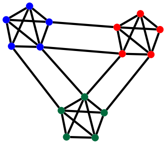
A Tighter Analysis of Spectral Clustering, and Beyond

Peter Macgregor and He Sun

Spectral Clustering



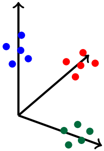
Graph G

$\rightarrow$

$$\begin{bmatrix} B_1 & C_{12} & C_{13} \\ C_{12}^\intercal & B_2 & C_{23} \\ C_{13}^\intercal & C_{23}^\intercal & B_3 \end{bmatrix}$$

Laplacian L_G

$\rightarrow$



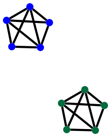
Spectral Embedding

Spectral Clustering Algorithm

Input: Graph G, number of clusters k.

1. Find k eigenvectors of the graph Laplacian matrix.
2. Embed vertices into $\mathbb{R}^k$ according to eigenvectors.
3. Perform k-means clustering in $\mathbb{R}^k$.

Why Does Spectral Clustering Work?



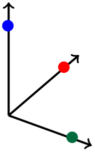
Graph G

$\rightarrow$

$$\begin{bmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{bmatrix}$$

Laplacian L_G

$\rightarrow$

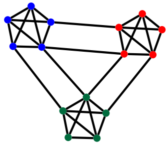


Spectral Embedding

Intuition

1. Suppose k clusters are disconnected.
2. Then, Laplacian matrix is block-diagonal.
3. First k eigenvectors are indicator vectors of clusters.
4. If small number of edges are added, eigenvectors don't change too much.

Why Does Spectral Clustering Work?



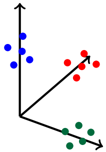
Graph G

$\rightarrow$

$$\begin{bmatrix} B_1 & C_{12} & C_{13} \\ C_{12}^{\intercal} & B_2 & C_{23} \\ C_{13}^{\intercal} & C_{23}^{\intercal} & B_3 \end{bmatrix}$$

Laplacian L_G

$\rightarrow$

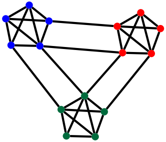


Spectral Embedding

Intuition

1. Suppose k clusters are disconnected.
2. Then, Laplacian matrix is block-diagonal.
3. First k eigenvectors are indicator vectors of clusters.
4. If small number of edges are added, eigenvectors don't change too much.

Why Does Spectral Clustering Work?



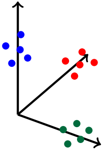
Graph G

$\rightarrow$

$$\begin{bmatrix} B_1 & C_{12} & C_{13} \\ C_{12}^{\intercal} & B_2 & C_{23} \\ C_{13}^{\intercal} & C_{23}^{\intercal} & B_3 \end{bmatrix}$$

Laplacian L_G

$\rightarrow$



Spectral Embedding

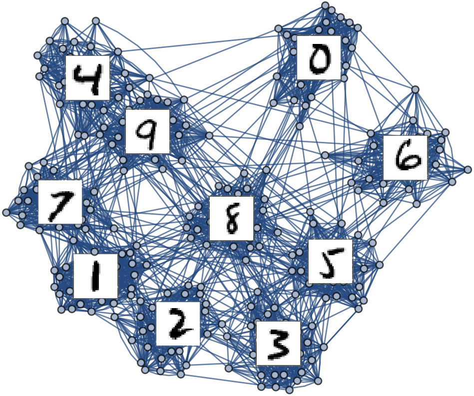
Our Results

We prove, for well-clustered graphs, that

1. there is a close connection between the indicator vectors of the clusters, and the Laplacian eigenvectors;
2. there is an upper bound on the number of vertices misclassified by the spectral clustering algorithm.

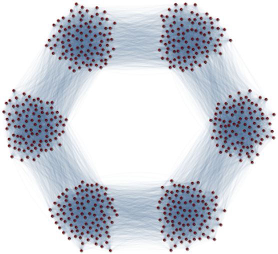
Both results significantly tighten the analysis by Peng, Sun, and Zanetti [SICOMP'17].

Graphs with Inter-Cluster Structure



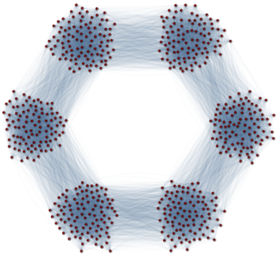
- In many graphs, cluster similarity is not symmetric.
- Can this be used to improve the performance of spectral clustering?

Meta-Graphs

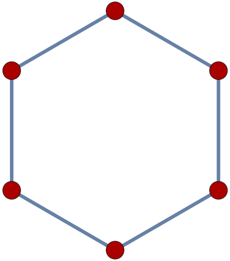


Graph G

Meta-Graphs

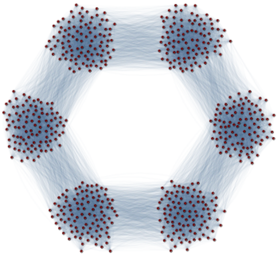


Graph G

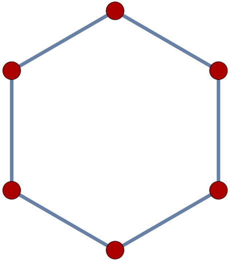


'Meta-Graph' M

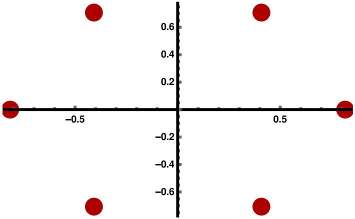
Meta-Graphs



Graph G

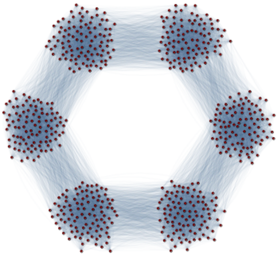


'Meta-Graph' $\mathcal{M}$

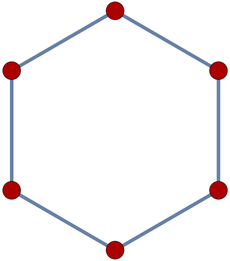


Spectral Embedding of $\mathcal{M}$

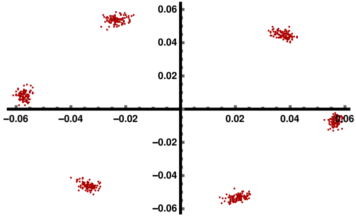
Meta-Graphs



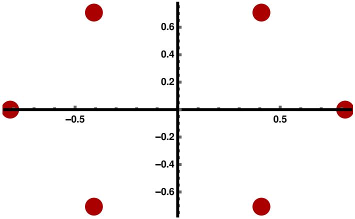
Graph G



'Meta-Graph' $\mathcal{M}$

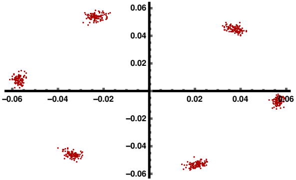


Spectral Embedding of G

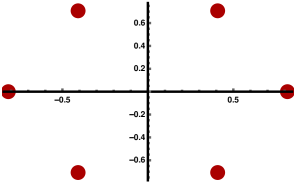


Spectral Embedding of $\mathcal{M}$

Spectral Clustering with Meta-Graphs



Spectral Embedding of G



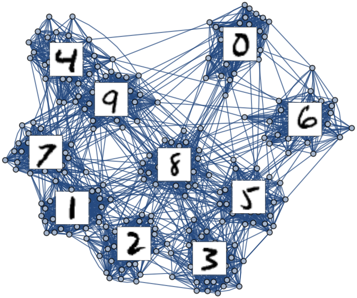
Spectral Embedding of M

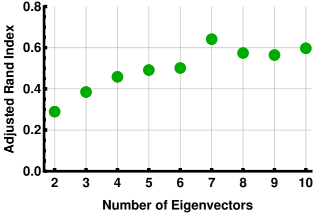
OUR RESULTS

We prove

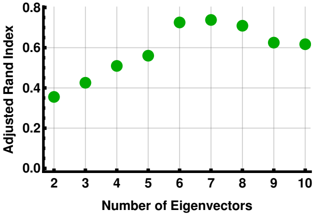
1. a close connection between the spectral embedding of the meta-graph M, and the eigenvectors of the Laplacian L_G;
2. if the meta-graph vertices are well-separated with $\ell < k$ eigenvectors, spectral clustering with ℓ eigenvectors performs well;
3. for graphs with certain structures, spectral clustering with $\ell < k$ eigenvectors performs better than spectral clustering with k eigenvectors.

Experimental Results - MNIST and USPS





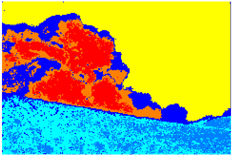
MNIST



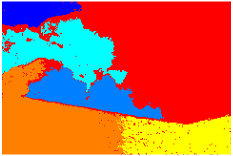
USPS

Experimental Results - BSDS

(a) Original Image



(b) 6 clusters found with 3 vectors



(c) 6 clusters found with 6 vectors

(d) Original Image

(e) 45 clusters found with 7 vectors

(f) 45 clusters found with 45 vectors

Using k eigenvectors on all 500 images in the dataset gives an average Rand Index of 0.71. Using $k/2$ eigenvectors gives an average of 0.74.

Summary

- A tighter analysis of the classical spectral clustering algorithm.
- Clustering structured graphs with fewer than k eigenvectors.
- Experimental evaluation of spectral clustering with fewer than k eigenvectors.