Title
Literature Review on Spectral Clustering

Permalink
https://escholarship.org/uc/item/08q7s99b

Author
Chen, Mengna

Publication Date
2013

Peer reviewed|Thesis/dissertation
Literature Review on Spectral Clustering

A thesis submitted in partial satisfaction of the requirements for the degree Master of Science in Statistics

by

Mengna Chen

2013
ABSTRACT OF THE THESIS

Literature Review on Spectral Clustering

by

Mengna Chen

Master of Science in Statistics
University of California, Los Angeles, 2013
Professor Yingnian Wu, Chair

The spectral clustering algorithm is an algorithm for putting N data points in an I-dimensional space into several clusters. Each cluster is parameterized by its similarity, which means that the points in the same group are similar and points in different groups are dissimilar to each other. Recently, spectral clustering has become an increasingly adopted tool and has been applied in many areas such as statistics, machine learning, pattern recognition, data mining, and image processing. This paper discusses these methods in details and later on introduce the relationship between spectral clustering and k-means clustering and spectral clustering's applications in image segmentation, educational data mining, entity resolution and speech separation.
The thesis of Mengna Chen is approved.

Nicolas Christou

Qing Zhou

Yingnian Wu, Committee Chair

University of California, Los Angeles

2013
# Table of Contents

Introduction ................................................................. 1-2  
Graph Construction ......................................................... 2-4  
Graph Laplacians and their properties ................................. 4-5  
Graph Partitioning ........................................................... 6-9  
Relationship with kernel K-means algorithm ......................... 9-10  
Applications ...................................................................... 11-22  
Conclusion ......................................................................... 23  
References .......................................................................... 24
List of Figures

**Figure 1** Image including the information on vertex and weight .......................... 2

**Figure 2** Image including the information on vertex and weight .......................... 2

**Figure 3** Segmentation results for natural scene images. Left to right: normalized cut, self-tuning spectral clustering and proposed method ................................................................. 14

**Figure 4** The technique for making a prediction for a test point ................................. 14

**Figure 5** The left column is for k-means clustering and the right column is for spectral clustering. The top row show the ASSISTment data scaled down by multi-dimensional scaling to three dimensions and the clusters identified by both k-means and spectral clustering. The rows below are the different planar views. ................................................................. 15

**Figure 6** The 5 fold cross validated errors using different prediction models from K=1 to 7 for both k-means and spectral clustering ................................................................. 16

**Figure 7** Top image for the spectrogram of English speakers, colored in black and grey; Bottom image for the blind segmentation obtained with proposed method ................................................. 21

**Figure 8** Top image for the spectrogram of English speakers, colored in black and grey; Bottom image for the blind segmentation obtained with proposed method ................................................. 22
1. Introduction

“Human brains are good at finding regularities in data. One way of expression regularity is to put a set of objects into groups that are similar to each other” (Mackay, 2003, p. 284). Clustering is a process of grouping a set of objects into classes with similar characteristics. In this research report, I will discuss spectral clustering, a more powerful and specialized clustering algorithm (compared to K-means).

There are several motivations for clustering as following:

- A good clustering has predictive power.
- Clusters can be useful in communication because they allow people to put objects with similar features into one category and to name them.
- Failures of one cluster model may draw special attention to interesting objects.
- Clusters may serve as models of learning processes in neural systems.

Spectral clustering derives its name from spectral analysis of a graph, which is how the data are represented. Spectral clustering techniques reduce dimensions using the eigenvalues of the similarity matrix of the data. The similarity matrix is provided as an input and consists of a quantitative assessment of the relative similarity of each pair of points in the dataset.

The spectral clustering algorithm is an algorithm for putting N data points in an I-dimensional space into several clusters. Each cluster is parameterized by its similarity, which means that the points in the same group are similar and points in different groups are dissimilar to each other. We start the algorithm by presenting the data points in the form of similarity graph, and then we need to find a partition of the graph so that the points within a group are similar and the points
between different groups are dissimilar to each other. The partition can be done in various ways such as minimum cut method, ratio cut method, and normalized and MinMaxCut Method. This paper will discuss these methods in details and later on introduce the relationship between spectral clustering and k-means clustering and spectral clustering's applications in different areas.

2. Graph Construction

Given a set of data points \( x_1, x_2, \ldots, x_n \) and denote \( s_{ij} \): similarity between data points \( x_i \) and \( x_j \), where \( s_{ij} \geq 0 \). Then we present the data points in the form of a similarity graph \( G=(V,E) \) which is an undirected graph with vertex set \( V = \{v_1, \ldots, v_n\} \). We assume that \( G \) is weighted and each edge between two vertices carries a weight \( w_{ij} > 0 \). And two vertices are connected if the similarity \( s_{ij} \) between the corresponding data points \( x_i \) and \( x_j \) is positive and the edge is weighted by \( s_{ij} \). After we finish the graph representation of data, we want to find the graph partitioning such that the edges between different groups have very low weights and the edges within a group have high weights (Luxburg, 2007).

2.1 Graph notation

In the similarity graph \( G=(V,E) \), we have:

- Vertex set \( V = \{v_1, \ldots, v_n\} \): Each \( v_i \) in this graph represents a data point \( x_i \).
- Weighted adjacency matrix \( W=(w_{ij}) \): We assume that \( G \) is weighted and each edge between two vertices \( v_i \) and \( v_j \) carries a weight \( w_{ij} \geq 0 \). If \( w_{ij} = 0 \), the two vertices are not connected. Otherwise, they are connected. \( w_{ij} \) is shown as the highlighted line in figure 1.
Figure 1 Image including the information on vertex and weight

- Degree $d_i = \sum_{j=1}^{n} w_{ij}$ as shown in figure 2.

Figure 2 Image including the information on vertex and weight

- Degree matrix $D$: the diagonal matrix with the degrees $d_1, \ldots, d_n$ on the diagonal.

- Indicator vector $\mathbb{1}A = (f_1, \ldots, f_n)^t \in \mathbb{R}^n$. Given a subset of vertices $A \subseteq V$. If $v_i \in A$, then $f_i = 1$; otherwise, $f_i = 0$.

- "Size" of the subset $A$:

  $|A| = \text{the number of vertices of } A$

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

- Connected: A subset $A$ of a graph is connected if two vertices in $A$ can be joined by a path such that all intermediate points also lie in $A$.

- Connected components: A subset $A$ is called a connected component if it is connected and if there are no connections between vertices in $A$ and $\bar{A}$. The sets $A_1, \ldots, A_k$ form a partition of
the graph if $A_i \cap A_j = \emptyset$ and $A_1 \cup \ldots \cup A_k = V$. (Liu)

### 2.2 Similarity graphs

There are several constructions to obtain a graph for a set of points, $x_1, \ldots, x_n$, and their distances $d_{jk} \in D$.

- **$\varepsilon$-neighborhood graph**: We connect all points whose distances are smaller than $\varepsilon$.

- **$k$-nearest neighbor graphs**: We connect vertex $v_i$ with vertex $v_j$ if $v_j$ is among the $k$-nearest neighbors of $v_i$. This $k$-nearest neighbor construction leads to a directed graph and there are two ways to make the graph undirected. The first method is to ignore the directions, e.g. connecting $v_i$ and $v_j$ with an undirected edge if $v_i$ is among the $k$-nearest neighbors of $v_j$ or the another way around. The second method is to connect $v_i$ and $v_j$ if $v_i$ is among the $k$-nearest neighbors of $v_j$ as well as $v_j$ is among the $k$-nearest neighbors of $v_i$.

- **Fully connected graph**: This results from connecting all points with positive similarity with each other. (Auffarth, 2007)

### 3. Graph Laplacians and their properties

Graph Laplacian matrices are the main tools for spectral clustering. This section will introduce different graph laplacians and their properties.

#### 3.1 Unnormalized graph Laplacian

The unnormalized graph Laplacian matrix is defined as $L = D - W$. And $L$ has the following properties:
• L is always symmetric and positive semi-definite.

• The multiplicity of 0 as an eigenvalue of L is the number of connected components of G.

• L has n non-negative, real-valued eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, where $\lambda_1$ is called the algebraic connectivity.

• The smallest non-trivial eigenvalue of L is called the spectral gap.

3.2 Normalized graph Laplacian

There are two ways to define the normalized graph Laplacian:

• $L_{\text{sym}} = D^{-\frac{1}{2}}L D^{-\frac{1}{2}} = 1 - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ (symmetric matrix)

• $L_{\text{rw}} = D^{-1}L = I - D^{-1}W$ (random walk)

And their properties are as following:

• $L_{\text{sym}}$ and $L_{\text{rw}}$ are positive semi-definite and have n non-negative, real-valued eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

• The multiplicity k of the eigenvalue 0 of both $L_{\text{sym}}$ and $L_{\text{rw}}$ equals the number of connected components $A_1, \ldots, A_k$ in the graph. (Auffarth, 2007)
4. Graph Partitioning

Clustering essentially means putting data points into several groups so that the points within a
group are similar and points in different groups are dissimilar to each other. For spectral
clustering, once the similarity graph is formed, we need to find the partition of the graph. In this
section, we will discuss several different methods for graph partitioning.

4.1 Minimum cut method

The minimum cut method partitions graph into several sets based on the minimum weight of
edges - that is, the weight of edges connecting vertices in these sets are minimum.

For two disjoint subsets A and B, we define

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

Then we need to solve the mincut problem in the way of choosing $A_1, ..., A_k$ that minimizes

$$\text{cut}(A_1, ..., A_k) = \sum_{i=1}^{k} \text{cut}(A_i, \bar{A}_i)$$

It is easy to solve the problem when $k=2$. However, this method doesn't usually result in an
unsatisfactory partition isolated vertices often become clusters. Two common ways to resolve the
unsatisfactory partition are RatioCut and the normalized cut Ncut and we will discuss these two
methods in detail in the following sections.
4.2 Ratio Cut Method

In RatioCut, the size of a subset $A$ of a graph is measured by its number of vertices $|A|$ and we define $\text{RatioCut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A_i})}{|A_i|}$ (Hagen & Kahng, 1992).

Our goal of using RatioCut is to solve the optimization problem

$$\min \text{RatioCut}(A_1, \ldots, A_k)$$

Say, we want to partition $V$ into $k$ sets $A_1, \ldots, A_k$. We will first define $k$ indicator vectors $h_i = (h_{1,i}, \ldots, h_{n,i})'$ by

$$h_{ij} = \begin{cases} 1/\sqrt{|A_i|} & \text{if } i \in A_j \\ 0 & \text{otherwise} \end{cases}$$

Now the RatioCut function can be rewritten using the unnormalized graph Laplacian due to the following calculation:

$$h_i^tLh_i = \sum_{i,j=1}^{n} w_{ij}(h_i - h_j)^2$$

$$= \sum_{i \in A_j, j \in \overline{A_j}} w_{ij}(\frac{1}{\sqrt{|A_i|}})^2 + \sum_{i \in A_j, j \in A_j} w_{ij}(- \frac{1}{\sqrt{|A_i|}})^2$$

$$= 2\text{cut}(A_i, \overline{A_i}) \times \frac{1}{|A_i|}$$

And

$$h_i^tLh_i = (H'LH)_{ii}$$

Plugging the things above, we get
\[ \text{RatioCut}(A_1, \ldots, A_k) = \frac{1}{2} \sum_{i=1}^{k} h_i' L h_i = \frac{1}{2} \sum_{i=1}^{k} (H' L H)_{ii} = \frac{1}{2} \text{Tr}(H' L H) \]

where Tr means the trace of a matrix. Thus, the problem now is \( \min_{A_1, \ldots, A_k} \text{Tr}(H' L H) \) subject to \( H' H = I \). And here we allow the entries of the matrix \( H \) to take arbitrary real values, which leads to the relaxed optimization problem \( \min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H' L H) \) subject to \( H' H = I \).

Referring to Section 5.2.2(6) of Lutkepohl, the solution is choosing \( H \) as the matrix which contains the first \( k \) eigenvectors of \( L \) as columns (1997). Meanwhile, we need to use the \( k \)-means algorithm on the rows of \( V \). Then we will have the general unnormalized spectral clustering algorithm. (Luxburg, 2006)

### 4.3 The normalized Ncut method

In the normalized Ncut, the size is measure by the weights of its edges \( \text{vol}(A) \) and we define:

\[ \text{Ncut}(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{\text{cut}(A_i, \overline{A_i})}{\text{vol}(A_i)} \]

The algorithm we use to implement the normalized Ncut method is very similar to the ones we used for RatioCut as above. The first step is to define the indicator vectors \( h_i = (h_{i,1}, \ldots, h_{i,n})' \) by

\[ h_{i,j} = \begin{cases} \frac{1}{\sqrt{\text{vol}(A_i)}} & \text{if } i \in A_j \\ 0 & \text{otherwise} \end{cases} \]

Then we set the matrix \( H \) as the matrix containing those \( k \) indicator vectors as columns. Since \( H' H = I, h_i' Dh_i = 1 \), and \( h_i' L h_i = 2 \text{cut}(A_i, \overline{A_i})/\text{vol}(A_i) \), we can rewrite the problem of minimizing Ncut as \( \min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(H' L H) \) subject to \( H' D H = I \).
Relaxing the discreteness condition and substituting \( U = D^{1/2}H \) leads to the standard trace minimization problem

\[
\min_{H \in \mathbb{R}^{n \times k}} \text{Tr}(U'D^{-1/2}L^{-1/2}U) \quad \text{subject to} \quad U'U = I.
\]

This problem can be solve by the matrix \( U \) containing the first \( k \) eigenvectors of \( L_{\text{sym}} \) as columns. According to Shi and Malik, the solution \( H \) consists of the first \( k \) eigenvectors of the matrix \( L_{\text{rw}} \), or the first \( k \) generalized eigenvectors of \( L_{\nu} = \lambda D_{\nu} \). (Luxburg, 2006)

5. **Relationship with kernel K-means algorithm**

K-means clustering is a method of cluster analysis in statistics. It is one of the simplest unsupervised learning algorithms that solve the common problem of clustering. The procedure follows a simple and easy way to classify a given data set through \( k \) clusters, which is a type of cluster that works by having the \( K \) clusters compete with each other for the right to own the data points. The main idea of K-means clustering is to put \( N \) data points into \( K \) clusters. The “\( K \)” in K-means clustering simply refers to the chosen number of clusters.

The major criticism of K-means is that “it is a ‘hard’ rather than ‘soft’ algorithm: points are assigned to exactly one cluster and all points assigned to a cluster are equals in that cluster” (Mackay, 2003, p. 288). In other words, each borderline point is assigned in one cluster, and has an equal contribution with all the other points in that cluster, and no contribution in any other clusters. However, the reality is that points located near the border between two or more clusters should play a partial role in determining the locations of all the clusters that they could be plausibly assigned to.
Two approaches to deal with the problem are kernel K-means and spectral clustering. Kernel k-means maps the data points to a higher-dimensional space using a nonlinear function and then partitions the points by linear separators in the new space. And the spectral clustering uses the eigenvectors of an affinity matrix to obtain a clustering of the data. These two techniques may seem to be totally different, but we will present how they are related by generalizing the k-means function to use both weights and kernels. Specifically, the weighted kernel k-means can be rewritten as a trace maximization problems whose relaxation can be solved with eigenvectors.

As discussed in section 4.3, the normalized Ncut method can be rewritten as a trace minimization problem, where we try to minimize Tr(U'D^{-1/2}LD^{-1/2}U). The minimization of Tr(U'D^{-1/2}LD^{-1/2}U) is equivalent to the maximization of Tr(U'D^{1/2}LD^{1/2}U). So we can see a direct relationship between the trace maximizations of the normalized cut and kernel k-means since the trace maximization of weighted kernel k-means is Tr(Y^T \tilde{D}^{1/2} \tilde{L}^{1/2} \tilde{D}^{1/2} Y), where Y is an n \times k orthogonal matrix (Dhillon, Guan and Kulis). If the affinity matrix W is positive definite, we can use the weighted kernel k-means to minimize the normalized cut.

According to Dhillon, Guan and Kulis, the advantage of the use of an iterative algorithm for the graph problem is that we can use different improvement methods, such as local search, to increase the quality of the results. And it is particularly useful when the affinity matrix is large and sparse, and a large number of eigenvectors need to be calculated.
6. Applications

Clustering has been extensively used in many areas, including in the statistics, machine learning, pattern recognition, data mining, and image processing. In this paper, I’m going to discuss its applications in image segmentation, educational data mining, entity resolution and speech separation.

6.1 Image segmentation

In digital image processing, segmentation is essential for image description and classification. Clusters can be formed for images based on pixel intensity, color, texture, location, or some combination of these. A big problem spectral clustering has in image segmentation is scalability.

“Spectral clustering involves the eigendecomposition of a pairwise similarity matrix, which is intractable for sufficiently large images. Down-sizing the image, however, will cause a loss of finer details and can lead to inaccurate segmentation results” (Tung, Wong, and Clausi, 2010). So Tung et al. (2010) proposed a method of spectral clustering to large images using a combination of blockwise processing and stochastic ensemble consensus.

The idea of this method is to perform an over-segmentation of the image at the pixel level using spectral clustering, and then merge the segments using a combination of stochastic ensemble consensus and a second round of spectral clustering at the segment level. And we use stochastic ensemble consensus o integrate both global and local image characteristics in determining the pixel classifications. This step also removes blockwise processing artifacts. (Tung et al., 2010)

Tung et al. (2010) also presented the experimental results on a set of natural scene images (from the Berkeley segmentation database) of the normalized cut, the self-tuning spectral clustering
and the proposed method as shown in figure 3. They conclude that "the proposed method achieves segmentation results that are comparable to or better than the other two methods. In particular, detailed structures are better preserved in the segmentation, as reflected in the higher recall values" (Tung et al., 2010). As we can see in figure 3, the proposed method results in better performance in fine details such as the distant mountains in image 1, the building outcropping in image 2, the pillars in image 3, the castle contour in image 4, the tree line in image 5, and the boundaries in image 6.

### 6.2 Educational Data Mining

With rapidly increasing data repositories from different educational areas, useful information and data in educational data mining is playing a significant role in student learning since it can answer important research question about student learning. K-means clustering is a simple and effective tool to monitor students’ academic performance by discovering the key characteristics from students’ performance and using these characteristics for future prediction. Furthermore, we are able to improve the student performance prediction by using spectral clustering. Trivedi, Pardos, Sarkozy and Heffernan implemented spectral clustering for analyzing data set of 628 students' state test scores from the 2004-2005 school year and the features included the various dynamic features. The data was collected using the ASSISTments tutor in two schools in Massachusetts and ASSISTments is an Intelligent Tutoring System developed at Worcester Polytechnic Institute, MA, USA. The prediction was the MCAS test scores for the same students in the following year. The technique for making a prediction for a test point includes the following steps and is shown in figure 4:

1. Divide the data into K clusters.
2. Apply a separate linear regression model to each cluster.

3. Each such predictor (such as linear regression) represents a model of the cluster and is called a cluster model. And the collection of cluster models is called a prediction model \( \text{PM}_K \), where \( K \) indicates the number of clusters. (Trivedi et al.)
Figure 3 Segmentation results for natural scene images. Left to right: normalized cut, self-tuning spectral clustering and proposed method.

Figure 4 The technique for making a prediction for a test point (Trivedi et al.)

Trivedi et al. showed the results of clustering using both k-means and spectral clustering as in figure 5. The data points are presented in different colors. And figure 6 below tells us that spectral clustering provides a significant improvement in prediction accuracy over k-means. Furthermore, the spectral clustering ensemble results are not only significant for the static condition (K=1) but also for k-means generated ensemble where K≥3.
Figure 5 The left column is for k-means clustering and the right column is for spectral clustering.

The top row show the ASSISTment data scaled down by multi-dimensional scaling to three dimensions and the clusters identified by both k-means and spectral clustering. The rows below are the different planar views. (Trivedi et al.)
6.3 Entity resolution

In many telecom and web applications, the need of entity resolution is getting bigger and bigger. Entity resolution is to identify whether the objects in the same source represent the same entity in the real-world. This problem arises often in the area of information integration when there lacks a unique identifier across multiple data sources to represent a real-world entity.

Blocking is an important technique for improving the computational efficiency of the algorithms for entity resolution. To solve the entity resolution problem, Shu, Chen, Xiong and Meng proposed an efficient spectral neighborhood (SPAN) algorithm based on spectral clustering. SPAN is an unsupervised and unconstrained algorithm and it is applicable in many applications where the number of blocks is unknown beforehand. (Shu et al.)

SPAN uses the vector space model in the way of representing each record by a vector of qgrams.
A qgram is a length q substring of blocking attribute value. And the algorithm is implemented in the following steps:

1. Define the similarity matrix for the records based on the vector space model.
2. Derive SPAN based on spectral clustering.
3. Use Newman-Girvan modularity as the stopping criterion for blocking.

Shu et al. compared SPAN with three common blocking algorithms, Sorted Neighborhood, Canopy Clustering and Bigram Indexing. The experiments were performed on both published synthetic data and real data and the results indicate:

1. SPAN is fast and scalable to large scale datasets while Canopy Clustering and Bigram Indexing are not.
2. SPAN outperforms the other three when data have low or medium noise.
3. SPAN is much more robust than Canopy Clustering and Bigram Indexing in respect to the tuning parameters because the performance of Canopy Clustering and Bigram Indexing require a large number of labeled data and thus are often not possible with data in the real-world applications. (Shu et al.)
Table 1

Accuracy comparison of the four blocking algorithms

<table>
<thead>
<tr>
<th>Datasets</th>
<th>SPAN prec.</th>
<th>SPAN recall</th>
<th>SPAN F1</th>
<th>CC prec.</th>
<th>CC recall</th>
<th>CC F1</th>
<th>BI prec.</th>
<th>BI recall</th>
<th>BI F1</th>
<th>SN prec.</th>
<th>SN recall</th>
<th>SN F1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<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>H1-500</td>
<td>0.763</td>
<td>0.517</td>
<td>0.617</td>
<td>0.608</td>
<td>0.679</td>
<td>0.634</td>
<td>0.577</td>
<td>0.340</td>
<td>0.428</td>
<td>0.226</td>
<td>0.264</td>
<td>0.243</td>
</tr>
<tr>
<td>H2-500</td>
<td>0.888</td>
<td>0.621</td>
<td>0.731</td>
<td>0.900</td>
<td>0.698</td>
<td>0.786</td>
<td>0.889</td>
<td>0.485</td>
<td>0.628</td>
<td>0.423</td>
<td>0.426</td>
<td>0.425</td>
</tr>
<tr>
<td>M1-500</td>
<td>0.980</td>
<td>0.760</td>
<td>0.862</td>
<td>0.756</td>
<td>0.808</td>
<td>0.781</td>
<td>0.722</td>
<td>0.695</td>
<td>0.708</td>
<td>0.508</td>
<td>0.573</td>
<td>0.539</td>
</tr>
<tr>
<td>M2-500</td>
<td>0.987</td>
<td>0.948</td>
<td>0.967</td>
<td>1.000</td>
<td>0.860</td>
<td>0.925</td>
<td>1.000</td>
<td>0.614</td>
<td>0.761</td>
<td>0.637</td>
<td>0.650</td>
<td>0.643</td>
</tr>
<tr>
<td>L1-500</td>
<td>0.999</td>
<td>0.952</td>
<td>0.975</td>
<td>0.988</td>
<td>0.961</td>
<td>0.974</td>
<td>0.862</td>
<td>0.864</td>
<td>0.863</td>
<td>0.570</td>
<td>0.680</td>
<td>0.620</td>
</tr>
<tr>
<td>L2-500</td>
<td>1.000</td>
<td>0.956</td>
<td>0.977</td>
<td>0.973</td>
<td>0.955</td>
<td>0.964</td>
<td>0.930</td>
<td>0.619</td>
<td>0.743</td>
<td>0.660</td>
<td>0.679</td>
<td>0.669</td>
</tr>
<tr>
<td>H1-1500</td>
<td>0.598</td>
<td>0.355</td>
<td>0.446</td>
<td>0.615</td>
<td>0.506</td>
<td>0.555</td>
<td>0.299</td>
<td>0.347</td>
<td>0.322</td>
<td>0.189</td>
<td>0.192</td>
<td>0.190</td>
</tr>
<tr>
<td>H2-1500</td>
<td>0.908</td>
<td>0.531</td>
<td>0.670</td>
<td>0.628</td>
<td>0.716</td>
<td>0.669</td>
<td>0.577</td>
<td>0.478</td>
<td>0.523</td>
<td>0.382</td>
<td>0.385</td>
<td>0.384</td>
</tr>
<tr>
<td>M1-1500</td>
<td>0.936</td>
<td>0.671</td>
<td>0.781</td>
<td>1.000</td>
<td>0.560</td>
<td>0.718</td>
<td>0.611</td>
<td>0.518</td>
<td>0.560</td>
<td>0.497</td>
<td>0.538</td>
<td>0.516</td>
</tr>
<tr>
<td>M2-1500</td>
<td>0.971</td>
<td>0.911</td>
<td>0.940</td>
<td>0.958</td>
<td>0.842</td>
<td>0.896</td>
<td>0.814</td>
<td>0.626</td>
<td>0.708</td>
<td>0.634</td>
<td>0.671</td>
<td>0.625</td>
</tr>
<tr>
<td>L1-1500</td>
<td>0.987</td>
<td>0.882</td>
<td>0.931</td>
<td>0.882</td>
<td>0.912</td>
<td>0.896</td>
<td>0.787</td>
<td>0.659</td>
<td>0.717</td>
<td>0.574</td>
<td>0.607</td>
<td>0.590</td>
</tr>
<tr>
<td>L2-1500</td>
<td>0.998</td>
<td>0.939</td>
<td>0.968</td>
<td>0.959</td>
<td>0.913</td>
<td>0.936</td>
<td>0.819</td>
<td>0.693</td>
<td>0.750</td>
<td>0.647</td>
<td>0.666</td>
<td>0.657</td>
</tr>
<tr>
<td>H1-5000</td>
<td>0.352</td>
<td>0.168</td>
<td>0.227</td>
<td>0.556</td>
<td>0.297</td>
<td>0.387</td>
<td>0.302</td>
<td>0.197</td>
<td>0.238</td>
<td>0.155</td>
<td>0.154</td>
<td>0.154</td>
</tr>
<tr>
<td>H2-5000</td>
<td>0.872</td>
<td>0.409</td>
<td>0.557</td>
<td>0.902</td>
<td>0.319</td>
<td>0.472</td>
<td>0.496</td>
<td>0.354</td>
<td>0.413</td>
<td>0.356</td>
<td>0.352</td>
<td>0.354</td>
</tr>
<tr>
<td>M1-5000</td>
<td>0.907</td>
<td>0.643</td>
<td>0.753</td>
<td>0.970</td>
<td>0.546</td>
<td>0.699</td>
<td>0.666</td>
<td>0.383</td>
<td>0.486</td>
<td>0.480</td>
<td>0.496</td>
<td>0.488</td>
</tr>
<tr>
<td>M2-5000</td>
<td>0.956</td>
<td>0.887</td>
<td>0.920</td>
<td>0.938</td>
<td>0.843</td>
<td>0.888</td>
<td>0.572</td>
<td>0.574</td>
<td>0.573</td>
<td>0.620</td>
<td>0.610</td>
<td>0.615</td>
</tr>
<tr>
<td>L1-5000</td>
<td>0.986</td>
<td>0.764</td>
<td>0.861</td>
<td>0.962</td>
<td>0.797</td>
<td>0.872</td>
<td>0.645</td>
<td>0.515</td>
<td>0.573</td>
<td>0.568</td>
<td>0.589</td>
<td>0.578</td>
</tr>
<tr>
<td>L2-5000</td>
<td>0.989</td>
<td>0.908</td>
<td>0.947</td>
<td>0.923</td>
<td>0.921</td>
<td>0.922</td>
<td>0.754</td>
<td>0.560</td>
<td>0.643</td>
<td>0.648</td>
<td>0.651</td>
<td>0.650</td>
</tr>
</tbody>
</table>

As shown in table 1, SPAN and Canopy Clustering significantly outperform Bigram Indexing and Sorted Neighborhood in terms of precision, recall and F1-measure, where precision is the proportion of correctly identified record pairs, recall is the proportion of correctly identified record pairs to the correct record pairs based on the ground-truth entities, and F1-measure is the harmonic average of precision and recall (Shu et al., 2010). Furthermore, we can easily tell that SPAN is the best algorithm among all these four based on the results shown in table 1.
Table 2

Time cost (seconds) of SPAN and Canopy Clustering on a big dataset

<table>
<thead>
<tr>
<th>No. of records</th>
<th>SPAN</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>10.58</td>
<td>11.37</td>
</tr>
<tr>
<td>5000</td>
<td>34.02</td>
<td>278.31</td>
</tr>
<tr>
<td>10000</td>
<td>50.11</td>
<td>3095.76</td>
</tr>
<tr>
<td>50000</td>
<td>259.57</td>
<td>*</td>
</tr>
<tr>
<td>100000</td>
<td>811.65</td>
<td>*</td>
</tr>
<tr>
<td>500000</td>
<td>3485.21</td>
<td>*</td>
</tr>
</tbody>
</table>

Since SPAN and CC are the two algorithms stand out in the comparison above, Shu et al. (2010) compared these two specifically with respect to time cost. The results in table 2 indicate that SPAN performs a lot better than CC in terms of time cost. Therefore, they concluded that SPAN is the best algorithm for entity resolution among these four.

6.4 Speech separation

While linkage algorithms and k-means algorithms are very popular in speech processing and robust to noise, they are only best suited for rounded linearly separable clusters. However, spectral clustering is able to find elongated clusters and is more robust to noise than the above two algorithms.

Bach and Jordan applied spectral clustering to data from four different male and female speakers with speech signals of duration 3 seconds based on a cost function that characterized how close the eigenstructure of a similarity matrix $W$ is to a partition $E$. According to Bach and Jordan, "minimizing this cost function with respect to the partition $E$ leads to a new clustering algorithm that takes the form of weighted k-means algorithms. Minimizing them with respect to $W$ yields a
Theoretical framework for learning the similarity matrix. The basic idea of their algorithm is to combine the knowledge of physical and psychophysical properties of speech with learning algorithms. The physical properties provide parameterized similarity matrices for spectral clustering and the psychophysical properties help generate segmented training data.

There were 15 parameters to estimate using Bach and Jordan's spectral learning algorithm. For testing, they used mixes from speakers which were different from those in the training set (the four different male and female speakers with speech signals of duration 3 seconds). Top image in Figure 7 shows the segmentation for two English speakers from the testing set when the two speech signals are known and the bottom image shows the segmentation by their algorithm.

Bach and Jordan's analyzed that the performance of the separation is good enough to obtain audible signals of reasonable quality even though some components of the "black" speaker are missing. Similarly, they also presented the segmentation results for French speakers in Figure 8 and pointed that their method "does not require knowing the speakers in advance in order to demix successfully; rather, it is only necessary that the two speakers have distinct pitches most of the time (another but less crucial condition is that on pitch is not too close to twice the other one)" (Bach & Jordan). As we can seen from the results, the proposed approach was successful in demixing the speech signals from two speakers.
Figure 7  Top image for the spectrogram of English speakers, colored in black and grey; Bottom image for the blind segmentation obtained with proposed method (Bach & Jordan).
Figure 8  Top image for the spectrogram of English speakers, colored in black and grey; Bottom image for the blind segmentation obtained with proposed method (Bach & Jordan).
7 Conclusion

Clustering is one of the most popular techniques used in many applications such as statistics, machine learning, pattern recognition, data mining, and image processing. And spectral clustering has become more and more widely used since it is a simple method in cluster analysis and often outperforms traditional clustering algorithms like k-means.

In this paper, I have introduced the theoretical background and procedures of spectral clustering. In short, it makes use of eigenvalues of the similarity matrix of the data to perform dimensionality reduction before clustering in fewer dimensions and then we find a partition of the graph so that the points within a group are similar and the points between different groups are dissimilar to each other. This partition can be done in various ways. This paper also presented the connection between weighted kernel k-means and spectral clustering and spectral clustering's applications in different areas such as image segmentation, educational data mining, entity resolution and speech separation.
References


