Spectral Graph Clustering

Benjamin Auffarth
Universitat de Barcelona
course report for *Técnicas Avanzadas de Aprendizaje*
at Universitat Politècnica de Catalunya

January 15, 2007

Abstract

Spectral clustering is a powerful technique in data analysis that has found increasing support and application in many areas. This report is geared to give an introduction to its methods, presenting the most common algorithms, discussing advantages and disadvantages of each, rather than endorsing one of them as the best, because, arguably, there is no black-box algorithm, which performs equally well for any data. We present results from previous studies and conclude that methods based on Ncut and multiway are most promising for general application.

Contents

1 Introduction **2**

2 Spectral clustering **2**
2.1 Notation . . . 2
2.2 Clustering . . . 3
2.3 Similarity graphs . . . 4
2.4 Laplacians and their properties . . . 5

3 Clustering algorithms **5**
3.1 Non-spectral clustering algorithms . . . 6
3.2 Spectral clustering algorithms . . . 7
3.2.1 Multiway . . . 8
3.2.2 Recursive . . . 8

4 Discussion **9**

5 Conclusions **10**

References **11**

1 Introduction

Spectral clustering uses information obtained from the eigenvalues and eigenvectors of their adjacency matrices for partitioning of graphs. It has many applications, such as in image segmentation (e.g. (Shi & Malik, 2000)) and social network analysis (e.g. (Newman, Watts, & Strogatz, 2002)). The methods are called spectral, because they make use of the spectrum of the adjacency matrix of the data to cluster the points.

Spectral clustering algorithms have found an increasing following, especially after (Shi & Malik, 2000) and (Ng, Jordan, & Weiss, 2002). As opposed to k-means clustering, which results in convex sets, spectral clustering can solve problems, such as intertwined spirals, because it does not make assumptions on the form of the cluster. Given a sparse similarity graph, spectral clustering can be implemented efficiently even for large data sets (cf. (Verma & Meila, 2003)). Further advantages of learning about spectral clustering algorithms were noted as follows (Luxburg, 2006):

- solution of clustering problems by standard linear algebra methods
- often more efficient than traditional algorithms (e.g. k-means, single linkage)

A concise introduction into the field of spectral clustering can be found in (Luxburg, 2006), which this report relies on heavily. The comparison of algorithms herein after summarizes (Verma & Meila, 2003), a systematic comparison of spectral clustering algorithms, which demonstrated that these complement and/or compete with existing methods with convincing results. Most of the presented algorithms work in combination with clustering algorithms, such as k-means (the knowledge of which this report presupposes, refer to (MacKay, 2003) for an overview about clustering).

Due to the subtle nature of the relationship between spectral parameters and the properties of datasets this report tries to outline the functioning of the algorithms in question rather than arguing for one of them as the best. Nevertheless comparison shows that some algorithms are more stable, have desirable properties, and have superior clustering results.

2 Spectral clustering

2.1 Notation

The notation for spectral graph theory foots on graph theory and appeals to intuition. Except for minor differences in the names of matrices, variables, and

counters, this notation seems pretty standard. This section introduces briefly the mathematical notations of the algorithms that will be presented.

A *graph* or *undirected graph* G can be written as an ordered pair $G := (V, E)$, where V is a set of *vertices* or *nodes*, and E is a set of pairs (unordered) of distinct vertices, called *edges* or *lines*. The vertices belonging to an edge are called the *ends*, *endpoints*, or *end vertices* of the edge. (Wikipedia, 2007b)

The *adjacency matrix* (also *similarity matrix* or *weight matrix*) of a finite directed or undirected graph G on n vertices is the $n \cdot n$ matrix where the nondiagonal entry w_{ij} is the number of edges from vertex i to vertex j, and the diagonal entry w_{ii} is either twice the number of loops at vertex i or just the number of loops (usages differ, depending on the mathematical needs; this report is not concerned with reflexive connections). The adjacency matrix is symmetric for undirected graphs. (Wikipedia, 2006a) In the following, we assume $w_{ij} = w_{ji} \geq 0$.

Given $A \in V$, its complement, $V \backslash A$ will be denoted as $\bar{A}$. $i \in A$ shall be shorthand for indices of $\{i | v_i \in A\}$. $|A|$ will denote the number of vertices in A. $vol(A)$ measures the size of A by the weights of its edges, i.e. $vol(A) := \sum i \in A d_i$.

The *degree* (or *valency*) of a vertex is the number of edge endpoints to the vertex. Loops are counted twice. The *degree matrix* D for G is a $n \times n$ square matrix defined as (Wikipedia, 2007a)

$$d_{i,j} := \begin{cases} deg(v_i) & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Intuitively, a subset $A \in V$ is *connected*, if pathes between any two points in A need only points in A. A is called *connected component* with respect to $\bar{A}$, if A is connected, and there are no edges between vertices in A and $\bar{A}$. Subsets $A_1, \ldots, A_k$ represent a *partitioning* of V, if, for all $1 \leq j, x \geq k$, $A_i \bigcap A_j = \emptyset$ and $\bigcup_{i=1}^{k} A_i = V$ (usually also defined nonempty).

2.2 Clustering

Generally speaking, clustering means partitioning of a graph, so that the edges between different groups have low similarity (low distance) and the edges within a group have high weight (low distance). The first requirement for the partitioning can be stated as the *minicut criterion*, which has to be minimized:

$$cut\,(A_1, \ldots, A_k) = \sum_{i=1}^{k} cut(A_k, \bar{A_k}), \text{ where}$$

$$cut(A, \bar{A}) = \sum_{i \in A_k, j \in \bar{A_k}} w_{ij}$$

Following the minicut requirement, often only a little group of points is isolated. For this reason, tweaks have been introduced:

- $RadioCut(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A_i})}{|A_i|}$. (Hagen & Kahng, 1992)
- $Ncut(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{cut(A_i, \bar{A_i})}{vol(A_i)}$. (Shi & Malik, 2000) (*Ncut* for "normalized cut").

As for the second requirement, within-cluster similarity means optimizing

$$\sum_{l=1}^{k} \sum_{i,j \in A_l} w_{ij}$$

Within-cluster similarity is maximized if $cut(A, \bar{A})$ is small and $vol(A)$ is big.

Therefore, Ncut implements the second criterion. Ncut can be interpreted as cutting through edges rarely transitions by a random walk. RadioCut, by maximizing $|A_i|$, as within-cluster similarity is not related to the number of vertices in A, does not implement this requirement.(Luxburg, 2006)

2.3 Similarity graphs

Given a set of points, $x_1, \ldots, x_n$, and their distances $d_{jk} \in D$ (not related to the degree), there are several constructions to obtain a graph (i.e. construct W), which are regularly used in spectral clustering.

The ϵ *-neighorhood graph* is the most simple possibility. All connections with distances below a threshold are set to 1. I.e.

$$\sum_{<ij>} w_{ij} = 1, \text{ with}$$

$$\left\{ < ij > | d_{ij} \in D, d_{ij} < \epsilon \right\}.$$

k-nearest neighbor graphs lead to a directed graph. The *k*-shortest distances from *i* are connected. In order to make the graph undirected, directions can be ignored, e.g. constructing a *k*-nearest neighbor graph, then doing either $A \bigcup A^{\mathrm{T}}$ (usually this is called *k*-nearest neighbor graph) or $A \bigcap A^{\mathrm{T}}$ (*mutual k-nearest neighbor graph*).

A *fully connected graph* results from connecting all points with positive similarity with each other.

2.4 Laplacians and their properties

The spectral algorithms presented here foot on eigenvectors of Laplacians, which are a combination of the weight and the degree matrix. For a more thorough and broader discussion of mathematical properties of Laplacians refer to (Mohar, 1997) and (Chung, 1997).

The (unnormalized) graph *Laplacian* is defined as $L = D - W$. For a graph G and its admittance matrix L with eigenvalues $\lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{n-1}$ these properties are important, in the context of spectral clustering. (Wikipedia, 2006c)

- L is always positive-semidefinite ($\forall i, \lambda_i \geq 0$).
- The multiplicity of 0 as an eigenvalue of L is the number of connected components of G.
- λ_1 is called the algebraic connectivity.
- The smallest non-trivial eigenvalue of L is called the spectral gap.

Noteworthy is furthermore, that matrices with identical off-diagonal elements have the same unnormalized Laplacians.

The *normalized graph Laplacian* is defined in two two distinct ways:

- $L_{sym} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = 1 - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ (symmetric matrix)
- $L_{rw} := D^{-1} L = I - D^{-1} W$. (random walk)

Some properties are

- L_{rw} and L_{sym} are positive semi-definite and have n non-negative real-valued eigenvalues $0 = \lambda_1 \leq \ldots \leq \lambda_n$.
- The multiplicity k of the eigenvalue 0 of both L_{rw} and L_{sym} equals the number of connected components $A, \ldots, A_k$ in the graph.

Refer to (Chung, 1997) for an in-depth discussion of normalized Laplacians.

3 Clustering algorithms

Spectral clustering became popular with, among others, (Shi & Malik, 2000) and (Ng et al., 2002). Its efficiency is mainly based on the fact that it does not make any assumptions on the form of the clusters. This property comes from the mapping of the original space to a eigen space. Algorithms differ basically in the number of eigenvectors they use for partioning.

Algorithms can be categorized based on the number of eigenvectors they use: (Verma & Meila, 2003)

- One eigenvector – recursively uses a single eigenvector on partitions (**recursive**).
- Many eigenvectors – Directly computes a **multiway partition** of the data.
- **Non spectral** – Grouping algorithms that can be used in conjunction with multiway spectral algorithms. Algorithms that are not spectral are shortly mentioned in the next section and some results comparing them to spectral algorithms will be presented in the discussion.

(Verma & Meila, 2003) outlines three steps in the algorithms.

- Normalization (preprocessing step): this was ignored here and also did not find more space in (Verma & Meila, 2003) beyond a mention.
- Spectral mapping: eigenvectors, usually based on a Laplacian, are computed as a mapping of the data points.
- Grouping: clustering algorithms group the points in original or mapped domain.

3.1 Non-spectral clustering algorithms

This short section is dedicated to non-spectral clustering algorithms. They are often used in spectral algorithms as post-processing step (see section 3.2). (Verma & Meila, 2003) compare different algorithms as post-processing steps after the spectral mapping. For some broader discussion of clustering algorithms in general, see (Shamir & Sharan, 2002).

The objective of *k-means* is to minimize the total intra-cluster variance, or, the squared error function, defined as

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

, where there are k clusters S_i, $i = 1, 2, \ldots, k$ and μ_i is the centroid or mean point of all the points $x_j \in S_i$. Starting by partitioning the input points into k initial sets, it calculates the mean point, or *centroid*, of each set, then iterates constructing a new partition by associating each point with the closest centroid and recalculating the centroids for the new clusters until no changes occur. (Wikipedia, 2006b) (refer to (MacKay, 2003) for more detail)

Ward's Algorithm (mentioned, but not explained in (Verma & Meila, 2003)) was introduced in (Ward, 1963)[1] is a agglomerative clustering algorithm. Given n

[1]Explanation found at `http://iv.slis.indiana.edu/sw/ward.html`.

data points (n sets), it reduces them to $n-1$ sets by considering the union of all possible $n(n-1)/2$ pairs and selecting a union by some criterion. A quantitative estimate of the loss associated with each stage in the grouping can be obtained according to:

$$ESS = \sum_{i=1}^{n} x_i^2 - \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right)^2$$

Single linkage proceeds at each iteration to construct agglomerations of data sets (initially points) that exceed similarity thresholds. This algorithm finds application in sequence alignment, e.g. biomedical data. (Shamir & Sharan, 2002)

The *anchor algorithm* was given by (Verma & Meila, 2003) (simplified from (Moore, 2000)). It produces a structure similar to a *ball-tree* or a *metric tree*. An anchor is analogous to a centroid in the k-means algorithm. k is the counter for the iterations of the algorithm, $\overline{k}$ indicates need for more iterations, min_n is the number of minimally desired points in a cluster and K the number of desired clusters. Initially a point is chosen randomly as an anchor and $k = 1$, $\overline{k} = 0$. Iterations proceed by constructing a cluster C_k, containing all points closer to x_k than to any other anchor (sorted by decreasing distance), if $|C_k| < min_n$ then $\overline{k}++$. Start new iteration if $k - \overline{k} < K$.

3.2 Spectral clustering algorithms

For the algorithms we assume data points $x_1, \ldots, x_n$ and their similarities $s_{ij} = s(x_i, x_j)$, where $1 \leq i, j \leq n$ and $s \in S$. S is symmetric and non-negative. They are rather similar, except for the different Laplacians they use. The change of representation from x to u with the help of the Laplacians enhances the cluster-properties of the data. (Luxburg, 2006)

Algorithms introduced here are *unnormalized spectral clustering*, (Shi & Malik, 2000) (SM), (Ng et al., 2002) (NJW), (Kannan, Vempala, & Vetta, 2004) (KVV), (Meila & Shi, 2000) (Multicut). Presentation follows (Luxburg, 2006) for the first three and (Verma & Meila, 2003) for the latter ones.

Unnormalized spectral clustering proceeds as in the following:
Inputs: k, the number of desired clusters, X, and S.

- Following the steps in section 2.3 construct W,
- then calculate L,
- then the first k eigenvectors $v_1, \ldots, v_k$ of L.

- Let $V \in \mathbb{R}^{nxk}$ contain $v_1, \ldots, v_k$ as columns and $y_i \in \mathbb{R}$, with $i = 1, \ldots, n$, correspond to the i-th row of V. Cluster the points y_i with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$, with $A_i = \left\{ j | y_j \in C_i \right\}$.

Note that similar to RadioCut, unnormalized spectral clustering does not optimize within-cluster similarity.(Luxburg, 2006)

3.2.1 Multiway

(Meila & Shi, 2000) proposed the Multiway algorithm (**Mcut**). It proceeds by computing $L_r w$, computes the k largest eigenvectors, forms the matrix V, whose columns are $v_1, \ldots, v_k$, and clusters the rows of V as points in a k-dimensional space. (cf. (Verma & Meila, 2003))

(Ng, Jordan, & Weiss, 2002) (**NJW**) works as follows:
Input: $S \in \mathbb{R}^{nxn}$, number of desired clusters k.

- Following the steps in section 2.3 construct W,
- Compute L_{sym} and its k first eigenvectors.
- Let $V \in \mathbb{R}^{nxk}$ contain $v_1, \ldots, v_k$ as columns. Let U be V normalized to row sums with norm 1. Let $y_i \in \mathbb{R}$, with $i = 1, \ldots, n$, correspond to the i-th row of U.
- Cluster the points y_i with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$, with $A_i = \left\{ j | y_j \in C_i \right\}$.

3.2.2 Recursive

(Shi & Malik, 2000) (**SM**) uses generalized eigenvectors, which correspond to the eigenvectors of matrix L_{rw}. The presentation of the algorithm is simplified according to (Luxburg, 2006)[2]. A thorough discussion can be found in (Verma & Meila, 2003).

Input: $S \in \mathbb{R}^{nxn}$, number of desired clusters k.

- Following the steps in section 2.3 construct W,
- Compute L_{rw} and its k first eigenvectors.

[2]Shi et alii provide their code for normalized cut image segmentation for download at `http://www.cis.upenn.edu/~jshi/GraphTutorial/index.html`

- Let $V \in \mathbb{R}^{nxk}$ contain $v_1, \ldots, v_k$ as columns and $y_i \in \mathbb{R}$, with $i = 1, \ldots, n$, correspond to the i-th row of V. Cluster the points y_i with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$, with $A_i = \left\{ j | y_j \in C_i \right\}$.

(Kannan, Vempala, & Vetta, 2004) (**KVV**) uses "Cheeger conductance" as criterion for the optimal cut. Conductance is a way to measure how hard it is to transition within a set of nodes. Strangely, except for (Verma & Meila, 2003) a google search provided no mention of "Cheeger conductance" [3]. (Verma & Meila, 2003) gives the Cheeger conductance of a clustering C as:
$\theta(C, \bar{C}) = \frac{cut(C,\bar{C})}{minvol(C),vol(C)}$

It also normalizes the similarity matrix to row sum one at every iteration of the algorithm. Except for these two differences, KVV is the identical to SM.

4 Discussion

(Luxburg, 2006) shows for arbitrary k that unnormalized spectral clustering corresponds to optimization by RadioCut and that normalized spectral clustering according to (Shi & Malik, 2000) corresponds to optimization by Ncut.

(Verma & Meila, 2003) compared the algorithms that were presented here plus some variations (refer to (Meila, 2002) for comparison measures of clustering algorithms). Targets for clusterings known, they compared especially mutual information between partitions according to clustering error measures. In the next paragraphs the results they found will be summarized.

For one dataset, *S100*, a dataset constructed to conform to optimal conditions for the multicut algorithms, as expected, the multiway algorithms performed the best, conductance-based algorithms underperforming.

As for the first real data set, gene expression data[4], the multiway spectral algorithms are the most stable of all algorithms. The best of recursive spectral are not too far behind and gain similar levels of errors and variation with increasing noise ratio. Linkage based clustering proved very sensitive to noise.

The third data set, NIST handwritten digits[5], the linkage algorithm again deteriorates and, as for multiway and recursive algorithms no significant difference could be found. With a reduced data set that was easier to cluster (well-separated digits, 0, 2, 4, 6, 7), multiway performed close to ceiling.

[3] Google search: `http://www.google.es/search?q="Cheeger+conductance"&btnG=Search`
[4] Provided by Ka Yee Yeung at `http://staff.washington.edu/kayee/model/`.
[5] available at `http://www.itl.nist.gov/iad/894.03/databases/defs/dbases.html`.

"Ideal" conditions for the weight matrix W for NJW and Mcut, called *block stochastic* are discussed in (Verma & Meila, 2003) and further discussed in (Meila & Shi, 2001). Block stochastic P is perfect for NJW and Mcut, where the clusters in the spectral domain are orthogonal, also ameliorated are recursive algorithms.

For Mcut and NJW, stability can be improved by using the top k eigenvectors of the generalized eigen space $Wx = \lambda Dx$, which is theoretically equivalent but numerically more stable. They showed equivalence, theoretically and practically, of the two algorithms, in the case of perfect W (block stochastic) and great similarity otherwise. (Verma & Meila, 2003)

Verma and Meilá showed spectral methods are competitive and more stable to noise than other tested algorithms. Spectral algorithm performed generally significantly better than the linkage algorithm even when the clusters are not well separated (digits sets). Ncut is better than conductance as criterion and multiway mostly better than recursive. As for the non-spectral methods, ward and k-means perform slightly more successful than anchor.

5 Conclusions

Spectral algorithms are a simple and efficient method for clustering. The step of preprocessing was ignored here. (Verma & Meila, 2003) mentions smoothing as a way to further improve results and explains preprocessing steps for different datasets, however, explicitly leaves out preprocessing for comparison purposes in most cases. Preprocessing is an important step and may help significantly elevate clustering results.

Choosing a similarity graph can be non-trivial and may require extensive preprocessing. However, once there is a similarity graph, the problem is linear and spectral methods do not suffer intrinsically from the problem of local optima. (cf. (Verma & Meila, 2003), (Meila & Shi, 2001))

Also beyond the scope of this report were the automatic determination of k, the number of clusters and the selection/weighting of different dimension of the weight matrix. (however see (Dy & Brodley, 2004) for the FSSEM approach)

We disqualified RadioCut and unnormalized spectral algorithms because they do not optimize within-cluster similarity, and discussed advantages and disadvantages of several other algorithms and conclude that spectral algorithms promise to give superior results than non-spectral methods, especially their properties of being not restricted to convex regions of similarity and the robustness to noise make them attractive. As for differences between spectral algorithms, (Verma & Meila, 2003) established that Ncut was superior to conductivity as criterion and the "near" equivalence of NJW and Mcut. They further showed in their article that multiway was mostly better than recursive.

References

Chung, F. R. K. (1997). *Spectral graph theory*. Providence, RI: American Mathematical Society.

Dy, J. G., & Brodley, C. E. (2004). Feature selection for unsupervised learning. *J. Mach. Learn. Res., 5*, 845–889.

Hagen, L., & Kahng, A. (1992). New spectral methods for radio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design, II(9)*, 1074–1085.

Kannan, R., Vempala, S., & Vetta, A. (2004). On clusterings: Good, bad and spectral. *J. ACM, 51*(3), 497–515.

Luxburg, U. von. (2006, August). A tutorial on spectral clustering. In (MPI-Technical Reports No. 149). Tubingen: Max Planck Institute for Biological Cybernetics.

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. `http://www.cambridge.org/0521642981`: Cambridge University Press. (Available from `http://www.inference.phy.cam.ac.uk/mackay/itila/`)

Meila, M. (2002). Comparing clusterings. In (Technical Report 418). Washington: University of Washington, Statistics department.

Meila, M., & Shi, J. (2000). Learning segmentation by random walks. *NIPS*, 873-879.

Meila, M., & Shi, J. (2001). A random walks view of spectral segmentation. *AISTATS 2001*.

Mohar, B. (1997). Some applications of laplace eigenvalues of graphs [NATO ASI Series C]. *G. Hahn and G. Sabidussi, editors, Graph Symmetry: Algebraic Methods and Applications, 497*, 227–275.

Moore, A. (2000, February). *The anchors hierarchy: Using the triangle inequality to survive high dimensional data* (Tech. Rep. No. CMU-RI-TR-00-05). Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

Newman, M., Watts, D., & Strogatz, S. (2002). Random graph models of social networks. *Proc. Natl. Acad. Sci. USA, 99*, 2566–2572.

Ng, A. Y., Jordan, M., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. *T. Dietterich, S. Becker, and Z. Ghahramani (Eds.) – Advances in Neural Information Processing Systems, 14*.

Shamir, R., & Sharan, R. (2002). Algorithmic approaches to clustering gene expression data. *Current Topics in Computational Molecular Biology, T.Jiang, T. Smith, Y. Xu, M. Q. Zhang (editors)*, 269–300.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*.

Verma, D., & Meila, M. (2003). A comparison of spectral clustering algorithms. *University of Washington, uw-cse-03-05-01*.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association, 58(301),* 236–244.

Wikipedia. (2006a). *Adjacency matrix — wikipedia, the free encyclopedia.* ([Online; accessed 9-January-2007])

Wikipedia. (2006b). *K-means algorithm — wikipedia, the free encyclopedia.* ([Online; accessed 11-January-2007])

Wikipedia. (2006c). *Laplacian matrix — wikipedia, the free encyclopedia.* ([Online; accessed 9-January-2007])

Wikipedia. (2007a). *Degree matrix — wikipedia, the free encyclopedia.* ([Online; accessed 9-January-2007])

Wikipedia. (2007b). *Graph (mathematics) — wikipedia, the free encyclopedia.* ([Online; accessed 9-January-2007])