

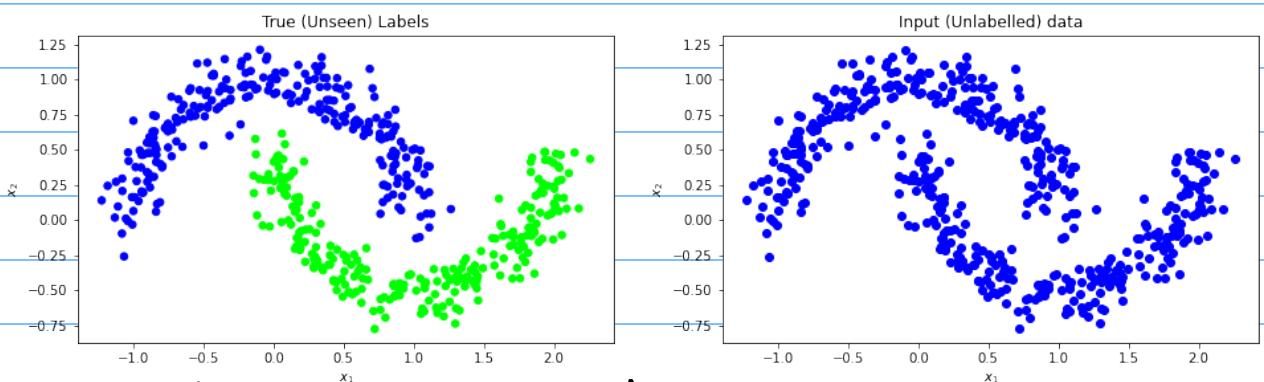
Lecture 20: Spectral Clustering

Equipped with knowledge of graph Laplacians we now introduce the spectral clustering algorithm.

Recall that our goal was to construct a feature map $F: \mathbb{R}^N \rightarrow \mathbb{R}^M$ so that clustering the transformed data $F(X)$ is easier/more efficient than directly clustering X .

20.1 Demo

We will convey spectral clustering with an example first.



Our first task is to construct the graph Laplacian on X & compute its eigen decomposition.

We use the indicator/set function weights

$$w_{ij} = \begin{cases} 1 & \text{if } \|x_i - x_j\| \leq r, \\ 0 & \text{if } \|x_i - x_j\| > r. \end{cases}$$

(you can also use other weights like Gaussian).

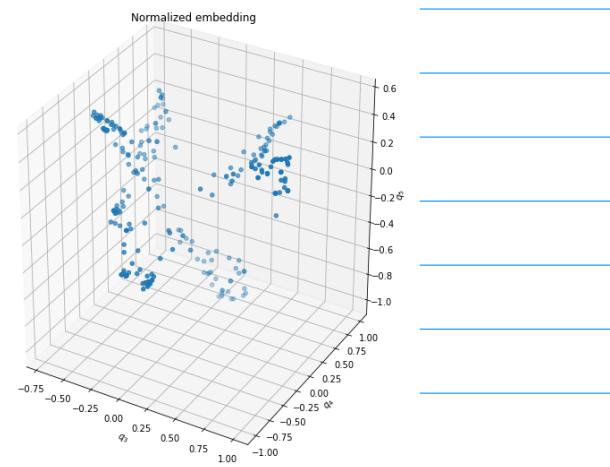
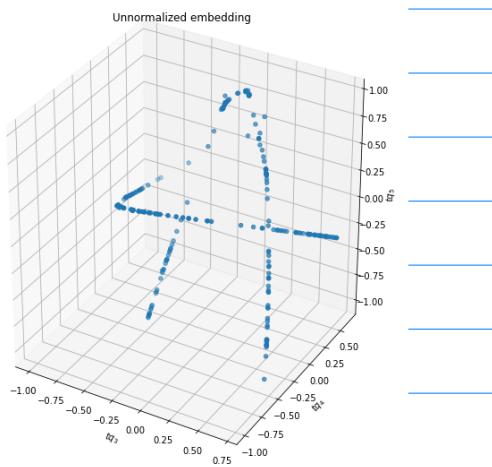
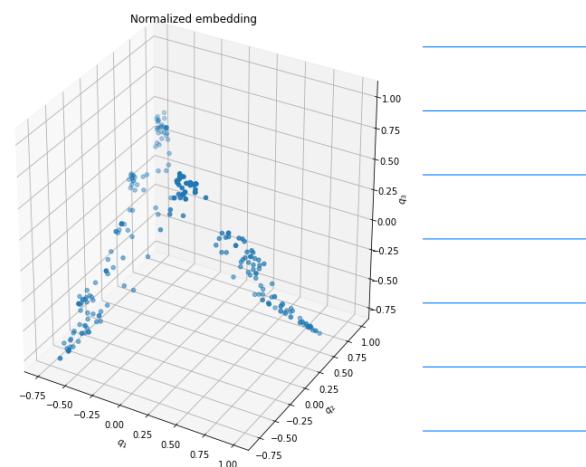
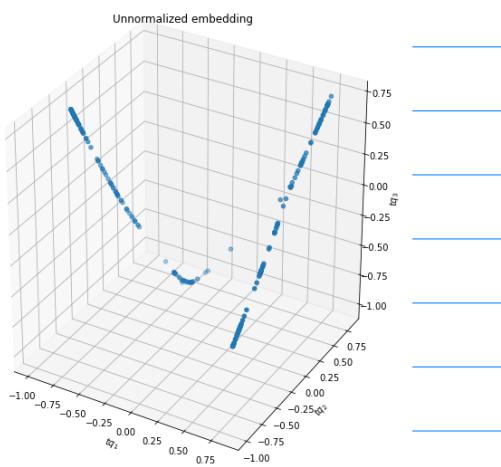
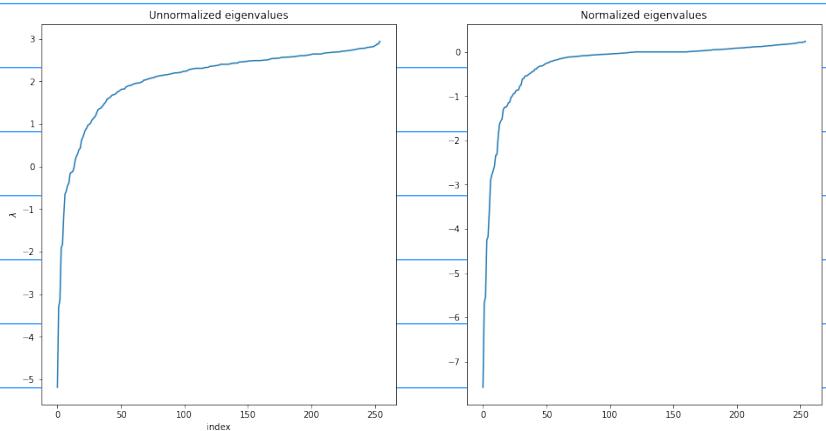
Then we construct the normalized & unnormalized Laplacians \tilde{L} & L & compute their eigendecompositions.

$$\tilde{L} = \tilde{Q} \tilde{\Lambda} \tilde{Q}^T, \quad L = Q \Lambda Q$$

* Important, we normalize columns of Q & \tilde{Q} so that their entries lie in $[-1, 1]$.

- first few eig. vals. are small

- spectrum grows rapidly.



write

$$\tilde{Q} = \begin{bmatrix} \tilde{q}_0 & \tilde{q}_1 & \cdots & \tilde{q}_{N-1} \end{bmatrix}, \quad \tilde{q}_j \in \mathbb{R}^N$$

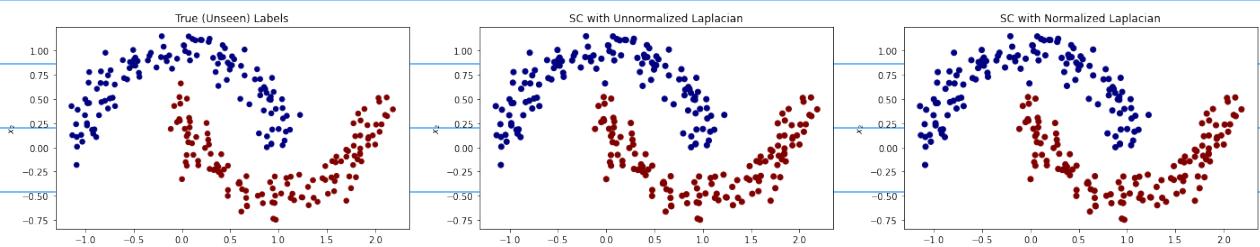
$$Q = \begin{bmatrix} q_0 & q_1 & \cdots & q_{N-1} \end{bmatrix}, \quad q_j \in \mathbb{R}^N$$

We now define the feature map

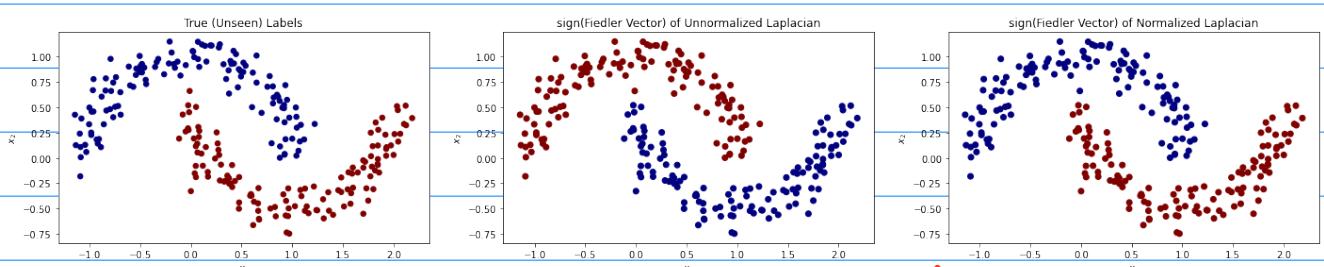
$$M \geq 1 \quad \tilde{F}(\underline{q}_j) = \begin{bmatrix} \tilde{q}_{1j} \\ \tilde{q}_{2j} \\ \vdots \\ \tilde{q}_{Mj} \end{bmatrix}, \quad F(q_j) = \begin{bmatrix} q_{1j} \\ q_{2j} \\ \vdots \\ q_{Mj} \end{bmatrix},$$

$$\tilde{F}: \mathbb{R}^d \rightarrow \mathbb{R}^M, \quad F: \mathbb{R}^d \rightarrow \mathbb{R}^M.$$

Finally we apply K-means to $\tilde{F}(X)$ or $F(X)$.



There is something even more fundamental here.
Let us simply plot $\text{sign}(\tilde{q}_1)$ & $\text{sign}(q_1)$.



* So clearly there is a lot of useful info in the eigenvectors of the laplacian matrix.

20.2 The spectral Clustering Algorithm

* Hence forth we consider the normalized Laplacian L but everything we discuss is true for \tilde{L} as well. Or any other graph Laplacian of your choice.

There are two key steps in SC.

(1) Construct feature map (Laplacian embedding)

$$F: \mathbb{R}^d \rightarrow \mathbb{R}^M$$

(2) Cluster/classify the mapped data $F(X)$.

Step ① requires the construction of W & in turn L . This is in many ways the key to the success of SC & is very much the subject of intense research.

A good strategy, also used in Sklearn is to write

$$L = Q \Delta Q^T$$

& normalize q_j so that their entries lie in $(-1, 1)$.

define

$$F(q_{1j}) = \begin{bmatrix} q_{1j} \\ q_{2j} \\ \vdots \\ q_{mj} \end{bmatrix} \quad \begin{array}{l} \text{jth column of } Q \\ \hookrightarrow \text{with } q_0 \text{ removed.} \end{array}$$

- * It is important in this context that q_j are ordered in increasing order of λ_j the eigenvalues of L .
- * The normalization of q_j is crucial! for ex if you normalize so that $\|q_j\|=1$ then k-means may not work so well.

Step 2 is more or less independent of Step 1. Basically, if F is constructed well then a standard clustering algorithm such as K-means can be used here.

20.3 So what is special about F ?

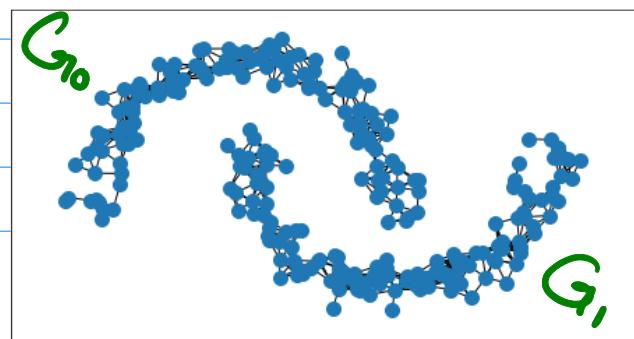
It turns out that the first few eigenvectors of L have a very nice, geometric interpretation.

We explain this in the binary clustering case, i.e., $K=2$. Let $G=\{X, W\}$ be our graph as before but we assume there are two distinct clusters in G .

$$G = G_0 \cup G_1$$

$$G_0 = \{X_0, W_0\}, \#X_0 = N_0$$

$$G_1 = \{X_1, W_1\}, \#X_1 = N_1$$



observe that G_0 & G_1 , are themselves proximity graphs with their own Laplacians L_0, L_1 .

Indeed the Laplacian of G_1 , is obtained precisely as the block diagonal matrix,

$$L = \begin{bmatrix} L_0 & \emptyset \\ \emptyset & L_1 \end{bmatrix}. \quad \begin{array}{l} \text{(we simply need to)} \\ \text{(reorder the } \alpha_j) \end{array}$$

On the other hand, we knew that L_0, L_1 have 0 eigenvalues with corresponding constant eigenvectors

$$L_0 \mathbf{1}_{N_0} = 0, \quad L_1 \mathbf{1}_{N_1} = 0$$

Thus, L has two 0 eigenvalues, ie
 $\lambda_0 = \lambda_1 = 0$,

$$L = \begin{bmatrix} L_0 & \emptyset \\ \emptyset & L_1 \end{bmatrix} \begin{bmatrix} \underbrace{\mathbf{1}_{N_0}}_{\tilde{\mathbf{q}}_0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} L_0 & \emptyset \\ \emptyset & L_1 \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \underbrace{\mathbf{1}_{N_1}}_{\tilde{\mathbf{q}}_1} \end{bmatrix} = 0$$

* Now observe that $\tilde{\mathbf{q}}_0$ & $\tilde{\mathbf{q}}_1$ are precisely the indicators of the X_0 & X_1 , i.e., the set of pts belonging to the sub graphs G_0 & G_1 !

* In practice we typically do not get $\tilde{\mathbf{q}}_0$ & $\tilde{\mathbf{q}}_1$ above. Instead, we get a linear combination of the two.

In fact the first eigenvector is always $\mathbf{q}_0 = \mathbf{1}$ & so the second vector

also known as the Fiedler vector, has the form

$$\underline{\mathbf{q}}_1 = \begin{bmatrix} \mathbf{1}_{N_0} \\ -\frac{N_0}{N_1} \mathbf{1}_{N_1} \end{bmatrix} \quad \left. \right\} N_0$$

thus, $\text{sign}(\underline{\mathbf{q}}_1)$
is our classifier!

adjusted so that

$$\underline{\mathbf{q}}_0^T \underline{\mathbf{q}}_1 = 0$$

