CS724: Topics in Algorithms
Spectral Clustering

Prof. Dan A. Simovici
We present a treatment of clusterings starting from a finite similarity space $S = (V, s)$ defined on the set of objects $V$.

**Definition**

The *similarity graph* associated to $S$ is the weighted graph $G_S = (V, E, s)$, where $E = \{(v, v') \in V \times V \mid s(v, v') > 0\}$ and the weight of an edge $(v, v')$ is $s(v, v')$ for $v, v' \in V$. 
A clustering of the objects in $V$ is a partition $\kappa = \{C_1, \ldots, C_n\}$ of $V$. The blocks $C_i$ of $\kappa$ are the clusters.

In terms of similarity spaces, the goal of any clustering algorithm is to gather in a cluster all objects that are similar to each other and to place in distinct clusters pairs of objects that have low similarities.
There are several ways to construct a similarity space (or a similarity graph) for a set of points $V = \{x_1, \ldots, x_n\}$.

For example, an undirected graph $G_t = (V, E_t)$ can be defined by

$$E_t = \{\{x_i, x_j\} \mid d(x_i, x_j) \leq t\},$$

where $t$ is a given threshold.
Another option is to use the $k$-nearest neighbor graph $G_{nn,k}$, where an edge $(v, w)$ exists if $w$ is among the $k$ nearest neighbors of $v$. This leads, of course to a directed graph; however, an undirected graph can be readily obtained by ignoring the orientation of the edges. An alternative undirected graph $G'_{nn,k}$ can be obtained by considering an edge $\{v, w\}$ if $w$ is among the $k$ closest neighbors of $v$ and $v$ is among the $k$ closest of $w$. 
Finally, it is possible to use a weighted complete graph on the set $V$ and define for each pair of objects a similarity measure $k(v, w)$. The function $k$ is referred in the specialized $\textbf{R}$ package kernlab as a \textit{kernel}.

A \textit{radial basis function} (rbf) is a real-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ whose value $f(x)$ depends only on the distance from the origin $\|x\|$, that is, $f(x) = f(\|x\|)$.

Examples of such kernels are $k(v, w) = e^{-\|v-w\|^2}$ named the \texttt{rbfdot} or $e^{-\|v-w\|}$ named the \texttt{laplacedot}, etc.
As usual, a clustering of the objects in $V = \{x_1, \ldots, x_n\}$, where $V \subseteq \mathbb{R}^m$ is a partition $\kappa = \{C_1, \ldots, C_k\}$ of $V$. The blocks $C_i$ of $\kappa$ are the clusters.

Let $\kappa = \{C_1, \ldots, C_k\}$ be a clustering of the objects of the set $V = \{v_1, \ldots, v_n\}$. For a block $C_i$ we denote by $\overline{C}_i$ the complement of $C_i$ relative the set $V$. Clearly, $\overline{C}_i = \bigcup\{C_j \mid j \in \{1, \ldots, k\} - \{i\}\}$ for every $i$, $1 \leq i \leq k$.

**Definition**

The *cut* of $\kappa$ is the number

$$
cut(\kappa) = \sum_{i=1}^{k} \cut(C_i, \overline{C}_i)
$$

$$
= \sum_{p=1}^{n} \sum_{q=1}^{n} \left\{ s(v_p, v_q) \mid v_p \text{ and } v_q \text{ belong to different clusters} \right\}
$$
Definition

Let $G = (V, E)$ be a graph and let $S$ be a set of vertices. The edge boundary of $S$ is the set of edges of $G$ that join $S$ to its complement. This set is denoted by $\partial(S)$. Clearly, $\partial(V - S) = \partial(S)$. 
Theorem

Let $G = (V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$ and let $S$ be a subset of $V$. Then

$$\alpha(G) \leq \frac{n|\partial(S)|}{|S|(n - |S|)},$$

where $\partial(S)$ is the edge boundary of the set $S$. 
Proof

Recall that we have shown that

\[ \alpha(G) = \min_x \{ x' L_G x \mid x \in S_n \} \]

\[ = \min_x \sum \{(x_i - x_j)^2 \mid x \in S_n, i < j \text{ and } \{v_i, v_j\} \in E\}, \]

where \( S_n = \{ x \in \mathbb{R}^n \mid x' 1_n = 0 \text{ and } \| x \| = 1 \} \).
Proof cont’d

Let \( r \in \mathbb{R}^n \) be a vector defined by

\[
 r_i = \begin{cases} 
 n - |S| & \text{if } v_i \in S, \\ 
 -|S| & \text{if } v_i \notin S,
\end{cases}
\]

for \( 1 \leq i \leq n \).

It is clear that \( r'1_n = 0 \), that is, \( r \) is orthogonal on \( 1_n \). Therefore, we have:

\[
\alpha(G) \leq \frac{\sum_{(v_i, v_j) \in E} (r_i - r_j)^2}{\|r\|^2} = \frac{n^2|\partial(S)|}{|S|(n - |S|)^2 + (n - |S|)|S|^2} = \frac{n|\partial(S)|}{|S|(n - |S|)}.
\]
Definition

Let $G = (V, E)$ be a graph. The conductance of $G$ is the number

$$cd(G) = \min \left\{ \frac{\partial(S)}{|S|} \mid S \subseteq V, |S| \leq \frac{|V|}{2} \right\}.$$
Example

To compute the conductance of a complete graph $\mathcal{K}_n$ note that each vertex $v$ is linked to $n - 1$ other vertices of the graph. Thus, for a set of vertices $S$ we have $\partial(S) = S \times (V - S)$, so $|\partial(S)| = |S|(n - |S|)$. Thus,

$$cd(\mathcal{K}_n) = \min \left\{ n - |S| \mid S \subseteq V, |S| \leq \frac{n}{2} \right\} = \left\lceil \frac{n}{2} \right\rceil.$$
Example

Let $G = (V, E)$ be a graph such that $|V| = n$. If $|S| \leq \frac{n}{2}$, then $\text{cd}(G) \leq \frac{\partial(S)}{|S|}$, so $|\partial(S)| \geq \text{cd}(G)|S|$. If $\text{cd}(G)$ is large, then the vertices of $S$ have many neighbors outside $S$.

Suppose that $\{G_n = (V_n, E_n) \mid n \in \mathbb{N}\}$ be a sequence of graphs with $|V_n| = n$ such that each graph $G_n$ is $k$-regular and the there exists a lower bound $b$ of the sequence $\{\text{cd}(G_n) \mid n \in \mathbb{N}\}$. We refer to such a sequence of graphs as an expander. Note that $|E_n| = \frac{kn}{2}$, which means that the graphs grow increasingly sparse.

The existence of a lower bound for conductances guarantees that there exist many neighbors for a set $S$ of vertices.
Theorem

Let $G = (V, E)$ be a graph. We have

$$cd(G) \geq \frac{\alpha(G)}{2}.$$
Proof

In the definition of the conductance we require $|S| \leq \frac{|V|}{2}$ so

$$\frac{|V|}{|V| - |S|} = \frac{1}{1 - \frac{|S|}{|V|}} \leq 2.$$ 

Therefore, since $\alpha(G) \leq \frac{n|\partial(S)|}{|S|(n - |S|)}$, it follows that

$$\alpha(G) \leq \frac{n|\partial(S)|}{|S|(n - |S|)} = \frac{|\partial(S)|}{|S|} \cdot \frac{n}{n - |S|} \leq 2cdG.$$ 

This theorem shows that $\frac{\alpha(G)}{2}$ provides a lower bound for the conductance of a graph, whose computation is intractable.
There are several criteria for choosing clusterings defined on similarity spaces. The simplest such criterion is the minimal value of $\text{cut}(\kappa)$. This will ensure that the objects of each cluster $C_i$ are as dissimilar as possible to the objects from the other clusters.

For bipartitions the algorithm is based on the observation that if $x, y$ are two vertices of a weighted graph $G = (V, E, w)$ and $\pi = \{X, Y\}$ is a bipartition of $V$ such that $x \in X$ and $y \in Y$, then the value of a minimum cut $\text{cut}(\pi)$ is the smaller of a minimum $(x, y)$-cut and a minimum cut of $G/\{x, y\}$, where $G/\{x, y\}$ is the graph obtained from $G$ by merging $x$ and $y$ and removing the edge $(x, y)$ if such an edge exists. Indeed, either there exists a minimum cut of $G$ that separates $x$ and $y$ (and in this case a minimum $(x, y)$-cut is a minimum cut of $G$), or there is no such cut and, in this case, a minimum cut of $G/\{x, y\}$ is a minimum cut of $\pi$. 

Prof. Dan A. Simovici
CS724: Topics in Algorithms Spectral Clust
Let $\kappa = \{C_1, \ldots, C_k\}$ be a partition of a set $V = \{v_1, \ldots, v_n\}$ of $n$ objects into $k$ clusters.

Starting from a similarity matrix $S \in \mathbb{R}^{n \times n}$ for the objects of $V$ we can define a *similarity graph* of $V$ as $G = (V, E, s)$, where $s(v_i, v_\ell) = s_{i\ell}$ for $1 \leq i, \ell \leq n$.

The *indicator vector* $c_j \in \mathbb{R}^n$ of the cluster $C_j$ is

$$
(c_j)_i = \begin{cases} 
\frac{1}{\sqrt{|C_j|}} & \text{if } v_i \in C_j \\
0 & \text{otherwise},
\end{cases}
$$

where $1 \leq i \leq n$ and $1 \leq j \leq i$. 
Since $\kappa$ is a partition of $V$, the matrix $C = (c_1 \cdots c_k)$ has an orthonormal set of columns, so $C'C = I_k$. Note that, in terms of the entries of $C$ we have

$$c_j = \begin{pmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{ij} \\ \vdots \\ c_{nj} \end{pmatrix}$$

for $1 \leq j \leq k$.

We claim that

$$c_j' L_G c_j = \frac{1}{2} \frac{cut(C_j, \bar{C}_j)}{|C_j|}$$
By a previous result we have:

\[ \mathbf{c}'_j L_G \mathbf{c}_j = \frac{1}{2} \sum_{i=1}^{k} \sum_{\ell=1}^{k} s_{i\ell} (c_{ij} - c_{\ell j})^2. \]

If \( v_i \in C_j \) and \( v_\ell \notin C_j \) we have \( c_{ij} = \frac{1}{\sqrt{|C_j|}} \) and \( c_{\ell j} = 0 \); in this case

\[ s_{i\ell} (c_{ij} - c_{\ell j})^2 = \frac{s_{ij}}{|C_j|}. \]

Otherwise, that is, if both \( v_i \) and \( v_\ell \) belong to \( C_j \), or neither vertex belongs to \( C_j \) we have

\[ s_{i\ell} (c_{ij} - c_{\ell j})^2 = 0. \]

This implies

\[ \mathbf{c}'_j L_G \mathbf{c}_j = \sum_{v_i \in C_j} \sum_{v_\ell \notin C_j} \frac{s_{ij}}{|C_j|} = \frac{1}{2} \frac{cut(C_j, \bar{C}_j)}{|C_j|} \]

and

\[ \mathbf{c}'_j L_G \mathbf{c}_j = (C' L_G C)_{jj}. \]
Since
\[ C' L_G C = \begin{pmatrix} c'_1 \\ \vdots \\ c'_k \end{pmatrix} L_G (c_1 \cdots c_k), \]
we have
\[
\sum_{j=1}^{k} c'_j L_G c_j = \sum_{j=1}^{k} (C' L_G C)_{jj} \\
= \text{trace}(C' L_G C) = \frac{1}{2} \sum_{j=1}^{k} \frac{\text{cut}(C_j, \tilde{C}_j)}{|C_j|} = \frac{1}{2} \text{cutratio}(\kappa).
\]

To minimize \text{cutratio}(\kappa) is tantamount to seeking the matrix \( C \) such that \( \text{trace}(C' L_G C) \) is minimal subjected to the constraint \( C' C = I_k \).

To obtain a practical solution this optimization problem is relaxed by allowing \( C \) to range over \( \mathbb{R}^{n \times k} \). By Ky Fan’s Theorem the minimum is obtained by choosing \( C \) such that its columns consist of the eigenvectors \( c_1, \ldots, c_k \) of \( L_G \) that correspond to the \( k \) smallest eigenvalues of the Laplacian \( L_G \).
The original set of points \( V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^m \) is transformed now into a set \( \{y_1, \ldots, y_n\} \) in a lower-dimensional space \( \mathbb{R}^k \), where \( y'_1, \ldots, y'_n \) are the rows of the matrix \( C \in \mathbb{R}^{k \times m} \) whose columns are the \( k \) eigenvectors \( c_1, \ldots, c_k \) of \( L_G \), as shown next.
Unnormalized Spectral Clustering

Data: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number $k$ of clusters

Result: A clustering $\kappa = \{C_1, \ldots, C_k\}$

- let $A$ be its weighted adjacency matrix;
- compute the ordinary Laplacian $L$;
- compute the first $k$ eigenvectors $c_1, \ldots, c_k$ of $L$;
- let $C = (c_1, \ldots, c_k) \in \mathbb{R}^{n \times k}$;
- define $y_1, \ldots, y_n \in \mathbb{R}^k$ such that $C' = (y_1 \ldots y_n)$;
- cluster $\{y_1, \ldots, y_n\} \subseteq \mathbb{R}^k$ using the $k$-means algorithm into $\kappa$;
Another approach to spectral clustering uses the normalized cut of a partition. As before, let $\kappa = \{C_1, \ldots, C_k\}$ be a partition of a set $V = \{v_1, \ldots, v_n\}$ of $n$ objects into $k$ clusters for which we have a similarity matrix $S \in \mathbb{R}^{n \times n}$. Define the characteristic vector $h_j$ of $C_j$ as

$$
(h_j)_i = \begin{cases} 
\frac{1}{\sqrt{\text{vol}(C_j)}} & \text{if } v_i \in C_j, \\
0 & \text{otherwise},
\end{cases}
$$

for $1 \leq j \leq k$ and let $H = (h_1 \cdots h_k)$ be the matrix of these vectors. We have

$$
\mathbf{h}_j^D G \mathbf{h}_j = \sum_{i=1}^{n} \sum_{\ell=1}^{n} (h_j)_i d_{i\ell} (h_j)_\ell.
$$
The non-zero terms in this sum are such that \( i = \ell \) and \( v_i \in C_j \). Thus, 
\[ h_j' D_G h_j = \frac{1}{\text{vol}(C_j)} \sum_{v \in C_j} d(v) = 1. \]
On the other hand, \( h_j' D_G h_m = 0 \) if \( j \neq m \), so \( H' D_G H = I_k \). A similar computation yields

\[
h_j' A_G h_j = \sum_{i=1}^{n} \sum_{\ell=1}^{n} (h_j)_i s_{i\ell}(h_j)_\ell = \frac{1}{\text{vol}(C_j)} \sum_{v_i, v_\ell \in C_j} s(v_i, v_\ell).
\]

These computations allow us to write

\[
h_j' L_G h_j = h_j'(D_G - A_G) h_j = I_k - h_j' A_G h_j = 1 - \frac{1}{\text{vol}(C_j)} \sum_{v_i, v_\ell \in C_j} s(v_i, v_\ell)
\]

\[
= \frac{\text{vol}(C_j) - \sum_{v_i, v_\ell \in C_j} s(v_i, v_\ell)}{\text{vol}(C_j)} = \frac{\text{cut}(C_j, \bar{C}_j)}{\text{vol}(C_j)}.
\]

Therefore,

\[
\text{trace}(H' L_G H) = \sum_{j=1}^{k} h_j' L_G h_j = \sum_{j=1}^{k} \frac{\text{cut}(C_j, \bar{C}_j)}{\text{vol}(C_j)} = \text{ncut}(\kappa).
\]
To minimize the normalized cut we need to minimize $\text{trace}(H'L_GH)$ subjected to the constraint $H'DH = I_k$. Let $M = D^{\frac{1}{2}}H$. Then, in terms of the matrix $M$, the optimization problem amounts to minimizing $\text{trace}(M'D^{-\frac{1}{2}}L_GD^{-\frac{1}{2}}M) = \text{trace}(M'L_{G,\text{sym}}M)$ subjected to the restriction $M'M = I_k$. By allowing $M$ to range over $\mathbb{R}^{n \times k}$, the optimum can be achieved by $M = (m_1, \ldots, m_k)$, where $m_1, \ldots, m_k$ are the first $k$ eigenvectors of the symmetric Laplacian $L_{G,\text{sym}}$. 
$D^{-\frac{1}{2}}m_1, \ldots, D^{-\frac{1}{2}}m_k$ are the first $k$ eigenvectors of the random walk Laplacian and these are exactly the columns of the matrix $H$. So, the optimal value of $H$ is obtained by choosing its columns to be equal to the eigenvectors that correspond to the first $k$ eigenvalues of $L_{G,\text{rw}}$. 
Next we discuss the implementation of spectral clustering in \( \mathbb{R} \).
We consider a set of 41 points in \( \mathbb{R}^2 \) placed into two squares and encoded as pairs of numbers in the matrix \( X \).

\[
X <- \text{matrix}(c(1,1,1,2,1,3,1,4,1,5,1,6,1,7, \\
11,1,11,2,11,3,11,4,11,5,11,6,11,7, \\
2,1,3,1,4,1,5,1,6,1,7,1,8,1,9,1,10,1, \\
2,7,3,7,4,7,5,7,6,7,7,7,8,7,9,7,10,7, \\
4,3.5,4,4,4,4.5, \\
5,3.5,5,4,5,4.5, \\
6,3.5,6,4,6,4.5), \\
nrow = 41, byrow=TRUE)
\]
The set of points in $\mathbb{R}^2$ looks like:
The function `neighbor_graph(X, k)` is used for building a $k$-nearest neighbor graph $G_{nn,k}$, where an edge $(v, w)$ exists if $w$ is among the $k$ nearest neighbors of $v$. The adjacency matrix $K$ of this graph is symmetrized (using the operation $K <- K + t(K)$ to yield the symmetric adjacency matrix of an undirected graph.

```r
neighbor_graph <- function(X, k) {
  D <- as.matrix(dist(X))
  K <- matrix(0, nrow = nrow(X), ncol = nrow(X))
  for(i in 1:nrow(X)) {
    neighbor_index <- order(D[i,])[2:k]
    K[i,][neighbor_index] <- 1
  }
  # K is a matrix having 1s in position (i,j) if j is among
  # the first k neighbors of i
  K <- K + t(K)
  K[K == 2] = 1
  return(K)
}
```
The function `spectral_clustering` makes use of the function `laplacian` and the function `neighbor_graph` defined above. The R script of this function is given next.

```r
spectral_clustering <- function(X,k,num_eig) {
  G = neighbor_graph(X,k)
  L = laplacian(G,FALSE)
  eig = eigen(L,symmetric=TRUE)
  n = nrow(L)
  return(eig$vectors[,,(n - num_eig):(n-1)])
  # this returns the eigenvectors of the num_eig smallest eigenvalues
}
```
Finally, the set of eigenvectors returned by `spectral_clustering` is clustered using the $k$-means function as in:

```r
X_sc <- spectral_clustering(X,k,num_eig)
X_final <- kmeans(X_sc,num_clust)
```
A direct application of the function `specc` of the package `kernlab`

```r
sc <- specc(X, centers=2, kernel=’’rbfdot’’)
```

followed by a call to the `pdf` function

```r
> pdf(’’squares.pdf’’)
> plot(X,pch=sc+22)
> dev.off()
```

will produce the plot shown next.