Mining Web Multi-resolution Community-based Popularity for Information Retrieval

Laurence A. F. Park Kotagiri Ramamohanarao

Department of Computer Science and Software Engineering
University of Melbourne, Australia
{lapark,rao}@csse.unimelb.edu.au

ACM Sixteenth Conference on Information and Knowledge Management

Global popularity

PageRank is a measure of global Web popularity. It uses the consensus of the entire Web to compute page popularity. Therefore it is suited to general queries.

Problem

Specialised queries require consensus from specialised communities, therefore are not suited to PageRank.

1. How do we compute a popularity list relative to a community?
2. How do we choose a list at query time?

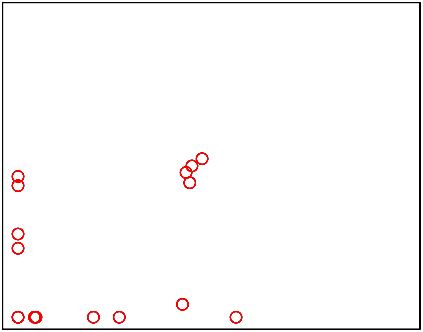
Outline

1. Multi-resolution popularity
2. Computing multi-resolution popularity
   - Pagerank's many solutions
   - Symmetric non-negative matrix factorisation
   - SNMF_1 - PageRank equivalence
   - Computing community popularity using SNMF
3. Using multi-resolution popularity
   - Query independent selection
   - Oracle selection
   - Rank based selection
   - Score based selection
4. Conclusion

Outline

Lowest resolution (Global Popularity)

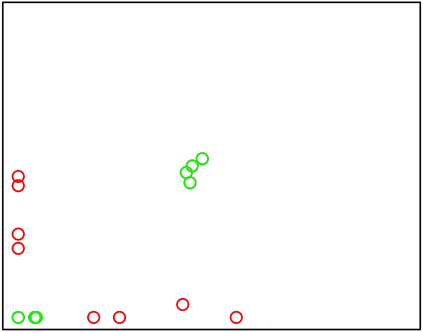
Where can I buy a CD?



General queries can use the consensus of the whole community (e.g. K-mart).

Medium resolution

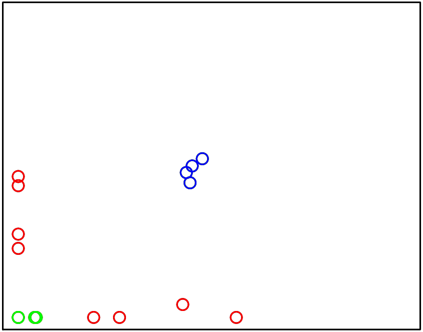
Where can I buy a movie soundtrack CD?



Specific queries cannot be answered by the general public and require specific knowledge (e.g. HMV).

High resolution

Where can I buy a 70's synthesiser movie soundtrack CD?



Specialised queries cannot be answered by specific groups and require specialised knowledge (e.g. Steve's super synthesiser music store).

Multi-resolution popularity lists for Web search

To use multi-resolution popularity lists, we must be able to:

1. generate popularity lists for each community in a given resolution
2. choose a popularity list once given a query

Outline

1. Multi-resolution popularity
2. Computing multi-resolution popularity
   - Pagerank's many solutions
   - Symmetric non-negative matrix factorisation
   - SNMF_1 - PageRank equivalence
   - Computing community popularity using SNMF
3. Using multi-resolution popularity
   - Query independent selection
   - Oracle selection
   - Rank based selection
   - Score based selection
4. Conclusion

PageRank

PageRank equation

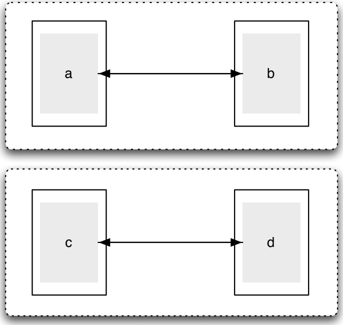
PageRank is the first eigenvalue of the weighted link matrix L:

$$p_i = \lambda \sum_{j \in B_i} \frac{p_j}{\#(l_j)} \Leftrightarrow \tilde{p} = \lambda \tilde{p} L$$

Note that there are many solutions to the eigenvalue problem. Using PageRank, we choose the solution with the greatest eigenvalue.

Problem with one popularity list

Simple example



PageRank solution

$$\tilde{p}_1 = [\ 0.5\ 0.5\ 0.5\ 0.5\ ]$$

Using one popularity list produces equal popularity for all pages, when we can clearly see that it should not be equal.

Choosing many eigenvectors

- By examining the other solutions that are offered by the eigenvalue decomposition, we may find popularity lists relative to various communities within the Web.
- Unfortunately, the eigenvectors may contain complex and negative elements, which do not provide an obvious order.

Problem

How can we compute the eigenvalue decomposition, with the constraint that the elements must be positive and real?

Non-negative matrix factorisation

Decompose the matrix A into matrices F and G:

$$A \approx FG^T$$
$$(d \times d) \approx (d \times n)(n \times d)$$

where F and G contain non-negative elements and provide the best approximation of A.

Non-negative matrix factorisation

Decompose the matrix A into matrices F and G:

$$A \approx FG^T$$
$$(d \times d) \approx (d \times n)(n \times d)$$

where F and G contain non-negative elements and provide the best approximation of A.

Symmetric non-negative matrix factorisation

We add the constraint that $F = G \Rightarrow A \approx FF^T$

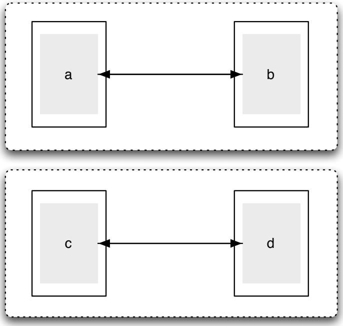
The equivalence of PageRank and SNMF

If we observe the $n = 1$ symmetric non-negative matrix factorisation, we find that it is proportional to PageRank:

$$F = \text{SNMF}_1(A) \propto \text{PageRank}\,(A)$$

This implies that SNMF_1 produces the same ranked list as PageRank

Computing community popularity using SNMF
Simple example revisited



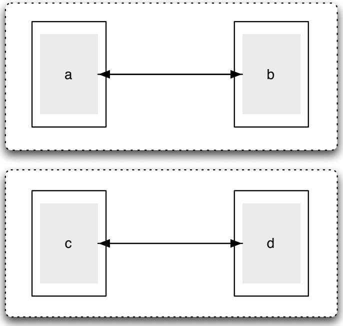
SNMF solution

$$SNMF_1 = [\ 0.5\ 0.5\ 0.5\ 0.5\ ]$$

Using multiple popularity lists, we are able to compute the popularity for each group.

Computing community popularity using SNMF

Simple example revisited



SNMF solution

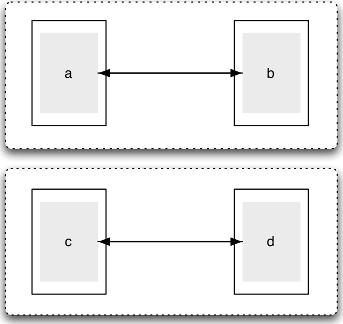
$$\text{SNMF}_1 = [\, 0.5\ 0.5\ 0.5\ 0.5 \,]$$

$$\text{SNMF}_2 = \begin{cases} [\, \color{red}{0.67}\ \color{red}{0.67}\ 0.00\ 0.00 \,] \\ [\, 0.05\ 0.05\ \color{red}{0.68}\ \color{red}{0.68} \,] \end{cases}$$

Using multiple popularity lists, we are able to compute the popularity for each group.

Computing community popularity using SNMF

Simple example revisited



SNMF solution

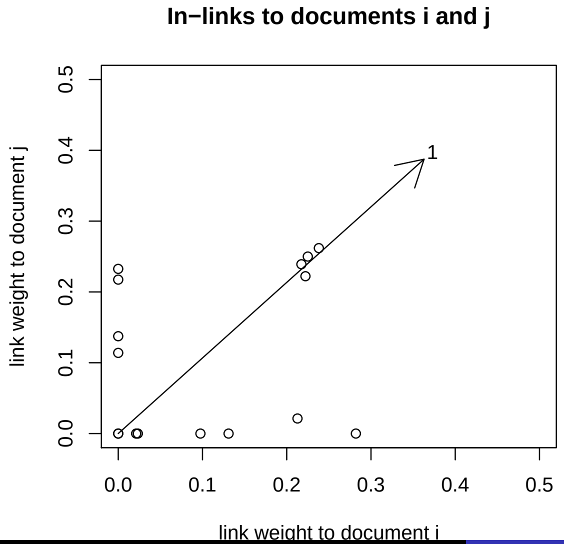
$$\text{SNMF}_1 = [\ 0.5\ 0.5\ 0.5\ 0.5\ ]$$

$$\text{SNMF}_2 = \begin{cases} [\ 0.67\ 0.67\ 0.00\ 0.00\ ] \\ [\ 0.05\ 0.05\ 0.68\ 0.68\ ] \end{cases}$$

$$\text{SNMF}_3 = \begin{cases} [\ 0.06\ 0.06\ 0.56\ 0.56\ ] \\ [\ 0.01\ 0.01\ 0.39\ 0.39\ ] \\ [\ 0.68\ 0.68\ 0.00\ 0.00\ ] \end{cases}$$

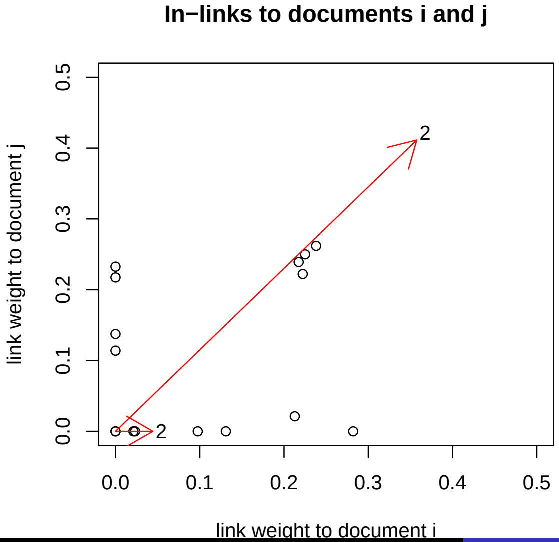
Using multiple popularity lists, we are able to compute the popularity for each group.

Multi-resolution popularity



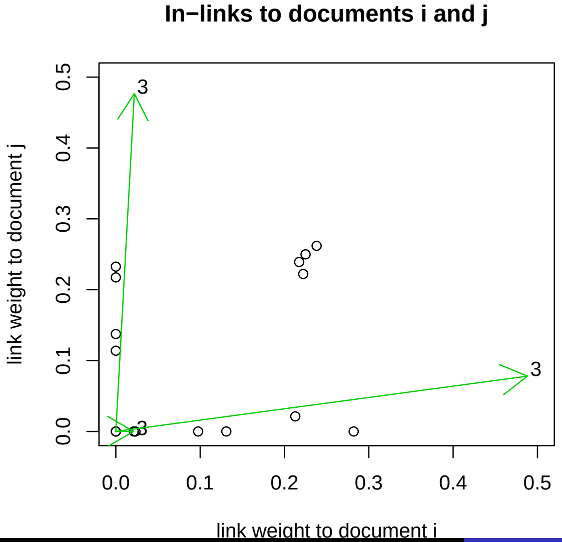
$SNMF_1$ (PageRank) computes a score based on the whole data set.

Multi-resolution popularity



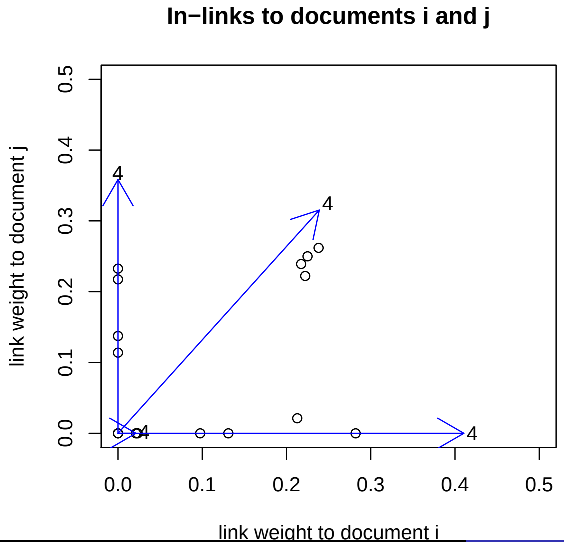
$SNMF_2$ Split the popularity between those that link to *i* and *j* and those that don't.

Multi-resolution popularity



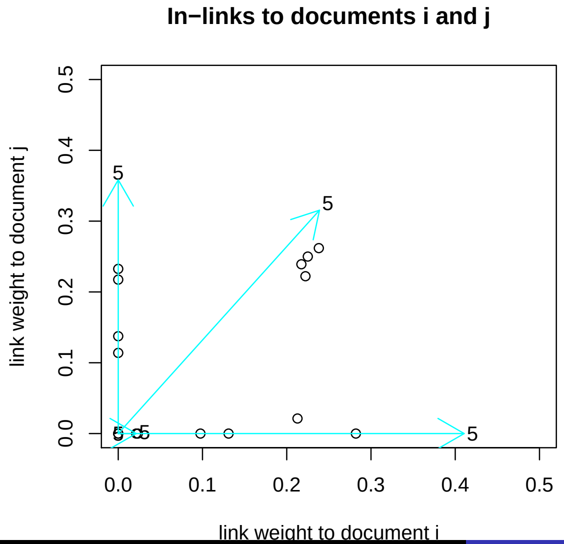
$SNMF_3$ splits further into those that link to *i* and those that link to *j*.

Multi-resolution popularity



SNMF_4 provides lists for those that link to *i*, those that link to *j*, those that link to both and those that link to neither.

Multi-resolution popularity



$SNMF_5$ introduces another list that may affect other documents.

Outline

1. Multi-resolution popularity
2. Computing multi-resolution popularity
   - Pagerank's many solutions
   - Symmetric non-negative matrix factorisation
   - SNMF_1 - PageRank equivalence
   - Computing community popularity using SNMF
3. Using multi-resolution popularity
   - Query independent selection
   - Oracle selection
   - Rank based selection
   - Score based selection
4. Conclusion

Experimental settings

- TREC GOV2 collection (25 million Web documents)
- 100 queries (topics 701-800)
- Computed 10 popularity lists (using resolutions 1,2,3,4).
- Typical Web searcher does not examine more than the top ten, therefore we used the measure Prec10.

Query Independent selection

Resolution	1	2		3		
Community	1	1	2	1	2	3
Prec10	0.36	0.38	0.38	0.39	0.38	0.38
PageRank ratio	1	1.06	1.04	1.06	1.03	1.05
Matched queries	22	26	28	20	23	19

Resolution	4			
Community	1	2	3	4
Prec10	0.37	0.38	0.40	0.38
PageRank ratio	1.03	1.05	1.10	1.04
Matched queries	22	22	22	23

Note that each community in each resolution provides a greater precision that the lowest resolution.

Oracle selection

The oracle method knows which community list to choose for each query. This shows the potential of using multi-resolution community based popularity lists.

Selection	Oracle
Prec10	0.544
PageRank ratio	1.497
Matched queries	100

Choosing a list

Problem

How do we choose the best list for a given query?

The best list should rank the initial query results higher and tighter than the other lists.
Qualities for matching list:

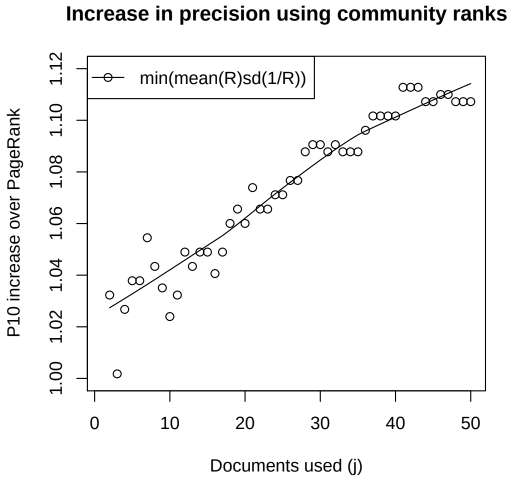
- minimise mean($R_{i,j}$)
- maximise mean($1/R_{i,j}$)
- minimise sd($R_{i,j}$)
- minimise sd($1/R_{i,j}$)
- maximise mean($S_{i,j}$)
- minimise mean($1/S_{i,j}$)
- minimise sd($S_{i,j}$)
- minimise sd($1/S_{i,j}$)

Rank based selection

Candidates	Rank in list 1	Rank in list 2
d_1	5	30
d_2	12	31
d_3	40	21
d_4	15	22
d_5	22	24
mean(R)	18.8	25.6
mean($1/R$)	0.08	0.04
sd(R)	13.3	4.6
sd($1/R$)	0.068	0.006

The candidate documents are top N scoring documents using term frequency matching.

Rank based selection result



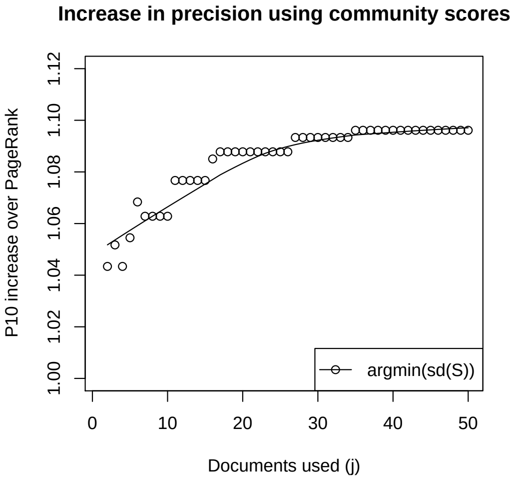
The precision increases with the number of candidate documents.

Score based selection

Candidates	Score in list 1	Score in list 2
d_1	0.31	0.03
d_2	0.18	0.03
d_3	0.09	0.07
d_4	0.12	0.06
d_5	0.11	0.04
mean(S)	0.162	0.046
mean($1/S$)	7.46	24.52
sd(S)	0.089	0.018
sd($1/S$)	3.09	8.97

The candidate documents are top N scoring documents using term frequency matching.

Score based selection results



The precision increases with the number of candidate documents.

Outline

Conclusions

- The Web contains many communities, therefore a single popularity list is not suitable for all queries.
- Multi-resolution popularity lists can be computed using Symmetric non-negative matrix factorisation.
- The lowest community resolution is equivalent to PageRank.
- We have shown that a 50% increase over PageRank is possible using four resolutions.
- By comparing the ranks of the candidate documents within each popularity list, we were able to achieve an 11% increase over PageRank.