Information Retrieval and Web Search Engines

Lecture 12: Link Analysis
July 8, 2010

Wolf-Tilo Balke and Joachim Selke
Institut für Informationssysteme
Technische Universität Braunschweig
Task: Try different classification algorithms

1. Choose three topics
   • Criteria?

2. Split collection into training and test set (70%—30%)

3. For each topic and each document representation (TF, TF-IDF, log entropy), evaluate the performance of different classifiers (precision, recall, balanced F measure)
   • Rocchio
   • Naïve Bayes
   • kNN, for at least three different values of k
   • (Soft-margin) SVM, for two kernels: linear and RBF ($\gamma = 1$)
Topic selection

The ten most frequent topics are...

- **earn (18.5%)** (reports about earnings)
- **acq (11.3%)** (company acquisitions)
- **money-fx (3.7%)** (currency and foreign exchange)
- grain (2.9%)
- crude (2.9%)
- trade (2.6%)
- interest (2.4%)
- ship (1.4%)
- wheat (1.4%)
- corn (1.2%)
**Performance of Rocchio on TF data:**

<table>
<thead>
<tr>
<th></th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.8%</td>
<td>79.8%</td>
<td>81.6%</td>
<td>80.7%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>19.1%</td>
<td>79.8%</td>
<td>82.0%</td>
<td>80.9%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>28.2%</td>
<td>37.6%</td>
<td>92.8%</td>
<td>53.6%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>28.6%</td>
<td>36.1%</td>
<td>93.0%</td>
<td>52.0%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>14.4%</td>
<td>24.8%</td>
<td>94.3%</td>
<td>39.3%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>14.2%</td>
<td>21.9%</td>
<td>89.4%</td>
<td>35.2%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>20.6%</td>
<td>45.9%</td>
<td>88.1%</td>
<td>56.0%</td>
<td>0.2 s</td>
</tr>
</tbody>
</table>
Performance of **Rocchio** on **TF–IDF** data:

<table>
<thead>
<tr>
<th></th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>17.9%</td>
<td>85.9%</td>
<td>83.4%</td>
<td>84.7%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>18.4%</td>
<td>85.5%</td>
<td>84.4%</td>
<td>84.9%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>24.2%</td>
<td>45.5%</td>
<td>96.3%</td>
<td>61.8%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>23.8%</td>
<td>44.4%</td>
<td>95.3%</td>
<td>60.6%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>9.5%</td>
<td>39.0%</td>
<td>97.9%</td>
<td>55.8%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>9.8%</td>
<td>34.1%</td>
<td>95.6%</td>
<td>50.3%</td>
<td></td>
</tr>
<tr>
<td><strong>Average (test)</strong></td>
<td>17.3%</td>
<td>54.7%</td>
<td>91.8%</td>
<td>65.3%</td>
<td>0.2 s</td>
</tr>
</tbody>
</table>
Performance of **Rocchio** on **log entropy** data:

<table>
<thead>
<tr>
<th></th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.3%</td>
<td>85.4%</td>
<td>85.1%</td>
<td>85.3%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>18.8%</td>
<td>85.3%</td>
<td>86.1%</td>
<td>85.7%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>24.1%</td>
<td>45.7%</td>
<td>96.2%</td>
<td>62.0%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>23.9%</td>
<td>44.3%</td>
<td>95.5%</td>
<td>60.5%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>9.8%</td>
<td>37.9%</td>
<td>97.6%</td>
<td>54.5%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>10.0%</td>
<td>33.8%</td>
<td>96.5%</td>
<td>50.1%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>17.6%</td>
<td>54.5%</td>
<td>92.7%</td>
<td>65.4%</td>
<td>0.2 s</td>
</tr>
</tbody>
</table>
Performance of naïve Bayes on ALL data:

- Ignoring all negative events
  ("term x does not occurs in document y")
- Using linear smoothing (weight 0.01)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>17.2%</td>
<td>90.0%</td>
<td>84.3%</td>
<td>87.1%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>17.0%</td>
<td>88.0%</td>
<td>80.5%</td>
<td>84.1%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>15.6%</td>
<td>72.6%</td>
<td>98.8%</td>
<td>83.7%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>16.7%</td>
<td>55.2%</td>
<td>83.3%</td>
<td>66.4%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>14.2%</td>
<td>26.7%</td>
<td>99.3%</td>
<td>42.1%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>14.9%</td>
<td>22.8%</td>
<td>97.3%</td>
<td>36.9%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>16.2%</td>
<td>55.3%</td>
<td>87.0%</td>
<td>62.5%</td>
<td>0.4 s</td>
</tr>
</tbody>
</table>
Performance of 1-NN on TF data:

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.3%</td>
<td>99.9%</td>
<td>99.4%</td>
<td>99.7%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>19.7%</td>
<td>85.4%</td>
<td>90.1%</td>
<td>87.7%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>11.4%</td>
<td>100.0%</td>
<td>99.8%</td>
<td>99.9%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>10.7%</td>
<td>73.5%</td>
<td>71.2%</td>
<td>72.3%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>3.8%</td>
<td>99.3%</td>
<td>98.8%</td>
<td>99.0%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>4.1%</td>
<td>66.8%</td>
<td>78.3%</td>
<td>72.1%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>11.5%</td>
<td>75.2%</td>
<td>79.9%</td>
<td>77.4%</td>
<td>104 s</td>
</tr>
</tbody>
</table>
Performance of 1-NN on TF–IDF data:

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.3%</td>
<td>99.9%</td>
<td>99.4%</td>
<td>99.7%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>21.0%</td>
<td>71.7%</td>
<td>80.8%</td>
<td>76.0%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>11.4%</td>
<td>100.0%</td>
<td>99.8%</td>
<td>99.9%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>10.8%</td>
<td>61.5%</td>
<td>60.0%</td>
<td>60.7%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>3.8%</td>
<td>99.3%</td>
<td>98.8%</td>
<td>99.0%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>4.0%</td>
<td>66.8%</td>
<td>77.4%</td>
<td>71.7%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>11.9%</td>
<td>66.7%</td>
<td>72.7%</td>
<td>69.5%</td>
<td>104 s</td>
</tr>
</tbody>
</table>
Performance of $1$-NN on log entropy data:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.3%</td>
<td>99.9%</td>
<td>99.4%</td>
<td>99.7%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>21.5%</td>
<td>68.5%</td>
<td>79.1%</td>
<td>73.4%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>11.4%</td>
<td>100.0%</td>
<td>99.8%</td>
<td>99.9%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>10.3%</td>
<td>63.6%</td>
<td>59.1%</td>
<td>61.2%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>3.8%</td>
<td>99.3%</td>
<td>98.8%</td>
<td>99.0%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>4.0%</td>
<td>66.9%</td>
<td>76.1%</td>
<td>71.2%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>11.9%</td>
<td>66.3%</td>
<td>71.4%</td>
<td>68.6%</td>
<td>104 s</td>
</tr>
</tbody>
</table>
Preliminary conclusions:

- Naïve Bayes is better than Rocchio on TF data, but comparable to Rocchio on the other two representations.
- 1-NN on TF is better than Naïve Bayes and Rocchio, regardless of the representation chosen.

- TF–IDF and log entropy are of comparable performance and perform sometimes better than TF and sometimes worse.
  - Rocchio: better than TF
  - 1-NN: worse than TF
Performance of 5-NN on TF data:

- Performance on TF-IDF and log entropy is slightly worse

<table>
<thead>
<tr>
<th></th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.3%</td>
<td>94.2%</td>
<td>93.5%</td>
<td>93.8%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>19.1%</td>
<td>90.2%</td>
<td>92.3%</td>
<td>91.2%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>8.8%</td>
<td>91.7%</td>
<td>70.8%</td>
<td>79.9%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>8.1%</td>
<td>84.3%</td>
<td>61.8%</td>
<td>71.3%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>3.5%</td>
<td>88.2%</td>
<td>81.6%</td>
<td>84.7%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>3.6%</td>
<td>73.0%</td>
<td>75.2%</td>
<td>74.1%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>10.3%</td>
<td>82.5%</td>
<td>76.4%</td>
<td>78.9%</td>
<td>104 s</td>
</tr>
</tbody>
</table>
Performance of 31-NN on TF data:

- Performance on TF-IDF and log entropy is slightly worse

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.4%</td>
<td>91.9%</td>
<td>91.9%</td>
<td>91.9%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>19.2%</td>
<td>89.4%</td>
<td>92.0%</td>
<td>90.7%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>4.9%</td>
<td>95.5%</td>
<td>40.9%</td>
<td>57.3%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>4.6%</td>
<td>94.3%</td>
<td>38.9%</td>
<td>55.0%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>3.3%</td>
<td>81.5%</td>
<td>70.4%</td>
<td>75.6%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>3.1%</td>
<td>72.3%</td>
<td>64.6%</td>
<td>68.2%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>9.0%</td>
<td>85.3%</td>
<td>65.2%</td>
<td>71.3%</td>
<td>104 s</td>
</tr>
</tbody>
</table>
Performance of **Linear SVM** on **TF** data:

- Performance on TF–IDF and log entropy is slightly worse
- $C = 0.1$ seems to give the best results

<table>
<thead>
<tr>
<th></th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>18.3%</td>
<td>98.7%</td>
<td>98.3%</td>
<td>98.5%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>18.6%</td>
<td>92.8%</td>
<td>92.5%</td>
<td>92.7%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>11.1%</td>
<td>99.5%</td>
<td>96.4%</td>
<td>97.9%</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>10.3%</td>
<td>84.2%</td>
<td>78.4%</td>
<td>81.2%</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>3.5%</td>
<td>97.8%</td>
<td>91.1%</td>
<td>94.3%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>3.0%</td>
<td>75.8%</td>
<td>65.0%</td>
<td>70.0%</td>
<td></td>
</tr>
<tr>
<td><strong>Average (test)</strong></td>
<td><strong>10.6%</strong></td>
<td><strong>84.3%</strong></td>
<td><strong>78.6%</strong></td>
<td><strong>81.3%</strong></td>
<td><strong>13 s</strong></td>
</tr>
</tbody>
</table>
Performance of **RBF SVM** on log entropy data:

- Performance on TF and TF-IDF is even worse
- We did not find a working value for $C$, maybe there is none...

<table>
<thead>
<tr>
<th></th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>earn (train)</td>
<td>10.7%</td>
<td>92.3%</td>
<td>53.5%</td>
<td>67.8%</td>
<td></td>
</tr>
<tr>
<td>earn (test)</td>
<td>10.3%</td>
<td>91.0%</td>
<td>50.5%</td>
<td>65.0%</td>
<td></td>
</tr>
<tr>
<td>acq (train)</td>
<td>0.0%</td>
<td>—</td>
<td>0.0%</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>acq (test)</td>
<td>0.0%</td>
<td>—</td>
<td>0.0%</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>money-fx (train)</td>
<td>0.1%</td>
<td>100.0%</td>
<td>3.5%</td>
<td>6.7%</td>
<td></td>
</tr>
<tr>
<td>money-fx (test)</td>
<td>0.1%</td>
<td>100.0%</td>
<td>2.2%</td>
<td>4.3%</td>
<td></td>
</tr>
<tr>
<td>Average (test)</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>500 s</td>
</tr>
</tbody>
</table>
## Summary (averages on test data):

<table>
<thead>
<tr>
<th>Method</th>
<th>Result size</th>
<th>Precision</th>
<th>Recall</th>
<th>F measure</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rocchio on TF–IDF</td>
<td>17.3%</td>
<td>54.7%</td>
<td>91.8%</td>
<td>65.3%</td>
<td>0.2 s</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>16.2%</td>
<td>55.3%</td>
<td>87.0%</td>
<td>62.5%</td>
<td>0.4 s</td>
</tr>
<tr>
<td>1-NN on TF</td>
<td>11.5%</td>
<td>75.2%</td>
<td>79.9%</td>
<td>77.4%</td>
<td>104 s</td>
</tr>
<tr>
<td>5-NN on TF</td>
<td>10.3%</td>
<td>82.5%</td>
<td>76.4%</td>
<td>78.9%</td>
<td>104 s</td>
</tr>
<tr>
<td>31-NN on TF</td>
<td>9.0%</td>
<td>85.3%</td>
<td>65.2%</td>
<td>71.3%</td>
<td>104 s</td>
</tr>
<tr>
<td>Linear SVM on TF</td>
<td>10.6%</td>
<td>84.3%</td>
<td>78.6%</td>
<td>81.3%</td>
<td>13 s</td>
</tr>
<tr>
<td>RBF SVM on log entropy</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>500 s</td>
</tr>
</tbody>
</table>

- There is no clear winner
- Differences among topics
- Differences regarding quality measures (precision, recall, …)
An Overview of Web Retrieval

A typical Web search engine:

- Users
- Web crawler
- Indexer
- Business model
- User interface
- Retrieval algorithms

Today’s topic
Lecture 12: Link Analysis

1. Link Structures
2. PageRank
3. HITS
Networks of **social interactions** are formed...

- Between academics by **co-authoring**

  *Optimal Preference Elicitation for Skyline Queries over Categorical Domains*

  Jongwuk Lee¹, Gae-won You¹, Seung-won Hwang¹, Joachim Selke², and Wolf-Tilo Balke²

- Between movie personnel by **directing and acting**
Social Networks (2)

- Between musicians, soccer stars, friends, and relatives

- Between countries via trading relations
Social Networks (3)

- Between people making phone calls
- Between people transmitting infections
– Between scientific papers through citations

Skyline computation: Skyline queries have been first studied as maximal vectors in [19]. Later, Börzsönyi et al. [5] introduced skylines queries in database applications. Next, Tan et al. [23] proposed progressive skyline computation using auxiliary structures. Kossmann et al. [18] improves (D&C) algorithm, and proposed nearest neighbor (NN) algorithm. Similarly, Papadias et al. [22] developed branch and bound skyline (BBS) algorithm which achieves I/O optimal property. Meanwhile Chomicki et al. [10] developed sort-filter-skyline (SFS) algorithm leveraging pre-sorting lists, and Godfrey et al. [12] proposed linear elimination-sort for skyline (LESS) algorithm with attractive average-case asymptotic complexity. Recently, there have been active research efforts to address “curse of dimensionality” problem of skyline queries [6,7,21] using inherent properties of skylines such as skyline frequency, k-dominant skylines, and k-representative skylines. All these efforts, however, focused only on numerical domains with inherent orders, and did not consider skyline queries over categorical domains.

– And, of course, between Web pages through links...
• It has been quite common for decades to model social networks using \textit{directed graphs}:

\begin{itemize}
  \item A directed graph
  \item Adjacency matrix
\end{itemize}

\[
A_{i,j} = 1 \quad \text{if and only if} \quad \text{node } i \text{ links to node } j
\]
Classical research questions:

- Which authors have a high **prestige** (or status)?
- Which countries are **well-connected**, which are **isolated**?
- Which people **connect** different communities?
• Using the graph model, it has been clear that in-degree is a good first-order indicator of prestige.

• In 1949, the sociologist John R. Seeley realized the recursive nature of prestige in a social network:
  – A person’s status is a function of the status of those who choose him.
  – And their status is a function of those who choose them.
  – And so *ad infinitum*…
A Model of Prestige

• Seeley **modeled prestige** as follows:
  – Every node $u$ has a notion of **prestige** $p(u)$ associated with it, which is simply a **positive real number**
  – Over all nodes, we represent the prestige score as a real column vector $p$ having exactly one entry for each node
  – **Recursiveness constraint:**
    The prestige of each node $u$ should be proportional to the total sum of prestige of all nodes that link to $u$, i.e.
    $$ p(u) = \alpha \cdot \sum_{v \rightarrow u} p(v) $$
  – **Equivalent fixpoint condition:**
    $$ p = \alpha \cdot A^T \cdot p $$
    • **Task:** Find numbers $p$ and $\alpha$ such that the condition holds
    • This approach fits well to ideas from linear algebra (later)
A Model of Prestige (2)

\[ p(u) = \alpha \cdot \sum_{v \rightarrow u} p(v) \]

\[ p = \alpha \cdot A^\top \cdot p \]

Example:

\[ p = (0.65, 0.65, 0, 0.4) \]
\[ \alpha = 0.62 \]

\[
\begin{array}{cccc}
A & 1 & 2 & 3 & 4 \\
1 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 0 & 0 \\
3 & 0 & 0 & 0 & 1 \\
4 & 1 & 1 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccc}
A^\top & 1 & 2 & 3 & 4 \\
1 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 0 & 1 \\
3 & 0 & 0 & 0 & 0 \\
4 & 1 & 0 & 1 & 0 \\
\end{array}
\]

Information Retrieval and Web Search Engines — Wolf-Tilo Balke and Joachim Selke — Technische Universität Braunschweig
Another interesting notion is **centrality**

**Definitions:**

- The **distance** $d(u, v)$ between two nodes $u$ and $v$ in a directed graph is the **smallest number of links** via which one can go from $u$ to $v$
- The **radius** of a node $u$ is $r(u) = \max_v d(u, v)$, i.e., the distance to $u$’s **most distant node**
- The **center** of the graph is $\arg \min_u r(u)$, i.e., the node that has the **smallest radius**
Centrality (2)

- The scientific citation graph:
  - Link a **paper** $u$ to a paper $v$, i.e. set $u \rightarrow v$, if $u$ is cited by $v$
  - Papers having a **small radius** are likely to be very **influential**

- The scientific collaboration graph:
  - Link two **authors** $u$ and $v$, i.e. set $u \leftrightarrow v$, if they **co-authored** a paper
  - The **Erdős number** of an author $u$ is his/her **distance to** the famous mathematician **Paul Erdős**
There are many other notions of centrality, e.g., **cuts**:

- A **cut** is a (usually small) number of **edges** that, when removed, disconnect a given pair of **vertices**.
- One may look for a small set of **vertices** that, when removed, will **decompose** the graph into two or more connected components.
- This is useful for the study of **epidemics, espionage, or suspected terrorist communication** on telephone networks.
Another important measure is **co-citation**

- If document \( u \) cites documents \( v \) and \( w \), then \( v \) and \( w \) are said to be co-cited by \( u \)

If documents \( v \) and \( w \) are **co-cited** by many documents, then \( v \) and \( w \) are somehow related to each other

In terms of the adjacency matrix \( A \):

- Link a **document** \( u \) to a paper \( v \), i.e. set \( u \rightarrow v \), if \( u \) cites \( v \)
- The number of documents co-citing \( v \) and \( w \) is the entry corresponding to \( v \) and \( w \) in the matrix \( A^T A \):

\[
A^T A[v, w] = \sum_u A^T[v, u] A[u, w]
\]

\[
= \sum_u A[u, v] A[u, w] = \left\{ u \mid u \rightarrow v \text{ and } u \rightarrow w \right\}
\]
• The entry in the $A^T A$ matrix corresponding to $[v, w]$ is the **co-citation index** of $v$ and $w$ and a **measure of relatedness** between $v$ and $w$

• One may use this pairwise relatedness measure in a clustering algorithm, such as **multidimensional scaling**

• MDS is similar to the **singular value decomposition**

• It uses a similarity matrix to **embed** the documents into a **low-dimensional Euclidean space** (e.g. a plane)

• **Visualizing clusters** based on co-citation reveals important **social structures** between and within link communities
(Boyack et al., 2005) visualized similarity data based on co-citations created from over 1 million journal articles published in 2000:

Each point represents a journal
• Classical IR:
  – The **worth of a document** with regard to a query is **intrinsic** to the document
  – **Documents are self-contained units**, and are generally descriptive and **truthful** about their contents

• Modern Web search:
  – Apply ideas from network analysis to the **Web graph**...
  – **Links are recommendations**
  – **Anchor texts** can be used as document descriptions
Assumption 1:
A hyperlink is signal of quality or popular interest

- In some sense, a link is a democratic vote
Assumption 2:
The anchor text of a link (or its surrounding text) describes the target page

Excerpt from Yahoo! Directory

IBM’s home page
(does not contain the term computer!)
Both assumptions clearly do not hold always
But exploiting them has proved to be much better than not exploiting them

We will concentrate on the first assumption:
“Links are quality signals”

Two highly popular algorithms:
– PageRank (Page et al., 1998)
– HITS (Kleinberg, 1999)
PageRank and HITS

• **PageRank**
  - Developed around the fall of 1996 at Stanford University by Larry Page and Sergey Brin, the founders of **Google**
  - **Idea:** Assign a *query-independent* measure of *prestige* to each Web resource

• **HITS**
  - Developed at the same time at IBM Almaden Research Center by Jon Kleinberg, a famous computer scientist
  - **Idea:** For any given query, assign *two measures* to each Web resource, a *hub score* and an *authority score*
    - **Hub:** A compilation of links to relevant Web resources
    - **Authority:** A resource that is relevant in itself
Before 1993:

- There are no search engines...
- Tim Berners-Lee maintains a list of Web servers:

**W3 servers**

*Note*: this page is here for historical interest only; the content hasn't been updated since late 1992.

For more up-to-date lists of web servers, see:

- [Netcraft](http://www.netcraft.com): Directory of Web Server Home Sites
- [WDVL](http://www.wdvl.com): Servers

**Webmaster**

This is a list of some WWW servers. It does not include all servers, and note that one server machine can serve many databases. See also: background on [WWW](http://www.w3.org), and data available by other protocols, data by subject, how to make a new server, test servers. If servers are marker "experimental", you should not expect anything. The top of the list is in reverse chronological order of addition.

- [NCSA](http://www.ncsa.uiuc.edu): National Center for Supercomputing Applictions, Urbana Champain, IL, USA. Experimental.
- [IN2P3](http://www.in2p3.fr): Lyon, France.
- [KVI](http://www.kvi.nl): Kernfysisch Versneller Instituut (nuclear physics accelerator institute), Groningen, Netherlands. VMS server.

- In Germany: LEO, “Link Everything Online”
1993–1998:
- Many new search engines, most popular: Lycos, AltaVista, Excite, Inktomi, HotBot, Ask Jeeves
- All of them mainly rely on **classical IR techniques** and focus on the **problem of scaling**

1998:
- **Google** is founded
- The first engine that heavily exploits the Web’s **link structure**
- Google’s success has a name: **PageRank**

1998–Today:
- Large companies try to **keep up with Google**
- Most noteworthy: Yahoo and Microsoft
The next big thing in Web search?

- Clustering?
- Natural language query processing?
- The “Semantic Web”?
- Advanced personalization?
- Open source search engines?
- Metasearch that works?
- Federated search?
- Innovative user interfaces?
- Something else?
Lecture 12: Link Analysis

1. Link Structures
2. PageRank
3. HITS
• **Problem:**
  
  – How to assign a *query-independent* measure of *prestige* to each Web resource?

• **A good but infeasible solution:**
  
  – Rank Web resources by their *popularity* (measured by traffic?)

• **The PageRank solution:**
  
  – Apply *John R. Seeley’s model* of prestige to the Web graph!
  – The number of in-links is correlated to a resource’s prestige
  – Links from good resources should count more than links from bad ones

\[
p(u) = \alpha \cdot \sum_{v \to u} p(v)
\]
Imagine a Web surfer doing a **random walk** on the Web:

- 90% of the time, the surfer clicks a **random hyperlink**
- 10% of the time, the surfer types in a **random URI**
- **PageRank** = The long-term visit rate of each node

This is a **crude, but useful, Web surfing model**

- No one chooses links with equal probability, surfing usually is topic-driven
- How to surf to a random page?
- What about the back button or bookmarks?
A more detailed version of the model:

1. Start at a random page, chosen uniformly
2. Flip a coin that shows “tails” with probability $\lambda$
3. If the coin shows “heads”
   AND the current page has a positive out-degree:
   - Randomly follow one of the pages out-links
   - Continue at (2)
   If the coin shows “tails”
   OR the current page has no out-links:
   - Surf to a random Web page, chosen uniformly
   - Continue at (2)
Example:

Adjacency matrix:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Set $\lambda = 0.25$

Transition matrix:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.75 + 0.05</td>
</tr>
<tr>
<td>2</td>
<td>0.75 + 0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>3</td>
<td>0.05</td>
<td>0.75 + 0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.375 + 0.05</td>
<td>0.05</td>
<td>0.375 + 0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>0.25 + 0.05</td>
<td>0.25 + 0.05</td>
<td>0.25 + 0.05</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Example (continued):

Transition matrix:

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>0.8</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>3</td>
<td>0.05</td>
<td>0.8</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>4</td>
<td>0.425</td>
<td>0.05</td>
<td>0.425</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>5</td>
<td>0.05</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.05</td>
</tr>
</tbody>
</table>

- If the surfer is at page 3 in step $t$
  - He/she will be at page 1 in step $t + 1$ with a probability of 5%
  - He/she will be at page 2 in step $t + 1$ with a probability of 80%
  - He/she will be at page 3 in step $t + 1$ with a probability of 5%
  - He/she will be at page 4 in step $t + 1$ with a probability of 5%
  - He/she will be at page 5 in step $t + 1$ with a probability of 5%
Example (continued):

- Let’s do a simulation
- If we start in state 1, what’s the probability of being in state $i$ after $t$ steps?

<table>
<thead>
<tr>
<th>$t$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t = 1$</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>$t = 2$</td>
<td>0.11</td>
<td>0.29</td>
<td>0.27</td>
<td>0.25</td>
<td>0.09</td>
</tr>
<tr>
<td>$t = 3$</td>
<td>0.36</td>
<td>0.27</td>
<td>0.17</td>
<td>0.07</td>
<td>0.13</td>
</tr>
<tr>
<td>$t = 4$</td>
<td>0.28</td>
<td>0.21</td>
<td>0.11</td>
<td>0.08</td>
<td>0.32</td>
</tr>
<tr>
<td>$t = 5$</td>
<td>0.24</td>
<td>0.21</td>
<td>0.16</td>
<td>0.13</td>
<td>0.26</td>
</tr>
<tr>
<td>$t = 6$</td>
<td>0.26</td>
<td>0.24</td>
<td>0.16</td>
<td>0.12</td>
<td>0.23</td>
</tr>
<tr>
<td>$t = 7$</td>
<td>0.27</td>
<td>0.23</td>
<td>0.15</td>
<td>0.11</td>
<td>0.24</td>
</tr>
<tr>
<td>$t = 8$</td>
<td>0.26</td>
<td>0.22</td>
<td>0.15</td>
<td>0.11</td>
<td>0.25</td>
</tr>
<tr>
<td>$t = 9$</td>
<td>0.26</td>
<td>0.23</td>
<td>0.15</td>
<td>0.11</td>
<td>0.25</td>
</tr>
</tbody>
</table>

The probability vector seems to converge...
• And indeed, **the probability vector converges** as \( t \) goes to infinity, for any initial probability vector.

• To make this point clear, we need some **linear algebra** and some **theory of stochastic processes**.

• **Definitions:**
  
  – Let \( n \) denote the number of nodes.
  
  – A **probability vector** is an \( n \)-dimensional vector such that
    (a) all entries are **non-negative** and
    (b) the **sum of entries is 1**
  
  – A **stochastic matrix** is an \( n \times n \) matrix such that
    (a) all entries are **non-negative** and
    (b) the **sum of each row is 1**
• Stochastic matrices are closely related to **Markov chains:**
  – A Markov chain consists of *n states* and an *n × n stochastic matrix* $T$
  – Each row and column of $T$ corresponds to a state, respectively
  – At any point in time, the Markov chain is in exactly one of these states
  – **Time is discrete,** i.e. it runs in discrete steps: $t = 0, 1, 2, \ldots$
  – From time step to time step, the chain’s current state changes according to the stochastic matrix $T$:
    $$\Pr(\text{state } v \text{ at time } t + 1 \mid \text{state } u \text{ at time } t) = T[u, v]$$
Convergence (3)

• In essence, a Markov chain is a probabilistic finite state machine.

• Knowledge about the current state of a Markov chain can be expressed by **probability vectors** of length $n$.

• Remember our example:
  
  – Knowing for sure that the current state of the chain is state $u$, can be expressed by a probability vector that is 1 at $u$’s place.
  
  – For example, $(0.2, 0.5, 0.3)$ means that the chain’s probability to be in the first, second, and third state is 20%, 50%, and 30%, respectively.
State transitions can be formalized using matrix–vector multiplication.

Let $T$ be a transition matrix and $p$ a probability vector that models the chain’s state probabilities at time $t$.

What are the state probabilities $p'$ at time $t + 1$?

$$p' = T^T \cdot p$$

Example ($n = 2$):

$$p = (p_1, p_2)$$

$$p' = (p_1', p_2')$$

$$p_1' = T_{1,1} \cdot p_1 + T_{2,1} \cdot p_2$$

$$p_2' = T_{1,2} \cdot p_1 + T_{2,2} \cdot p_2$$
Convergence (5)

• Now we have everything we need to talk about convergence properties of the Markov chain

• Let \( p_0 \) be some initial probability state vector

• Let \( p_t \) denote the probability state vector at time \( t \)

• Then, for any \( t \), we have

\[
 p_{t+1} = T^T \cdot p_t
\]

• Clearly, convergence of \( p_t \) as \( t \to \infty \) means that \( p_t \) converges to a vector \( p \) such that

\[
 p = T^T \cdot p
\]

• Well, what we are looking for is an eigenvector of \( T^T \) corresponding to the eigenvalue \( 1 \)
• According to the **Perron–Frobenius theorem** from linear algebra the following is true:
  
  – Every stochastic matrix containing **only positive entries** has **1 as one of its eigenvalues**
  
  – Furthermore, **1 is the largest eigenvalue** of the matrix
  
  – There is **only one eigenvector** having the eigenvalue **1**

• Since we do a **random teleport** with probability \( \lambda > 0 \) in the random surfer model, the theorem applies

• Therefore, we can be sure that there is a probability vector \( p \) satisfying \( p = T^T \cdot p \)

• Such a vector \( p \) is called the Markov chain’s **stationary probability vector**
• In the **random surfer model** there is a **unique stationary probability vector** $p$.

• Node $u$’s **PageRank** is its stationary probability $p[u]$.

<table>
<thead>
<tr>
<th>$t$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t = 0$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$t = 1$</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.8</td>
</tr>
<tr>
<td>$t = 2$</td>
<td>0.11</td>
<td>0.29</td>
<td>0.27</td>
<td>0.25</td>
<td>0.09</td>
</tr>
<tr>
<td>$t = 3$</td>
<td>0.36</td>
<td>0.27</td>
<td>0.17</td>
<td>0.07</td>
<td>0.13</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$t \to \infty$</td>
<td>0.26</td>
<td>0.23</td>
<td>0.15</td>
<td>0.11</td>
<td>0.25</td>
</tr>
</tbody>
</table>

• This fits **Seeley’s notion of prestige**:

$$p(u) = \alpha \cdot \sum_{v \rightarrow u} p(v)$$
PageRank (2)

- PageRank was invented by Larry Page at Stanford
- PageRank is **patented** as US patent 6,285,999
  - “Method for node ranking in a linked database”
    - The method for computing the PageRank and related stuff are patented!
  - Patent was assigned to Stanford University (not to Google)
  - **Google has exclusive license rights**
  - Stanford received **1.8 million shares in Google** in exchange for use of the patent
  - These shares were sold in 2005 for **336 million dollars**
Result list sorted by PageRank

Result list sorted using IR methods

Query: “university”
• How to compute the PageRank?
• A very simple method for eigenvalue and eigenvector computation is the so-called **power iteration**, which can be applied to any quadratic matrix \( A \):

1. Start with an arbitrary initial vector \( b_0 \)
2. Set \( i = 0 \)
3. Set \( b_{i+1} = A \cdot b_i \)
4. Set \( b_{i+1} = b_{i+1} / |b_{i+1}| \), i.e. normalize \( b_{i+1} \) to unit length
5. Set \( i = i + 1 \)
6. GOTO 2
• One can prove that the power iteration converges to the eigenvector of $A$ having the largest eigenvalue.
• In our case, the largest eigenvalue is 1.
  – The power iteration finds the stationary probability vector $p$.
• How many iterations are needed?
  – Actually, the number is quite low since we don’t need a perfect result anyway…
• How to compute the PageRank for a Web graph containing 60 billion nodes?
  – Use a highly scalable distributed algorithm
  – Actually, this is one of Google’s secrets…
A search engine myth:
“PageRank is the most important component of ranking”

The reality:
- There are several components that are at least as important:
  Anchor text, phrases, proximity, …
- Google uses **hundreds of different features** for ranking
- There are rumors that PageRank in its original form
  (as presented here) has a negligible effect on ranking
- However, variants of PageRank are
  still an essential part of ranking
- Addressing **link spam** is difficult and crucial!
• A disadvantage of PageRank is that it computes only a single overall score for each web resource
  – A web resource might be unimportant from a global view but highly important for a specific topic

• **Topic-sensitive PageRank** tries to address this issue:
  – Define a set of popular **topics** (e.g. football, Windows, Obama)
  – Use **classification** algorithms to assign each Web resource to one (or more) of these topics
  – For each topic, compute a **topic-sensitive PageRank** by limiting the random teleports to pages of the current topic
  – At query time, detect the query’s topics and use the corresponding PageRank scores...
### Example (query: bicycling):

<table>
<thead>
<tr>
<th>NoBias</th>
<th>Arts</th>
</tr>
</thead>
</table>
| “RailRiders Adventure Clothing”
www.RailRiders.com
www.Waypoint.org/default.html
www.Gorp.com/
www.FloridaCycling.com/ | “Photo Contest & Gallery (Bicycling)”
www.bikescape.com/photogallery/ |

<table>
<thead>
<tr>
<th>Business</th>
<th>Computers</th>
</tr>
</thead>
</table>
| “Recumbent Bikes and Kit Aircraft”
www.rans.com
www.BreakawayBooks.com
java.oreilly.com/bite-size/
www.carboom.com | “GPS Pilot”
www.gpspilot.com
www.wireless.gr/wireless-links.htm
www.linkstosales.com
www.LiftExperts.com/lifts.html |

<table>
<thead>
<tr>
<th>Games</th>
<th>Kids and Teens</th>
</tr>
</thead>
</table>
| “Definition Through Hobbies”
www.flickr.com/”gretchen/hobbies.html
www.BellaOnline.com/sports/
www.npr.org/programs/wesun/puzzle/will.html
www.trygve.com/ | “Camp Shohola For Boys”
www.shohola.com
www.EarthForce.org
www.WeissmanTours.com

<table>
<thead>
<tr>
<th>Recreation</th>
<th>Science</th>
</tr>
</thead>
</table>
| “Adventure travel”
www.gorp.com/
www.gorp.com/gorp/activity/main.htm
www.outdoor-pursuits.org/ | “Coast to Coast by Recumbent Bicycle”
hypertextbook.com/bent/
www.SiestaSoftware.com/
www.BenWiens.com/benwiens.html
www.SusanJeffers.com/jeffbio.htm |

<table>
<thead>
<tr>
<th>Shopping</th>
<th>Sports</th>
</tr>
</thead>
</table>
| “Cycling Clothing & Accessories for Women”
www.TeamEstrogen.com/
www.ShopOutdoors.com/
www.bike.com/ | “Swim, Bike, Run, & Multisport”
www.multisports.com/
www.BikeRacing.com/
www.CycleCanada.com/
www.bikescape.com/photogallery/ |
Comparison to PageRank (precision at 10):

![Comparison to PageRank chart](chart.png)
• **Eliminate navigational links:**
  - Most web pages contain navigational structures
  - The **quality assumption** does only hold if a hyperlink was created as a result of *editorial judgment*
  - Therefore, navigational links should be removed before computing the PageRank

• **Eliminate nepotistic links:**
  - Nepotism = favoritism based on kinship
  - Links between *pages authored by the same person* also are problematic
  - Again, they should be removed before doing any computations
  - Unfortunately, it’s much harder to detect them than detecting navigational links…
• Web pages having the highest PageRank:
  http://www.seocompany.ca/pagerank/pr-10-pages.php

• Google Toolbar:
  http://toolbar.google.com
More Applications

• The PageRank can be used for **crawling**:
  – Decide how deep to crawl a web site
  – Decide how often to update a resource

• Other applications:
  – Impact factors of journals:
    [http://www.eigenfactor.org](http://www.eigenfactor.org)
    [http://www.scimagojr.com](http://www.scimagojr.com)
  – Generate reading lists based on Wikipedia:
    [http://www.maxtility.com/scgi-bin/wikiosity.cgi](http://www.maxtility.com/scgi-bin/wikiosity.cgi)
  – Rank doctoral programs:
  – Many more…
Lecture 12: Link Analysis

1. Link Structures
2. PageRank
3. HITS
• HITS stands for hyperlink induced topic search
• Invented by Jon Kleinberg

• Problem setting:
  – For any information need, there are hubs and authorities
    • Authority: Definitive high-quality information (query-dependent!)
    • Hub: Comprehensive lists of links to authorities (query-dependent!)
  – To a certain degree, each page is a hub as well as an authority

• Task:
  – Given a query, estimate the degree of authority and hubness of each Web page
• **Obvious:**
  The authority and hubness scores are query-dependent, therefore the computation has to be done at query time

• **Idea:**
  – **Given:** A query $q$
  – Send $q$ to a **standard IR system** to collect
    a **root set** $R$ of nodes in the Web graph
  – Collect the **base set** $V_q$ of nodes, which includes $R$ as well as all nodes that are connected to $R$ by an in-link or out-link

Root set
• Idea (continued):
  – Finally, compute hub and authority scores on the base set

• Hubs and authority scores are defined similar to prestige:
  – Let $A$ be the base set’s adjacency matrix
  – Denote the nodes’ hub scores by a vector $h$ and their authority scores by a vector $a$
  – A recursive definition of $h$ and $a$:
    \[
    a = \alpha \cdot A^T \cdot h \quad \quad \quad \quad \quad \quad \quad h = \beta \cdot A \cdot a
    \]
    – Again, $\alpha$ and $\beta$ are proportionality constants
  – The authority score of a page is proportional to the sum of hub scores of the pages linking to it
  – The hub score of a page is proportional to the sum of authority scores of the pages to which it links
By combining both equations we arrive at:

\[ a = \alpha \cdot A^T \cdot h \quad \text{and} \quad h = \beta \cdot A \cdot a \]

• As we see:
  – The authority vector \( a \) is an eigenvector of \( A^T A \)
  – The hub vector \( h \) is an eigenvector of \( A A^T \)

• Kleinberg decided to take the principal eigenvectors in each case, i.e. the eigenvectors corresponding to the eigenvalues with the highest absolute values

• Again, they can be computed using the power iteration
Example (query: japan elementary schools):

<table>
<thead>
<tr>
<th>Hubs</th>
<th>Authorities</th>
</tr>
</thead>
<tbody>
<tr>
<td>schools</td>
<td>The American School in Japan</td>
</tr>
<tr>
<td>LINK Page-13</td>
<td>The Link Page</td>
</tr>
<tr>
<td>∵ú¬{ǐšw¬Z</td>
<td>%œ¬Èë¬s—§¬ä¬c¬¬šw¬Zfz=[f¬fy=][fw</td>
</tr>
<tr>
<td>¼ë¬ë¬š¬šw¬Zfz=[f¬fy=][fw</td>
<td>Kids’ Space</td>
</tr>
<tr>
<td>100 Schools Home Pages (English)</td>
<td>§¬Èé¬s¬§¬Èé¬¼¬¬šw¬Z</td>
</tr>
<tr>
<td>K-12 from Japan 10/met and Education</td>
<td>{èé¬ç¬šw¬¬¬šw¬Z</td>
</tr>
<tr>
<td><a href="http://www...iglobe.ne.jp/~KESAN">http://www...iglobe.ne.jp/~KESAN</a></td>
<td>KEIMEI GAKUEN Home Page (Japanese)</td>
</tr>
<tr>
<td>,l,fj¬¬šw¬Z,µ‘N,‘P¬œē</td>
<td>Shiranuma Home Page</td>
</tr>
<tr>
<td>∵œ¬š¬——š¬œ¬š¬—œ¬¬šw¬Z</td>
<td>fuzoku-es.fukui-u.ac.jp</td>
</tr>
<tr>
<td>Koulutus ja oppilaitokset</td>
<td>welcome to Miasa E&amp;J school</td>
</tr>
<tr>
<td>TOYODA HOMEPAGE</td>
<td>%œ¬ë¬ë¬š¬šw¬Zfz=[f¬fy=][fw</td>
</tr>
<tr>
<td>Education</td>
<td><a href="http://www...p/~m_maru/index.html">http://www...p/~m_maru/index.html</a></td>
</tr>
<tr>
<td>Cay’s Homepage(Japanese)</td>
<td>fukui haruyama-es HomePage</td>
</tr>
<tr>
<td>¬y¬¬šw¬Z,šw¬Zfz=[f¬fy=][fw</td>
<td>Torisu primary school</td>
</tr>
<tr>
<td>UNIVERSITY</td>
<td>goo</td>
</tr>
<tr>
<td>%œ¬¬š¬¬šw¬Z DRAGON97-TOP</td>
<td>Yakumo Elementary,Hokkaido,Japan</td>
</tr>
<tr>
<td></td>
<td>FUZOKU Home Page</td>
</tr>
<tr>
<td></td>
<td>Kamishibun Elementary School...</td>
</tr>
</tbody>
</table>
As PageRank, **HITS has been patented:**

- US patent 6,112,202
- “Method and system for identifying authoritative information resources in an environment with content-based links between information resources”
- Inventor: Jon Kleinberg
- **Assignee: IBM**
Connection to LSI/SVD

• There is a direct mapping between finding the singular value decomposition of $A$ and finding an eigen-decomposition of $A^T A$ and $A A^T$

• A short recap from Lecture 4:
  – Let $A = U S V$ be the SVD of $A$
  – **Theorem:**
    - $U$’s columns are the eigenvectors of $A A^T$, the matrix $S^2$ contains the corresponding eigenvalues
    - Similarly, $V$’s rows are the eigenvectors of $A^T A$, $S^2$ again contains the eigenvalues

• Therefore, HITS is equivalent to running the SVD on the adjacency matrix of the base set
• If the query is ambiguous (e.g. “Java” or “jaguar”) or polarized (e.g. “abortion” or “cold fusion”), the base set will contain a few, almost disconnected, link communities.

• Then, the principal eigenvectors found by HITS will reveal hubs and authorities in the largest link community.

• One can tease of this structure by computing not only the principal eigenvectors but some more.
PageRank can be precomputed, HITS has to be computed at query time
  - HITS is very expensive

Different choices regarding the formal model
  - HITS models hubs and authorities
  - HITS uses a subset of the Web graph
  - But: We could also apply PageRank to a subset and HITS on the whole Web graph…

On the Web, a good hub usually is also a good authority

The difference between HITS and PageRank is not that large…
• Spam detection
• Metasearch
• Privacy issues