Applications
of
Nonnegative Matrices:
Ranking
and
Clustering

Amy Langville
Mathematics Department
College of Charleston
Collaborators

- Carl Meyer, N. C. State University
- David Gleich, Stanford University
- Michael Berry, University of Tennessee-Knoxville
- College of Charleston (current and former students)
  - Luke Ingram
  - Ibai Basabe
  - Kathryn Pedings
  - Neil Goodson
  - Colin Stephenson
  - Emmie Douglas
Nonnegative Matrices

- Ranking webpages
- Ranking sports teams
- Recommendation systems
- Meta-algorithms
  - Rank and rating aggregation
  - Cluster aggregation
Ranking webpages
1998: enter Link Analysis

- uses hyperlink structure to focus the relevant set
- combine traditional IR score with popularity score
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It’s huge.
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 400 X bigger than Surface Web
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

• It’s huge.
  – over 10 billion pages, average page size of 500KB
  – 20 times size of Library of Congress print collection
  – Deep Web - 400 X bigger than Surface Web

• It’s dynamic.
  – content changes: 40% of pages change in a week, 23% of .com change daily
  – size changes: billions of pages added each year
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

• **It’s huge.**
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 400 X bigger than Surface Web

• **It’s dynamic.**
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year

• **It’s self-organized.**
  - no standards, review process, formats
  - errors, falsehoods, link rot, and spammers!
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It’s huge.
  - over 10 billion pages, average page size of 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 400 X bigger than Surface Web

- It’s dynamic.
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year

- It’s self-organized.
  - no standards, review process, formats
  - errors, falsehoods, link rot, and spammers!

A Herculean Task!
Web Information Retrieval

IR before the Web = traditional IR
IR on the Web = web IR

How is the Web different from other document collections?

- It’s huge.
  - over 10 billion pages, each about 500KB
  - 20 times size of Library of Congress print collection
  - Deep Web - 400 X bigger than Surface Web

- It’s dynamic.
  - content changes: 40% of pages change in a week, 23% of .com change daily
  - size changes: billions of pages added each year

- It’s self-organized.
  - no standards, review process, formats
  - errors, falsehoods, link rot, and spammers!

- Ah, but it’s hyperlinked!
  - Vannevar Bush’s 1945 memex
Elements of a Web Search Engine
The Ranking Module (generates popularity scores)

- Measure the importance of each page
The Ranking Module (generates popularity scores)

- Measure the importance of each page

- The measure should be Independent of any query
  - Primarily determined by the link structure of the Web
  - Tempered by some content considerations
The Ranking Module (generates popularity scores)

- Measure the importance of each page
- The measure should be Independent of any query
  - Primarily determined by the link structure of the Web
  - Tempered by some content considerations
- Compute these measures off-line long before any queries are processed
The Ranking Module (generates popularity scores)

- Measure the importance of each page
- The measure should be Independent of any query
  - Primarily determined by the link structure of the Web
  - Tempered by some content considerations
- Compute these measures off-line long before any queries are processed
- Google’s PageRank\(^\text{©}\) technology distinguishes it from all competitors
The Ranking Module (generates popularity scores)

- Measure the importance of each page
- The measure should be Independent of any query
  - Primarily determined by the link structure of the Web
  - Tempered by some content considerations
- Compute these measures off-line long before any queries are processed

- Google’s PageRank© technology distinguishes it from all competitors

Google’s PageRank = Google’s $$$$$
Results 1 - 10 of about 122,000,000 for business intelligence (0.10 seconds)

SAS Business Intelligence
www.SAS.com  Get Better Answers Faster w/ SAS' Award-winning BI Software. Get Info

Business intelligence - Wikipedia, the free encyclopedia
Business intelligence (BI) is a business management term which refers to applications and technologies which are used to gather, provide access to, ... en.wikipedia.org/wiki/Business_intelligence - 43k - Cached - Similar pages

Business Intelligence .com :: The Resource for Business Intelligence
The Business Intelligence resource for business and technical professionals covering a wide range of topics including Performance Management, Data Warehouse ... www.businessintelligence.com/ - 74k - Apr 15, 2007 - Cached - Similar pages

Business Intelligence and Performance Management Software ...
Business intelligence and business performance management software. Reporting, analytics software, budgeting software, balanced scorecard software, ... Stock quote for COGN www.cognos.com/ - 32k - Cached - Similar pages

Oracle Business Intelligence Solutions
The First Comprehensive, Cost-Effective BI Solution Only Oracle delivers a complete, pre-integrated technology foundation to reduce the cost and complexity ... www.oracle.com/solutions/ business_intelligence/index.html - 55k - Cached - Similar pages

Business Intelligence - Management Best Practice Reports
Business Intelligence: Providers of independent reports containing best practice advice, proprietary research findings and case studies for senior managers ... www.business-intelligence.co.uk/ - 18k - Cached - Similar pages

Intelligent Enterprise: Better Insight for Business Decisions

SQL Database Management
Enterprise Data Mgmt Solutions From Dell™. Find Out More Here www.dell.com

Business Intelligence
See what business intelligence can do for you (free interactive demo). www.InformationBuilders.com

MCITP: BI Cert Boot Camp
9-Day MCITP Certification Boot Camp Business Intelligence All Inclusive www.mcseclasses.com

Business Intelligence
Improve information integrity with real-time data integration software www.DataMirror.com

Love Data?
Empower yourself with MS BI Tools via SetFocus' Master's Program www.SetFocus.com

Business Intelligence
Take Your Pick

Amount of Internet search results that Web surfers typically scan before selecting one

- First page of search results: 39%
- First two pages: 19%
- More than first three pages: 10%
- First three pages: 9%
- A few search results*: 23%

*Top results without reading through the whole page

Note: Sample size is 2,369 people
Sources: JupiterResearch; iProspect
Business intelligence - Wikipedia, the free encyclopedia
Business intelligence (BI) is a business management term which refers to applications and technologies which are used to gather, provide access to, ... en.wikipedia.org/wiki/Business_intelligence - 43k - Cached - Similar pages

Business Intelligence - The Resource for Business Intelligence
The Business Intelligence resource for business and technical professionals covering a wide range of topics including Performance Management, Data Warehouse ... www.businessintelligence.com/ - 74k - Apr 15, 2007 - Cached - Similar pages

Business Intelligence and Performance Management Software ...
Business intelligence and business performance management software. Reporting, analytics software, budgeting software, balanced scorecard software, ... www.cognos.com/ - 32k - Cached - Similar pages

Oracle Business Intelligence Solutions
The First Comprehensive, Cost-Effective BI Solution Only Oracle delivers a complete, pre-integrated technology foundation to reduce the cost and complexity ... www.oracle.com/solutions/ business_intelligence/index.html - 55k - Cached - Similar pages

Business Intelligence - Management Best Practice Reports
Business Intelligence: Providers of independent reports containing best practice advice, proprietary research findings and case studies for senior managers ... www.business-intelligence.co.uk/ - 18k - Cached - Similar pages

Intelligent Enterprise: Better Insight for Business Decisions

Sponsored Links
SQL Database Management
Enterprise Data Mgmt Solutions From Dell™. Find Out More Here www.dell.com

Business Intelligence
See what business intelligence can do for you (free interactive demo). www.InformationBuilders.com

MCITP: BI Cert Boot Camp
9-Day MCITP Certification Boot Camp Business Intelligence All Inclusive www.mcseclasses.com

Business Intelligence
Improve information integrity with real-time data integration software www.DataMirror.com

Love Data?
Empower yourself with MS BI Tools via SetFocus' Master's Program www.SetFocus.com

Business Intelligence
The Next Frontiers

The New Age of Google

The Search Giant Has Changed Our Lives. Can Anybody Catch These Guys? By Steven Levy

PLUS: The Future of Digital Voting

Google founders Larry Page and Sergey Brin
Google’s PageRank

(Lawrence Page & Sergey Brin 1998)

The Google Goals

● Create a PageRank $r(P)$ that is not query dependent
  ▶ Off-line calculations — No query time computation

● Let the Web vote with in-links
  ▶ But not by simple link counts
    ─ One link to $P$ from Yahoo! is important
    ─ Many links to $P$ from me is not

● Share The Vote
  ▶ Yahoo! casts many “votes”
    ─ value of vote from Yahoo! is diluted
  ▶ If Yahoo! “votes” for $n$ pages
    ─ Then $P$ receives only $r(Y)/n$ credit from $Y$
Google’s PageRank

(Lawrence Page & Sergey Brin 1998)

The Google Goals

- Create a PageRank $r(P)$ that is not query dependent
  - Off-line calculations — No query time computation
- Let the Web vote with in-links
  - But not by simple link counts
    - One link to $P$ from Yahoo! is important
    - Many links to $P$ from me is not
- Share The Vote
  - Yahoo! casts many “votes”
    - value of vote from Yahoo! is diluted
  - If Yahoo! “votes” for $n$ pages
    - Then $P$ receives only $r(Y)/n$ credit from $Y$
The Google Goals

- Create a PageRank $r(P)$ that is not query dependent
  - Off-line calculations — No query time computation
- Let the Web vote with in-links
  - But not by simple link counts
    - One link to $P$ from Yahoo! is important
    - Many links to $P$ from me is not
- Share The Vote
  - Yahoo! casts many “votes”
    - value of vote from Yahoo! is diluted
  - If Yahoo! “votes” for $n$ pages
    - Then $P$ receives only $r(Y)/n$ credit from $Y$
Google’s PageRank

(Lawrence Page & Sergey Brin 1998)

The Google Goals

- Create a PageRank $r(P)$ that is not query dependent
  - Off-line calculations — No query time computation
- Let the Web vote with in-links
  - But not by simple link counts
    - One link to $P$ from Yahoo! is important
    - Many links to $P$ from me is not
- Share The Vote
  - Yahoo! casts many “votes”
    - value of vote from Yahoo! is diluted
  - If Yahoo! “votes” for $n$ pages
    - Then $P$ receives only $r(Y)/n$ credit from $Y$
PageRank

The Definition

\[ r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \]

\[ B_P = \{ \text{all pages pointing to } P \} \]

\[ |P| = \text{number of out links from } P \]
PageRank

The Definition

\[ r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \]

\[ B_P = \{ \text{all pages pointing to } P \} \]

\[ |P| = \text{number of out links from } P \]

Successive Refinement

Start with \( r_0(P_i) = 1/n \) for all pages \( P_1, P_2, \ldots, P_n \)
PageRank

The Definition

\[ r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \]

\( B_P = \{ \text{all pages pointing to } P \} \)

\(|P| = \text{number of out links from } P\)

Successive Refinement

Start with \( r_0(P_i) = 1/n \) for all pages \( P_1, P_2, \ldots, P_n \)

Iteratively refine rankings for each page

\[ r_1(P_i) = \sum_{P \in B_{P_i}} \frac{r_0(P)}{|P|} \]
PageRank

The Definition

\[ r(P) = \sum_{P \in B_P} \frac{r(P)}{|P|} \]

\[ B_P = \{ \text{all pages pointing to } P \} \]

\[ |P| = \text{number of out links from } P \]

Successive Refinement

Start with \( r_0(P_i) = \frac{1}{n} \) for all pages \( P_1, P_2, \ldots, P_n \)

Iteratively refine rankings for each page

\[ r_1(P_i) = \sum_{P \in B_{P_i}} \frac{r_0(P)}{|P|} \]

\[ r_2(P_i) = \sum_{P \in B_{P_i}} \frac{r_1(P)}{|P|} \]
PageRank

The Definition

\[ r(P) = \sum_{P \in \mathcal{B}_P} \frac{r(P)}{|P|} \]

\[ \mathcal{B}_P = \{\text{all pages pointing to } P\} \]

\[ |P| = \text{number of out links from } P \]

Successive Refinement

Start with \( r_0(P_i) = \frac{1}{n} \) for all pages \( P_1, P_2, \ldots, P_n \)

Iteratively refine rankings for each page

\[ r_1(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_0(P)}{|P|} \]

\[ r_2(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_1(P)}{|P|} \]

\[ \vdots \]

\[ r_{j+1}(P_i) = \sum_{P \in \mathcal{B}_{P_i}} \frac{r_j(P)}{|P|} \]
In Matrix Notation

After Step $k$

$$\pi_k^T = [r_k(P_1), r_k(P_2), \cdots, r_k(P_n)]$$
In Matrix Notation

After Step $k$

$\pi^T_k = [r_k(P_1), r_k(P_2), \cdots, r_k(P_n)]$

$\pi^T_{k+1} = \pi^T_k H$ where $h_{ij} = \begin{cases} 1/|P_i| & \text{if } i \to j \\ 0 & \text{otherwise} \end{cases}$
In Matrix Notation

After Step $k$

- $\pi_k^T = [r_k(P_1), r_k(P_2), \cdots, r_k(P_n)]$

- $\pi_{k+1}^T = \pi_k^T H$ where $h_{ij} = \begin{cases} \frac{1}{|P_i|} & \text{if } i \rightarrow j \\ 0 & \text{otherwise} \end{cases}$

- PageRank vector = $\pi^T = \lim_{k \to \infty} \pi_k^T = \text{eigenvector for } H$

Provided that the limit exists
Tiny Web

\[
H = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}
\]
\[
H = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
0 & 1/2 & 1/2 & 0 & 0 & 0 \\
\end{pmatrix}
\]
Tiny Web

\[ H = \begin{pmatrix}
    P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
    P_1 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\
    P_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
    P_3 & 0 & 0 & 0 & 0 & 0 & 0 \\
    P_4 & 0 & 0 & 0 & 0 & 0 & 0 \\
    P_5 & 0 & 0 & 0 & 0 & 0 & 0 \\
    P_6 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \]
$$H = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
P_1 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\
P_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
P_3 & 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
\end{pmatrix}$$
Tiny Web

\[ H = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
0 & 1/2 & 1/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/2
\end{pmatrix} \]
Tiny Web

\[
H = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
0 & 1/2 & 1/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 \\
0 & 0 & 0 & 1/2 & 0 & 1/2 \\
\end{pmatrix}
\]
\[
H = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
0 & 1/2 & 1/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 \\
0 & 0 & 0 & 1/2 & 0 & 1/2 \\
0 & 0 & 0 & 1 & 0 & 0 \\
\end{pmatrix}
\]
A random walk on the Web Graph
A random walk on the Web Graph

PageRank = $\pi_i = \text{amount of time spent at } P_i$
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote

Markov chain
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and \textit{quality} of votes cast for that page.

Hyperlink as vote
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote

![Diagram showing hyperlink structure]
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and quality of votes cast for that page.
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote
Rank each page corresponding to a search term by number and quality of votes cast for that page.
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote

page 2 is a dangling node
A random walk on the Web Graph

PageRank $= \pi_i = \text{amount of time spent at } P_i$

Dead end page (nothing to click on) — a “dangling node”
A random walk on the Web Graph

PageRank = $\pi_i = \text{amount of time spent at } P_i$

Dead end page (nothing to click on) — a “dangling node”

$\pi^T = (0, 1, 0, 0, 0, 0) = \text{e-vector}$  $\Rightarrow$ Page $P_2$ is a “rank sink”
The Fix

Allow Web Surfers To Make Random Jumps
Ranking with a Random Surfer

- Rank each page corresponding to a search term by number and *quality* of votes cast for that page.

Hyperlink as vote

surfer “teleports”
The Fix

Allow Web Surfers To Make Random Jumps

Replace zero rows with

\[ \frac{e^T}{n} = \left( \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \right) \]

\[
S = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
0 & 1/2 & 1/2 & 0 & 0 & 0 \\
1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\
1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
0 & 0 & 0 & 0 & 1/2 & 1/2 \\
0 & 0 & 0 & 1/2 & 0 & 1/2 \\
0 & 0 & 0 & 1 & 0 & 0 
\end{pmatrix}
\]
The Fix

Allow Web Surfers To Make Random Jumps

— Replace zero rows with $\frac{e^T}{n} = \left( \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \right)$

\[
S = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
P_1 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\
P_2 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\
P_3 & 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
P_4 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\
P_5 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\
P_6 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{pmatrix}
\]

— $S = H + \frac{ae^T}{6}$ is now row stochastic $\implies \rho(S) = 1$
The Fix

Allow Web Surfers To Make Random Jumps

— Replace zero rows with \( \frac{e^T}{n} = \left( \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \right) \)

\[
\mathbf{S} = \begin{pmatrix}
P_1 & P_2 & P_3 & P_4 & P_5 & P_6 \\
P_1 & 0 & 1/2 & 1/2 & 0 & 0 & 0 \\
P_2 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\
P_3 & 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\
P_4 & 0 & 0 & 0 & 0 & 1/2 & 1/2 \\
P_5 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \\
P_6 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{pmatrix}
\]

— \( \mathbf{S} = \mathbf{H} + \frac{a e^T}{6} \) is now row stochastic \( \implies \rho(\mathbf{S}) = 1 \)

— Perron says \( \exists \ \pi^T \geq 0 \) s.t. \( \pi^T = \pi^T \mathbf{S} \) with \( \sum_i \pi_i = 1 \)
Nasty Problem

The Web Is Not Strongly Connected
Nasty Problem

The Web Is Not Strongly Connected

S is reducible

\[
S = \begin{pmatrix}
P_1 & P_2 & P_3 \\
\begin{pmatrix}
P_1 & P_2 & P_3 \\
P_4 & P_5 & P_6 \\
0 & 1/2 & 1/2 \\
1/6 & 1/6 & 1/6 \\
1/3 & 1/3 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1/2 & 1/2 \\
1/2 & 0 & 1/2 \\
1 & 0 & 0
\end{pmatrix}
\end{pmatrix}
\]
Nasty Problem

The Web Is Not Strongly Connected

- **S** is reducible

\[
S = \begin{pmatrix}
P_1 & P_2 & P_3 & | & P_4 & P_5 & P_6 \\
P_1 & 0 & 1/2 & 1/2 & | & 0 & 0 & 0 \\
P_2 & 1/6 & 1/6 & 1/6 & | & 1/6 & 1/6 & 1/6 \\
P_3 & 1/3 & 1/3 & 0 & | & 0 & 1/3 & 0 \\
P_4 & 0 & 0 & 0 & | & 0 & 1/2 & 1/2 \\
P_5 & 0 & 0 & 0 & | & 1/2 & 0 & 1/2 \\
P_6 & 0 & 0 & 0 & | & 1 & 0 & 0 \\
\end{pmatrix}
\]

- Reducible \(\implies\) PageRank vector is not well defined

- Frobenius says **S** needs to be *irreducible* to ensure a unique \(\pi^T > 0\) s.t. \(\pi^T = \pi^T S\) with \(\sum_i \pi_i = 1\)
Irreducibility Is Not Enough

Could Get Trapped Into A Cycle \((P_i \rightarrow P_j \rightarrow P_i)\)
Irreducibility Is Not Enough

Could Get Trapped Into A Cycle \((P_i \rightarrow P_j \rightarrow P_i)\)

- The powers \(S^k\) fail to converge
Irreducibility Is Not Enough

Could Get Trapped Into A Cycle \((P_i \rightarrow P_j \rightarrow P_i)\)

— The powers \(S^k\) fail to converge

— \(\pi_{k+1}^T = \pi_k^T S\) fails to convergence
Irreducibility Is Not Enough

Could Get Trapped Into A Cycle \((P_i \rightarrow P_j \rightarrow P_i)\)

- The powers \(S^k\) fail to converge

- \(\pi_{k+1}^T = \pi_k^T S\) fails to convergence

Convergence Requirement

- Perron–Frobenius requires \(S\) to be primitive
Irreducibility Is Not Enough

Could Get Trapped Into A Cycle \((P_i \rightarrow P_j \rightarrow P_i)\)

- The powers \(S^k\) fail to converge
- \(\pi_k^{T+1} = \pi_k^{T} S\) fails to convergence

Convergence Requirement

- Perron–Frobenius requires \(S\) to be primitive
- No eigenvalues other than \(\lambda = 1\) on unit circle
Irreducibility Is Not Enough

Could Get Trapped Into A Cycle \((P_i \rightarrow P_j \rightarrow P_i)\)

- The powers \(S^k\) fail to converge
- \(\pi_{k+1}^T = \pi_k^T S\) fails to converge

Convergence Requirement

- Perron–Frobenius requires \(S\) to be primitive
- No eigenvalues other than \(\lambda = 1\) on unit circle
- Frobenius proved \(S\) is primitive \(\iff S^k > 0\) for some \(k\)
The Google Fix

Allow A Random Jump From Any Page

\[ G = \alpha S + (1 - \alpha)E > 0, \quad E = ee^T/n, \quad 0 < \alpha < 1 \]
The Google Fix

Allow A Random Jump From Any Page

\[ G = \alpha S + (1 - \alpha)E > 0, \quad E = ee^T/n, \quad 0 < \alpha < 1 \]

\[ G = \alpha H + uv^T > 0 \quad u = \alpha a + (1 - \alpha)e, \quad v^T = e^T/n \]
The Google Fix

Allow A Random Jump From Any Page

\[ G = \alpha S + (1 - \alpha)E > 0, \quad E = ee^T/n, \quad 0 < \alpha < 1 \]

\[ G = \alpha H + uv^T > 0 \quad u = \alpha a + (1 - \alpha)e, \quad v^T = e^T/n \]

PageRank vector \[ \pi^T = \text{left-hand Perron vector of } G \]
Ranking with a Random Surfer

- If a page is “important,” it gets lots of votes from other important pages, which means the random surfer visits it often.

- Simply count the number of times, or *proportion of time*, the surfer spends on each page to create ranking of webpages.
Ranking with a Random Surfer

- If a page is “important,” it gets lots of votes from other important pages, which means the random surfer visits it often.

- Simply count the number of times, or proportion of time, the surfer spends on each page to create ranking of webpages.

### Proportion of Time

<table>
<thead>
<tr>
<th>Page</th>
<th>Proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 1</td>
<td>0.04</td>
</tr>
<tr>
<td>Page 2</td>
<td>0.05</td>
</tr>
<tr>
<td>Page 3</td>
<td>0.04</td>
</tr>
<tr>
<td>Page 4</td>
<td>0.38</td>
</tr>
<tr>
<td>Page 5</td>
<td>0.20</td>
</tr>
<tr>
<td>Page 6</td>
<td>0.29</td>
</tr>
</tbody>
</table>

### Ranked List of Pages

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page 4</td>
</tr>
<tr>
<td>Page 6</td>
</tr>
<tr>
<td>Page 5</td>
</tr>
<tr>
<td>Page 2</td>
</tr>
<tr>
<td>Page 1</td>
</tr>
<tr>
<td>Page 3</td>
</tr>
</tbody>
</table>
The Google Fix

Allow A Random Jump From Any Page

— \( G = \alpha S + (1 - \alpha)E \geq 0 \), \( E = ee^T/n \), \( 0 < \alpha < 1 \)

— \( G = \alpha H + uv^T \geq 0 \)

— PageRank vector \( \pi^T = \text{left-hand Perron vector of } G \)

Some Happy Accidents

— \( x^T G = \alpha x^T H + \beta v^T \) \quad \text{Sparse computations with the original link structure}
The Google Fix

Allow A Random Jump From Any Page

\[ G = \alpha S + (1 - \alpha)E > 0, \quad E = ee^T/n, \quad 0 < \alpha < 1 \]

\[ G = \alpha H + uv^T > 0 \quad u = \alpha a + (1 - \alpha)e, \quad v^T = e^T/n \]

PageRank vector \[ \pi^T = \text{left-hand Perron vector of } G \]

Some Happy Accidents

\[ x^T G = \alpha x^T H + \beta v^T \quad \text{Sparse computations with the original link structure} \]

\[ \lambda_2(G) = \alpha \quad \text{Convergence rate controllable by Google engineers} \]
The Google Fix

Allow A Random Jump From Any Page

- $G = \alpha S + (1 - \alpha)E > 0$, $E = ee^T/n$, $0 < \alpha < 1$

- $G = \alpha H + uv^T > 0$ $u = \alpha a + (1 - \alpha)e$, $v^T = e^T/n$

- PageRank vector $\pi^T = \text{left-hand Perron vector of } G$

Some Happy Accidents

- $x^T G = \alpha x^T H + \beta v^T$ Sparse computations with the original link structure

- $\lambda_2(G) = \alpha$ Convergence rate controllable by Google engineers

- $v^T$ can be any positive probability vector in $G = \alpha H + uv^T$
The Google Fix

Allow A Random Jump From Any Page

- \( G = \alpha S + (1 - \alpha)E \geq 0, \quad E = ee^T/n, \quad 0 < \alpha < 1 \)
- \( G = \alpha H + uv^T > 0 \)
  \( u = \alpha a + (1 - \alpha)e, \quad v^T = e^T/n \)
- PageRank vector \( \pi^T = \text{left-hand Perron vector of } G \)

Some Happy Accidents

- \( x^TG = \alpha x^TH + \beta v^T \quad \text{Sparse computations with the original link structure} \)
- \( \lambda_2(G) = \alpha \quad \text{Convergence rate controllable by Google engineers} \)
- \( v^T \) can be any positive probability vector in \( G = \alpha H + uv^T \)
- The choice of \( v^T \) allows for personalization
Google matrix $G > 0$

Perron-Frobenius guarantees

- existence of $\pi$
- uniqueness of $\pi$
- convergence of algorithm
- rate of convergence: $\lambda_2(G) = \alpha$
Ranking Sports Teams
PageRank applied to Sports

(joint work with Carl Meyer)

- webpages vote with hyperlinks
- losing teams vote with point differentials

- not as successful at ranking and predicting winners as mHITS
each team $i$ gets both offensive rating $o_i$ and defensive rating $d_i$

- mHITS Thesis: A team is a good defensive team (i.e., deserves a high defensive rating $d_j$) when it holds its opponents (particularly strong offensive teams) to low scores. A team is a good offensive team (i.e., deserves a high offensive rating $O_i$) when it scores many points against its opponents (particularly opponents with high defensive ratings).

$$
\begin{pmatrix}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
\text{Duke} & 0 & 7 & 21 & 7 & 0 \\
\text{Miami} & 52 & 0 & 34 & 25 & 27 \\
\text{UNC} & 24 & 16 & 0 & 7 & 3 \\
\text{UVA} & 38 & 17 & 5 & 0 & 14 \\
\text{VT} & 45 & 7 & 30 & 52 & 0 \\
\end{pmatrix}
$$

Graph

Point Matrix $P \geq 0$
mHITS Equations

Summation Notation

\[ d_j = \sum_{i \in I_j} p_{ij} \frac{1}{o_i} \quad \text{and} \quad o_i = \sum_{j \in L_i} p_{ij} \frac{1}{d_j} \]

Matrix Notation: iterative procedure

\[ d^{(k)} = P^T \frac{1}{o^{(k)}} \quad \text{and} \quad o^{(k)} = P \frac{1}{d^{(k-1)}} \]

- related to the Sinkhorn-Knopp algorithm for matrix balancing (uses successive row and column scaling to transform \( P \geq 0 \) into doubly stochastic matrix \( S \))
- P. Knight uses Sinkhorn-Knopp algorithm to rank webpages
mHITS Results: tiny NCAA
(data from Luke Ingram)

<table>
<thead>
<tr>
<th>Team</th>
<th>Off. Rating $o$</th>
<th>Off. Rank</th>
<th>Def. Rating $d$</th>
<th>Def. Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>.4</td>
<td>5th</td>
<td>140</td>
<td>5th</td>
</tr>
<tr>
<td>Miami</td>
<td>1.8</td>
<td>1st</td>
<td>67</td>
<td>2nd</td>
</tr>
<tr>
<td>UNC</td>
<td>.6</td>
<td>4th</td>
<td>97</td>
<td>4th</td>
</tr>
<tr>
<td>UVA</td>
<td>1.0</td>
<td>3rd</td>
<td>80</td>
<td>3rd</td>
</tr>
<tr>
<td>VT</td>
<td>1.4</td>
<td>2nd</td>
<td>34</td>
<td>1st</td>
</tr>
</tbody>
</table>

$r = o/d$ Rating

<table>
<thead>
<tr>
<th>Team</th>
<th>$r = o/d$ Rating</th>
<th>$r = o/d$ Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>0.0003</td>
<td>5th</td>
</tr>
<tr>
<td>Miami</td>
<td>0.027</td>
<td>2nd</td>
</tr>
<tr>
<td>UNC</td>
<td>0.006</td>
<td>4th</td>
</tr>
<tr>
<td>UVA</td>
<td>0.012</td>
<td>3rd</td>
</tr>
<tr>
<td>VT</td>
<td>0.041</td>
<td>1st</td>
</tr>
</tbody>
</table>
Weighted mHITS

- Weighted Point Matrix $\mathbf{\tilde{P}} \geq 0$

  $$\tilde{p}_{ij} = w_{ij} \ p_{ij} \quad (w_{ij} = \text{weight of matchup between teams } i \ \text{and } j)$$

- possible weightings $w_{ij}$

---

**Graphs:**

- **Linear:** $w$ vs. $t$ from $t_0$ to $t_f$
- **Logarithmic:** $w$ vs. $t$ from $t_0$ to $t_f$
- **Exponential:** $w$ vs. $t$ from $t_0$ to $t_f$
- **Step Function:** $w$ vs. $t$ from $t_0$ to $t_f$
mHITS Results: full NCAA
(image from Neil Goodson and Colin Stephenson)
mHITS Results: full NCAA

- weightings can easily be applied to all ranking methods
  ⇒ interesting possibilities for weighted webpage ranking

<table>
<thead>
<tr>
<th>Method</th>
<th>ESPN score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Massey Linear</td>
<td>1450</td>
</tr>
<tr>
<td>Massey Log</td>
<td>1450</td>
</tr>
<tr>
<td>mHITS Log</td>
<td>1450</td>
</tr>
<tr>
<td>Massey Step</td>
<td>1420</td>
</tr>
<tr>
<td>Massey Exponential</td>
<td>1400</td>
</tr>
<tr>
<td>mHITS Step</td>
<td>1320</td>
</tr>
<tr>
<td>Massey Uniform</td>
<td>1310</td>
</tr>
<tr>
<td>mHITS Linear</td>
<td>1310</td>
</tr>
<tr>
<td>mHITS Uniform</td>
<td>1310</td>
</tr>
<tr>
<td>Colley Linear</td>
<td>1100</td>
</tr>
<tr>
<td>Colley Log</td>
<td>1010</td>
</tr>
</tbody>
</table>
mHITS Point Matrix $P \geq 0$

Perron-Frobenius guarantees (for irreducible $P$ with total support)

- existence of $o$ and $d$
- uniqueness of $o$ and $d$
- convergence of mHITS algorithm
- rate of convergence of mHITS algorithm

$$\sigma_2^2(S), \text{ where } S = D(1/o) P D(1/d)$$
Recommendation Systems
**Recommendation Systems**

\[ A \geq 0 \]

<table>
<thead>
<tr>
<th></th>
<th>User 1</th>
<th>User 2</th>
<th>...</th>
<th>User n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item 1</td>
<td>1</td>
<td>5</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>Item 2</td>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>Item m</td>
<td>0</td>
<td>1</td>
<td>...</td>
<td>2</td>
</tr>
</tbody>
</table>

- **purchase history matrix**

- **Find similar users**
- **Find similar items**
- **Cluster users into classes**

**EX:** use NMF so that \( A_{m \times n} \approx W_{m \times k}H_{k \times n} \)
NMF Algorithm: Lee and Seung 2000

Mean Squared Error objective function

\[
\min ||A - WH||^2_F
\]
\[
s.t. \quad W, H \geq 0
\]

\[
W = \text{abs} \text{randn}(m,k));
\]
\[
H = \text{abs} \text{randn}(k,n));
\]
for i = 1 : maxiter
\[
H = H .* (W^T A) ./ (W^T WH + 10^{-9});
\]
\[
W = W .* (AH^T) ./ (WHH^T + 10^{-9});
\]
end

Many parameters affect performance (k, obj. function, sparsity constraints, algorithm, etc.).

— NMF is not unique!

(proof of convergence to fixed point based on E-M convergence proof)
Interpretation with NMF

- Columns of $W$ are the underlying basis vectors, i.e., each of the $n$ columns of $A_{m \times n}$ can be built from $k$ columns of $W_{m \times k}$.

- Columns of $H$ give the weights associated with each basis vector.

$$A_{*1} \approx WH_{*1} = \begin{bmatrix} \vdots \end{bmatrix} w_1 h_{11} + \begin{bmatrix} \vdots \end{bmatrix} w_2 h_{21} + \cdots + \begin{bmatrix} \vdots \end{bmatrix} w_k h_{k1}$$

$$A_{*2} = \begin{pmatrix} 5 \\ 0 \\ 0 \\ 11 \\ 1 \end{pmatrix} \approx WH_{*1} = \begin{bmatrix} 1.4 \\ 0 \\ 0 \\ 2.4 \\ 0 \end{bmatrix} .7 + \begin{bmatrix} 0 \\ 3.2 \\ 5.1 \\ 0 \\ 0 \end{bmatrix} .003 + \cdots + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 4.4 \\ 2.8 \end{bmatrix} .2$$

- $W, H \geq 0 \Rightarrow$ immediate interpretation (additive parts-based rep.)
Netflix

17,770 movies, ≈ .5 million users

\[
A = \begin{pmatrix}
1 & 5 & \ldots & 0 \\
0 & 0 & \ldots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 1 & \ldots & 2 \\
\end{pmatrix}
\]

- Find similar users
- Find similar movies
- Cluster users and movies into classes
- Rank users
- Rank movies
Clustering Netflix
(movie-movie matrix; 17770 x 17770)
Netflix

17770 movies, \( \approx 0.5 \text{ million users} \)

\[
\begin{pmatrix}
\text{User 1} & \text{User 2} & \ldots & \text{User n} \\
\text{movie 1} & 1 & 5 & \ldots & 0 \\
\text{movie 2} & 0 & 0 & \ldots & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\text{movie m} & 0 & 1 & \ldots & 2
\end{pmatrix}
\]

- Find similar users
- Find similar movies
- Cluster users and movies into classes
- Rank users
- Rank movies
mHITS on Netflix

each movie \( i \) gets a rating \( m_i \) and each user gets a rating \( u_j \)

- mHITS Thesis: A movie is a good (i.e., deserves a high rating \( m_i \)) if it gets high ratings from good (i.e., discriminating) users. A user is good (i.e., deserves a high rating \( u_j \)) when his or her ratings match the true rating of a movie.

\[
\begin{align*}
\text{mHITS Netflix Algorithm}
\end{align*}
\]

\[
\begin{align*}
\mathbf{u} &= \mathbf{e}; \\
\text{for } i &= 1 : \text{maxiter} \\
\mathbf{m} &= \mathbf{A} \mathbf{u}; \\
\mathbf{m} &= \frac{5(\mathbf{m} - \min(\mathbf{m}))}{\max(\mathbf{m}) - \min(\mathbf{m})}; \\
\mathbf{u} &= \frac{1}{((\mathbf{R} - (\mathbf{R} > 0) \odot (\mathbf{em}^T))^2 \mathbf{e})}; \\
\end{align*}
\]
Netflix mHITS results

subset: 1500 “super users”  (rate $\geq$ 1000 movies)

1st
Raiders of the Lost Ark

2nd
Silence of the Lambs

3rd
The Sixth Sense

4th
Shawshank Redemption

5th
LOR: Fellowship of the Ring

6th
The Matrix

7th
LOR: The Two Towers

8th
Pulp Fiction

9th
LOR: The Return of the King

10th
Forrest Gump

11th
The Usual Suspects

12th
American Beauty

13th
Pirates of the Carribbean: Black Pearl

14th
The Godfather

15th
Braveheart
The Enron Email Dataset

(data from Mike Berry)

- PRIVATE email collection of 150 Enron employees during 2001
- 92,000 terms and 65,000 messages
- Term-by-Message Matrix

\[
\begin{pmatrix}
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\text{subpoena} & 2 & 0 & 1 & \cdots \\
\text{dynegy} & 0 & 3 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{pmatrix}
\]
Text Mining Applications

- Data compression
- Find similar terms
- Find similar documents
- Cluster documents
- Cluster terms
- Topic detection and tracking
Unclustered Enron A
NMF-clustered Enron A
### Clustering the Enron Email Dataset
*(image from Mike Berry)*

<table>
<thead>
<tr>
<th>Feature Index ($k$)</th>
<th>Cluster Size</th>
<th>Topic Description</th>
<th>Dominant Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>497</td>
<td>California</td>
<td>ca, epuc, gov, socalgas, sempra, org, sce, gmssr, aelaw, ci</td>
</tr>
<tr>
<td>23</td>
<td>43</td>
<td>Louise Kitchen named top woman by Fortune</td>
<td>evp, fortune, britain, woman, ceo, avon, fiorinal, cfo, hewlett, packard</td>
</tr>
<tr>
<td>26</td>
<td>231</td>
<td>Fantasy football</td>
<td>game, wr, qb, play, rb, season, injury, updated, fantasy, image</td>
</tr>
<tr>
<td>33</td>
<td>233</td>
<td>Texas longhorn football newsletter</td>
<td>UT, orange, longhorn[s], texas, true, truorange, recruiting, oklahoma defensive</td>
</tr>
<tr>
<td>34</td>
<td>65</td>
<td>Enron collapse</td>
<td>partnership[s], fastow, shares, sec, stock, shareholder, investors, equity, lay</td>
</tr>
<tr>
<td>39</td>
<td>235</td>
<td>Emails about India</td>
<td>dahhol, dpc, india, mseqb, maharashtra, indian, lenders, delhi, foreign, minister</td>
</tr>
<tr>
<td>46</td>
<td>127</td>
<td>Enron collapse</td>
<td>dow, debt, reserved, wall, copyright jones, cents, analysts, reuters, spokesman</td>
</tr>
</tbody>
</table>
Tracking Enron clusters over time
(image from Mike Berry)
Meta-Algorithms
Rank Aggregation
Rank Aggregation

- average rank
- Borda count
- simulated data
- graph theory
## Average Rank

<table>
<thead>
<tr>
<th></th>
<th>mHITS ($r = o/d$)</th>
<th>Massey</th>
<th>Colley</th>
<th>Average Rating</th>
<th>Average Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>5th</td>
<td>5th</td>
<td>5th</td>
<td>$\bar{5}$</td>
<td>5th</td>
</tr>
<tr>
<td>Miami</td>
<td>2nd</td>
<td>1st</td>
<td>1st</td>
<td>$\bar{1.3}$</td>
<td>1st</td>
</tr>
<tr>
<td>UNC</td>
<td>4th</td>
<td>4th</td>
<td>3rd</td>
<td>3.6</td>
<td>4th</td>
</tr>
<tr>
<td>UVA</td>
<td>3rd</td>
<td>3rd</td>
<td>4th</td>
<td>3.3</td>
<td>3rd</td>
</tr>
<tr>
<td>VT</td>
<td>1st</td>
<td>2nd</td>
<td>2nd</td>
<td>1.6</td>
<td>2nd</td>
</tr>
</tbody>
</table>
Borda Count

- for each ranked list, each item receives a score equal to the number of items it outranks.

<table>
<thead>
<tr>
<th></th>
<th>mHITS (r = o/d)</th>
<th>Massey</th>
<th>Colley</th>
<th>Borda Count</th>
<th>Borda Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5th</td>
</tr>
<tr>
<td>Miami</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>11</td>
<td>1st</td>
</tr>
<tr>
<td>UNC</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4th</td>
</tr>
<tr>
<td>UVA</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>3rd</td>
</tr>
<tr>
<td>VT</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>10</td>
<td>2nd</td>
</tr>
</tbody>
</table>
3 ranked lists

- VT beats Miami by 1 point, UVA by 2 points, . . .
- Miami beats UVA by 1 point, UNC by 2 points, . . .
- UVA beats UNC by 1 point, Duke by 2 points
- UNC beats Duke by 1 point

repeat for each ranked list ⇒ generates game scores for teams

Simulated Game Data
Simulated Data

Ranked Lists from Various Methods

- Markov
- Colley
- mHITS

generates

Simulated Game Data

<table>
<thead>
<tr>
<th></th>
<th>Team 1</th>
<th>Team 2</th>
<th>...</th>
<th>Team n</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>7-4</td>
<td>17-8</td>
<td>...</td>
<td>6-21</td>
</tr>
<tr>
<td>Team 2</td>
<td>1-5</td>
<td>4-2</td>
<td></td>
<td>12-10</td>
</tr>
<tr>
<td>Team n</td>
<td>3-11</td>
<td>6-8</td>
<td></td>
<td>13-19</td>
</tr>
<tr>
<td></td>
<td>1-3</td>
<td>15-21</td>
<td></td>
<td>9-12</td>
</tr>
</tbody>
</table>

that becomes input to

Combiner Method

that creates

Aggregated List
Graph Theory
more voting

- ranked lists are used to form weighted graph
- possible weights
  * $w_{ij} =$ # of ranked lists having $i$ below $j$
  * $w_{ij} =$ sum of rank differences of lists having $i$ below $j$

- run algorithm (e.g., Markov, PageRank, HITS) to determine most important nodes
Aggregation

Rating Aggregation
Rating Aggregation

rating vectors

- form rating differential matrix $\mathbf{R}$ for each rating vector
Rating Aggregation

crating differential matrices $R \geq 0$

- differing scales $\rightarrow$ normalize
Rating Aggregation

rating differential matrices

\[
\begin{align*}
\tilde{R}_{\text{Massey}} & = \begin{pmatrix}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
0 & 0 & 0 & 0 & 0 \\
.1792 & 0 & 0 & 0 & 0 \\
.1367 & 0 & 0 & 0 & 0 \\
.0892 & 0 & 0 & 0 & 0 \\
.1783 & 0 & 0 & 0 & 0 \\
\end{pmatrix} \\
\tilde{R}_{\text{Colley}} & = \begin{pmatrix}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
0 & 0 & 0 & 0 & 0 \\
.2 & 0 & 0 & 0 & 0 \\
.1 & 0 & 0 & 0 & 0 \\
.0517 & 0 & 0 & 0 & 0 \\
.1517 & 0 & 0 & 0 & 0 \\
\end{pmatrix} \\
\tilde{R}_{mHITS} & = \begin{pmatrix}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
0 & 0 & 0 & 0 & 0 \\
.1237 & 0 & 0 & 0 & 0 \\
.0155 & 0 & 0 & 0 & 0 \\
.0464 & 0 & 0 & 0 & 0 \\
.1959 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\end{align*}
\]
Rating Aggregation

rating differential matrices

- combine into one matrix \[ \rightarrow \text{AVERAGE} \]
Rating Aggregation

rating differential matrices

- combine into one matrix  → AVERAGE
Rating Aggregation

average rating differential matrix $\mathbf{R}_{\text{average}} \geq 0$

\[
\bar{\mathbf{R}}_{\text{average}} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0.1676 & 0 & 0.1058 & 0.1052 & 0.0164 & 0 \\
0.0841 & 0 & 0 & 0.0161 & 0 & 0 \\
0.0624 & 0 & 0.0167 & 0 & 0 & 0 \\
0.1753 & 0.0241 & 0.1135 & 0.1129 & 0 & 0
\end{pmatrix}
\]

- run ranking method
  - Markov method on $\mathbf{R}_{\text{average}}^T$
  - row sums of $\mathbf{R}_{\text{average}}$ / col sums of $\mathbf{R}_{\text{average}}$
  - Perron vector of $\mathbf{R}_{\text{average}}$
Rating Aggregation

average rating differential matrix $\mathbf{R}_{average} \geq 0$

- run ranking method
  - Markov method on $\mathbf{R}_{average}^T$
  - row sums of $\mathbf{R}_{average}$ / col sums of $\mathbf{R}_{average}$
  - Perron vector of $\mathbf{R}_{average}$

<table>
<thead>
<tr>
<th>Team</th>
<th>Method 1 $r = o/d$</th>
<th>Method 2 Markov $r$</th>
<th>Method 3 Perron $r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>0</td>
<td>.020</td>
<td>.27</td>
</tr>
<tr>
<td>Miami</td>
<td>16.4</td>
<td>.465</td>
<td>.58</td>
</tr>
<tr>
<td>UNC</td>
<td>.4</td>
<td>.025</td>
<td>.34</td>
</tr>
<tr>
<td>UVA</td>
<td>3</td>
<td>.024</td>
<td>.33</td>
</tr>
<tr>
<td>VT</td>
<td>26.0</td>
<td>.466</td>
<td>.61</td>
</tr>
</tbody>
</table>
Cluster Aggregation

many clustering algorithms = many clustering results

⇒ Can we combine many results to make one super result?

Cluster Aggregation Algorithm

1. Create aggregation matrix $F \geq 0$

   $f_{ij} = \# \text{ of methods having items } i \text{ and } j$
   in the same cluster

2. Run favorite clustering method on $F$
Cluster Aggregation Example

Method 1

<table>
<thead>
<tr>
<th>item</th>
<th>cluster assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Method 2

<table>
<thead>
<tr>
<th>item</th>
<th>cluster assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Method 3

<table>
<thead>
<tr>
<th>item</th>
<th>cluster assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

Cluster Aggregated Graph

Cluster Aggregated Results

Fiedler using just one eigenvector

Fiedler using two eigenvectors
Conclusions

- Applied problems often have nonnegative data.

- This nonnegativity is exploitable structure.
  - ⇒ proofs for convergence and rates of convergence
  - ⇒ proofs for existence and uniqueness

- Nonnegativity is easily interpretable. Often it pays to maintain nonnegativity throughout the modeling process.
  - ⇒ nonnegative matrix factorizations
  - ⇒ nonnegative rating vectors