Spectral Embedding and Locally Linear Embedding

MUSTAFA HAJIJ

Manifolds

Mathematically, a manifold is a space that looks locally like a patch. The dimension of this patch is called the dimension of the manifold.

Manifolds

Mathematically, a manifold is a space that looks locally like a patch. The dimension of this patch is called the dimension of the manifold.

Surfaces are examples of manifolds. In particular,

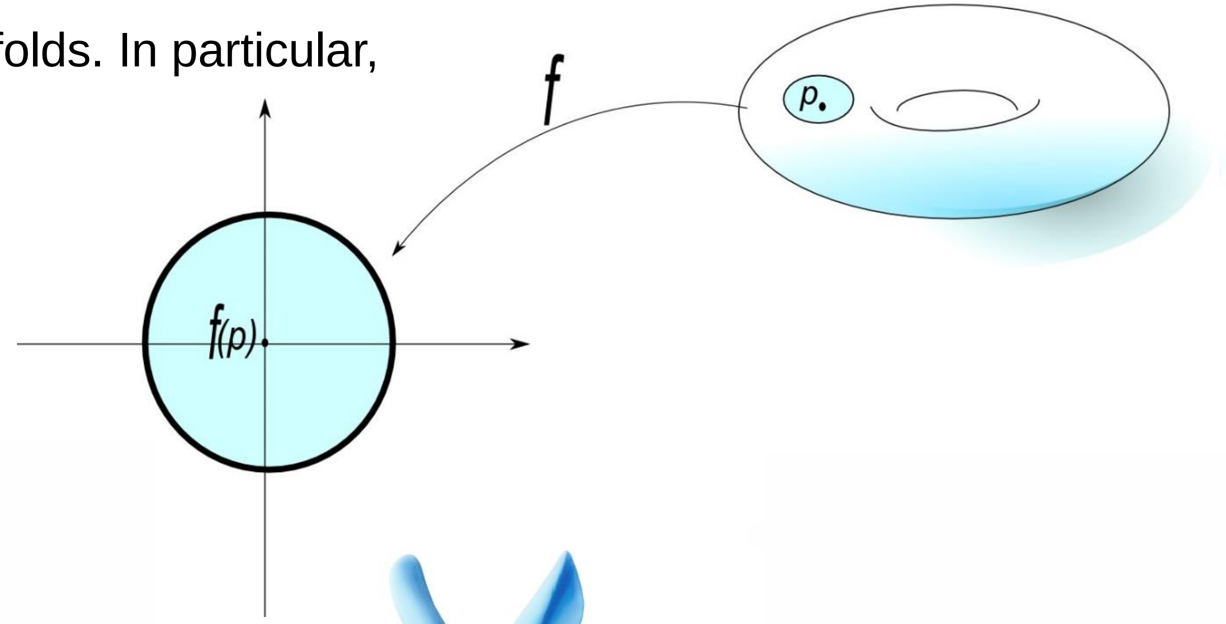
surfaces are 2-manifolds.

Manifolds

Mathematically, a manifold is a space that looks locally like a patch. The dimension of this patch is called the dimension of the manifold.

Surfaces are examples of manifolds. In particular,

surfaces are 2-manifolds.

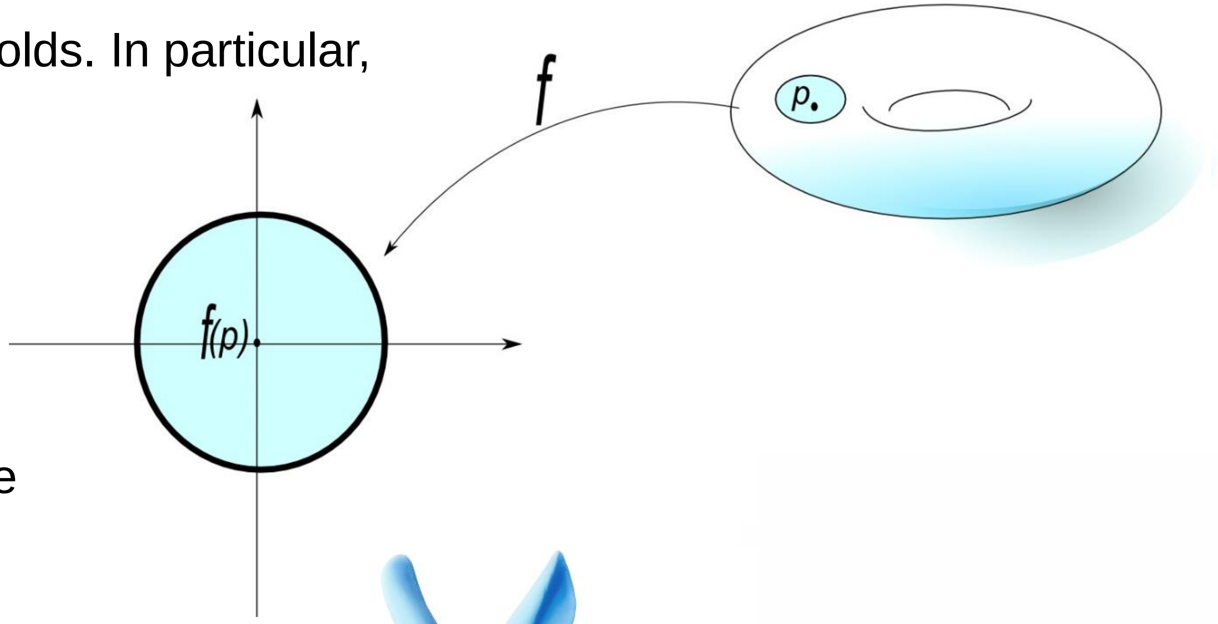


Manifolds

Mathematically, a manifold is a space that looks locally like a patch. The dimension of this patch is called the dimension of the manifold.

Surfaces are examples of manifolds. In particular,

surfaces are 2-manifolds.



For every point *p* on the surface

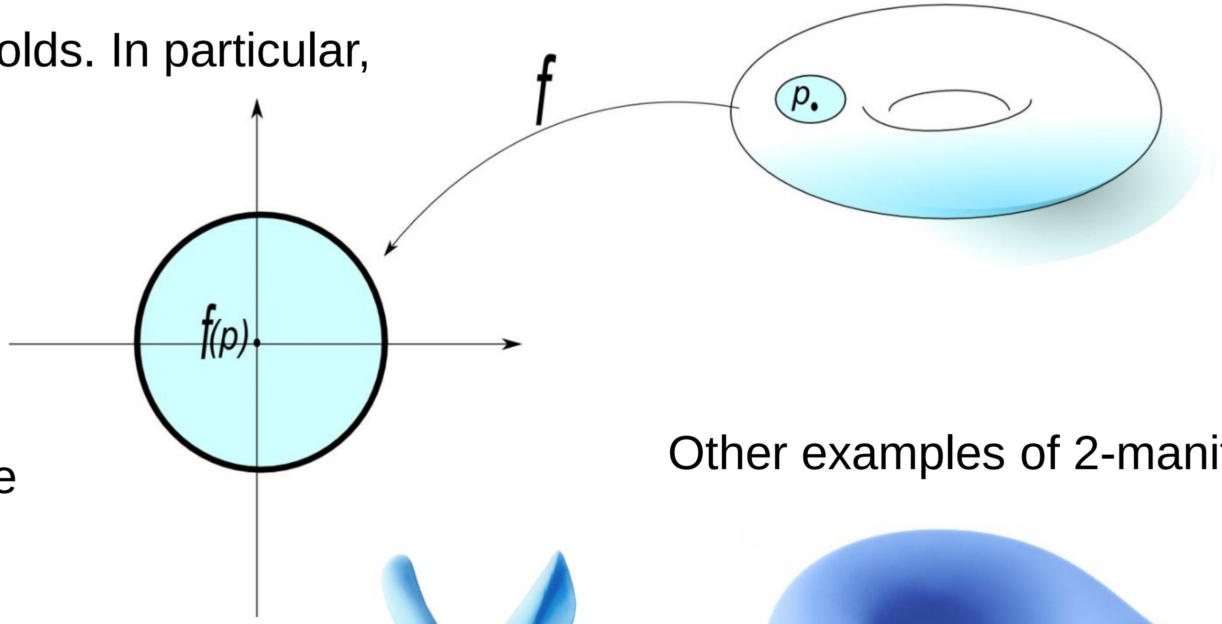
there exists a small disk

that is mapped via a map *f* to the unit disk in the plane.

Manifolds

Mathematically, a manifold is a space that looks locally like a patch. The dimension of this patch is called the dimension of the manifold.

Surfaces are examples of manifolds. In particular, surfaces are 2-manifolds.

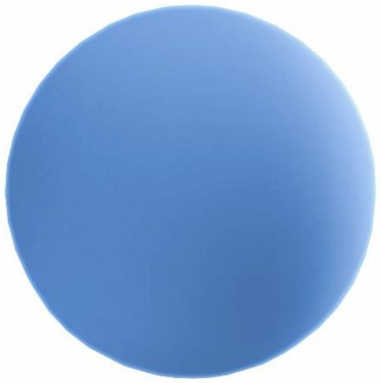


For every point *p* on the surface

there exists a small disk

that is mapped via a map *f* to the unit disk in the plane.

Other examples of 2-manifolds



Locally linear embedding

PCA and MDS, do not give good results in nonlinear dimensionality reduction problems.

Locally linear embedding

PCA and MDS, do not give good results in nonlinear dimensionality reduction problems.

LLE recovers the global data by collecting local information surrounding each point in the data and then stitch these information together

Locally linear embedding

PCA and MDS, do not give good results in nonlinear dimensionality reduction problems.

LLE recovers the global data by collecting local information surrounding each point in the data and then stitch these information together

Locally linear embedding does not require the computation of the pair-wise distance matrix of the data such as ISOMAP and MDS.

LLE

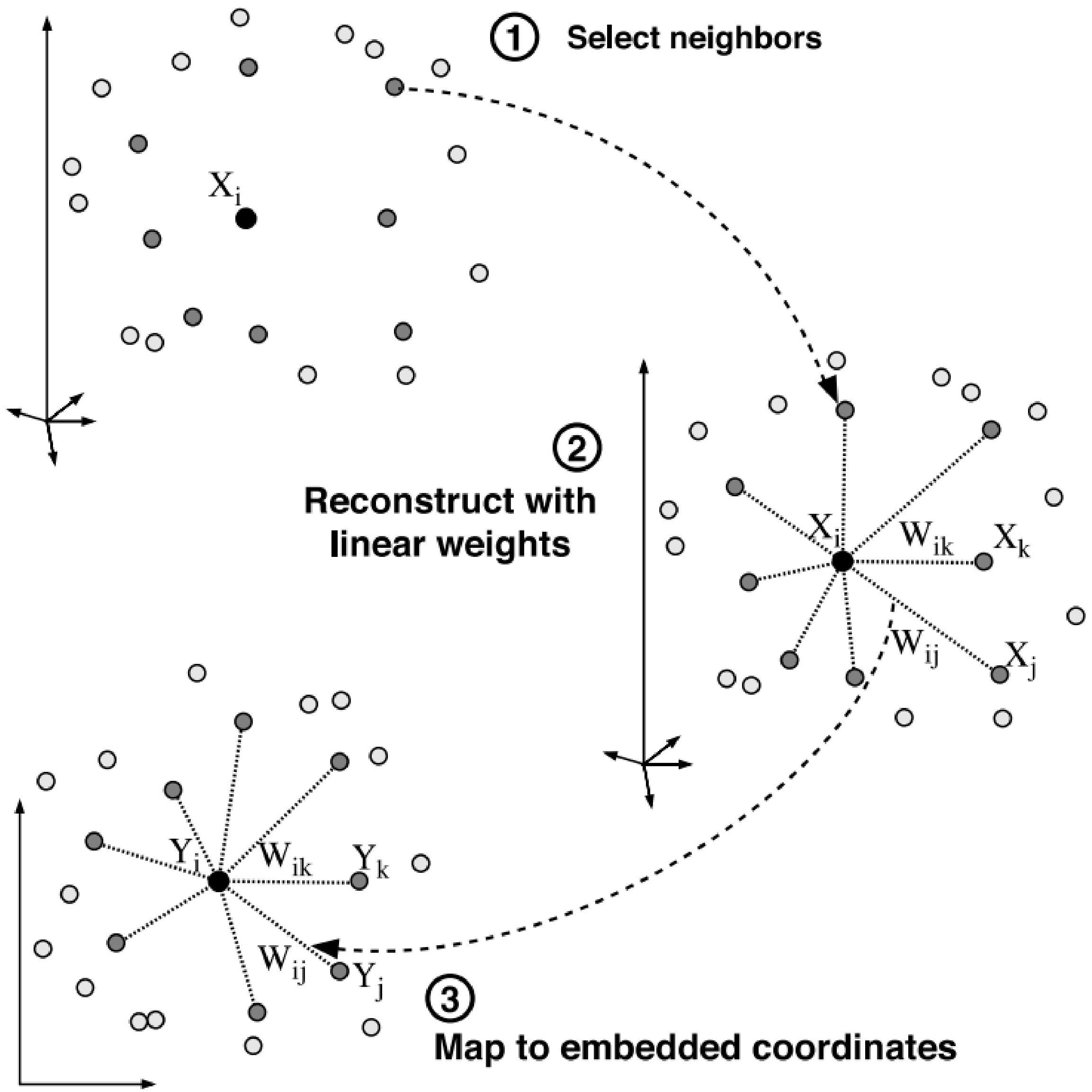


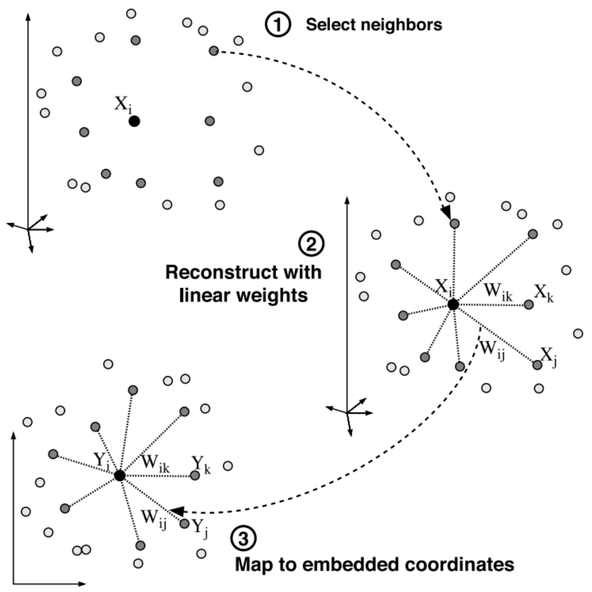
Image source, [original paper](#)

Basic idea

We need to optimize two objective functions in LLE :

I- $$\varepsilon(W) = \sum_i \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2$$

Here we are given the points X_i and we are optimizing the weights W_{ij}



Basic idea

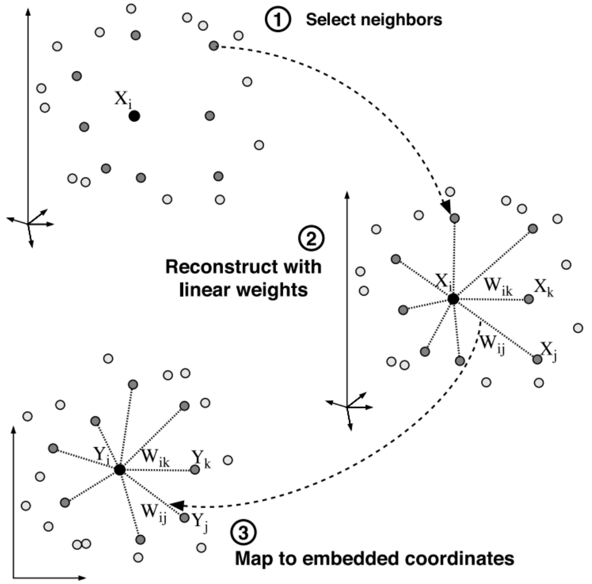
We need to optimize two objective functions in LLE :

I- $$\varepsilon(W) = \sum_i \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2$$

Here we are given the points X_i and we are optimizing the weights W_{ij}

II- $$\Phi(Y) = \sum_i \left| \vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j \right|^2$$

Here we fix the weights W_{ij} while optimizing the coordinates Y_i



Basic idea

Steps :

1- Identify the k-NN graph

Basic idea

Steps :

1- Identify the k-NN graph

2-Compute Wij such that $\sum_{i} \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2$ is minimal

Subject to the constrains :

(a) Wij=0 when Xj does not belong to the set of neighbors of Xi;

(b) $\sum_{j=1}^{k} W_{ij} = 1$

Basic idea

Steps :

1- Identify the k-NN graph

2-Compute Wij such that $$\sum_{i} \left| \vec{X}_i - \Sigma_j W_{ij} \vec{X}_j \right|^2$$ is minimal

Subject to the constrains :

(a) Wij=0 when Xj does not belong to the set of neighbors of Xi;

(b) $\sum_{j=1}^{k} W_{ij} = 1$

3-Compute Yi such that $$\sum_{i} \left| \vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j \right|^2$$ is minimal

Details

Define $V_i = [X_{i(1)} X_{i(2)} \quad \ldots X_{i(k)}]$

Details

Define $V_i = [X_{i(1)} X_{i(2)} \quad \dots X_{i(k)}]$

$$W_i = [W_{i1} W_{i2} \quad \dots W_{ik}]^T$$

Details

Define $V_i = [X_{i(1)} X_{i(2)} \quad \dots X_{i(k)}]$

$$W_i = [W_{i1} W_{i2} \quad \dots W_{ik}]^T$$

Hence $$\sum_{j=1}^{k} W_{ij} X_{i(j)} = V_i W_i$$

Details

Define $V_i = [X_{i(1)} X_{i(2)} \ \ \ldots X_{i(k)}]$

$$W_i = [W_{i1} W_{i2} \ \ \ldots W_{ik}]^T$$

Hence $$\sum_{j=1}^{k} W_{ij} X_{i(j)} = V_i W_i$$

So the objective function now can be written as :

$$|X_i - V_i W_i|^2$$

Details

Define $V_i = [X_{i(1)} X_{i(2)} \ \ \ldots X_{i(k)}]$

$$W_i = [W_{i1} W_{i2} \ \ \ldots W_{ik}]^T$$

Hence $$\sum_{j=1}^{k} W_{ij} X_{i(j)} = V_i W_i$$

So the objective function now can be written as :

$$|X_i - V_i W_i|^2$$

Now construct a matrix

$$[X_i \ \ X_i \ldots X_i]$$

Details

Define $V_i = [X_{i(1)} X_{i(2)} \ \ldots X_{i(k)}]$

$$W_i = [W_{i1} W_{i2} \ \ldots W_{ik}]^T$$

Hence $$\sum_{j=1}^{k} W_{ij} X_{i(j)} = V_i W_i$$

So the objective function now can be written as :

$$|X_i - V_i W_i|^2$$

Now construct a matrix

$$[X_i \ \ X_i \ldots X_i]$$

And write

$$[X_i \ \ X_i \ldots X_i] = X_i e^T \text{ where } \quad e^T = [1, \ldots, 1]_{1*k}$$

Details

Note that $X_i = \mathrm{X_i e^T W_i}$ Hence, write the objective function:

$$|X_i e^T W_i - V_i W_i|^2$$

Details

Note that $X_i = X_i e^T W_i$ Hence, write the objective function:

$$|X_i e^T W_i - V_i W_i|^2 = |(X_i e^T - V_i) W_i|^2$$

Details

Note that $X_i = X_i e^T W_i$ Hence, write the objective function:

$$|X_i e^T W_i - V_i W_i|^2 = |(X_i e^T - V_i)W_i|^2$$

$$= W_i^T (X_i e^T - V_i)^T * (X_i e^T - V_i)W_i$$

Details

Note that $X_i = X_i e^T W_i$ Hence, write the objective function:

$$|X_i e^T W_i - V_i W_i|^2 = |(X_i e^T - V_i) W_i|^2$$

$$= W_i^T (X_i e^T - V_i)^T * (X_i e^T - V_i) W_i$$

Define $$G := (X_i e^T - V_i)^T (X_i e^T - V_i)$$

Details

Note that $X_i = X_i e^T W_i$ Hence, write the objective function:

$$|X_i e^T W_i - V_i W_i|^2 = |(X_i e^T - V_i) W_i|^2$$

$$= W_i^T (X_i e^T - V_i)^T * (X_i e^T - V_i) W_i$$

Define $$G := (X_i e^T - V_i)^T (X_i e^T - V_i)$$

$The\ matrix\ G$ is called the Gram matrix

Details

Hence, write the objective function:

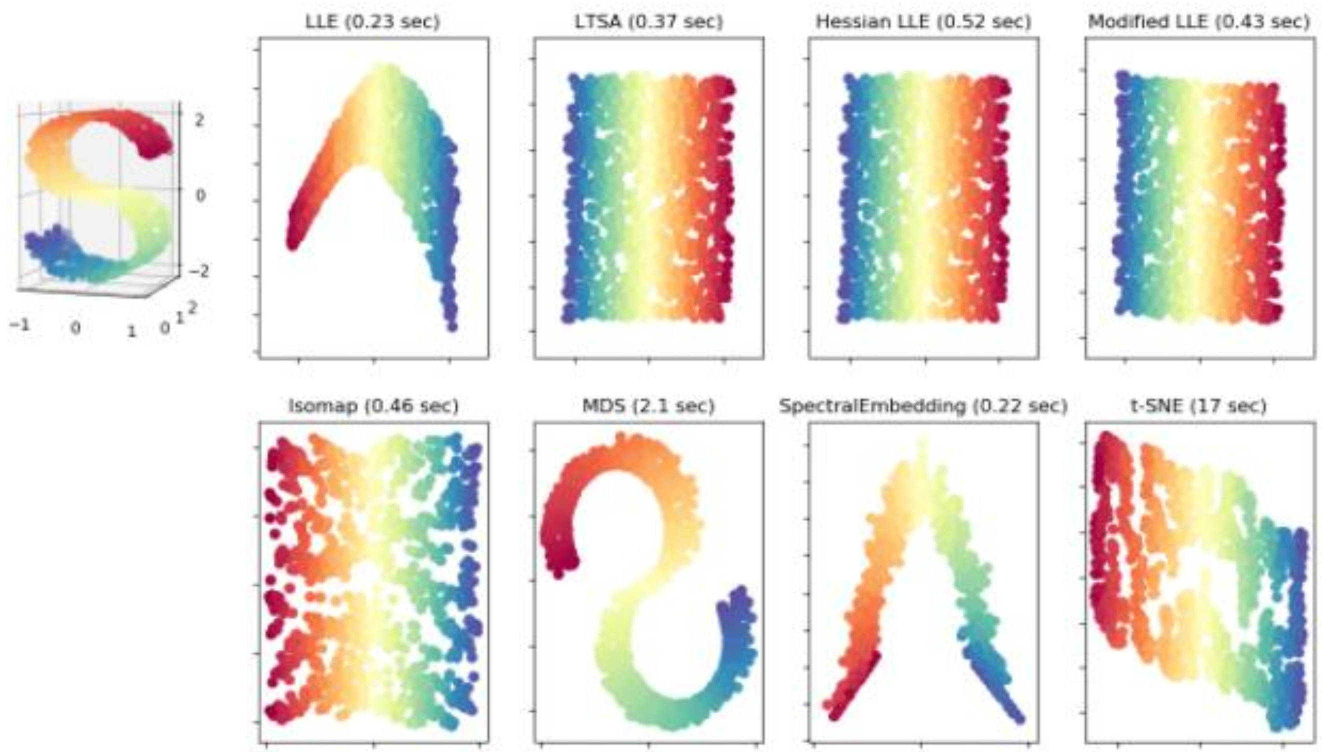
$$min \sum_{i} W_i^T G \mathrm{W_i}$$

Subject to $\sum_{j=1}^{k} W_{ij} = 1$

This is a well-known optimization problem and it can be solved using least square methods

LLE and its variation in Sklearn

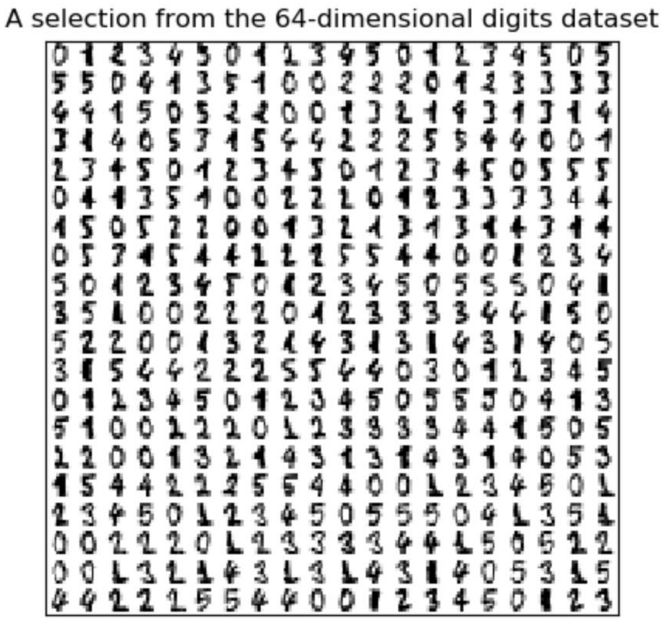
MDS is implemented in [Sklearn]



[Example]

Part II : Spectral Embedding

Spectral Embedding :example



spectral embedding

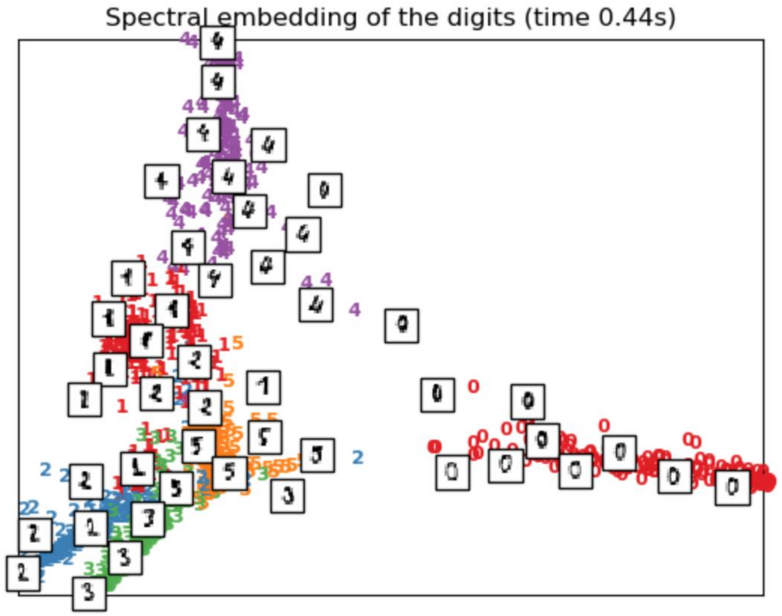
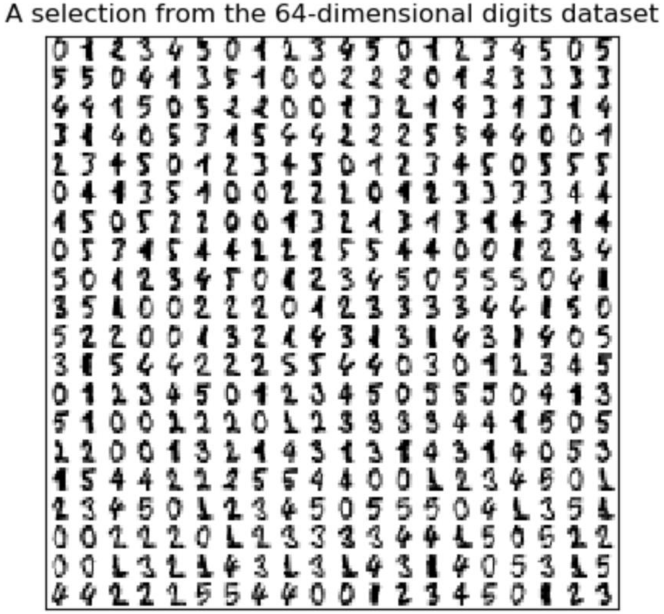


Image source: sklearn example

- 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

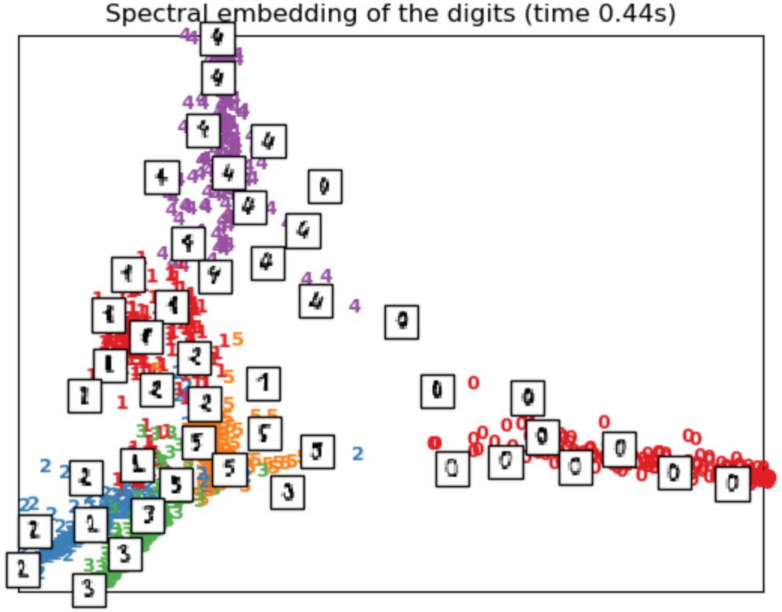


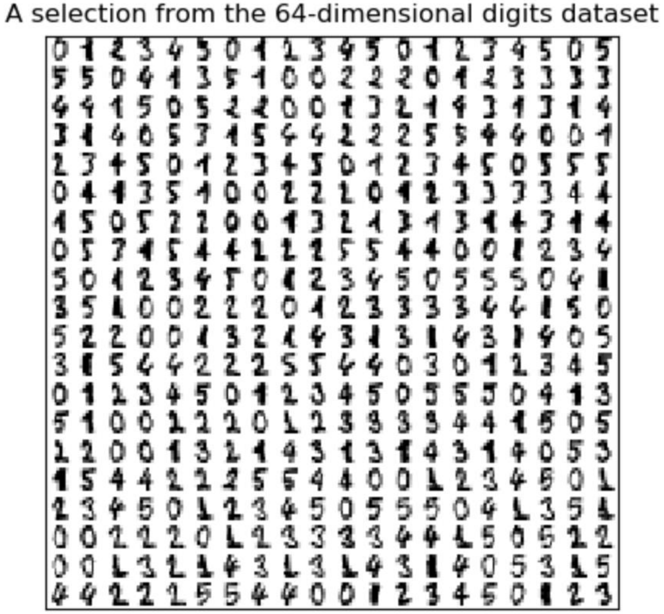
Image source: sklearn example

spectral embedding

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

$w = [w_1, w_2]$

Spectral Embedding :example



spectral embedding

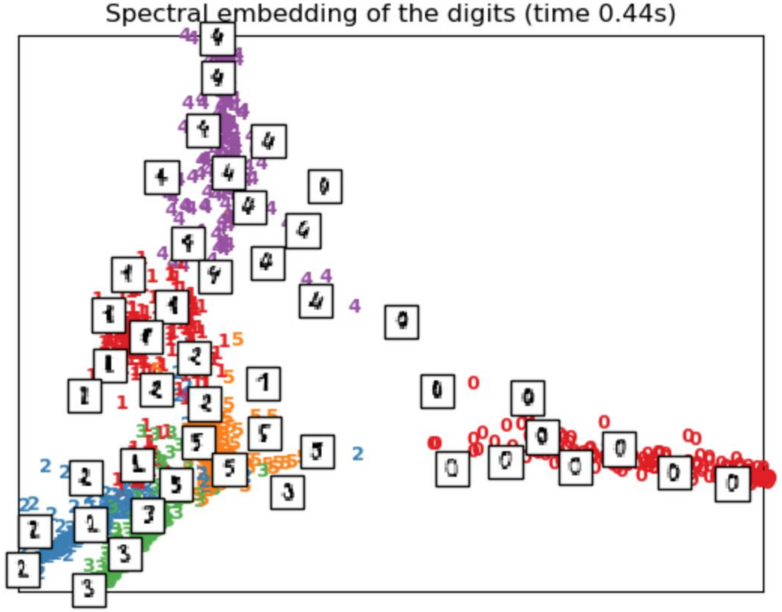


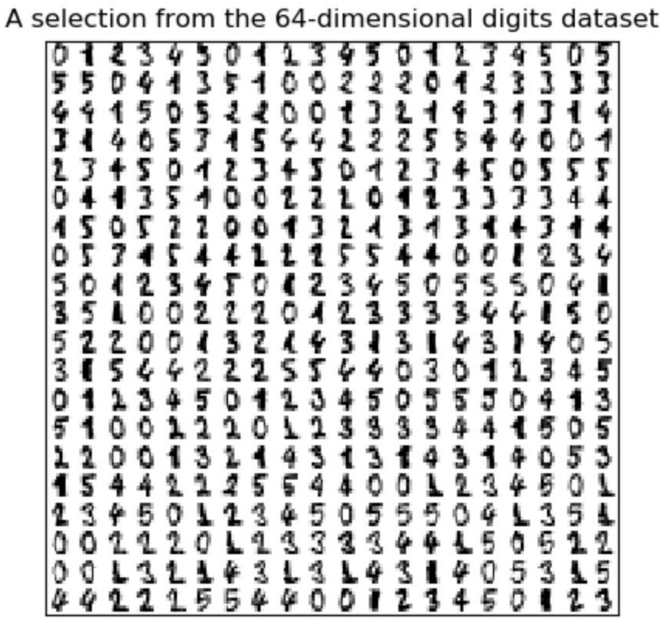
Image source: sklearn example

spectral embedding

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

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

Spectral Embedding :example



spectral embedding

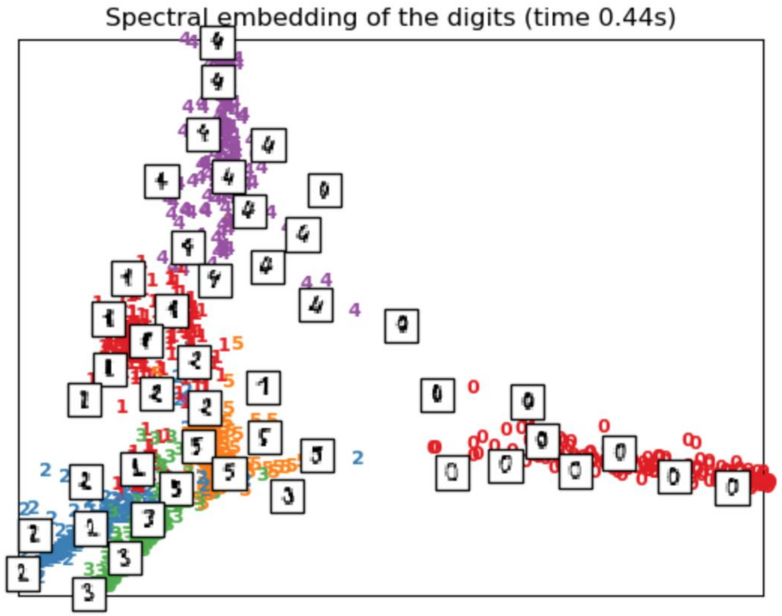


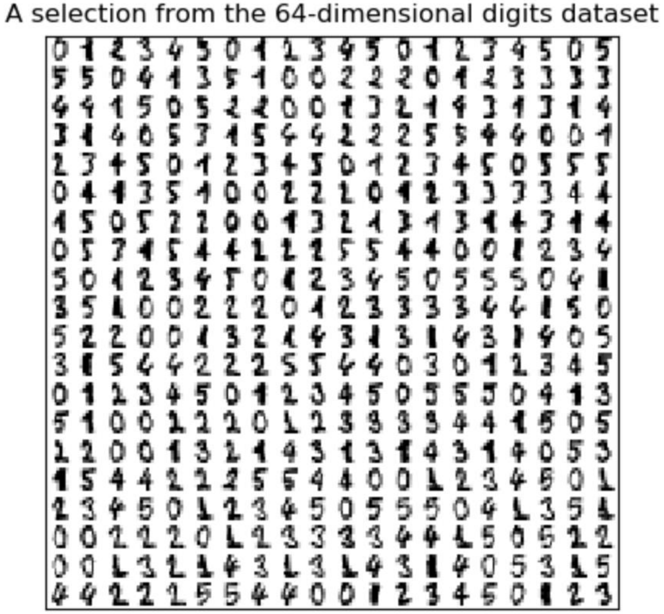
Image source: sklearn example

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



spectral embedding

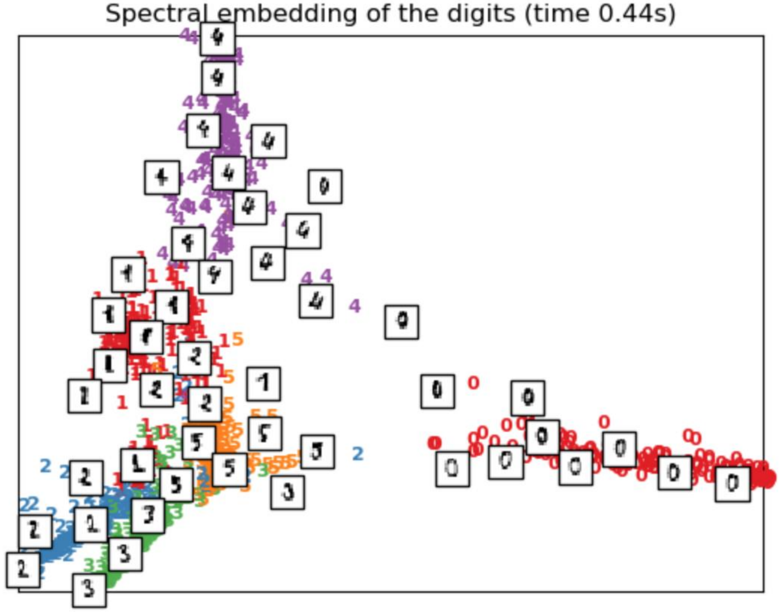


Image source: sklearn example

spectral embedding

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

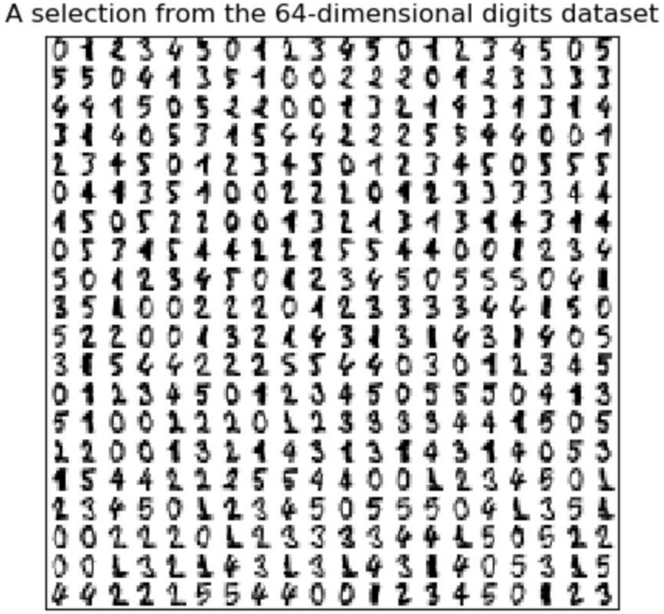
$w = [w_1, w_2]$

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 $k = 2$.

Spectral Embedding :example



spectral embedding

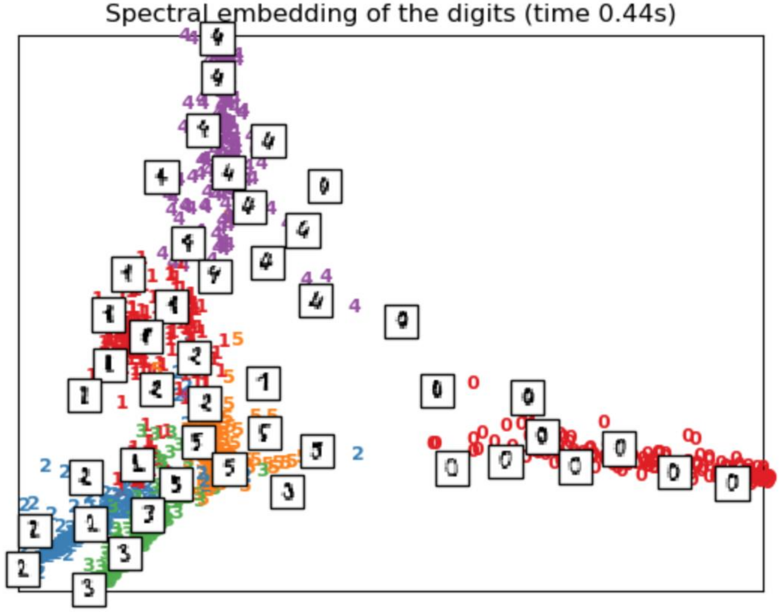


Image source: sklearn example

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

Eigenvectors

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

Column vectors

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.

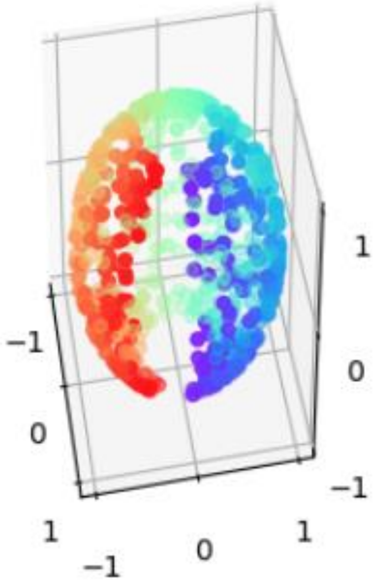
- Compute the Laplacian of the graph L(G).

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

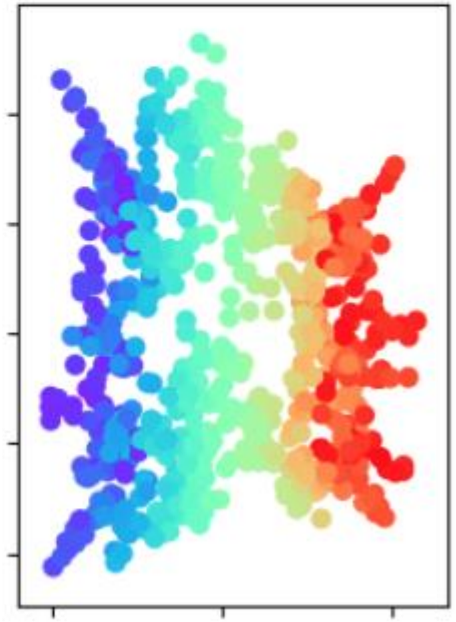
- Form W from V by normalizing the rows of W (making every row a unit vector).

- 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



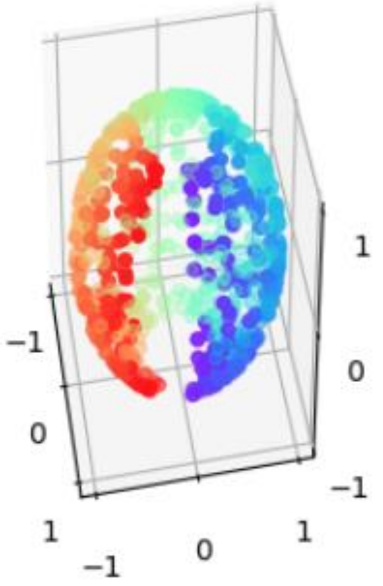
x=[x1,x2,x3]

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

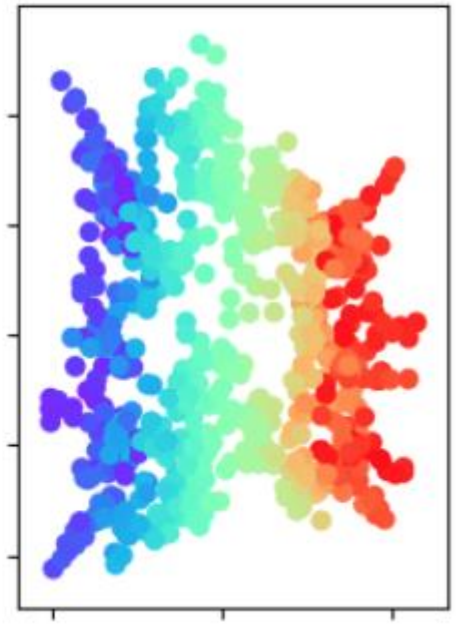


Image source: sklearn example

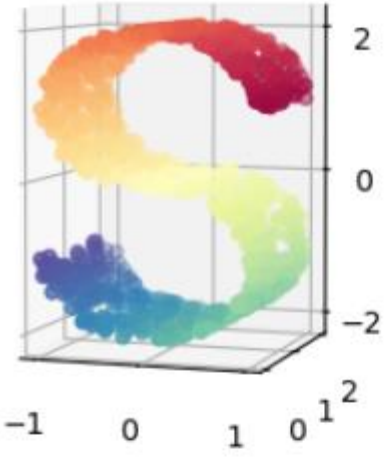
x=[x1,x2,x3]

w=[w1,w2]

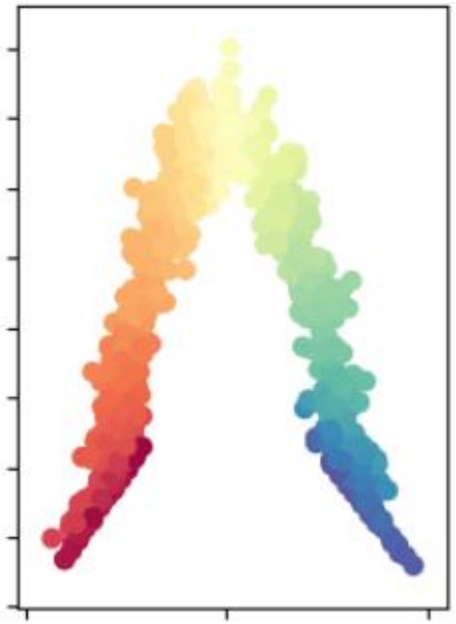
In general the results of spectral embedding can better reveal or exaggerate useful underlying structures in the input data.

Spectral embedding usually put the points that are highly similar closer to each other and the points that are dissimilar far away from each other

Spectral Embedding :more examples



spectral embedding



x=[x1,x2,x3]

w=[w1,w2]

Image source: sklearn example

Spectral Embedding :graphs

We can use spectral embedding to give an embedding for a graph.

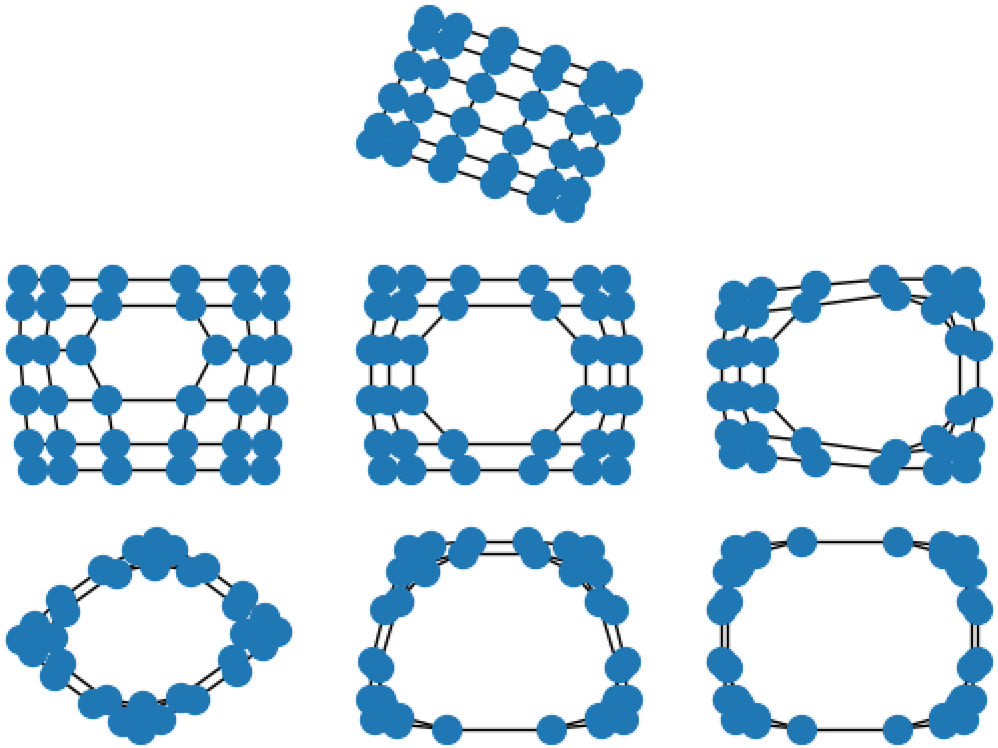


Image source:
networks example