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)

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

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: $U = \{ u_1 , u_2 , \ldots , u_{|U|} \}$
- A set of items: $P = \{ p_1 , p_2 , \ldots , p_{|P|} \}$
- A set of opinions: $t_{i,k} = (u_i , p_k , r_{i,k})$
- Correlation between items:

$$C_{i,k} = \mathrm{corr} (p_i , p_k)$$

NOTE: the correlation matrix has to be normalized

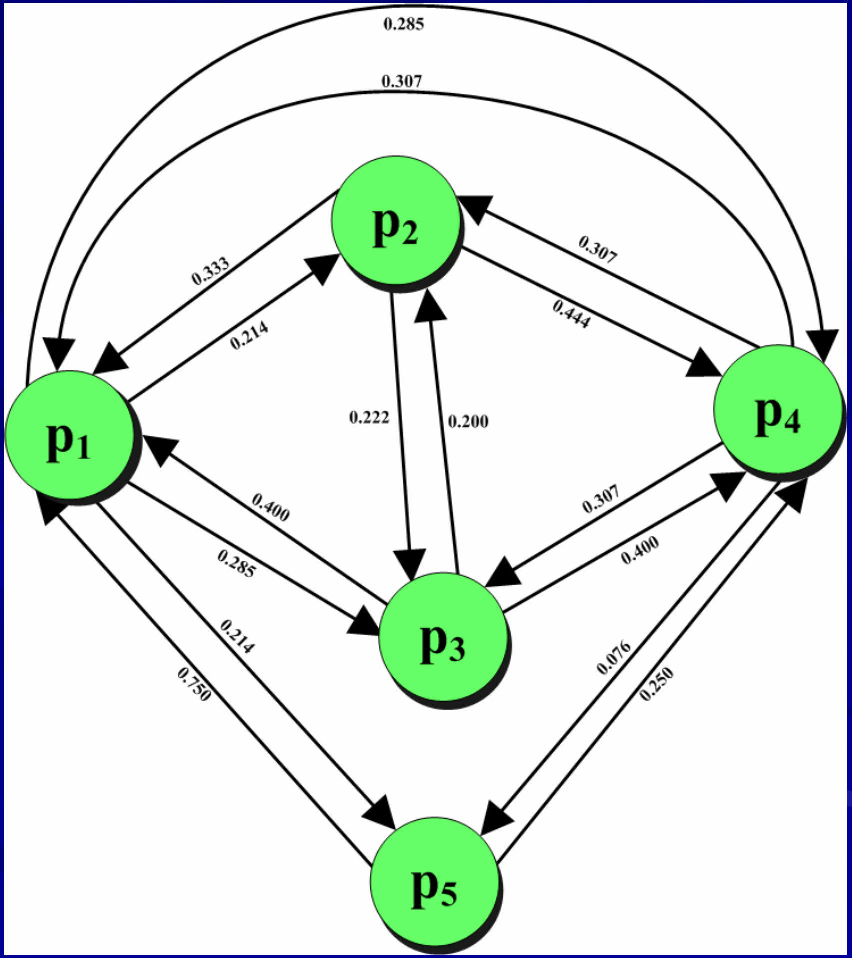
Goal

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:

$$\mathbf{IR}(t+1) = \alpha \cdot \mathcal{C} \cdot \mathbf{IR}(t) + (1-\alpha) \cdot \mathbf{d}$$

- **IR** ItemRank values vector for user *u*
- IR_i ItemRank value for item p_i
- **d** preference vector for user *u*

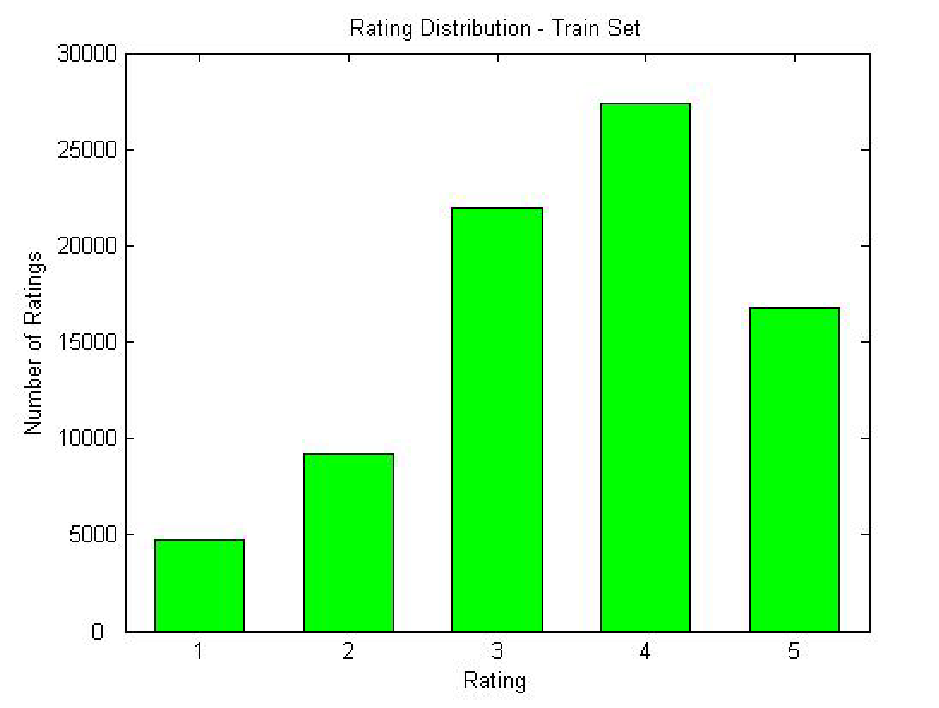
- About 20 iterations to converge

ItemRank for Movies

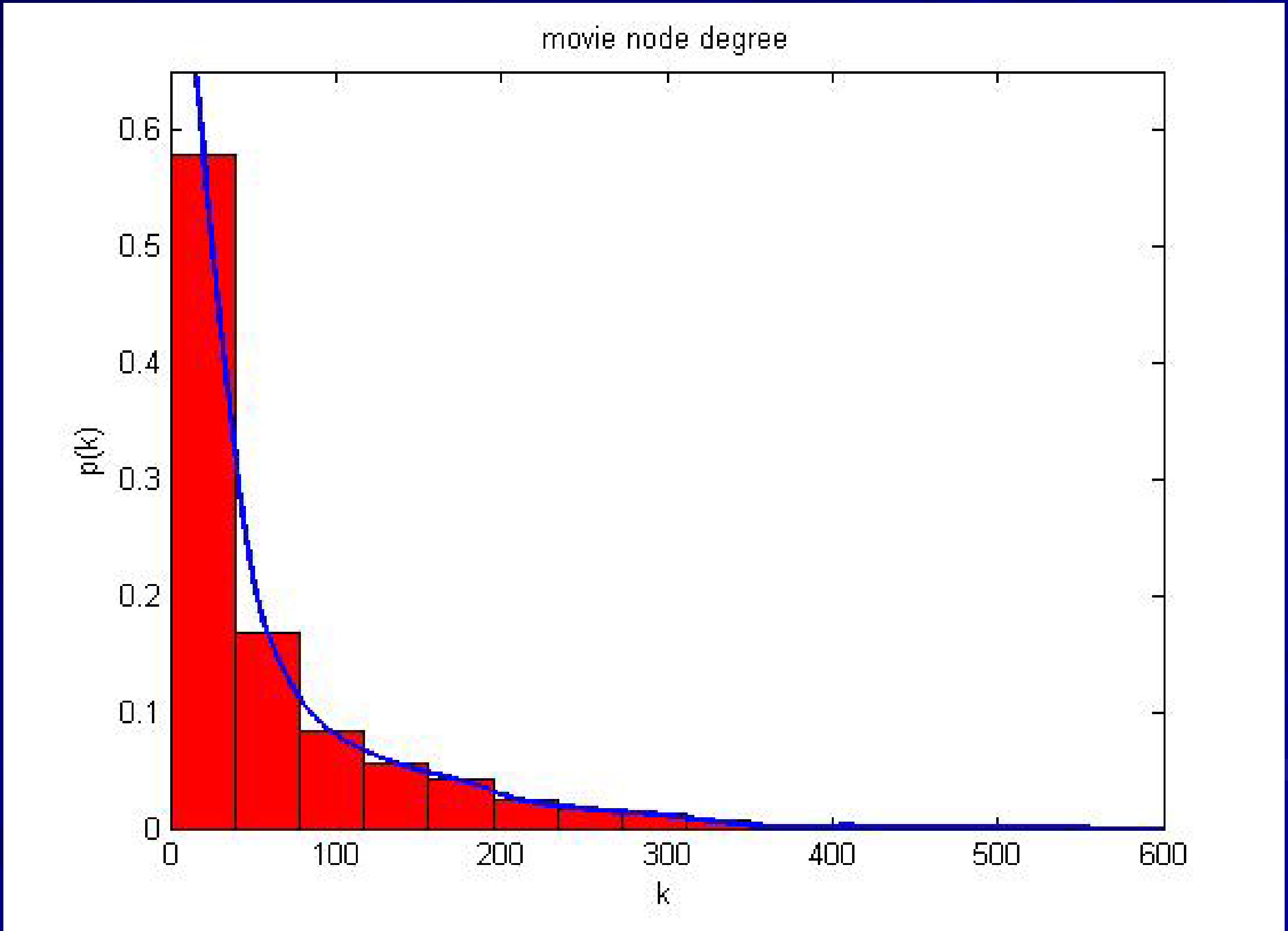
- Movie Graph
- User *u* rates some movies (non-zero entries in **d**)
- Correlation Graph built by co-occurrence of movies in user preference lists
- ItemRank suggests good movies to be watched

Movie DataSet

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



Movie DataSet



Experimental Results

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

MaxF	CT	PCA CT	One-way	Return	L^+	ItemRank	Katz	Dijkstra
84.07	84.09	84.04	84.08	72.63	87.23	**87.76**	85.83	49.96
0	+0.02	-0.03	+0.01	-11.43	+3.16	**+3.69**	+1.76	-34.11

- ItemRank with binary graph:

	ItemRank		ItemRank (binary graph)	
	micro DOA	Macro DOA	micro DOA	Macro DOA
SPLIT 1	87.14	87.73	71.00	72.48
SPLIT 2	86.98	87.61	70.94	72.91
SPLIT 3	87.20	87.69	71.17	72.98
SPLIT 4	87.08	87.47	70.05	71.51
SPLIT 5	86.91	88.28	70.00	71.78
Mean	**87.06**	**87.76**	**70.63**	**72.33**

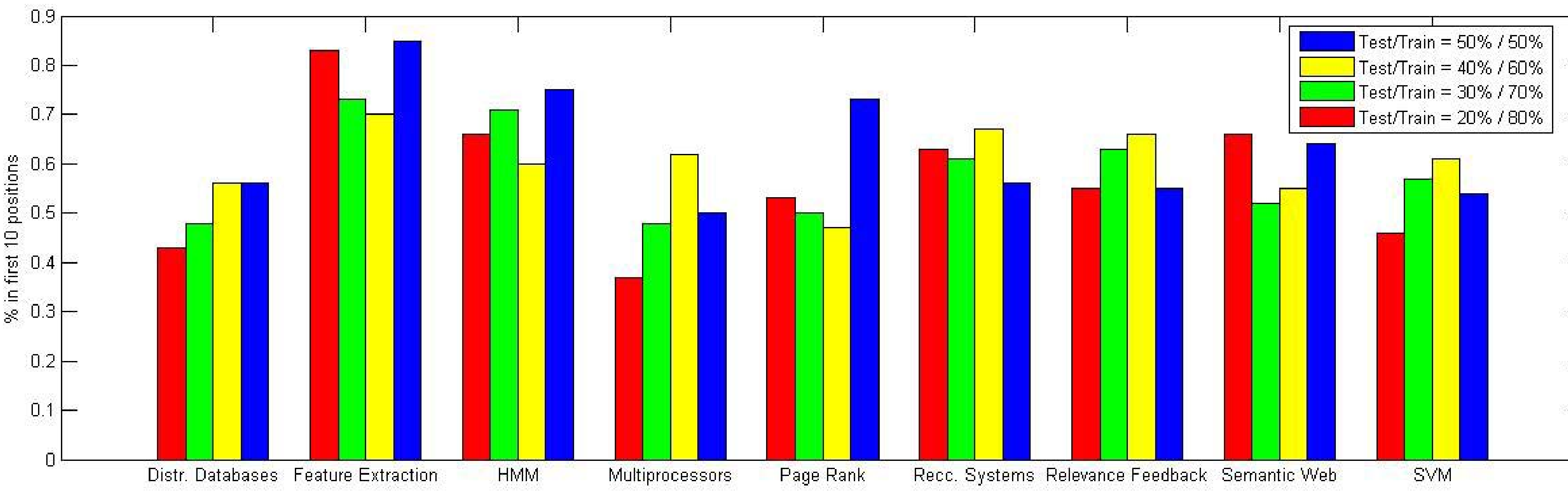
ItemRank for Papers

- 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

Paper DataSet

Topic	Number of Papers
Distributed Databases	4,961
Feature Extraction	3,463
Hidden Markov Model	3,978
Multiprocessors	4,292
Page Rank	5,225
Recommendation System	4,367
Relevance Feedback	6,278
Semantic Web	6,423
Support Vector Machine	4,206

Experimental Results



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