Fast Effective Clustering for Graphs and Document Collections

William W. Cohen

Machine Learning Dept. and Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Joint work with: Frank Lin
Outline

• Background on spectral clustering
• “Power Iteration Clustering”
  - Motivation
  - Experimental results
• Analysis: PIC vs spectral methods
• PIC for sparse bipartite graphs
  - Motivation & Method
  - Experimental Results
Spectral Clustering: Graph = Matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Spectral Clustering: Graph = Matrix
Transitively Closed Components = “Blocks”

Of course we can’t see the “blocks” unless the nodes are sorted by cluster…
Spectral Clustering: Graph = Matrix  Vector = Node → Weight

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>_</td>
<td>1</td>
<td>1</td>
<td>_</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>_</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>_</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>_</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>_</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>_</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>_</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>_</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>1</td>
<td>1</td>
<td>_</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>_</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

M

V

A

A 3
B 2
C 3
D
E
F
G
H
I
J

M

M

Diagram
Spectral Clustering: Graph = Matrix
\[ M \times v_1 = v_2 \] "propogates weights from neighbors"
**Spectral Clustering: Graph = Matrix**

\[ W \ast v_1 = v_2 \] “propogates weights from neighbors”

*W: normalized so columns sum to 1*

\[
W \ast v_1 = v_2
\]
Spectral Clustering: Graph = Matrix

$Wv_1 = v_2$ “propogates weights from neighbors”

$W \cdot v = \lambda v : v$ is an eigenvector with eigenvalue $\lambda$

Q: How do I pick $v$ to be an eigenvector for a block-stochastic matrix?
Spectral Clustering: Graph = Matrix

$W^*v_1 = v_2$ “propogates weights from neighbors”

$W \cdot v = \lambda v : v$ is an eigenvector with eigenvalue $\lambda$

How do I pick $v$ to be an eigenvector for a block-stochastic matrix?
Spectral Clustering: Graph = Matrix
$W^*v_1 = v_2$ “propogates weights from neighbors”

$W \cdot v = \lambda v : v$ is an eigenvector with eigenvalue $\lambda$

[Shi & Meila, 2002]
Spectral Clustering: Graph = Matrix

$W \cdot v = \lambda v : v$ is an eigenvector with eigenvalue $\lambda$

\[ W \cdot v = \lambda v \]

[Shi & Meila, 2002]
Spectral Clustering: Graph = Matrix
\[ W^* v_1 = v_2 \] "propogates weights from neighbors"

\[ W \cdot v = \lambda v : v \text{ is an eigenvector with eigenvalue } \lambda \]

If \( W \) is connected but roughly block diagonal with \( k \) blocks then

- the top eigenvector is a constant vector
- the next \( k \) eigenvectors are roughly piecewise constant with "pieces" corresponding to blocks
Spectral Clustering: Graph = Matrix
$W \cdot v = \lambda v : v$ is an eigenvector with eigenvalue $\lambda$

If $W$ is connected but roughly block diagonal with $k$ blocks then
• the “top” eigenvector is a constant vector
• the next $k$ eigenvectors are roughly piecewise constant with “pieces” corresponding to blocks

Spectral clustering:
• Find the top $k+1$ eigenvectors $v_1, \ldots, v_{k+1}$
• Discard the “top” one
• Replace every node $a$ with $k$-dimensional vector $x_a = < v_2(a), \ldots, v_{k+1}(a) >$
• Cluster with $k$-means
Spectral Clustering: Pros and Cons

• Elegant, and well-founded mathematically
• Tends to avoid local minima
  - Optimal solution to relaxed version of mincut problem (Normalized cut, aka NCut)
• Works quite well when relations are approximately transitive (like similarity, social connections)
• Expensive for very large datasets
  - Computing eigenvectors is the bottleneck
  - Approximate eigenvector computation not always useful
• Noisy datasets sometimes cause problems
  - Picking number of eigenvectors and $k$ is tricky
  - “Informative” eigenvectors need not be in top few
  - Performance can drop suddenly from good to terrible
Experimental results: best-case assignment of class labels to clusters

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>NCut Accuracy</th>
<th>NCut Macro-F1</th>
<th>NJW Accuracy</th>
<th>NJW Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3</td>
<td>0.673</td>
<td>0.570</td>
<td>0.807</td>
<td>0.806</td>
</tr>
<tr>
<td>PenDigits01</td>
<td>2</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>PenDigits17</td>
<td>2</td>
<td>0.755</td>
<td>0.753</td>
<td>0.755</td>
<td>0.754</td>
</tr>
<tr>
<td>UBMCMBlog</td>
<td>2</td>
<td>0.953</td>
<td>0.953</td>
<td>0.953</td>
<td>0.953</td>
</tr>
<tr>
<td>AGBlog</td>
<td>2</td>
<td>0.520</td>
<td>0.342</td>
<td>0.520</td>
<td>0.342</td>
</tr>
<tr>
<td>20ngA</td>
<td>2</td>
<td>0.955</td>
<td>0.955</td>
<td>0.955</td>
<td>0.955</td>
</tr>
<tr>
<td>20ngB</td>
<td>2</td>
<td>0.505</td>
<td>0.344</td>
<td>0.550</td>
<td>0.436</td>
</tr>
<tr>
<td>20ngC</td>
<td>3</td>
<td>0.613</td>
<td>0.621</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20ngD</td>
<td>4</td>
<td>0.469</td>
<td>0.432</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>-</td>
<td>0.716</td>
<td>0.663</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Adamic & Glance
“Divided They Blog:…” 2004
**Spectral Clustering: Graph = Matrix**  
\[ M \ast v_1 = v_2 \]  
“propogates weights from neighbors”

\[
\begin{array}{cccccccccc}
A & B & C & D & E & F & G & H & I & J \\
\_ & 1 & 1 & 1 & & & & & & \\
B & 1 & \_ & 1 & & & & & & \\
C & 1 & 1 & \_ & & & & & & \\
D & & 1 & 1 & & & & & & \\
E & & 1 & \_ & 1 & & & & & \\
F & & 1 & 1 & \_ & & & & & \\
G & & & 1 & 1 & & & & & \\
H & & & \_ & 1 & 1 & & & & \\
I & 1 & 1 & \_ & 1 & & & & & \\
J & 1 & 1 & 1 & \_ & & & & & \\
\end{array}
\]

\[
\begin{array}{cccc}
A & 3 & & \\
B & 2 & & \\
C & 3 & & \\
D & & 1 & 1 & \\
E & & 1 & \_ & 1 \\
F & & 1 & 1 & \_ \\
G & & 1 & 1 & \_ \\
H & & \_ & 1 & 1 \\
I & 1 & 1 & \_ & 1 \\
J & 1 & 1 & 1 & \_ \\
\end{array}
\]
Repeated averaging with neighbors as a clustering method

- Pick a vector $v^0$ (maybe at random)
- Compute $v^1 = Wv^0$
  - i.e., replace $v^0[x]$ with weighted average of $v^0[y]$ for the neighbors $y$ of $x$
- Plot $v^1[x]$ for each $x$
- Repeat for $v^2, v^3, ...$

- Variants widely used for semi-supervised learning
  - clamping of labels for nodes with known labels
- Without clamping, will converge to constant $v^*$
- What are the dynamics of this process?
Repeated averaging with neighbors on a sample problem...

- Create a graph, connecting all points in the 2-D initial space to all other points
  - Weighted by distance
- Run power iteration for 10 steps
- Plot node id \( x \) vs \( v^{10}(x) \)
  - Nodes are ordered by actual cluster number
Repeated averaging with neighbors on a sample problem...

(a) 3 Circles PIC result
(b) Embedding at $t = 10$
(c) Embedding at $t = 50$
(d) Embedding at $t = 100$
Repeated averaging with neighbors on a sample problem...

(a) 3Circles PTC result

(b) Embedding at $t = 10$

(c) Embedding at $t = 50$

(d) Embedding at $t = 100$

(e) Embedding at $t = 200$

(f) Embedding at $t = 400$
Repeated averaging with neighbors on a sample problem...

(a) 3Circles PIC result
(b) Embedding at $t = 10$
(c) Embedding at $t = 50$
(d) Embedding at $t = 100$

(e) Embedding at $t = 200$
(f) Embedding at $t = 400$
(g) Embedding at $t = 600$
(h) Embedding at $t = 1000$

very small
PIC: Power Iteration Clustering
run power iteration (repeated averaging w/ neighbors) with early stopping

1. Pick an initial vector \( \mathbf{v}^0 \).
2. Set \( \mathbf{v}^{t+1} \leftarrow \frac{W \mathbf{v}^t}{\|W \mathbf{v}^t\|_1} \) and \( \delta^{t+1} \leftarrow |\mathbf{v}^{t+1} - \mathbf{v}^t| \).
3. Increment \( t \) and repeat above step until \( |\delta^t - \delta^{t-1}| \approx 0 \).
4. Use \( k \)-means to cluster points on \( \mathbf{v}^t \) and return clusters \( C_1, C_2, ..., C_k \).

- \( \mathbf{v}^0 \): random start, or “degree matrix” \( D \), or ...
- Easy to implement and efficient
- Very easily parallelized
- Experimentally, often better than traditional spectral methods
- Surprising since the embedded space is 1-dimensional!
Experiments

• “Network” problems: natural graph structure
  - PolBooks: 105 political books, 3 classes, linked by copurchaser
  - UMBCBlog: 404 political blogs, 2 classes, blogroll links
  - AGBlog: 1222 political blogs, 2 classes, blogroll links

• “Manifold” problems: cosine distance between classification instances
  - Iris: 150 flowers, 3 classes
  - PenDigits01,17: 200 handwritten digits, 2 classes (0-1 or 1-7)
  - 20ngA: 200 docs, misc.forsale vs soc.religion.christian
  - 20ngB: 400 docs, misc.forsale vs soc.religion.christian
  - 20ngC: 20ngB + 200 docs from talk.politics.guns
  - 20ngD: 20ngC + 200 docs from rec.sport.baseball
**Experimental results:**

best-case assignment of class labels to clusters

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>NCut Accuracy</th>
<th>Macro-F1</th>
<th>NJW Accuracy</th>
<th>Macro-F1</th>
<th>PIC Accuracy</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3</td>
<td>0.673</td>
<td>0.570</td>
<td>0.807</td>
<td>0.806</td>
<td>0.980</td>
<td>0.980</td>
</tr>
<tr>
<td>PenDigits01</td>
<td>2</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>PenDigits17</td>
<td>2</td>
<td>0.755</td>
<td>0.753</td>
<td>0.755</td>
<td>0.754</td>
<td>0.755</td>
<td>0.753</td>
</tr>
<tr>
<td>UBMCCBlog</td>
<td>2</td>
<td>0.953</td>
<td>0.953</td>
<td>0.953</td>
<td>0.953</td>
<td>0.948</td>
<td>0.948</td>
</tr>
<tr>
<td>AGBBlog</td>
<td>2</td>
<td>0.520</td>
<td>0.342</td>
<td>0.520</td>
<td>0.342</td>
<td>0.957</td>
<td>0.957</td>
</tr>
<tr>
<td>20ngA</td>
<td>2</td>
<td>0.955</td>
<td>0.955</td>
<td>0.955</td>
<td>0.955</td>
<td>0.960</td>
<td>0.960</td>
</tr>
<tr>
<td>20ngB</td>
<td>2</td>
<td>0.505</td>
<td>0.344</td>
<td>0.550</td>
<td>0.436</td>
<td>0.905</td>
<td>0.904</td>
</tr>
<tr>
<td>20ngC</td>
<td>3</td>
<td>0.613</td>
<td>0.621</td>
<td>0.635</td>
<td>0.639</td>
<td>0.737</td>
<td>0.730</td>
</tr>
<tr>
<td>20ngD</td>
<td>4</td>
<td>0.469</td>
<td>0.432</td>
<td>0.535</td>
<td>0.534</td>
<td>0.580</td>
<td>0.570</td>
</tr>
</tbody>
</table>

| Average        |   | 0.716        | 0.663    | 0.746        | 0.713    | 0.869        | 0.867    |

Table 1: Clustering performance of PIC and spectral clustering algorithms on several real datasets.
## Experiments: run time and scalability

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size</th>
<th>NCut Runtime</th>
<th>NJW Runtime</th>
<th>PIC Runtime</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>150</td>
<td>589</td>
<td>242</td>
<td>59</td>
<td>6</td>
</tr>
<tr>
<td>PenDigits01</td>
<td>200</td>
<td>965</td>
<td>326</td>
<td>56</td>
<td>6</td>
</tr>
<tr>
<td>PenDigits17</td>
<td>200</td>
<td>1197</td>
<td>528</td>
<td>62</td>
<td>6</td>
</tr>
<tr>
<td>UBMCMBlog</td>
<td>404</td>
<td>4205</td>
<td>1589</td>
<td>85</td>
<td>21</td>
</tr>
<tr>
<td>AGBlog</td>
<td>1222</td>
<td>114821</td>
<td>58145</td>
<td>211</td>
<td>34</td>
</tr>
<tr>
<td>20ngA</td>
<td>200</td>
<td>1113</td>
<td>355</td>
<td>72</td>
<td>15</td>
</tr>
<tr>
<td>20ngB</td>
<td>400</td>
<td>4085</td>
<td>1864</td>
<td>139</td>
<td>13</td>
</tr>
<tr>
<td>20ngC</td>
<td>600</td>
<td>13070</td>
<td>6383</td>
<td>190</td>
<td>13</td>
</tr>
<tr>
<td>20ngD</td>
<td>800</td>
<td>33191</td>
<td>16295</td>
<td>278</td>
<td>11</td>
</tr>
</tbody>
</table>

Time in millisecond
Analysis: why is this working?

eigenvectors $e_1, \ldots, e_n$
eigenvalues $\lambda_1, \ldots, \lambda_n$, 

$s_a = \langle e_1(a), \ldots, e_k(a) \rangle,$

$spec(a, b) \equiv \|s_a - s_b\|_2 = \sqrt{\sum_{i=2}^{k} (e_i(a) - e_i(b))^2}$

$pic^t(v^0; a, b) \equiv |v^t(a) - v^t(b)|$
Analysis: why is this working?

\[
\text{eigenvectors } e_1, \ldots, e_n \quad \text{eigenvalues } \lambda_1, \ldots, \lambda_n,
\]

\[
s_a = \langle e_1(a), \ldots, e_k(a) \rangle,
\]

\[
pic^t(v^0; a, b) \equiv |v^t(a) - v^t(b)|
\]

\[
v^t = Wv^{t-1} = W^2v^{t-2} = \ldots = W^tv^0
\]

\[
= c_1W^te_1 + c_2W^te_2 + \ldots + c_nW^te_n
\]

\[
= c_1\lambda_1^te_1 + c_2\lambda_2^te_2 + \ldots + c_n\lambda_n^te_n
\]

\[
pic^t(a, b) = \left| [e_1(a) - e_1(b)]c_1\lambda_1^t \right|
\]

\[
+ \sum_{i=2}^{k} [e_i(a) - e_i(b)]c_i\lambda_i^t
\]

\[
+ \sum_{j=k+1}^{n} [e_j(a) - e_j(b)]c_j\lambda_j^t
\]
Analysis: why is this working?

\[ \text{eigenvalues } \lambda_1, \ldots, \lambda_n, \]
\[ s_a = \langle e_1(a), \ldots, e_k(a) \rangle, \]
\[ \text{spec}(a, b) \equiv \|s_a - s_b\|_2 = \sqrt{\sum_{i=2}^{k} (e_i(a) - e_i(b))^2} \]
\[ \text{pic}^t(a, b) = \begin{bmatrix} e_1(a) - e_1(b) \end{bmatrix} c_1 \lambda_1^t \]
\[ + \sum_{i=2}^{k} [e_i(a) - e_i(b)] c_i \lambda_i^t \]
\[ + \sum_{j=k+1}^{n} [e_j(a) - e_j(b)] c_j \lambda_j^t \]

L2 distance
scaling?
differences might cancel?
“noise” terms
Analysis: why is this working?

• If
  - eigenvectors $e_2, \ldots, e_k$ are approximately piecewise constant on blocks;
  - $\lambda_2, \ldots, \lambda_k$ are “large” and $\lambda_{k+1}, \ldots$ are “small”;
    • e.g., if matrix is block-stochastic
  - the $c_i$’s for $v^0$ are bounded;
  - for any $a,b$ from distinct blocks there is at least one $e_i$ with $e_i(a)-e_i(b)$ “large”

• Then exists an $R$ so that
  - $\text{spec}(a,b)$ small $\Leftrightarrow R^*\text{pic}(a,b)$ small
Analysis: why is this working?

- Sum of differences vs sum-of-squared differences
- "soft" eigenvector selection
### Ncut with top k eigenvectors

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>Purity</th>
<th>NMI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3</td>
<td>0.6733</td>
<td>0.7235</td>
<td>0.7779</td>
</tr>
<tr>
<td>PenDigits01</td>
<td>2</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>PenDigits17</td>
<td>2</td>
<td>0.7550</td>
<td>0.2066</td>
<td>0.6301</td>
</tr>
<tr>
<td>PolBooks</td>
<td>3</td>
<td>0.8476</td>
<td>0.5745</td>
<td>0.8447</td>
</tr>
<tr>
<td>UBMCBlog</td>
<td>2</td>
<td>0.9530</td>
<td>0.7488</td>
<td>0.9104</td>
</tr>
<tr>
<td>AGBlog</td>
<td>2</td>
<td>0.5205</td>
<td>0.0060</td>
<td>0.5006</td>
</tr>
<tr>
<td>20ngA</td>
<td>2</td>
<td>0.9600</td>
<td>0.7594</td>
<td>0.9232</td>
</tr>
<tr>
<td>20ngB</td>
<td>2</td>
<td>0.5050</td>
<td>0.0096</td>
<td>0.5001</td>
</tr>
<tr>
<td>20ngC</td>
<td>3</td>
<td>0.6183</td>
<td>0.3295</td>
<td>0.6750</td>
</tr>
<tr>
<td>20ngD</td>
<td>4</td>
<td>0.4750</td>
<td>0.2385</td>
<td>0.6312</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td>0.7308</td>
<td>0.4596</td>
<td>0.7393</td>
</tr>
</tbody>
</table>

### Ncut with top 10 eigenvectors: weighted

**Table 2.** Clustering performance of eigenvalue-weighted NCut on several real datasets. For all measures a higher number means better clustering. **Bold numbers** are the highest in its row.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>Purity</th>
<th>NMI</th>
<th>RI</th>
<th>Purity weighted by $\lambda_i$</th>
<th>NMI</th>
<th>RI</th>
<th>Purity weighted by $\lambda_i^{15}$</th>
<th>NMI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>3</td>
<td>0.6667</td>
<td>0.6507</td>
<td>0.7254</td>
<td>0.9800</td>
<td>0.9306</td>
<td>0.9741</td>
<td>0.9800</td>
<td>0.9306</td>
<td>0.9741</td>
</tr>
<tr>
<td>PenDigits01</td>
<td>2</td>
<td>0.7000</td>
<td>0.2746</td>
<td>0.5800</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>PenDigits17</td>
<td>2</td>
<td>0.7000</td>
<td>0.1810</td>
<td>0.5800</td>
<td>0.7550</td>
<td>0.2066</td>
<td>0.6301</td>
<td>0.7550</td>
<td>0.2066</td>
<td>0.6301</td>
</tr>
<tr>
<td>PolBooks</td>
<td>3</td>
<td>0.4857</td>
<td>0.1040</td>
<td>0.4413</td>
<td>0.8476</td>
<td>0.5861</td>
<td>0.8514</td>
<td>0.8476</td>
<td>0.5861</td>
<td>0.8514</td>
</tr>
<tr>
<td>UBMCBlog</td>
<td>2</td>
<td>0.9505</td>
<td>0.7400</td>
<td>0.9059</td>
<td>0.9505</td>
<td>0.7400</td>
<td>0.9059</td>
<td>0.9505</td>
<td>0.7400</td>
<td>0.9059</td>
</tr>
<tr>
<td>AGBlog</td>
<td>2</td>
<td>0.9493</td>
<td>0.7143</td>
<td>0.9037</td>
<td>0.9509</td>
<td>0.7223</td>
<td>0.9066</td>
<td>0.9509</td>
<td>0.7223</td>
<td>0.9066</td>
</tr>
<tr>
<td>20ngA</td>
<td>2</td>
<td>0.5600</td>
<td>0.0685</td>
<td>0.5072</td>
<td>0.9600</td>
<td>0.7594</td>
<td>0.9232</td>
<td>0.9600</td>
<td>0.7594</td>
<td>0.9232</td>
</tr>
<tr>
<td>20ngB</td>
<td>2</td>
<td>0.7125</td>
<td>0.2734</td>
<td>0.5903</td>
<td>0.9450</td>
<td>0.7042</td>
<td>0.8961</td>
<td>0.9450</td>
<td>0.7042</td>
<td>0.8961</td>
</tr>
<tr>
<td>20ngC</td>
<td>3</td>
<td>0.6867</td>
<td>0.3866</td>
<td>0.6546</td>
<td>0.6617</td>
<td>0.3772</td>
<td>0.7025</td>
<td>0.6617</td>
<td>0.3772</td>
<td>0.7025</td>
</tr>
<tr>
<td>20ngD</td>
<td>4</td>
<td>0.4763</td>
<td>0.2365</td>
<td>0.6368</td>
<td>0.4875</td>
<td>0.2555</td>
<td>0.6425</td>
<td>0.4875</td>
<td>0.2555</td>
<td>0.6425</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td>0.6888</td>
<td>0.3630</td>
<td>0.6525</td>
<td>0.8538</td>
<td>0.6282</td>
<td>0.8432</td>
<td>0.8538</td>
<td>0.6282</td>
<td>0.8432</td>
</tr>
<tr>
<td>Dataset</td>
<td>k</td>
<td>Purity</td>
<td>NMI</td>
<td>RI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>----</td>
<td>--------</td>
<td>------</td>
<td>-----</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iris</td>
<td>3</td>
<td>0.6733</td>
<td>0.7235</td>
<td>0.7779</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PenDigits01</td>
<td>2</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PenDigits17</td>
<td>2</td>
<td>0.7550</td>
<td>0.2066</td>
<td>0.6301</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PolBooks</td>
<td>3</td>
<td>0.8476</td>
<td>0.5745</td>
<td>0.8447</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UBMCMBlog</td>
<td>2</td>
<td>0.9530</td>
<td>0.7488</td>
<td>0.9104</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGBlog</td>
<td>2</td>
<td>0.5205</td>
<td>0.0060</td>
<td>0.5006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20ngA</td>
<td>2</td>
<td>0.9600</td>
<td>0.7594</td>
<td>0.9232</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20ngB</td>
<td>2</td>
<td>0.5050</td>
<td>0.0096</td>
<td>0.5001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20ngC</td>
<td>3</td>
<td>0.6183</td>
<td>0.3295</td>
<td>0.6750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20ngD</td>
<td>4</td>
<td>0.4750</td>
<td>0.2385</td>
<td>0.6312</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>0.7308</td>
<td>0.4596</td>
<td>0.7393</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>k</th>
<th>Purity</th>
<th>NMI</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purity</td>
<td></td>
<td>0.9800</td>
<td>0.9306</td>
<td>0.9741</td>
</tr>
<tr>
<td>NMI</td>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>RI</td>
<td></td>
<td>0.7550</td>
<td>0.2066</td>
<td>0.6301</td>
</tr>
<tr>
<td>e_i weighted by $\lambda_i$</td>
<td></td>
<td>0.8476</td>
<td>0.5861</td>
<td>0.8514</td>
</tr>
<tr>
<td>e_i weighted by $\lambda_i^{15}$</td>
<td></td>
<td>0.9530</td>
<td>0.7488</td>
<td>0.9104</td>
</tr>
</tbody>
</table>

Table 2. Clustering performance of eigenvalue-weighted NCut on several real datasets. For all measures a higher number means better clustering. Bold numbers are the highest in its row.
Summary of results so far

- Both PIC and Ncut embed each graph node in a space where distance is meaningful.
- Distances in “PIC space” and Eigenspace are closely related.
  - At least for many graphs suited to spectral clustering.
- PIC does “soft” selection of eigenvectors.
  - Strong eigenvalues give high weights.
- PIC gives comparable-quality clusters.
  - But is much faster.
Outline

• Background on spectral clustering
• “Power Iteration Clustering”
  - Motivation
  - Experimental results
• Analysis: PIC vs spectral methods
• PIC for sparse bipartite graphs
  - “Lazy” Distance Computation
  - “Lazy” Normalization
  - Experimental Results
Motivation: Experimental Datasets are...

• “Network” problems: natural graph structure
  - PolBooks: 105 political books, 3 classes, linked by copurchaser
  - UMBCBlog: 404 political blogs, 2 classes, blogroll links
  - AGBlog: 1222 political blogs, 2 classes, blogroll links
  - Also: Zachary’s karate club, citation networks, ...

• “Manifold” problems: cosine distance between all pairs of classification instances
  - Iris: 150 flowers, 3 classes
  - PenDigits01,17: 200 handwritten digits, 2 classes (0-1 or 1-7)
  - 20ngA: 200 docs, misc.forsale vs soc.religion.christian
  - 20ngB: 400 docs, misc.forsale vs soc.religion.christian
  - ...

  Gets expensive fast
Lazy computation of distances and normalizers

• Recall PIC’s update is
  \[ v^t = W * v^{t-1} = D^{-1} A * v^{t-1} \]
  \[ \text{1 is a column vector of 1's} \]
  \[ \text{where \( D \) is the [diagonal] degree matrix: } D = A * 1 \]

• My favorite distance metric for text is length-normalized TFIDF:
  - Def’n: \( A(i,j) = \langle v_i, v_j \rangle / ||v_i|| * ||v_j|| \)
  - Let \( N(i,i) = ||v_i|| \) ... and \( N(i,j) = 0 \) for \( i \neq j \)
  - Let \( F(i,k) = \text{TFIDF weight of word } w_k \text{ in document } v_i \)
  - Then: \( A = N^{-1} F^T F N^{-1} \)
Lazy computation of distances and normalizers

- Recall PIC's update is
  \[ v^t = W \ast v^{t-1} = D^{-1}A \ast v^{t-1} \]
  
  - where \( D \) is the [diagonal] degree matrix: \( D = A \ast 1 \)
  
  - Let \( F(i,k) = \text{TFIDF weight of word } w_k \text{ in document } v_i \)
  
  - Compute \( N(i,i) = ||v_i|| \) ... and \( N(i,j) = 0 \) for \( i \neq j \)
  
  - **Don't** compute \( A = N^{-1}F^T F N^{-1} \)
  
  - Let \( D(i,i) = N^{-1}F^T F N^{-1} \ast 1 \) where \( 1 \) is an all-1's vector
    
    - Computed as \( D = N^{-1}(F^T F (N^{-1} \ast 1)) \) for efficiency
  
  - New update:
    
    - \( v^t = D^{-1}A \ast v^{t-1} = D^{-1} N^{-1}F^T F N^{-1} \ast v^{t-1} \)

- Equivalent to using TFIDF/cosine on all pairs of examples but requires only sparse matrices
Experimental results

• RCV1 text classification dataset
  - 800k + newswire stories
  - Category labels from *industry* vocabulary
  - Took single-label documents and categories with at least 500 instances
  - Result: 193,844 documents, 103 categories

• Generated 100 random category pairs
  - Each is all documents from two categories
  - Range in size and difficulty
  - Pick category 1, with $m_1$ examples
  - Pick category 2 such that $0.5m_1 < m_2 < 2m_1$
### Results

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC-Avg</th>
<th>NMI-Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>57.59</td>
<td>-</td>
</tr>
<tr>
<td>k-means</td>
<td>69.43</td>
<td>0.2629</td>
</tr>
<tr>
<td>NCUTevd</td>
<td>77.55</td>
<td>0.3962</td>
</tr>
<tr>
<td>NCUTiram</td>
<td>61.63</td>
<td>0.0943</td>
</tr>
<tr>
<td>PIC</td>
<td>76.67</td>
<td>0.3818</td>
</tr>
</tbody>
</table>

- **NCUTevd**: Ncut with exact eigenvectors
- **NCUTiram**: Implicit restarted Arnoldi method
- No stat. signif. diffs between NCUTevd and PIC
Results
Results

Size vs Runtime of PIC and NCut (log-log)
Results

• Linear run-time implies constant number of iterations

• Number of iterations to “acceleration-convergence” is hard to analyze:
  - Faster than a single complete run of power iteration to convergence
  - On our datasets
    • 10-20 iterations is typical
    • 30-35 is exceptional
(a) $R^2 = 0.0424$

(b) $R^2 = 0.0552$

(c) $R^2 = 0.0007$

(d) $R^2 = 0.0134$
Thanks to...

- NIH/NIGMS
- NSF
- Microsoft LiveLabs
- Google