Spectral Clustering

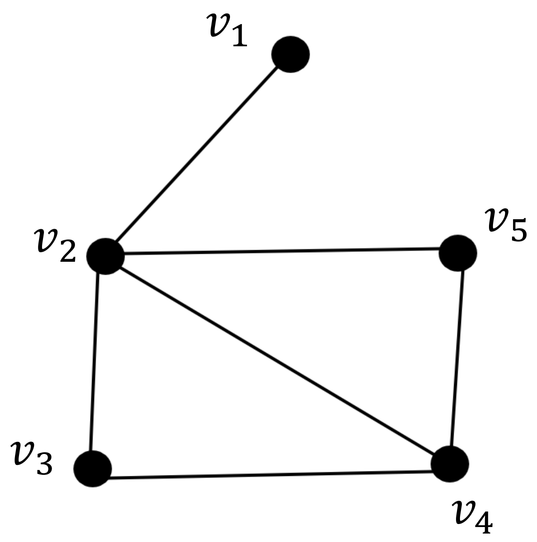
Mustafa Hajij

Graph Laplacian

Let G be a graph on n nodes. The Graph Laplacian is an n by n matrix given by :

$$L = D - A$$

Where D is the degree matrix and A is the adjacency matrix



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Symmetric Graph Laplacian

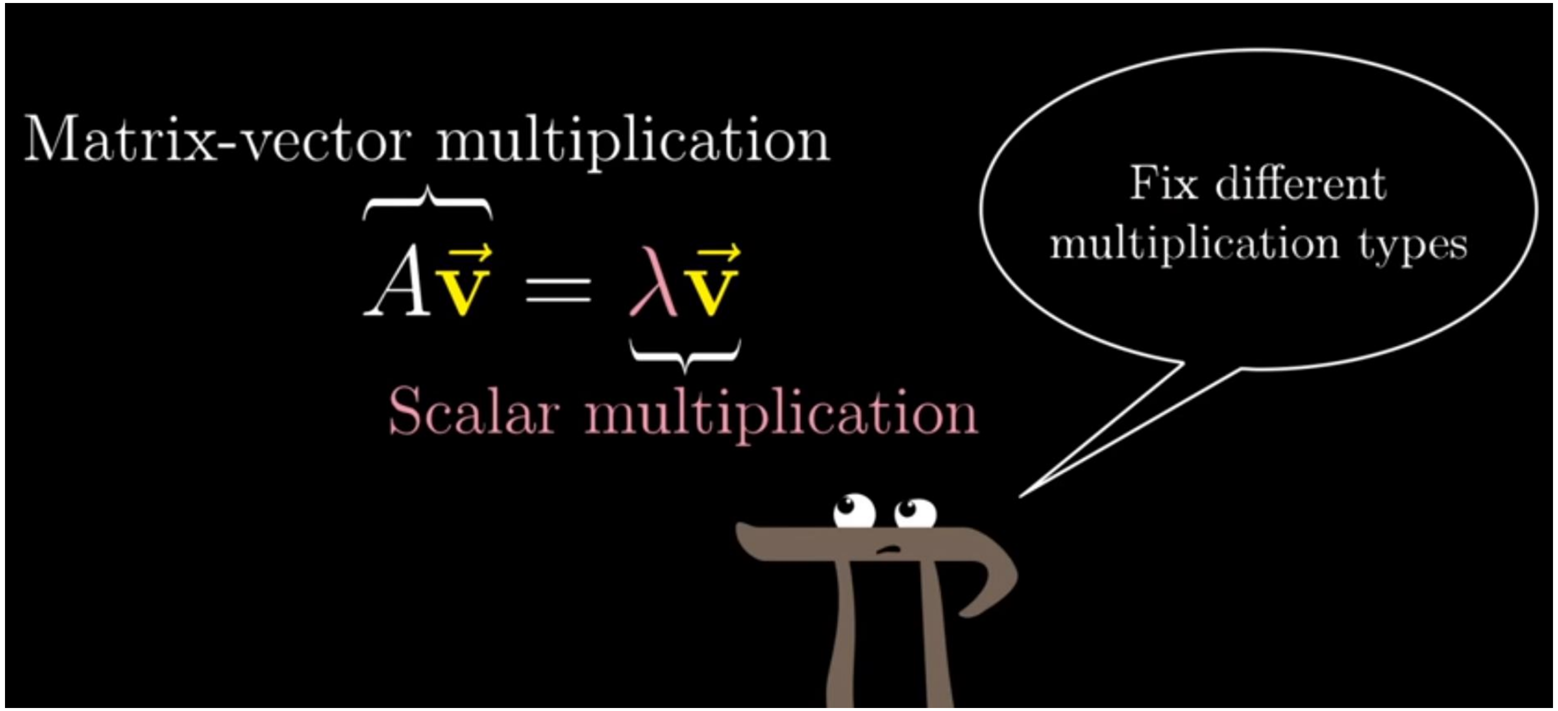
$$L^{\text{sym}} := D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2},$$

Explicitly this is given by:

$$L^{\text{sym}}_{i,j} := \begin{cases} 1 & \text{if } i = j \text{ and } \deg(v_i) \neq 0 \\ -\frac{1}{\sqrt{\deg(v_i)\deg(v_j)}} & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases}$$

Eigenvalues and Eignenvector of a matrix

Watch this lecture for a review of the concepts of the eigenvalues and eigenvectors



Eigenvalues and Eignenvector of a symmetric matrix

A squared matrix is symmetric if $A = A^T$

Eigenvalues and Eignenvector of a symmetric matrix

A squared matrix is symmetric if $A = A^T$

$$\begin{bmatrix} 3 & 2 & 5 \\ 2 & 5 & 4 \\ 5 & 4 & 7 \end{bmatrix}$$

Symmetric

$$\begin{bmatrix} 5 & 1 & 3 \\ 2 & 0 & 2 \\ 0 & 1 & 5 \end{bmatrix}$$

Non-symmetric

Eigenvalues and Eignenvector of a symmetric matrix

A squared matrix is symmetric if $A = A^T$

$$\begin{bmatrix} 3 & 2 & 5 \\ 2 & 5 & 4 \\ 5 & 4 & 7 \end{bmatrix}$$

Symmetric

$$\begin{bmatrix} 5 & 1 & 3 \\ 2 & 0 & 2 \\ 0 & 1 & 5 \end{bmatrix}$$

Non-symmetric

Symmetric matrices are very important in engineering and the solution of many problems come down to finding eigenvectors and eigenvalues for some symmetric matrix

Eigenvalues and Eignenvector of a symmetric matrix

A squared matrix is symmetric if $A = A^T$

$$\begin{bmatrix} 3 & 2 & 5 \\ 2 & 5 & 4 \\ 5 & 4 & 7 \end{bmatrix}$$

Symmetric

$$\begin{bmatrix} 5 & 1 & 3 \\ 2 & 0 & 2 \\ 0 & 1 & 5 \end{bmatrix}$$

Non-symmetric

Symmetric matrices are very important in engineering and the solution of many problems come down to finding eigenvectors and eigenvalues for some symmetric matrix

If $A \in R^{n \times n}$ is a symmetric matrix then it has an orthonormal set of eigenvectors $u_1, u_2, \ldots, u_n$ corresponding to (not necessarily distinct) eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$

Eigenvalues and Eignenvector of a symmetric matrix

A squared matrix is symmetric if $A = A^T$

$$\begin{bmatrix} 3 & 2 & 5 \\ 2 & 5 & 4 \\ 5 & 4 & 7 \end{bmatrix}$$

Symmetric

$$\begin{bmatrix} 5 & 1 & 3 \\ 2 & 0 & 2 \\ 0 & 1 & 5 \end{bmatrix}$$

Non-symmetric

Symmetric matrices are very important in engineering and the solution of many problems come down to finding eigenvectors and eigenvalues for some symmetric matrix

If $A \in R^{n \times n}$is a symmetric matrix then it has an orthonormal set of eigenvectors $u_1, u_2, \ldots, u_n$ corresponding to (not necessarily distinct) eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$

The graph Laplacian is a symmetric matrix

Eigenvalues and Eignenvector of a symmetric matrix

Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

A squared matrix is symmetric is positive semi-definite if all its eigenvalues are larger than or equal to zero

Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

A squared matrix is symmetric is positive semi-definite if all its eigenvalues are larger than or equal to zero

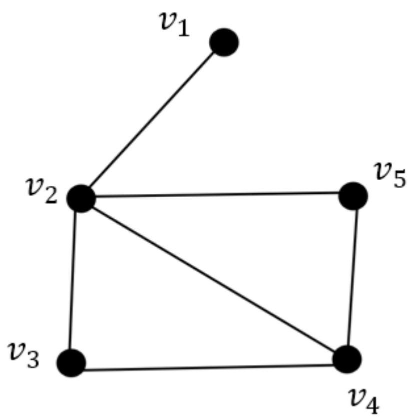
The graph Laplacian is a symmetric positive semi-definite matrix

Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

A squared matrix is symmetric is positive semi-definite if all its eigenvalues are larger than or equal to zero

The graph Laplacian is a symmetric positive semi-definite matrix

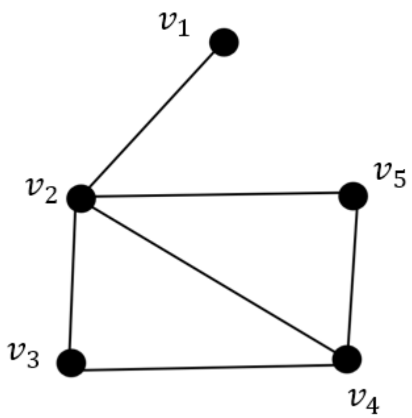


Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

A squared matrix is symmetric is positive semi-definite if all its eigenvalues are larger than or equal to zero

The graph Laplacian is a symmetric positive semi-definite matrix



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

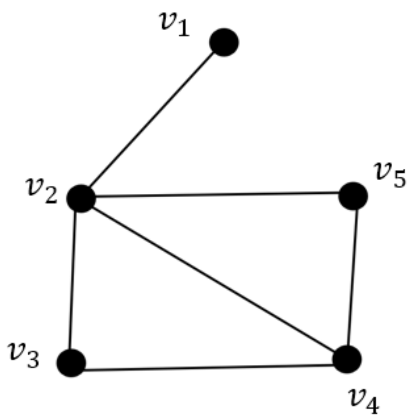
$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix}$$

Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

A squared matrix is symmetric is positive semi-definite if all its eigenvalues are larger than or equal to zero

The graph Laplacian is a symmetric positive semi-definite matrix



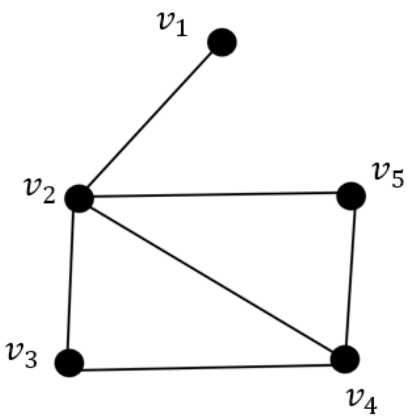
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Eigenvalues and Eignenvector of a symmetric positive definite matrix

A squared matrix is symmetric is positive definite if all its eigenvalues are larger than zero

A squared matrix is symmetric is positive semi-definite if all its eigenvalues are larger than or equal to zero

The graph Laplacian is a symmetric positive semi-definite matrix



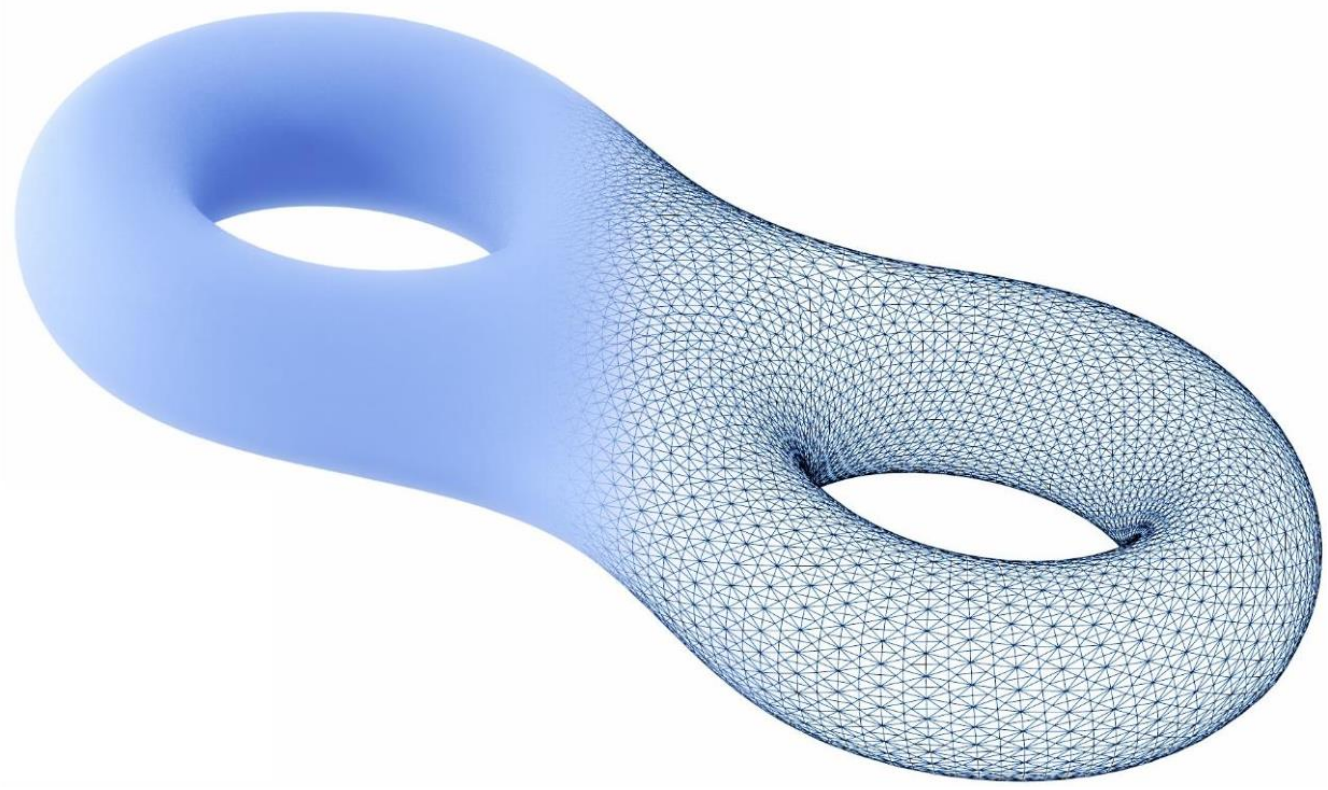
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{bmatrix} \rightarrow L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & -1 & 0 & -1 & 2 \end{bmatrix}$$

Solve its eigenvalues →

All of them are going to be real and non-negative

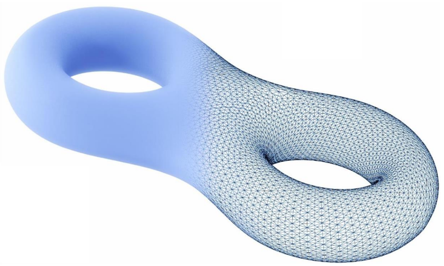
Eigenvalues and Eignenvector of the Laplacian

We can think about a mesh as a graph. We can compute the eigenvalues
And eigenvectors of the Laplacian of this graph



Eigenvalues and Eignenvector of the Laplacian

We can think about a mesh as a graph. We can compute the eigenvalues
And eigenvectors of the Laplacian of this graph



The first 10 eigenvectors of this mesh

Eigenvalues and Eignenvector in Python

In python you can compute the eigenvalues and the eigenvectors of a matrix : numpy.linalg.eig

From the data to the graph

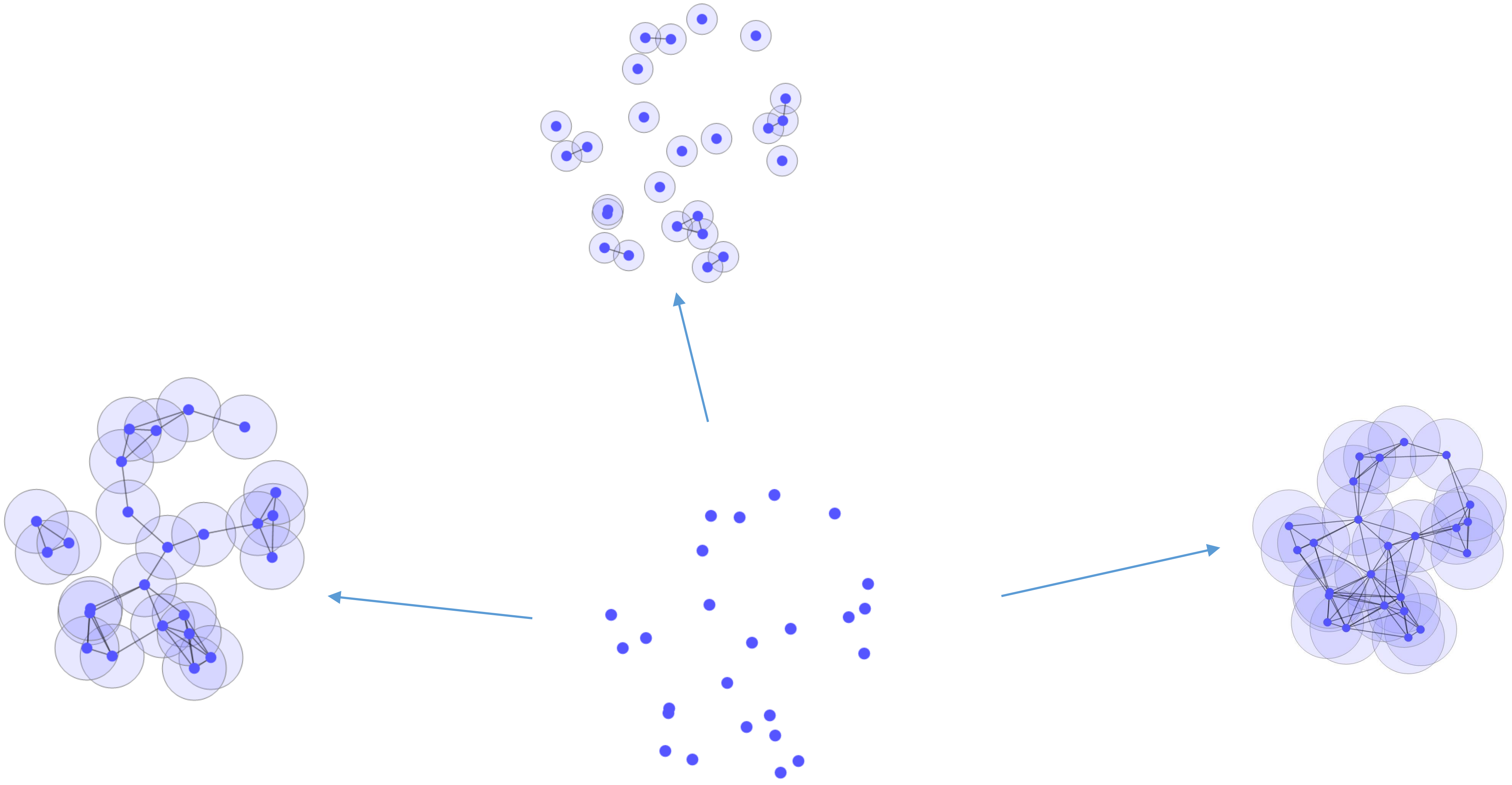
Given the data $X = \{p_1, p_2, \ldots, p_n\}$, we begin by constructing a graph G on the top of the data X:

- The points in X are the vertices of the graph
- The edges in the graph and their weights are determined by how close together and are in X

Three common methods to construct graphs :

- The neighborhood graphs ($\varepsilon -$ neighborhood graph or the knn graph)
- The complete graph on the set X.

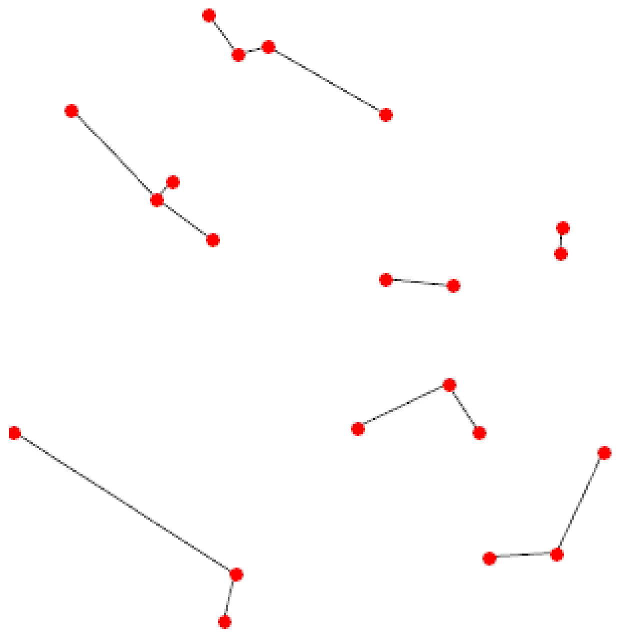
Similarity Graph:ε- Neighborhood Graph



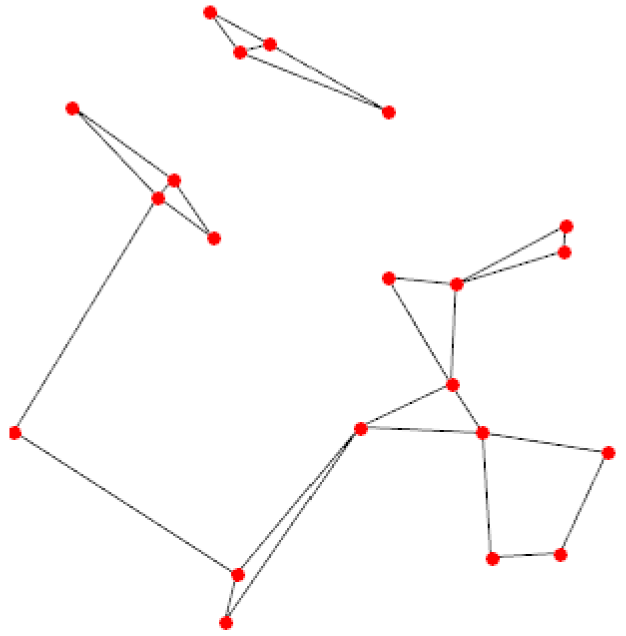
$$Construct\ the\ \varepsilon - neighborhood\ graph$$

A common problem here is which ε we should choose?

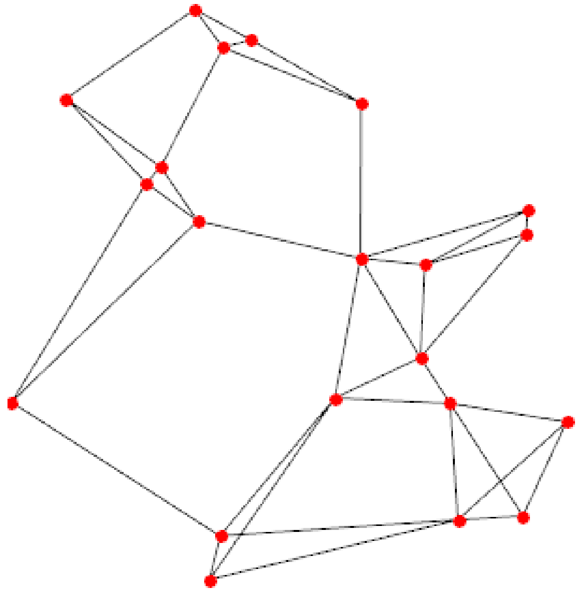
Similarity Graph: KNN Graph



Example of 1-NN graph



Example of 2-NN graph



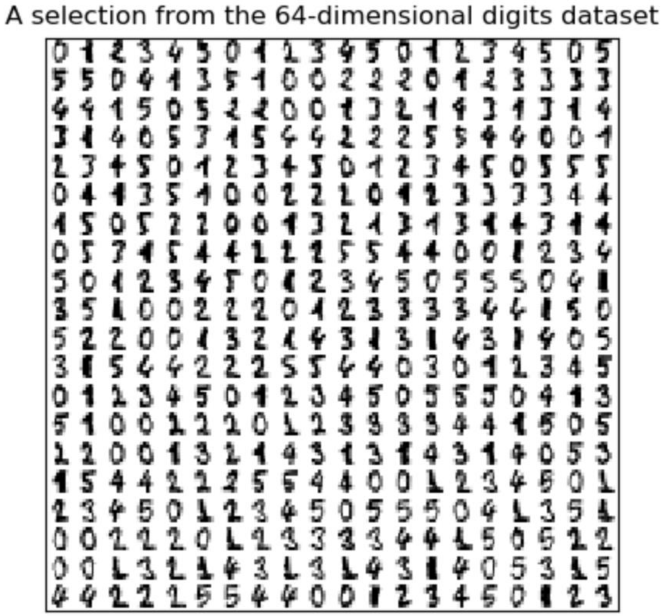
Example of 3-NN graph

Similarity Graph: The fully connected graph

Suppose that we are given a set of points $X = \{p_1, p_2, \dots, p_n\}$ in R^d. Another way to construct a graph on the top of the data X is by connecting all points in X to each other. In this case we weight all edges by $s_{ij} := s(x_i, x_j)$ defined as follows :

$$s(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$$

Spectral Embedding :example



spectral embedding

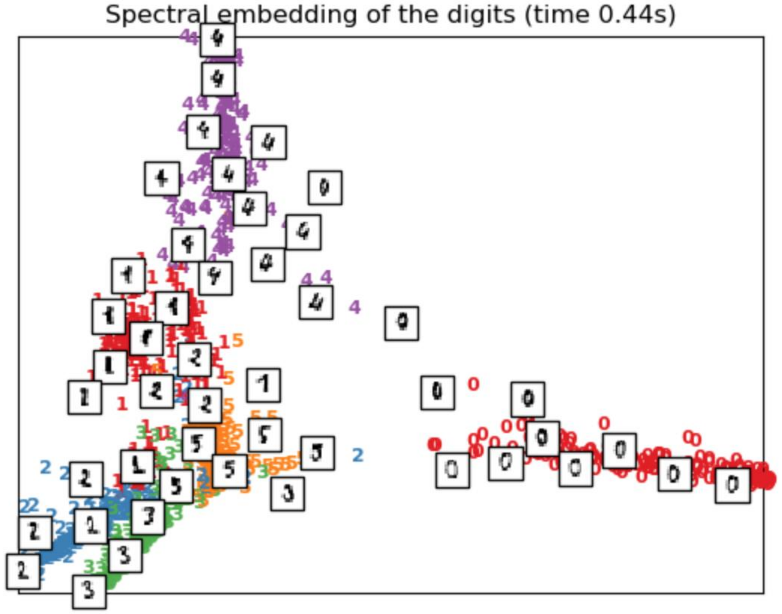
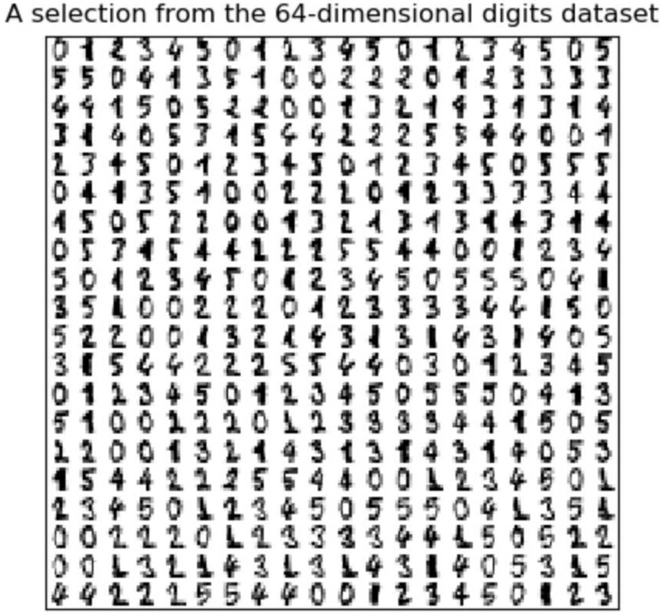


Image source: sklearn example

- The spectral embedding can unfold the nonlinear structures in a data in a high-dimensional feature space so that they become much easier to handle and understand.

Spectral Embedding :example



spectral embedding

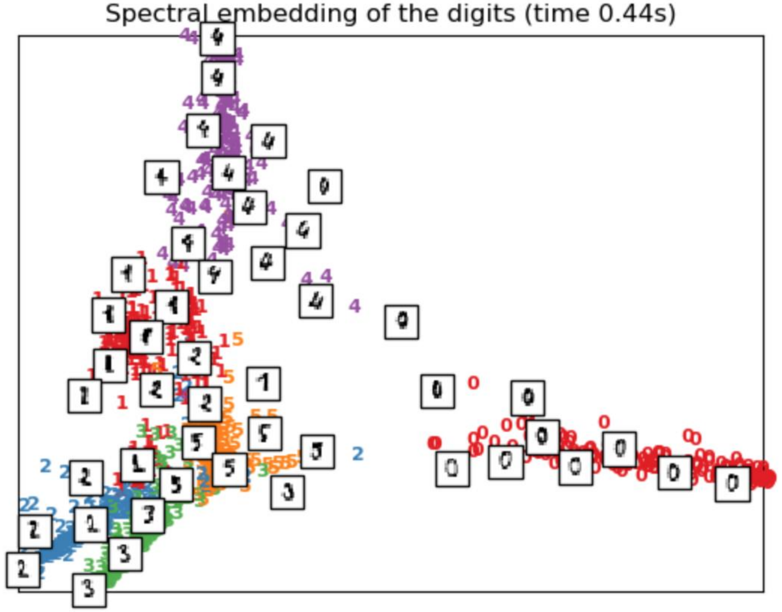


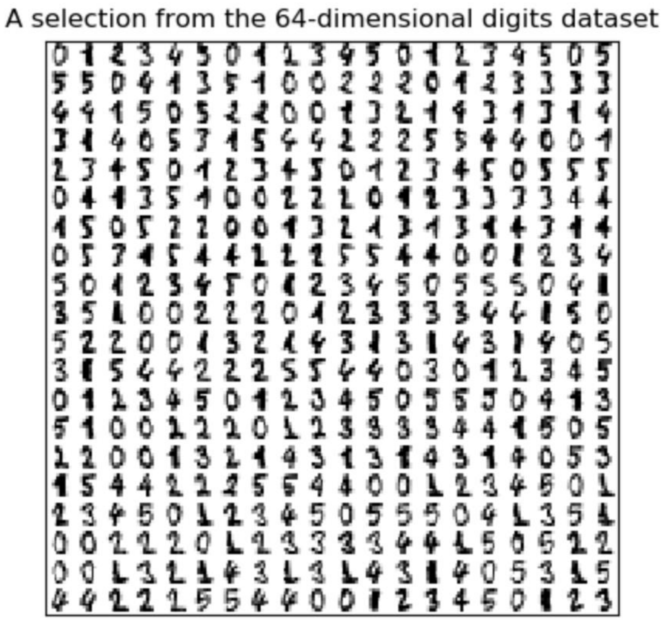
Image source: sklearn example

spectral embedding

$x = [x_1, \ldots, x_{64}]$

$w = [w_1, w_2]$

Spectral Embedding :example



spectral embedding

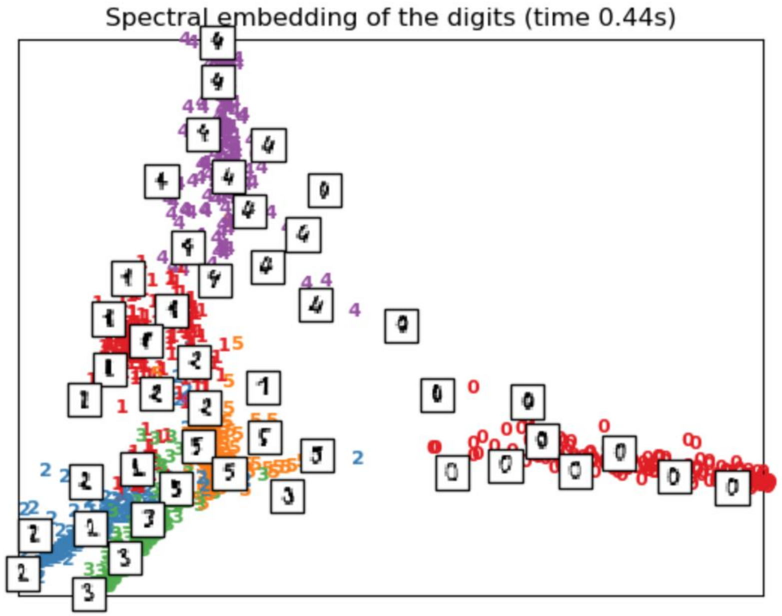
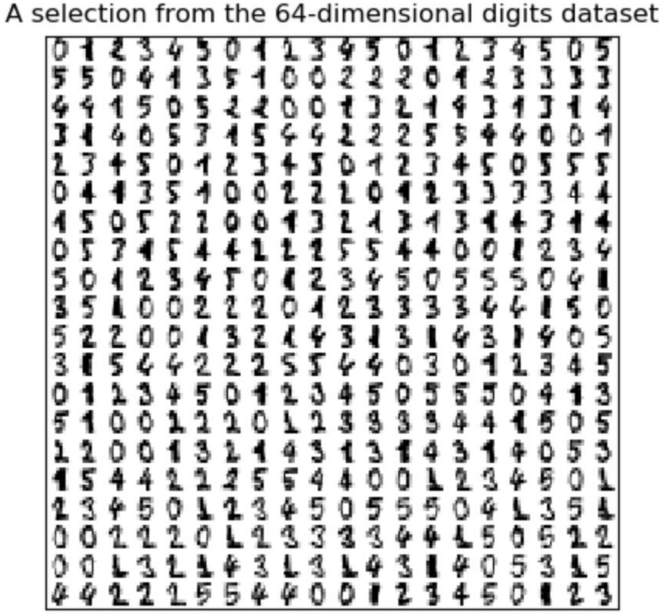


Image source: sklearn example

$x = [x_1, \dots, x_{64}]$ spectral embedding $w = [w_1, w_2]$

Consider the digit dataset. This dataset can be thought of as a high-dimensional data with $d = 64$.

Spectral Embedding :example



spectral embedding

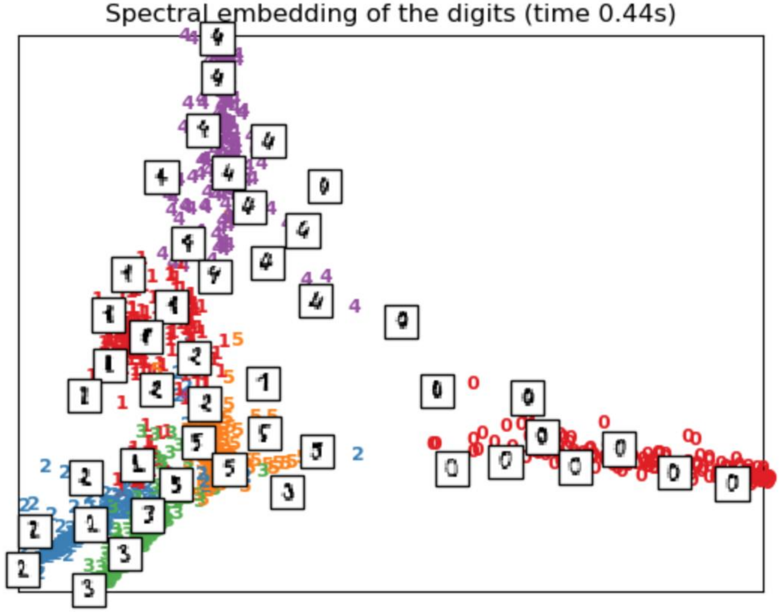


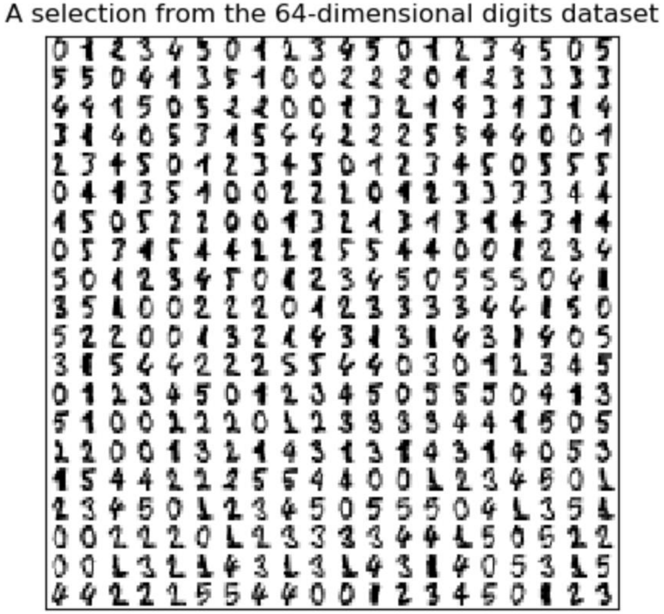
Image source: sklearn example

$x = [x_1, \dots, x_{64}]$ — spectral embedding → $w = [w_1, w_2]$

Consider the digit dataset. This dataset can be thought of as a high-dimensional data with $d = 64$.

So every image can be thought of as a vector $x = [x_1, \dots, x_{64}]$

Spectral Embedding :example



spectral embedding

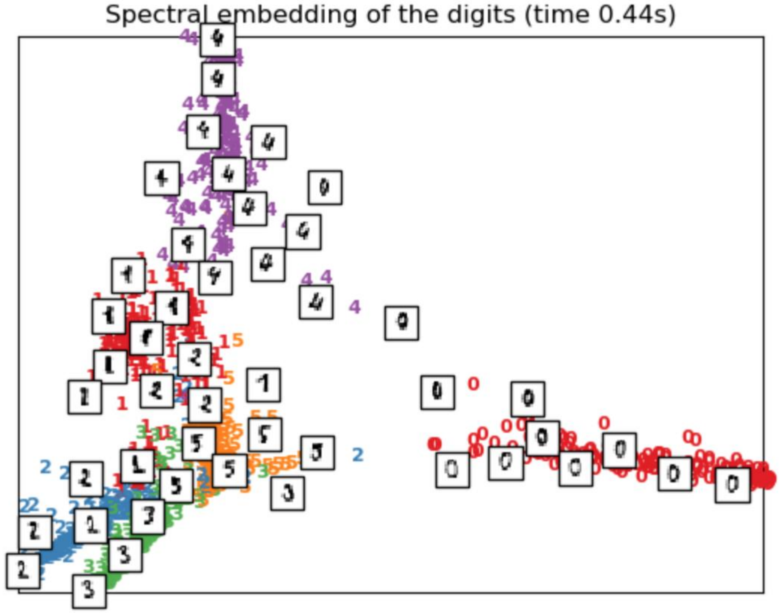


Image source: sklearn example

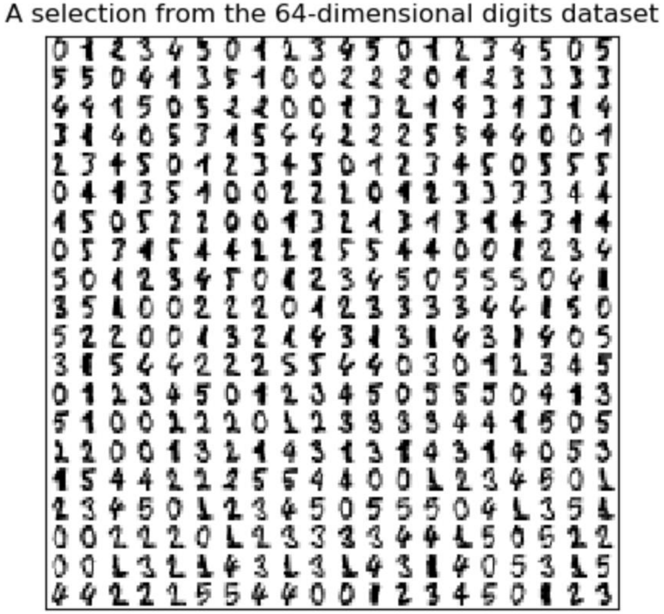
$x = [x_1, \ldots, x_{64}]$ spectral embedding $w = [w_1, w_2]$

Consider the digit dataset. This dataset can be thought of as a high-dimensional data with $d = 64$.

So every image can be thought of as a vector $x = [x_1, \ldots, x_{64}]$

Spectral embedding assigns to the point x new coordinates $w = [w_1, \ldots, w_k]$ where $k \leq 64$. Usually we choose $d << k$. In the example above we choose $k = 2$.

Spectral Embedding :example



spectral embedding

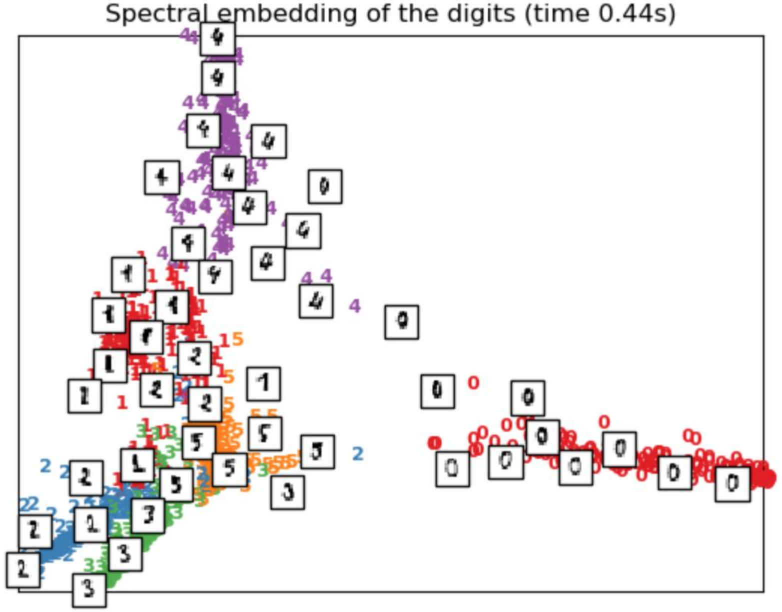


Image source: sklearn example

$x = [x_1, \dots, x_{64}]$ spectral embedding $w = [w_1, w_2]$

Consider the digit dataset. This dataset can be thought of as a high-dimensional data with $d = 64$.

So every image can be thought of as a vector $x = [x_1, \dots, x_{64}]$

Spectral embedding assigns to the point x new coordinates $w = [w_1, \dots, w_k]$where $k \leq 64$. Usually we choose $d << k$. In the example above we choose $k = 2$.

But how exactly do we construct this new vector w ?

Spectral Embedding : general steps

- Construct a similarity graph phase : A similarity graph for the data X is chosen from the many available neighborhood graphs we studied in earlier lectures.

- The spectral embedding phase :In this step we use the eigenvectors of the Laplacian of the similarity graph to construct new coordinates.

Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V

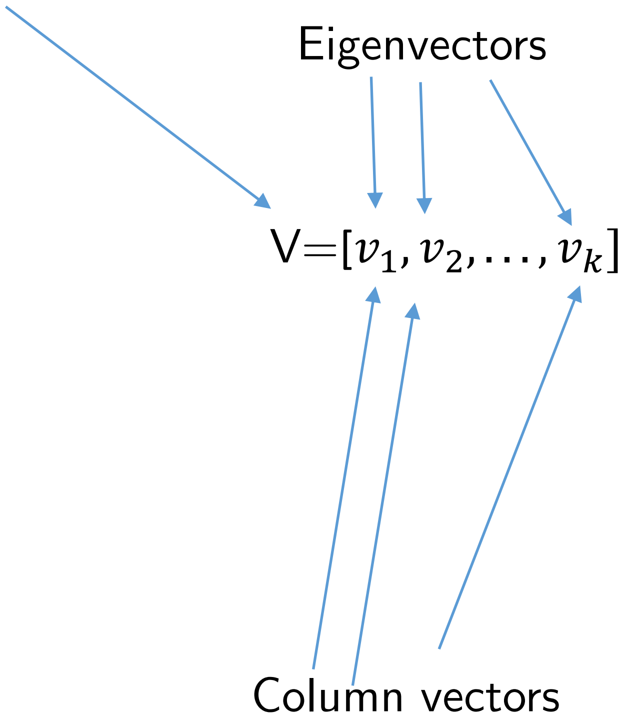
Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V



Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.
- Compute the Laplacian of the graph L(G).
- Compute top k eigenvectors of L and place them as columns in a matrix V
- Form W from V by normalizing the rows of W (making every row a unit vector).

Eigenvectors

$$V=[v_1, v_2, \ldots, v_k]$$

Column vectors

$$v_{ij} = \frac{u_{ij}}{\left(\sum_{l=1}^{k} u_{il}^2\right)^2}$$

Spectral Embedding : Algorithm

Input : a data set X consists of a points in R^d. The number of dimensions $k \leq d$

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

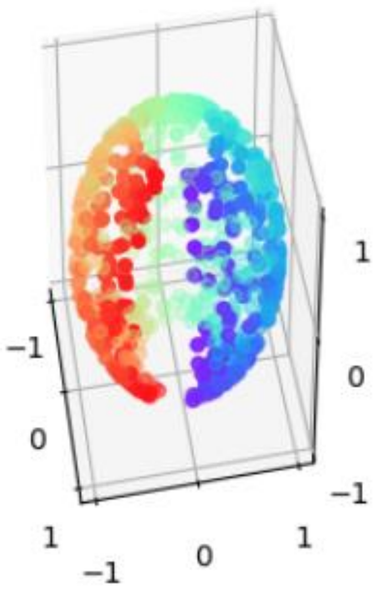
- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V

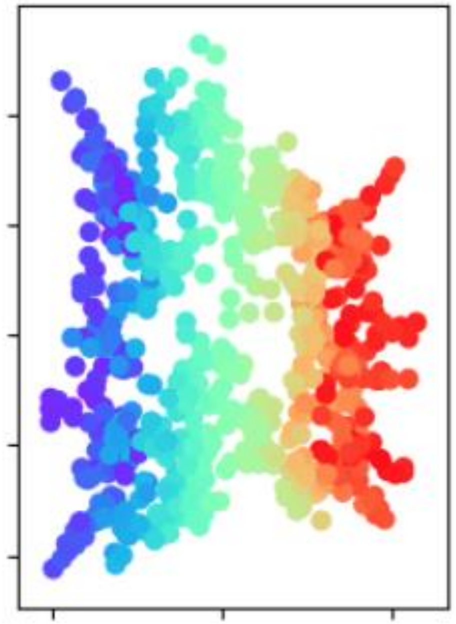
- Form W from V by normalizing the rows of W (making every row a unit vector).

- Each row w_i in the matrix W is, by definition, the spectral embedding of the point x_i from the original data

Spectral Embedding :more examples



spectral embedding



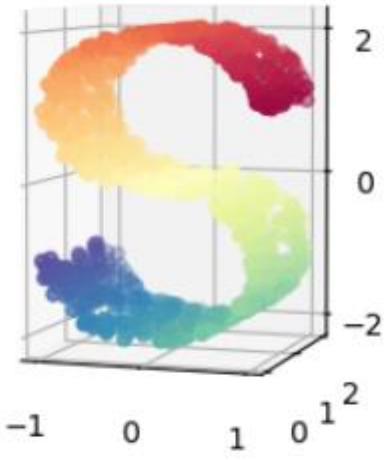
x=[x1,x2,x3]

w=[w1,w2]

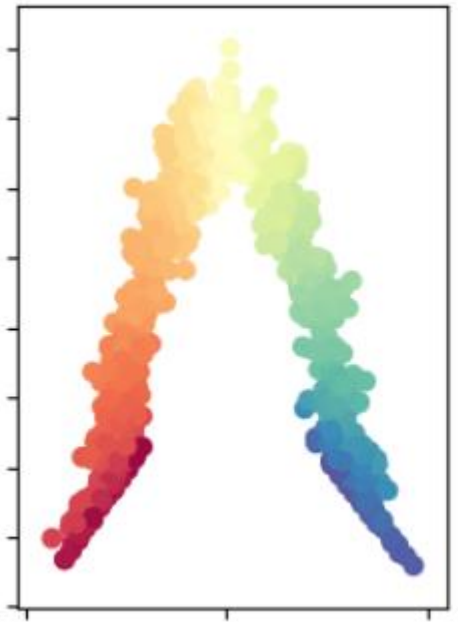
Image source: sklearn example

In general the results of spectral embedding can better reveal or exaggerate useful underlying structures in the input data.

Spectral Embedding :more examples



spectral embedding

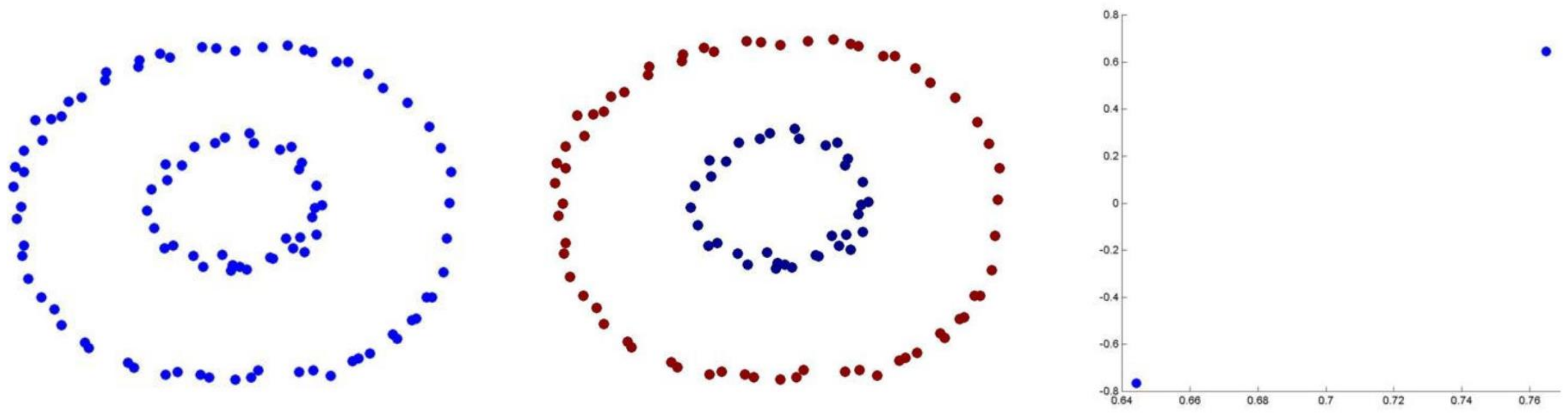


x=[x1,x2,x3]

w=[w1,w2]

Image source: sklearn example

Spectral Clustering example



(a) (b) (c) Image [source]()

- In the above example the dataset (a) is mapped to its clustering embedding (figure c) which can be trivially classified using k-means (figure c).
- This example shows that via by a transforming the data into the spectral domain, certain intrinsic shape structures are revealed.

Spectral Clustering : general steps

- Construct a similarity graph phase : A similarity graph for the data X is chosen among the many available neighborhood graphs we studied in earlier lectures.

- The spectral embedding phase :In this step we use the eigenvectors of the Laplacian of the similarity graph to construct new coordinates for the points in X in which the clusters are more obvious.

- Using one of the classical clustering algorithms to cluster the points in the spectral space and induce this clustering results to the original points.

Spectral Clustering : general steps

- Construct a similarity graph phase : A similarity graph for the data X is chosen among the many available neighborhood graphs we studied in earlier lectures.

- The spectral embedding phase :In this step we use the eigenvectors of the Laplacian of the similarity graph to construct new coordinates for the points in X in which the clusters are more obvious.

- Using one of the classical clustering algorithms to cluster the points in the spectral space and induce this clustering results to the original points.

Spectral embedding

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V

- Form W from V by normalizing the rows of W (making every row a unit vector).

- Run K-means to cluster the row vectors of W

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.

- Compute the Laplacian of the graph L(G).

- Compute top k eigenvectors of L and place them as columns in a matrix V

- Form W from V by normalizing the rows of W (making every row a unit vector).

- Run K-means to cluster the row vectors of W

- x_i is assigned to cluster α iff row i of W is assigned to cluster α.

Spectral Clustering : Algorithm

Input : a data set X consists of a points in R^d. The number of clusters k

- Construct a similarity graph G=G(X) of the data. This can be the k-NN graph for instance.
- Compute the Laplacian of the graph L(G).
- Compute top k eigenvectors of L and place them as columns in a matrix V
- Form W from V by normalizing the rows of W (making every row a unit vector).

(Spectral embedding)

- Run K-means to cluster the row vectors of W
- x_i is assigned to cluster α iff row i of W is assigned to cluster α.

More Spectral Clustering: examples

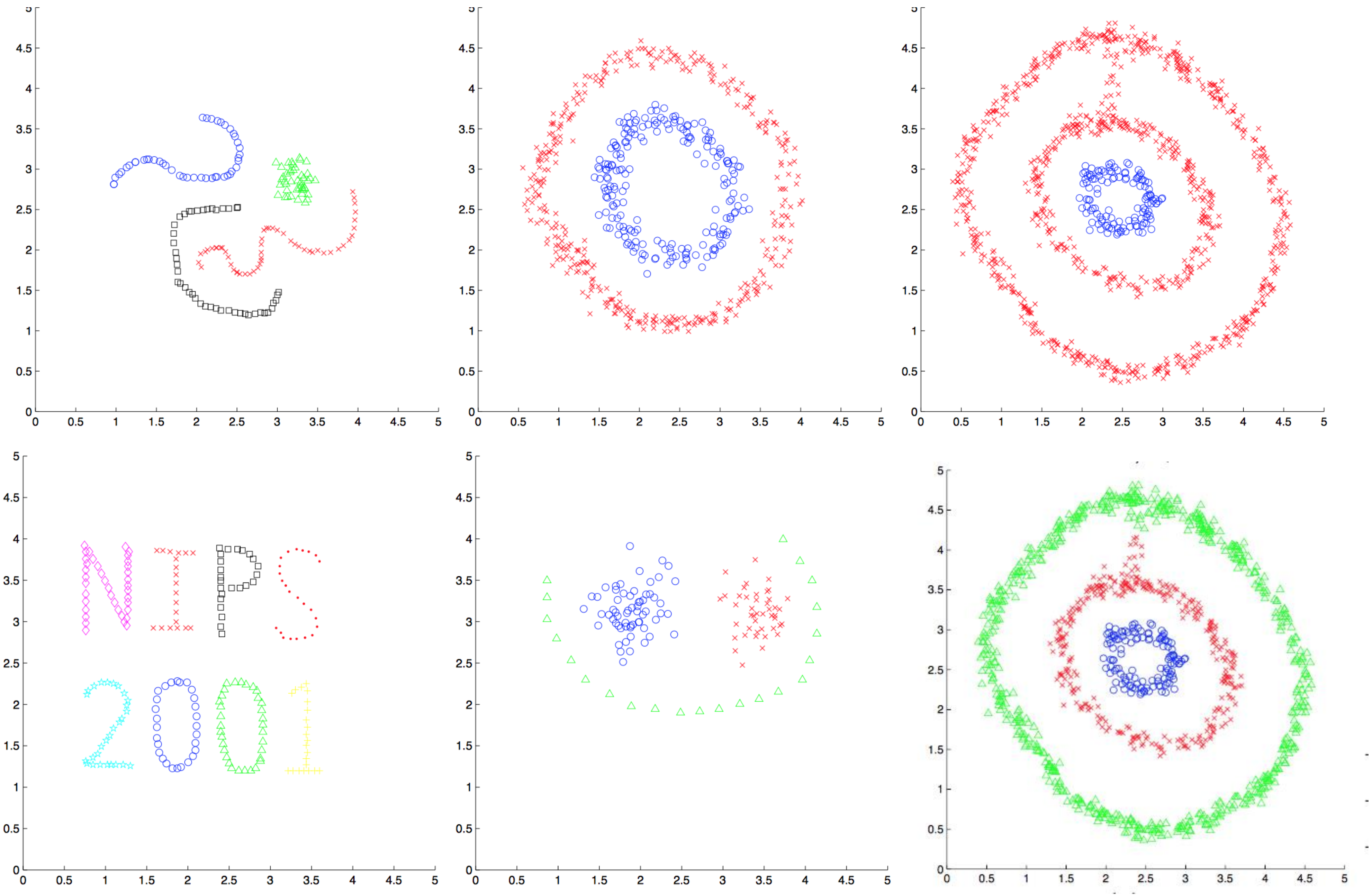
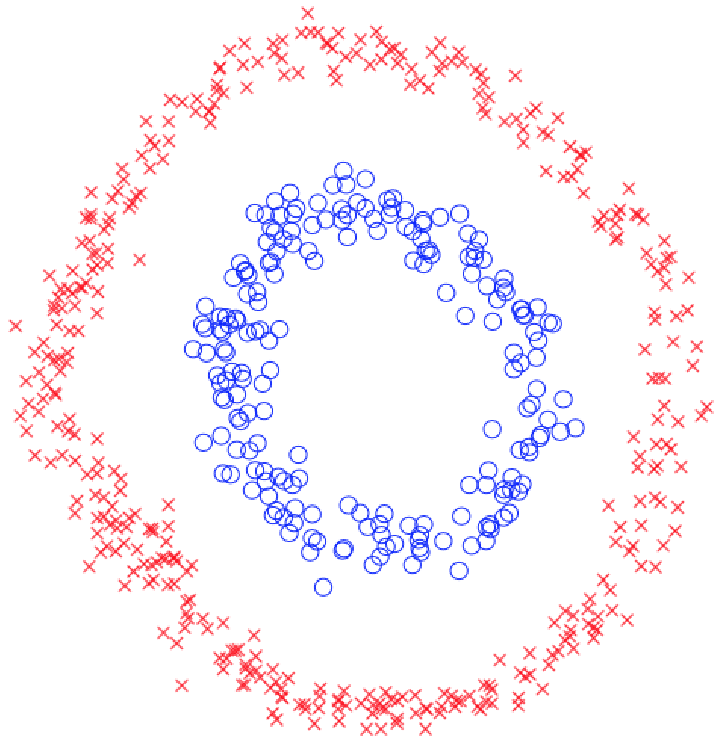
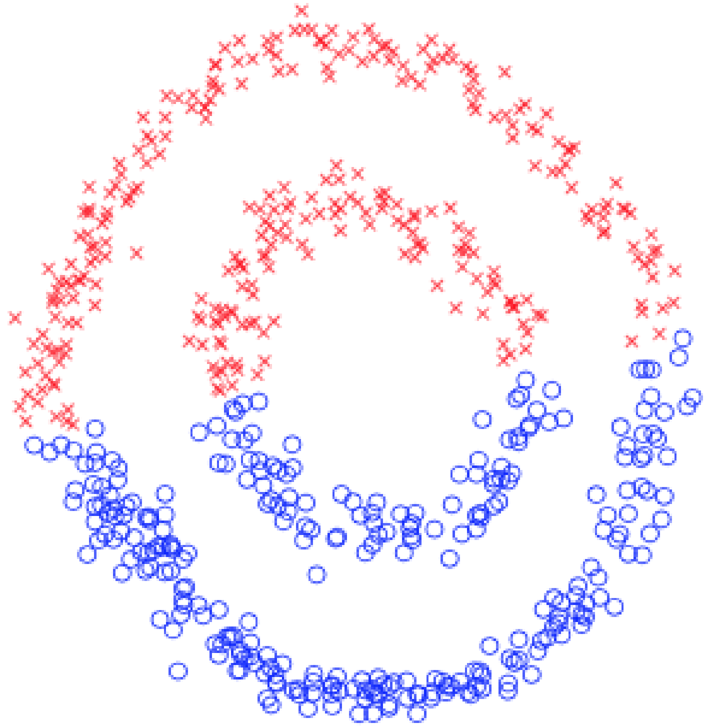


image source

Spectral Clustering vs K-means



Spectral clustering

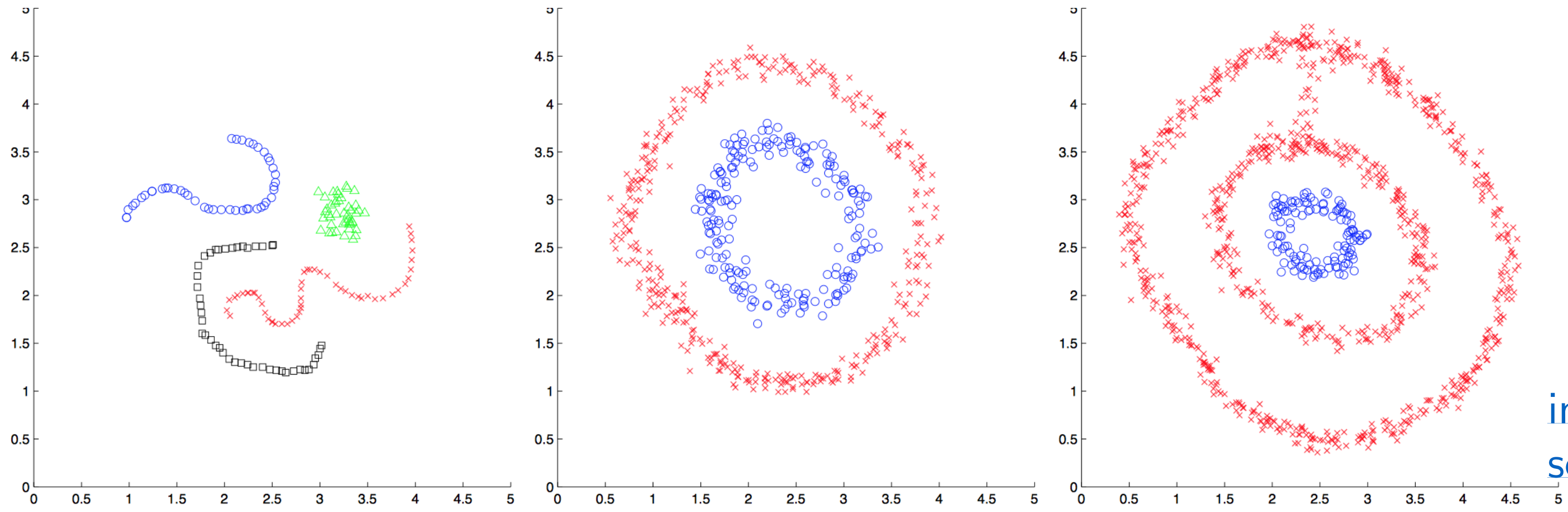


K-means clustering

image source

Spectral Clustering/Embedding: remarks

Spectral clustering does not put any assumption on the shape of data.



[image source](#)

Spectral clustering can be used to for clustering non-linearly separable data.

The results of spectral embedding can better reveal or exaggerate useful underlying structures in the input data.