# 基于节点结构相似和排序的协同过滤算法[1]

刘淇，陈恩红
中国科学技术大学计算机科学与技术系，合肥 (230027)
E-mail: cheneh@ustc.edu.cn

**摘　要:** 本文提出一类基于图中的节点结构相似与带重启动的随机游走相结合的协同过滤推荐算法。该方法不仅具有较高的准确度，且具有良好的可扩展性。另外，该推荐算法可以在所有的用户和项之间产生相似排序，从而避免低覆盖率造成的推荐不准确。文中提出了两类不同的实现方法，分别是基于项协同过滤的项排序算法和基于用户协同过滤的用户排序算法，通过在一个标准数据集 MovieLens 上的测试表明了算法的有效性。

**关键词:** 推荐算法; 协同过滤; 结构相似; 排序; 随机游走
**中图分类号:** TP31

## 1. 引言

推荐系统根据用户与系统的交互历史以及用户的个人信息等构建用户的兴趣模型，预测用户可能感兴趣的产品或项。推荐系统大致可以分为三类[1]：基于协同过滤的方法(collaborative filtering)[2][4]，基于内容的方法(content-based filitering)[5][8]和将二者结合的混合推荐算法(hybrid recommendation)[6][7]。其中，基于协同过滤（CF）的方法在保持较好推荐效果的同时，其实现和维护代价都比较低，因而得到了最广泛的应用。协同过滤算法又分为 Memory-based 和 Model-based 两类[1]。

Model-based 算法首先利用已有的用户评价数据建立一个模型，然后根据该模型进行评价预测，例如基于贝叶斯网络的方法[11]，基于最大熵的方法[10]。但通常 Model-based 方法的模型建立和更新非常耗时，且模型经常不能像 Memory-based 方法一样覆盖所有的用户。Memory-based 算法是一种启发式的方法，它根据用户以往的所有评价去推荐。该方法一般是先为每个用户寻找相似的用户，再根据相似用户对给定项的评分以及用户相似度对项进行排序[1]。

近来，也提出了一些基于图中顶点相似计算的方法来进行协同推荐[15][17][19]。在其它领域中，[12] 总结和比较了大量的计算图中结点相似的方法用来进行连边预测。[13]也提出了一种新的度量网络中顶点结构相似的方法。

在图结构中，经常用随机游走（Random walk）对节点间的相似度进行衡量[15][16][19]，该方法以两个节点间的平均首达时间（ACT,average commute time）为标准。但 ACT 的问题在于它对图中远离节点 i,j 的部分有很强的依赖，即使节点是紧密相连的时候也是一样，所以经常会造成计算得到的相似度与实际节点的相似度有较大偏差。作为抵消该依赖的一种方法，可以让节点 i 到节点 j 的游走周期性“重启动”，即每一步以一定的概率 c 返回 i 重新走步，这样几乎不会走到图中偏远的部分。随机重启也是进行网页排序的 PageRank 算法[14]的基本思想。[18]用带重启动的随机游走(RWR，random walk with restart)来发现已知多媒体对象中各个媒体属性间的关联，而[21]则用 RWR 来发现情感与音乐属性的关联从而进行音乐推荐。类似地，[17]也将 PageRank 算法的思想用到推荐系统中。

实际应用中，推荐算法不仅要求较高的预测准确度，还要具备良好的扩展性，但是目前

[1]本课题得到国家教育部基于教育部博士点基金项目“中国科技论文在线”模式的科技论文网络发表平台的个性化服务研究（2007105）的资助。

的推荐算法都有其不足之处。例如，基于聚类的方法往往不能有效地覆盖所有的项或者用户，而基于图的方法虽然可以为所有的用户-项对产生相似排序，但往往消耗大量的空间资源，可扩展性较差，而且其推荐准确度也有待提高[15][19]。本文提出一种基于图中节点相似和排序的协同过滤算法 SSR(Structural Similarity and Ranking)。首先，根据结构相似寻找图中节点间的初步相似度，再利用 RWR 对图中的节点进行排序，从而产生推荐序列。SSR 算法实现过程中考虑了几种典型的节点结构相似方法，并比较了基于项和基于用户的协同过滤，即分别将项作为图中节点为给定用户对所有项进行排序的算法，以及将用户作为图中节点为给定项进行用户排序的算法。由于图中的结点表示项或者用户，从而有效地缩小了图的规模，降低空间消耗，使得算法具有良好的可扩展性。在用于推荐的通用数据集上的实验结果表明，该方法可以得到更高的预测准确度。

本文接下来将首先介绍 SSR 算法模型；在第 3 节，将对 SSR 算法与其他方法进行对比实验，并对实验结果进行分析；最后对全文进行总结。

## 2. 算法描述

协同过滤是为给定用户推荐其可能感兴趣的项，但也可以将给定的项推荐给可能对其感兴趣的用户。对于前者，SSR 将建立由项组成的关联图，然后用给定用户的用户向量在相应的项关联矩阵上进行 RWR，从而得到所有项的排序，排名靠前即得分较高的项将被推荐给用户(Item_Rank，因为实验中使用的 item 为电影，所以又称 Movie_Rank)。而对于后者，SSR 将建立用户组成的关联图，然后用给定项的项向量在相应的用户关联矩阵上进行 RWR，得到所有用户的排序，该项将被推荐给得分较高的用户(User_Rank)。本文，I 表示项的集合，U 表示用户的集合；n，m 分别表示项和用户的个数。

### 2.1 关联矩阵

将用户与项之间的关系表示为一个二部图 G=<X，E>，其中顶点集 X=U∪I，用户 $U_p$ 如果对某项 $I_q$ 表现出兴趣，则为 $U_p$ 与 $I_q$ 建立一条连边。然后，将该二部图分别在两个维度(用户，项)上进行投影，投影后节点 i 与 j 间的连边权重 σ(i,j)表示节点 i 与 j 的结构相似度。如下是几种常用的结构相似计算方法：

$$\sigma_{unmorm}(i,j)=|\Gamma_i\cap\Gamma_j| \quad (1)$$

$$\sigma_{Jaccard}(i,j)=\frac{|\Gamma_i\cap\Gamma_j|}{|\Gamma_i\cup\Gamma_j|} \quad (2)$$

$$\sigma_{cosine}(i,j)=\frac{|\Gamma_i\cap\Gamma_j|}{\sqrt{|\Gamma_i|*|\Gamma_j|}} \quad (3)$$

$$\sigma_{\min}(i,j)=\frac{|\Gamma_i\cap\Gamma_j|}{\min(|\Gamma_i|,|\Gamma_j|)} \quad (4)$$

其中，$\Gamma_i$ 为投影前节点 i 的邻居节点集合。σ(i,j)可以分别由公式（1）（2）（3）（4）得到。

这里的每个投影图就是由项或者用户组成的关联图。相应地，可以得到关联图对应的矩阵 CC，其中当 i≠j 时，$CC_{i,j}=\sigma(i,j)$，且 $CC_{i,i}=0$。再对 CC 以列为单位进行归一化得到随机矩阵 C，C 可以视为关联图的关联矩阵，$C_{i,j}$ 表示节点 i 对于节点 j 的关联系数，即对节点 j 而言节点 i 的重要程度。这样，关联图中所有节点对间的关联程度都可以用 C 中相应的元素表示。可以看到 CC 是一个对称矩阵，而 C 则不再具有这个性质。

### 2.2 SSR算法

SSR 算法的目标是通过排序的方式估计用户与项之间的关系。SSR 算法用向量表示查询

节点，向量中的每一维都代表关联图中的一个节点，其值则表示相应节点与该查询节点的亲和度。SSR 采用带重启动的随机游走（RWR）方式预测一个节点（或向量）Y 对查询节点（或向量）X 的重要性或关联程度。考虑从节点向量 X 开始的随机游走，每走一步保证如下两点：首先，如果一个节点 Z 与节点 P 联系紧密，而 P 又是与 X 相关联的，则通过 P，X 与 Z 也会建立一定的联系。其次，由于 Z 是通过间接的方式与 X 建立联系的，所以相比较 Z 与 P 及 P 与 X，Z 与 X 的联系强度会有一定的“衰减”。或者说，X 游走到 P 以后会以一定的概率沿 P 走向与 P 有联系的节点，除此之外，它还有可能返回节点 X 重新走步。定义 steady-state 概率向量 $S_X(Y)$ 表示从节点 X 开始的随机游走到达节点 Y 的概率。那么 $S_X$（Y）就表示了 Y 相对于 X 的亲密程度。

定义 $O_q$ 为查询对象，它可以是一个用户或一个项。如果 $O_q$ 是一个用户，SSR 算法要得到与它关系最紧密的项，而如果 $O_q$ 是项，SSR 算法得到与它关系最紧密的用户。这里若关联图或关联矩阵 C 由 k 个节点组成，则 $O_q$ 对应的 steady-state 概率向量 $S_q=(S_q(1),...S_q(k))$ 即为所求。定义归一化向量 $V_{Oq}$ 是根据 $O_q$ 在训练集中的记录而建立的节点向量，在归一化 $V_{Oq}$ 之前向量 $V_{oq}'$ 第 j 维的元素定义如下：

$$V_{Oq}^{'j}=\begin{cases}1 & if\ O_q \text{ is user and } O_q\ rated\ I_j, or\\ & if\ O_q\ is\ item\ and\ U_j\ rated\ O_q\\ 0 & otherwise\end{cases} \quad (5)$$

本文设计的推荐算法 SSR 如图 1 所示：

输入：表示用户与项之间关系的二部图 G，待推荐对象 $O_q$，参数 c

输出：对 $O_q$ 的推荐序列

1. 根据公式（1）（2）（3）或（4）得到 G 的一维投影图对应的关联图，若 $O_q \in U$，则投影得到关于项的关联图，若 $O_q \in I$，则投影得到关于用户的关联图
2. 令关联图对应的矩阵 CC 对角线的元素为 0，再以列为单位将其归一化，得到关联矩阵 C
3. 由（5）得到向量 $V'_{Oq}$，并对其归一化得到向量 $V_{Oq}$
4. t=0，初始化 $S_{Oq}=V_{Oq}$
5. while($S_{Oq}$ 不收敛||t<MAX_LOOP)
6. {
7. $S_{Oq}=(1-c)*C*S_{Oq}+c*V_{Oq}$
8. t++
9. }
10. 将 $S_{Oq}$ 中的元素按照降序排序
11. 若 $S_{Oq}(j)$ 排名靠前且 $V_{Oq}(j)=0$，则将 $S_{Oq}(j)$ 对应的项（用户）推荐给 $O_q$，得到 $O_q$ 的推荐序列

图 1 SSR 算法流程

在推荐算法 SSR 中，可调参数 c 为随机游走重启动的概率，(1-c) 为衰减因子，用来衡量信息的衰减程度。实际应用中，平均情况下，t=20 时，$S_{Oq}$ 即可收敛。

## 2.3 评价标准

本文采用 DOA（degree of agreement）[20]为标准对实验结果进行评价。根据使用对象

的不同将其扩展为用户的 DOA（U_DOA）和项的 DOA（I_DOA），其思想是：

对于 U_DOA，先定义 $NW_{ui}=n/(L_{ui} \cup T_{ui})$，其中 n 为所有项的个数，$L_{ui}$ 为用户 $U_i$ 在训练集中评价过的项集合，$T_{ui}$ 为 $U_i$ 在测试集中评价的项集合。

$$check\_order_{Ui}(I_j, I_k) = \begin{cases} 1, if\ (predict\_rank_{Ij} \geq prdict\_rank_{Ik}) \\ 0, otherwise \end{cases}$$

从而对于用户 Ui，其 DOA 得分定义如下：

$$DOA_{Ui} = \frac{\sum_{(j \in T_{Ui}, k \in NW_{Ui})} check\_order\ (I_j, I_k)}{|T_{Ui}| * |NW_{Ui}|}$$

I_DOA 的定义与 U_DOA 类似，这里不再赘述。通过定义可以看出，随机预测的 DOA 值大约为 50%，而理想情况下的 DOA 值为 100%。并且，实验中以所有用户(项)的 DOA 取平均作为总体的效果评价。

$$U\_DOA = \frac{\sum_{U_i} DOA_{Ui}}{|U|} \qquad 或 \qquad I\_DOA = \frac{\sum_{I_i} DOA_{Ii}}{|I|}$$

# 3. 实验

## 3.1 MovieLens数据集

MovieLens[9][3]是一个进行电影推荐的网站，GroupLens 小组从中整理出了适用于各种推荐场景的一个标准数据集。在本文使用的数据集中，对用户进行了处理，只考虑那些对超过 20 部电影打分的用户，总共包含 943 个用户（m=943）对 1，682 部电影（n=1，682）的评分，共约 100，000 个评分。

实验中使用数据集的方法与[15][17][19]中类似：对于参数训练部分（主要是对参数 c 的测试)，将整个数据集分成训练集与测试集两部分，而测试部分则使用五对数据集进行 5-交叉测试，每次用 80%的评分数据做训练集 20%的评分数据作为测试集。

## 3.2 参数选择及实验结果

在 PageRank 算法中重启动概率 c 经常取为 0.15，而在实验中，通过各个方法对 c 在(0，1）之间变化时的表现情况可以发现，c 越接近于 1，SSR 算法的推荐效果越好。其实这一方面与所用的关联图的半径有关[18]，一方面与数据的性质有关，也就是在当前的 MovieLens 数据集中，项（用户）的相似度绝大部分比例依赖于项（用户）间的直接路径数。接下来的实验中若无特殊说明，选择 c=0.99。

表 1 显示了不同的结构相似对应的 Movie_Rank 和 User_Rank 方法在进行 5-交叉测试时的表现：

表 1 5-交叉测试结果

| | Movie_Rank（I_DOA） | | | | User_Rank（U_DOA） | | | |
|---|---|---|---|---|---|---|---|---|
| | Unnorm | Jaccard | Cosine | Min | Unnorm | Jaccard | Cosine | Min |
| DOA(%) | 89.09 | 90.37 | 90.69 | 80.71 | 78.80 | 78.67 | 79.05 | 73.66 |

表 2 显示了使用 MovieLens 作为数据集的不同推荐算法的效果比较，这些算法的具体细节和实现可以参见[15][17][19]。这里 SSR 算法选择基于 Movie_Rank 的 Cosine 结构相似（M_Cosine）为代表与其他算法进行比较，其中的 ItemRank[17]与本文基于公式（1）的 Movie_Rank 实现方法类似。对于其中的每个算法，给出了其 5-交叉测试后的平均实验结果，以及与传统的 MaxF 算法的表现差异比较。MaxF 算法简单地将项按被浏览次数进行降序排序，每次都将其没有看过且排序最靠前的项推荐给给定用户。MaxF 算法也是比较的基准算法。

表 2 不同算法效果比较

| | Movie_Rank | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | MaxF | CT | PCA CT | One-way | Return | $L^+$ | ItemRank | Katz | SSR（M_Cosine） |
| DOA（%） | 84.07 | 84.09 | 84.04 | 84.08 | 72.63 | 87.23 | 87.76 | 85.83 | 90.69 |
| Difference with MaxF(in %) | 0 | +0.02 | -0.03 | +0.01 | -11.43 | +3.16 | +3.69 | +1.76 | +6.62 |

## 3.3 讨论

首先，可以发现在同一个数据集中，对电影排序的推荐效果明显好于对用户排序的结果，其实这与 MovieLens 数据集的性质有关。MovieLens 数据集对用户进行了处理，只保留了至少评价了 20 部电影的用户，以保证每个用户的兴趣都可以得到确切的反映。而数据集中的电影却没有经过类似处理。即数据集中存在许多被很少用户看过的电影，也有的电影可能被许多用户都看过，所以为电影构造节点向量的时候，会有一部分电影的节点向量相当稀疏，而另有一部分电影的节点向量相当稠密，这些都会造成区分度的下降。

在计算效率方面，给定关联矩阵 C 时，SSR 只要经过较少次数的迭代就可在 $O(n^2)$ 或 $O(m^2)$ 时间里得到一次推荐，而 $L^+$或者 CT 算法只能一次得到所有推荐，所以只能定时更新推荐而无法做到实时更新推荐。但 SSR 算法的一个问题在于，如果关联矩阵 C 中有一个节点发生了更新，则整个关联矩阵都需要更新，这将消耗 $O(n^2)$或 $O(m^2)$ 的时间，所以可以选择定时更新关联矩阵 C 而实时更新节点向量的方法，以得到更好的推荐效果。

最后是算法的空间复杂度和可扩展性。当基于项过滤对项进行排序时，关联矩阵 C 的大小与用户数无关，这是一个很有用的性质。因为一般的应用中，用户数目比项的数目有更高的数量级[17]，而且一般用户可能只对少数的一些项感兴趣，所以可以将用户的兴趣直接用链表的形式表现出来，可以节省巨大的空间资源。如果用 $k_u$ 表示每个用户平均感兴趣的项个数，那么此时算法的空间复杂度为 $O(n^2+m*k_u)$，即使用户数目不断增加，用户对项的兴趣也不断增加，SSR 算法都可以有效地应用。当有新的项时，SSR 和 ItemRank 只需要 $O(n)$的空间消耗，其 $L^+$与 CT 等则需要 $O(m+n)$。而当增加一个新用户时 ItemRank 和 SSR 需要 $O(k_u)$的空间，而 $L^+$，CT 等算法却需要 $O(m+n)$的空间，问题在于在一般的应用中 $m>>n>>k_u$。而基于用户过滤的用户排序，虽然空间消耗要高于对项排序的方法，但仍然小于 $L^+$与 CT 等方法，这里不再赘述。

# 4. 结论

本文提出了一种基于节点的结构相似与带重启动的随机游走相结合的协同过滤推荐算法，它可以为每个用户预测其可能感兴趣的项，也可以将项推荐给那些可能对其感兴趣的用户。通过在同一个数据集上与其它一些比较好的算法的对比实验结果表明，该方法不仅有更高的推荐准确度，同时有较低的时空复杂度以及较好的可扩展性，尤其是基于项过滤的项排

序算法可以构建应对海量数据的推荐系统[17]。

尽管基于项过滤的项排序算法在推荐效果和时空复杂性方面都表现优异，但通过基于用户过滤的用户排序算法的不佳表现可以发现，作为一个典型的协同过滤算法，该方法仍不能很好地应对实际应用中出现的数据稀疏和冷启动等问题。如何在保持算法有效性的前提下解决这些问题，将是未来工作的进一步研究方向。

## 参考文献

[1] Gedimiinas Adomavicus, and Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions [J]. IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6, June 2005, pages 734-749

[2] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering [J], IEEE Internet Computing, Jan./Feb. 2003, pages: 76-80

[3] B.N. Miller, I. Albert, S.K. Lam, etal. MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System [J]. Proc. Int'l Conf. Intelligent User Interfaces, 2003, pages: 263-266

[4] Resnick, P., Iacovou, N., Suchak, M, etal. Grouplens: an open architecture for collaborative filtering of Netnews [J]. Proceeding of the CSCW conference, Chapel Hill, NC, 1994, pages: 175-186

[5] Kim, J.W., Lee, B.H., Shaw, M.J., etal. Application of decision-tree induction techniques to personalized advertisements on Internet storefronts [J]. International Journal of Electronic Commerce, 5(3), 2001, pages: 45-62

[6] M. Balabanovic, Y. Shoham. Fab: Content-Based, Collaborative Recommendation [J]. Comm. ACM, vol. 40, no. 3, 1997, pages 66-72

[7] M. Pazzani. A Framework for Collaborative, Content-Based, and Demographic Filtering [J]. Artificial Intelligence Rev., Dec.1999, pages. 393-408

[8] Souvik Debnath, Niloy Ganguly, Pabitra Mitra. Feature Weighting in Content Based Recommendation System Using social network analysis[J]. WWW /Poster Paper, April, 2008, pages 1041-1042

[9] http://www.movielens.umn.edu

[10] D. Pavlov, D. Pennock. A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains[J]. Proc. 16th Ann. Conf. Neural Information Processing Systems (NIPS '02), 2002, pages: 1441-1448

[11] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of Predictive Algorithms for Collaborative Filtering [J]. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, 1998, pages 43-52

[12] David Liben-Nowell, Jon Kleinberg. The Link-Prediction Problem for Social Networks[J]. Journal of the American Society for Information Science and Technology, 58(7): 2007, pages 1019–1031

[13] E. A. Leicht, Petter Holme, and M. E. J. Newman. Vertex similarity in networks[J]. Physical Review E 73, 026120 (2006)

[14] Brin, S., and Page, L. The anatomy of a large-scale hypertextual Web search engine[J]. Computer Networks and ISDN Systems, 30(1–7), 1998, pages 107–117

[15] F. Fouss, A. Pirotte, and M. Sarens. Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation[J]. IEEE Transactions on Knowledge and Data Engineering 19 (3), 2007, pages 355–369

[16] Luh Yen & Marco Saerens, Amin Mantrach, etal. a family of dissimilarity measures between nodes generalizing both shortest-path and the Commute-time Distances[J]. Proceedings of the ACM SIGKDD international conference on Knowledge Discovery and Data Mining (KDD), 2008, pages 785-793

[17] Marco Gori, Augusto Pucol. A random-walk Based Scoring Algorithm with Application to Recommender Systems for Large-Scale E-commerce[J]. WEBKDD'06, 2006 pages: 127-146

[18] Pan, J. Y., Yang, H. J., Faloutsos, C. and, Duygulu, P. Automatic Multimedia Cross-modal Correlation Discovery[J]. In Proc. of ACM Intl. Conference on Knowledge Discovery and Data Mining (KDD'04), 2004 pages: 653-658

[19] F. Fouss, A Pirotte, J. M. Renders, etal. A novel way of computing dissimilarities between nodes of a graph, with application to collaborative filtering[J]. In IEEE/WIC/ACM International Joint Conference on Web Intelligence, 2005, pages 550-556

[20] S. Siegel and J. Castellan. Nonparametric Statistics for the Behavioral Sciences [M]. McGraw-Hill, 1988

[21] FF Kuo, MF Chiang, MK Shan, etal. Emotion-based music recommendation by association discovery from film music[J]. Proceedings of the 13th annual ACM international conference on Multimedia. 2005, Pages: 507 – 510.

# Collaborative Filtering Based on Nodes' Structural Similarity and Ranking

Liu Qi, Chen Enhong
Department of Computer Science and Technology, University of Science and Technology of China, Hefei (230027)

**Abstract**

This paper provides a novel collaborative based recommender algorithm based on combination of structural similarity of graph nodes and random walk with restart. This method not only performs with high precision but also is extendable. Besides, because this method generates similarity rank between each pair of user and item, this makes it can avoid inaccuaracy caused by poor coverage rate. We put forward two ways to carry out our algorithm, namely the item-collaborative-based item ranking method and the user-collaborative-based user ranking method. The effectiveness of the proposed method is demonstrated by experiments on a widely used benchmark dataset.

**Keywords:** recommender algorithms; collaborative filtering; structural simiality; ranking, random walk

**作者简介：**

刘淇（1986—），男，硕士研究生，主要研究方向为数据挖掘，推荐系统，社会网络；

陈恩红（1968—），男，博士，教授，博导，IEEE 高级会员（Senior Member），中国科技大学多媒体计算与通信教育部-微软重点实验室副主任,中国人工智能学会知识工程专委会、机器学习专委会委员，中国计算机学会人工智能与模式识别专委会委员，担任 20 余个国际学术会议的程序委员或主席。主要研究方向：语义 Web、 机器学习与数据挖掘、网络信息处理、约束满足问题 。