Spectral Techniques for Clustering

Nicola Rebagliati

29 April, 2010
Introduction

Spectral Clustering

A Probabilistic Algorithm

Conclusions
Outline

1 Introduction
   - Motivations
   - Graph Laplacians

2 Spectral Clustering
   - Introduction
   - Spectral Clustering: Relaxation
   - Spectral Clustering: Rounding
   - Algorithm
   - Examples

3 A Probabilistic Algorithm
   - Motivations
   - A Spectral Bound
   - A Probabilistic Algorithm

4 Conclusions
Typical supervised learning problems start with two ingredients:

- data $x_i \in \mathcal{R}^d$
- labels $y_i \in \{-1, +1\}$

that form a **training set**: $\{(x_i, y_i)\}_{i=1}^n$

The objective is to **learn** a function $f : \mathcal{R}^d \rightarrow \{-1, +1\}$ such that

$$f(x_i) \approx y_i$$

Many issues:

- Labels contain errors.
- Generalize on future, unseen, data.
- Usually we have few data with respect to the dimensionality $d$. 
In other settings the training set is partially labeled, or without labels:

- **some labels**  Semisupervised Learning
- **no labels**   Unsupervised Learning, i.e. Clustering

**Intrinsic geometry of dataset:**
- Often data live on a low-dimensional manifold
- Finding the manifold make it easier to learn from the data
Intrinsic geometry

Key concept is that of locality:
- neighborhood should only be "local"
- "correct" distances are geodesic (can be completely different from euclidean distances)

Figure: The swiss roll
Similarity Graphs

We use similarity graphs to approximate the underlying manifold. A similarity graph is a weighted symmetric graph, with weights in $[0, 1]$. (0 means dissimilarity, 1 means equality)

Different similarity functions ($n \times n \rightarrow [0, 1]$):

- **Gauss** $w(x_i, x_j) := e^{-\frac{|x_i - x_j|^2}{\sigma^2}}$.
- **KNN** Each data is connected to its $K$ nearest neighbors.
- **$\epsilon$-thres** Each vector $x_i$ is connected to $x_j$ if and only if $\|x_i - x_j\|_2^2 \leq \epsilon$.

The parameters $\sigma, K, \epsilon$ decide the locality of neighborhood. The result is a similarity matrix $W \in \mathcal{R}^{n \times n}$. 
We can reduce the data dimensionality by studying the graph laplacian of $W$. If $D \in \mathcal{R}^{n \times n}$ is a diagonal matrix with the degree of each vertex in its diagonal:

- **Combinatorial Laplacian matrix**: $L_U(W) = D - W$  
  [Fiedler, 1973]

- **Normalized Laplacian matrix**: $L_N(W) = I - D^{-1/2}WD^{-1/2}$  
  [Chung, 1997]

We will focus on the Normalized Laplacian $L_N(W)$
Basic properties of Graph Laplacians

- $L_N(W)$ is a symmetric matrix with positive eigenvalues.
- Eigenvalues (and consequently eigenvectors) are ordered from the smaller $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \leq 2$
- We denote the $i$-th eigenvector as $u_i$. $L_N(W)u_i = \lambda_i u_i$
Spectral Clustering: Spectral Embedding

We want to embed data from an high dimensional space to a smaller one. \((\mathcal{R}^d \rightarrow \mathcal{R}^M, \ M \ll d)\)

The first \(M\) eigenvectors of \(L_N(W)\) give the coordinates of the Spectral Embedding of the dataset [Belkin and Niyogi, 2003]:

- \(u_i(j)\) is the \(j\)-entry of the \(i\)-th eigenvector
- \(\mathcal{R}^d \ni x_j \mapsto [u_1(j), u_2(j), \ldots, u_M(j)] \in \mathcal{R}^M\)
Literature

- Spectral Clustering [Shi and Malik, 1997, Ng et al., 2001, Bach and Jordan, 2006]
- Spectral Graph Theory [Chung, 1997]
Clustering problems are easy to explain, the most common is:

“Divide this set of data into $K$, meaningful, classes”

Indeed even if clustering literature is really vast, clustering itself is still not well-understood from the theoretical point of view.
A simple clustering algorithm is the $K$-means.
The idea is to find $K$ “prototypical” vectors. Each of them is a class mean, for this reason they are also called “centroids”:

$$\mu_j = \frac{1}{n} \sum_{i \in C_j} x_i$$

Each point $x_j$ is assigned to its closest centroid.
The solution centroids minimize the distortion functional

$$\text{Distortion}(X, K) = \min_{C_1, \ldots, C_K} \sum_{j=1}^{K} \sum_{i \in C_j} \|x_i - \mu_j\|_2^2$$
Typical outcome of $K$-means

Figure: $K$-means with two classes.
Back to similarity graphs, what is a reasonable cost functional (like the distortion)?
Example: small weights mean that data are dissimilar.
The quality of a graph partition $A = (A_1, \ldots, A_K)$, with $K$ classes, can be measured with its cut value:

$$\text{cut}(A, W) := \sum_{k=1}^{K} \sum_{i \in A_k} \sum_{j \notin A_k} w(v_i, v_j).$$

In practice [Wu and Leahy, 1993] minimizing the cut leads to poor clusterings with unbalanced partitions.
Graph Cutting

Figure: The minimum cut

Figure: A more balanced cut
The Normalized Cut [Shi and Malik, 1997]

Introducing a normalization leads to better cuts, the normalized cut:

- Minimize the cut.
- Maximize connectivity inside each cluster, the volume.

\[ ncut(A, W) := \sum_{k=1}^{K} \frac{cut(A_k)}{vol(A_k)}. \]

Minimizing the normalized cut is an NP-complete problem [Shi and Malik, 1997] that we denote as \( NCut(G, K) \).
Spectral Clustering: Roadmap

Spectral Clustering are heuristic for minimizing the normalized cut, in order:

- Matrix representation of partitions
- Rewrite the normalized cut as the trace of a matrix multiplication problem
- Relaxing the minimization of the normalized cut
- Recover a partition from the relaxed solution (Rounding)
Matrix representation $H_A \in \mathcal{R}^{n \times K}$ of a partition $A$ is a weighted indicator matrix

$$H_A(i, j) = \begin{cases} \sqrt{\frac{d_i}{\text{vol}(A_j)}} & \text{if } i \in A_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

- $H_A$ has orthonormal columns (thus $H_A^T H_A = I$)
- We denote the set of these matrices as $H_{n,K}$
Relationship between the Normalized Cut and the Normalized Laplacian Matrix

Example, partition $B := \{\{1,2,3\}, \{4,5\}, \{6,7\}\}$

$H_B \in H_{7,3}$

\[
H_B = \begin{pmatrix}
\sqrt{\frac{d_1}{\text{vol}(A_1)}} & 0 & 0 \\
\sqrt{\frac{d_2}{\text{vol}(A_1)}} & 0 & 0 \\
\sqrt{\frac{d_3}{\text{vol}(A_1)}} & 0 & 0 \\
0 & \sqrt{\frac{d_4}{\text{vol}(A_2)}} & 0 \\
0 & \sqrt{\frac{d_5}{\text{vol}(A_2)}} & 0 \\
0 & 0 & \sqrt{\frac{d_6}{\text{vol}(A_3)}} \\
0 & 0 & \sqrt{\frac{d_7}{\text{vol}(A_3)}}
\end{pmatrix}
\]
Relationship between the Normalized Cut and the Normalized Laplacian Matrix

We can now rewrite, equivalently, the \textit{ncut} functional with \( K \) classes as the trace of a matrix multiplication

\[ L_N(W) = I - D^{-1/2}WD^{-1/2} \]

\[ ncut(A, W) = \sum_{k=1}^{K} \frac{cut(A_k)}{vol(A_k)} = Tr(H_A^T L_N(W) H_A). \]

And the related minimization problem:

\[ NCut(G, K) = \min_{H_A \in H_{n,K}} Tr(H_A^T L_N(W) H_A). \]

The conceptual framework changes, now we deal with matrices and matrix multiplications rather than partitions and cuts.
Spectral Clustering: Relaxation

Key idea of Spectral Clustering: drop the constraint $H_A \in H_{n,K}$ and obtain a “relaxed” functional
$H_A$ is substituted by $M \in \mathcal{R}^{n \times K}$, $M^T M = I$

$$\text{RelaxedNCut}(G, K) = \min_{M \text{ s.t. } M^T M = I} \text{Tr}(M^T L_N(W) M).$$

The solution is given by $U_K \in \mathcal{R}^{n \times K}$, the first $K$ eigenvectors of $L_N(W)$.

$$U_K = \begin{pmatrix} \vdots & \ldots & \ldots & \vdots \\ \mathbf{u}_1 & \mathbf{u}_2 & \ldots & \mathbf{u}_K \\ \vdots & \ldots & \ldots & \vdots \end{pmatrix}, \text{Tr}(U_K^T L_K(W) U_K) = \sum_{i=1}^{K} \lambda_i \quad (3)$$
Spectral Clustering: Relaxation

- If the similarity graph is made of $K$ connected components, $A$:
  $$U_K = H_A$$
- Important, the reverse is not true: $U_K$ does not necessarily represent a partition! (Usually $U_K \notin H_{n,K}$)
- We need a procedure to recover a partition from $U_K$. 
We can find all eigenvectors $U$ in $O(n^3)$ time with gaussian elimination.
In practice the power iteration method (Matlab function `eigs`) is faster:
- compute only $U_K$, not whole spectrum
- It can be used with sparse similarity matrices
The rounding step converts $U_K$ into a partition matrix $H_A$

We use a distance between matrices:

$$\text{dist}(A, B) = \left\| AA^T - BB^T \right\|_F$$

The rounding problem can be formalized as:

$$Rounding(U_K, K) = \arg\min_{H_A \in H_{n, K}} \text{dist}(U_K, H_A)^2$$

A slight modification of $K$-means is a rounding algorithm.
Spectral Clustering

\[ ncut(A, W) = \sum_{k=1}^{K} \frac{cut(A_k)}{vol(A_k)} = Tr(H_A^T L_N(W) H_A). \]

Summing up: Spectral Clustering techniques consist of two steps:

- **Relaxation** Find the top \( K \) eigenvectors of the Normalized Laplacian, as we have just seen
- **Rounding** Round these eigenvectors to obtain a resulting partition of \( K \) classes

Or: Minimize the distortion of the \( K \)-dimensional embedding
Ng-Jordan-Weiss Spectral Clustering [Ng et al., 2001]

Simple and significant example of Spectral Clustering. Given a dataset, a target number of classes $K$ and a parameter $\sigma$:

1. Form the similarity matrix $W$, $W_{i,j} = e^{-||v_i - v_j||^2/\sigma^2}$
2. Compute the first $K$ eigenvectors, $U_K$, of $L_N(W)$
3. Normalize each row of $U_K$ to have squared norm 1.
4. Return a partition that attempt to minimize distortion on $U_K$ rows. This is done by running $K$-means algorithm.
Comments

Does NJW (and Spectral Clustering in general) minimize the normalized cut?  
A meaningful quantity to look at: the spectral gap $\lambda_{K+1} - \lambda_K$ ($K$ is the number of classes)

- When the spectral gap is large we know it works. (See also [Meilă, 2006])
- In general, not necessarily (minimizing the normalized cut is NP-complete)
We show two examples of Spectral Clustering

- In the first example NJW works and correctly divide a dataset, two half moon that are not linearly separable
- In the second example, a contribution of my thesis, NJW gives a non-optimal partition
An example: moon data

Two half moons are not linearly separable

- 200 data in each half moon.
- The similarity $w(x_i, x_j) = e^{-\frac{||v_i - v_j||^2}{0.3^2}}$.
- $K = 2$. 

Nicola Rebagliati
An example: moon data

**Figure:** Spectral embedding given by the first two eigenvectors $[u_1, u_2]$
An example: moon data

Figure: NJW divides the two half moons
There are examples where Spectral Clustering result is really different from the optimal one.

- [Guattery and Miller, 1995] is a graph-theoretical example
- In my thesis I show examples arising from points in the plane. In particular it seems that more than $K$ eigenvectors should be needed.
Two rectangles and a segment [Rebagliati and Verri, 2010]

- 350 data. (100 in the first rectangle, 100 in the second rectangle and 150 in the segment)
- The similarity between points $p_i, p_j$ is

$$w(p_i, p_j) := e^{-\|v_i - v_j\|^2/0.22^2}$$
NJW Solution (not optimal)

Figure: The solution given by NJW, 3 classes, its normalized cut is 0.16.
Figure: Reference solution, chosen by visual inspection, its normalized cut is 0.09 (56\% of 0.16).
Small spectral gap and Eigenvectors swapped

A more careful analysis shows that $\lambda_3$ and $\lambda_4$ are close (thus a small spectral gap), in this case they tend to “swap”, i.e. they are not stable w.r.t noise and invert their position.

- Thus the fourth eigenvectors $u_4$ could have been in place of the third eigenvector $u_3$.
- Applying NJW with matrix $[u_1, u_2, u_4] \in \mathcal{R}^{n \times 3}$ gives the reference partition.

But is the reference solution optimal?
Two main questions

Spectral Clustering rounds the first $K$ eigenvectors. But we showed that we can recover good solutions from more than $K$ eigenvectors. This naturally prompts two questions:

- More than $K$ eigenvectors should be needed, but how many should we choose?
- How do we round the chosen eigenvectors? (Rounding was defined for matrix $\in R^{n \times K}$)

This two questions aim at generalizing spectral clustering.
We are able to show a bound that helps choosing the number of eigenvectors, the approach is that of using the matrix representation $H_A \in \mathcal{R}^{n \times K}$ of a partition $A$.

- $L_N(W) = U \Lambda U^T$. ($U = [u_1, u_1, \ldots, u_n, ] \in \mathcal{R}^{n \times n}$)
- Choose $M \geq K$
- Divide $U = [U_M, U_e]$. ($U_M \in \mathcal{R}^{n \times M}$, $U_e \in \mathcal{R}^{n \times n-M}$)
- Decompose $H_A = [U_M, U_e] \begin{bmatrix} C_A \\ E_A \end{bmatrix}$
A Spectral Bound

\[ H_A = [U_M, U_e] \begin{bmatrix} C_A \\ E_A \end{bmatrix} \]

- \((C_A \in \mathbb{R}^{M \times K}, E_A \in \mathbb{R}^{n-M \times K})\).
- We call \(\|E_A\|_F^2\) the \(Residual(H_A, M)\).
- If the \(Residual(H_A, M)\) is small, almost all the information on \(H_A\) is in \(C_A\) (Notice matrix dimensions, \(M \ll n\)).
- If the \(Residual(H_A, M)\) is small, we can discard \(U_e\).
- \(C_A\), “C” stands for Compression.
A Spectral Bound [Rebagliati and Verri, 2009]

\[ Residual(H_A, M) \leq \frac{ncut(A,W) - \sum_{i=1}^{K} \lambda_i}{\lambda_{M+1} - \lambda_K} . \]

- The numerator \( ncut(A, W) - \sum_{i=1}^{K} \lambda_i \) depends on the relaxation
- The denominator \( \lambda_{M+1} - \lambda_K \) is an extended spectral gap
How do we round $U_M$?

Let $A^*$ be a partition minimizing the normalized cut.

$$H_A^* = [U_M, U_e] \begin{bmatrix} C^*_A \\ E^*_A \end{bmatrix}$$

- For simplicity consider the $Residual(H_A^*, M) \approx 0$,
- $U_M C^*_A \approx H_A^*$
- Note $C^*_A^T C^*_A \approx I$

If we find $C^*_A$ we have found also $H_A^*$. (Note: $C^*_A \in \mathcal{R}^{M \times K}$ is much smaller than $H_A^* \in \mathcal{R}^{n \times K}$)

The probabilistic algorithm tries to sample $N \approx C^*_A$, thus $U_M N$ is a candidate best partition.
Input: a graph $G$ with similarity matrix $W$, parameters $K$, $M$.

- Compute the first $M \geq K$ eigenvectors $U_M$ of the Normalized Laplacian $L_N(W)$
- For $k=1:\max\_rep$ times:
  - sample a matrix $N \in \mathcal{R}^{M \times K}$ such that $N^T N = I$
  - $(A)_k := \text{round } U_M N$ (for example using NJW)
  - compute normalized cut of $(A)_k$
- Return the $(A)_k$ with the minimum normalized cut value
I omitted some tricks that help saving (some) computation.

The algorithm is easily parallelizable.

Choosing parameter $M$ in not easy:

- The bigger the $M$ the more $C_A$ represents $H_A$.
- The smaller the $M$ the more eigenvectors we discard and the easiest is to sample $N \approx C_A$. 
Consider again the example with two rectangles and a segment.

**Figure:** Reference solution, chosen by visual inspection, its normalized cut is 0.09.
Figure: Behavior of the Spectral Bound $\frac{\text{ncut}(A,W) - \sum_{i=1}^{K} \lambda_i}{\lambda_{M+1} - \lambda_K}$, an upper bound on the Residual($H_A^*, M$).
Choose $M = 9$, $\text{max\_rep} = 10^4$, the reference solution is found.
Experiments on a benchmark [Rebagliati and Verri, 2010]

We compared results with those of the Graph Partitioning Archive [Soper et al., 2000]
We distributed the probabilistic algorithm in the DISI labs thanks to the ShareGrid service managed by Marco Ferrante.
Experiments on a benchmark [Rebagliati and Verri, 2010]

<table>
<thead>
<tr>
<th>graph</th>
<th>vertices</th>
<th>best Alg.1</th>
<th>$\overline{E}(\text{Alg.1})$</th>
<th>FMC</th>
<th>bench.</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>2851</td>
<td>189</td>
<td>196.9 (± 4 )</td>
<td>260</td>
<td>189 [PROBE]</td>
</tr>
<tr>
<td>3elt</td>
<td>4720</td>
<td>92</td>
<td>99.8 (± 5 )</td>
<td>117</td>
<td>90 [PROBE]</td>
</tr>
<tr>
<td>uk</td>
<td>4824</td>
<td>21</td>
<td>25.6 (± 2 )</td>
<td>31</td>
<td>20 [mpM4.0]</td>
</tr>
<tr>
<td>add32</td>
<td>4960</td>
<td>11</td>
<td>14.5 (± 3 )</td>
<td>23</td>
<td>11 [Ch2.0]</td>
</tr>
</tbody>
</table>

**Table:** Some results for the balanced cut on benchmark graphs from the Graph Partitioning Archive, $K = 2$ and $M = 10$. We repeated $10^2$ times the probabilistic algorithm with $\max\_\text{rep} = 10^4$. Column FMC is the results of the Fiedler Median Cut. Column **bench.** is the value of the benchmark partition.
Outline

1 Introduction
   - Motivations
   - Graph Laplacians

2 Spectral Clustering
   - Introduction
   - Spectral Clustering: Relaxation
   - Spectral Clustering: Rounding
   - Algorithm
   - Examples

3 A Probabilistic Algorithm
   - Motivations
   - A Spectral Bound
   - A Probabilistic Algorithm

4 Conclusions
Conclusions

Spectral Clustering is an interesting heuristic that does not necessarily work.

Contributions:

- Examples
- A spectral bound for choosing the number of eigenvectors
- Testing of a simple, probabilistic, algorithm whose results are close, or equal, to the state of the art.
Conclusions

Future work:

- Specialize the spectral bound to restricted classes of graphs
- Study a more efficient version of the probabilistic algorithm

Thanks for your attention!


From graphs to manifolds - weak and strong pointwise consistency of graph laplacians. pages 470–485.


Technical Report DISI-TR-09-04, DISI.

Rebagliati, N. and Verri, A. ((submitted) 2010).
A randomized algorithm for spectral clustering.
accepted to ESANN 2010.

Normalized cuts and image segmentation.

A combined evolutionary search and multilevel approach to graph partitioning.

Nonlinear dimensionality reduction by locally linear embedding.
