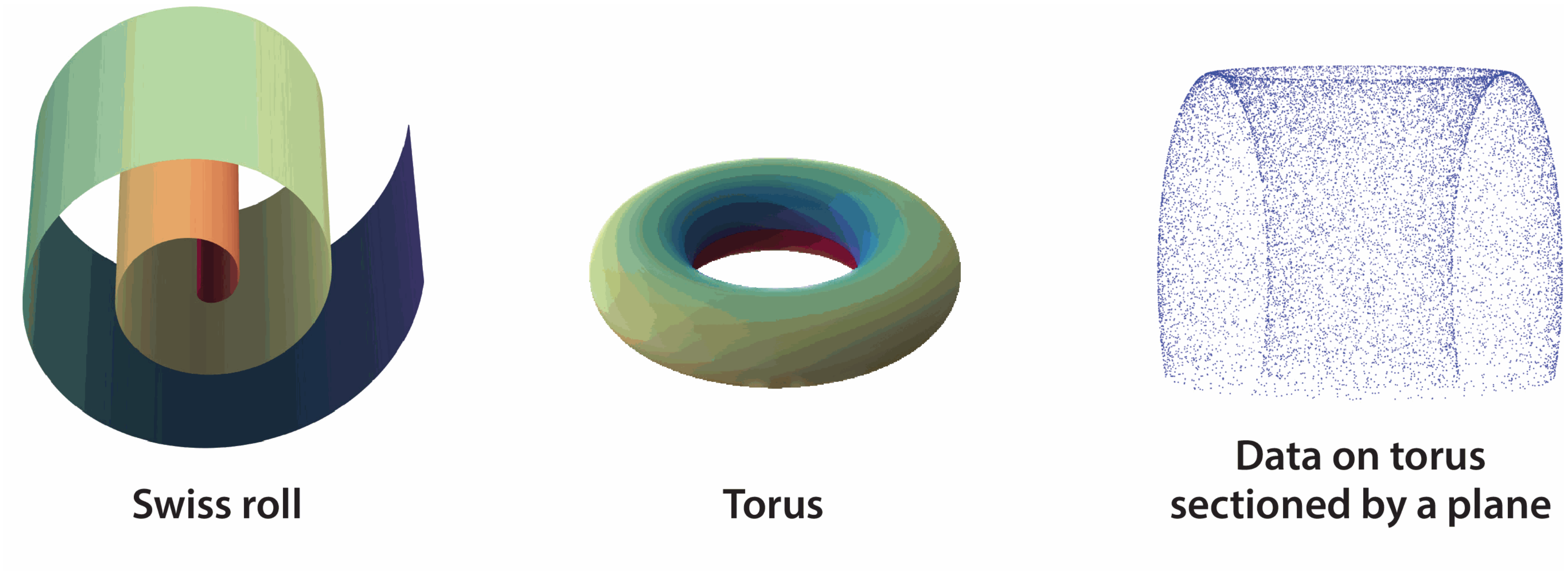
# *MACHINE LEARNING*

## Spectral Clustering & Nonlinear Embeddings

## Interactive exercise session

# Manifold Learning



Meilă M, Zhang H. 2024
*Annu. Rev. Stat. Appl.* 11:393–417

Figure 2 Examples of manifolds with intrinsic dimension d = 2. (Left) A Swiss roll. (Middle) A torus (hollow). (Right) 1,000 points sampled from a torus sectioned by a plane.

*Dataset resides on an unknown nonlinear manifold.*

# Manifold Learning

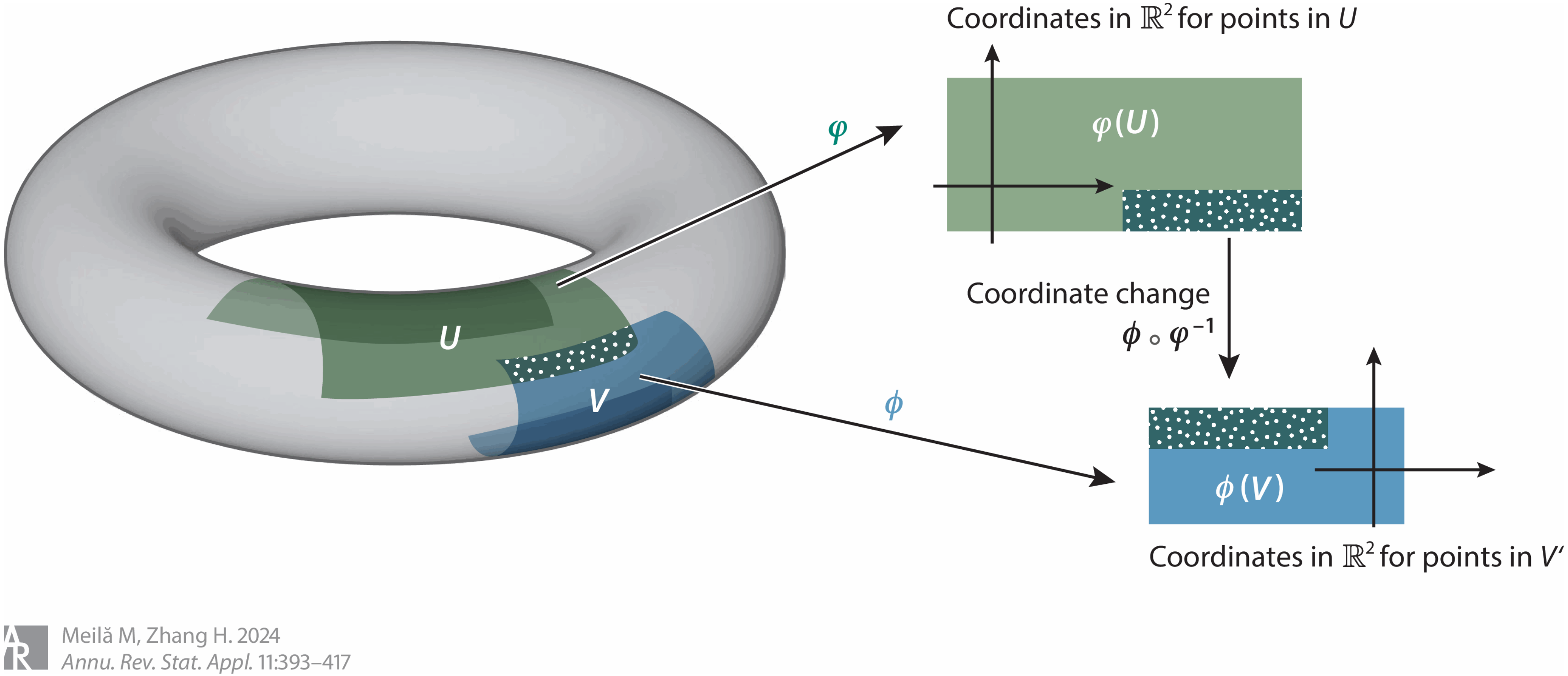
Principle:

Dataset resides on an unknown nonlinear manifold.

Identify the manifold and construct a new representation.

Project the data onto this representation to simplify further computations.

# Manifold Learning

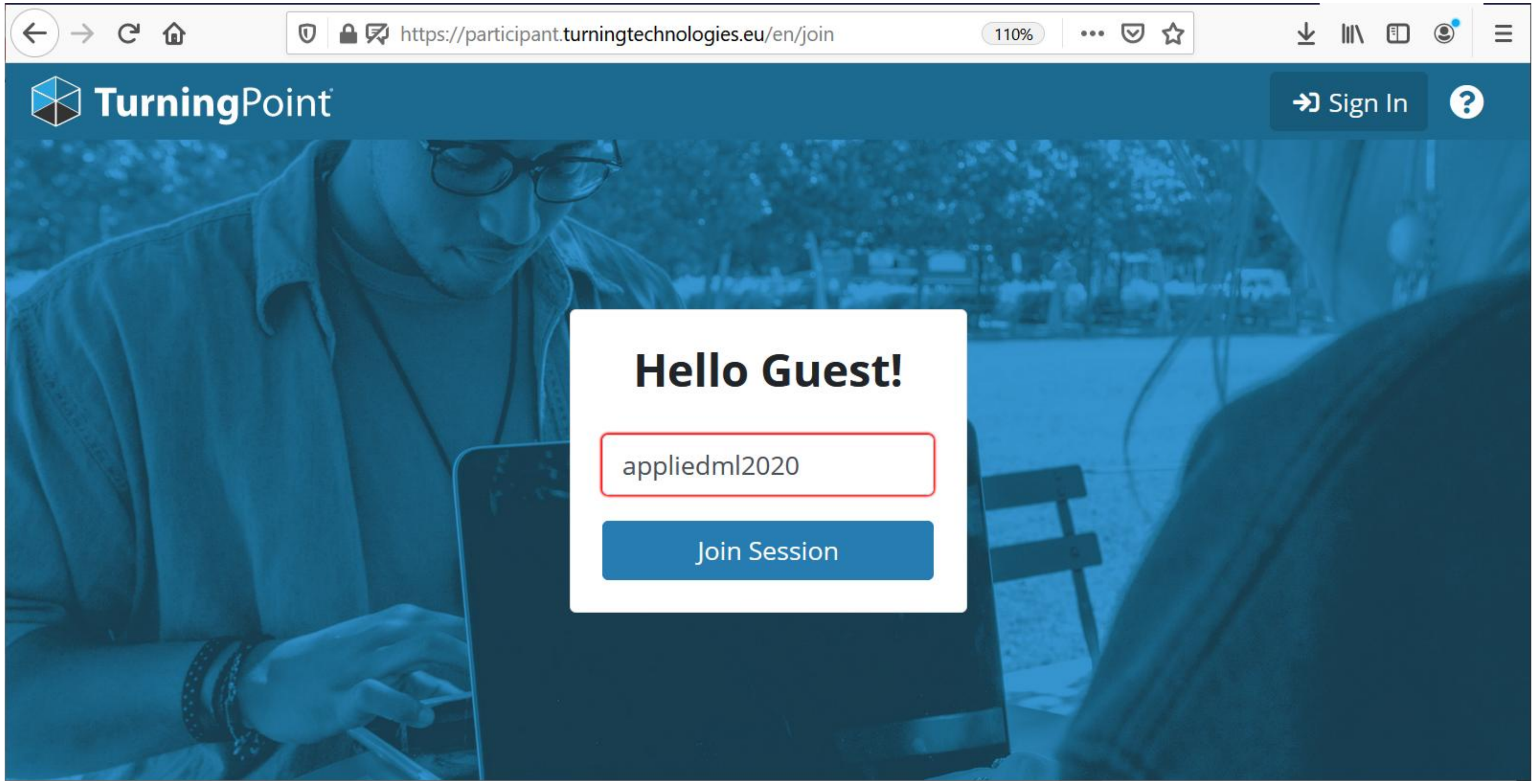

Coordinates in $\mathbb{R}^2$ for points in $U$
$\varphi(U)$
$\varphi$
$U$
$V$
Coordinate change
$\phi \circ \varphi^{-1}$
$\phi$
$\phi(V)$
Coordinates in $\mathbb{R}^2$ for points in $V'$

Meilă M, Zhang H. 2024
*Annu. Rev. Stat. Appl.* 11:393–417

# Launch polling system

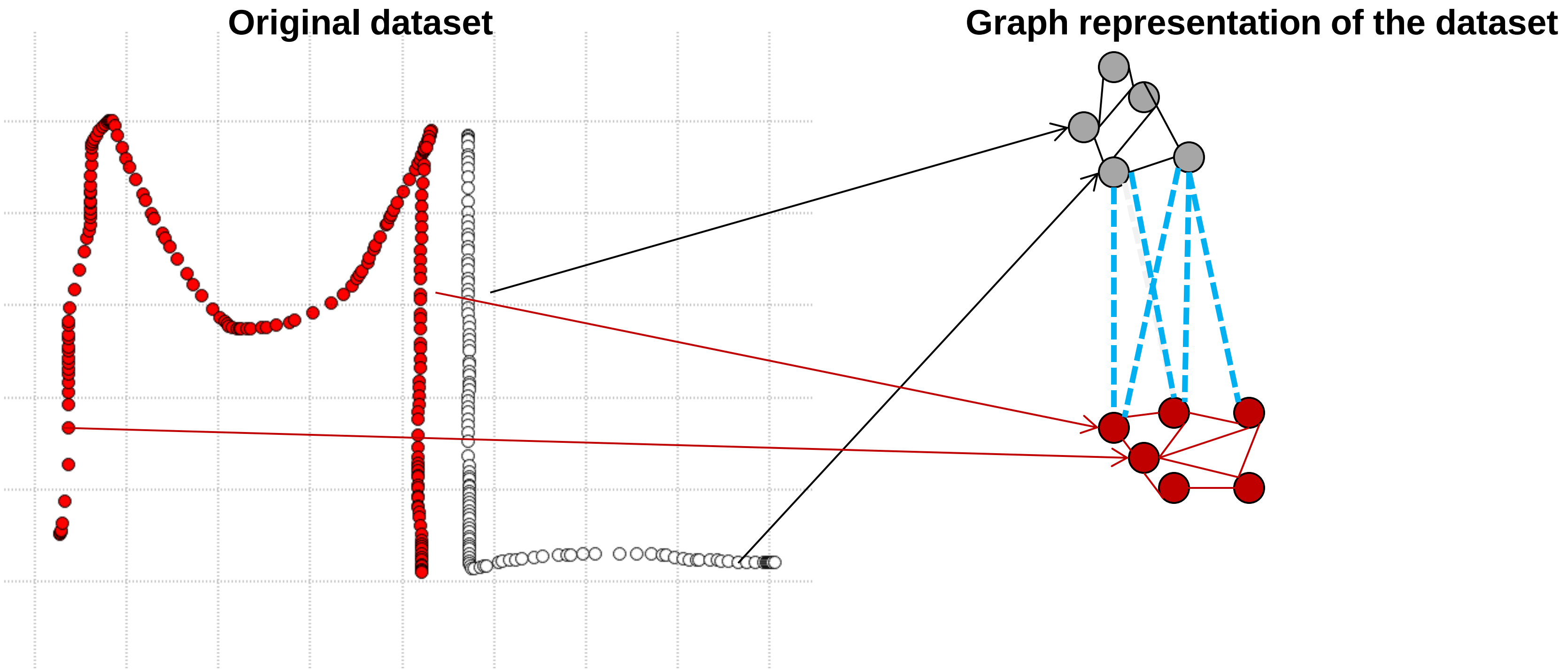
https://participant.turningtechnologies.eu/en/join

Acces as GUEST and enter the session id: *appliedml2020*



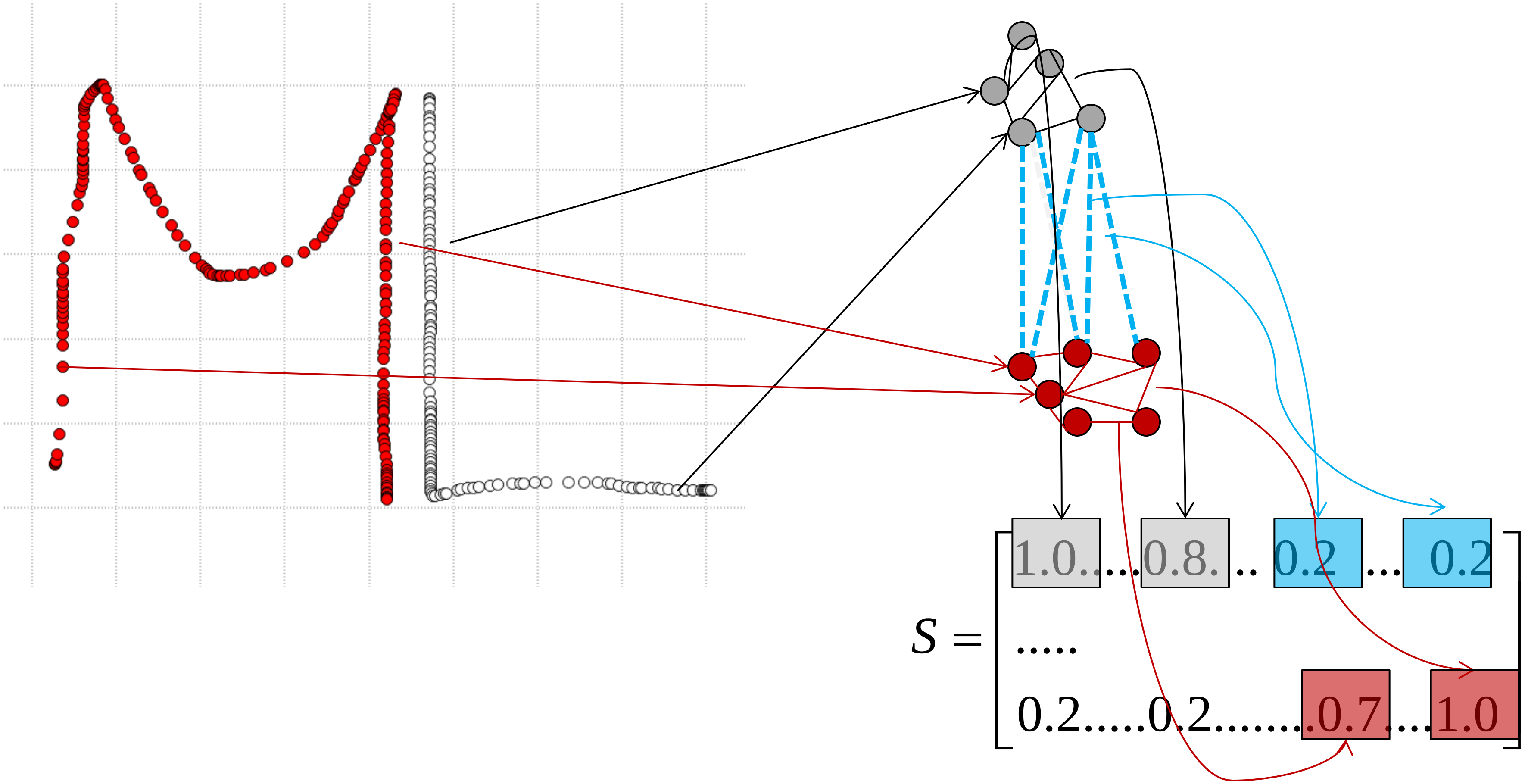
appliedml2020

# Embed Data in a Graph



- Build a *similarity graph*
- Each vertex on the graph is a datapoint

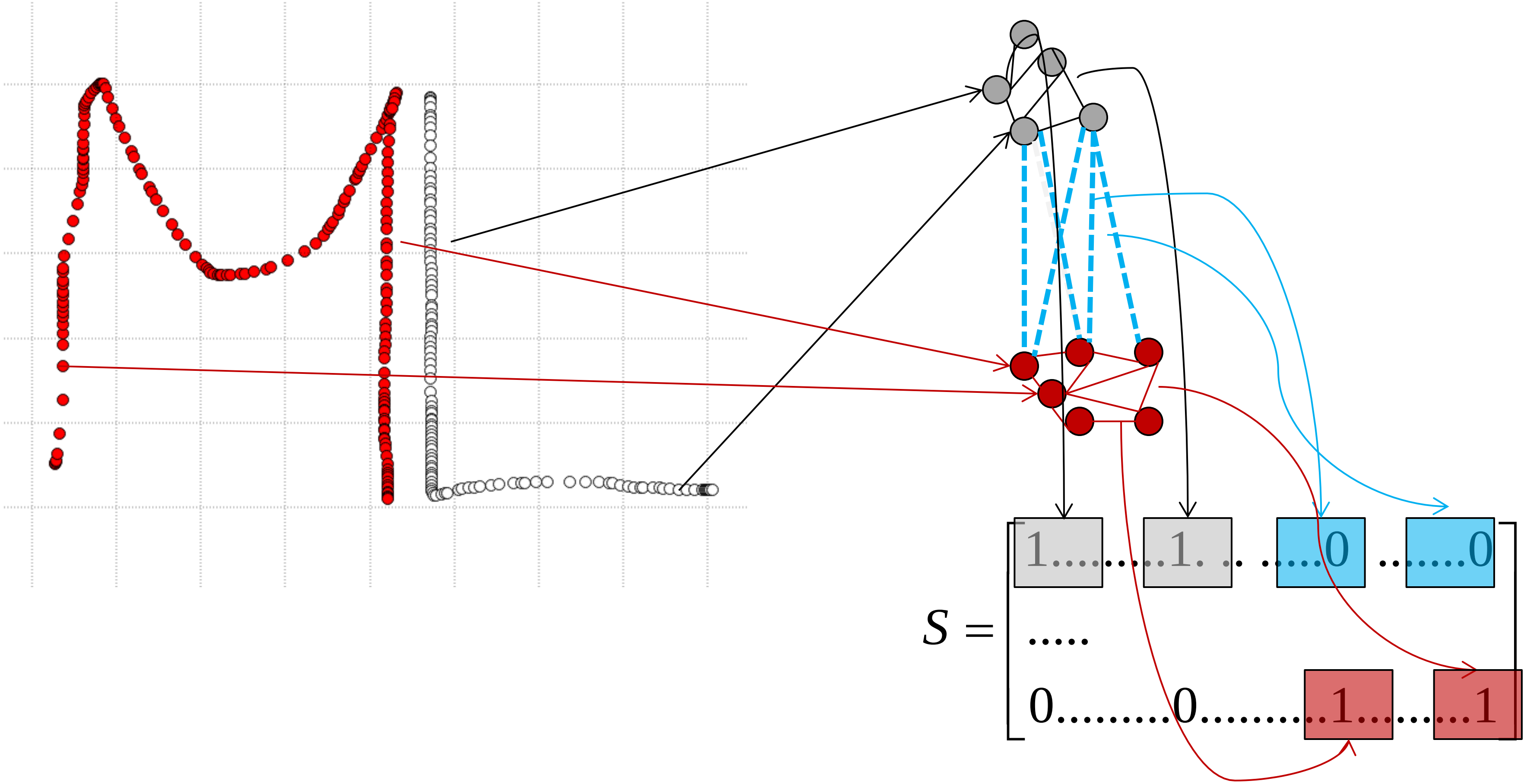
# Measure Distances in Graph



$$S = \begin{bmatrix} 1.0 \ldots\ldots 0.8 \ldots 0.2 \ldots 0.2 \\ \ldots\ldots \\ 0.2 \ldots\ldots 0.2 \ldots\ldots 0.7 \ldots 1.0 \end{bmatrix}$$

Construct the similarity matrix (adjacency matrix) *S* to denote whether points are close or far away to weight the edges of the graph.

# Connected Components in a Graph



$S =$

1………1. .. …..0 ..…..0

…..

0………0………1………1

If all blue connections have value zero in the similarity matrix, then the graph has 2 **connected components** (i.e. two disconnected blocks of datapoints; all datapoints within a block are connected).

# Spectral Clustering: example

$$S = \begin{bmatrix} 1.0 & 0.9 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 1.0 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 1.0 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 0.9 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 1.0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

Looking at the similarity matrix, what can you say about the distribution of points?

A. There are 4 datapoints.

B. There are 8 datapoints.

C. I do not know.

# Spectral Clustering: example

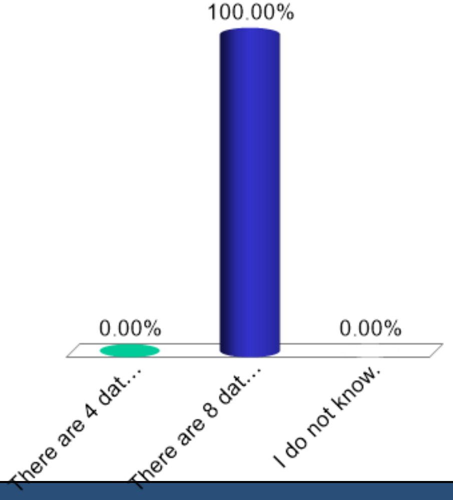
$$S = \begin{bmatrix} 1.0 & 0.9 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 1.0 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 1.0 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 0.9 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 1.0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

Looking at the similarity matrix, what can you say about the distribution of points?

A. There are 4 datapoints.

B. There are 8 datapoints. ✓

C. I do not know.



# Spectral Clustering: example

$$S = \begin{bmatrix} 1.0 & 0.9 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 1.0 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 1.0 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 0.9 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 1.0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

Looking at the similarity matrix, what can you say about the distribution of points?

A. The points are grouped in one cluster.

B. The points are grouped in 2 clusters.

C. The distribution of the points is uniform in each cluster.

D. I do not know.

# Spectral Clustering: example

$$S = \begin{bmatrix} 1.0 & 0.9 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 1.0 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 1.0 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 0.9 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 1.0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

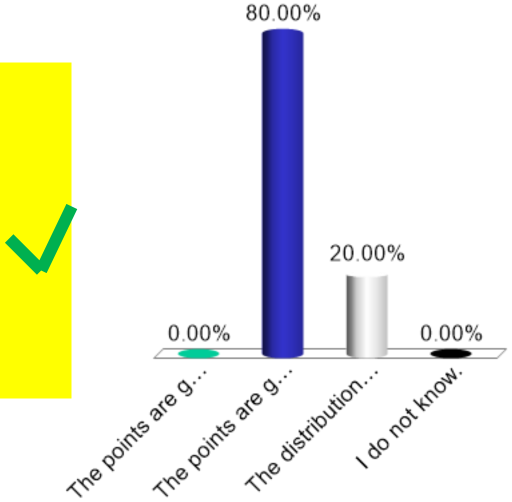
Looking at the similarity matrix, what can you say about the distribution of points?

A. The points are grouped in one cluster.

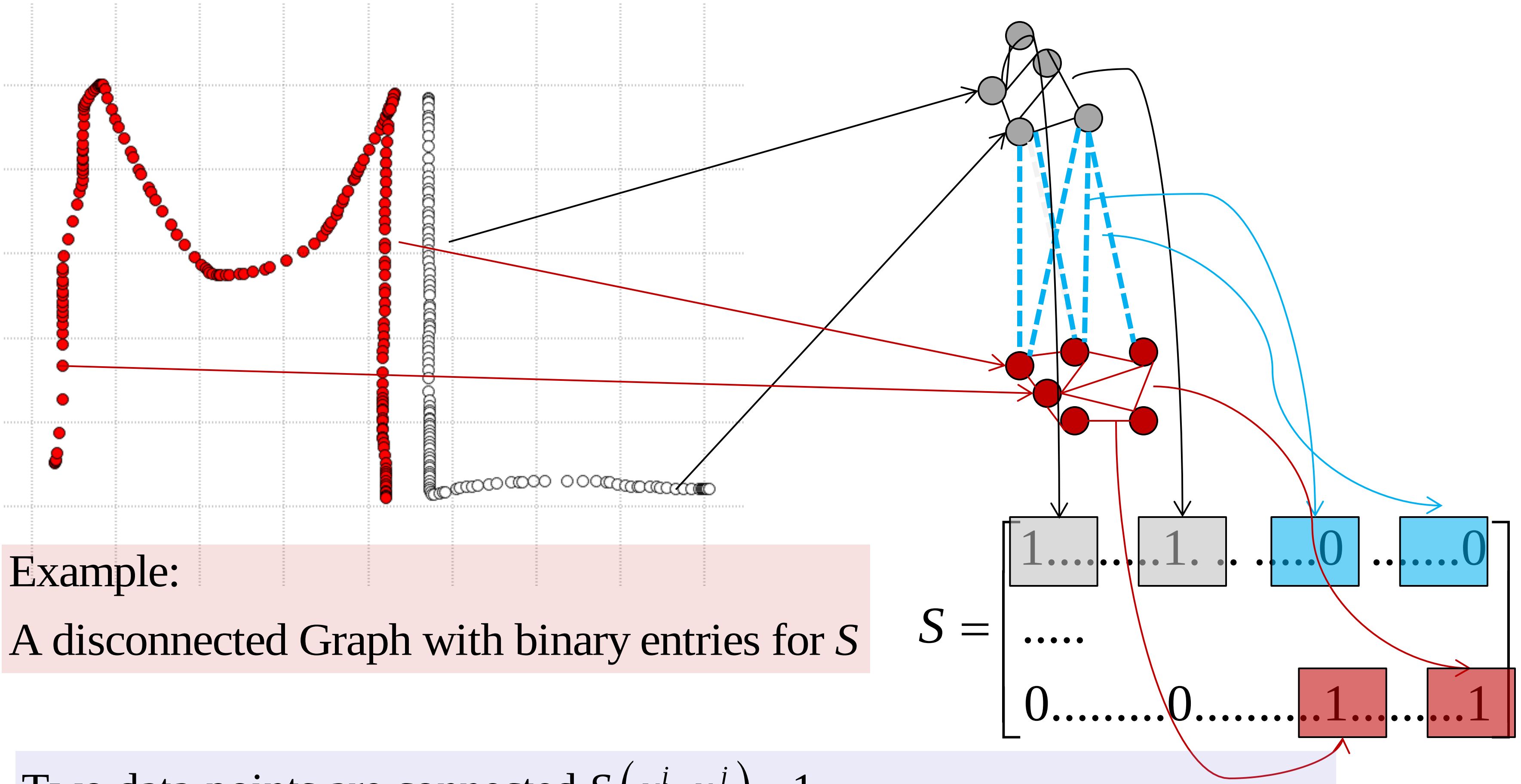
B. The points are grouped in 2 clusters. ✓

C. The distribution of the points is uniform in each cluster. ✓

D. I do not know.



# Disconnected Graphs



Example:

A disconnected Graph with binary entries for $S$

$$S = \begin{bmatrix} 1\ldots\ldots1\ldots\ldots0\ldots\ldots0 \\ \ldots.. \\ 0\ldots\ldots0\ldots\ldots1\ldots\ldots1 \end{bmatrix}$$

Two data-points are connected $S\left(x^i, x^j\right) = 1$

if a) the similarity between them is higher than a threshold;

or b) if they are k-nearest neighbors $\left(\text{according to the similarity metric}\right)$.

# Graph Laplacian

Eigenvalue decomposition of the Graph Laplacian matrix:

$$L = U \Lambda U^T$$

All eigenvalues of $L$ are positive and the smallest eigenvalue of $L$ is zero:

$\Rightarrow$ If we order the eigenvalues by increasing order:

$$\lambda_1 = 0 \leq \lambda_2 \leq .... \leq \lambda_M .$$

Theorem (see annexes):

If the graph has $k$ connected components, then the eigenvalue $\lambda$=0 has multiplicity $k$.

# Spectral Clustering

**Let us do exercise I**

# Spectral Clustering: Exercise I

Consider a two-dimensional dataset composed of two points.

a) Build a similarity matrix using a threshold function on  Euclidean (norm-2) distance. The metric outputs 1 if the points are close enough according to a threshold and zero otherwise. Consider two cases: when the two datapoints are close or far.

b) For each of the two cases above, build the Laplacian matrix, perform an eigenvalue decomposition and discuss the eigenvalues. $L = D - S$

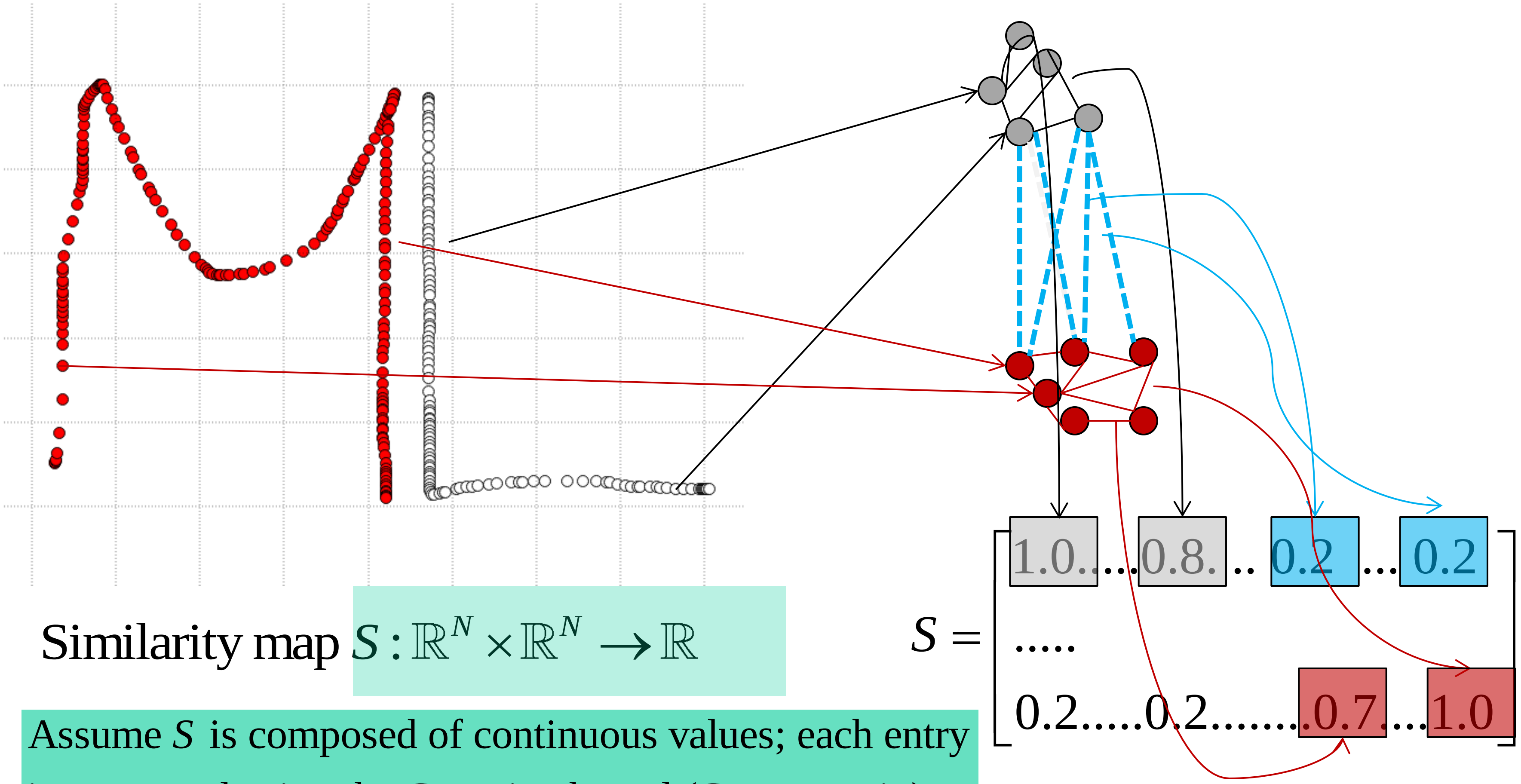
# Role of the eigenvalues in spectral clustering

$\Rightarrow$ The multiplicity of the eigenvalue 0 determines the number of connected components in a graph.

→ Identifying the number of clusters using the eigenvalue decomposition of the Laplacian matrix is then immediate (using above) *when* the similarity matrix is sparse.

→ What happens when the similarity matrix is full?

# Spectral Clustering



$$S = \begin{bmatrix} 1.0 \ldots 0.8 \ldots 0.2 \ldots 0.2 \\ \ldots.. \\ 0.2 \ldots.. 0.2 \ldots\ldots. 0.7 \ldots 1.0 \end{bmatrix}$$

Similarity map $S : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$

Assume $S$ is composed of continuous values; each entry is computed using the Gaussian kernel (Gram matrix)

$$S\left(x^i, x^j\right) = e^{-\frac{\left\|x^i - x^j\right\|^2}{2\sigma^2}}$$

# Spectral Clustering: exercise II

Consider a two-dimensional dataset composed of two points (assume again two cases – points are close to one another or are far apart).

a) Build a similarity matrix using a RBF kernel. Build the Laplacian matrix, perform an eigenvalue decomposition and discuss the eigenvalues and eigenvectors, for each of the two cases above.

b) Repeat (a) using a homogeneous polynomial kernel with p=2.

# Information entailed in the eigenvalues

$$S = \begin{bmatrix} 1.0 & 0.9 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 1.0 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 1.0 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 0.9 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 1.0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

What can you say about the eigenvalues of the associated Laplacian matrix?

A. Only one eigenvalue is zero.

B. Two eigenvalues are zero.

C. Four eigenvalues are zero.

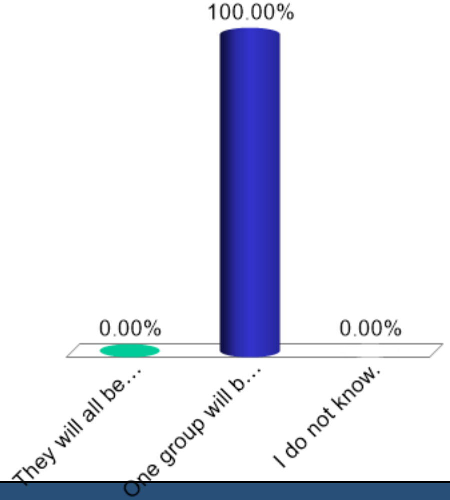
D. I do not know.

# Information entailed in the eigenvalues

$$S = \begin{bmatrix} 1.0 & 0.9 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 1.0 & 0.9 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 1.0 & 0.9 & 0 & 0 & 0 & 0 \\ 0.9 & 0.9 & 0.9 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 1.0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 1.0 & 0.1 \\ 0 & 0 & 0 & 0 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

What can you say of the **non zero** eigenvalues of the associated Laplacian matrix?

A. They will all be equal.
B. One group will be much larger than another group.
C. I do not know.



# Spectral Clustering: finding the clusters

This provides an indication of the number of clusters *K*
*(K: # of groups of eigenvalues with similar values).*

We do not yet know how the points are partitioned in the clusters!
Let us see now how we can infer the clusters from the eigenvalue decomposition.

# Spectral Clustering: finding the clusters

To find the clusters:

A. Project points on the K eigenvectors with largest eigenvalues.

B. Project points on the K eigenvectors with associated eigenvalue zero.

C. Project points on the K eigenvectors with smallest eigenvalues.

D. I do not know.

# Spectral Clustering: finding the clusters

To find the clusters:

A. Apply K-means on points in original space.

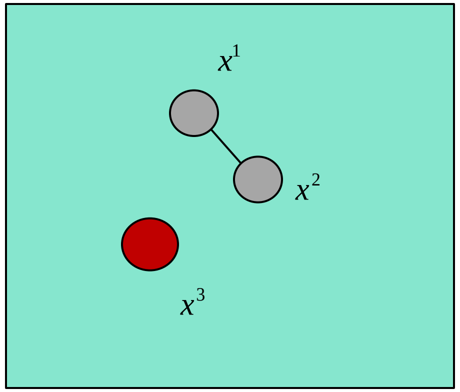
B. Apply K-means on points in projected space.

C. Apply K-means on points in both original and projected space.

D. I do not know.

# Finding the clusters: example

Example: 3 datapoints in a graph composed of 2 partitions



The similarity matrix is $S = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$L$ has eigenvalue $\lambda$=0 with multiplicity two.

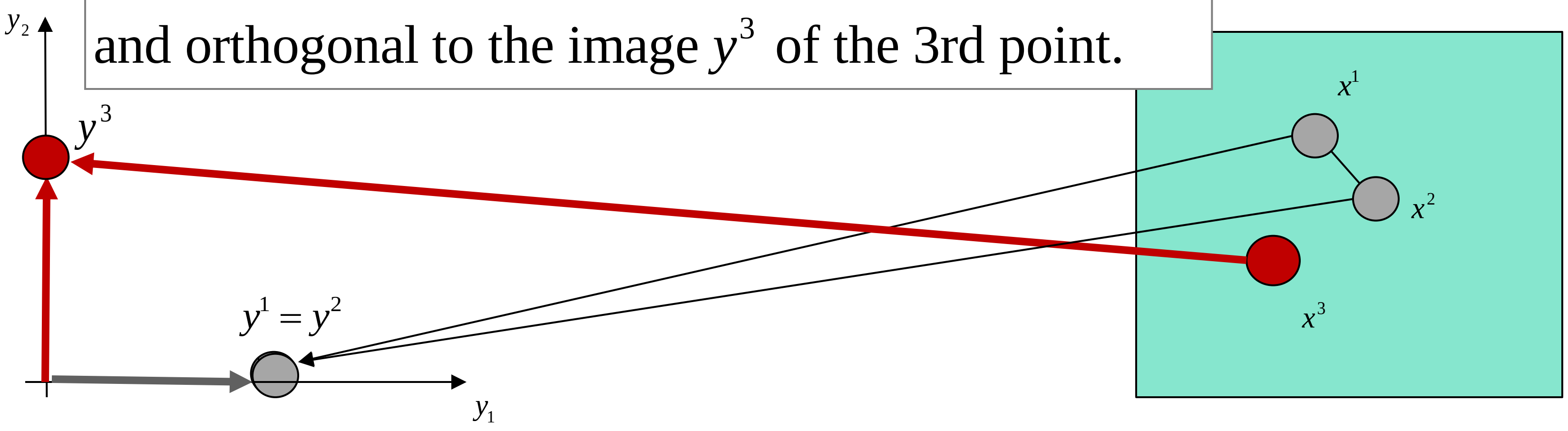
The eigenvectors of $L$ are:

$$e^1 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \quad e^2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad e^3 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

The images of the points are given by:

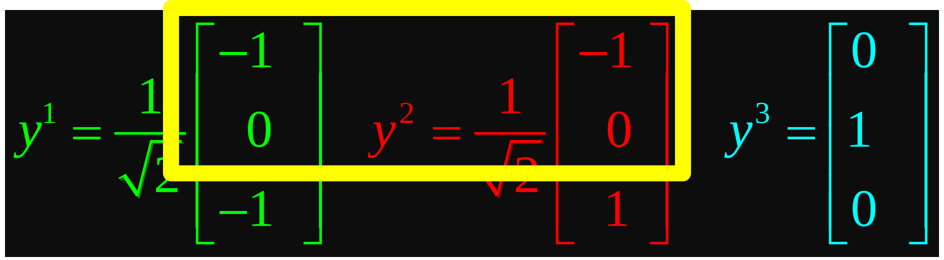
$$y^1 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ -1 \end{bmatrix} \quad y^2 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \quad y^3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

The images $y^1, y^2$ of the datapoints are superposed (when considering the first two dimensions only) and orthogonal to the image $y^3$ of the 3rd point.



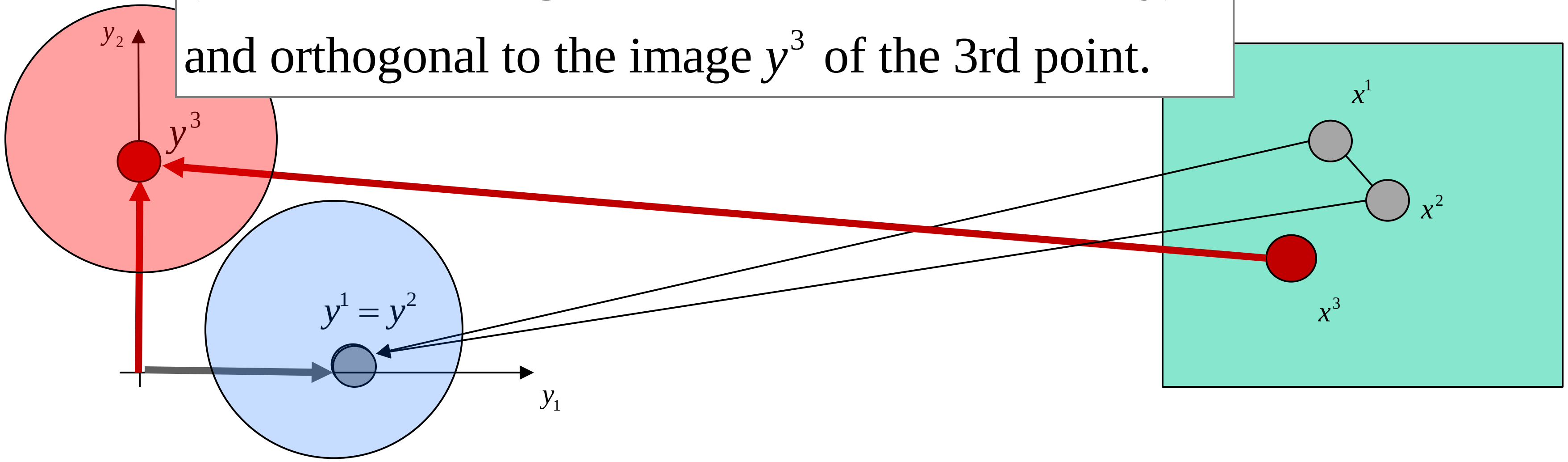
The coordinates of the images $y^1, y^2$ of the datapoints $x^1, x^2$ for the first two eigenvectors are equal.

The images of the points are given by:



$$y^1 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\0\\-1\end{bmatrix} \quad y^2 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\0\\1\end{bmatrix} \quad y^3 = \begin{bmatrix}0\\1\\0\end{bmatrix}$$

The images $y^1, y^2$ of the datapoints are superposed (when considering the first two dimensions only) and orthogonal to the image $y^3$ of the 3rd point.

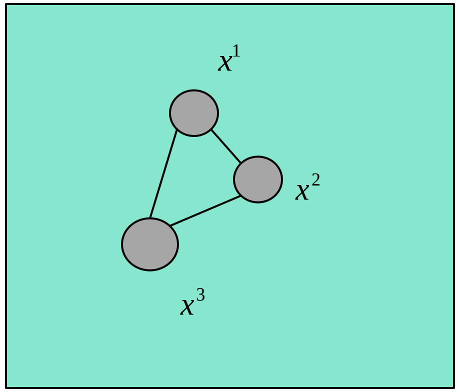


To discover the clusters, run K-means in image space (y-space).

# Finding the clusters: example

Example: 3 datapoints in a fully connected graph

$$S = \begin{bmatrix} 1 & 0.9 & 0.02 \\ 0.9 & 1 & 0.02 \\ 0.02 & 0.02 & 1 \end{bmatrix} \qquad L = \begin{bmatrix} 0.92 & -0.90 & -0.02 \\ -0.90 & 0.92 & -0.02 \\ -0.02 & -0.02 & 0.04 \end{bmatrix}$$



$L$ has eigenvalue $\lambda_1$=0 with multiplicity 1.

The second eigenvalue is small $\lambda_2 = 0.06$,

whereas the 3rd one is large, $\lambda_3 = 1.82$.

The coordinates of the images $y^1, y^2$ of the datapoints $x^1, x^2$ for the first two eigenvectors are again equal.

with associated eigenvectors :

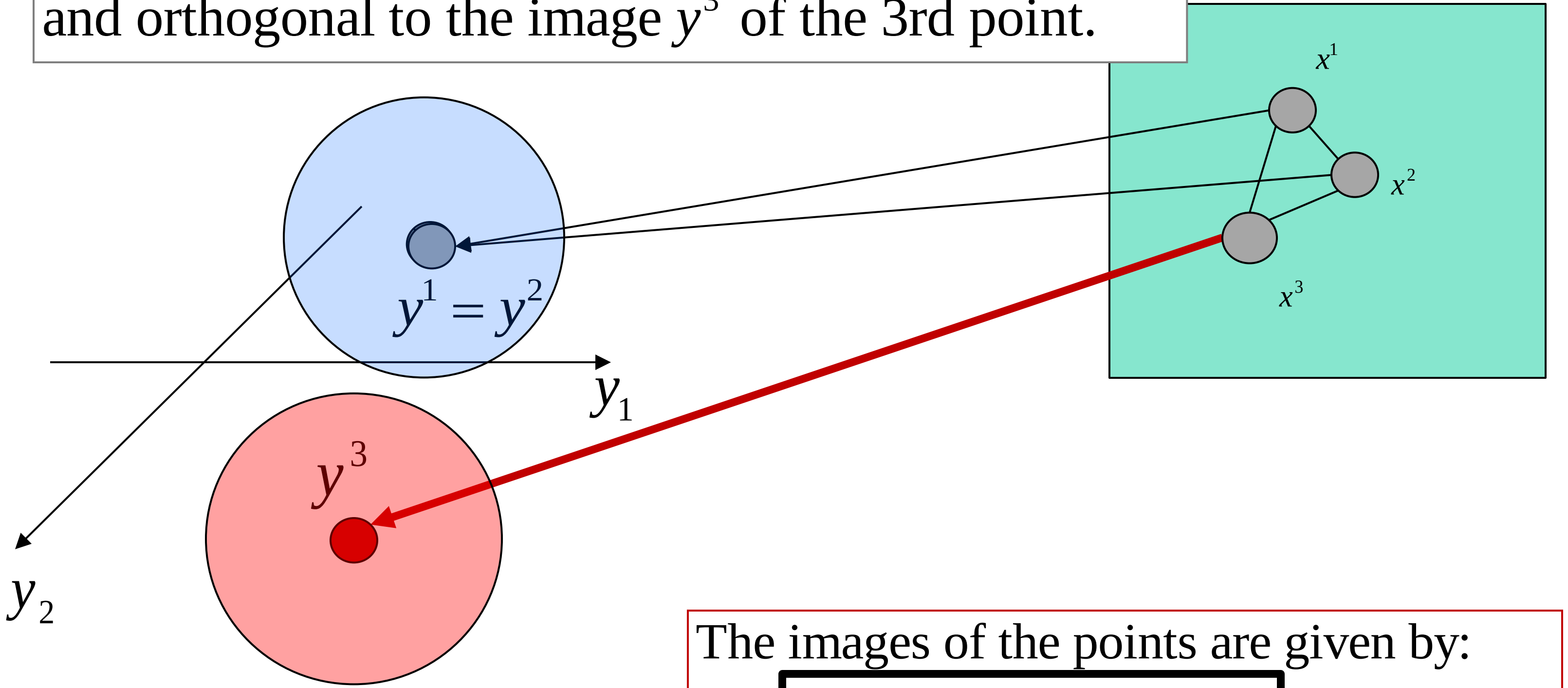
$$e^1 = \frac{1}{\sqrt{3}}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \; e^2 = \begin{bmatrix} -0.4 \\ -0.4 \\ 0.8 \end{bmatrix}, \; e^3 = \begin{bmatrix} -0.7 \\ 0.7 \\ 0.0 \end{bmatrix}$$

The images of the points are given by:

$$\boxed{y^1 = \begin{bmatrix} 1/\sqrt{3} \\ -0.4 \\ -0.7 \end{bmatrix}, y^2 = \begin{bmatrix} 1/\sqrt{3} \\ -0.4 \\ 0.7 \end{bmatrix}}, y^3 = \begin{bmatrix} 1/\sqrt{3} \\ 0.8 \\ 0.0 \end{bmatrix}$$

It makes sense to group eigenvectors with smallest eigenvalues.

The images $y^1, y^2$ of the datapoints are superposed (when considering the first two dimensions only) and orthogonal to the image $y^3$ of the 3rd point.



The images of the points are given by:

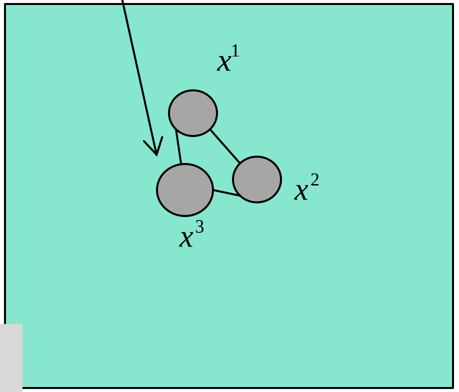
$$y^1 = \begin{bmatrix} 1/\sqrt{3} \\ -0.4 \\ -0.7 \end{bmatrix}, y^2 = \begin{bmatrix} 1/\sqrt{3} \\ -0.4 \\ 0.7 \end{bmatrix}, y^3 = \begin{bmatrix} 1/\sqrt{3} \\ 0.8 \\ 0.0 \end{bmatrix}$$

To discover the clusters, run K-means in image space (y-space).

The 3rd point is now closer to the two other points

Example: 3 datapoints in a fully connected graph

The similarity matrix is $S = \begin{bmatrix} 1 & 0.9 & 0.8 \\ 0.9 & 1 & 0.7 \\ 0.8 & 0.7 & 1 \end{bmatrix}$



$L$ has eigenvalue $\lambda$=0 with multiplicity 1. The second and third eigenvalues are both large $\lambda_2 = 2.23,\ \lambda_3 = 2.57$.

with associated eigenvectors :

$$e^1 = \frac{1}{\sqrt{3}}\begin{bmatrix}1\\1\\1\end{bmatrix},\ e^2 = \begin{bmatrix}-0.21\\-0.57\\0.79\end{bmatrix},\ e^3 = \begin{bmatrix}-0.78\\0.57\\0.21\end{bmatrix}$$

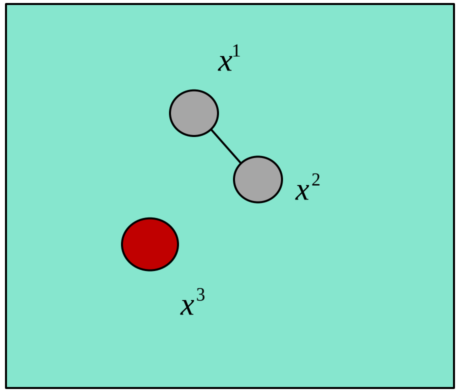
Entries are no longer equal!

The images of the points are given by:

$$y^1 = \begin{bmatrix}1/\sqrt{3}\\-0.21\\-0.78\end{bmatrix},\ y^2 = \begin{bmatrix}1/\sqrt{3}\\-0.57\\0.57\end{bmatrix},\ y^3 = \begin{bmatrix}1/\sqrt{3}\\0.79\\0.21\end{bmatrix}$$

# Equivalency to other non-linear Embeddings

What about kernel PCA?



The eigenvalue decomposition of $S$ (equiv. to kPCA on Gram matrix) yields the set of dual eigenvectors:

$$\alpha^1 = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\\0\end{bmatrix},\ \alpha^2 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ \alpha^3 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\-1\\0\end{bmatrix}$$

$\lambda_1 = 2,\ \lambda_2 = 1,\ \lambda_3 = 0.$

With spectral clustering, the eigenvectors of $L$ are:

$$e^1 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\-1\\0\end{bmatrix},\ e^2 = \begin{bmatrix}0\\0\\1\end{bmatrix},\ e^3 = \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\\0\end{bmatrix}$$

$\lambda_1 = 0,\ \lambda_2 = 0,\ \lambda_3 = 2.$

→ The dual eigenvectors with non-zero eigenvalues are aligned with the set of eigenvectors of the Laplacian matrix!

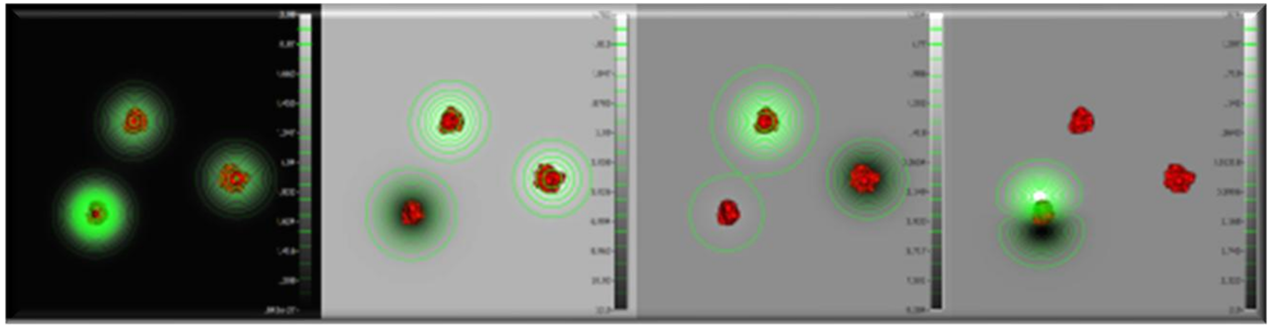
Careful: this is not true in arbitrary cases!

# Kernel PCA as preprocessing before K-means?

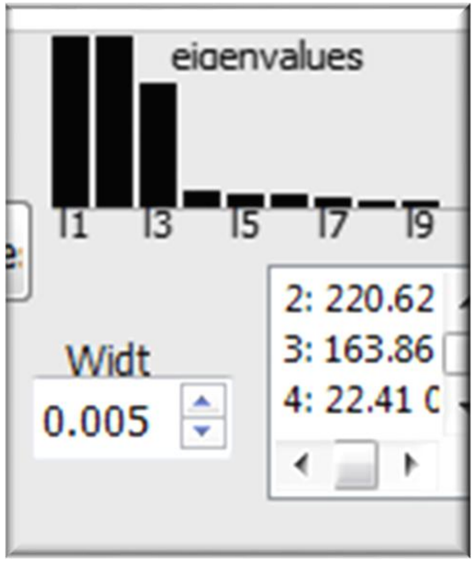
# KPCA as pre-processing before kernel K-means

The choice of parameters  in kernel K-Means can be initialized by doing a readout of the Gram matrix after kernel PCA.

The number of large eigenvalues = number of clusters (here 3)



Projections on dual eigenvectors 1 to 4 (from left to right)

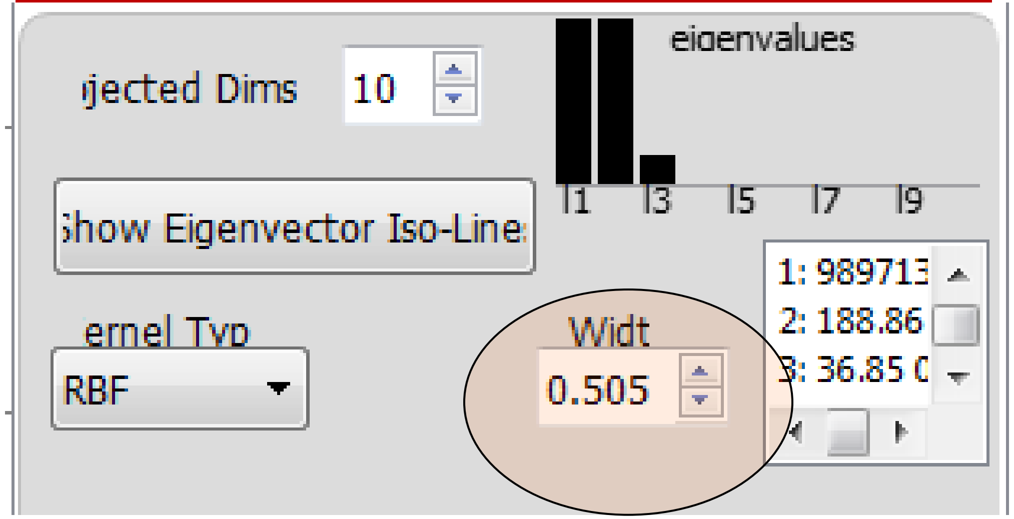


2: 220.62
3: 163.86
4: 22.41 0

# KPCA as pre-processing before kernel K-means

**The choice of kernel and kernel's hyperparameters determines the number of clusters**

The number of large eigenvalues = number of clusters (here 2)

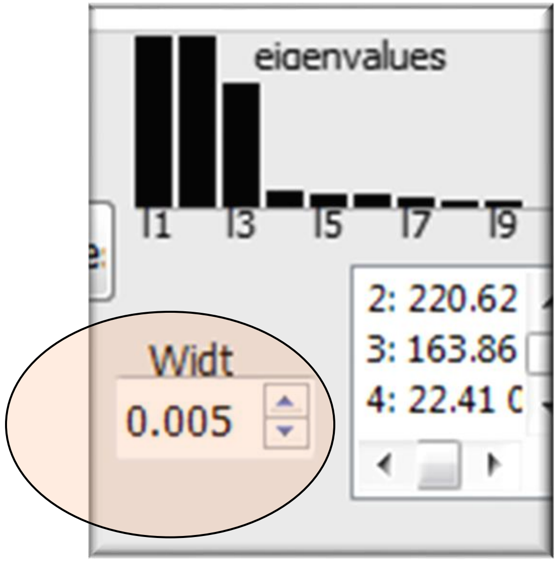


jected Dims 10

ernel Typ RBF

Widt 0.505

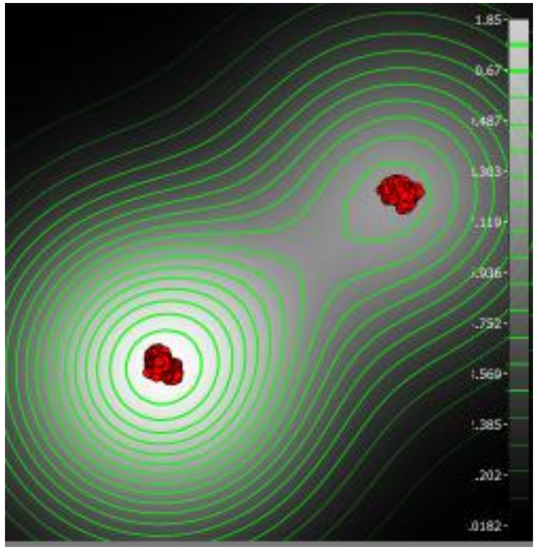
1: 989713
2: 188.86
3: 36.85 0

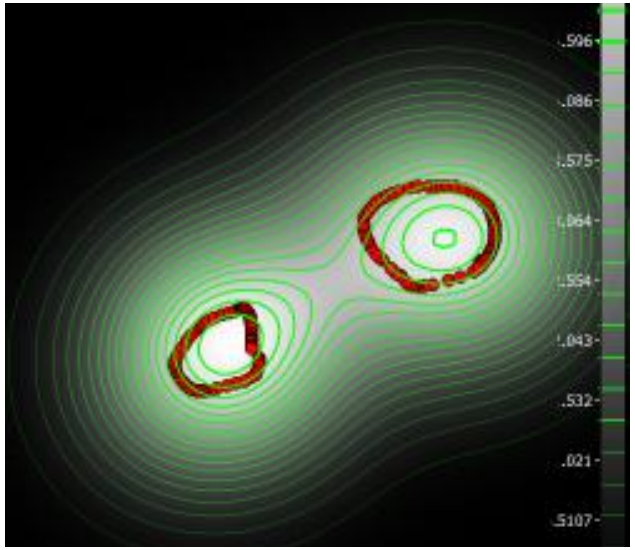


Widt 0.005

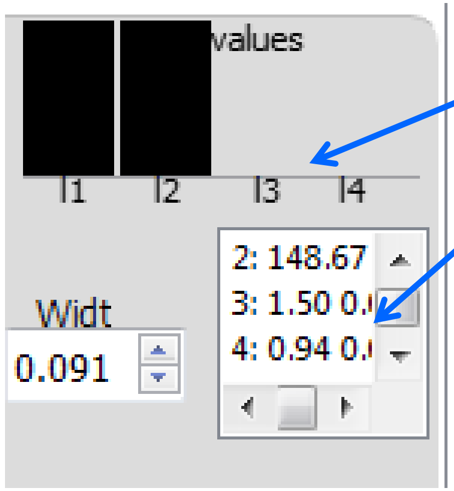
2: 220.62
3: 163.86
4: 22.41 0

# Looking at eigenvalue distribution in Kernel PCA projections can help determine the tightness of the clusters.



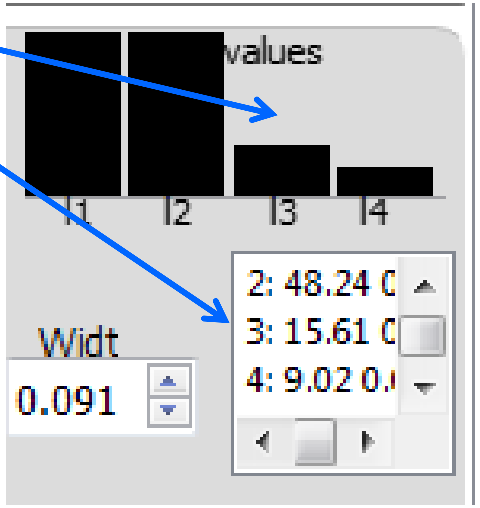


**The smallest eigenvalues entail information about dispersion within clusters**
**The smaller, the tighter the cluster.**



2: 148.67
3: 1.50 0.(
4: 0.94 0.(

Widt
0.091



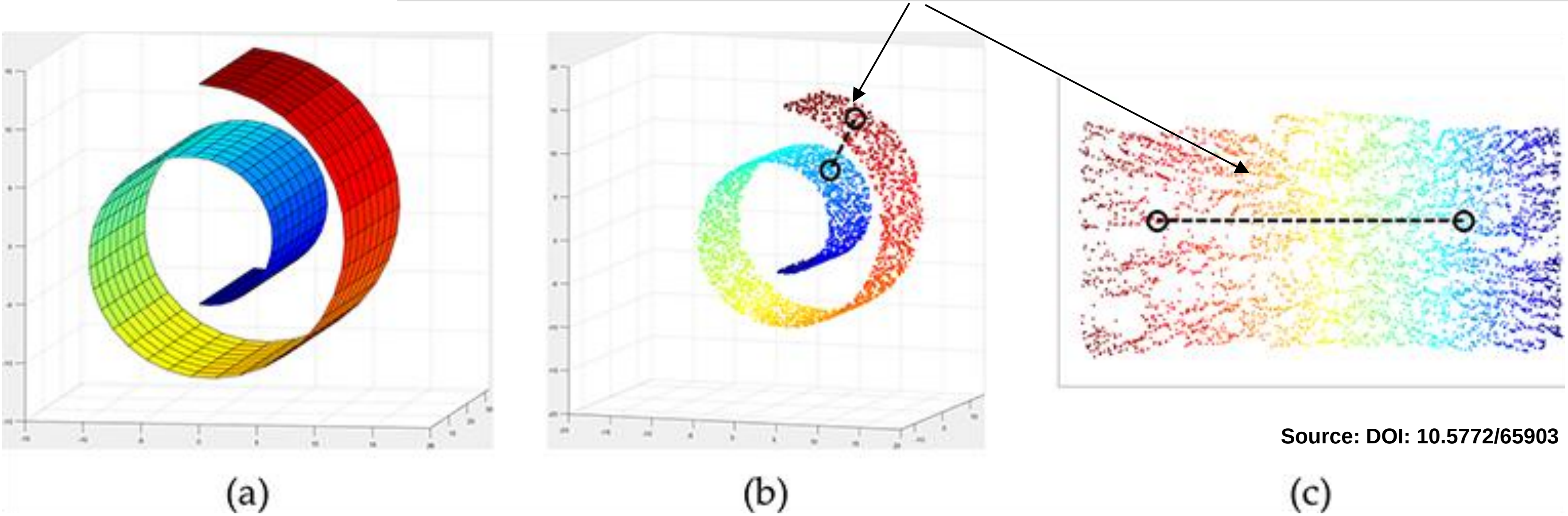
2: 48.24 (
3: 15.61 (
4: 9.02 0.(

Widt
0.091

# Non-linear embeddings

# Non-linear embeddings

Points close to one another are actually far apart on the manifold

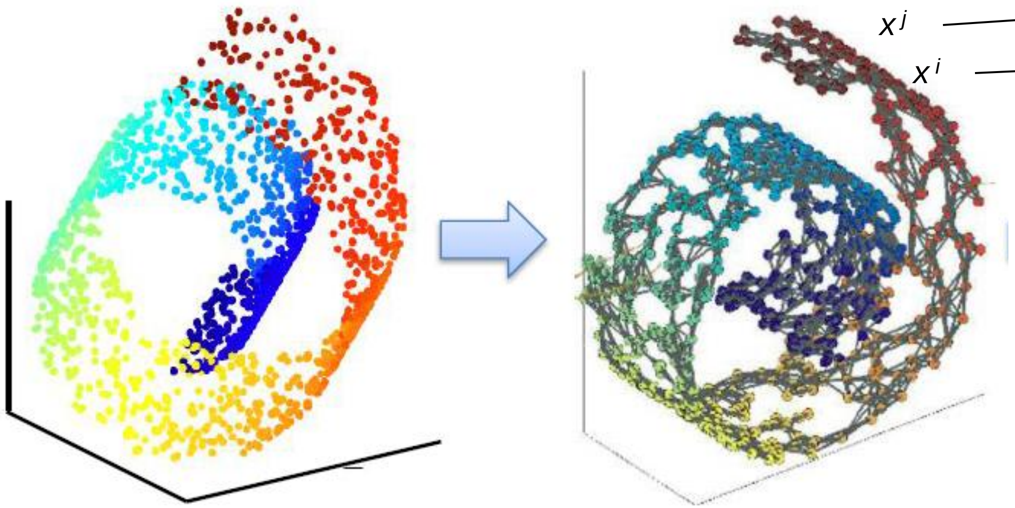


Source: DOI: 10.5772/65903

If one could flatten the manifold, this would be advantageous for:

- Visualizing the data
- Reducing the dimensionality
- linearly separating the group of data

# Laplacian Eigenmaps



Sample datapoints

Construct graph from datapoints

Similarity Matrix:

$$S = \begin{bmatrix} 1 & 0.5 & 0 & 0 \,..... \\ 0.5 & 1 & 0 & \\ 0 & 0 & 1 \,........... & \\ ....... & & & \end{bmatrix}$$

**Adjacency graph:**

We put an edge (rbf kernel) between nodes *i* and *j* if:

- $x^i$ is among *k* nearest neighbors of $x^j$ or
- if $x^j$ is among *k* nearest neighbors of $x^i$

(the relation is symmetric)

# Laplacian Eigenmaps

Solve the generalized eigenvalue problem:

$$Le^i = \lambda De^i \Leftrightarrow \left(I - \boxed{D^{-1}} S\right)e^i = \lambda e^i$$

$e^i$ : $M$ eigenvectors

If $D$ not invertible, solve:

$$\min_y e^T Le \quad \text{such that } \; e^T De = 1.$$

Similarity Matrix:

$$S = \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0.3 & 1 \end{bmatrix}$$

Symetrized and normalized form of the Graph Laplacian

Ensures minimal distortion while offering a rescaling.

Similarity Matrix:

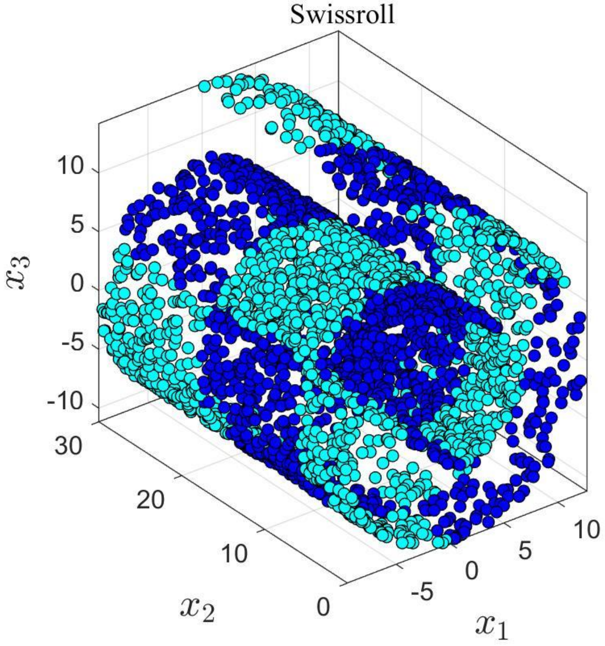
$$I - D^{-1}S = \begin{bmatrix} 0.33 & -0.33 & 0 & 0 \\ -0.33 & 0.33 & 0 & 0 \\ 0 & 0 & 0.23 & -0.23 \\ 0 & 0 & 0.23 & 0.23 \end{bmatrix}$$

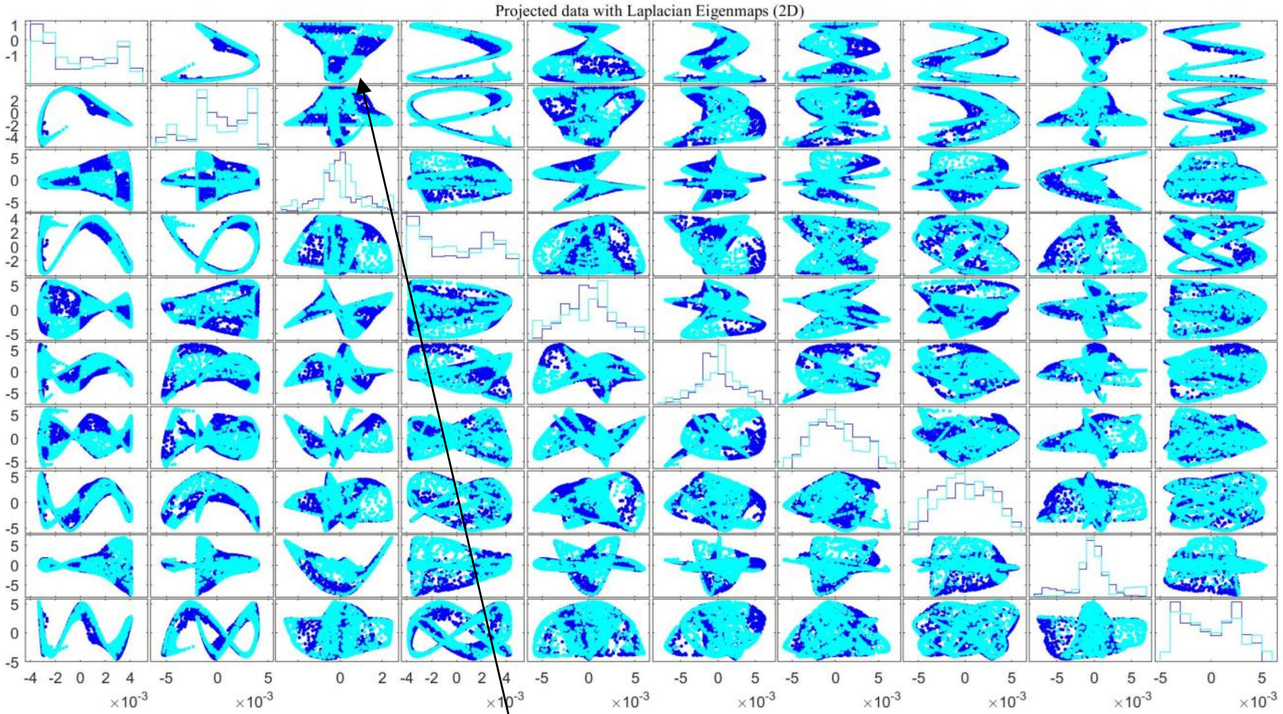
Reflected in the eigenvalues

| | | | | |
|---|---|---|---|---|
| Eigenvalues of $L$ : | 0 | 0 | 0.46 | 0.66 |
| Eigenvalues of $I - D^{-1}S$ : | 0 | 0 | 0.6 | 1.0 |

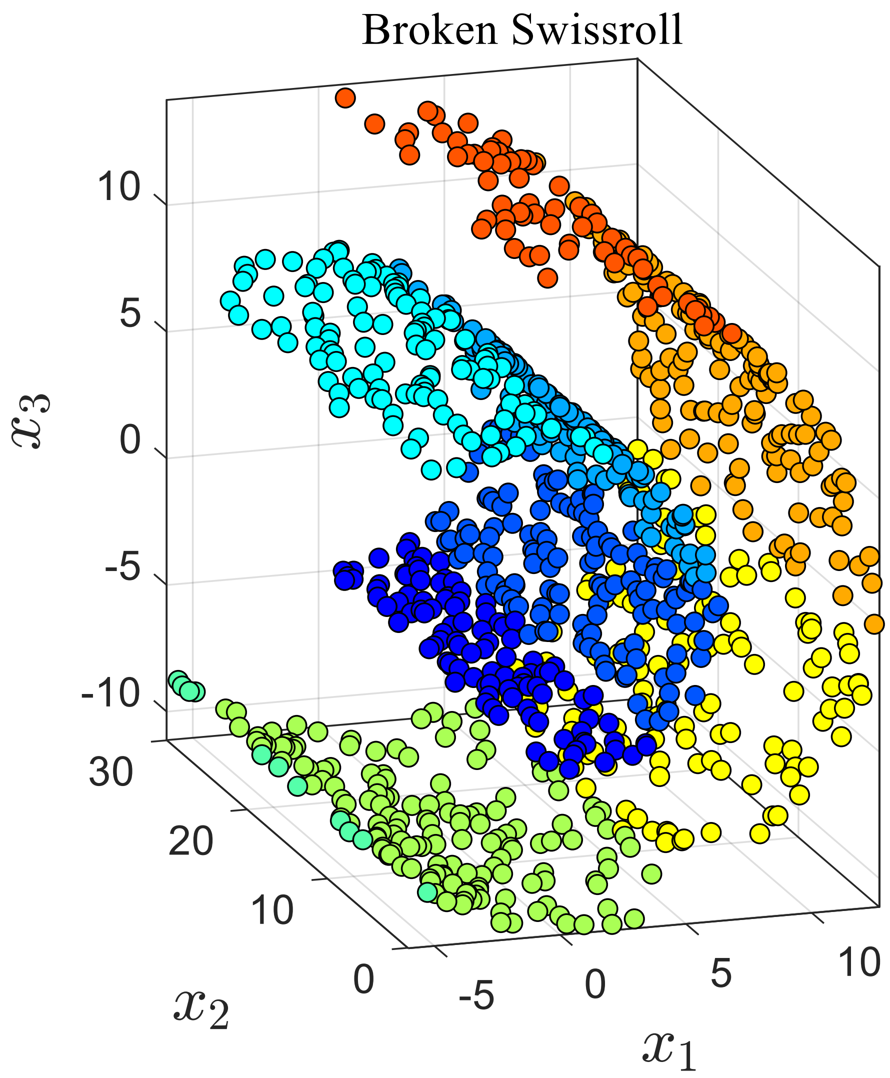
# Laplacian Eigenmaps

Data used in the practice session





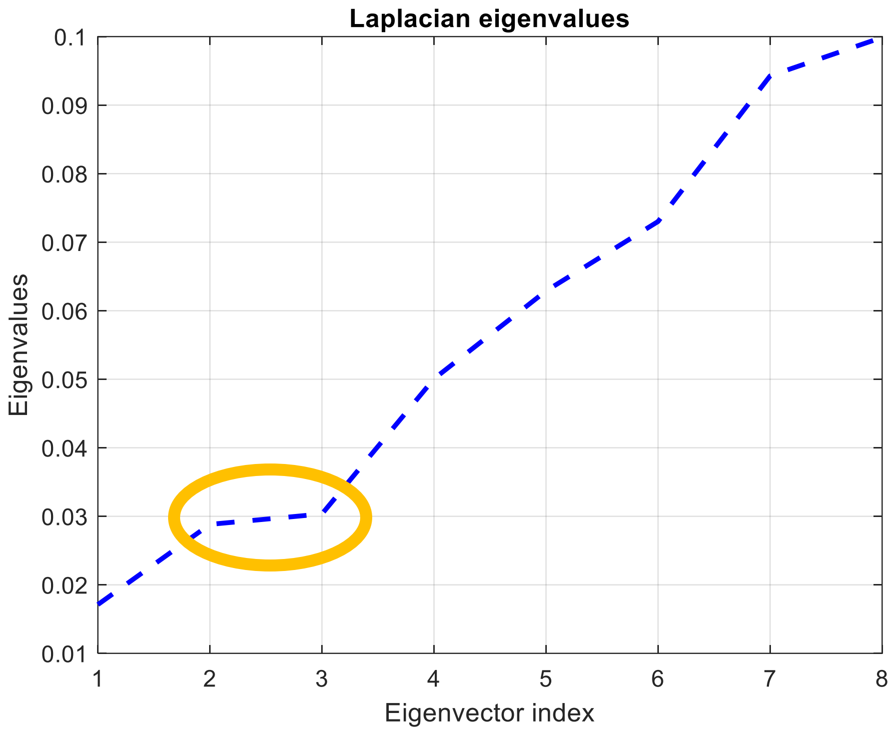
The projections on the pair $e^2, e^3$ generate a flat embedding that enables a piece-wise linear partitioning.



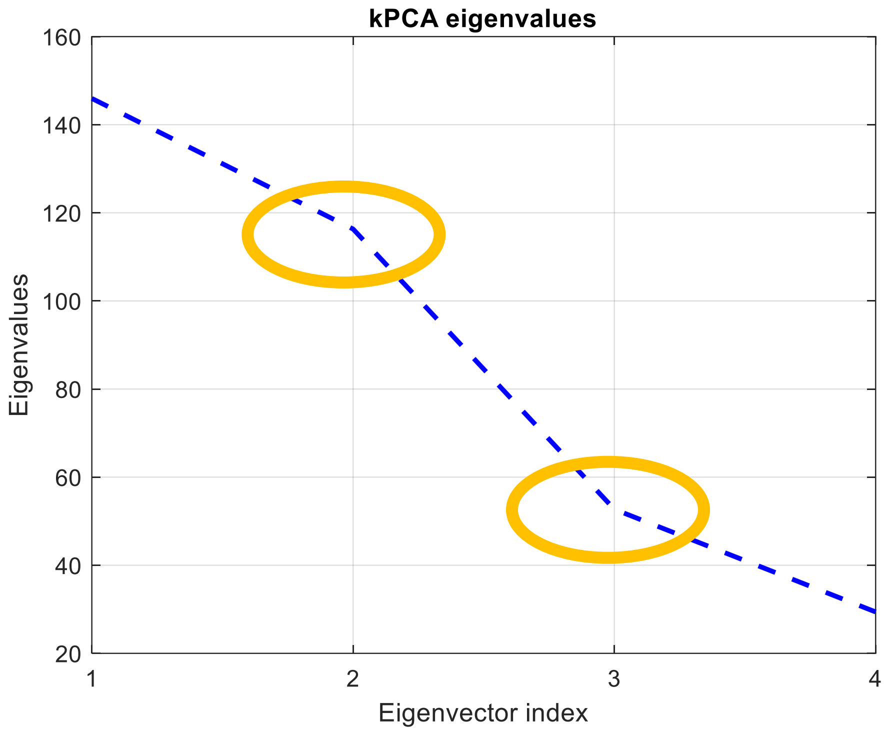
**How to determine the best embedding?**

# Comparison of eigenvalues

Decomposition of the Laplacian

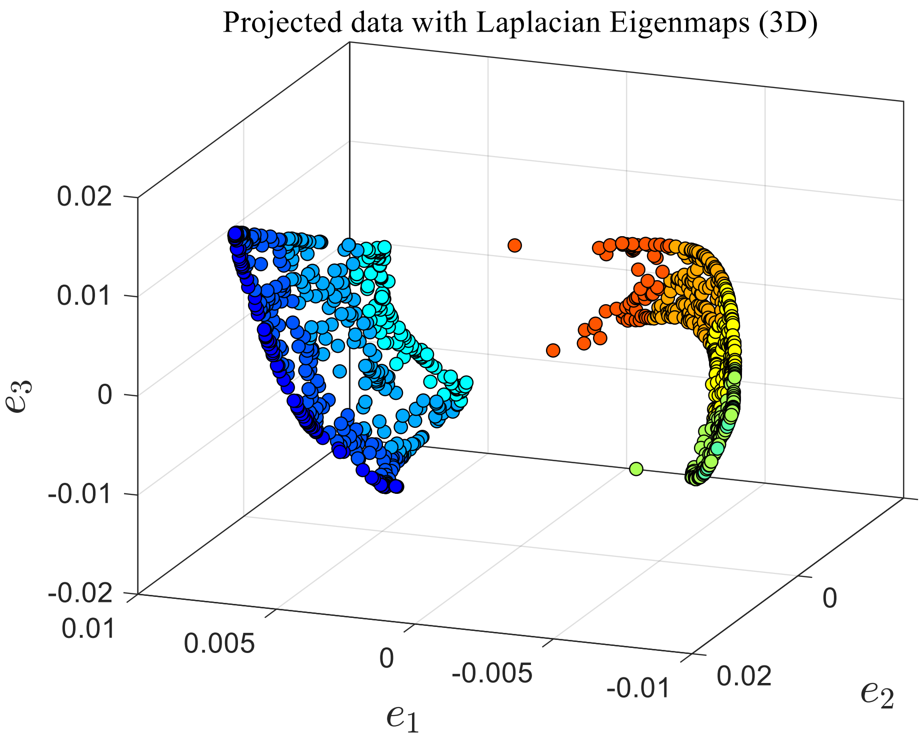


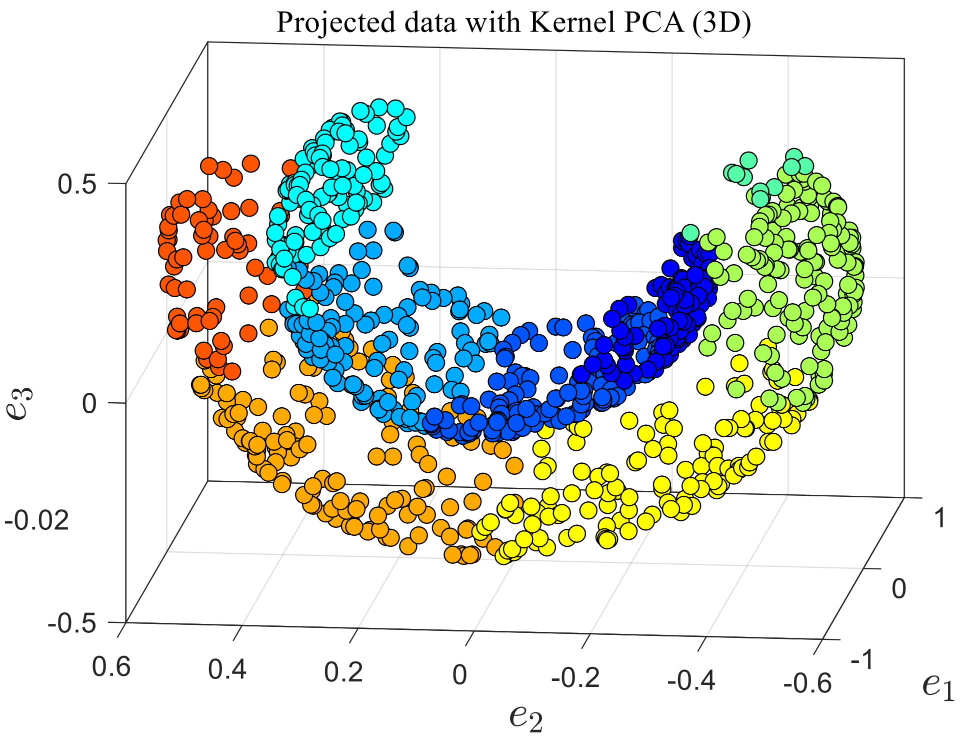
Decomposition of the Gram Matrix



Looking at the Laplacian eigenvalues eases the identification of the number of clusters.

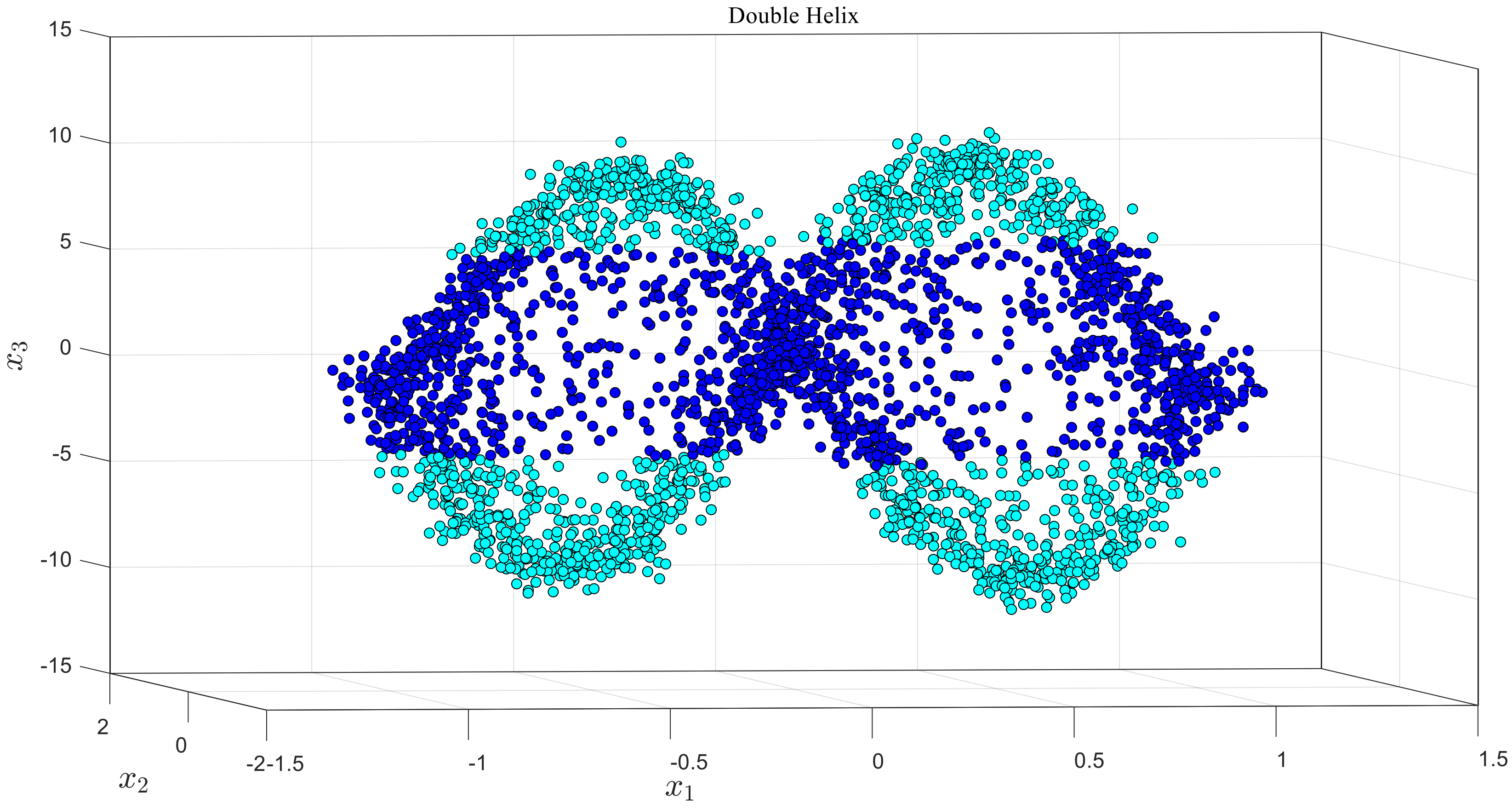
# Comparison of projections


Projected data with Laplacian Eigenmaps (3D)


Projected data with Kernel PCA (3D)

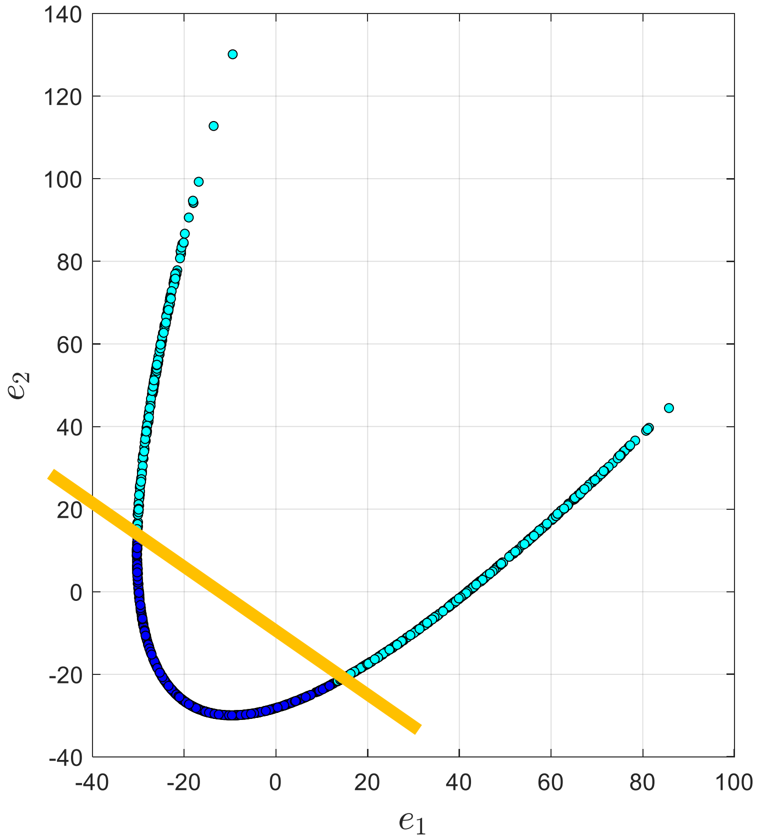
Clusters are better separated in the Laplacian projections.

# Tightly embedded clusters



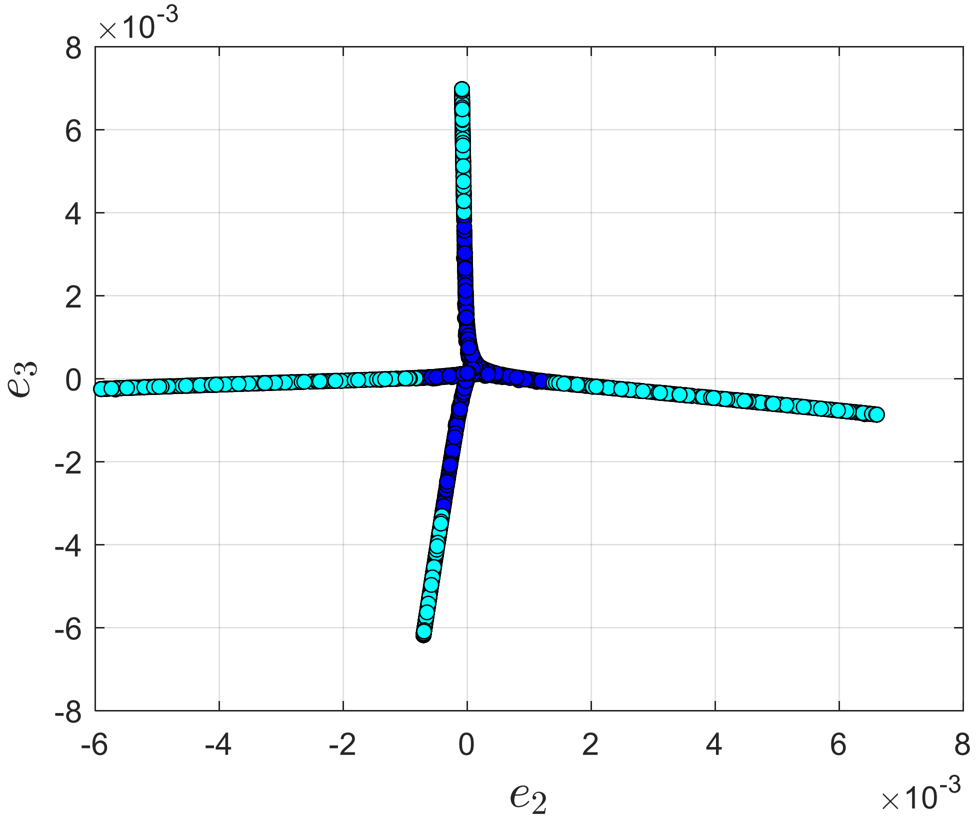
**When the groups of datapoints are intermingled along different dimensions.**

# Double Helix kPCA Projection



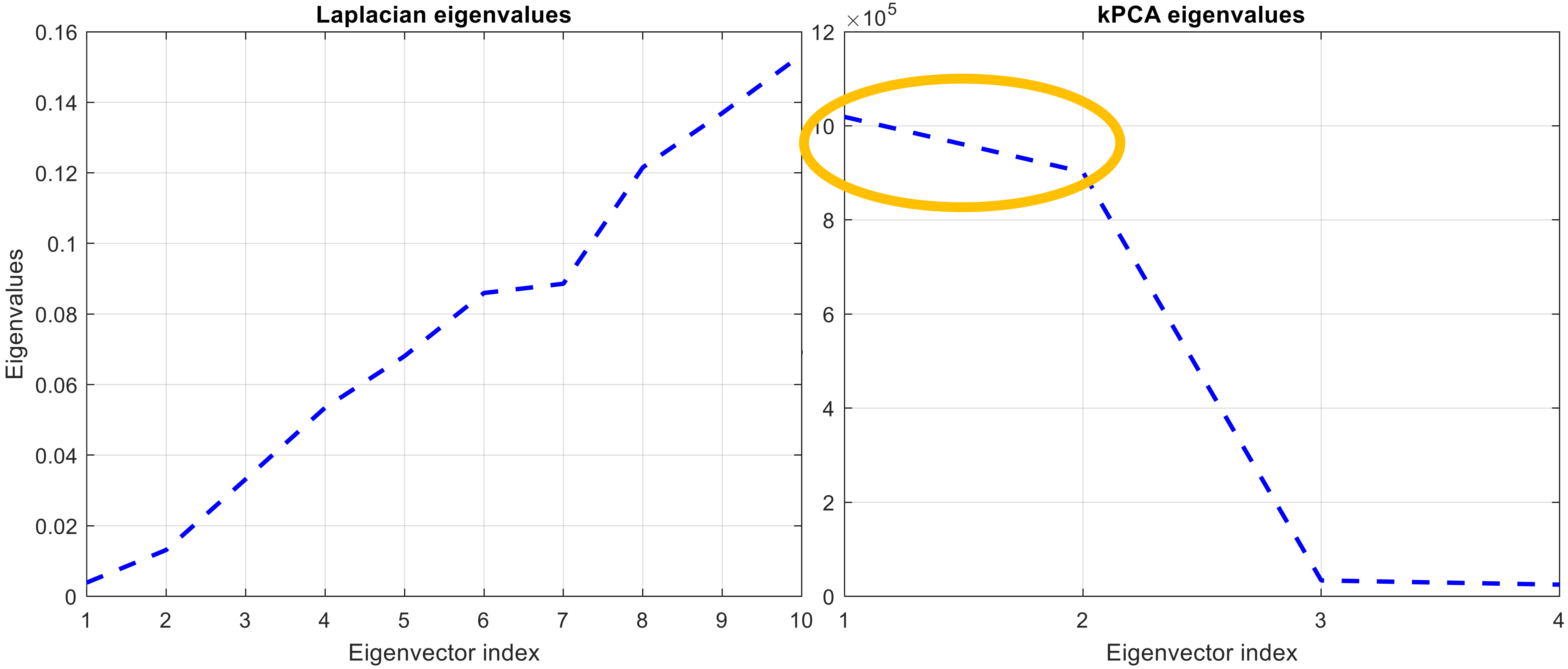
Projection after kernel PCA with inhomogeneous polynomial kernel

# Double Helix Laplacian Projections



Projection using Laplacian eigenmaps

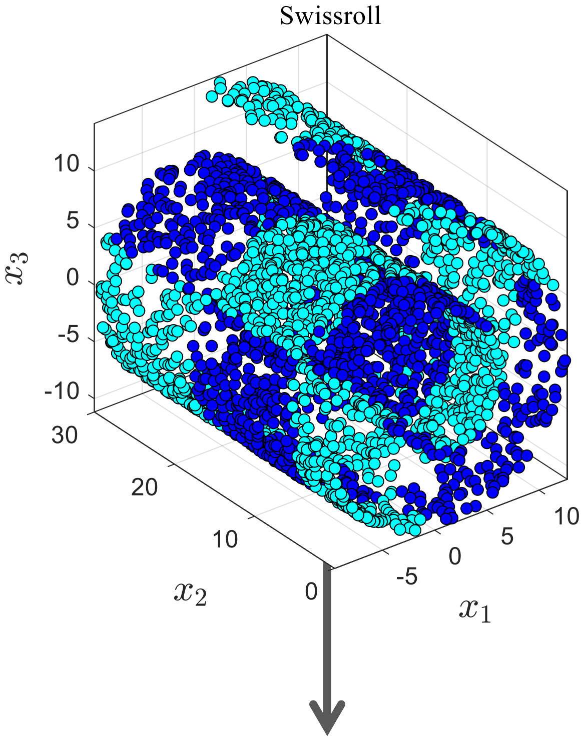
# Comparison of eigenvalues

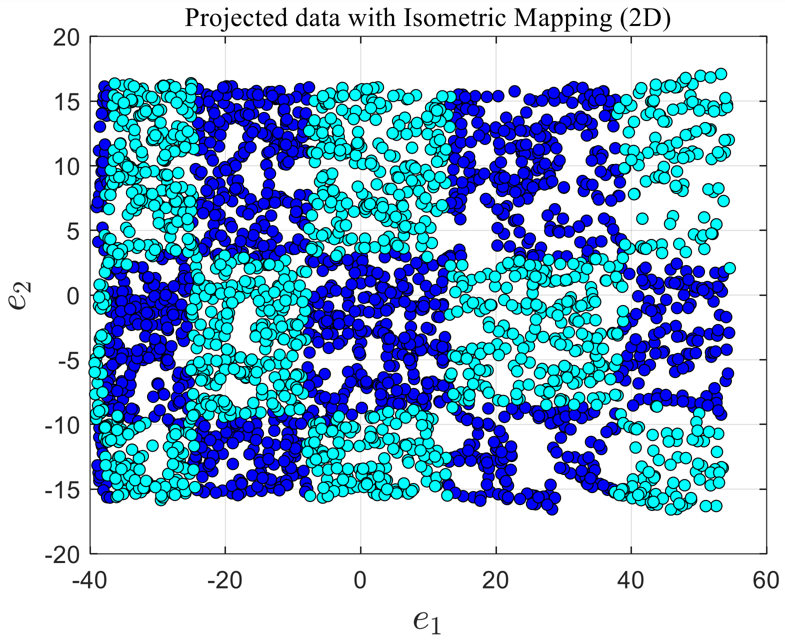


kPCA does a better job at extracting the number of clusters and projections

Laplacian projections are best suited when the clusters are well separated spatially (disconnected graph)

# Isomap embedding





Compute pairwise distances:

$$S_{ij} = \left( \min_{k-nearest\ \text{neighbours}} d\left(x^i, x^j\right) \right)^2$$

Isomap is an extension of Multidimensional scaling.

Do an eigendecomposition of the centered similarity matrix.

# Multi-Dimensional Scaling (MDS)

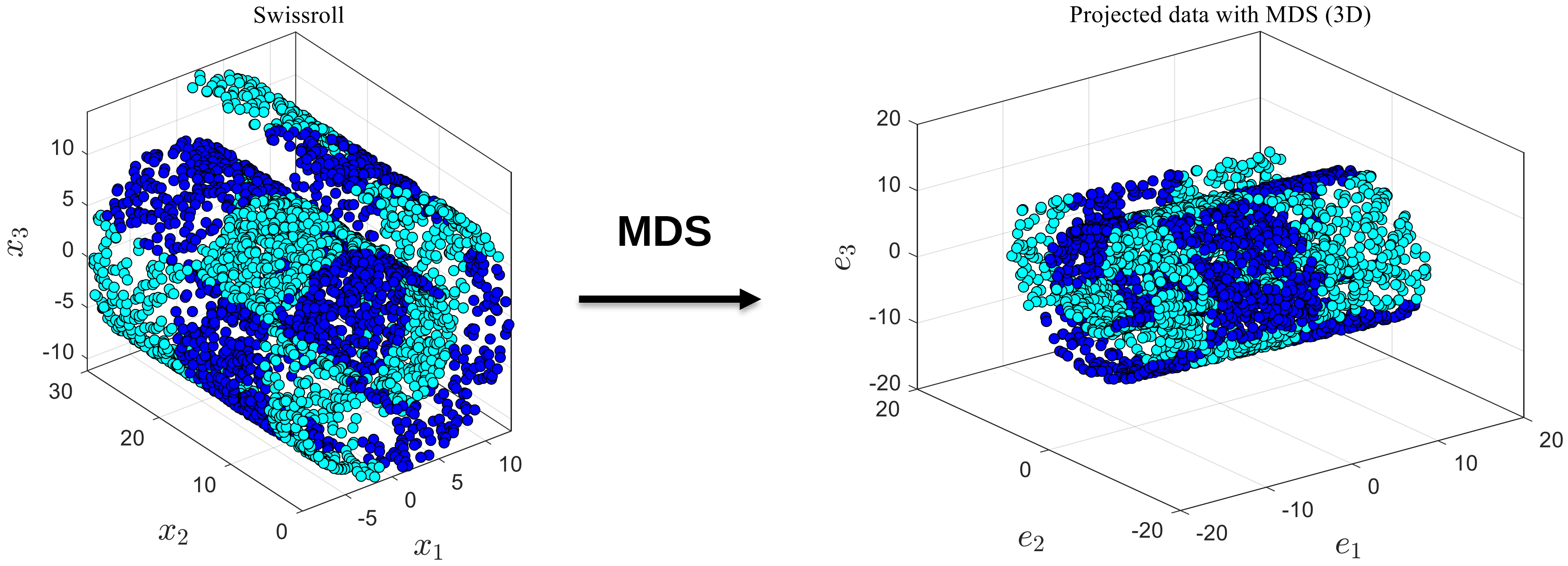
1) Build similarity matrix from Gram Matrix, using linear kernel

$S_{ij} = \left\langle x^i, x^j \right\rangle$ ~ squared pairwise distance $\left\| x^i - x^j \right\|$

2) Center the similarity matrix: $S'_{ij} = S_{ij} - \frac{1}{M}\sum_{k=1}^{M} S_{ik} - \frac{1}{M}\sum_{k=1}^{M} S_{kj} + \frac{1}{M^2}\sum_{k,l=1}^{M} S_{kl}$

3) Decompose $S'$ to obtain eigenvectors $e^i$, $i = 1 \ldots M$.

4) Generate scaled projections $y_i^j = \sqrt{\lambda_i}\, e_j^i$

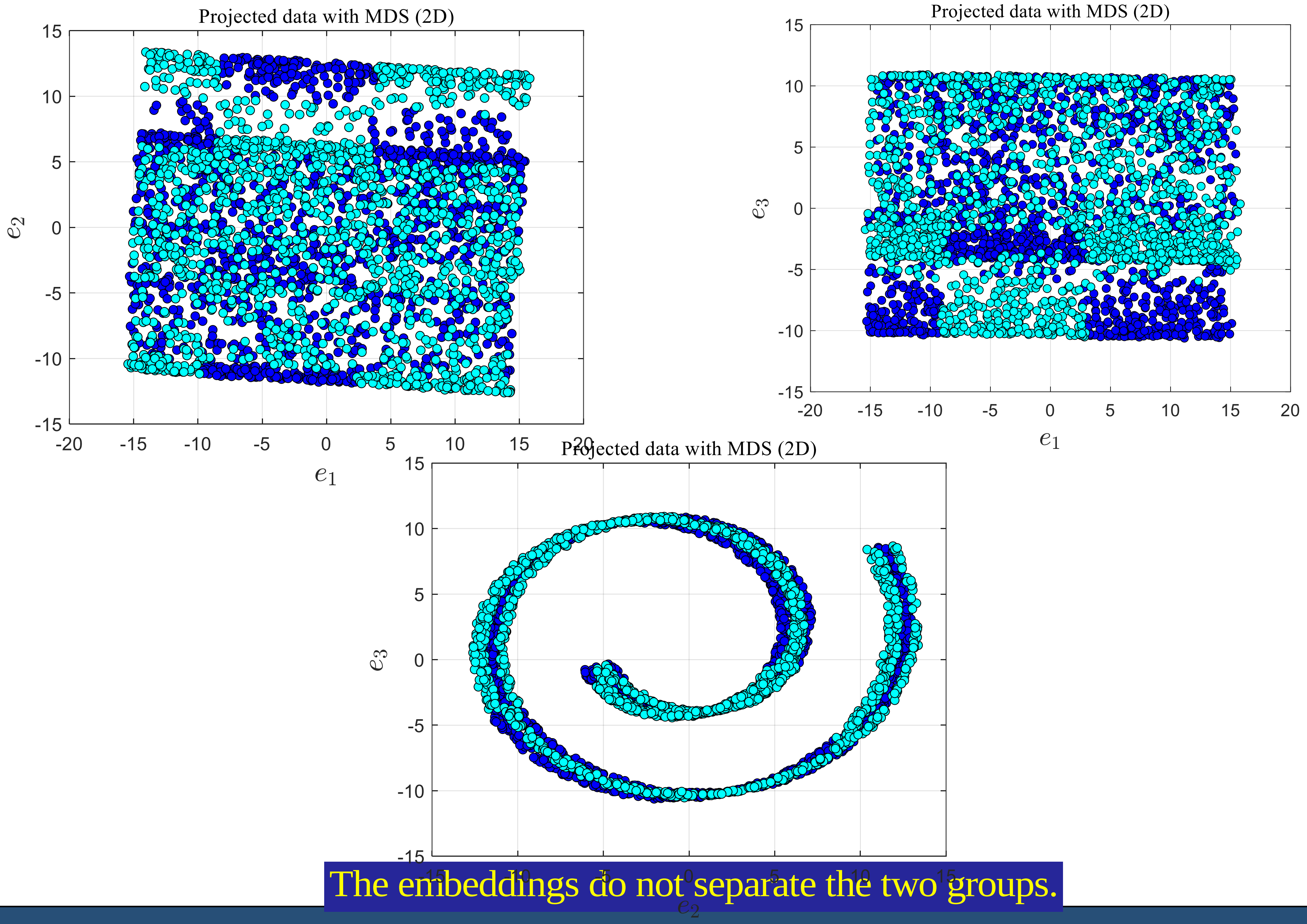


Swissroll

Projected data with MDS (3D)

Flattens and normalizes but does not separate.

# Multi-Dimensional Scaling (MDS)



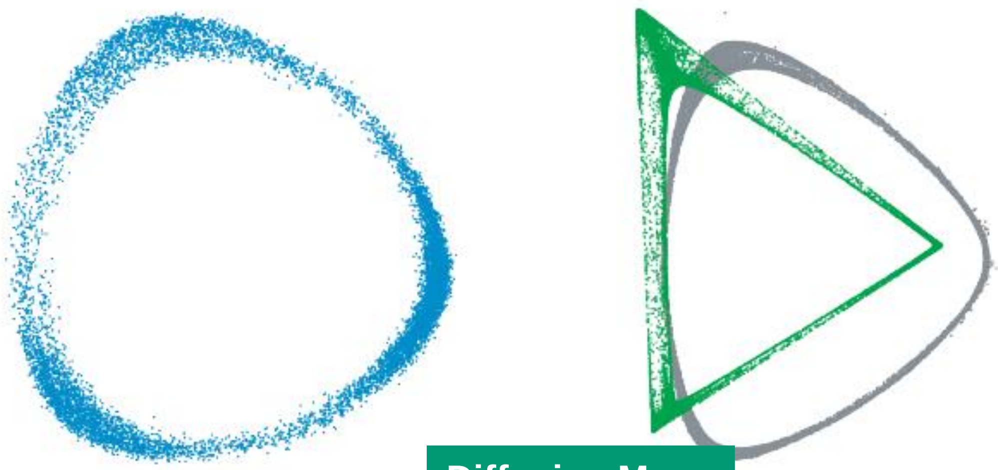
The embeddings do not separate the two groups.

# Manifold Learning Methods: **Caveats**

- Knowing the intrinsic dimension of the manifold can help recovering the projections
- The result of the algorithm heavily depends on setting well the hyperparameters such as the type of neighborhood graph (*k*-NN or radius neighbor) and neighborhood scale
- The algorithms heavily depend on the sampling method, that can affect measure of local density (and hence of relative scaling). Non uniform sampling can lead to undersired local deformation of the space.

Laplacian Eigenmap use: $L = I - D^{-1}S$

Diffusion maps use: $L = I - \left( \underset{\text{Rows normalization}}{D^{-1}S} + \underset{\text{Column normalization}}{SD^{-1}} \right)$



Meilă M, Zhang H. 2024
*Annu. Rev. Stat. Appl.* 11:393–417

*Effects of graph construction and renormalization, when the sampling density is highly nonuniform*

# Summary: techniques to generate non-linear embeddings

We have seen how to use decomposition of the following set of matrices:

Gram matrix: kPCA, kCCA, MDS

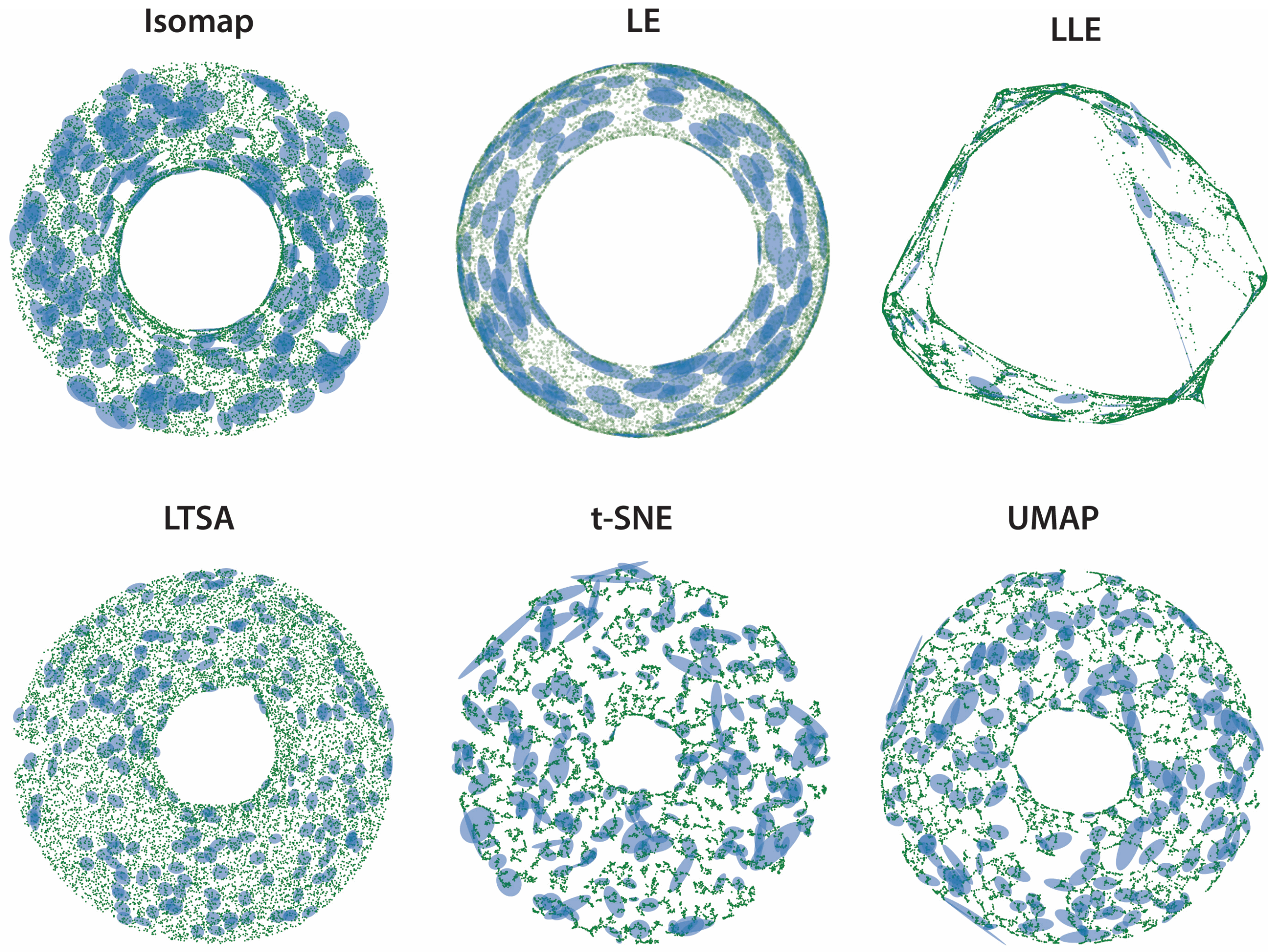
Graph Laplacian: Laplacian Eigenmaps

See also supplement on moodle for other techniques:

- Maximum variance unfolding
- Local Linear Embeddings (LLE)

# Manifold Learning Methods: **Trade-offs**

*Sensitivity and properties of learned representations depending on parameters' and algorithmic choices.*



Meilă M, Zhang H. 2024
*Annu. Rev. Stat. Appl.* 11:393–417