Data Mining using Matrix and Graphs

Chris Ding, Tao Li, and Fei Wang

University of Texas at Arlington

CHQDing@uta.edu
A fast growing trend is to use matrix factorization, eigenvectors, tensor, etc to solve challenging problems in data mining.

In many data mining conference, about ¼ papers utilize matrix algorithms. For problem size ~ 1000, usually a few matlab codes are sufficient.
Why Matrix and Eigenvectors?

Matrix and Linear algebra

• relatively simple
  – in comparison to probabilistic, information-theoretic, graph-theoretic approaches

• well-developed branch of mathematics
  – knowledge accumulated since 1700

• many mature software available
  – developed by scientific computing community
Matrix Model Approach: Major Accomplishments

- Laplacian Spectral Clustering (Rcut, Ncut, Mcut)
  - Delicate/refined graph clustering objectives can be solved via eigenvectors of Laplacian matrix
- K-means Clustering Solution given by PCA
  - Generalized to kernel K-means clustering
- A matrix factorization solves a clustering problem
- Tensor decompositions do dimension reduction and clustering simultaneous
- A large number of eigenvector-based dimension deduction (low-dimension embedding)
  - Laplacian embedding, Isomap, LDA
- Semi-supervised Learning (Label propagation)
- Data Ranking
  - Rank webpages (PageRank in google)
- Broad application areas
  - bioinformatics, text mining (collaborative filtering), computer vision, finance, scientific data analysis/visualization
Many unsupervised learning methods are closely related in a simple way.
Spectral Clustering of Supreme Court Justices based on their voting records

Number of times (%) two Justices voted in agreement

<table>
<thead>
<tr>
<th></th>
<th>Ste</th>
<th>Bre</th>
<th>Gin</th>
<th>Sou</th>
<th>O’Co</th>
<th>Ken</th>
<th>Reh</th>
<th>Sca</th>
<th>Tho</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stevens</td>
<td>–</td>
<td>62</td>
<td>66</td>
<td>63</td>
<td>33</td>
<td>36</td>
<td>25</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>Breyer</td>
<td>62</td>
<td>–</td>
<td>72</td>
<td>71</td>
<td>55</td>
<td>47</td>
<td>43</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>Ginsberg</td>
<td>66</td>
<td>72</td>
<td>–</td>
<td>78</td>
<td>47</td>
<td>49</td>
<td>43</td>
<td>28</td>
<td>26</td>
</tr>
<tr>
<td>Souter</td>
<td>63</td>
<td>71</td>
<td>78</td>
<td>–</td>
<td>55</td>
<td>50</td>
<td>44</td>
<td>31</td>
<td>29</td>
</tr>
<tr>
<td>O’Connor</td>
<td>33</td>
<td>55</td>
<td>47</td>
<td>55</td>
<td>–</td>
<td>67</td>
<td>71</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>Kennedy</td>
<td>36</td>
<td>47</td>
<td>49</td>
<td>50</td>
<td>67</td>
<td>–</td>
<td>77</td>
<td>58</td>
<td>59</td>
</tr>
<tr>
<td>Rehnquist</td>
<td>25</td>
<td>43</td>
<td>43</td>
<td>44</td>
<td>71</td>
<td>77</td>
<td>–</td>
<td>66</td>
<td>68</td>
</tr>
<tr>
<td>Scalia</td>
<td>14</td>
<td>25</td>
<td>28</td>
<td>31</td>
<td>54</td>
<td>58</td>
<td>66</td>
<td>–</td>
<td>79</td>
</tr>
<tr>
<td>Thomas</td>
<td>15</td>
<td>24</td>
<td>26</td>
<td>29</td>
<td>54</td>
<td>59</td>
<td>68</td>
<td>79</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 1: From the voting record of Justices 1995 Term – 2004 Term, the number of times two justices voted in agreement (in percentage). (Data source: from July 2, 2005 New York Times. Originally from Legal Affairs; Harvard Law Review)

Consider this data as pairwise similarity between Justices

C. Ding, Data Mining using Matrix Algorithms
A simple example with profound consequence

Spectral Clustering of Supreme Court Justices

- U.S. Supreme court has large influence on public policy
- In 2005, Justice O’Conner retired. Later same year, Justice Rehnquist passed away.
- Justices Roberts and Alito are appointed
- Will the court (with new justices) change direction?
- Many discussions in the news media.
Spectral Clustering of Supreme Court Justices

• Three groups in the Supreme Court:
  - Left leaning group, center-right group, right leaning group.

• The center-right group has two members changed
  - Could have profound change in future court decisions?
Vector Data vs Graph Data

- Most Data come as collection of feature vectors
  - Standard machine learning/statistics deal with this data
- Graph Data
  - Network data (internet, web, biological networks)
  - Pairwise similarity
  - Kernel constructed from vector data
- Graph data vs vector data
  - Many machine learning methods use vector data
  - But most can be modified to use graph data
  - Graph are easier to deal with
- We use matrix algebra (adjacency matrix of a graph) as the main algorithmic approach, rather than traditional BFS, DFS, type algorithm.

C. Ding, Data Mining using Matrix Algorithms
Spectral Clustering
(Laplacian matrix based clustering)
Some historical notes

• Fiedler, 1973, 1975, graph Laplacian matrix
• Donath & Hoffman, 1973, bounds
• Hall, 1970, Quadratic Placement (embedding)
• Pothen, Simon, Liou, 1990, Spectral graph partitioning (many related papers there after)
• Hagen & Kahng, 1992, Ratio-cut
• Chan, Schlag & Zien, multi-way Ratio-cut
• Chung, 1997, Spectral graph theory book
• Shi & Malik, 2000, Normalized Cut
Spectral Gold-Rush of 2001
9 papers on spectral clustering

- Ding, He & Zha, KDD 2001. Perturbation analysis of Laplacian matrix on sparsely connected graphs
- Ng, Jordan & Weiss, NIPS 2001, K-means algorithm on the embedded eigen-space
- Dhillon, KDD 2001, Bipartite graph clustering
- Zha et al, CIKM 2001, Bipartite graph clustering
- Gu et al, K-way Relaxation of NormCut and MinMaxCut
Spectral Graph Partitioning

MinCut: \textbf{min} cutsize

cutsize = \# of cut edges

Constraint on sizes: |A| = |B|
2-way Spectral Graph Partitioning

Partition membership indicator: \( q_i = \begin{cases} 
1 & \text{if } i \in A \\
-1 & \text{if } i \in B
\end{cases} \)

\[ J = \text{CutSize} = \frac{1}{4} \sum_{i,j} w_{ij} [q_i - q_j]^2 \]

\[ = \frac{1}{4} \sum_{i,j} w_{ij} [q_i^2 + q_j^2 - 2q_iq_j] = \frac{1}{2} \sum_{i,j} q_i [d_i \delta_{ij} - w_{ij}] q_j \]

\[ = \frac{1}{2} q^T (D - W) q \]

Relax indicators \( q_i \) from discrete values to continuous values, the solution for \( \min J(q) \) is given by the eigenvectors of

\[ (D - W) q = \lambda q \]

(Fiedler, 1973, 1975)

(Pothen, Simon, Liou, 1990)
Recovering Partitions

Subgraph $A, B$ are determined by:

$$A = \{ i \mid q_2(i) < 0 \}, \quad B = \{ i \mid q_2(i) \geq 0 \}$$
Spectral Clustering

**minimize** cutsize  without explicit size constraints

**But where to cut ?**

Need to balance sizes
Clustering Objective Functions

• **Ratio Cut** (Hangen & Kahng, 1992)
  \[ J_{Rcut}(A,B) = \frac{s(A,B)}{|A|} + \frac{s(A,B)}{|B|} \]

• **Normalized Cut** (Shi & Malik, 2000)
  \[ J_{Ncut}(A,B) = \frac{s(A,B)}{d_A} + \frac{s(A,B)}{d_B} \]
  \[ = \frac{s(A,B)}{s(A,A) + s(A,B)} + \frac{s(A,B)}{s(B,B) + s(A,B)} \]

• **Min-Max-Cut** (Ding et al, 2001)
  \[ J_{MMC}(A,B) = \frac{s(A,B)}{s(A,A)} + \frac{s(A,B)}{s(B,B)} \]

\[ s(A,B) = \sum_{i \in A} \sum_{j \in B} w_{ij} \]

\[ d_A = \sum_{i \in A} d_i \]
Spectral Graph Clustering

\[
s(A,B) = \sum_{i \in A} \sum_{j \in B} w_{ij}
\]

Balance weight
Balance size
Balance volume

\[
s(A,A) = \sum_{i \in A} \sum_{j \in A} w_{ij}
\]

Max within-cluster similarities (weights)

Min between-cluster similarities (weights)
Embedding in Principal Subspace

Cluster Self-Aggregation (analyzed using perturbation analysis) explains why spectral clustering is effective

(Hall, 1970: quadratic placement, embedding graph nodes in eigenspace)
Spectral Embedding: Self-aggregation

• Compute $K$ eigenvectors of the Laplacian.
• Embed objects in the $K$-dim eigenspace

(Ding, 2004)
Spectral embedding is not topology preserving

700 3-D data points form 2 interlock rings

In eigenspace, they shrink and separate
Perturbation Analysis

Eigenvectors are piece-wise constant functions
Perturbation Analysis Example

\[ \sum W \]

1\text{st order solution}

\[ \lambda_2 = 0.300, \bar{\lambda}_2 = 0.268 \]

Between-cluster connections suppressed

Within-cluster connections enhanced

Effects of self-aggregation