基于偏好的二分图网络模型 Top-N 推荐①

陈添辉, 林世平, 郭 昆, 廖寿福

(福州大学 数学与计算机科学学院学院, 福州 350108)

摘 要: 针对推断网络(NBI)的二分图方法中只是考虑用户是否评价过项目, 却没有利用用户评分高低这一局限性, 提出基于偏好的推断网络(PNBI)推荐方法. 该方法在推断网络的基础上, 考虑单个用户对项目评分高低体现了该用户对项目的喜好程度, 在“用户-项目”的资源分配过程中, 将资源分配给评分值较大的评分项, 该方法能克服NBI算法中无法使用低评分值数据的缺陷. 考虑到数据的稀疏性问题, 采用倒排表的方法来节省相似度的运算次数, 加速算法. 在 MovieLens 数据集上的实验表明, PNBI 二分图推荐算法在准确率、覆盖率和召回率三个方面均优于 NBI 二分图推荐算法.

关键词: 偏好; 二分图; 推荐算法; 倒排表

Top-N Recommendation Based on Preference Bipartite Network

CHEN Tian-Hui, LIN Shi-Ping, GUO Kun, LIAO Shou-Fu

(College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China)

Abstract: In view of the bipartite graph method of network-based inference (NBI) only considered whether users evaluated the project or not, but not given their scores, the thesis proposed a preferential network-based inference (PNBI) recommended method. Based on inference network, the method takes into account that user's rating values for the program reflects his degree of preference. In the "User-Item" resource allocation process, the method allocates resources to the item that gets a higher score, this method can overcome the NBI algorithm's disadvantage of failing to use low score value. Considering the sparsity of data, the method uses inverted list to discrese the number of calculation to accelerate the algorithm. Experiments on MovieLens dataset show that, PNBI bipartite graph recommended algorithm outperforms NBI bipartite graph recommended algorithm in accuracy, coverage and recall.

Key words: preference; bipartite graph; recommendation algorithm; inverted

随着互联网的发展, 信息呈爆炸性的增长. 从大量的可选信息中选择出有用信息对于许多用户来说是困难的, 许多优秀的信息也在大量无效的数据下被掩盖. 如何挖掘出大量数据中有用的信息就变得非常有现实意义, 个性化推荐等一系列技术就是在这样的背景下产生的. 推荐系统通过对用户历史行为数据记录进行分析, 从中挖掘出一些个性化信息, 并通过这些个性化信息来给用户推荐相关产品.

推荐系统主要可以分为: 协同过滤(Collaborative Filter)系统[1,2], 基于内容(content-based)系统[3,4], 混合(hybird)系统[5]和基于图模型(graph-based)系统[6,7]. 由于用户的行为数据可以用二分图来表示, 因此很多基于图的算法可以应用在基于图模型的推荐系统上[8].

Zhou Tao 等[9]提出的NBI(network-based inference)推荐方法, 该方法将计算项目-项目之间关系转为“从项目到用户”和“从用户到项目”两次的资源分配问题. 由于二分图模型对数据的处理是将其二元化(为 0 或者为 1), 这就忽略了用户对项目评分值高低分的影响.

① 基金项目:国家自然科学基金(61300104)

收稿时间:2014-08-12;收到修改稿时间:2014-09-16

© 中国科学院软件研究所 http://www.c-s-a.org.cn

本文提出一种基于用户偏好的二分图资源分配方法, 实验表明 PNBI 在准确率, 召回率和覆盖率等方面均优于 NBI 算法.

文章第 2 部分介绍算法模型和算法具体实现, 第 3 部分对实验结果进行分析, 第 4 部分总结.

1 二分图算法与改进

近年来, 基于网络图模型的推荐方法越来越受到关注[10], 常见的模型主要是二分图. 典型的二分图如图 1(a)所示, 对于一个二分图 $G(V,E)$, 所有顶点 V 被分为集合 X 和集合 Y 两个顶点集, 集合 X 和集合 Y 必须满足 $X\cap Y=\varnothing$ 且 $X\cup Y=V$. 对于任意的边 $e_{ij}\in E$, 顶点 i 和顶点 j 必须属于不同的顶点集.

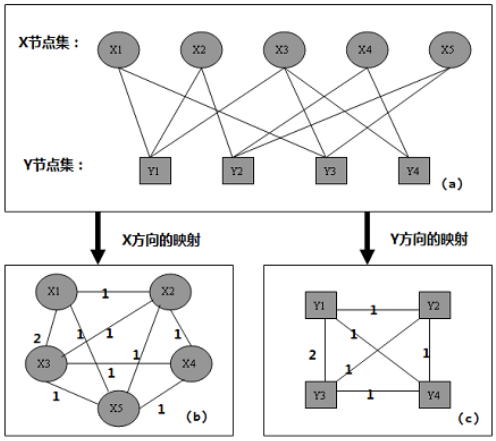

图 1 二分图基本模型

二分图经过映射后, 得出的是各个顶点集内部节点和节点之间的关系, 如图 1(b)和 1(c)所示.

1.1 二分图与推荐算法

对于一个由 N 个用户和 M 个项目组成的推荐系统, 定义用户为一个集合 $U=\{u_1,u_2,\cdots,u_N\}$, 项目为一个集合 $I=\{i_1,i_2,\cdots i_M\}$, 记用户-项目的评分矩阵为 $R=(r_{u,i})_{N\times M}$, $r_{u,i}$ 表示用户 u 对项目 i 的评分. 把 X 节点集替换为用户集合 U, Y 节点集替换为项目节点集 I, 因此就可以将该推荐系统模型运用于二分图中.

记节点 i 到节点 j 的权重为 w_{ij}, 如何表示边与边之间的关系呢? Zhou 等在文献[9]中提出一种全新的资源分配方案: 假设原始项目集 I 中的资源为 $\{x,y,z\}$ (如图 2(a)所示), 资源分配分为两步: (1)资源平均地从项目集 I 流向用户集 U , 如图 2(b)所示; (2)资源再从用户集 U 平均地流回项目集 I , 分配过程如图 2(c)所示. 假设分配后的资源为 $\{x',y',z'\}$, 则由图可得到公式(1):

$$\begin{pmatrix}x'\\y'\\z'\end{pmatrix}=\begin{pmatrix}11/18 & 1/6 & 5/18\\ 1/9 & 5/12 & 5/18\\ 5/18 & 5/12 & 4/9\end{pmatrix}\begin{pmatrix}x\\y\\z\end{pmatrix} \tag{1}$$

假设公式(1)中的 3×3 矩阵为资源分配矩阵 $W=(w_{jk})$, 则对于一个二分图 $G(X,Y,E)$, 经过分配后的资源权重 w_{jk} 如公式 2.

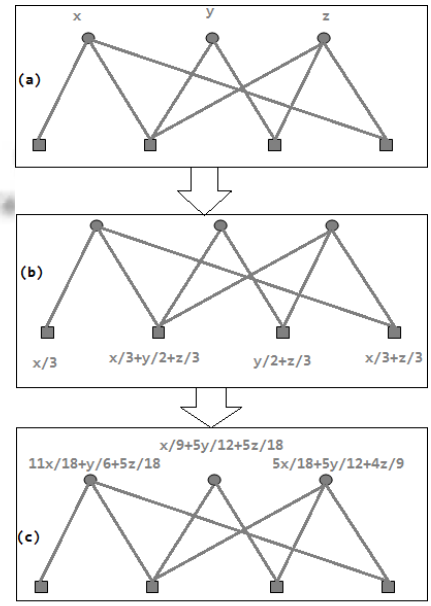

图 2 NBI 资源分配

$$w_{jk}=\frac{1}{d(i_j)}\sum_{l=1}^{m}\frac{a_{jl}a_{kl}}{d(u_l)} \tag{2}$$

公式 2 中, $d(x)$ 表示为 x 顶点的度, $A_{(n\times m)}$ 为邻接矩阵, 且

$$a_{jl}=\begin{cases}1, & i_ju_l\in E,\\ 0, & \text{其他}\end{cases} \tag{3}$$

1.2 改进的二分图算法

在许多评分系统中, 往往是使用一个分值来度量用户对某一物品的喜好程度的, 典型的例子是 Yahoo 音乐推荐系统, 该系统允许用户的评分范围为 1-5 分, 5 分代表“非常喜欢”, 4 分代表“喜欢”, 3 分代表“一般”, 2 分代表“不喜欢”, 1 分代表“非常不喜欢”. 对于单个用户来说, 评分的高低往往代表着用户对物品的喜好程度[11,12], 本文基于这个问题, 利用用户的正反馈偏好来改进原始的二分图算法.

定义 1. 图 1(b)的过程为第一次资源分配, 图 1(c)

© 中国科学院软件研究所 http://www.c-s-a.org.cn

的过程为第二次资源分配.

经过第一次分配, 用户节点集 U 中存在着从 I 节点集分配过来的资源, 对于原始的算法, 在第二次资源分配时, 直接平均地再分配回 I 节点集. 考虑到用户对物品评分的高低反应了用户对物品的喜好程度, 在第二次资源分配过程中, 对于某一个用户, 仅将资源分配给评分值大于等于该项目的资源. 也就是说, 如果用户 u 评价了项目 i,k , 当且仅当 $r_{ui}>=r_{uk}$ 时, 用户 u 对物品 k 的评分在第二次资源分配的过程中才能将资源分配给项目 i . 由于在原始的二分图模型中添加了用户偏好, 故称这种二分图模型为基于偏好的推断网络(PNBI)的二分图方法. 第二次资源分配原则如图 3 所示.

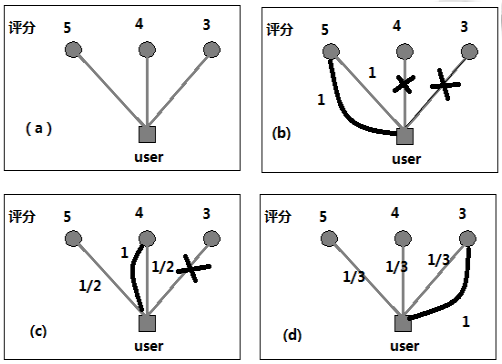

图 3　PNBI 第二次资源分配图

假设用户 $user$ 分别对三个项目编号为 x,y,z 做的评分为 5 分, 4 分和 3 分, 如图 3(a)所示. 对于项目 y 来说, 假设第一次分配给 $user$ 的资源为 1, 在做第二次资源分配时, 由于评分大于等于 4 分的只有两个项目 x,y , 故该资源在回流时就不考虑项目 z , 如图 3(c)所示. 图 3(b)和图 3(d)以此类推.

经过资源分配, 得到资源分配矩阵 W 后, 可以计算出每一个项目最后获得的资源大小, 如公式(4):

$$f' = W \cdot f \tag{4}$$

对于任一用户, 将该用户还未评分过的项目按 f' 降序排序, 那么前 N 个项目就是 Top-N 个推荐项目.

1.3 算法描述

NBI 算法在进行资源分配时的时间复杂度为 $O(m\times m)$, 当项目数 m 很大时, 算法运行速度很慢. 考虑到数据的稀疏性, 即存在大量 $a_{ji}a_{kl}=0$ 的无效计算, 故本次算法使用倒排表[8]来减少运算次数, 加速算法. 倒排表的思想是仅计算 $a_{ji}a_{kl}\neq 0$ 的情况.

推荐流程主要为两个算法, 算法 1 为计算“项目-项目”之间的权重矩阵, 算法 2 是对每个用户进行项目推荐.

算法 1. 计算资源分配矩阵.

输入: 格式为(用户 id, 项目 id, 评分)的训练集 TRAIN, 近邻个数 K 的取值

输出: 资源分配矩阵 W

Foreach data in TRAIN

1) 生成每个用户 i 评价过的项目 id, 记 $user(i)$
2) 生成每个项目 j 被哪些 id 的用户评价过, 记 $item(j)$
3) 记录评分矩阵 $R=(r_{ij})$. 表示用户 i 对项目的评分 j
4) Endfor
5) 记 $item(j)$ 的大小为 $d(j)$
6) Foreach u in $user$
7) Foreach i in $user(u)$
8) $lq\_item(i)$.empty()
9) Foreach l in $user(u)$
10) If $r_{ul}\geq r_{ui}$ then
11) $lq\_item(i).insert(l)$
12) Endif
13) Endfor
14) Foreach l in $lq\_item(i)$
15) $w_{ui}=w_{ui}+1/(d(i)\bullet lq\_item(i).size())$
16) Endfor
17) EndFor
18) Endfor

算法 2. 为用户做 top-n 推荐.

输入: 资源分配矩阵 W , 推荐个数 N 的取值

输出: 推荐列表 RecList

1) Foreach $i=1:M$
2) $nkid=top\_k\_id(w_i)$
3) Endfor
4) Foreach $i=1:N$
5) $\hat{r}_i=0$
6) Foreach $item1$ in $user(i)$
7) Foreach $item2$ in $nkid$
8) $\hat{r}_{i,item2}=\hat{r}_{i,item2}+w[item1][item2]$
9) Endfor
10) Endfor
11) $RecList=top\_n(\hat{r}_i)$

© 中国科学院软件研究所 http://www.c-s-a.org.cn

12) Endfor

其中, $top\_k\_id(w)$ 表示得到前 k 个最大数的编号, 在算法中表示得到前 k 个最相似的项目.

1.4 算法复杂度

算法 1 中 1-5 步的时间复杂度为 $O(t)$, t 为训练集数据长度. 算法 1 的 7-19 步的复杂度是由每个用户评分项目的长度决定的, 假设平均长度为 t/n, n 为用户数目, 则时间复杂度为 $O(t^2/n)$. 易知 t/n>1, 故算法 1 的时间复杂度为 $O(t^2/n)$. 算法 2 中 6-10 步的复杂度为 $O(tk/n)$, 故算法 2 的时间复杂度为 $O(tk)$. 其中, k 为最近邻居数量.

2 实验分析

实验数据采用明尼苏达大学(University of Minnesota)GroupLens 团队的 MovieLens 数据集(数据集详细信息见 http://www.grouplens.org/node/73)来对算法进行验证. 该数据集包含 943 个用户和 1682 部电影的 10 万条评分记录, 用 1-5 分来分别表示 5 个不同的等级, 某一用户对某一部电影的评分值越高说明越喜欢该电影. 本次实验中, 将数据集划分为训练集和测试集两部分, 其中 80%作为训练集, 20%作为测试集. 对于测试集中的实验数据, 本实验假定评分大于等于 4 分的电影才是用户真正喜欢的电影. 即在评估结果时, 推荐的电影必须出现在测试集中, 并且用户对其评分值不低于 4 分, 该电影才认为命中.

本次实验选取用户没有挑选过的最热门的前 10 个作为推荐结果. 由于邻居个数 K 值对算法的准确率和覆盖率都会产生影响[8], 故实验对比 K 的不同取值下, 准确率和覆盖率的结果, 分别如图 4、图 5 所示.

从图 4 可以看出, 准确率并不是和邻居个数 K 成正比的, 考虑极端情况, 当 K 取值为 0 时, 算法退化为随机推荐, 准确率明显较低; 当 K 的取值非常大时, 算法挖掘的关联信息基本用完, 所以准确率趋于一个稳定. 从图 4 知, 当 K 的值在 5-100 之间时, 准确率增加的非常快速. 从图 5 覆盖率的曲线趋势发现, 随着 K 值的增大, 算法的覆盖率首先急剧上升, 然后急剧下降, 最终收敛. 这是因为当考虑较小近邻数时, 较冷门的物品由于没有受到热门物品的权重影响, 算法的 Top-N 列表中更倾向于推荐个性化的物品, 取得了推荐的多样性, 但是当 K 大于一定阈值后, 大量热门物品的加入使得算法趋近于热门推荐. 对比图 4 中两条曲线可知, PNBI 算法(菱形红色)的准确率在不同的 K 值下效果均优于原始的 NBI 算法(方形蓝色). 对比图 5 中两条曲线, PNBI 算法在覆盖率方面也略优于 NBI 算法, 该结果表明 PNBI 算法在挖掘用户有效信息方面优于 NBI 算法.

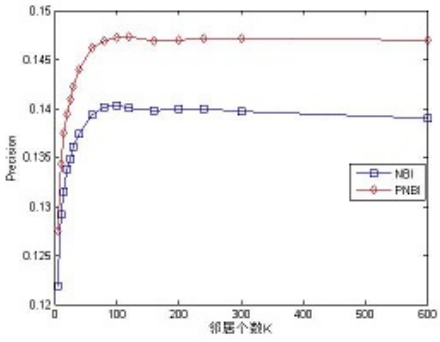

图 4 PNBI 和 NBI 准确率对比图

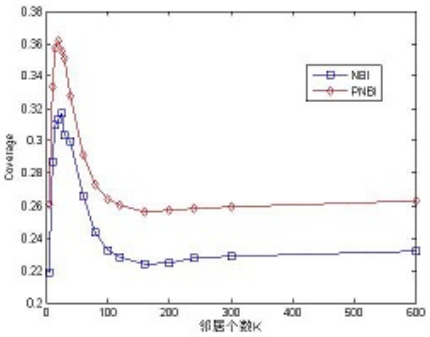

图 5 PNBI 和 NBI 覆盖率对比图

综合图 4, 图 5 可得, 推荐算法在使用少量的邻居个数的情况下, 就能达到很好的推荐效果. 这也说明了用户行为数据具有相关性相关联系, 这也证明了推荐算法的有效性.

表 1 算法运行时间对比图

没有使用倒排表的 NBI	使用倒排表的 PNBI 算法
46.0525s	18.6686s

表 1 显示了在 Windows7 64-bit, 4G 内存, Visual Stdio2012 环境下没有使用倒排技术和使用了倒排技术 NBI 算法的平均运行时间对比, 仿真实验结果表明, 使用倒排技术能有效加速算法.

3 结论

针对二分图算法没有对用户数据评分做区分和丢弃较低评分值等缺陷, 提出基于用户偏好的推荐算法, 仿真实验表明, 算法在准确率、覆盖率等均更优. 算法

© 中国科学院软件研究所 http://www.c-s-a.org.cn

在计算过程中, 考虑到评分矩阵稀疏性的特性, 采用倒排表的方法实现算法, 加速了算法的运行.

算法在推荐精度上还有待提高的地方, 当用户的评分信息没有在训练集中出现, 或者在训练集中没有太多共同评分项时, 推荐效果较差的, 因为此时算法退化成物品冷启动问题, 该问题除了使用人工进行分类标记外, 暂时并没有太多有效的办法, 并且由于本次算法仅仅使用的是用户的评分数据, 故对于物品冷启动是无能为力的. 算法的初始权重分配方式导致算法倾向于推荐热门的电影, 接下来拟采用降低热门电影初始权重的方式来改进这一缺点.

参考文献

1 Goldberg D, Nichols D, Oki BM, et al. Using collaborative filtering to weave an information tapestry. Communications of the ACM, 1992, 35(12): 61–70.

2 Resnick P, Iacovou N, Suchak M, et al. GroupLens: An open architecture for collaborative filtering of netnews. Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work. 1994. 175–186.

3 Belkin NJ, Croft WB. Information filtering and information retrieval: Two sides of the same coin? Communications of the ACM, 1992, 35(12): 29–38.

4 Balabanović M, Shoham Y. Fab: Content-based, collaborative recommendation. Communications of the ACM, 1997, 40(3): 66–72.

5 Ungar LH, Foster DP. Clustering methods for collaborative filtering. AAAI Workshop on Recommendation Systems. 1998.

6 刘建国,周涛,汪秉宏.个性化推荐系统的研究进展.自然科学进展,2009,19(1):1–15.

7 Breese JS, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence. 1998. 43–52.

8 项亮.推荐系统实践.北京:人民邮电出版社,2012.

9 Zhou T, REN J, Medo M, et al. Bipartite network projection and personal recommendation. Physical Review E, 2007, 76(4): 046115.

10 Shi L. Trading-off among accuracy, similarity, diversity, and long-tail: A graph-based recommendation approach. Proc. of the 7th ACM Conference on Recommender Systems. 2013. 57–64.

11 Aiolli F. Efficient top-n recommendation for very large scale binary rated datasets. Proc. of the 7th ACM Conference on Recommender Systems. ACM, 2013. 273–280.

12 Jin R, Si L, Zhai C. Preference-based graphic models for collaborative filtering. Proc. of the Nineteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 2002.

© 中国科学院软件研究所 http://www.c-s-a.org.cn