Notes on the SDP relaxation of k-means

The SDP relaxation of k-means

Iguchi et al.[1] formulated a semidefinite progmming approach to k-means as follows[2]:

$$\begin{aligned} \arg\max_{Z} \; & -\mathrm{Tr}(D_2 Z) \\ \text{s.t. } & \mathrm{Tr}(Z) = k \\ & Ze = e \\ & Z \geq 0 \text{ element-wise} \\ & Z \text{ is positive semidefinite} \end{aligned}$$

Where

- $D_2 = [d_{ij}] = [||x_i - x_j||^2]$
- $x_1, ..., x_n \in \mathbb{R}^q$
- The number of clusters, k, is known
- $e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^q$

Note that without the SDP relaxation, we have a rigid structure for Z where $z_{ij} = \begin{cases} n_k^{-1} & x_i, x_j \text{ in same cluster } k \\ 0 & \text{else} \end{cases}$

Equating the trace formulation of k-means to kernel k-means

We can see that the data matrix $X = \begin{bmatrix} x_1^\top \\ \vdots \\ x_n^\top \end{bmatrix}$ is not explicitly in the objective, although squared Euclidean distances are. We can rewrite this as a kernel formulation by noting that $D_2 = \kappa(B)$ where B is a kernel matrix[3]:

[1] https://arxiv.org/abs/1505.04778

[2] the notation is slightly different here

[3] In the following steps we use the fact that Z is symmetric, $Ze = e$, $e^\top Z = e^\top$, and $e^\top e = n$

$$
\begin{aligned}
-\mathrm{Tr}(D_2 Z) &= -\mathrm{Tr}(\kappa(B)Z) \\
&= -\mathrm{Tr}((be^\top - 2B + eb^\top)Z) \\
&= 2\mathrm{Tr}(BZ) - \mathrm{Tr}(be^\top Z) - \mathrm{Tr}(eb^\top Z) \\
&= 2\mathrm{Tr}(BZ) - \mathrm{Tr}(be^\top) - \mathrm{Tr}(Zeb^\top) \\
&= 2\mathrm{Tr}(BZ) - \mathrm{Tr}(be^\top) - \mathrm{Tr}(eb^\top) \\
&= 2\mathrm{Tr}(BZ) - 2\mathrm{Tr}(be^\top) \\
&= 2\mathrm{Tr}(BZ) - 2\mathrm{Tr}(B)
\end{aligned}
$$

... where $b = \mathrm{diag}(B)$, the vector of diagonal entries of B. Note that if we think of B as a weight matrix for an undirected graph, $\mathrm{Tr}(B) = 0$. Similarly, if we impose that the diagonal entries of B are equal to 1 (e.g., B is a correlation matrix), then $\mathrm{diag}(B) = n$. Either way, $\mathrm{Tr}(B)$ does not depend on Z, so we can ignore it in the objective, and we can see that $-\arg\max_Z \mathrm{Tr}(D_2 Z) = \arg\max_Z \mathrm{Tr}(BZ)$, which is just the typical kernel formulation of k-means:

$$
\begin{aligned}
&\arg\max_Z \mathrm{Tr}(BZ) \\
&\quad \text{s.t. } \mathrm{Tr}(Z) = k \\
&\qquad z_{ij} = \begin{cases} n_k^{-1} & x_i, x_j \text{ in same cluster } k \\ 0 & \text{else} \end{cases}
\end{aligned}
$$

Similarly, we can go from a kernel formulation of k-means to one based on squared Euclidean distances by noting that $D_2 = \tau(B)$. For simplicity of notation, we will rewrite $\arg\max_x f(x) = \arg\max_x 2f(x)$.

$$
\begin{aligned}
2\mathrm{Tr}(BZ) &= 2\mathrm{Tr}(\tau(D_2)Z) \\
&= \mathrm{Tr}(-PD_2PZ) \\
&= -\mathrm{Tr}((I - n^{-1}ee^\top)D_2(I - n^{-1}ee^\top)Z) \\
&= -\mathrm{Tr}((D_2 - n^{-1}D_2ee^\top - n^{-1}ee^\top D_2 + n^{-2}ee^\top ee^\top D_2)Z) \\
&= -\mathrm{Tr}(D_2Z) + n^{-1}\mathrm{Tr}(D_2ee^\top Z) + n^{-1}\mathrm{Tr}(ee^\top D_2 Z) - n^{-1}\mathrm{Tr}(ee^\top D_2 Z) \\
&= -\mathrm{Tr}(D_2Z) + 2n^{-1}\mathrm{Tr}(D_2ee^\top) - n^{-1}\mathrm{Tr}(D_2ee^\top) \\
&= -\mathrm{Tr}(D_2Z) + n^{-1}\mathrm{Tr}(D_2ee^\top)
\end{aligned}
$$

Since the second and third terms do not depend on Z, we can discard them, and we get $\arg\max_Z \mathrm{Tr}(BZ) = \arg\max_Z -\mathrm{Tr}(D_2Z)$.[4]

Equating the SDP relaxation of k-means to the SDP relaxation of ratio cut

The ratio cut objective is:

$$
\arg\min_Z \mathrm{Tr}(LZ)
$$

[4] We can rewrite $\mathrm{Tr}(D_2ee^\top) = \sum_{i,j} d_{ij}^2 = 2\sum_{i<j} d_{ij}^2$

where L is the combinatorial graph Laplacian and Z has the same structure as before. If we relax the optimization problem by not enforcing Z to have this structure, we can see that:

$$\arg\min_Z \text{Tr}(LZ) = \arg\max_Z \text{Tr}(L^\dagger Z)$$

where $L^\dagger$ is the generalized inverse of L. Since $L^\dagger$ is positive semidefinite, it can be thought of as a kernel matrix, and we can apply the $\tau(\cdot)$ transformation to it to obtain D_2. In this case, D_2 is the expected commute time of the graph that generated L.

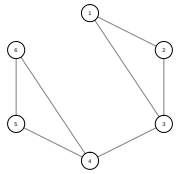
The argmin and argmax equivalence is not true in general if we force Z to have the structure that we want. It also is not true if we apply the SDP constraints (namely $Z \geq 0$ element-wise). One question of interest is under what conditions can we equate the two objectives under the SDP constraints.

Examples

Example 1

Here we look at a case where $\arg\min_Z \text{Tr}(LZ) = \arg\max_Z \text{Tr}(L^\dagger Z)$ under the SDP restrictions ($Ze = e$, $\text{Tr}(Z) = k$, Z is positive semidefinite, $Z \geq 0$ element-wise). In fact, in this example, not only do the two problems have the same solution, the solution coincides with the solution to the unrelaxed ratio cut problem.

Here we have a very simple graph with just six vertices. The "intuitive cut" for $k = 2$ is the $3 - 4$ cut, which happens to also be the solution to the (fully constrained) ratio cut problem.



The RatioCut-SDP solution (Ling and Strohmer)[5] yields:

```
rc.sdp(L, k = 2)$Z %>%
  MASS::fractions()
```

```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1/3  1/3  1/3    0    0    0
[2,] 1/3  1/3  1/3    0    0    0
[3,] 1/3  1/3  1/3    0    0    0
[4,]   0    0    0  1/3  1/3  1/3
[5,]   0    0    0  1/3  1/3  1/3
[6,]   0    0    0  1/3  1/3  1/3
```

The k-means SDP solution using $L^\dagger$ as a kernel matrix (adapted from Peng and Wei)[6] yields:

```
kmeans.sdp(L.dagger, k = 2)$Z %>%
  MASS::fractions()
```

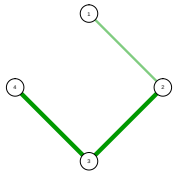
```
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 1/3  1/3  1/3    0    0    0
[2,] 1/3  1/3  1/3    0    0    0
[3,] 1/3  1/3  1/3    0    0    0
```

[5] https://arxiv.org/abs/1806.11429
[6] http://www.optimization-online.org/DB_FILE/2005/04/1114.pdf

```
[4,]    0    0    0  1/3  1/3  1/3
[5,]    0    0    0  1/3  1/3  1/3
[6,]    0    0    0  1/3  1/3  1/3
```

Example 2

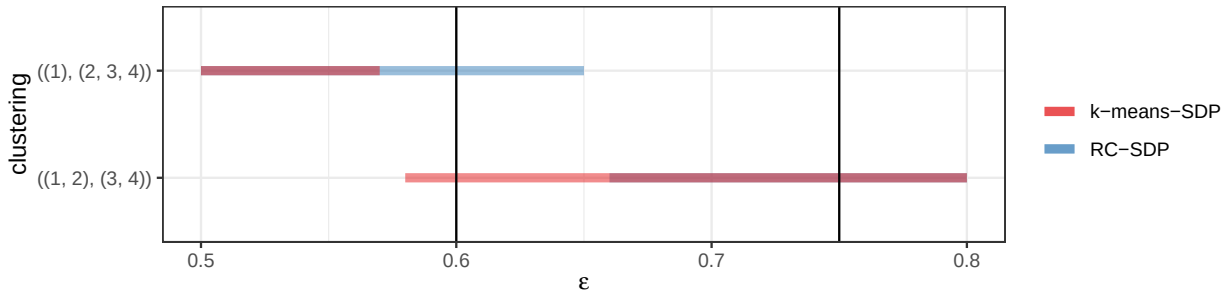
Let's try the epsilon graph example from the text. This graph has four vertices connected in series by three edges. The $2-3$ and $3-4$ edges have weight 1 but the $1-2$ edge has weight $\epsilon \in (0,1)$.



Recall that the graph Laplacian is of the form

$$\begin{bmatrix} \epsilon & -\epsilon & 0 & 0 \\ -\epsilon & 1+\epsilon & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$

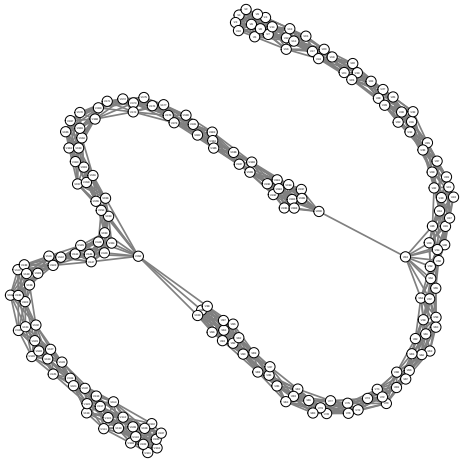
We previously showed that when $\epsilon \in (0, 0.75)$, the optimal ratio cut is the $1-2$ cut, and when $\epsilon \in (0.75, 1)$, the optimal ratio cut is the $2-3$ cut. However, for kernel k-means, the optimal clustering is $\{\{1\}, \{2,3,4\}\}$ when $\epsilon \in (0, 0.6)$ and $\{\{1,2\}, \{3,4\}\}$ when $\epsilon \in (0.6, 1)$. One thing that might be of interest is whether RatioCut-SDP and k-means-SDP yields the same results.



Interestingly, both RatioCut-SDP and k-means-SDP fail to find the correct ϵ where the optimal cut switches from $1-2$ to $2-3$.

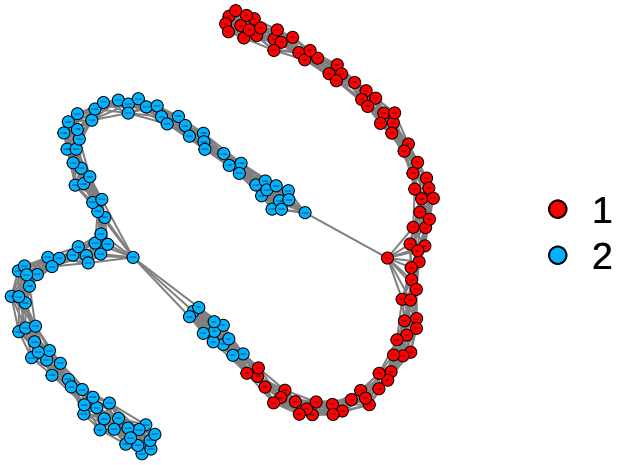
Example 3

Here we look at the "spiral graph":

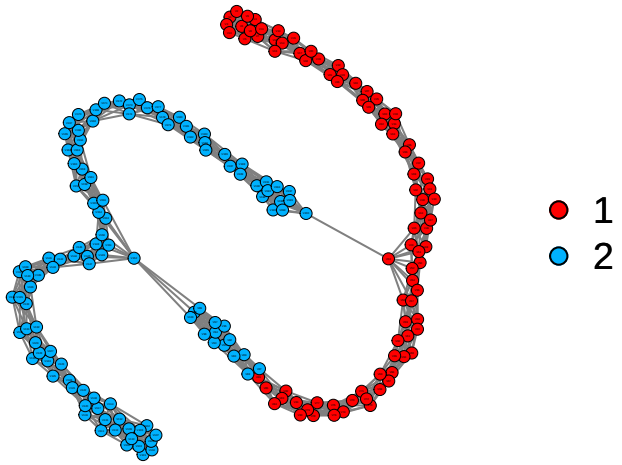


We know that the optimal ratio cut clustering is $\{\{1, ..., 100\}, \{101, ..., 200\}\}$. This also happens to coincide with the optimal k-means clustering. We also know that the spectral clustering relaxation of ratio cut fails to provide the optimal ratio cut (even when the k-means rounding step is initialized with the correct clustering).

Applying RatioCut-SDP to this graph:



Here we see that RatioCut-SDP makes the same mistake the spectral clustering relaxation of ratio cut makes. We can also try k-means SDP:



Strangely, k-means-SDP makes a similar mistake RatioCut-SDP makes, while performing k-means on the embedding of $L^\dagger$ is able to find the optimal ratio cut (although Lloyd's algorithm runs into local minima issues as we add more embedding dimensions, so it's not always practically feasible).

It's also worth noting that these optimization problems takes a while to solve.

RatioCut-SDP optimality criterion

Ling and Strohmer outlines the following as being required for RatioCut-SDP to find the optimal ratio cut:

$$||D_\delta||_{op} < \frac{\lambda_{k+1}(L_{iso})}{4}$$

where

- W_{iso} is the weight matrix given that the optimal ratio cut is known and the edges of W are cut accordingly
- $W_\delta = W - W_{iso}$, the edges that are to be cut for the optimal ratio cut
- D_{iso}, D_δ, L_{iso}, and L_δ are the degree and combinatorial graph Laplacian matrices constructed from W_{iso} and W_δ
- $||\cdot||_{op}$ is the operator norm
- $\lambda_i(A)$ is the i^{th} eigenvalue of matrix A

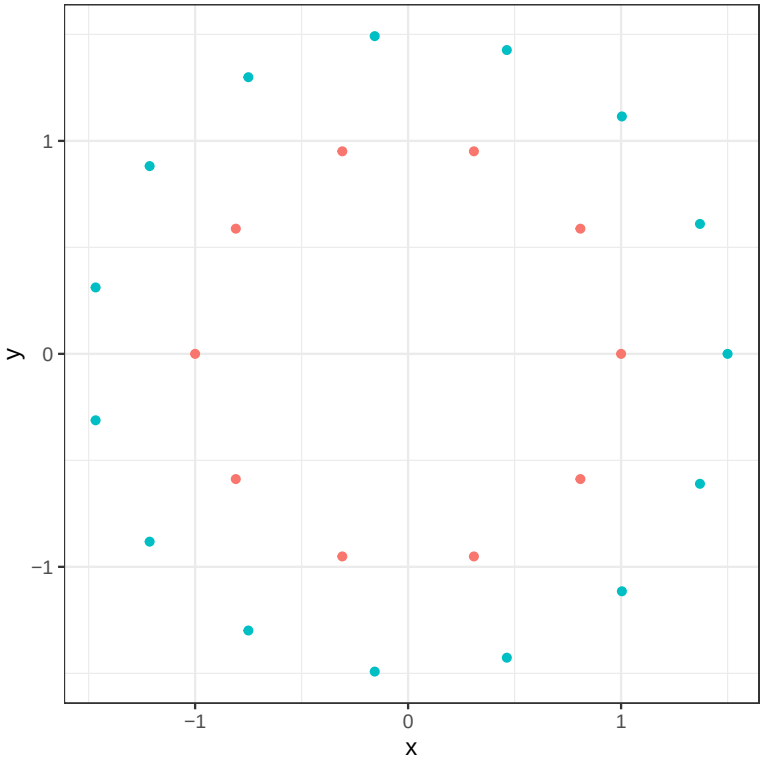
Examples 1-3 all violate this criterion.

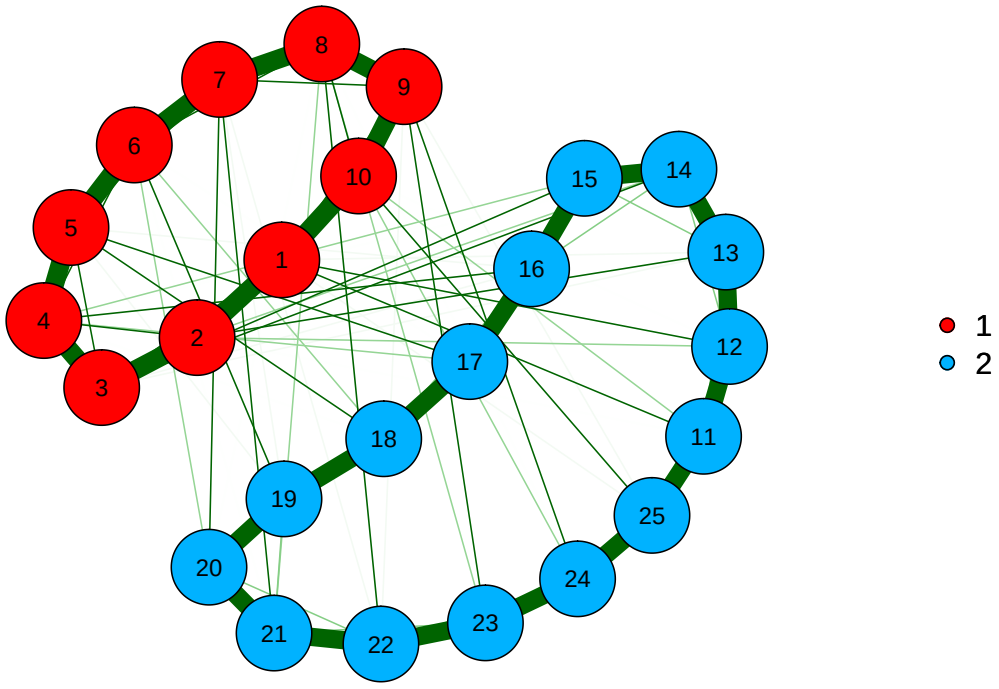
Intuitively, the criterion says that the within-cluster connectivity must be large compared to the magnitude of the betweeen-cluster edges. Note that L_{iso} has k zero eigenvalues, one for each subgraph, and $\lambda_{k+1}(L_{iso})$ is the smallest "fiedler value" among the subgraphs. The fiedler value characterizes how tightly connected a graph (or in this case, subgraph) is. The first k eigenvectors of L_{iso} produce exactly the solution to the (fully constrained) ratio cut.

Examples from the Paper

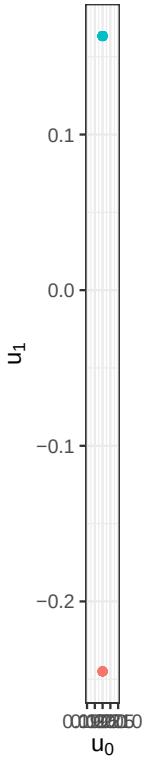
Concentric Circles

Here, we will set $r_2 = \frac{3}{2}$, $r_1 = 1$, $n = 10$, $m = 15$. Theorem 4.1 says if $\gamma \leq \left(2 + \frac{\log 4m}{\log \frac{m}{2\pi}}\right)^{-1} \left(\frac{\frac{n^2\Delta^2}{16} - 1}{2}\right)$, where $\Delta = \frac{r_2 - r_1}{r_1}$, then a graph constructed using the heat kernel with $\sigma^2 = \frac{16 r_1^2 \gamma}{n^2 \log \frac{m}{2\pi}}$ will always be solved by RatioCut-SDP. Here, we will set γ directly at the boundary of the condition.



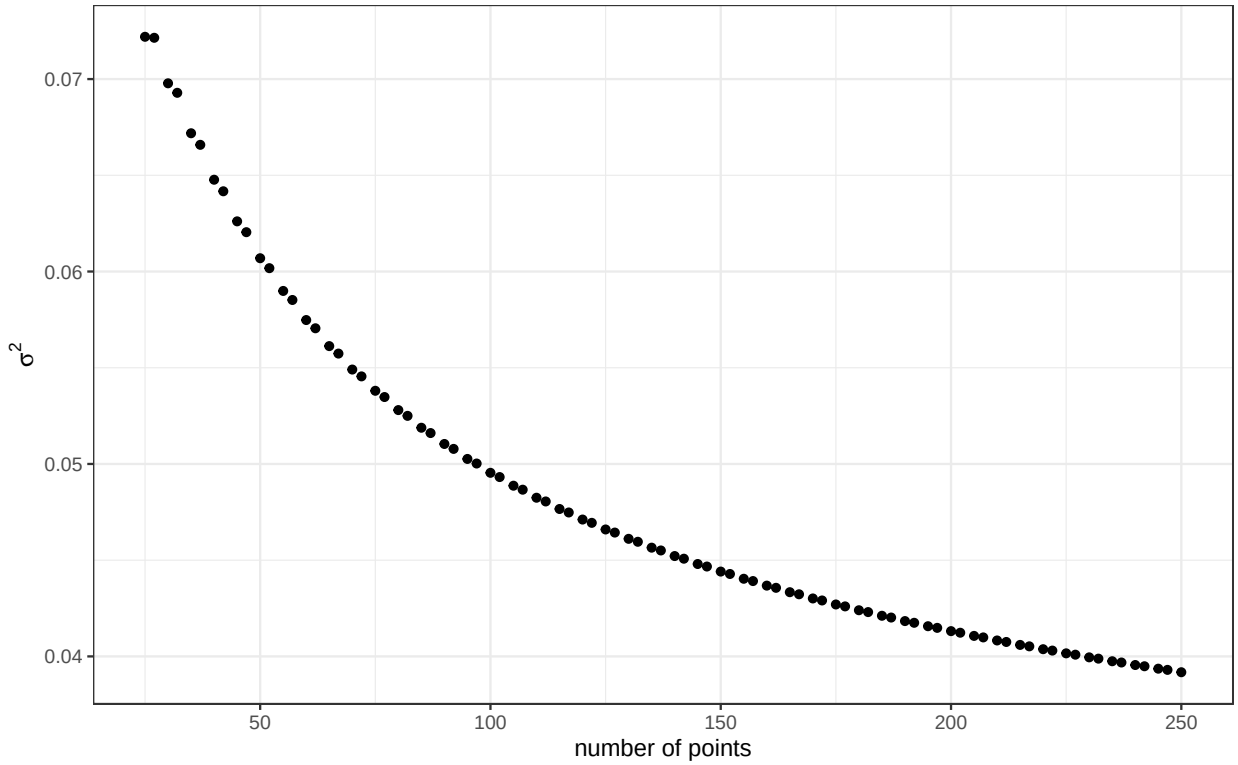


Looking at the original data and the heat kernel graph, it seems like this should be a very "easy" problem to solve. Sure enough, spectral clustering is able to solve this very easily. In fact, the 1-dimensional Laplacian eigenmap has the points in each cluster almost lying on top of one another:



```
clustering  1  2
         1 10  0
         2  0 15
```

We can also see that σ^2 is a decreasing function of the sample sizes, so we actually get better separation as we increase n (and therefore m), and spectral clustering is even more able to separate the two clusters.



This illustrates the main criticism of the Ling and Strohmer result: the condition under which RatioCut-SDP is guaranteed to work is so stringent that it only captures the "easy" problems. Perhaps this points to other methods being guaranteed under this condition.

Alternative Formulation for $k = 2$

In von Luxburg's tutorial, she proposes an alternative but equivalent formulation of the ratio cut objective when $k = 2$:

$$\begin{aligned} &\arg\min_f \mathrm{Tr}(f^\top L f) \\ &\text{s.t. } f \in \mathbb{R}^n \\ &f_i = \begin{cases} \sqrt{\frac{n_2}{nn_1}} & v_i \in C_1 \\ -\sqrt{\frac{n_1}{nn_2}} & v_i \in C_2 \end{cases} \end{aligned}$$

Note that f has the following properties:

- $\sum f_i = 0$

- $||f||_2 = 1$
- $f^\top e = 0$

Since this discrete optimization problem is NP-hard, von Luxburg's proposed solution is to relax the problem by replacing the original constraint on f with its three observed properties. The solution to this is the Fiedler vector, or eigenvector of L that corresponds to its second smallest eigenvalue. This induces the same embedding in $\mathbb{R}^1$ as in the arbitrary k formulation, $\arg\min_H \operatorname{Tr}(H^\top LH)$.

As a reminder, $H \in \mathbb{R}^{n\times k}$ is defined as:

$$h_{ij} = \begin{cases} n_j^{-1/2} & v_i \in C_j \\ 0 & \text{else} \end{cases}$$

The intuition behind the $\operatorname{Tr}(H^\top LH)$ formulation is that if we replace L with L_{iso}, the combinatorial graph Laplacian of the graph with k disconnected parts, then the first k eigenvectors of L_{is} are of the form of H (up to a permutation), and by the Rayleigh-Ritz theorem, this is the global minimizer.

However, once we introduce edges that connect the disjoint subgraphs, the eigenvectors of L no longer take on this form, no matter how small the inter-cluster edge weights are. A slightly more formal statement is:

Let $\epsilon = \max W_\delta$, the largest inter-cluster edge weight (and assume that W describes a connected graph). Then as long as $\epsilon > 0$, even as $\epsilon \to 0$, $\begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix} \not\to H$, where v_j are the k eigenvectors of L that correspond to its k smallest eigenvalues.

But we do have the following (or I think we do—I don't have a proof):

Proposition: Let $G_{iso} = (V, E_{iso})$ be a graph with two disconnected subgraphs, with corresponding weight matrix W_{iso}. Then let $\epsilon > 0$ and $G(\epsilon) = (V, E(\epsilon))$ be a connected graph constructed from G_{iso} such that $||D_\delta|| = \epsilon$ (where D and D_{iso} are the diagonal degree matrices of W and W_{iso} and $D_\delta = D - D_{iso}$, as described by Ling and Strohmer). Let $L(\epsilon)$ be the combinatorial graph Laplacian of $G(\epsilon)$.
Then as $\epsilon \to 0$, the second eigenvector of $L(\epsilon)$, $v_2 \to f$, where f is of the form described above.

In other words, no matter how close L and L_{iso} are to each other, the first two eigenvectors of L will never approach the first two eigenvectors of L_{iso}, which is in part the intuition behind the justification for spectral clustering to approximate the ratio cut solution. Perhaps we should use the alternative formulation $\arg\min_f \operatorname{Tr}(f^\top Lf)$ instead.[7]

Alternative SDP Problem

Based on this, we can try to form an alternative SDP problem using f instead of H:

Let $\Phi = ff^\top$. Then we can observe:

- $\Phi = \Phi^\top, \Phi \in \mathbb{R}^{n\times n}$
- $\Phi e = ff^\top e = f0 = \overrightarrow{0}$
- $\operatorname{Tr}(\Phi) = \operatorname{Tr}(ff^\top) = \operatorname{Tr}(f^\top f) = \operatorname{Tr}(1) = 1$ (this should be $k-1$ in the general case)
- Φ is positive semidefinite
  - Φ has rank 1 (or $k-1$ in the general case)
  - $\Phi f = ff^\top f = f1 = f$
- $\phi_{ij} = \begin{cases} \frac{n_2}{nn_1} & v_i, v_j \in C_1 \\ \frac{n_1}{nn_2} & v_i, v_j \in C_2 \\ -\frac{1}{n} & \text{else} \end{cases}$

[7] Note: I'm still working on generalizing $f \in \mathbb{R}^{n\times(k-1)}$ for the arbitrary k case. It's straightforward to describe this as a simplex in $\mathbb{R}^{k-1}$ with point masses on the vertices proportional to the cluster sizes such that the center of mass is the origin, but the actual closed form is taking some time to figure out.

Rewriting $\text{Tr}(f^\top Lf) = \text{Tr}(Lff^\top) = \text{Tr}(L\Phi)$, we can state a relaxation of the ratio cut problem as:

$$\begin{aligned} \arg\min_{\Phi} & \text{Tr}(L\Phi) \\ \text{s.t. } & \Phi \text{ is PSD} \\ & \Phi e = 0 \\ & \text{Tr}(\Phi) = 1 \\ & \Phi \geq -\frac{1}{n} \text{ element-wise} \end{aligned}$$

... with the hope that this will let us relax the optimality criterion from Ling and Strohmer.

Proximities

We noted that as $\epsilon \to 0$ (i.e., $L \to L_{iso}$ while keeping the graph generating L connected), v_2, the Fiedler vector of L, approaches f. Then it would be intuitive to believe that $||v_2 - f|| \leq g(\epsilon)$ for some monotone increasing function g. We saw previously that performing 2-means clustering on v_2 (treating it as an embedding in $\mathbb{R}^1$) is equivalent to finding n_1 and n_2 such that $||v_2 - f||$ is minimized. Furthermore, we saw that the 2-means clustering solution is not equivalent to the ratio cut solution. So minimizing $||v_2 - f||$ does not give us the ratio cut solution in the general case. However, if we can put an upper bound on $||v_2 - f||$, then perhaps we can define criteria for which the 2-means clustering solution on v_2 is equivalent to the ratio cut solution (and hopefully this will be looser than the Ling-Strohmer criterion for RatioCut-SDP).