A Random-Walk Based Scoring Algorithm with Application to Recommender Systems for Large-Scale E-Commerce

Marco Gori and Augusto Pucci

Dipartimento di Ingegneria dell’Informazione
University of Siena
Via Roma 56, 53100 Siena (ITALY)
Recommending Problem

- Too many resources (products, movies, books, papers, web pages...)
- Resource filtering is time consuming
- Help users during the filtering process
- Personalized ranking of resources
- Suggest **useful** resources to a given user
Problem Ingredients

- A set of users: \( U = \{ u_1, u_2, ..., u_{|U|} \} \)
- A set of items: \( P = \{ p_1, p_2, ..., p_{|P|} \} \)
- A set of opinions: \( t_{i,k} = (u_i, p_k, r_{i,k}) \)
- Correlation between items:
  \[
  C_{i,k} = \text{corr}(p_i, p_k)
  \]

NOTE: the correlation matrix has to be normalized
Goal

Properly rank items in set $P$ with respect to:

- User $u$ preferences
- Other users’ preferences
- Item correlation matrix $C$
The Idea

- Correlation Graph obtained by Correlation Matrix
- Spread user preferences over the Correlation Graph
ItemRank Algorithm

- Iterative equation:

\[ \text{IR}(t + 1) = \alpha \cdot C \cdot \text{IR}(t) + (1 - \alpha) \cdot d \]

- \( \text{IR} \) ItemRank values vector for user \( u \)
- \( \text{IR}_i \) ItemRank value for item \( p_i \)
- \( d \) preference vector for user \( u \)

- About 20 iterations to converge
ItemRank for Movies

- Movie Graph
- User $u$ rates some movies (non-zero entries in $d$)
- Correlation Graph built by co-occurrence of movies in user preference lists
- ItemRank suggests good movies to be watched
Movie Data Set

- 943 users
- 1,682 movies
- 100,000 ratings
- Ratings from 1 to 5
- 5 standard training/test splittings

![Rating Distribution - Train Set](image)
Movie Data Set

WebKDD 2006 Workshop on Knowledge Discovery on the Web, Aug. 20, 2006, at KDD 2006, Philadelphia, PA, USA
Experimental Results

Macro-DOA and difference with MaxF (in %):

<table>
<thead>
<tr>
<th>MaxF</th>
<th>CT</th>
<th>PCA CT</th>
<th>One-way</th>
<th>Return</th>
<th>(L^+)</th>
<th>ItemRank</th>
<th>Katz</th>
<th>Dijkstra</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.07</td>
<td>84.09</td>
<td>84.04</td>
<td>84.08</td>
<td>72.63</td>
<td>87.23</td>
<td>87.76</td>
<td>85.83</td>
<td>49.96</td>
</tr>
<tr>
<td>0</td>
<td>+0.02</td>
<td>-0.03</td>
<td>+0.01</td>
<td>-11.43</td>
<td>+3.16</td>
<td>+3.69</td>
<td>+1.76</td>
<td>-34.11</td>
</tr>
</tbody>
</table>

ItemRank with binary graph:

<table>
<thead>
<tr>
<th>SPLIT</th>
<th>ItemRank</th>
<th>ItemRank (binary graph)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>micro DOA</td>
<td>Macro DOA</td>
</tr>
<tr>
<td>SPLIT 1</td>
<td>87.14</td>
<td>87.73</td>
</tr>
<tr>
<td>SPLIT 2</td>
<td>86.98</td>
<td>87.61</td>
</tr>
<tr>
<td>SPLIT 3</td>
<td>87.20</td>
<td>87.69</td>
</tr>
<tr>
<td>SPLIT 4</td>
<td>87.08</td>
<td>87.47</td>
</tr>
<tr>
<td>SPLIT 5</td>
<td>86.91</td>
<td>88.28</td>
</tr>
<tr>
<td>Mean</td>
<td>87.06</td>
<td>87.76</td>
</tr>
</tbody>
</table>
ItemRank for Papers

- Paper Graph
- User $u$ selects some interesting papers (non-zero entries in $d$)
- Correlation Graph is a weighted citation graph connecting papers
- ItemRank suggests good reference candidates
<table>
<thead>
<tr>
<th>Topic</th>
<th>Number of Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed Databases</td>
<td>4,961</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>3,463</td>
</tr>
<tr>
<td>Hidden Markov Model</td>
<td>3,978</td>
</tr>
<tr>
<td>Multiprocessors</td>
<td>4,292</td>
</tr>
<tr>
<td>Page Rank</td>
<td>5,225</td>
</tr>
<tr>
<td>Recommendation System</td>
<td>4,367</td>
</tr>
<tr>
<td>Relevance Feedback</td>
<td>6,278</td>
</tr>
<tr>
<td>Semantic Web</td>
<td>6,423</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>4,206</td>
</tr>
</tbody>
</table>
Experimental Results

The image shows a bar chart with the title "Experimental Results." The chart compares different categories such as "Distr. Databases," "Feature Extraction," "HMM," "Multiprocessors," "Page Rank," "Rec. Systems," "Relevance Feedback," "Semantic Web," and "SVM." Each category has bars representing different test/train split ratios: 50%/50%, 40%/60%, 30%/70%, and 20%/80%. The y-axis represents the percentage in the first 10 positions, while the x-axis lists the categories.
Performance Issues

BAD NEWS

- ItemRank computation is user dependent
- Correlation Graph can be huge

GOOD NEWS

- User clustering is possible (also soft-clustering)
- C matrix is sparse: efficient way to compute PageRank-like algorithms
Future Works

- Scalability improvements:
  - Users soft-clustering
  - Efficient computation with sparse matrices
- Correlation matrix learning
- Comparison with graph regularization frameworks
- Applications to new datasets