按需印刷平台中的相似搜索研究

张明西，张雷洪，吕巍，孙刘杰
（上海理工大学，上海 200093）

摘要：目的 研究按需印刷平台中的相似搜索效率问题。**方法** 利用用户与产品之间的“购买”关系构建“用户–产品”关系，基于P–Rank提出一种高效的相似搜索方法POD–Rank，用于从“用户–产品”关系中发现相似产品。POD–Rank相似搜索过程依据“用户–产品”关系离线计算用户相似性，并利用用户相似性在线计算产品相似性，而后进一步提出优化的在线查询处理算法，以降低查询处理的时间开销。**结果** POD–Rank的计算时间开销和存储开销显著低于P–Rank，而且能够快速响应查询请求。**结论** POD–Rank的相似性计算开销为P–Rank的0.03%，存储开销为P–Rank的0.06%，计算效果与P–Rank接近，能够满足按需印刷平台中大规模产品数据处理的需求。

关键词： 按需印刷；P–Rank；相似搜索；“用户–产品”关系图
中图分类号： TS801.8　　**文献标识码：** A　　**文章编号：** 1001–3563(2015)23–0135–05

Similarity Search over Print–on–demand Platform

ZHANG Ming–xi, *ZHANG Lei–hong*, *LYU Wei*, *SUN Liu–jie*
(Shanghai University of Science and Technology, Shanghai 200093, China)

ABSTRACT: The aim of this work was to study the efficiency problem of similarity search over Print–On–Demand (POD) Platform. A "user–product" relation graph was built by utilizing the purchasing relationship between user and product, the similarity between products was measured according to the structure of "user–product" relation graph. For improving the efficiency, we proposed a similarity search method, POD–Rank, which divided the computation process into 2 steps. In the first step, we computed the similarity between users in an off–line manner; and in the second step, we computed the similarity between the query and each candidate product based on user similarity in an online manner. For further reducing the response time of on–line query processing, we proposed an optimized online query processing algorithm by skipping the unnecessary accumulation operations on zero–values. The space cost and pre–computation time cost of POD–Rank were evidently lower than those of P–Rank with little effectiveness loss and short online query time. By adopting the 2–step similarity computation method, the time cost was significantly reduced, the computation time cost was only 0.03% of that of P–Rank, the size of similarity matrix was only 0.06% of that of P–Rank, and the effectiveness was close to that of P–Rank. This method can therefore be efficiently applied to processing of large datasets of POP platform.

KEY WORDS: Print–On–Demand; P–Rank; similarity search; "user–product" relation graph

互联网和计算机技术已经广泛应用于印刷行业，并改变了传统印刷行业的生产模式[1—4]。按需印刷(Print–on–demand)是传统印刷行业与现代信息技术的产物，已经广泛应用于各大企业，如互动出版社、当当网等网络书店都采用了按需印刷的生产模式。具体地说，按需印刷是指按照用户的个性化印刷要求，

收稿日期： 2015–05–23
基金项目： 上海市教委科研创新项目(15ZZ074)；上海高校青年教师培养资助计划(ZZSLG14021)；上海出版传媒研究院招标课题(SAYB1410)；上海理工大学博士启动基金(1D–14–309–001)
作者简介： 张明西(1985—)，男，安徽亳州人，上海理工大学讲师，主要研究方向为Web信息检索、按需印刷、社会网络分析。

依指定的地点和时间直接将所需资料的文件数据进行数码印刷、装订。按需印刷注重按需出版，对于发行量较小的图书，特别是专业和学术性图书，按需印刷是一种最佳的出版方式，这种方式能有效节约生产成本。

数字资产管理平台是按需印刷平台构建的一个重要模块，主要用于存储和管理产品和用户数据，涉及范围包括产品数据链管理[1]、产品聚类分析[17]、产品相似搜索[5—6]等，其中相似搜索是数据管理的一个重要研究方向，在近几年得到了广泛关注和深入研究。在按需印刷平台中，相似搜索的主要任务是从海量数据中发现相似的产品，这为产品聚类分析、产品推荐、用户行为分析等实际应用提供重要依据。

现有相似搜索方法很多，比如SimRank[5]，P-Rank[6]，NetSim[7]等，其中P-Rank（Penetrating Rank）基于入链（In-link）和出链（Out-link）计算相似性，具有更高的精确度，能有效地从资产数据库中发现相似的产品。然而P-Rank的相似性计算的开销巨大，不适用于大规模数据的处理。现有相似搜索的优化方法[7—16]没有同时考虑入链和出链信息，并不适用于P-Rank，而且可能会忽略潜在相似的结果。

基于上述讨论，文中针对按需印刷平台中的相似搜索效率问题进行研究，在P-Rank研究工作基础上提出一种高效的按需印刷平台中的相似搜索方法POD-Rank。首先将P-Rank重写为矩阵形式，基于重写的P-Rank模型优化相似搜索过程：在离线阶段，依据“用户-产品”关系图计算用户相似性，结合用户相似性在线计算产品之间的相似性。在查询处理阶段，对查询处理过程进行优化，采用累加的思想提高相似性在线计算效率，并提出一种高效查询处理算法，以提高查询处理效率。

1 预备知识

1.1 “用户-产品”关系图

在按需印刷平台中，用户与印刷产品之间的“购买”行为形成了一种“购买”关系，所有用户和产品之间的这种“购买”关系就构成了“用户-产品”关系图（User-Product Graph），该图被定义为一个有向二分图$G=(V,E)$，其中$V=V_u\cup V_p$表示实体集合，V_u和V_p分别表示用户和产品类型的实体，E表示用户和产品之间的“购买”关系，有序对$(u,p)\in E$表示用户$u(u\in V_u)$和产品$p(p\in V_p)$的“购买”关系。

“用户-产品”关系的一个示例见图1，其中链接关系为用户和产品之间的“购买”关系，比如(u_1,p_1)，(u_1,p_3)等，在相同类型的实体之间不存在任何链接关系。

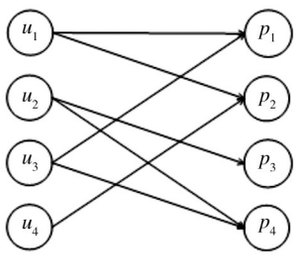

图1 用户与产品的关系

Fig.1 Examples of user-product relation

1.2 P-Rank模型

P-Rank主要思想可以概括为：如果2个实体都被相似的实体所引用，那么这2个实体是相似的；如果2个实体都引用了相似的实体，那么这2个实体也是相似的。P-Rank采用迭代的方法进行相似性计算，采用$R_t(a,b)$表示实体$a\in V$和$b\in V$在第t次迭代的相似性，计算步骤：

当$t=0$时，如果$a=b$，那么$R_0(a,b)=1$，否则$R_0(a,b)=0$；

当$t>0$时，如果$a=b$，那么$R_0(a,b)=1$，否则

$$R_t(a,b)=\frac{\lambda C}{|I(a)||I(b)|}\sum_{i=1}^{|I(a)|}\sum_{j=1}^{|I(b)|}R_{t-1}(I_i(a),I_j(b))+\frac{(1-\lambda)C}{|O(a)||O(b)|}\sum_{i=1}^{|O(a)||O(b)|}R_{t-1}(O_i(a),O_j(b)) \tag{1}$$

式中：C为阻尼因子，$C\in(0,1)$；λ为平衡因子，$\lambda\in(0,1)$，用于平衡入链和出链对相似性的影响，通常设置为0.5；$I(a)$为a的指入邻居（In-neighbor）的集合；$O(a)$为a的指出邻居（Out-neighbor）的集合。相似性计算的时间复杂度为$O(tn^2d)$，空间复杂度为$O(n^2)$，这里的n是图G中的实体规模，d表示图G中的平均度。

2 优化算法

2.1 P-Rank模型的重写

为了提高相似性计算效率，首先将P-Rank模型重写为矩阵形式，这时$\boldsymbol{R}_t$表示第t次迭代的相似性矩阵。

当$t=0$时，$\boldsymbol{R}_0(a,b)=\boldsymbol{I}_n$，这里的$\boldsymbol{I}_n$为单位矩阵；

当$t>0$时,$\boldsymbol{R}_t$计算公式为:

$$\boldsymbol{R}_t=(1-\lambda)C\boldsymbol{P}\boldsymbol{R}_{t-1}\boldsymbol{P}^{\mathrm{T}}+\lambda C\boldsymbol{P}^{\mathrm{T}}\boldsymbol{R}_{t-1}\boldsymbol{P}+(1-C)\boldsymbol{I}_n \quad (2)$$

式中:$\boldsymbol{P}$为概率转移矩阵,矩阵元素表示图G中实体之间的转移概率;$\boldsymbol{P}^{\mathrm{T}}$为$\boldsymbol{P}$的转置;$\boldsymbol{I}_n$为$n\times n$的单位矩阵。这时相似性计算的时间复杂度降低为$O(tn^2d)$,空间复杂度不变。尽管通过重写在一定程度上降低了计算开销,但是仍然难以满足实际需求。

2.2 POD-Rank模型

为了满足按需印刷平台中大规模产品数据处理的需求,文中基于P-Rank的基本思想提出一种高效的相似搜索方案POD-Rank,用于"用户-产品"关系图中产品相似性计算。POD-Rank相似性计算过程分为离线和在线2个阶段。在离线阶段,首先依据"用户-产品"关系图来计算用户之间的相似性,并不计算产品的相似性,这是因为在计算产品相似性时仅用到用户相似性计算结果,而没用到产品相似性的计算结果。这样可以节省计算产品相似性的计算开销。

对于给定的图G,产品相似性矩阵计算公式为:

$$\boldsymbol{R}_{\mathrm{u}}=(1-\lambda)C\boldsymbol{P}_{\mathrm{up}}\boldsymbol{P}_{\mathrm{up}}{}^{\mathrm{T}}+\lambda C\boldsymbol{P}_{\mathrm{up}}{}^{\mathrm{T}}\boldsymbol{P}_{\mathrm{up}}+(1-C)\boldsymbol{I}_{\mathrm{u}} \quad (3)$$

式中:$\boldsymbol{I}_{\mathrm{u}}$为$n_{\mathrm{u}}\times n_{\mathrm{u}}$的单位矩阵,这里的$n_{\mathrm{u}}$是用户规模;$\boldsymbol{P}_{\mathrm{up}}$为在图$G$中的用户到产品之间的转移概率矩阵;$\boldsymbol{P}_{\mathrm{up}}{}^{\mathrm{T}}$为$\boldsymbol{P}_{\mathrm{up}}$的转置。

在计算$\boldsymbol{P}_{\mathrm{up}}\boldsymbol{P}_{\mathrm{up}}{}^{\mathrm{T}}$时,对于每个行元素$u_1$,仅与图$G$中这样的$u_2=\{u_2|\forall p\in O(u_1):u_2\in I(p)\}$对应的列元素相乘;在计算$\boldsymbol{P}_{\mathrm{up}}{}^{\mathrm{T}}\boldsymbol{P}_{\mathrm{up}}$时,对于每个行元素$u_1$,仅与图$G$中这样的$u_2=\{u_2|\forall p\in I(u_1):u_2\in O(p)\}$对应的列元素相乘。这样就节省了多余的相似性为零时的计算操作,从而将在离线阶段相似性计算的时间复杂度降低为$O(nd^2)$,显著低于P-Rank的时间复杂度。

在查询处理阶段,依据用户相似性的计算结果来计算实体相似性。由等式(4)—(5)可以得到用户之间的相似性矩阵:

$$\boldsymbol{R}_{\mathrm{p}}=(1-\lambda)C\boldsymbol{P}_{\mathrm{pu}}{}^{\mathrm{T}}\boldsymbol{R}_{\mathrm{u}}\boldsymbol{P}_{\mathrm{pu}}+\lambda C\boldsymbol{P}_{\mathrm{pu}}\boldsymbol{R}_{\mathrm{u}}\boldsymbol{P}_{\mathrm{pu}}{}^{\mathrm{T}}+(1-C)\boldsymbol{I}_{\mathrm{p}} \quad (4)$$

式中:$\boldsymbol{I}_{\mathrm{p}}$为$n_{\mathrm{p}}\times n_{\mathrm{p}}$的单位矩阵,这里的$n_{\mathrm{p}}$是产品规模;$\boldsymbol{P}_{\mathrm{pu}}$为在图$G$中的产品到用户的转移概率矩阵;$\boldsymbol{P}_{\mathrm{pu}}{}^{\mathrm{T}}$为$\boldsymbol{P}_{\mathrm{pu}}$的转置。这时,产品$p_1\in V_{\mathrm{p}}$和$p_2\in V_{\mathrm{p}}$的相似性为:

$$\boldsymbol{R}_{\mathrm{p}}(p_1,p_2)=(1-\lambda)C\boldsymbol{P}(p_1)\boldsymbol{R}_{\mathrm{u}}\boldsymbol{P}^{\mathrm{T}}(p_2)+\lambda C\boldsymbol{P}^{\mathrm{T}}(p_1)\boldsymbol{R}_{\mathrm{u}}\boldsymbol{P}(p_2)+(1-C) \quad (5)$$

式中:$\boldsymbol{P}(p_1)$和$\boldsymbol{P}(p_2)$为产品p_1和p_2在概率转移矩阵$\boldsymbol{P}$中对应的行向量;$\boldsymbol{P}^{\mathrm{T}}(p_1)$和$\boldsymbol{P}^{\mathrm{T}}(p_2)$为产品$p_1$和$p_2$在概率转移矩阵$\boldsymbol{P}$的转置矩阵中对应的行向量。

POD-Rank在线查询处理的直接方法为:给定查询$q\in V_{\mathrm{p}}$,依据公式(5)计算q与所有候选产品$p_i\in V_{\mathrm{p}}$的相似性,然后返回最相似的k个产品。然而,这种方法在计算过程中涉及了多余的零值累加操作,这样就会显著增加相似搜索查询响应的时间开销。

为了进一步提高查询处理效率,下面提出一种优化的在线查询算法,见算法1。算法输出为"用户-产品"关系图$G(V,E)$、矩阵$\boldsymbol{R}_{\mathrm{u}}$、查询$q$和参数$k$,算法输出为与查询$q$最相关的top-$k$个产品序列。算法将累加过程中到零值累加操作进行了剪枝,仅考虑R_1中的非零元素,并将这样的元素累加到相似性计算结果中,通过这种方法节省了大量的时间开销,显著提高了查询效率。

算法1 优化的在线查询处理算法:

Input: Graph G(V, E), matrix R_u, query q and parameter k;

Output: Top-k most similar sorted products;

1: for all x∈I(q) do

2: for all y∈{y|R_u(x,y)>0} do

3: for all p∈O(y) do

4: $R_p(q,p)\leftarrow R_p(q,p)+\frac{\lambda C}{|I(q)||I(p)|}R_u(x,y)$;

5: end for

6: end for

7: end for

8: for all x∈O(q) do

9: for all y∈{y|R_u(x,y)>0} do

10: for all p∈I(y) do

11: $R_p(q,p)\leftarrow R_p(q,p)+\frac{(1-\lambda)C}{|O(q)||O(p)|}R_u(x,y)$;

12: end for

13: end for

14: end for

15: Sort products according to similarities and output;

3 结果

3.1 设置

实验机器配置为CPU为Intel(R) Xeon(R),频率

为2.27 GHz和2.26 GHz(双处理器),内存为16.0 GB,操作系统为Windows Server 2008 R2 Enterprise,开发环境为VS C++2010。实验在亚马逊数据真实集(http://snap.stanford.edu/data/amazon-meta.html)上进行。从该数据集中选择有代表性的6521个实体构建"用户-产品"关系图,包含2502个用户、4019本图书、15 251条"购买"关系。

实验将论文提出的方法(POD-Rank)及其优化方法(Opt-POD-Rank)与P-Rank进行对比。采用NDCG(Normalized Discounted Cumulative Gain)[18]评估实验效果。对于给定查询实体,NDCG在第k个位置的值(NDCG@k)根据实体在结果序列中的位置以及他们的相似性评估等级来计算,采用P-Rank在第10次迭代的相似性值为打分基准。

实验效率评估方面主要对比离线计算的时间和存储开销以及在线查询处理的时间开销进行评估。其中,存储开销方面主要是对比相似性矩阵中的非零元素规模。

3.2 效率测试

在亚马逊数据集上的测试结果显示,P-Rank相似性计算时间开销为19 013 s,而POD-Rank仅5 s,POD-Rank的计算时间显著低于P-Rank,仅为P-Rank的0.03%,这是因为P-Rank需要进行多次迭代,而POD-Rank无需进行迭代计算,同时在计算过程中节省了多余的零值累加操作。

在相似性矩阵规模方面的测试结果显示,P-Rank相似性矩阵的非零元素数为16 192 570,而POD-Rank的非零元素数为897 362,显著低于P-Rank,仅为P-Rank的0.06%,这是因为现实网络比较稀疏,用户相似性矩阵的非零元素占的比例并不是很大,而P-Rank相似性矩阵经过多次迭代后几乎达到满阵。

k为10,20,30,40,50时,POD-Rank的查询处理时间分别为369.173,371.691,391.125,403.980,409.109 ms,Opt-POD-Rank的查询处理时间分别为20.911,20.274,22.131,24.091,24.742 ms。可知POD-Rank和Opt-POD-Rank在线查询时间开销随着k取值增加而增加,但增加趋势比较缓慢。POD-Rank查询时间在360~410 ms之间,Opt-POD-Rank将查询时间降低为20~25 ms之间,仅为POD-Rank的5%左右。

上述实验结果表明,POD-Rank显著降低了相似搜索的计算开销,查询响应的时间开销较低,能够满足按需印刷平台中大规模数据处理的需求。

3.3 效果测试

k为10,20,30,40,50时,POD-Rank的NDCG值分别为0.999 151,0.998 549,0.998 591,0.998 998,0.999 613,P-Rank的NDCG值都为1。可知P-Rank的NDCG值不随取值变化而变化,始终为1,POD-Rank的NDCG值稍低,但非常接近1。当k取值增加时,NDCG值呈现出逐渐减小的趋势,这是因为返回结果中排序在第1的结果是查询本身,所以在k为1时,NDCG值也为1,而之后结果的并不是理想序列。结果表明,POD-Rank返回结果的效果与P-Rank是非常接近的。

表1给出POD-Rank在Amazon数据集上关于3个不同查询的返回结果的top-5个结果。其中,查询"谁写的圣经?"是一个关于圣经的图书,该查询对应的返回结果排序中,前5个排序位置对应的结果显然都与宗教相关。比如,排序1的结果是查询本身,排序2的"诗篇的神学导论"是关于神学导论的图书,排序3的"出乎意料的智慧:旧约全书的主要启示"是关于旧约全书的图书,排序4的"传道书之雅歌篇"是一本传道方面的图书,排序5的"何西阿书,阿莫斯,弥迦书"也是一本与圣经相关的图书。查询"现实生活中的Linux安全:入侵防御、探测和恢复"是一本关于信息安全与防范方面的图书。显然,排序1,2,3,4,5都是与信息安全相关的图书。另外,查询"化学工程热力学导论"返回结果也可以类似的分析,这里就不做赘述。通过实例分析可知,POD-Rank在现实数据中应用时,返回的产品都是与查询相似的,能够从大规模的数据中发现相似的图书产品。

表1 关于不同查询的返回结果

Tab. 1 Returned result on different queries

排序	谁写的圣经?	现实生活中的Linux安全:入侵防御、探测和恢复	化学工程热力学导论
1	谁写的圣经?	现实生活中的Linux安全:入侵防御、探测和恢复	化学工程热力学导论
2	诗篇的神学导论	安全指南完整版	热力学原理及工程技术应用
3	出乎意料的智慧:旧约全书的主要启示	红帽Linux系统安全与优化	统计力学与热力学导论
4	传道书之雅歌篇	网络安全	对流热交换(第2版)
5	何西阿书,阿莫斯,弥迦书	Linux 系统安全:保护您的服务器和工作站的黑客指南	热电技术手册

4 结语

在按需印刷平台中的相似搜索效率方面进行研究,提出了一种高效的相似搜索方法POD-Rank。与传统相似搜索方法相比,POD-Rank相似性计算并不需要进行多次迭代,时间开销和存储开销都比较低。在查询处理阶段进行优化,节省了零值累加操作,降低了查询响应时间,返回结果也具有较高的精确度。

未来研究工作将围绕按需印刷平台中的相似性增量式计算展开,着重研究相似性的动态计算方法以及相似性矩阵的局部更新问题。另外,课题还将研究相似搜索在印刷产品聚类分析、印刷产品推荐等方面的应用。

参考文献:

[1] 皮阳雪,何永志. 一种基于数据链管理的印刷ERP解决方案[J]. 包装工程,2014,35(11):122—127.
PI Yang-xue,HE Yong-zhi. A Solution of Printing ERP Based on Data Chain Management[J]. Packaging Engineering,2014,35(11):122—127.

[2] 丁毅,苏杰. 论ERP系统和EXCEL在包装企业管理中的综合应用[J]. 包装工程,2012,33(11):135—137.
DING Yi,SU Jie. Application of ERP System and EXCEL in Packaging Enterprise Management[J]. Packaging Engineering,2012,33(11):135—137.

[3] 孔玲君,姜中敏,高雪玲. 数字印刷品非均匀性的综合评价[J]. 包装工程,2014,35(1):114—119.
KONG Ling-jun,JIANG Zhong-min,GAO Xue-ling. Comprehensive Evaluation of Non-uniformity of Digital Prints[J]. Packaging Engineering,2014,35(1):114—119.

[4] 朱佳. 基于BPMN的按需印刷流程的设计与实现[D]. 北京:北京印刷学院,2013.
ZHU Jia. The Design and Application of Printing on Demand Process Based on BPMN[D]. Beijing: Beijing Institute of Graphic Communication,2013.

[5] JEH G,WIDOM J. Simrank:A Measure of Structural-context Similarity[C]// In Proceedings of KDD,2002:538—543.

[6] ZHAO P,HAN J,SUN Y. P-rank:A Comprehensive Structural Similarity Measure over Information Networks[C]// In Proceedings of CIKM,2009:553—562.

[7] ZHANG M,HU H,HE Z,et al. Top-*k* Similarity Search in Heterogeneous Information Networks with X-star Network Schema[J]. Expert Systems with Applications,2015,42(2):699—712.

[8] LI C,HAN J,HE G. Fast Computation of SimRank for Static and Dynamic Information Networks[C]// In Proceedings of EDBT,2010.

[9] ZHAO X,XIAO C,LIN C,et al. A Partition-based Approach to Structure Similarity Search[J]. VLDB,2013,7(3):169—180.

[10] LI P,LIU H,XU J,et al. Fast Single-pair Simrank Computation[C]// In Proceedings of SDM,2010:571—582.

[11] CAI Y,CONG G,JIA X,et al. Efficient Algorithms for Computing Link Based Similarity in Real World Networks[C]// In Proceedings of ICDM,2009:734—739.

[12] SUN Y,YU Y,HAN J. Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema[C]// In Proceedings of KDD,2009:797—806.

[13] LEE P,LAKS V S,JEFFREY X Y. On Top-*k* Structural Similarity Search[C]// In Proceedings of ICDE,2012:774—785.

[14] CAI Y,LIU H,HE J,et al. An Adaptive Method for Efficient Similarity Calculation[C]// In Proc of DASFAA,2009:339—353.

[15] SUN Y,HAN J,YAN X,et al. Pathsim:Meta Path-based Top-*k* Similarity Search in Heterogeneous Information Networks[C]// In Proceedings of VLDB,2011:992—1003.

[16] ZHANG M,HE Z,HU H,et al. E-rank:A Structural-based Similarity Measure in Social Networks[C]// In WI,2012:415—422.

[17] 张树勋,朱霞,陈俊斌,等. 基于模糊聚类分析的军用附属油料包装规格优化研究[J]. 包装工程,2014,35(11):28—32.
ZHANG Shu-xun,ZHU Xia,CHEN Jun-bin,et al. Optimization of Packaging Specifications for Military Auxiliary Oil Based on Fuzzy Clustering Analysis[J]. Packaging Engineering,2014,35(11):28—32.

[18] ARVELIN K,KEKALAINEN J. Cumulated Gain-based Evaluation of It Techniques[J]. ACM Transactions on Information Systems,2002,20:422—446.