The Science of Ranking Items:
from webpages to teams to movies

Amy Langville
Mathematics Department
College of Charleston

PhilOpt 1/30/2009
Collaborators

- Carl Meyer, N. C. State University
- College of Charleston (current and former students)
  - Luke Ingram
  - Kathryn Pedings
  - Neil Goodson
  - Colin Stephenson
  - Emmie Douglas
Outline

• Popular Ranking Methods
  — Massey
  — Colley
  — Markov
  — HITS

• New Ranking Methods
  — Rank Differential Method
  — Rating Differential Method

• Aggregation
  — Methods of Comparison
  — Rank Aggregation
  — Rating Aggregation
Popular Ranking Methods
Ranking

Problem

given data of $n$ items, create a ranked list of these items
\(\text{(e.g., pairwise comparisons)}\)

Applications

- webpages (PageRank, HITS, SALSA, ...)
- sports teams (Massey, Colley, Markov, mHITS, ...)
- recommendation systems (Netflix movies, Amazon books, ...)

Related Problem

cluster $n$ items into groups
The Data

\[ A \geq 0 \]

**Sports Examples**

\[ A = \text{team} \begin{pmatrix} \text{team} \\ \text{stats} \end{pmatrix} ; \quad A = \text{team} \begin{pmatrix} \text{team} \\ \text{stat differentials} \end{pmatrix} \]

\[ A = \text{game} \begin{pmatrix} \text{team} \\ \text{wins} & \text{losses} \end{pmatrix} \]
Popular Ranking Methods

- **Massey**: symmetric linear system $Mr = p$
- **Colley**: s.p.d linear system $Cr = b$
- **Markov**: irreducible eigensystem $Vr = r$, where $V \geq 0$
- **mHITS**: Sinkhorn-Knopp algorithm on $P \geq 0$
Popular Ranking Methods

- **Massey**: symmetric linear system $Mr = p$
- **Colley**: s.p.d linear system $Cr = b$
- **Markov**: irreducible eigensystem $Vr = r$, where $V \geq 0$
- **mHITS**: Sinkhorn-Knopp algorithm on $P \geq 0$

Ranking methods are adapted to fit the application (webpages, sports teams, movies, etc.)
Popular Ranking Methods

Markov Method
Markov Method

Voting Matrix $V \succeq 0$

- losers vote with points given up (or some other statistic)
- winners and losers vote with points given up
- losers vote with point differentials

$$V = \begin{pmatrix}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
\text{Duke} & 0 & 45 & 3 & 31 & 45 \\
\text{Miami} & 0 & 0 & 0 & 0 & 0 \\
\text{UNC} & 0 & 18 & 0 & 0 & 27 \\
\text{UVA} & 0 & 8 & 2 & 0 & 38 \\
\text{VT} & 0 & 20 & 0 & 0 & 0
\end{pmatrix}$$

- vote with multiple statistics

$$V = \alpha_1 V_1 + \alpha_2 V_2 + \cdots + \alpha_k V_k$$
## Fair Weather Fan’s Random Walk

- **row normalize to make $V$ stochastic**

\[
V = \begin{pmatrix}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
0 & 45/124 & 3/124 & 31/124 & 45/124 \\
1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\
0 & 18/45 & 0 & 0 & 27/45 \\
0 & 8/48 & 2/48 & 0 & 38/48 \\
0 & 1 & 0 & 0 & 0
\end{pmatrix}
\]

- **Check/enforce irreducibility**

- **Solve eigensystem:** $Vr = r$

- **$r$** = fair weather fan’s long-term visit proportions

- **not as successful at ranking and predicting winners as mHITS** (coming soon)
Nonnegativity and Markov

- enforce irreducibility, aperiodicity
\[ \bar{V} = V + \epsilon e e^T \geq 0 \]

- P-F guarantees existence and uniqueness of \( r \)

- P-F guarantees convergence and rate of convergence of power method on \( V \)
Popular Ranking Methods

mHITS Method
mHITS Method

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).

![Graph](image)

**Point Matrix** $P \geq 0$

$P = \begin{pmatrix}
  0 & 7 & 21 & 7 & 0 \\
  52 & 0 & 34 & 25 & 27 \\
  24 & 16 & 0 & 7 & 3 \\
  38 & 17 & 5 & 0 & 14 \\
  45 & 7 & 30 & 52 & 0
\end{pmatrix}$
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>

Graph showing connections and ratings:

- Duke to Miami: 38
- Duke to VT: 52
- Miami to VT: 7
- UNC to UVA: 5
- UVA to VT: 14
- VT to UNC: 25
- Duke to UVA: 7
- UNC to Miami: 17
- Duke to UNC: 25
- VT to Duke: 5
- Miami to UVA: 14
- VT to Miami: 7
- UVA to UNC: 25
- VT to UVA: 14

<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 $\tilde{P} \geq 0$

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

- possible weightings $w_{ij}$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{weighting.png}
\end{figure}
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
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

$$\sigma_2^2(S), \text{ where } S = D(1/o)PD(1/d)$$
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.

**mHITS Netflix Algorithm**

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

17770 movies, ≈ .5 million users

David Gleich’s subset: 1500 “super users” (rate ≥ 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
New Ranking Methods
## Ranking Philosophies

### Old

<table>
<thead>
<tr>
<th>Team</th>
<th>Data</th>
<th>Method</th>
<th>Rating Vector</th>
<th>Ranking Vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>7-4 17-8 6-21</td>
<td><strong>Massey</strong></td>
<td>(-1.4)</td>
<td>6</td>
</tr>
<tr>
<td>Team 2</td>
<td>1-5 4-2 12-10</td>
<td><strong>Mr = p</strong></td>
<td>4.3</td>
<td>3</td>
</tr>
<tr>
<td>Team n</td>
<td>3-11 6-8 13-19</td>
<td></td>
<td>.5</td>
<td>5</td>
</tr>
<tr>
<td>Team n</td>
<td>1-3 15-21 9-12</td>
<td></td>
<td>1.2</td>
<td>4</td>
</tr>
<tr>
<td>Team n</td>
<td>11-10 8-4 5-2</td>
<td></td>
<td>-3.1</td>
<td>7</td>
</tr>
<tr>
<td>Team n</td>
<td>11-9 12-8 11-8</td>
<td></td>
<td>5.3</td>
<td>2</td>
</tr>
<tr>
<td>Team n</td>
<td></td>
<td></td>
<td>7.9</td>
<td>1</td>
</tr>
</tbody>
</table>
### Old

#### Data

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

#### Method

\[ \text{Massey} \]

\[ Mr = p \]

#### Rating Vector

\[ \begin{bmatrix} -1.4 \\ 4.3 \\ .5 \\ 1.2 \\ -3.1 \\ 5.3 \\ 7.9 \end{bmatrix} \]

#### Ranking Vector

\[ \begin{bmatrix} 6 \\ 3 \\ 5 \\ 4 \\ 7 \\ 2 \\ 1 \end{bmatrix} \]

### New

#### Data

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

#### Method

\[ \text{Rank Differential} \]

#### Ranking Vector

\[ \begin{bmatrix} 6 \\ 3 \\ 5 \\ 4 \\ 7 \\ 2 \\ 1 \end{bmatrix} \]
New Ranking Methods

Rank Differential Method
- Every ranking vector is a permutation
- Relative positions matter
- One-to-one mapping:

\[
\begin{bmatrix}
\text{ranking vector}
\end{bmatrix} \leftrightarrow \begin{bmatrix}
\text{rank differential matrix} \ R
\end{bmatrix}
\]
Ranking Vector

- Every ranking vector is a permutation
- Relative positions matter
- One-to-one mapping:

```
Example:

\[
\begin{bmatrix}
2 \\
1 \\
3
\end{bmatrix}
\leftarrow
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 2 \\
0 & 0 & 0
\end{bmatrix}
\]
```
Ranking Vector

- Every ranking vector is a permutation
- Differences in position have meaning
- One-to-one mapping:

  \[
  \begin{pmatrix}
  2 \\
  1 \\
  3 \\
  \end{pmatrix}
  \leftrightarrow
  \begin{pmatrix}
  0 & 0 & 1 \\
  1 & 0 & 2 \\
  0 & 0 & 0 \\
  \end{pmatrix}
  \]

  Example:

- Every rank differential matrix $\mathbf{R}$ is a reordering of the fundamental rank differential matrix $\hat{\mathbf{R}}$

  \[
  \hat{\mathbf{R}} = \begin{pmatrix}
  0 & 1 & 2 & 3 & 4 & 5 \\
  \begin{pmatrix}
  1 & 0 & 1 & 2 & 3 & 4 \\
  2 & 0 & 0 & 1 & 2 & 3 \\
  3 & 0 & 0 & 0 & 1 & 2 \\
  4 & 0 & 0 & 0 & 0 & 1 \\
  5 & 0 & 0 & 0 & 0 & 0 \\
  \end{pmatrix}
  \end{pmatrix}
  \]

  corresponds to ranking vector

  \[
  \begin{pmatrix}
  1 \\
  2 \\
  3 \\
  4 \\
  5 \\
  \end{pmatrix}
  \]
Rank Differential Method

Input Matrix: \( D \)  
Output: permutation Matrix: \( q \)
Rank Differential Method

Input Matrix: $D$  Output: permutation Matrix: $q$

GOAL: **Find symmetric reordering of $D$ that**

$$\min_q \| D(q, q) - \hat{R} \|$$

$D =$ data differential matrix  (e.g., Markov voting matrix $V$)

$q =$ permutation vector

$\hat{R} =$ fundamental rank differential matrix
## Rank Differential Example

\[
\begin{align*}
D &= \begin{pmatrix}
Duke & Miami & UNC & UVA & VT \\
Duke & 0 & 0 & 0 & 0 & 0 \\
Miami & 45 & 0 & 18 & 8 & 20 \\
UNC & 3 & 0 & 0 & 2 & 0 \\
UVA & 31 & 0 & 0 & 0 & 0 \\
VT & 45 & 0 & 27 & 38 & 0 \\
\end{pmatrix} \\
\hat{R} &= \begin{pmatrix}
1 & 0 & 1 & 2 & 3 & 4 & 5 \\
2 & 0 & 0 & 1 & 2 & 3 & \ \\
3 & 0 & 0 & 0 & 1 & 2 & \ \\
4 & 0 & 0 & 0 & 0 & 1 & \ \\
5 & 0 & 0 & 0 & 0 & 0 & \\
\end{pmatrix}
\end{align*}
\]

- Find ordering of teams that brings \( D \) closest to \( \hat{R} \)
### Rank Differential Example

#### Matrix D:

<table>
<thead>
<tr>
<th></th>
<th>Duke</th>
<th>Miami</th>
<th>UNC</th>
<th>UVA</th>
<th>VT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duke</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Miami</td>
<td>45</td>
<td>0</td>
<td>18</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>UNC</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>UVA</td>
<td>31</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VT</td>
<td>45</td>
<td>0</td>
<td>27</td>
<td>38</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Matrix R:

<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</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Find ordering of teams that brings $D$ closest to $\hat{R}$

Normalize
## Rank Differential Example

\[
D = \begin{pmatrix}
Duke & Miami & UNC & UVA & VT \\
Duke & 0 & 0 & 0 & 0 & 0 \\
Miami & .19 & 0 & .08 & .03 & .08 \\
UNC & .01 & 0 & 0 & .01 & 0 \\
UVA & .13 & 0 & 0 & 0 & 0 \\
VT & .19 & 0 & .11 & .16 & 0 \\
\end{pmatrix}
\]

\[
\hat{R} = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & .05 & .10 & .15 & .20 \\
2 & 0 & 0 & .05 & .10 & .15 \\
3 & 0 & 0 & 0 & .05 & .10 \\
4 & 0 & 0 & 0 & 0 & .05 \\
5 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

- Find ordering of teams that brings \( D \) closest to \( \hat{R} \)

**Normalize**
### Rank Differential Example

**D**

\[
D = \begin{pmatrix}
Duke & Miami & UNC & UVA & VT \\
Duke & 0 & 0 & 0 & 0 & 0 \\
Miami & .19 & 0 & .08 & .03 & .08 \\
UNC & .01 & 0 & 0 & .01 & 0 \\
UVA & .13 & 0 & 0 & 0 & 0 \\
VT & .19 & 0 & .11 & .16 & 0 \\
\end{pmatrix}
\]

**R**

\[
R = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & .05 & .10 & .15 & .20 \\
2 & 0 & 0 & .05 & .10 & .15 \\
3 & 0 & 0 & 0 & .05 & .10 \\
4 & 0 & 0 & 0 & 0 & .05 \\
5 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

- **Optimal ordering:** \( q = [5, 2, 4, 3, 1] \)

**D(q, q)**

\[
D(q, q) = \begin{pmatrix}
Duke & Miami & UNC & UVA & VT \\
Duke & 0 & 0 & .16 & .11 & .19 \\
Miami & .08 & 0 & .03 & .01 & .16 \\
UNC & 0 & 0 & 0 & 0 & .13 \\
UVA & 0 & 0 & .01 & 0 & .01 \\
VT & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
\hat{R} = \begin{pmatrix}
1 & 2 & 3 & 4 & 5 \\
1 & 0 & .05 & .10 & .15 & .20 \\
2 & 0 & 0 & .05 & .10 & .15 \\
3 & 0 & 0 & 0 & .05 & .10 \\
4 & 0 & 0 & 0 & 0 & .05 \\
5 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]
Rank Differential Example

Reordered $\mathbf{D}$

$\hat{\mathbf{R}}$
Rank Differential Example

Reordered $D$

$mHITS$ method

Duke $5^{th}$

Miami $2^{nd}$

UNC $3^{rd}$

UVA $4^{th}$

VT $1^{st}$

rank differential method

Duke $5^{th}$

Miami $2^{nd}$

UNC $4^{th}$

UVA $3^{rd}$

VT $1^{st}$
Gottlieb 1982
Gottlieb 1982
The Optimization Problem

- $\min_q \| D(q, q) - \hat{R} \|$ is equivalent to

  $\min_Q \| Q^T D Q - \hat{R} \|

  \text{s.t.} \quad Q e = e

  \quad e^T Q = e^T

  \quad q_{ij} \in \{0, 1\}$  \hfill (BIP)
The Optimization Problem

- \( \min_q \| D(q, q) - \hat{R} \| \) is equivalent to

\[
\min_Q \| Q^T D Q - \hat{R} \|
\]

s.t. \( Q e = e \)

\( e^T Q = e^T \)

\( q_{ij} \in \{0, 1\} \) \hspace{1cm} (BIP)

- If the Frobenius norm is used, \( \min_q \| Q^T D Q - \hat{R} \|_F \iff \max_Q \operatorname{trace}(Q^T D Q \hat{R}) \)

\[
\max_Q \operatorname{trace}(Q^T D Q \hat{R})
\]

s.t. \( Q e = e \)

\( e^T Q = e^T \)

\( q_{ij} \in \{0, 1\} \) \hspace{1cm} (QBIP)
Normalization is unnecessary

\[
\max_Q \quad \text{trace} \left( Q^T \alpha D Q \beta \hat{R} \right) \iff \max_Q \quad \alpha \beta \quad \text{trace} \left( Q^T D Q \hat{R} \right)
\]

- Normalizing \( D \) and \( \hat{R} \) only changes the objective function by a scalar multiple, thus, the optimal solution is unchanged.
Calculating the Objective Function

\[ f = \text{trace}(Q^T D Q \hat{R}) \]

- The structure of \( \hat{R} \) makes calculating \( f \) cheap.

**INPUT:**
- \( D \), data matrix of size \( n \times n \)
- \( q \), permutation vector of size \( n \times 1 \)

**OUTPUT:** \( f \)

```matlab
[sortedq, index] = sort(q);
f = 0;
for i = 2:n
    for j = 1:i-1
        f = f + (i-i) * D(index(i),index(j));
    end
end
```
Solving the Optimization Problem

\[
\min_q \|D(q, q) - \hat{R}\|
\]

- Huge solution space: \( \exists n! \) permutations \( q \) (related to TSP; NP-hard)

**Evolutionary Algorithm**

- initialize population with \( k=10 \) solutions \( \{x_1, x_2, \ldots, x_k\} \)
- until convergence
- compute fitness \( \|D(x_i, x_i) - \hat{R}\| \) for each \( x_i \)
- create new population by
  - copy 3 fittest \( x_i \) into next generation
  - pair 6 fittest \( x_i \) and mate with rank aggregation (coming soon)
  - mutate next 3 \( x_i \) with flip, invert, reversal operators
  - insert 1 immigrant \( x_i \) using random permutation

guaranteed to converge to global min (Fogel, Michelawicz) but slow
New Ranking Methods

Rating Differential Method
Rating Differential Method

- Rank mapping:

- Rating Mapping:
Rating Differential Method

- **Rank mapping:**
  - Example:

- **Rating Mapping:**
  - Example:
Rating Differential Method

- Rank mapping:
  
- Rating Mapping:
  
- No fundamental rating differential matrix BUT there is a fundamental form for rating differential matrix
A rating differential matrix $\mathbf{R}$ is in *fundamental form* if

- $r_{ij} = 0$, $\forall i \geq j$ (strictly upper triangular)
- $r_{ij} \leq r_{ik}$, $\forall i \geq j \leq k$ (ascending order across rows)
- $r_{ij} \geq r_{kj}$, $\forall j \geq i \leq k$ (descending order down columns)
A rating differential matrix $R$ is in fundamental form if

\[
\begin{align*}
    r_{ij} &= 0, \quad \forall \ i \geq j \quad \text{(strictly upper triangular)} \\
    r_{ij} &\leq r_{ik}, \quad \forall \ i \geq j \leq k \quad \text{(ascending order across rows)} \\
    r_{ij} &\geq r_{kj}, \quad \forall \ j \geq i \leq k \quad \text{(descending order down columns)}
\end{align*}
\]

**GOAL:** Find symmetric reordering of $D$ so that $D(q,q)$

\[
\min\limits_q \text{(\# violations of fundamental form constraints)}
\]
Rating Differential Method

A rating differential matrix $R$ is in *fundamental form* if

- $r_{ij} = 0$, $\forall i \geq j$ (strictly upper triangular)
- $r_{ij} \leq r_{ik}$, $\forall i \geq j \leq k$ (ascending order across rows)
- $r_{ij} \geq r_{kj}$, $\forall j \geq i \leq k$ (descending order down columns)

**GOAL:** Find symmetric reordering of $D$ so that $D(q,q)$

$$
\min_{q} \left( \text{# violations of fundamental form constraints} \right)
$$

(or)

$$
\min_{q} \left( \text{WEIGHT of violations of fundamental form constraints} \right)
$$
Rating Differential Example

$D = \begin{pmatrix}
Duke & Miami & UNC & UVA & VT \\
Duke & 0 & 0 & 0 & 0 & 0 \\
Miami & 45 & 0 & 18 & 8 & 20 \\
UNC & 3 & 0 & 0 & 2 & 0 \\
UVA & 31 & 0 & 0 & 0 & 0 \\
VT & 45 & 0 & 27 & 38 & 0 \\
\end{pmatrix}$

$D(q, q) = \begin{pmatrix}
Miami & VT & UNC & UVA & Duke \\
Miami & 0 & 20 & 18 & 8 & 45 \\
VT & 0 & 0 & 27 & 38 & 45 \\
UNC & 0 & 0 & 0 & 2 & 3 \\
UVA & 0 & 0 & 0 & 0 & 31 \\
Duke & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}$
### Rating Differential Example

#### Rating Matrix \( D \):

\[
D = \begin{pmatrix}
Duke & Miami & UNC & UVA & VT \\
Duke & 0 & 0 & 0 & 0 & 0 \\
Miami & 45 & 0 & 18 & 8 & 20 \\
UNC & 3 & 0 & 0 & 2 & 0 \\
UVA & 31 & 0 & 0 & 0 & 0 \\
VT & 45 & 0 & 27 & 38 & 0
\end{pmatrix}
\]

#### Rating Vector \( D(q, q) \):

\[
D(q, q) = \begin{pmatrix}
Miami & VT & UNC & UVA & Duke \\
Miami & 0 & 20 & 18 & 8 & 45 \\
VT & 0 & 0 & 27 & 38 & 45 \\
UNC & 0 & 0 & 0 & 2 & 3 \\
UVA & 0 & 0 & 0 & 0 & 31 \\
Duke & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

#### Ranking Methods

- **mHITS method**
  - Duke: 5th
  - Miami: 2nd
  - UNC: 4th
  - UVA: 3rd
  - VT: 1st

- **Rank differential method**
  - Duke: 5th
  - Miami: 2nd
  - UNC: 3rd
  - UVA: 4th
  - VT: 1st

- **Rating differential method**
  - Duke: 5th
  - Miami: 1st
  - UNC: 3rd
  - UVA: 4th
  - VT: 2nd
Solving the Optimization Problem

\[
\min_q \quad (# \text{ violations of fundamental form constraints})
\]

- Huge solution space: \( \exists n! \) permutations \( q \) (related to TSP; NP-hard)

**Evolutionary Algorithm**

- Initialize population with \( k=10 \) solutions \( \{x_1, x_2, \ldots, x_k\} \)
- Until convergence
  - Compute fitness \( \|D(x_i, x_i) - \hat{R}\| \) for each \( x_i \)
  - Create new population by
    - Copy 3 fittest \( x_i \) into next generation
    - Pair 6 fittest \( x_i \) and mate with rank aggregation
    - Mutate next 3 \( x_i \) with flip, invert, reversal operators
    - Insert 1 immigrant \( x_i \) using random permutation

End

Guaranteed to converge to global min (Fogel, Michalewicz) but slow
Solving the Optimization Problem

\[
\min_q \quad (\# \text{ violations of fundamental form constraints})
\]

- Huge solution space: \( \exists n! \) permutations \( q \) (related to TSP; NP-hard)

**Evolutionary Algorithm**

---

initialize population with \( k=10 \) solutions \( \{x_1, x_2, \ldots, x_k\} \)
until convergence

compute fitness \( \# \text{ violations} \) for each \( x_i \)
create new population by

- copy 3 fittest \( x_i \) into next generation
- pair 6 fittest \( x_i \) and mate with rank aggregation
- mutate next 3 \( x_i \) with flip, invert, reversal operators
- insert 1 immigrant \( x_i \) using random permutation

end

---

guaranteed to converge to global min (Fogel, Michelawicz) but slow
A New Algorithm for Ranking Sports Teams Using Evolutionary Optimization

Kathryn Pedings (kathryn.pedings@gmail.com) and Dr. Amy Langville
College of Charleston Department of Mathematics

Abstract
Evolutionary optimization is an algorithm using Darwinian ideas of mating, mutating, fitness, and survival of the fittest. Its use has been limited to intractable problems, but it was our belief that the algorithm could be modified to be successful with tractable problems such as sports ranking.

1. Never before used on sports ranking.
2. We have shown that with some datasets our evolutionary algorithm is better than other well-known algorithms.
3. Sophisticated changes made to speed up the algorithm.

Hillside Form and Point-Differential Matrices

Acknowledgements
- Emmie Douglas for her input and scholarly advice.
- Dr. Amy Langville for her original Evolutionary Optimization code, datasets, and continued guidance throughout the research process.

Results
The dataset used is the basketball data for the Southern Conference for the 2007-2008 season.

<table>
<thead>
<tr>
<th>Massey Ranking</th>
<th>Evolutionary Optimization</th>
<th>Colley Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Viol = 269</td>
<td># of Viol = 265</td>
<td># of Viol = 278</td>
</tr>
<tr>
<td>Davidson</td>
<td>Davidson</td>
<td>Davidson</td>
</tr>
<tr>
<td>UNC - G</td>
<td>UNC - G</td>
<td>App St.</td>
</tr>
<tr>
<td>GA So.</td>
<td>GA So.</td>
<td>Chatt.</td>
</tr>
<tr>
<td>App St.</td>
<td>App St.</td>
<td>GA So.</td>
</tr>
<tr>
<td>Chatt.</td>
<td>Chatt.</td>
<td>UNC - G</td>
</tr>
<tr>
<td>CoF C</td>
<td>CoF C</td>
<td>Elon</td>
</tr>
<tr>
<td>Elon</td>
<td>W. Car.</td>
<td>CoF C</td>
</tr>
<tr>
<td>Wofford</td>
<td>Furman</td>
<td>Wofford</td>
</tr>
<tr>
<td>W. Car.</td>
<td>Furman</td>
<td>Furman</td>
</tr>
<tr>
<td>Furman</td>
<td>Citadel</td>
<td>W. Car.</td>
</tr>
<tr>
<td>Citadel</td>
<td></td>
<td>Citadel</td>
</tr>
</tbody>
</table>

Future Work
- Use to rank other items such as books, movies, graduate school student applicants, etc.
- Enter a bracket to ESPN for March Madness.

Work Cited
Aggregation
Aggregation
Methods of Comparison
Methods of Comparison

Many methods, which is best?

Qualitative

- $\mathcal{R}^1$ plots
- bipartite line graphs

Quantitative

- distance between two ranked lists
  * Kendall’s $\tau$
  * Spearman’s footrule
- distance to aggregated list
- # of hillside violations
Methods of Comparison

$\mathbb{R}^1$ plots

2008 SoCon results (Neil Goodson and Colin Stephenson)
### Methods of Comparison

**bipartite line graphs**

#### 2005 NCAA basketball

<table>
<thead>
<tr>
<th>Markov</th>
<th>RPI</th>
<th>Massey</th>
</tr>
</thead>
<tbody>
<tr>
<td>Villanova</td>
<td>Duke</td>
<td>Texas</td>
</tr>
<tr>
<td>Connecticut</td>
<td>Villanova</td>
<td>Connecticut</td>
</tr>
<tr>
<td>Syracuse</td>
<td>Connecticut</td>
<td>Duke</td>
</tr>
<tr>
<td>Texas</td>
<td>Memphis</td>
<td>Kansas</td>
</tr>
<tr>
<td>Pittsburgh</td>
<td>OhioSt</td>
<td>NorthCarolina</td>
</tr>
<tr>
<td>WestVirginia</td>
<td>Tennessee</td>
<td>Villanova</td>
</tr>
<tr>
<td>Kansas</td>
<td>Iowa</td>
<td>Florida</td>
</tr>
<tr>
<td>Iowa</td>
<td>Texas</td>
<td>UCLA</td>
</tr>
<tr>
<td>Georgetown</td>
<td>UCLA</td>
<td>Memphis</td>
</tr>
<tr>
<td>Marquette</td>
<td>Gonzaga</td>
<td>Washington</td>
</tr>
<tr>
<td>Duke</td>
<td>Pittsburgh</td>
<td>OhioSt</td>
</tr>
<tr>
<td>Illinois</td>
<td>NorthCarolina</td>
<td>Illinois</td>
</tr>
<tr>
<td>MichiganSt</td>
<td>LSU</td>
<td>LSU</td>
</tr>
<tr>
<td>OhioSt</td>
<td>Illinois</td>
<td>Pittsburgh</td>
</tr>
<tr>
<td>SetonHall</td>
<td>Florida</td>
<td>Arkansas</td>
</tr>
<tr>
<td>NorthCarolina</td>
<td>Oklahoma</td>
<td>Iowa</td>
</tr>
<tr>
<td>TexasA&amp;M</td>
<td>Syracuse</td>
<td>Tennessee</td>
</tr>
<tr>
<td>Oklahoma</td>
<td>MichiganSt</td>
<td>Georgetown</td>
</tr>
<tr>
<td>Wisconsin</td>
<td>Nevada</td>
<td>MichiganSt</td>
</tr>
<tr>
<td>Cincinnati</td>
<td>Kansas</td>
<td></td>
</tr>
</tbody>
</table>
Methods of Comparison

distance between two ranked lists

Kendall’s $\tau$ on full lists of length $n$

$$-1 \leq \tau = \frac{n_c - n_d}{\binom{n}{2}} \leq 1$$

$n_c = \#$ concordant pairs

$n_d = \#$ discordant pairs

$\tau = 1$, lists in complete agreement

$\tau = -1$, one list is reverse of the other
Methods of Comparison

distance between two ranked lists

Kendall’s $\tau$ on full lists of length $n$

$$-1 \leq \tau = \frac{n_c - n_d}{\binom{n}{2}} \leq 1$$

$n_c = \# \text{ concordant pairs}$

$n_d = \# \text{ discordant pairs}$

$\tau = 1$, lists in complete agreement

$\tau = -1$, one list is reverse of the other

What about partial lists, like top-$k$ lists?
Methods of Comparison

bipartite line graphs

2005 NCAA basketball
Kendall’s Tau on partial lists

\[ \tau = \frac{n_c - n_d - n_u}{\binom{n}{2} - n_u} \]

where:
- \( n_c = \# \) concordant pairs
- \( n_d = \# \) discordant pairs
- \( n_u = \# \) unlabeled pairs

Bounds

\[ \frac{-\binom{n}{2}}{\binom{n}{2} - n_u} \leq \tau \leq \frac{\binom{n}{2}}{\binom{n}{2} - n_u} \]
Kendall’s Tau on partial lists

\[ \tau = 0.67 \]
\[ \tau = 0.06 \]
Methods of Comparison

distance between two ranked lists

Spearman’s footrule on full lists \( l \) and \( \tilde{l} \) of length \( n \)

\[
0 \leq \phi = \sum_{i=1}^{n} |l(i) - \tilde{l}(i)|
\]

\( \phi = \|l - \tilde{l}\|_1 \)
Methods of Comparison

distance between two ranked lists

Spearman’s footrule on full lists \( l \) and \( \tilde{l} \) of length \( n \)

\[
0 \leq \phi = \sum_{i=1}^{n} |l(i) - \tilde{l}(i)| \quad \phi = \|l - \tilde{l}\|_1
\]

BUT disagreements in lists are given equal weight
Methods of Comparison

distance between two ranked lists

Weighted footrule on full lists \( l \) and \( \tilde{l} \) of length \( n \)

\[
\phi = \frac{\sum_{i=1}^{n} |l(i) - \tilde{l}(i)|}{\min\{l(i), \tilde{l}(i)\}}
\]
Methods of Comparison

distance between two ranked lists

Weighted footrule on full lists \( l \) and \( \tilde{l} \) of length \( n \)

\[
\phi = \frac{\sum_{i=1}^{n} |l(i) - \tilde{l}(i)|}{\min\{l(i), \tilde{l}(i)\}}
\]

What about partial lists, like top-\( k \) lists?
Weighted Footrule on partial lists

\[ \phi = 0.05 \quad \phi = 0.27 \]
Weighted Footrule on partial lists

**Weighted Footrule $\phi$ Measure for Comparing Partial Lists of Length $k$**

The weighted footrule measure $\phi$ between two partial lists $l$ and $\tilde{l}$, both of length $k$, is built from individual $\phi_i$ values and normalized so that $0 \leq \phi \leq 1$. 

$$\phi = \frac{\sum_{i=1}^{k} \phi_i}{\phi(l, l^c)},$$

where 

$$\phi(l, l^c) = -2k + 2x \sum_{i=1}^{k} 1/i.$$ 

Each item $i$ belongs to one of two following classes, and thus its contribution $\phi_i$ to $\phi$ is calculated accordingly.

- For item $i \in l \cap \tilde{l}$ (i.e., an item appearing in both lists $l$ and $\tilde{l}$),
  
  $$\phi_i = \frac{|l(i) - \tilde{l}(i)|}{\min(l(i), \tilde{l}(i))}.$$ 

- For item $i \in (l \cup \tilde{l})/(l \cap \tilde{l})$ (i.e., an item appearing in only one list, which without loss of generality, we assume is $l$),
  
  $$\phi_i = \frac{|l(i) - x|}{\min(l(i), x)},$$

where $x$ is defined as

$$x = \frac{k - 4\lfloor k/2 \rfloor + 2(k + 1) \sum_{i=1}^{\lfloor k/2 \rfloor} 1/i}{\sum_{i=1}^{k} 1/i}.$$
Weighted Footrule vs. Kendall Tau

Kendall Tau: \[ \tau = 0.95 \quad \tau = 0.95 \]

weighted footrule: \[ \phi = 0.34 \quad \phi = 1.53 \]
But which list is best?

Methods of Comparison

distance to aggregated list

\[
\phi = 0.05 \quad \phi = 0.27
\]

mHITS   Aggregate   Markov
Methods of Comparison

# of hillside violations

Results

The dataset used is the basketball data for the Southern Conference for the 2007-2008 season.
Aggregation

Rank Aggregation
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>5</td>
<td>5th</td>
</tr>
<tr>
<td>Miami</td>
<td>2nd</td>
<td>1st</td>
<td>1st</td>
<td>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>
Simulated Data

3 ranked lists

<table>
<thead>
<tr>
<th>mHITS</th>
<th>Massey</th>
<th>Colley</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>VT</td>
<td>Miami</td>
</tr>
<tr>
<td>2nd</td>
<td>Miami</td>
<td>VT</td>
</tr>
<tr>
<td>3rd</td>
<td>UVA</td>
<td>UVA</td>
</tr>
<tr>
<td>4th</td>
<td>UNC</td>
<td>UNC</td>
</tr>
<tr>
<td>5th</td>
<td>Duke</td>
<td>Duke</td>
</tr>
</tbody>
</table>

mHITS

- 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>Team</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>6-21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-5</td>
<td>4-2</td>
<td>12-10</td>
<td></td>
</tr>
<tr>
<td>Team 2</td>
<td>3-11</td>
<td>6-8</td>
<td>13-19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1-3</td>
<td>15-21</td>
<td>9-12</td>
<td></td>
</tr>
<tr>
<td>Team n</td>
<td>11-10</td>
<td>8-4</td>
<td>5-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11-9</td>
<td>12-8</td>
<td>11-8</td>
<td></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 $j$
  - $w_{ij} = \text{sum of rank differences of lists having } i \text{ below } j$

- 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

rating differential matrices $\mathbf{R} \geq 0$

- differing scales $\rightarrow$ NORMALIZE
Rating Aggregation

rating differential matrices

\[
\tilde{R}_{\text{Massey}} = \begin{pmatrix}
0 & 0 & 0.09 & 0.0008 \\
0.1792 & 0 & 0.1092 & 0 \\
0.1367 & 0 & 0 & 0.0008 \\
0.0892 & 0.0192 & 0 & 0 \\
0.1783 & 0.1083 & 0.0892 & 0 \\
\end{pmatrix}
\]

\[
\tilde{R}_{\text{Colley}} = \begin{pmatrix}
0 & 0 & 0.1483 & 0.0483 \\
0 & 0.2 & 0.1 & 0 \\
0.1 & 0 & 0 & 0.0483 \\
0.0517 & 0 & 0 & 0 \\
0.1517 & 0.0517 & 0.1 & 0 \\
\end{pmatrix}
\]

\[
\tilde{R}_{mHITS} = \begin{pmatrix}
0 & 0 & 0.0773 & 0 \\
0.1237 & 0 & 0.1082 & 0.0464 \\
0.0155 & 0 & 0.0309 & 0 \\
0.0464 & 0.0309 & 0 & 0 \\
0.1959 & 0.0722 & 0.1804 & 0 \\
\end{pmatrix}
\]
Rating Aggregation

rating differential matrices

- combine into one matrix → AVERAGE
Rating Aggregation

rating differential matrices

- combine into one matrix → **AVERAGE**
Rating Aggregation

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

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 \\
0.1676 & 0 & 0.1058 & 0.1052 & 0.0164 \\
0.0841 & 0 & 0 & 0.0161 & 0 \\
0.0624 & 0 & 0.0167 & 0 & 0 \\
0.1753 & 0.0241 & 0.1135 & 0.1129 & 0
\end{pmatrix}
\]

- 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 $\mathbf{R}_{\text{average}} \geq 0$

- 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 $\mathbf{R}_{\text{average}}$

<table>
<thead>
<tr>
<th>Team</th>
<th>Method 1 Method 2</th>
<th>Method 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r = o/d$ Markov $r$</td>
<td>Perron $r$</td>
</tr>
<tr>
<td>Duke</td>
<td>0 $5^{th}$ .20 $5^{th}$ .27 $5^{th}$</td>
<td></td>
</tr>
<tr>
<td>Miami</td>
<td>16.4 $2^{nd}$ .465 $2^{nd}$ .58 $2^{nd}$</td>
<td></td>
</tr>
<tr>
<td>UNC</td>
<td>.4 $3^{rd}$ .025 $3^{rd}$ .34 $3^{rd}$</td>
<td></td>
</tr>
<tr>
<td>UVA</td>
<td>3 $4^{th}$ .024 $4^{th}$ .33 $4^{th}$</td>
<td></td>
</tr>
<tr>
<td>VT</td>
<td>26.0 $1^{st}$ .466 $1^{st}$ .61 $1^{st}$</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

• several methods for rating and ranking items begin by building nonnegative matrices
  * Markov: stationary vector of \( \mathbf{V} \geq 0 \)
  * mHITS: Sinkhorn-Knopp on \( \mathbf{P} \geq 0 \)
  * Rank Differential: reordering of \( \mathbf{D} \geq 0 \)
  * Rating Differential: reordering of \( \mathbf{D} \geq 0 \)

• nonnegative matrix theory many times insures existence, uniqueness, convergence.

• sometimes nonnegativity is forced to guarantee these properties

• rank and rating aggregation also build nonnegative matrices: \( \mathbf{W} \geq 0, \mathbf{R} \geq 0 \)