# Multiresolution Web Link Analysis Using Generalized Link Relations

Laurence A.F. Park and Kotagiri Ramamohanarao, *Member*, *IEEE*

**Abstract**—Web link analysis methods such as PageRank, HITS, and SALSA have focused on obtaining global popularity or authority of the set of Web pages in question. Although global popularity is useful for general queries, we find that global popularity is not as useful for queries in which the global population has less knowledge of. By examining the many different communities that appear within a Web page graph, we are able to compute the popularity or authority from a specific community. Multiresolution popularity lists allow us to observe the popularity of Web pages with respect to communities at different resolutions within the Web. Multiresolution popularity lists have been shown to have high potential when compared against PageRank. In this paper, we generalize the multiresolution popularity analysis to use any form of Web page link relations. We provide results for both the PageRank relations and the In-degree relations. By utilizing the multiresolution popularity lists, we achieve a 13 percent and 25 percent improvement in mean average precision over In-degree and PageRank, respectively.

**Index Terms**—Symmetric nonnegative matrix factorization, PageRank, SALSA, in-degree, Web link analysis.

## 1 Introduction

When searching the Web, it is common to find that information of interest to the majority of the population can be easily found, but specialized information requires more effort in the form of many query reformulations until the desired information is located or the search is abandoned.

The ranked list of Web pages provided by a search engine is computed based on the similarity of the provided query to each Web page, where the Web page with the greatest similarity score is ranked first. Many features of the Web page and the Web are used to compute the similarity score. One important feature that is directly related to the popularity of the Web page is the Web link analysis score.

The Web link analysis score of a page depends on the links to and from the page and therefore is a measure of the popularity of the page. Hence, the Web link analysis score is high for pages of general interest (having high popularity) and low for pages of specialized interest (having low popularity).

Web link analysis has the form of variants of PageRank [1], HITS [2], and SALSA [3].

PageRank is a method of global link analysis that provides a score to every page on the Web determined by the PageRank score of the pages linking to them. By using PageRank, we measure the importance of a page in terms of its global popularity; by global popularity, we mean that it is popular relative to the population of Web pages. This implies that pages that concern topics that are popular to only a small population will not receive a high PageRank score. For example, a search for *toasters* will usually present pages from Websites such as *amazon.com* when using PageRank due to the high global popularity of the site's pages. Pages from sites specializing in toasters would have high local popularity within the "toaster specialist" population, but since these specialist sites do not have a high global popularity, they would not appear within the top search results.

HITS and SALSA are link analysis methods that use only those Web pages that are relevant to the query and the set of neighboring Web pages. By performing the link analysis on only a small subset of the Web, HITS and SALSA provide a more focused link analysis than PageRank, but as in PageRank, they provide an authority score based on the global consensus of all of the chosen Web pages. This implies that if a Web page from any globally popular site appears in the list of query results, it is likely to obtain the highest rank.

Recent work [4] has shown how we can decompose the Web link graph used by PageRank into multiple popularity lists, where each popularity list comes from a certain resolution of the Web and is focused on a certain interest. The popularity list with the lowest resolution is equivalent to PageRank, since PageRank computes global popularity. The popularity lists with higher resolutions establish a version of PageRank that is local to some community within the Web. It was shown that by using multiresolution popularity lists, a significant increase in retrieval precision can be obtained over PageRank.

A recent study comparing the effectiveness of various link analysis methods [5] showed that SALSA and simple in-degree counts provided better page rankings than PageRank and HITS. In this paper, we introduce a method of computing multiresolution popularity lists for any given method of link analysis that can be represented in the form of a set of Web page relations. This includes PageRank, HITS, SALSA, and In-degree.

- *L.A.F. Park is with the School of Computing and Mathematics, University of Western Sydney, Locked Bag 1797, Penrith South DC NSW 1797, Australia. E-mail: l.park@uws.edu.au.*
- *K. Ramamohanarao is with the Department of Computer and Software Engineering, The University of Melbourne, VIC 3010, Australia. E-mail: kotagiri@unimelb.edu.au.*

*Manuscript received 17 Nov. 2008; revised 14 Jan. 2010; accepted 18 Apr. 2010; published online 10 May 2011.*
*Recommended for acceptance by B.C. Ooi.*
*For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-11-0606.*
*Digital Object Identifier no. 10.1109/TKDE.2011.107.*

1041-4347/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

| Cricket | |
|---|---|
| $a$ | 1 |
| $b$ | 2 |
| $c$ | 3 |
| $d$ | 4 |
| $e$ | 5 |
| $f$ | 6 |
| $g$ | 7 |

(a)

↔

| Indoor Cricket | |
|---|---|
| $a$ | 3 |
| $b$ | 7 |
| $c$ | 6 |
| $d$ | 1 |
| $e$ | 4 |
| $f$ | 5 |
| $g$ | 2 |

(b)

↔

| Indoor Cricket strategy | |
|---|---|
| $a$ | 2 |
| $b$ | 7 |
| $c$ | 5 |
| $d$ | 3 |
| $e$ | 4 |
| $f$ | 6 |
| $g$ | 1 |

(c)

Fig. 1. An example of the popularity of seven books (labeled $a$ to $g$) with respect to three different resolutions of the cricketing community (the lowest resolution being the general cricket community, the second resolution being the general indoor cricket community, and the third resolution being the indoor cricket community strategists). We can see that the popularity of each book is dependent on the community.

This paper provides the following major contributions:

- A generalized method of computing multiresolution community popularity lists for any given set of Web page relations (Section 2).
- An analysis of the feature vectors produced when performing our multiresolution analysis and a discussion of their relationship to the relational link matrix (Sections 2.2, 2.3, and 2.4).
- A comparison between the effectiveness of multiresolution popularity lists based on PageRank and in-degree (Section 3).
- A new open problem regarding the optimal selection of a multiresolution community popularity list when given a query (Section 3.3).

This paper will proceed as follows: Section 2 describes the multiresolution link analysis process and shows how we are able to decompose any relational link matrix into a set of feature vectors for our Web pages. Section 3 examines the effect of our multiresolution link analysis decomposition and selection on the PageRank relational link matrix and the In-degree relational link matrix.

## 2 Multiresolution Link Analysis

Link analysis on the Web is the method of using the information about which pages link to other pages, in an attempt to improve search results or gain insight into the structure of the Web. In this section, we will examine a deeper form of link analysis called multiresolution link analysis.

### 2.1 Multiresolution Communities

A community is a collection of people that have a common interest. Most people belong to many communities and most communities have more than one person belonging to them, so it is a many-to-many relationship. The specificity of the interest that binds the community determines the size of the community. If the interest is very broad, then the community will contain many members; on the other hand, if the interest is very specific, then the community will contain few members.

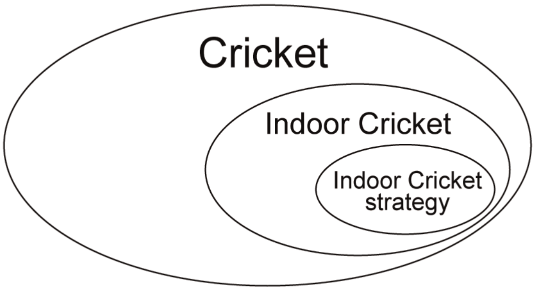


Fig. 2. Three resolutions of the Cricket community. The lowest resolution contains those interested in cricket, while the highest resolution contains those who are interested in Indoor Cricket strategy.

This implies that every community of more than one person contains subcommunities specializing in some interest within the community, where the broadest community is the set of all people and the narrowest community is the community of one person. Therefore, each community is defined by its type of interest and the resolution of the interest. The lowest resolution community would focus on issues that affect the whole population, while a higher resolution community would focus on more specialized issues. Note that even though each of the communities at each resolution has a common interest, they still have an opinion of items of which they have no interest.

#### 2.1.1 *An Example of Community Resolutions*

The example in Fig. 2, shows the Cricket[1] community, the smaller Indoor Cricket community contained within the Cricket community, and the Indoor Cricket strategy community contained within the Indoor Cricket community. The consensus of each of these communities provides us with the popularity of cricket articles $a$ to $g$ with respect to three resolutions of the cricketing community shown in Fig. 1. We will now discuss these resolutions in depth.

The lowest resolution, labeled "Cricket," contains the popularity with respect to the largest community, having the most general knowledge. If we had no knowledge of the game of cricket and we wanted basic knowledge such as how and where it is played, we would choose the paper that is most popular to this community.

If instead, we were interested in knowing more about the rules of indoor cricket (a more specific interest), the articles that are most popular to the cricket community would not be as useful. The majority of the population within the cricket community would know about indoor cricket, but not have a detailed understanding of it (most would know that it is played indoor, but not know the specifics of the rules). Therefore, to obtain knowledge of the rules of indoor cricket, we must increase the resolution of the community that we are receiving advice from (to a finer resolution), and thus choose the paper that is most popular to the indoor cricket community.

Now that we know the rules and want to start playing indoor cricket, we should equip ourselves with knowledge of the strategy that is used to try to win a game of indoor cricket. To do so, we must increase the resolution of the

[1] For those unfamiliar with the game of cricket, it is a bat and ball game with some similarities to baseball.

community, since our wanted knowledge is more specialized, and obtain popularity advice from the indoor cricket strategy community.

We can see that the cricketing community would contain those who are interested in cricket and not indoor cricket, and those that are interested in cricket and also indoor cricket. And likewise, the indoor cricketing community would contain those that are interested in the strategy of the game, and those that are not interested in the strategy and just playing for fun. Therefore, the communities are not independent, but overlap at different resolutions.

Now that we have an understanding on the different resolution that exists within communities and their associated popularity lists, we can make use of them when searching the Web. To use the lists effectively, we must be able to compute the lists, select the most appropriate list, then use the information from the selected list.

### 2.1.2 Computing the Community Popularity Lists

To compute community popularity lists at multiple resolutions, we must choose the desired resolution and then compute the basis for that resolution. For example, the lowest (broadest) resolution treats the whole population as one community; therefore, the computed basis is the best one-dimensional basis that fits the Web link data, containing the popularity of each Web page. To compute the next resolution, we compute the best two-dimensional basis for the Web link data. Each of the two basis vectors provides us with a ranked list of Web pages, that when combined, gives the best two-dimensional approximation to the link data. Each of the two lists contains the popularity of each Web page with respect to some community. By assuming more communities, we increase the dimensionality of the basis computed, and hence obtain more ranked lists, that when combined, provide a better approximation to the original link data.

### 2.1.3 Selecting a Community Popularity List

Each of the community's popularity lists computed from our multiresolution decomposition has some interest at some specific resolution. We can use the knowledge of these communities to help us further our understanding of some topic. But to do so, we must know which community has our interest as their interest.

Therefore, the next challenge is to derive a method of selecting the community popularity list that best suits our information need.

### 2.1.4 Obtaining the Information

Once we find a specific community popularity list where the associated community's interests are the same as those of our inquisition, we can the use the popularity information to determine which Web pages suit our need. To do so, we obtain a set of candidate pages that match our interest, and rank them according to the ranks provided in the community popularity list.

## 2.2 Web page Feature Vectors

Communities within the Web can be found by locating Web pages that contain similar features. Before we can measure a set of Web pages for similarity, we must first obtain the features that are to be compared.

TABLE 1
A Relational Matrix for a Set of Four Webpages, Generated Using the Inner Product of the Pages Feature Vectors

| Page | Page | | | |
|---|---|---|---|---|
| | 1 | 2 | 3 | 4 |
| 1 | 42 | 28 | 25 | 52 |
| 2 | 28 | 127 | 75 | 43 |
| 3 | 25 | 75 | 67 | 41 |
| 4 | 52 | 43 | 41 | 101 |

To show how to compute the Web page-community relationships, we will assume we know the Web page-community relationships and show how they are related to the Web page-Web page relationships.

The ideal space for Web page vectors would contain a feature for every community or interest in the Web. The value of the feature would represent the association of the Web page to the community. For example, a set of Web pages may have the following vectors:

| Page | Community | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 1 | **6** | 2 | 1 |
| 2 | **10** | 1 | 3 | **4** | 1 |
| 3 | 5 | **5** | 2 | 3 | 2 |
| 4 | 1 | 1 | **7** | 1 | **7** |

From this, we can see that page 2 has the highest association to community 1 (with an association value of 10) and community 4, page 3 has the highest association to communities 1 and 2, and page 4 has the highest association to communities 3 and 5. Using these page vectors, we are able to compute the similarity using the inner product. Therefore, the similarity between pages 1 and 2 is 28, and the similarity between pages 1 and 3 is 25. The set of all similarities is shown in Table 1. From this table, we can deduce that page 1 is most similar to page 4, page 2 is most similar to page 3. From examining the Web page vectors, we can see the similarities in that pages 1 and 4 both have a high association to community 3, and that pages 2 and 3 have high associations to communities 1 and 4.

If we represent the set of page vectors as a matrix $A$, then the table of page relationships is given in matrix form as

$$R = AA^T,$$

where $R$ is the relational matrix containing the Web page similarity values.

Unfortunately, we do not have a set of Web page vectors $A$ with community features, and there is no simple method of extracting them from the Web page set. But we are able to generate an estimate of the relational matrix $R$ that can be computed using the hyperlink graph within the set of Web pages. Using this estimate of the relational matrix $R$, we can compute the decomposition $AA^T$ to provide us with each Web pages relationship to each community. We will discuss how to perform this decomposition in Section 2.4.

## 2.3 Relational Link Matrices

The relational matrix for a set of Web pages can be derived based on the hyperlinks traversing from one page to another. A simple link-based relational matrix would contain a row and column associated with each Web page. Each element of the matrix would contain a 1 if there was a link that came from the Web page associated with the column and pointed to the Web page associated with the row, and a 0 otherwise. This simple link matrix has the problem that it can be easily biased by pages that contain many links. In this section, we will examine three forms of known relational link matrix generation methods that do not suffer from this problem, which we can use to estimate $R$.

### 2.3.1 *PageRank*

The PageRank [1] of a Web page is the probability of arriving at that page after in infinite random walk on the Web link graph. PageRank is computed using the following equation:

$$a_j = (1-\beta)\sum_{i|i\to j}\frac{a_i}{\mathrm{out}_i} + \frac{\beta}{N}, \qquad (1)$$

where $a_i$ is the PageRank score of page $i$, $\mathrm{out}_i$ is the out degree of page $i$, $N$ is the number of Web pages, and $\beta$ is the random jump probability. The last term of the equation is the probability of arriving at page $j$ after a random jump, where the probability of arriving at each page is equally likely.

Equation (1) can be rewritten in vector notation as

$$\begin{aligned}\tilde{a} &= (1-\beta)L^T D_{\mathrm{out}}^{-1}\tilde{a} + \beta\tilde{n}\\ &= \left[(1-\beta)L^T D_{\mathrm{out}}^{-1} + \beta C\right]\tilde{a},\end{aligned}$$

where each element $l_{i,j}$ of the link matrix $L$ is

$$l_{i,j} = \begin{cases}1, & \text{if page } i \text{ links to page } j,\\ 0, & \text{otherwise,}\end{cases} \qquad (2)$$

the diagonal matrix $D_{\mathrm{out}}$ contains the out degree of each page, $\tilde{n}$ is a vector containing the constants $1/N$, and $C$ is the matrix containing the set of constant column vectors $\tilde{n}$. Note that $\tilde{n} = C\tilde{a}$ due to $\sum_i a_i = 1$. The PageRank equation can be simplified to

$$\tilde{a} = R_{\mathrm{PR}}\tilde{a},$$

where the stochastic matrix $R_{\mathrm{PR}} = (1-\beta)L^T D_{\mathrm{out}}^{-1} + \beta C$. We can see that $R$ is a relational matrix that contains the information on how each Web page is related to each other Web page.

### 2.3.2 *HITS*

HITS link analysis [2] computes the rank of authorities and hubs using mutual re-enforcement. The authority and hub score of each page is computed using

$$a_j = \sum_{i|i\to j} h_i \quad h_j = \sum_{i|j\to i} a_i,$$

where $a_i$ and $h_i$ are the authority and hub score of page $i$, respectively. By substituting the hub equation into the authority equation, we can obtain a single equation for the authority computation:

$$a_j = \sum_{i|i\to j}\sum_{k|i\to k} a_k.$$

Given the link matrix $L$ from (2), we can rewrite the HITS authority equation as

$$\tilde{a} = L^T L\tilde{a} = R_{\mathrm{HITS}}\tilde{a},$$

where $R_{\mathrm{HITS}} = L^T L$ is the relation matrix between each Web page.

### 2.3.3 *SALSA*

If we examine the formation of the SALSA [3] authority computation:

$$a_j = \sum_{i|i\to j}\sum_{k|i\to k}\frac{a_k}{\mathrm{in}_k \mathrm{out}_i},$$

we can see that it is a normalized version of the HITS authority equation. The normalization gives SALSA the stochastic property as found in PageRank. We can write the SALSA authority equation in vector notation as

$$\begin{aligned}\tilde{a} &= L^T D_{\mathrm{out}}^{-1} L D_{\mathrm{in}}^{-1}\tilde{a}\\ &= R_{\mathrm{SALSA}}\tilde{a},\end{aligned}$$

where the relational matrix $R_{\mathrm{SALSA}} = L^T D_{\mathrm{out}}^{-1} L D_{\mathrm{in}}^{-1}$ contains the relationships between the set of Web pages.

We have shown that each of the link analysis methods provides an estimate of $R$, that can be reduced to the form of

$$\tilde{a} = R\tilde{a},$$

which is an eigenvalue equation with the eigenvalue equal to 1. The ranked list produced by each of these methods is the eigenvector associated with the largest eigenvalue. It was previously shown [4] that that largest eigenvector $\tilde{a}$ also satisfied the following property:

$$\tilde{a} = \mathrm{argmin}_{\tilde{v}}\left(\|R - \tilde{v}\tilde{v}^T\|\right),$$

where $R$ is a stochastic matrix. This implies that the best one-dimensional approximation of the relational matrix is given by the outer product of the first eigenvector to itself. Therefore, the eigenvalue decomposition can be used to obtain the best suited one-dimensional feature vector of the set of Web pages. This single feature would be a dominant feature across the Web and is used in PageRank, HITS, and SALSA to rank all pages within the Web.

A second factor can be produced by examining the second eigenvector, but unfortunately it is likely to contain imaginary values from the complex domain due to the relational matrix being nonsymmetric. There is no simple method of ranking complex values; therefore, analysis is left to only the first eigenvector.

## 2.4 Multiresolution Link Analysis Using Symmetric Nonnegative Matrix Factorization (SNMF)

As described in the previous section, by using the eigenvalue decomposition, we are able to compute the one-dimensional approximation of the given relational matrix. For algorithms such as PageRank, HITS, and SALSA, this one-dimensional approximation is used as the popularity or authority rank of each of the Web pages.

Symmetric nonnegative matrix factorization [6] is a method of decomposing a relational matrix into its object vectors, assuming that the relations are measured using the inner product. The decomposition has the form

$$R \approx AA^T,$$

where $R$ is our known link relational matrix and $A$ is the computed set of nonnegative factors of $R$. SNMF was previously used to decompose the PageRank link matrix into various community-based popularity lists [4]. In fact, if we decompose the matrix $R$ into its one-dimensional factors:

$$R \approx \tilde{a}\tilde{a}^T,$$

it was shown in [4] that we obtain the first eigenvector of $R$ (the best one-dimensional approximation of $R$). The relations found in the PageRank link matrix are the fraction of PageRank that is distributed from page $a$ to $b$. If we think in terms of votes, the relations are the fraction of votes page $a$ gives to page $b$. By performing SNMF on this relational matrix, we reveal the feature vectors associated with each of the Web pages that produce the relations in the relational matrix.

Using the example relations from Table 1, we can use SNMF to decompose the relations into their feature vectors. We are not able to identify the number of features that original feature vectors used; therefore, we will begin by assuming one feature. The resulting one-dimensional Web page vector is presented in the table below:

| Page | SNMF Factor |
|---|---|
| | 1 |
| 1 | 4.57 |
| 2 | 9.87 |
| 3 | 7.24 |
| 4 | 7.56 |

By computing only one factor, we have produced a global measure of relatedness, based on the relationships of all Web pages. The results say that page 2 has the highest relationship to every other page, followed by page 4, then page 3, and page 1 has the smallest relationship. By computing only one factor, we have taken all of the information in the relational matrix, and compacted it into one dimension to give a single score to each page (as also done in PageRank, HITS, and SALSA). We have done this on the assumption that the set of Web pages contains only one community.

By decomposing the relational matrix into two factors, we assume that there are two communities. The two-factor SNMF on Table 1 produces:

| Page | SNMF Factors | |
|---|---|---|
| | 1 | 2 |
| 1 | 2.00 | 5.39 |
| 2 | 11.05 | 1.00 |
| 3 | 7.17 | 2.21 |
| 4 | 2.97 | 9.45 |

Each of these factors provides us with a finer view of the Web page relations when compared to the single factor previously computed. By examining the first factor, we see that pages 2 and 3 produce high scores, while pages 1 and 4 produce low scores. The second factor shows pages 1 and 4 producing high scores and pages 2 and 3 producing low scores. Each of these factors is representative of a community interest. Therefore, these factors are consistent to our findings in Section 2.2 that pages 1 and 4 are related, and pages 2 and 3 are related.

By decomposing the relational matrix into three factors, we obtain:

| Page | SNMF Factors | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| 1 | 1.07 | 2.32 | 5.21 |
| 2 | 7.72 | 8.19 | 0.33 |
| 3 | 7.66 | 1.86 | 2.18 |
| 4 | 2.12 | 2.80 | 9.26 |

These three factors represent three community interests. We can see that the first and third factors are similar to the factors found in the two-dimensional decomposition. The second factor shows page 2 having a high score, while the remaining pages have low scores. This implies that page 2 belongs to a community that is not found in other pages (which could be due to its relatively high score of feature 1).

From these three decompositions, we have six community popularity lists covering three resolutions. The lowest resolution, providing a global measure of similarity, is equivalent to the lists we compute with PageRank, HITS, and SALSA. The remaining lists are higher resolution lists that divide the Web into communities and provide similarity values for each page to each of the communities. This implies that queries that are associated with a community would obtain more precise results when using an associated community popularity list and not an unrelated community popularity list.

In the example, we computed six community popularity lists (by computing the one-, two-, and three-factor SNMF). For a given data set, the greater the number of community popularity lists computed, the better the results as long as the list selection method is effective. For example, a system using three lists (using one- and two-factor SNMF) will always provide equal or better results than a system with one list. This is because the list of the latter system is included in the three lists of the former system. If the lists are chosen properly at query time, and the best list is the one list that both have in common, both systems will provide the same results. If the best list is not the list that both have in common, then the system with three lists will provide better results.

Note that the communities computed from the relational link matrix are not defined by the content of the Web pages, but defined by the Web links between them. The factors computed from SNMF of the link relational matrix compute the most likely features of the Web pages to produce the given link structure; we are calling the features communities, and the value associated with these features is the Web page-community relationships.

## 3 Experimentation

When seeking information and facing multiple communities, we must decide which community best suits our query. Once we have selected a community, the community will direct us to their popular answer. Now that we have shown how to obtain the multiresolution community popularity lists for a given relational matrix, we need to examine how to choose a community popularity list based on our set of candidate Web pages returned by our query.

In this section, we will examine the effect of several methods of community selection using our multiresolution community popularity lists to assist during the search process. Our experimental environment consists of the WT10G Web document collection used in TREC-9 and TREC-2001,[2] with all 100 queries (queries 451 to 550) and the associated relevance judgements. The WT10G Web document collection is a 10 gigabyte crawl of the Web, containing 1.69 million Web pages. Analysis has shown that this Web page collection has similar characteristics to the Web and therefore is a good representative sample of the Web [7]. Therefore, the WT10G Web document collection is ideal for our experimental purposes.

To perform our experiments, we generated community popularity lists for the first four resolutions using one-, two-, three-, and four-dimensional SNMF, where the lowest resolution treats the Web page collection as if it came from one community, the second resolution contains two communities, the third resolution contains three communities, and the fourth resolution contains four communities. We implemented SNMF as suggested in [6].

To evaluate our experiments, we examined precision at 10 documents (Prec10) and the number of matched queries. Precision at 10 documents is the average number of relevant Web pages found in the top 10 ranked Web pages. It has been observed that the typical Web user does not examine more than the first 10 top ranked Web pages [8]; therefore, precision at 10 is appropriate for Web retrieval evaluation. Matched queries is the number of times we issued a query and our choice of list was the best choice from our set of community popularity lists across all resolutions. Therefore, for 100 queries, a matched query value of 40 implies that the best list was chosen for 40 queries and the best list was not chosen for the remaining 60 queries. The best list for a query is defined as the list, that when used, provides the maximum precision at 10 for that query.

The information retrieval process used was:

1. When given a query, obtain a set of candidate Web pages based on the Web page content.
2. Select a community popularity list based on the distribution of the candidate Web pages in each list.
3. Rerank the set of candidate Web pages using the ranking found in the selected community popularity list.

Note that a Web search engine would use additional features to rank the pages presented to the user, and that each search engine uses different features. In our experiments, we are examining the effect that the community popularity lists have on the search results; therefore, we have simplified the ranking process to observe this effect.

The Zettair text search engine[3] was used to retrieve the set of candidate Web pages for each query.

### 3.1 Choice of Relational Matrices

It was previously shown that we can produce effective multiresolution community popularity lists using the PageRank relational matrix. Therefore, we will provide results of this method as a baseline.

Experimental evidence has shown that use of SALSA and In-degree for information retrieval produces similar precision results to each other, and better precision results than HITS and PageRank [5]. Therefore, we will focus our attention on SALSA and In-degree.

It was shown that the authority list computed using SALSA on a given set of Web pages is equivalent to computing the In-degree of the same set of Web pages [3]. We can show this by substituting the vector of in-degree values for each page into the eigenvalue equation for SALSA. By doing this, we find that the in-degree vector is the eigenvector associated with the largest eigenvalue. The SALSA eigenvalue equation is

$$\begin{aligned} R_{\text{SALSA}}\tilde{v} &= \tilde{v}\lambda \\ L^T D_{\text{out}}^{-1} L D_{\text{in}}^{-1} \tilde{v} &= \tilde{v}\lambda. \end{aligned}$$

If we choose $\tilde{v} = \tilde{d}_{\text{in}} = \text{diag}(D_{\text{in}})$, we obtain

$$\begin{aligned} L^T D_{\text{out}}^{-1} L D_{\text{in}}^{-1} \tilde{d}_{\text{in}} &= \tilde{d}_{\text{in}}\lambda \\ L^T D_{\text{out}}^{-1} L \tilde{1} &= \tilde{d}_{\text{in}}\lambda \\ L^T D_{\text{out}}^{-1} \tilde{d}_{\text{out}} &= \tilde{d}_{\text{in}}\lambda \\ L^T \tilde{1} &= \tilde{d}_{\text{in}}\lambda \\ \tilde{d}_{\text{in}} &= \tilde{d}_{\text{in}}\lambda, \end{aligned}$$

where $\tilde{d}_{\text{out}} = \text{diag}(D_{\text{out}})$. Therefore, $\tilde{d}_{\text{in}}$ is an eigenvector of $L^T D_{\text{out}}^{-1} L D_{\text{in}}^{-1}$, with corresponding eigenvalue $\lambda = 1$.

Since $R_{\text{SALSA}}$ is a stochastic matrix, its largest eigenvalue is 1. Therefore, the In-degree vector is the eigenvector associated with the largest eigenvalue of $R_{\text{SALSA}}$, and hence the steady state solution when using SALSA.

The link matrix used with SALSA is the set of pages associated with the query and their neighbors, and therefore query dependent. If instead we used the entire collection of Web pages (removing the dependence on the query), the steady state solution to SALSA is exactly the in-degree of each page. Therefore, we define $R_{\text{in-degree}}$ as

[2] http://trec.nist.gov.

[3] http://www.seg.rmit.edu.au/zettair/.

TABLE 2
The Precision after 10 Documents (Prec10) Obtained Using the 10 Computed Popularity Lists Computed from the PageRank Relational Matrix

| Resolution | 1 | 2 | | 3 | | | 4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Community | 1 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 3 | 4 |
| Prec10 | 0.115 | 0.128 | 0.115 | 0.133 | 0.129 | 0.115 | 0.134 | 0.117 | 0.139 | 0.124 |
| PageRank ratio | 1.000 | 1.115 | 1.000 | 1.159 | 1.123 | 1.000 | 1.168 | 1.017 | 1.212 | 1.079 |
| In-degree ratio | 0.904 | 1.008 | 0.904 | 1.048 | 1.016 | 0.904 | 1.056 | 0.920 | 1.096 | 0.976 |
| Matched queries | 37 | 40 | 38 | 39 | 38 | 35 | 39 | 33 | 40 | 36 |

*The PageRank and In-degree ratios are the ratio between the chosen popularity list and PageRank or In-degree, respectively. We can see that the first list provides the same precision as PageRank and every other list is greater. The matched queries row shows the number of queries in which the associated list provided the best precision.*

$$R_{\text{in-degree}} := R_{\text{SALSA}}, \text{ where } L \text{ contains all Web pages.}$$

For our link analysis, we will be using the entire set of Web pages for our link matrix $L$. This implies that even though we are using the SALSA relational matrix, we are unable to show results for SALSA due to our community popularity list computation being query independent. To make it clear that we are computing community relationships based on the whole relational link matrix, we have labeled our experiments as In-degree and not SALSA.

### 3.2 Query Independent Selection

To begin our investigation, we will examine the effect of each community popularity list individually. Since we are not basing our choice of community popularity list on the query, our list selection is query independent. Table 2 provides a baseline measure, showing the precision at 10 and the number of matched queries for each community popularity list, computed using the PageRank link metric. The columns of the table are divided to show the list's community number and resolution. At resolution 1, the community popularity list provides the same Web page rankings as PageRank. Included in this table are the PageRank ratio and In-degree ratio. These values are the precision at 10 documents for the selected list divided by the precision at 10 documents when using either PageRank or In-degree on the set of 100 queries. We can see that the PageRank ratio for the first resolution is 1, showing their equivalence. The results show that each of the 10 community popularity lists has a PageRank ratio of 1 or greater, implying that by picking one of the 10 lists at random, we should perform at least as good as PageRank, if not better.

Also included in the table is a matched queries number. This number identifies how many queries achieved their greatest precision when using the associated list. For example, 37 of the queries obtained the greatest precision when using list 1 from community 1. There may be multiple lists that provide the greatest precision for a query; therefore, the sum of the matched queries does not have to sum to 100. It is interesting to note that higher matched queries does not imply a greater precision. If we compare resolution 4, community 2 to resolution 1, community 1, we see that the former has a greater precision at 10 documents, while the latter has a greater number of matched queries. This difference implies that by using resolution 4, community 2, the queries that we matched have more relevant Web pages in the top 10 than those that were matched using resolution 1, community 1.

Table 3 displays the precision at 10 documents and matched query results when using the in-degree link-based relational matrix to build the multiresolution community popularity lists. At resolution 1, the community popularity list provides the same Web page rankings as when using in-degree. This table includes the PageRank ratio and In-degree ratio as in Table 2. We can see that the In-degree ratio is 1 for the first community popularity list, reflecting the equivalence of In-degree and the first community popularity list generated using the In-degree relational matrix.

It is interesting to see that the precision at 10 documents using the in-degree link metric provides higher precision for the majority of the community popularity lists when compared to the precision when using the PageRank link metric (eight of the 10 lists), while the reverse is shown when examining the matched queries (the PageRank method has a greater number of matched queries for seven of the 10 lists).

TABLE 3
The Precision after 10 Documents (Prec10) Obtained Using the 10 Computed In-Degree Popularity Lists

| Resolution | 1 | 2 | | 3 | | | 4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Community | 1 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 3 | 4 |
| Prec10 | 0.128 | 0.140 | 0.118 | 0.120 | 0.140 | 0.122 | 0.146 | 0.132 | 0.102 | 0.138 |
| PageRank ratio | 1.106 | 1.212 | 1.027 | 1.044 | 1.212 | 1.062 | 1.265 | 1.141 | 0.885 | 1.195 |
| In-degree ratio | 1.000 | 1.096 | 0.928 | 0.944 | 1.096 | 0.960 | 1.144 | 1.032 | 0.800 | 1.080 |
| Matched queries | 25 | 33 | 30 | 30 | 38 | 31 | 38 | 39 | 29 | 36 |

*The PageRank and In-degree ratios are the ratio between the chosen popularity list and either PageRank or In-degree. The matched queries row shows the number of queries in which the associated list provided the best precision.*

TABLE 4
The Precision after 10 Documents (Prec10) Obtained Using the Best Community Popularity List for Each Query

| Selection | Oracle method | |
|---|---|---|
| | PageRank | In-degree |
| Prec10 | 0.237 | 0.268 |
| PageRank ratio | 2.062 | 2.327 |
| In-degree ratio | 1.864 | 2.104 |
| Matched queries | 100 | 100 |

*This score shows the best possible score that can be obtained using our 10 PageRank multiresolution community popularity lists and the 10 In-degree multiresolution community popularity lists. We can see that the PageRank oracle method provides a 106.2 percent improvement over PageRank and the In-degree oracle method provides a 110.4 percent improvement over In-degree.*

## 3.3 Oracle Selection

In this section, we move on to examining the effect on precision of choosing a community popularity list based on individual queries. The first method of list selection we will examine is the oracle method. The oracle method chooses the best list for the query based on the precision that list will provide. This is not a method we can use in practice, since we would not have prior knowledge of the precision for each list, but it is presented in this paper to provide an upper bound for our experimental results. Using this upper bound, we obtain a better understanding as to how well our proposed query-based community popularity list selection methods function. The results of using this oracle method using the PageRank and In-degree relational matrices are shown in Table 4.

By computing the multiresolution community popularity lists from the PageRank relational matrix, we find that we are able to achieve a possible 106 percent increase in precision over the PageRank baseline and an 86 percent increase in precision over the in-degree baseline.

By computing the multiresolution community popularity lists from the in-degree relational matrix, the oracle selection method shows that we have the potential to achieve a 133 percent increase over the PageRank baseline and 110 percent increase in precision over the In-degree baseline. For the remainder of this paper, we will focus our attention on finding a query dependent list selection metric that will help us to utilize the benefits of our community popularity lists.

## 3.4 Rank-Based Selection

To effectively use the set of community popularity lists, we must be able to choose the appropriate list at query time to use for the given query. To do so, we first compute the set of candidate Web pages that are most similar to the query (by using a text retrieval system and choosing the top ranked Web pages for the given query), then we compute which community popularity list is the most similar to the set of candidate Web pages.

Once a list has been chosen, the set of candidate Web pages are ranked according to their rank in the chosen community popularity list. In this section, we will examine the effect of the systems precision when community popularity lists are chosen based on the rank of the set of candidate Web pages within the list. Therefore, once the set of candidate Web pages have been selected, we compute a score for every community popularity list; the list that achieves the highest score is selected and used to rank the candidate Web pages.

TABLE 5
An Example Set of Webpages Returned by a Query and Their Associated Score and Ranks from Three Community Popularity Lists

| Web Page | List 1 | | List 2 | | List 3 | |
|---|---|---|---|---|---|---|
| | Score | Rank | Score | Rank | Score | Rank |
| 1 | 0.07 | 10 | 0.04 | 13 | 0.13 | 7 |
| 2 | 0.02 | 32 | 0.03 | 17 | 0.08 | 24 |
| 3 | 0.12 | 5 | 0.09 | 11 | 0.10 | 18 |
| 4 | 0.09 | 9 | 0.08 | 12 | 0.05 | 42 |

### 3.4.1 Simple Rank-Based Metrics

Before we describe the metrics used to score the lists, we must first discuss the desirable properties of a selected list. Each list provides an ordered set of all of the Web pages, ordered according to some community interest. Therefore,

- if a set of pages were directly related to the community interest, they would appear near the top of the list,
- if a set of pages were not directly related to the community interest, but they were related in some sense, they would appear nearby each other in the list, and
- if a set of pages were totally unrelated to the community popularity list, they would appear scattered throughout the list.

From this description of desirable properties, it seems that a list is appropriate for a query if the mean rank of the set of pages within the list is minimal. It also suggests that a list is desirable if the standard deviation of the ranks is minimal for a given set of pages. To account for these properties, we have constructed four metrics for list selection:

$$\begin{aligned}\min(\text{mean}) &:= \text{argmin}_c(\text{mean}_d(R_{c,d}))\\ \min(\text{sd}) &:= \text{argmin}_c(\text{sd}_d(R_{c,d}))\\ \max(\text{imean}) &:= \text{argmax}_c(\text{mean}_d(R^{-1}_{c,d}))\\ \min(\text{isd}) &:= \text{argmin}_c(\text{mean}_d(R^{-1}_{c,d}))\end{aligned}$$

where $R_{c,d}$ is the rank of the $d$th candidate Web page in the $c$th community popularity list and $R^{-1}_{c,d}$ is its reciprocal. Therefore, $\min(\text{mean})$ returns the number of the list that provides the minimum mean rank of the set of candidate Web pages.

### 3.4.2 Combined Rank-Based Metrics

We will also use four combination metrics, that take into account both the mean and standard deviation of the ranks:

$$\begin{aligned}\min(\text{mean.sd}) &:= \text{argmin}_c(\text{mean}_d(R_{c,d}) \times \text{sd}_j(R_{c,d}))\\ \min(\text{mean.isd}) &:= \text{argmin}_c\left(\text{mean}_d(R_{c,d}) \times \text{sd}_j\left(R^{-1}_{c,d}\right)\right)\\ \max(\text{imean/sd}) &:= \text{argmax}_c\left(\text{mean}_d\left(R^{-1}_{c,d}\right)/\text{sd}_j(R_{c,d})\right)\\ \max(\text{imean/isd}) &:= \text{argmax}_c\left(\text{mean}_d\left(R^{-1}_{c,d}\right)/\text{sd}_j\left(R^{-1}_{c,d}\right)\right)\end{aligned}$$

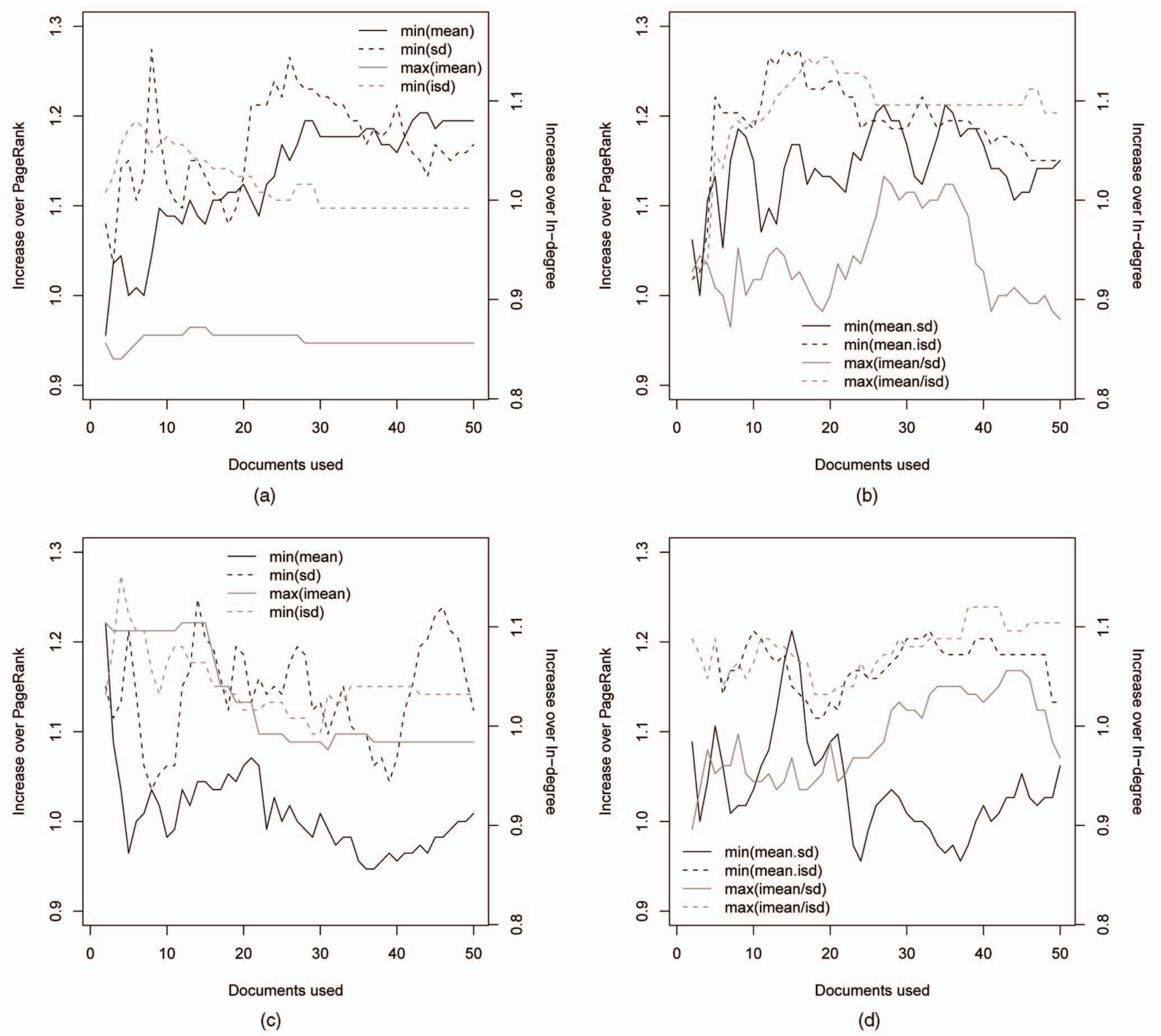


Fig. 3. Increase in precision at 10 documents when using the rank of the candidate Web pages in the community popularity lists to select a list. The top two plots show results computed using the PageRank relational matrix, while the bottom two plots show results computed using the in-degree relational matrix. The *x*-axis shows the number of candidate Web pages used during the list selection computation. The left *y*-axis shows the increase in precision at 10 documents when compared to ranking the candidate Web pages with PageRank. The right *y*-axis shows the increase in precision at 10 when compared to ranking the candidate Web pages with in-degree. (a) Simple rank metrics on PageRank relations. (b) Combined rank metrics on PageRank relations. (c) Simple rank metrics on In-degree relations. (d) Combined rank metrics on In-degree relations.

For each of these metrics, we examined the effect of using the top $n$ candidate Web pages as the set $R_{c,d}$, where $n$ ranges from 2 to 50 (note that $n = 1$ was excluded since the standard deviation of one value is not defined).

### 3.4.3 Example of Rank-Based Metric Use

Before examining the results, we will provide an example of how the metrics are used to choose a community popularity list. After issuing a query, the four Web pages in Table 5 were returned by the system waiting to be ranked. To rank the Web pages, we must choose a community popularity list and then rank the Web pages, respectively. If we use $\min(\text{mean.sd})$ to select the community popularity list, we must first choose the number of candidate Web pages that are to take part. For this small example, we choose three. We then proceed to compute the mean community rank of each of the top three ranks from each list. From List 1, we obtain $(10 + 32 + 5)/3 = 15.667$; from List 2, we obtain $(13 + 17 + 11)/3 = 13.667$; and from List 3, we obtain $(7 + 24 + 18)/3 = 16.333$. So from these results, we want the list that provides the minimum mean, which is List 2. The Web pages would then be ordered by List 2 and presented to the user. If we used $\min(\text{sd})$, we would compute the results in the same way, but use standard deviation instead of mean. This would give us a score of 14.364 for List 1, 3.055 for List 2, and 8.621 for List 3. Therefore, using this metric, we would also choose List 2. For $\max(\text{imean})$ and $\min(\text{isd})$, we would follow the same process, but use the reciprocal of the ranks, rather than the ranks, for the mean and standard deviation computation.

When using the combined metrics, we simply combine the scores from the simple metrics to obtain the list scores.

### 3.4.4 Rank-Based Metric Results

A set of plots showing the results of using the metrics with the given ranges of candidate Web pages is shown in Fig. 3. From these results, we can see from Fig. 3a that simply choosing the list that has the minimum mean rank provides a steady increase in precision as the number of candidate Web pages used increases. The results for the minimum standard deviation fluctuate, therefore making it hard to choose the number of candidate Web pages that provide the best precision. The remaining simple metrics (max(imean) and min(isd)) both stabilize at precisions lower than the In-degree precision, and are therefore not effective.

Two of the combined metrics (min(mean.isd) and max(imean/isd)) used on the PageRank relation community popularity lists in Fig. 3b peak in the 10 to 20 Web page range, providing at least a 20 percent increase over PageRank and at least a 10 percent increase over In-degree.

From the In-degree community popularity lists, we can see that max(imean) in Fig. 3c is stable in the range of 1 to 15 Web pages providing an increase of 10 percent over the In-degree precision. The min(isd) metric also stabilizes as the number of Web pages used nears 50. For the combined metrics in Fig. 3d, we see that max(imean/isd) begins to stabilize near the 50 Web page mark providing an increase of about 10 percent over the In-degree precision. The remaining metrics fluctuate, therefore making it hard to choose the right number of candidate Web pages to use for list selection.

## 3.5 Score-Based Selection

When generating community popularity lists for a given relational matrix, we obtain a set of scores for each of the Web pages. These scores are then used to rank the set of Web pages. In the previous section, we examined the effect of choosing a community popularity list, based on the ranks of the candidate Web pages within the list. In this section, we will examine the effect of the choosing a list based on the scores of the candidate Web pages within the list.

### 3.5.1 Simple Score-Based Metrics

Within each list:

- The Web pages with the greatest score are the pages that are most representative of the list. Therefore, when examining the set of lists, we should choose those that assign the set of candidate Web pages the greatest score.
- Webpages that have similar scores within a list are likely to be related; therefore, we should also seek the list that has our set of candidate Web pages with similar scores.

Based on these qualities, we have chosen four simple metrics for choosing a community popularity list, based on the mean and standard deviation:

$$\begin{aligned}
\max(\text{mean}) &:= \text{argmax}_c(\text{mean}_d(S_{c,d}))\\
\min(\text{sd}) &:= \text{argmin}_c(\text{sd}_d(S_{c,d}))\\
\min(\text{imean}) &:= \text{argmin}_c\left(\text{mean}_d\left(S_{c,d}^{-1}\right)\right)\\
\min(\text{isd}) &:= \text{argmin}_c\left(\text{mean}_d\left(S_{c,d}^{-1}\right)\right)
\end{aligned}$$

where $S_{c,d}$ is the score of the $d$th candidate Web page in the $c$th community popularity list and $S_{c,d}^{-1}$ is its reciprocal. Therefore, $\max(\text{mean})$ returns the number of the list that provides the maximum mean score of the set of candidate Web pages.

### 3.5.2 Combined Score-Based Metrics

We will also use four combination metrics, that take into account both the mean and standard deviation of the scores:

$$\begin{aligned}
\max(\text{mean/sd}) &:= \text{argmax}_c(\text{mean}_d(S_{c,d})/\text{sd}_d(S_{c,d}))\\
\max(\text{mean/isd}) &:= \text{argmax}_c\left(\text{mean}_d(S_{c,d})/\text{sd}_d\left(S_{c,d}^{-1}\right)\right)\\
\min(\text{imean.sd}) &:= \text{argmin}_c\left(\text{mean}_d\left(R_{c,d}^{-1}\right) \times \text{sd}_d(R_{c,d})\right)\\
\min(\text{imean.isd}) &:= \text{argmin}_c\left(\text{mean}_d\left(R_{c,d}^{-1}\right) \times \text{sd}_d\left(R_{c,d}^{-1}\right)\right)
\end{aligned}$$

### 3.5.3 Example of Score-Based Metric Use

Before proceeding with our experimental results, we will provide a simple example of the use of our score-based metrics for community popularity list selection. After issuing a query, the four Web pages in Table 5 were returned by the system waiting to be ranked. To rank the Web pages, we must choose a community popularity list and then rank the Web pages, respectively. If we use $\max(\text{mean})$, we must first select the number of candidate Web pages that are to take part. In this example, we choose three. We then proceed to compute the mean of each of the top three scores from each list. From List 1, we obtain $(0.07 + 0.02 + 0.12)/3 = 0.07$; from List 2, we obtain $(0.04 + 0.03 + 0.09)/3 = 0.053$; and from List 3, we obtain $(0.13 + 0.08 + 0.10)/3 = 0.103$. From this set of score, we choose the list that provides the maximum mean score, being List 3. The Web pages are then presented to the user in the order from List 3.

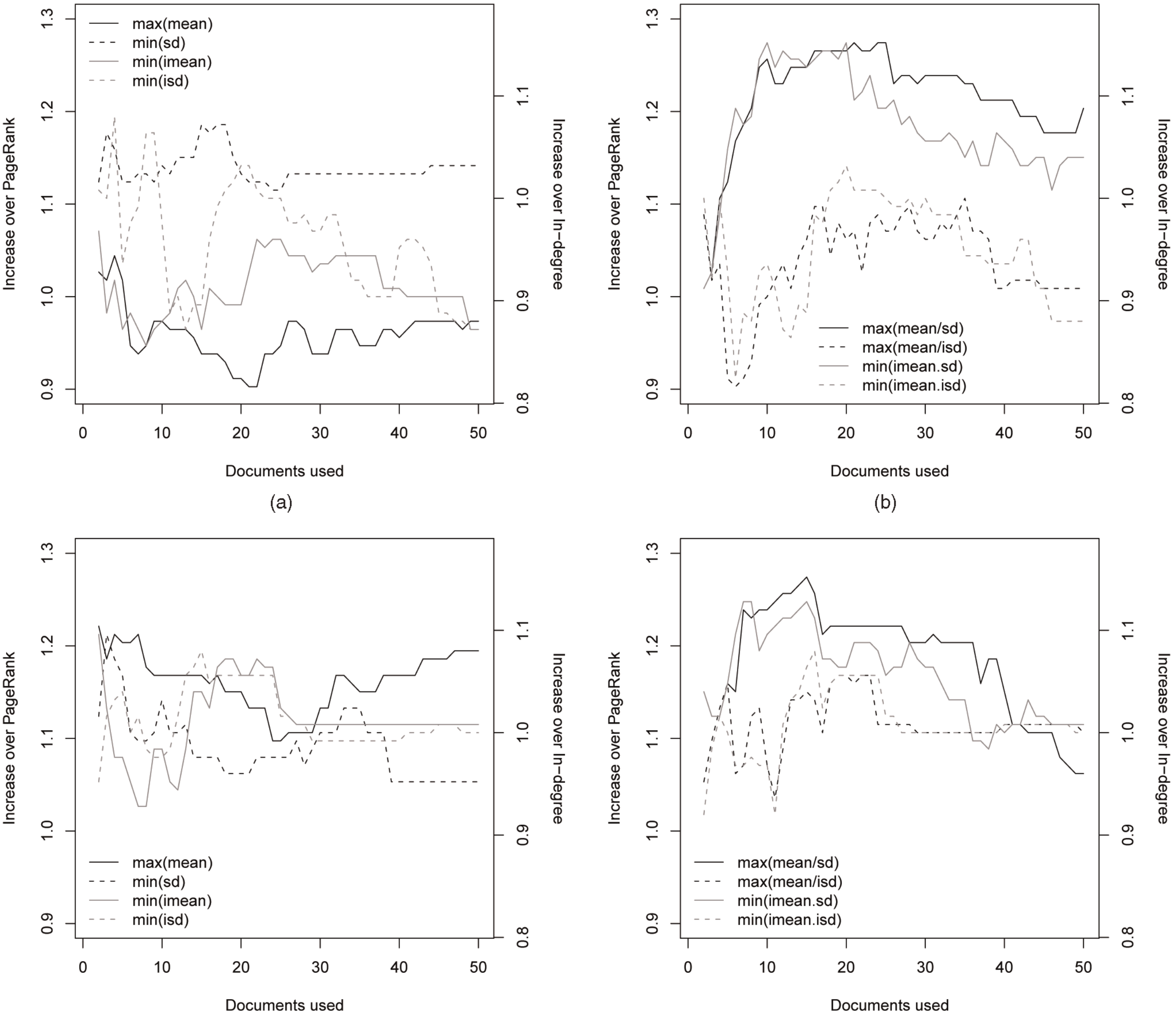
If instead we chose to use $\min(\text{sd})$, we would obtain 0.05 for List 1, 0.032 for List 2, and 0.025 for List 3. We then select the list that provides the minimum standard deviation, which again is List 3. List 3 is then used to order the Web page results. The same process is applied when using $\min(\text{imean})$ and $\min(\text{isd})$ where the reciprocal of each score is used in the place of each score.

If using a combined metric, we simply compute the score for each of its components and combine them to obtain the list score.

### 3.5.4 Score-Based Metric Results

As with our rank-based experiments, we examined the effect of using the top $n$ candidate Web pages as the set $S_{c,d}$ for each of the metrics, where $n$ ranges from 2 to 50. A set of plots showing the results of using the described metrics on the PageRank and In-degree community popularity lists is provided in Fig. 4.

From the first set of plots (Fig. 4a), we can see that only the min(sd) metric from this set of simple metrics using the



(a)

(b)

(c)

(d)

Fig. 4. Increase in precision at 10 documents when using the score of the candidate Web pages in the community popularity lists to select a list. The top two plots show results computed using the PageRank relational matrix, while the bottom two plots show results computed using the in-degree relational matrix. The *x*-axis shows the number of candidate Web pages used during the list selection computation. The left *y*-axis shows the increase in precision at 10 documents when compared to ranking the candidate Web pages with PageRank. The right *y*-axis shows the increase in precision at 10 when compared to ranking the candidate Web pages with in-degree. (a) Simple score metrics on PageRank relations. (b) Combined score metrics on PageRank relations. (c) Simple score metrics on In-degree relations. (d) Combined score metrics on In-degree relations.

PageRank community popularity lists provides stable results. If we examine the combined metrics in Fig. 4b, we can see that max(mean/sd) and min(imean.sd) provide large increases in precision (of 25 percent over PageRank and 15 percent over In-degree) for the 10 to 20 Web page range.

When using the set of simple metrics on the In-degree community popularity lists (decomposed from the In-degree relations) in Fig. 4c, we can see that max(mean) is stable for the 1 to 10 Web page range, while the remaining simple metrics fluctuate for most of the Web page range.

When observing the combined metrics in Fig. 4d, we notice that the max(mean/sd) and the min(imean.sd) metrics provide a high increase in precision around the 10 to 20 Web page range, as they did on the PageRank community popularity lists.

### 3.6 Complexity

To use our multiresolution community popularity lists, we must first compute the lists offline, and then as queries arrive, select a list and apply it to the query.

To compute the community popularity lists, we first build the relational matrix and then apply SNMF. Construction of the PageRank relational matrix requires a simple pass through the data set since it is simply a weighted link matrix. The In-degree relational matrix is the combination of two weighted link matrices. Therefore, the computation requires a matrix multiply between two $N \times N$ matrices, where $N$ is the number of Web pages. For our data set containing 1.69 million Web pages, this multiplication took about 4 hours. Note that if the Web pages change, the relational matrix can be updated by including the differences, and therefore the whole matrix does not need to be recomputed. To speed up this process, an investigation could be made into performing the SNMF before multiplying, taking into account that the SNMF needs to incorporate the multiplication. The one-factor SNMF decomposition runs in a similar time to an eigenvalue decomposition used when computing PageRank. The $n$-factor SNMF decomposition time increases linearly with $n$. Therefore, computation of the 10 community popularity lists in our experiments took approximately 10 times the time of computing the single PageRank list. Each of the set of community popularity lists is computed independently; therefore, our four sets of lists could have been computed in parallel taking only four times the time of computing the single PageRank list.

Concerning query times, the community popularity lists can be stored in an inverted index and the scores computed using an accumulator, just as Web page scores are computed from the text they contain. Since the number of elements being processed is only a fraction of the number of elements processed during the candidate Web page computation, the time required for list selection and Web page reranking would be the same fraction the time required to compute the candidate Web page scores.

## 4 Related Work

The use of multiresolution community popularity lists presented in this paper is novel in that we do not have to predefine the communities, and that we are able to identify the community interests at the general resolution and also at finer more specific resolutions.

Topic sensitive PageRank [9] and personalized PageRank [10] are similar to the work presented in this paper, in that they produce PageRank vectors that are biased toward a collection of Web pages by altering the PageRank jump probability, but they are different in that the biasing must be predefined. Personalized PageRank does this by examining the Web user usage patterns and Topic sensitive PageRank does this by first identifying collections of Web pages belonging to the same topic (using the Open Directory Project categories) and then using these collections to produce topic biased PageRank vectors. The multiresolution community popularity lists produced in this paper are generated based on the structure of the Web and therefore do not need maintenance of predefined topic categories and do not need to keep track of Web usage.

A variation of HITS, called TOPHITS [11], uses a relational tensor to store link and anchor text relations. By using a tensor decomposition, similar to the singular value decomposition, authority and hub vectors are computed along with anchor text vectors. The anchor text vectors are then used to determine the topic focus of each authority and hub. Again, this work is similar to ours in that various vectors are computed for ranking, but the work differs in that the ranking vectors are assigned topics, where the topics are extracted from Web anchor text. Our multiresolution community vectors are not associated with predefined text and therefore are not limited to a subset of queries.

## 5 Conclusion

Current link analysis methods used in Web search engines seek to obtain a single list of popularity or authority scores in order to rank Web pages more adequately. By using only a single list, we find that search engines are able to respond well to general queries in which the majority of the Web community would have an opinion of. Unfortunately, it also means that search engines find it difficult to supply information regarding specific queries that the majority of the Web would not have an opinion of.

Previous work on multiresolution link analysis has shown that we are able to extend PageRank to produce multiple lists that center around various portions of the Web. Each of the lists provides a rank of every page on of the Web relative to some community.

In this paper, we further examined multiresolution link analysis using symmetric nonnegative matrix factorization. We showed how methods such as PageRank, HITS, and SALSA treat some variant of the Web link matrix as a Web page relational matrix. The list that each method produces is the one-dimensional decomposition of the relational matrix. Hence, each method is attempting to decompose the Web page relational matrix into its Web page feature vectors. We were therefore able to show how to perform a generalized form of multiresolution link analysis that is able to use any Web page relational matrix.

To show the benefits of our generalized model, we computed a set of community popularity lists using the PageRank relational matrix and the In-degree relational matrix. Using various list selection methods, we showed that we were able to achieve a 25 percent increase over PageRank and a 13 percent increase over In-degree. We also showed that when using the best list selection method, we can obtain a potential 132 percent increase in precision over PageRank and a 110 percent increase in precision over In-degree. This large gap between our results and the optimal results shows that there is much room for improvement. This provides the Information Retrieval community with a new open problem of finding the optimal metric to select the best community popularity lists for each query in order to benefit from these large increases of precision.

## Acknowledgments

The authors would like the thank Ms. Quynh-Chi Nguyen for the use of her programming skills, and the Australian Research Council for their funding of this work.

## References

[1] L. Page, S. Brin, R. Motwani, and T. Winograd, "The Pagerank Citation Ranking: Bringing Order to the Web," technical report, Stanford Digital Library Technologies Project, www.citeseer.ist.psu.edu/page98pagerank.html, 1998.

[2] J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *J. ACM,* vol. 46, no. 5, pp. 604-632, 1999.

[3] R. Lempel and S. Moran, "SALSA: The Stochastic Approach for Link-Structure Analysis," *ACM Trans. Information Systems,* vol. 19, no. 2, pp. 131-160, Apr. 2001.

[4] L.A.F. Park and K. Ramamohanarao, "Mining Web Multi-Resolution Community-Based Popularity for Information Retrieval," *Proc. ACM Conf. Information and Knowledge Management,* pp. 545-552, Nov. 2007.

[5] M. Najork, "Comparing the Effectiveness of HITS and SALSA," *Proc. ACM 16th Conf. Information and Knowledge Management,* M.J. Silva, A.A.F. Laender, R. Baeza-Yates, D.L. McGuinness, B. Olstad, Ø.H. Olsen, and A.O. Falcão, eds., pp. 157-164, 2007.

[6] C. Ding, X. He, and H.D. Simon, "On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering," *Proc. SIAM Int'l Conf. Data Mining (SDM '05),* pp. 606-610, Apr. 2005.

[7] I. Soboroff, "Does WT10g Look Like the Web?" *SIGIR '02: Proc. 25th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 423-424, 2002.

[8] Y. Zhang, L.A.F. Park, and A. Moffat, "Click-Based Evidence for Decaying Weight Distributions in Search Effectiveness Metrics," *Information Retrieval,* vol. 13, pp. 1-24, 2010.

[9] T.H. Haveliwala, "Topic-Sensitive Pagerank," *WWW '02: Proc. 11th Int'l Conf. World Wide Web,* pp. 517-526, 2002.

[10] G. Jeh and J. Widom, "Scaling Personalized Web Search," *WWW '03: Proc. 12th Int'l Conf. World Wide Web,* pp. 271-279, 2003.

[11] T.G. Kolda, B.W. Bader, and J.P. Kenny, "Higher-Order Web Link Analysis Using Multilinear Algebra," *Proc. Fifth IEEE Int'l Conf. Data Mining,* pp. 242-249, 2005.

**Laurence A.F. Park** received the BE (Hons.) and BSc degrees from the University of Melbourne, Australia, in 2000, and the PhD degree from the University of Melbourne in 2004. He joined the Computer Science Department at the University of Melbourne as a research fellow in 2004, and was promoted to senior research fellow in 2008. He joined the School of Computing and Mathematics at the University of Western Sydney as a lecturer in computational mathematics and statistics in 2009, where he is currently investigating methods of large-scale data mining and machine learning. During this time, he has been made an honorary senior fellow of the University of Melbourne. He is a member of the IEEE Signal Processing Society.

**Kotagiri Ramamohanarao** received the PhD degree from Monash University. He was awarded the Alexander von Humboldt Fellowship in 1983. He has been at the University Melbourne since 1980 and was appointed a professor in computer science in 1989. He held several senior positions including the head of Computer Science and Software Engineering, the head of the School of Electrical Engineering and Computer Science at the University of Melbourne, and the research director for the Cooperative Research Centre for Intelligent Decision Systems. He served on the Editorial Boards of the Computer Journal. At present, he is on the Editorial Boards for *Universal Computer Science*, and *Data Mining and VLDB (Very Large Data Bases) Journal*. He was the program cochair for VLDB, PAKDD, DASFAA, and DOOD conferences. He is a steering committee member of IEEE ICDM, PAKDD, and DASFAA. He received distinguished contribution award for Data Mining. He is a fellow of the Institute of Engineers Australia, a fellow of Australian Academy Technological Sciences and Engineering, and a fellow of Australian Academy of Science. He was awarded Distinguished Contribution Award in 2009 by the Computing Research and Education Association of Australasia. He is a member of the IEEE.