Spectral Clustering

Jeyanthi S N

CptS 580
Topics in Machine Learning
Seminar Series

Nov 1, 2011
Outline

- Coverage
- Reference
- Linear Algebra + spectral theory
- Problems with spectral methods
- Kernel functions
- The paper
- Remnants
Coverage

- Motivation, basics - y
- Simple / ideal case discussions - y
- Comparison with already known techniques – y (?)
- Family of spectral algorithms – n
- State of the art – y(?)
- Detailed linear algebra derivations – n
- Applications – n
Reference

A tutorial on spectral clustering

Statistics and Computing, Vol. 17, No. 4. (1 December 2007), pp. 395-416
Laplacian (for $K_4$)

$\begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}$

$W = \text{Adj}$

$D$

$L = D - W$

$\lambda_1 = 0, \lambda_2 = 4, \lambda_3 = 4, \lambda_4 = 4$
Laplacian (Disconnected G)

\[
\begin{bmatrix}
-0.70711 & 0.00000 & 0.00000 & 0.70711 \\
0.00000 & 0.70711 & -0.70711 & 0.00000 \\
-0.70711 & 0.00000 & 0.00000 & -0.70711 \\
0.00000 & 0.70711 & 0.70711 & 0.00000
\end{bmatrix}
\]

Number of Zero eigen values = number of connected components of G

\[\lambda_1 = 0, \lambda_2 = 0, \lambda_3 = 2, \lambda_4 = 2\]
General case

Eigen vectors

-0.40825  0.46471  0.70711  -0.17352  -0.23071  -0.18452
-0.40825  0.46471  -0.70711  -0.17352  -0.23071  -0.18452
-0.40825  0.26096  -0.00000  0.34703  0.46141  0.65719
-0.40825  -0.46471  0.00000  0.39160  -0.65573  0.18452
-0.40825  -0.26096  -0.00000  0.34703  0.46141  -0.65719
-0.40825  -0.46471  0.00000  -0.39160  -0.65573  0.18452

-1.0131e-15  4.3845e-01  3.0000e+00  3.0000e+00  3.0000e+00  4.5616e+00

2  -1  -1  -0  -0  -0
-1  2  -1  -0  -0  -0
-1  -1  3  -0  -1  -0
-0  -0  -0  2  -1  -1
-0  -0  -1  -1  3  -1
-0  -0  -0  -1  -1  2

\[ \lambda_s \]

L
Spectral clustering - main algorithms

Input: Similarity matrix $S$, number $k$ of clusters to construct

- Build similarity graph
- Compute the first $k$ eigenvectors $v_1, \ldots, v_k$ of the matrix $L$ for unnormalized spectral clustering
  
  or

  $L_{rw}$ for normalized spectral clustering

- Build the matrix $V \in \mathbb{R}^{n \times k}$ with the eigenvectors as columns
- Interpret the rows of $V$ as new data points $Z_i \in \mathbb{R}^k$

<table>
<thead>
<tr>
<th></th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Z_1$</td>
<td>$v_{11}$</td>
<td>$v_{12}$</td>
<td>$v_{13}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$Z_n$</td>
<td>$v_{n1}$</td>
<td>$v_{n2}$</td>
<td>$v_{n3}$</td>
</tr>
</tbody>
</table>

- Cluster the points $Z_i$ with the $k$-means algorithm in $\mathbb{R}^k$. 
Normalized laplacians

- Commonly used laplacians
  - Satisfy all the required properties

\[
L_{\text{sym}} := D^{-1/2} LD^{-1/2} = I - D^{-1/2} WD^{-1/2}
\]
\[
L_{\text{rw}} := D^{-1} L = I - D^{-1} W
\]

- As many laplacians as there are authors
Clustering using graph cuts

Clustering: within-similarity high, between similarity low

minimize \( \text{cut}(A, B) := \sum_{i \in A, j \in B} w_{ij} \)

Balanced cuts:

\[
\begin{align*}
\text{RatioCut}(A, B) & := \text{cut}(A, B) \left( \frac{1}{|A|} + \frac{1}{|B|} \right) \\
\text{Ncut}(A, B) & := \text{cut}(A, B) \left( \frac{1}{\text{vol}(A)} + \frac{1}{\text{vol}(B)} \right)
\end{align*}
\]

Mincut can be solved efficiently, but RatioCut or Ncut is NP hard.
Spectral clustering: relaxation of RatioCut or Ncut, respectively.
Solving Balanced Cut Problems

Relaxation for simple balanced cuts:

\[ \min_{A,B} \text{cut}(A, B) \text{ s.t. } |A| = |B| \]

Choose \( f = (f_1, \ldots, f_n)' \) with \( f_i = \begin{cases} 1 & \text{if } X_i \in A \\ -1 & \text{if } X_i \in B \end{cases} \)

- \( \text{cut}(A, B) = \sum_{i \in A, j \in B} w_{ij} = \frac{1}{4} \sum_{i,j} w_{ij}(f_i - f_j)^2 = \frac{1}{4} f' L f \)
- \( |A| = |B| \implies \sum_i f_i = 0 \implies f^t 1 = 0 \implies f \perp 1 \)
- \( \|f\| = \sqrt{n} \sim \text{const.} \)

\[ \min_f f' L f \text{ s.t. } f \perp 1, f_i = \pm 1, \|f\| = \sqrt{n} \]

Relaxation: allow \( f_i \in \mathbb{R} \)

By Rayleigh: solution \( f \) is the second eigenvector of \( L \)

Reconstructing solution: \( X_i \in A \iff f_i \geq 0, X_i \in B \) otherwise
K-way Graph Cut Objective functions

- **Ratiocut**$(A_1, \ldots, A_k)$: \[
\frac{1}{2} \sum_{i=1}^{i=k} \left( \frac{W(A_i, \bar{A}_i)}{|A_i|} \right) = \sum_{i=1}^{i=k} \left( \frac{cut(A_i, \bar{A}_i)}{|A_i|} \right)
\]

- **Ncut**$(A_1, \ldots, A_k)$: \[
\frac{1}{2} \sum_{i=1}^{i=k} \left( \frac{W(A_i, \bar{A}_i)}{vol(A_i)} \right) = \sum_{i=1}^{i=k} \left( \frac{cut(A_i, \bar{A}_i)}{vol(A_i)} \right)
\]

$A \subset V$, $\bar{A} = V \setminus A$, complement of $A$,

$|A|$: the number of vertices $\in A$

$W(A, B) := \sum_{i \in A, j \in B} w_{ij}$

$vol(A) := \sum_{i \in A} d_i$

$d_i := \sum_{j=1}^{j=n} w_{ij}$
Solving balanced cut

- Relaxing Ratiocut leads to unnormalized spectral clustering
- Relaxing Ncut leads to normalized spectral clustering
- For \( k > 2 \) (number of partitions), relaxation results in trace minimization problem
- Ratiocut is easier, we will see how to do trace minimization for that problem
Ratio cut – Trace minimization

\[ A_1 \cup A_2 \ldots \cup A_k = V, \ A_i \cap A_j = \emptyset, \ A_i \neq \emptyset \]

Define \( k \) indicator vectors \( h_j = (h_{1,j}, \ldots h_{n,j})' \),

\[ h_{i,j} = \frac{1}{\sqrt{|A_j|}} \text{ if } v_i \in A_j, \ 0 \text{ otherwise} \]

\[ (i=1, \ldots, n; \ j=1, \ldots, k) \]

\[ H \in \mathbb{R}^{n \times k}, \ H = [h_1, \ldots, h_k], \ H' H = I \]

\[ h_i' Lh_i = \frac{\text{cut}(A_i, \bar{A}_i)}{|A_i|} \]

\[ h_i' Lh_i = (H' LH)_{ii} \]
Trace minimization

\[
\text{Ratiocut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} (H' LH)_ii = \text{Tr}(H' LH)
\]

Problem of minimizing Ratiocut:

\[
\min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H' LH) \quad \text{s.t} \quad H' H = I
\]

Use Rayleigh-Ritz procedure
Issues

- Constructing similarity graph \( W \)
  - K-NN, \( \varepsilon \)-neighborhood, Gaussian
    - All of them have parameters
    - Gaussian – similarity graph not sparse
    - For K-\(\text{nn}, \varepsilon \)-based, make sure number of connected components is less than required number of clusters

- Clustering is sensitive to changes in \( W \), and its parameters
Different similarity Graphs

Datasets of different densities

* two half moons
* One Gaussian cloud
Issues (2)

- Computing Eigen vectors
- Number of clusters
- Laplacian
  - Regular graph: all three similar
  - Degree distribution has long tails, then better to use $L_{rw}$
- Etc (some to follow in state of the art)
Kernel functions – a glance

Let the equation of the decision boundary of these two class points be

$$w_1 x^2(1) + \sqrt{2} w_2 x(1) x(2) + w_3 x^2(2) = 0$$

Quadratic discriminant

$$x = \begin{pmatrix} x(1) \\ x(2) \end{pmatrix} \quad z = \begin{pmatrix} z(1) \\ z(2) \\ z(3) \end{pmatrix} \quad \phi : \mathbb{R}^2 \to \mathbb{R}^3$$
Kernel functions (2)

\[ \phi(x) = \phi \left( \begin{bmatrix} x(1) \\ x(2) \end{bmatrix} \right) = \begin{bmatrix} z(1) = x^2(1) \\ z(2) = \sqrt{2} x(1) x(2) \\ z(3) = x^2(2) \end{bmatrix} = z \]

\[ \phi - \text{space decision boundary:} \quad w_1 z_1 + w_2 z_2 + w_3 z_3 = 0 \]

Linear discriminant in z-space
Kernel functions (3)

Let $x_1, x_2 \in \mathbb{R}^2$, $z_1, z_2 \in \mathbb{R}^3$, $\phi$ defined as above,

Dot product in z-space is as follows

$$\left\langle z_1, z_2 \right\rangle = z_1 \cdot z_2 = z_1(1)z_2(1) + z_1(2)z_2(2) + z_1(3)z_2(3)$$

$$= x_1^2(1)x_2^2(1) + 2x_1(1)x_1(2)x_2(1)x_2(2) + x_1^2(2)x_2^2(2)$$

$$= (x_1(1)x_2(1) + x_1(2)x_2(2))^2$$

$$= \left\langle x_1, x_2 \right\rangle^2$$

$$= K(x_1, x_2)$$

Kernel trick – $K$ operates in input space itself
A word about Mercer

- This decade old theorem tells us that any ‘reasonable’ kernel function corresponds to some feature space.

- A mercer kernel matrix is always positive semi definite
  - Most useful property for constructing kernels
  - Eg: gram matrix (also an affinity matrix) is a kernel matrix

- Any PSD matrix can be regarded as a kernel matrix, that is an inner product matrix in some space – Nello Cristianini
Include weights and kernels in the objective function of k-means

\[ D(\{\pi_j\}_{j=1}^k) = \sum_{j=1}^k \sum_{a \in \pi_j} w(a) \| \phi(a) - m_j \|^2 \]

where

\[ m_j = \frac{\sum_{b \in \pi_j} w(b) \phi(b)}{\sum_{b \in \pi_j} w(b)} \]

\( \pi_j \) := clusters, \( w(a) \) := weight for each point 'a'
- Rewrite in trace maximization form using linear algebra tricks

\[ m_j = \Phi_j \frac{W_j e}{s_j}, \quad \text{where} \quad \Phi = [\phi(a_1), \ldots, \phi(a_n)], \quad W_j := \text{diag}(w \in \pi_j), \]

\[ s_j = \sum_{a \in \pi_j} w(a), \quad \text{trace}(AA^T) = \text{trace}(A^T A) = \|A\|_F^2 \]
Graph partitioning objectives naturally lead to trace maximization

Link the two trace maximizations with appropriate substitution
Paper – some comments

- This way other objectives like RatioAssociation can be solved through kernel k-means
  - Details in the tech report by same authors
  - Key is to get trace maximization form

- Complexity
  - kernel k-means, please refer the paper
  - Spectral Clustering – between quadratic and cubic
    - Less for special cases

- Interesting point: Spectral clustering to get initial partition
State of the art

- Original form is not scalable:
  - Parallel Spectral Clustering, Yangqiu Song, Chih-Jen Lin, Machine Learning and Knowledge Discovery in Databases (2008), pp. 374-389
  - Fast approximate Spectral Clustering: Donghui Yan et al, SIGKDD 2009

- Map-Reduce for Machine Learning on Multicore

- Out of sample extension:
  - Spectral Embedded Clustering: A Framework for In-Sample and Out-of-Sample Spectral Clustering, Feiping Nie, Ivor W. Tsang, IJCAI'09
    - Argues that in very high dimensional space spectral clustering fails (manifold assumption fail to hold)
State of the art (2)

- Can the two techniques (K-means, Spec clustering) be combined to form more efficient approach
  - Integrated KL (K-means – Laplacian) Clustering: A New Clustering Approach by Combining Attribute Data and Pairwise Relations, Fei Wang, Chris Ding, and Tao Li, SIAM 2009

- Outlier handling (?)
- Hard / soft clustering (?)
Other clustering algorithms

- Partitioning
- Hierarchical
- Density based
- Spectral clustering is all of the three above
  - Top down clustering
  - Can capture arbitrary shaped points
  - Mutually k-nn, ε-neighborhood are density based similarity graph constructions
Relation to Dimensionality Reduction

- Objective of LE is same as solving Ncut
- All DR methods also rely on EVD
- Isomap is global DR technique (preserve geodesic distances)
  - LLE, SC, LE are local
- Difference in last step:
  - Clustering: threshold the values in eigen vector to split the points, possibly by another clustering algorithm
  - LE : eigenvector is the first component of the reduced dimension representation of the data points
Kernel Based Clustering


  - Adapts from Kernel PCA, decomposition of gram matrix, non-parametric density estimation
  - Claims to provide accurate clustering, estimate model complexity, all parameters, probabilistic outcome for each point assignment, equivalence to SC
Discussion questions

- Is Euclidean distance a generic distance?
- Effect of nature of attributes (discrete or continuous) on a clustering algorithm
References

- A Unified View of Kernel k-means, Spectral Clustering and Graph Partitioning, Inderjit Dhillon, Yuqiang Guan, Brian Kulis, Technical Report TR-04-25, 2005
- www.stanford.edu/~boyd/ee263/lectures/symm.pdf
- www.cs.yale.edu/homes/spielman/561/lect02-09.pdf
- people.inf.ethz.ch/arbenz/ewp/Lnotes/lsevp.pdf
- http://web.eecs.utk.edu/~dongarra/etemplates/node80.html#6462
- http://www.svms.org/kernels/
- Introduction to SVMs (and other kernel based learning methods), N.Cristianini, J.S.Taylor
Some selected literature on spectral clustering

Of course I recommend the following 😊


The three articles which are most cited:


Nice historical overview on spectral clustering; and how relaxation can go wrong:
