Lecture 15 - Laplacian embeddings & spectral clustering

15.1 GL's continued

In the last lecture we took a closer look at the spectrum of graph Laplacian matrices. We saw that if G is a disconnected graph with M-connected components, ie,

$$G = \{X, W\} \quad G = \bigcup_{i=1}^{M} G_i, \quad G_i = \{X_i, W_i\}$$

Then, after possibly re-ordering the vertices of G we can write

$$W = \begin{bmatrix} W_1 & & & \emptyset \\ & W_2 & & \\ & & \ddots & \\ \emptyset & & & W_M \end{bmatrix} \implies L = \begin{bmatrix} L_1 & & & \emptyset \\ & L_2 & & \\ & & \ddots & \\ \emptyset & & & L_M \end{bmatrix}$$

This in turn implied that

$$\mathrm{Null}(L) = \mathrm{Span}\{\mathbb{1}_{G_i}\}_{i=1}^{M}$$

where $\mathbb{1}_{G_i}$ is the indicator vector of subgraph G_i, ie,

$$(\mathbb{1}_{G_i})_j = \begin{cases} 1 & \text{if } x_j \in X_i \\ 0 & \text{otherwise} \end{cases}$$

This is an important discovery, & the cornerstone of graphical alg such as spectral clustering & diffusion maps. That is, if G has M-clusters / subgraphs that are disconnected, then L has an M-dim null space spanned by indicators of those clusters.

In real world applications we never deal with truly disconnected graphs. But rather weakly connected graphs. Broadly speaking these are graphs whose weight matrices are nearly block diag.

$$W = \begin{bmatrix} W_1 & & \mathcal{O}(\varepsilon) \\ & W_2 & \\ \mathcal{O}(\varepsilon) & \ddots & W_M \end{bmatrix} \quad \text{or} \quad \|W - \tilde{W}\| = \mathcal{O}(\varepsilon)$$

where $\tilde{W}$ is block-diag. One can use perturbation analysis to show that the intuition from the disconnected case generalizes, ie;

Suppose G is weakly connected with M clusters. Let L be the graph Laplacian on G with eigenpairs $(\lambda_i, \underline{v}_i)_{i=1}^{n}$ (in increasing order) then

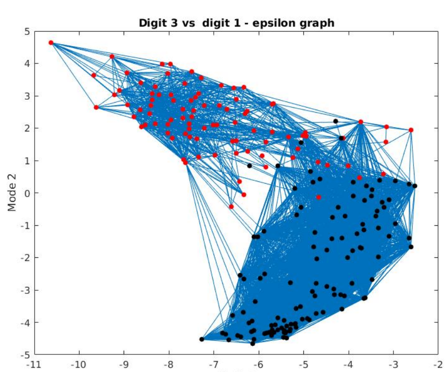
$$0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_M = O(\varepsilon) \leq \lambda_{M+1} \cdots$$

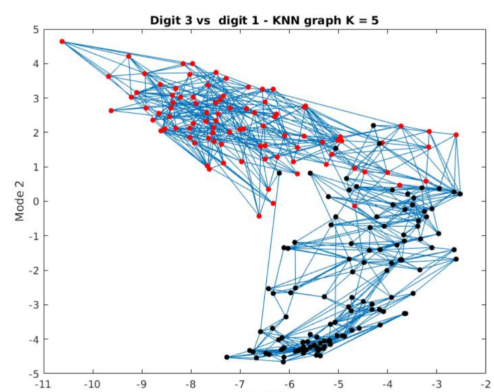
& that $\text{span}\{\underline{v}_i\}_{i=1}^{M} \approx \text{span}\{\mathbb{1}_{G_i}\}_{i=1}^{M}$.

See Hoffmann et al (2020) for precise proofs & statement.

As an example we can take a quick look at MNIST. We only consider the digits (1, 3)

PCA modes ↳

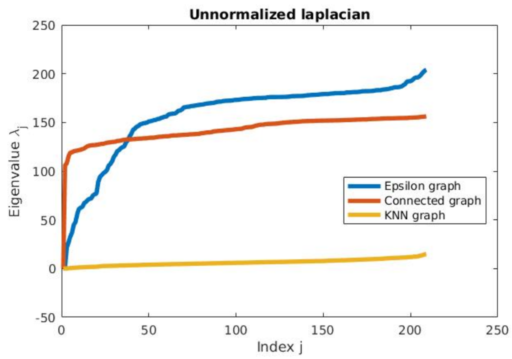




$\eta(t) = \mathbb{1}_{\{t \leq r\}}$

(Code on Syll. Page on Canvas)

We can now compute the spectrum of GL's on these graphs.

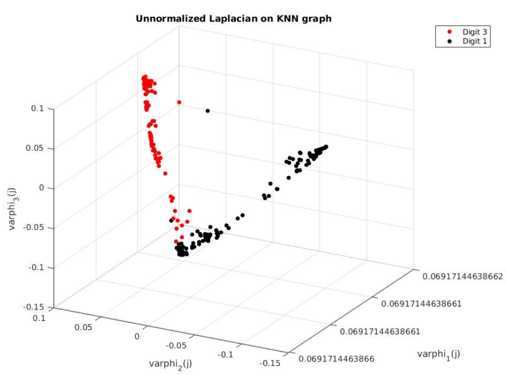


ans =

0.0000	-0.0000	0.0000
0.9986	105.9650	0.0414
21.7370	107.7931	0.1239
26.7655	115.2667	0.3269
32.7235	119.0158	0.4694
36.8659	119.2605	0.5636
44.9942	120.4345	0.7535
47.6328	120.8167	0.9381
55.8548	121.0942	1.0437
61.2600	121.5685	1.1925
62.4118	122.0143	1.2445
63.5510	122.5396	1.2964
67.2153	123.2783	1.4437
68.3995	124.6266	1.5142
70.3951	125.6829	1.5807
71.4500	126.1747	1.6946
71.9585	126.4159	1.7231
75.0720	126.6149	1.7855
76.3888	126.7153	1.8322
76.7784	127.0169	1.9612
88.4589	127.2401	2.0386
93.9583	127.6779	2.2692
95.9388	127.9136	2.5263
97.9050	128.0107	2.6009
99.1648	128.6159	2.6359
101.0590	129.1393	2.6964
105.5531	129.4036	2.7285
107.3645	129.5846	2.7688
110.1842	129.8969	2.8167

prox graph | η-Gaussian | K-NN

$\underline{v}_3$



$\underline{v}_2$ $\underline{v}_1$

It turns out, the eigenvectors $\underline{v}_1, \dots, \underline{v}_M$ tell us a lot about the clusters to which the vertices belong to.

Suppose G is weakly connected & only has two clusters $M = 2$ then

$\underline{v}_1 = \mathbb{1}$ & λ_1 $\quad \underline{v}_2 \approx \begin{bmatrix} \alpha \mathbb{1}_{G_1} \\ -\beta \mathbb{1}_{G_2} \end{bmatrix}$ & $\lambda_2 = O(\varepsilon)$

Note, α & β are found by ensuring $\underset{\sim}{v}_2 \perp \underset{\sim}{v}_1$. The important point is that $\underset{\sim}{v}_2$ will serve as a good classifier for the pts in X. This line of thinking leads to the idea of Laplacian or spectral embedding / feature maps.

15.2 Laplacian embeddings of data sets

The observations we made above lead to an elegant & simple pre-processing algorithm that enables the exploitation of geometric info, such as clusters in a data set

Alg: Constructing Laplacian embeddings.

Input X

step 1) Build a graph $G = \{X, W\}$

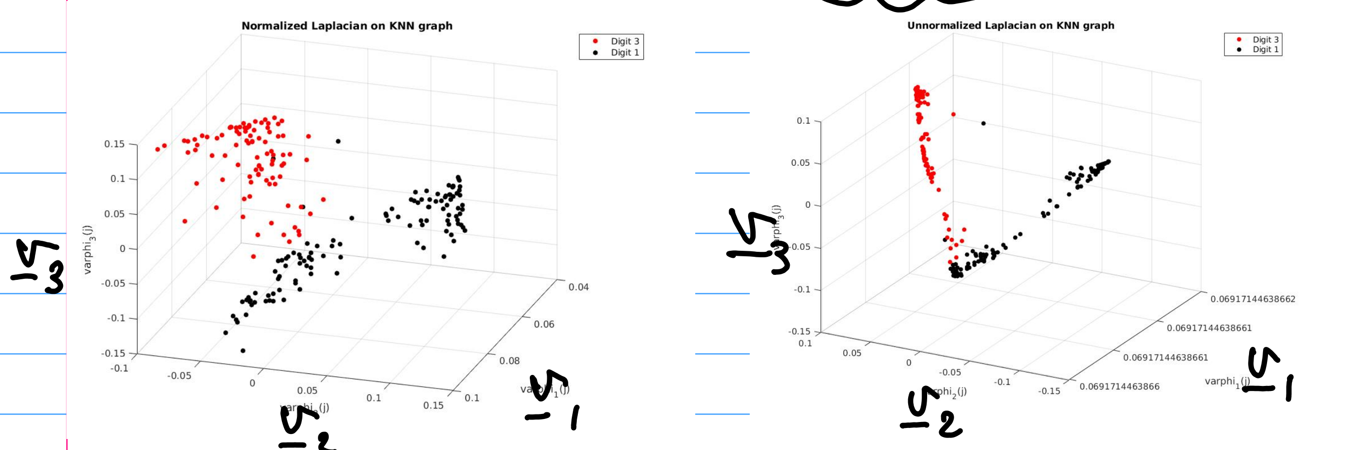
step 2) Compute first few eig. pairs of a GL on G, $\{\lambda_j, \underset{\sim}{v}_j\}_{j=1}^{J}$ for $J \leq n$

Step 3) define feature map

$$F : X \to \mathbb{R}^J, \qquad F(\underline{x}_i) := \begin{bmatrix} (\underline{v}_1)_i \\ (\underline{v}_2)_i \\ \vdots \\ (\underline{v}_J)_i \end{bmatrix}$$

The map F is called the Laplacian embedding of X.

$J = 3$



Food for thought: the map F above is a featuremap just like the ones we defined in RKHS methods. In fact, we can take

$$K(\underline{x}, \underline{x}') = F(\underline{x})^T F(\underline{x}'), \quad \underline{x}, \underline{x}' \in X.$$

what is this kernel?

(6) Combining Laplacian embeddings with the k-means clustering alg. yields the

Spectral clustering algorithm.

Alg: Spectral Clustering:

Input: data set X

step 1) Build graph $G = \{X, W\}$

Step 2) Pick M - Possible # of clusters

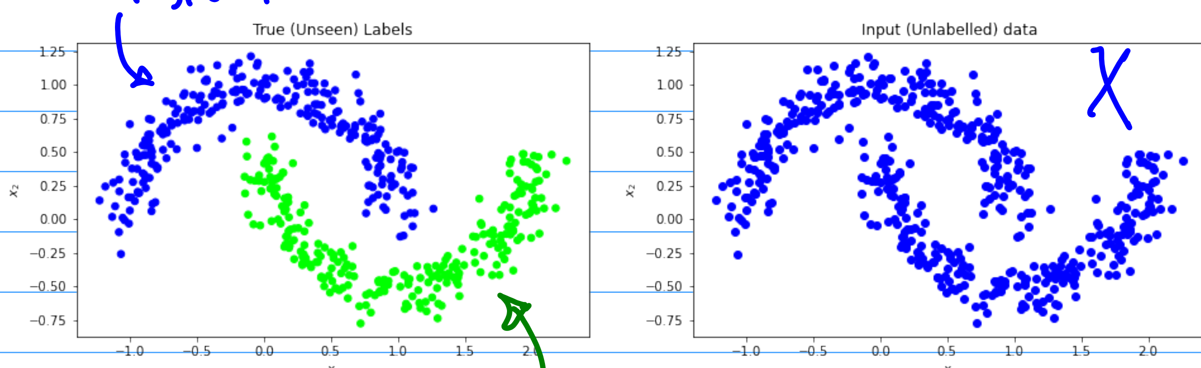
step 3) Build Laplacian embedding

$$F : X \to \mathbb{R}^M$$

step 4) Perform K-means clustering on $F(X)$ - the embedded data.

A quick demo on two moons data set

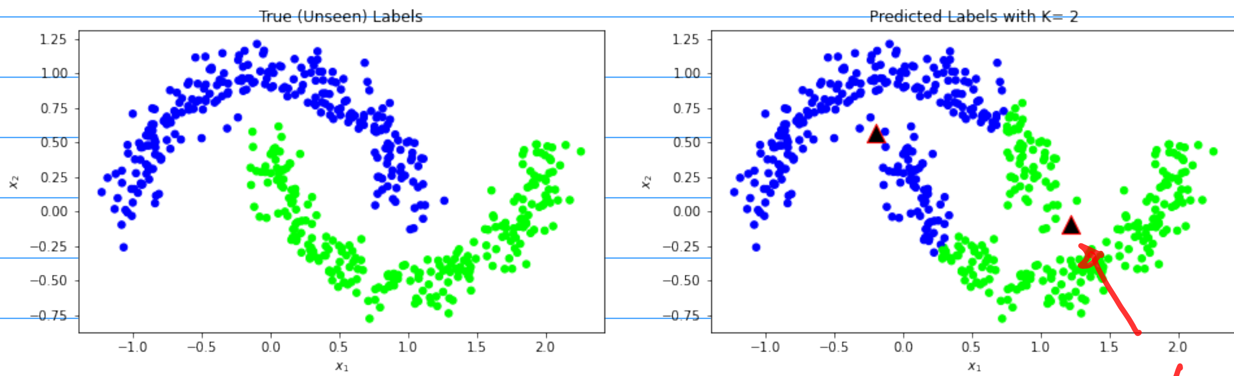
Cluster 1



X

cluster 2

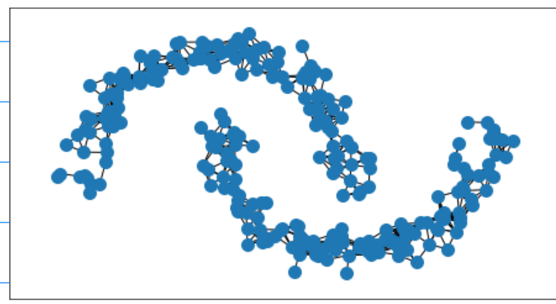
What if we just did K-means?



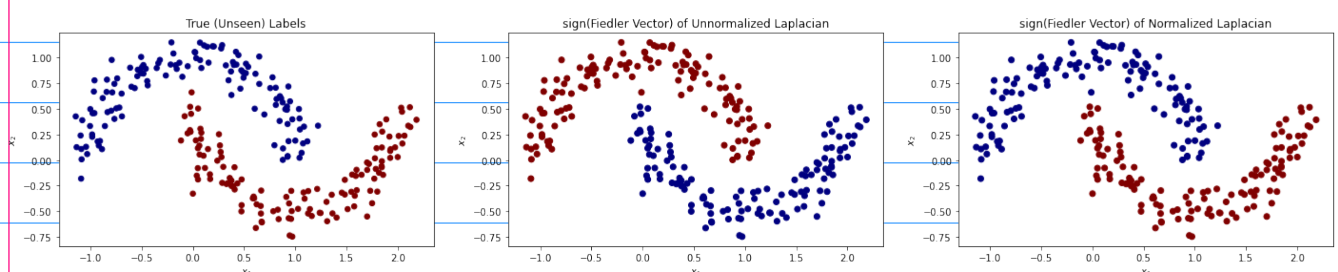
cluster means

Enhance using spec. clustering

graph G



Fiedler vector ($\underline{v}_2$).



Spectral clustering output: