# A Topical PageRank Based Algorithm for Recommender Systems

Liyan Zhang Kai Zhang
School of Software, Tsinghua University
ly-zhang06@mails.tsinghua.edu.cn

Chunping Li
School of Software, Tsinghua University
cli@tsinghua.edu.cn

## ABSTRACT

In this paper, we propose a Topical PageRank based algorithm for recommender systems, which aim to rank products by analyzing previous user-item relationships, and recommend top-rank items to potentially interested users. We evaluate our algorithm on MovieLens dataset and empirical experiments demonstrate that it outperforms other state-of-the-art recommending algorithms.

## Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; Information filtering

## General Terms

Algorithms, Experimentation

## Keywords

Recommender System, Topical PageRank

## 1. INTRODUCTION

Recommender systems are automatic tools making personalized suggestions to users by analyzing previous interaction information. Solutions in this area are mainly classified into three categories: Content-based Approaches, Collaborative Filtering, and Hybrid Approaches.

Recently, graph based recommending algorithms have been proposed and achieved outstanding prediction performance[1]. However, some of them(e.g. [2]) fail to take into account potential features of items (e.g. item genre). In this paper, we present a Topical PageRank based algorithm, which considers item genre to rank items for users and recommends the top-ranked items to users correspondingly. Experimental results on MovieLens dataset show the superiority of the proposed algorithm over existing graph based recommending algorithms.

## 2. ALGORITHM

Our basic idea of the proposed algorithm lies in the investigation of correlation between ranking and recommender systems since we can recommend top-rank items to users. With this observation, we attempt to correlate ranking algorithms for web search with recommender systems. Specifically, we attempt to leverage Topical PageRank[3], a recently proposed superior ranking algorithm, to rank items and then recommend users with top-rank items.

Generally speaking, the input of a recommender system can be treated as a user-item matrix consisting of tuples, each denoted as $t_{k,i} = (u_k, m_i, r_{ki})$, where $u_k$ is one of users, $m_i$ is one of items, $r_{ki}$ is an evaluation score, generally an integer ranging from 1 to 5. Our goal is to compute rank score for each item by analyzing the given evaluation scores with respect to each user, and then suggest the top-rank items to users.

### 2.1 Data Model: Correlation Graph

A key point for a recommending algorithm is exploiting the item correlations. As an initial step, we first establish a correlation graph among items. Experiences reveal that correlation between items can be indicated by user preference lists. Consequently, we can reasonably assume that item $m_i$ and item $m_j$ are highly related, if they tend to co-occur in preferences lists of different users. We define $\mu_{i,j}$ as the set of users $u_k$ choosing both item $m_i$ and item $m_j$ in the training set of user-item tuples $Tr$, that is:

$$\mu_{i,j} = \begin{cases} u_k : (t_{k,i} \in Tr) \wedge (t_{k,j} \in Tr), & i \neq j, \quad (1) \\ \emptyset, & i = j. \quad (1') \end{cases}$$

Next, we define a $|M| * |M|$ ($|M|$ is the total number of items) matrix $\tilde{\mathcal{M}}$, with each element $\tilde{\mathcal{M}}_{i,j}$ representing the number of users choosing both item $m_i$ and item $m_j$. Alternatively, each $\tilde{\mathcal{M}}_{i,j}$ is denoted as $|\mu_{i,j}|$, where $|\ |$ denotes the cardinality of a set. Furthermore, we normalize matrix $\tilde{\mathcal{M}}$ to obtain a stochastic matrix-correlation matrix $\mathcal{M}$, with each element $\mathcal{M}_{i,j} = \tilde{\mathcal{M}}_{i,j}/\omega_j$, where $\omega_j$ is the sum of entries in $j-th$ column of $\tilde{\mathcal{M}}$. Thus, we obtain a correlation graph $\mathbb{G}$ on basis of the correlation matrix $\mathcal{M}$, in which an edge between item $m_i$ and $m_j$ is established if and only if $\mathcal{M}_{i,j} > 0$.

### 2.2 Topical PageRank based Algorithm

The above item correlation graph bears two properties: propagation and attenuation, which are two key features for PageRank algorithm. A. Pucci has developed a PageRank based algorithm, known as PaperRank[2], for paper recommender systems. We follow a similar way but leverage Topical PageRank algorithm for recommending by exploiting genres of items. Specifically, we rank items for different users with Topical PageRank given the same item correla-


*SIGIR'08,* July 20–24, 2008, Singapore.
ACM 978-1-60558-164-4/08/07.

tion graph, and recommend top-rank items to potentially interested users.

We first briefly introduce Topical PageRank algorithm [3]. This algorithm takes into account the topic of pages, and introduces the PageRank score of a page $p$ in topic $z$

$$A_z(p) = d \sum_{q:q\to p} \frac{\alpha A_z(q) + (1-\alpha)C_z(q)A(q)}{O(q)} + \frac{1-d}{N}C_z(p) \quad (2)$$

where $q$ denotes any page linked to page $p$ and $O(q)$ represents the degree of page $q$. $C_z(p)$ is the probability with which page $p$ belongs to topic $z$, $N$ is the number of pages, $d$ and $\alpha$ are link related and topic related parameters respectively. The equivalent matrix form of Equation(2) for different page-topic pairs is defined as:

$$A = d\alpha G \bullet A + d(1-\alpha)G \bullet F_{CA} + (1-d)\frac{C}{N} \quad (3)$$

where $G$ is the normalized connectivity matrix for graph, $F_{CA}$ is an assistant matrix, $\bullet$ means matrix product.

With the similar idea, we define item rank score matrix for different item-genre pairs as

$$R = d\alpha\mathcal{M} \bullet R + d(1-\alpha)\mathcal{M} \bullet F + (1-d)I \quad (4)$$

where the $|M| * n$ matrix $R$ ($n$ is number of item genres) contains $R_{ig}$ indicating predicted rank score for item $m_i$ on genre $g$ as to a given user, $\mathcal{M}$ is the item correlation matrix, $F$ is a $|M| * n$ assistant matrix, and $I$ is a $|M| * n$ matrix associated with the original user-item evaluation score. Note that we choose $d$=0.15, $\alpha$=0.1 in the following experiments empirically. The iterative formula for calculating $R^{u_k}$ for different users $u_k$ is given by:

$$\begin{cases} R_{ig}^{u_k}(0) = \dfrac{1}{|M| * n}(1 \le i \le |M|, 1 \le g \le n) \\ F_{ig}(t) = (\sum_{g=1}^{n} R_{ig}^{u_k}(t-1)) * P_{ig} \\ R^{u_k}(t) = d\alpha\mathcal{M}R^{u_k}(t-1) + d(1-\alpha)\mathcal{M}F(t) + (1-d)I^{u_k} \end{cases} \quad (5)$$

In formula(5), the first equation initializes $R^{u_k}$ and the second one computes assistant matrix $F$, where $P_{ig}$ refers the probability with which item $m_i$ belongs to genre $g$. Each element of $I^{u_k}$ in the third equation is defined as $I_{ig} = \frac{\tilde{I}_{ig}}{\sum_{g=1}^{n}\tilde{I}_{ig}}$, with $\tilde{I_{ig}} = r_{ki} * P_{ig}$ for user $u_k$ on item $m_i$. Given totally $\kappa$ users, we can get $\kappa$ different $I^{u_k}$, and thus obtain $\kappa$ corresponding matrix $R^{u_k}$ by iteratively computing formula(5).

After getting matrix $R^{u_k}$ for each user $u_k$, we introduce a $\kappa * |M|$ matrix $TR$, where $TR_{ki}$ denotes the estimated item rank score for user $u_k$ to item $m_i$, defined as

$$TR_{ki} = \sum_{g=1}^{n}(R_{ig}^{u_k} * P_{ig}) \quad (6)$$

where the higher $TR_{ki}$ is, the more user $u_k$ prefers item $m_i$ to other lower score items. We finally recommend top-rank items in vector $TR_k$ to user $u_k$.

# 3. EXPERIMENTS

We evaluated our algorithm on MovieLens dataset containing 100,000 ratings for 1,682 movies by 943 users, which is constructed from the popular MovieLens Site for recommending movies. The dataset has been divided into 5 predefined splits, each with 80% of ratings as training set $Tr$ and 20% as testing set $Te$.

**Table 1: DOA comparison of our algorithm (TR) with PaperRank (PR) on the 5 splits.**

| | Split1 | Split2 | Split3 | Split4 | Split5 | **Mean** |
|---|---|---|---|---|---|---|
| PR | 87.69 | 87.71 | 87.65 | 87.51 | 88.14 | 87.74 |
| TR | 89.05 | 89.12 | 89.26 | 88.97 | 89.01 | **89.08** |

**Table 2: Overall DOA comparisons of our algorithm (TR) with other scoring algorithms.**

| | L+ | PR | Katz | Dijkstra | **TR** |
|---|---|---|---|---|---|
| DOA | 87.18 | 87.74 | 85.26 | 49.65 | **89.08** |

In order to compare our algorithm with other promising graph based approaches, we chose the degree of agreement ($DOA$) as the performance measure. We first introduce $DOA$ for a specific user. As to a particular user $u_k$, the whole movie set can be divided into 3 subsets, movies in training set $Tr_{u_k}$, testing set $Te_{u_k}$ and unwatched set $Nw_{u_k}$. Intuitively, movies watched by the user should rank higher than movies unwatched. Thus, we define a Boolean function $f_{u_k}$ to represent the comparison result between a pair of movies from different subsets. If the score $TR_{ki}$ ($m_i \in Te_{u_k}$) is higher than $TR_{kj}$ ($m_j \in Nw_{u_k}$), $f_{u_k}$ is set 1, otherwise, $f_{u_k}$ is 0. Finally, we denote

$$DOA_{u_k} = \frac{\sum_{(m_i \in Te_{u_k}, m_j \in Nw_{u_k})} f_{u_k}(m_i, m_j)}{|Te_{u_k}| \bullet |Nw_{u_k}|}$$

which represents the percentage of correct orders with regard to total order pairs for a given user. Computing the average of $DOA_{u_k}$ for all users, we could obtain the final performance index $DOA$.

The results in Table 1 demonstrate that the proposed Topical PageRank based algorithm (TR) outperforms PaperRank (PR) in each split, indicating prediction improvement by exploiting item genre. In Table 2, comparisons with other recommending algorithms mentioned in [1] further prove the superiority of our algorithm.

# 4. CONCLUSIONS

In this paper, we present a Topical PageRank based algorithm for recommender systems, which performs better than other algorithms on MovieLens Dataset.

# 5. ACKNOWLEDGEMENT

This work was supported by National Nature Science Funding of China under Grant No. 90718022.

# 6. REFERENCES

[1] F. Fouss, A. Pirotte, and M. Saerens. A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation. In *Web Intelligence*, pages 550-556, 2005.

[2] M. Gori and A. Pucci. Research paper recommender systems: A random-walk based approach. In *Web Intelligence*, pages 778-781, 2006.

[3] L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *SIGIR*, pages 91-98, 2006.