MACHINE LEARNING

Spectral Clustering
Outline of Today’s Lecture

• Introduce the principle of spectral clustering
• Show extension for other transformations of the space
  • Multi-dimensional scaling
  • Laplacian Eigenmaps
  • Isomaps

• Exercise the principle of eigen-decomposition underlying these methods
Non-Linear Manifolds

PCA and Kernel PCA belong to a more general class of methods that create non-linear manifolds based on *spectral decomposition*. (spectral decomposition of matrices is more frequently referred to as an eigenvalue decomposition.)

Depending on which matrix we decompose, we get a different set of projections.

- PCA decomposes the covariance matrix of the dataset \( \rightarrow \) generates rotations and projections in the original space
- kernel PCA decomposes the Gram matrix \( \rightarrow \) generates partitions of the space by regrouping the datapoints (tight clusters with RBF, quadrans for polynomial kernel)
Non-Linear Manifolds

• Spectral clustering decomposes the *Graph Laplacian matrix*: The Graph Laplacian is a matrix representation of a graph.

• Eigenvalue decomposition of this matrix determines *relationships* across datapoints induced by the similarity across datapoints embedded in the graph.

• The spectral decomposition of the *Graph Laplacian matrix* can be used to generate various projections, including scaling of the space, flattening and clustering.
Embed Data in a Graph

- Build a *similarity graph*
- Each vertex on the graph is a datapoint
Measure Distances in Graph

Construct the similarity matrix $S$ to denote whether points are close or far away to weight the edges of the graph:

$$
S = 
\begin{bmatrix}
0.9 & 0.8 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.7 & 0.9 \\
\end{bmatrix}
$$
Disconnected Graph (binary entries):
Two data-points are connected (S=1) if:

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

b) or if they are k-nearest neighbors (according to the similarity metric)
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; datapoints within a block are connected).
Connected Components in a Graph

• Next, we will see a method to discover the number of connected components.

• Knowing this number allows to identify clusters according to the similarity matrix chosen.
Graph Laplacian

Given a similarity matrix $S = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$ (4x4 example)

Construct the diagonal matrix $D$ composed of the sum on each line of $S$:

$$D = \begin{bmatrix} \sum_i S_{1i} & \ldots & 0 \\ 0 & \sum_i S_{2i} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & \ldots & \ldots & \sum_i S_{4i} \end{bmatrix}$$

$$L = D - S = \begin{bmatrix} \sum_i S_{1i} - 1 & 0 & 0 & -1 \\ 0 & \sum_i S_{2i} - 1 & -1 & 0 \\ 0 & -1 & \sum_i S_{3i} - 1 & 0 \\ -1 & 0 & 0 & \sum_i S_{4i} - 1 \end{bmatrix}$$

and then, build the Graph Laplacian matrix: $L = D - S$

$L$ is positive semi-definite $\rightarrow$ spectral decomposition possible
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:

\[ \Rightarrow \text{If we order the eigenvalues by increasing order:} \]

\[ \lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_M. \]

Theorem (see annexes):

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

\[ \Rightarrow \text{The multiplicity of the eigenvalue 0 determines the number of connected components in a graph.} \]

\[ \Rightarrow \text{The associated eigenvectors identify these connected components.} \]
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.
Spectral Clustering

⇒ The multiplicity of the eigenvalue 0 determines the number of connected components in a graph.
⇒ The associated eigenvectors identify these connected components.

→ 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

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)

\[
S(x^i, x^j) = e^{-\frac{\|x^i - x^j\|^2}{2\sigma^2}}
\]
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.
Spectral Clustering

When the similarity matrix is not sparse, the eigenvalue decomposition of the Laplacian matrix, yields rarely a solution with more than one eigenvalue zero.

We then have one eigenvector with one eigenvalue zero. All other eigenvalues are positive.

→ The first eigenvalue is then still zero but with multiplicity 1 only (fully connected graph)!

However, some of the other positive eigenvalues may be very close to 0.

Idea: the smallest eigenvalues (close to zero) provide also information on the partitioning of the graph (see solution exercise II)
Spectral Clustering

Algorithm in the general case ($S$ not binary)
1) Build the Laplacian matrix: $L = D - S$
2) Do the eigenvalue decomposition of the Laplacian matrix: $L = U \Lambda U^T$
3) Order the eigenvalues by increasing order:
   $$\lambda_1 = 0 \leq \lambda_2 \leq \ldots \leq \lambda_M$$
4) Apply a threshold on the eigenvalues, such that small $\lambda \to 0$
5) Determine the number of clusters by looking at the multiplicity of $\lambda = 0$ after step 4

This provides an indication of the number of clusters $K$.

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

Eigenvalues of the Laplacian matrix in $U$:

$$
U = \begin{bmatrix}
\begin{array}{ccc}
e_1^1 & e_2^1 & \cdots & e_{1}^M \\
e_1^2 & e_2^2 & \cdots & e_{1}^M \\
\vdots & \vdots & \ddots & \vdots \\
e_1^M & e_2^M & \cdots & e_{1}^M
\end{array}
\end{bmatrix}
$$

$$y^i = \begin{bmatrix}
e_i^1 \\
\vdots \\
e_i^M
\end{bmatrix}
$$

Construct an embedding of each of the $M$ datapoints $x^i$ through $y^i$.

This amounts to a non-linear mapping

$$X = \{x^i\}_{i=1}^M \xrightarrow{\phi} Y = \{y^i\}_{i=1}^M$$
Spectral Clustering

Construct an embedding of each of the \( M \) datapoints \( x^i \) through \( y^i \).

Points well grouped in original space generate grouped images \( y^i \).

→ Reduce dimensionality by picking \( K < M \) eigenvectors \( e^i \), \( i = 1...K \), on which the projections of \( y^i \), \( i = 1...M \), are well grouped.
Spectral Clustering

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:

$e^1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$

$e^2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

$e^3 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

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:

$y^1 = \begin{bmatrix} -1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix}$

$y^2 = \begin{bmatrix} -1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$

$y^3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$
Spectral Clustering

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:

$$y^1 = \begin{bmatrix} -1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix}, \quad y^2 = \begin{bmatrix} -1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}, \quad y^3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$
Spectral Clustering

Example: 3 datapoints in a fully connected graph

\[ S = \begin{bmatrix} 1 & 0.9 & 0.02 \\ 0.9 & 1 & 0.02 \\ 0.02 & 0.02 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 0.92 & -0.90 & -0.02 \\ -0.90 & 0.92 & -0.02 \\ -0.02 & -0.02 & 0.04 \end{bmatrix} \]

\( L \) has eigenvalue \( \lambda = 0 \) with multiplicity 1.

The second eigenvalue is small \( \lambda_2 = 0.06 \), whereas the 3rd one is large, \( \lambda_3 = 1.82 \).

with associated eigenvectors:

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

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

The images of the points are given by:

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

It makes sense to group eigenvectors with smallest eigenvalues.
Spectral Clustering

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:

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

The images of the points are given by:

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

Step 1: Embedding in y

Idea: Points close to one another have almost the same coordinates on the eigenvectors of $L$ with small eigenvalues.

Step 1: Do an eigenvalue decomposition of the Laplacian matrix $L$ and project the images of the datapoints onto the first $K$ eigenvectors with smallest eigenvalues (hence reducing the dimensionality of the images $y$).
Step 2:
Perform K-Means on the set of $y^1, \ldots, y^M \in \mathbb{R}^K$ vectors. Cluster datapoints $x$ according to their clustering after K-means on $y$. 
Spectral Clustering: exercise III

Consider a dataset composed of four points with two pairs of points that are close to each other, one pair being far from the other. More formally, assume that the similarity matrix looks as follows:

\[
S = \begin{bmatrix}
1 & 0.8 & 0 & 0 \\
0.8 & 1 & 0 & 0 \\
0 & 0 & 1 & 0.5 \\
0 & 0 & 0.5 & 1
\end{bmatrix}
\]

a) What are the eigenvalues and eigenvectors of \( L = D - S \)? How many connected components do you obtain?

b) What are the eigenvalues and eigenvectors of \( S \)? What do you notice? How could you infer clusters of points? (Hint: Look at the ratio of the eigenvalues)
Equivalency to other non-linear Embeddings

Example: 3 datapoints in a graph composed of 2 partitions

\[ S = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

\( L \) has eigenvalue \( \lambda = 0 \) with multiplicity two.

The eigenvectors of \( L \) are:

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

\( \lambda_1 = 0, \quad \lambda_2 = 0, \quad \lambda_3 = 2. \)

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

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

\( \lambda_1 = 2, \quad \lambda_2 = 1, \quad \lambda_3 = 0. \)

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

Careful: this is not true in arbitrary cases!
Equivalency to other non-linear Embeddings

kernel PCA: Eigenvalue decomposition of the matrix of similarity \( S \)

\[
S = U D U^T
\]

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 2)
The choice of kernel and kernel’s hyperparameters determine also the number of existing clusters kernel width.

From top to bottom, projections onto first 3 dual eigenvectors with RBF Kernel using kernel width of 0.8, 1.5, 2.5, resp.
Kernel PCA projections can help determine the kernel width. The largest eigenvalues grow as we get a better clustering.
There exists several variants on the Laplacian non-linear mappings, see a few examples in the next slides.
Laplacian Eigenmaps

Solve the generalized eigenvalue problem: $Le = \lambda De \Leftrightarrow \left( I - D^{-1}S \right) e = \lambda e$

If $D$ not invertible, solve: $\min_{y} e^{T}Le$ such that $e^{T}De = 1$.

Ensures minimal distortion while preventing arbitrary scaling.

Projections of images $y^{i}, i = 1...M$ on each vector $e^{i}, i = 1...K$ generate different embeddings of the datapoints.

Image courtesy from A. Singh
Laplacian Eigenmaps

Solve the generalized eigenvalue problem: $Le = \lambda De \iff \left( I - D^{-1}S \right)e = \lambda e$

If $D$ not invertible, solve: $\min_y e^T Le$ such that $e^T De = 1$.

Ensures minimal distortion while preventing arbitrary scaling.

The projections on the pair $e^1, e^3$ generate a flat embedding that enables a linear partitioning.
Multi-Dimensional Scaling (MDS)

Performs a scaled projection from the similarity matrix:

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

2) Then perform eigenvalue decomposition of \( S' \) yielding eigenvectors \( e^i, i = 1...M \).

3) Consider only the eigenvectors with positive eigenvalues.

4) Generate scaled projections \( y^j_i = \sqrt{\lambda_i} e^i_j \) (see example of isomap)

Flattens and normalizes but does not separate very well.
Isomap

Generalization of MDS using geodesic distances to generate $S$.

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

The geodesic distances encapsulates well the neighbouring. Combined with the MDS flattening of the space, they allow to extract well the 2 classes.
Variants to generate non-linear Embeddings

Eigenvalue decomposition of the following set of matrices:

–Graph Laplacian: \( L = D - S \)

–Scaled similarity matrix: \( D^{-1}S \) (Laplace Eigenmaps)

–Centered similarity matrix \( S'_{ij} = S_{ij} - \frac{1}{M} \sum_{k=1}^{M} S_{ik} - \frac{1}{M} \sum_{k=1}^{M} S_{kj} + \frac{1}{M^2} \sum_{k,l=1}^{M} S_{kl} \)

and normalized projections (Multidimensional Scaling - MDS)

All lead to a non-linear embedding yielding to a grouping of the points in the projected space of images \( Y = \{ y^i \}_{i=1}^{M} \).

Applying K-means on these projections amounts to spectral clustering.
Summary

We have seen several ways in which to perform a non-linear embedding of the space, namely:

- Kernel PCA - appropriate for data that live in a single modality
- Kernel CCA – appropriate to compare embeddings across different modalities encoding the data
- Kernel K-means: proceed to clustering and non-linear embedding simultaneously
- Spectral clustering: performs K-means after non-linear embedding using variants on projections onto the eigenvectors of the similarity matrix.

All these methods can generally be referred to as spectral methods. They are very powerful, if used well!

Their power relies on you choosing well the kernel. To this end, this part of the class is meant to have you develop a good understanding of the type of deformations of the space generated by a specific kernel and its parameters.