In the previous lecture, we discussed how \( K \)-means can be used to cluster points in \( \mathbb{R}^p \).

Spectral clustering:

- Very popular clustering method.
- Often outperforms other methods such as \( K \)-means.
- Can be used for various “types” of data (not only points in \( \mathbb{R}^p \)).
- Easy to implement. Only uses basic linear algebra.
In the previous lecture, we discussed how $K$-means can be used to cluster points in $\mathbb{R}^p$.

Spectral clustering:

- Very popular clustering method.
- Often outperforms other methods such as $K$-means.
- Can be used for various “types” of data (not only points in $\mathbb{R}^p$).
- Easy to implement. Only uses basic linear algebra.

Overview of spectral clustering:
Spectral clustering: overview

In the previous lecture, we discussed how $K$-means can be used to cluster points in $\mathbb{R}^p$.

Spectral clustering:

- Very popular clustering method.
- Often outperforms other methods such as $K$-means.
- Can be used for various “types” of data (not only points in $\mathbb{R}^p$).
- Easy to implement. Only uses basic linear algebra.

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects.
In the previous lecture, we discussed how $K$-means can be used to cluster points in $\mathbb{R}^p$.

Spectral clustering:

- Very popular clustering method.
- Often outperforms other methods such as $K$-means.
- Can be used for various “types” of data (not only points in $\mathbb{R}^p$).
- Easy to implement. Only uses basic linear algebra.

Overview of spectral clustering:

1. Construct a *similarity matrix* measuring the similarity of pairs of objects.
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
Spectral clustering: overview

In the previous lecture, we discussed how $K$-means can be used to cluster points in $\mathbb{R}^p$.

Spectral clustering:

- Very popular clustering method.
- Often outperforms other methods such as $K$-means.
- Can be used for various “types” of data (not only points in $\mathbb{R}^p$).
- Easy to implement. Only uses basic linear algebra.

Overview of spectral clustering:

1. Construct a similarity matrix measuring the similarity of pairs of objects.
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the graph Laplacian.
Spectral clustering: overview

In the previous lecture, we discussed how $K$-means can be used to cluster points in $\mathbb{R}^p$.

Spectral clustering:

- Very popular clustering method.
- Often outperforms other methods such as $K$-means.
- Can be used for various “types” of data (not only points in $\mathbb{R}^p$).
- Easy to implement. Only uses basic linear algebra.

Overview of spectral clustering:

1. Construct a similarity matrix measuring the similarity of pairs of objects.
2. Use the similarity matrix to construct a (weighted or unweighted) graph.
3. Compute eigenvectors of the graph Laplacian.
4. Cluster the graph using the eigenvectors of the graph Laplacian using the $K$-means algorithm.
We will use the following notation/conventions:

- \( G = (V, E) \) a graph with vertex set \( V = \{v_1, \ldots, v_n\} \) and edge set \( E \subseteq V \times V \).
We will use the following notation/conventions:

- $G = (V, E)$ a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$.
- Each edge carries a weight $w_{ij} \geq 0$. 

The adjacency matrix of $G$ is $W = W_G = (w_{ij})_{n \times n}$. We will assume $W$ is symmetric (undirected graphs).

The degree of $v_i$ is $d_i := \sum_{j=1}^{n} w_{ij}$.

The degree matrix of $G$ is $D := \text{diag}(d_1, \ldots, d_n)$.

We denote the complement of $A \subset V$ by $\overline{A}$.

If $A \subset V$, then we let $1_A = (f_1, \ldots, f_n)^T \in \mathbb{R}^n$, where $f_i = 1$ if $v_i \in A$ and 0 otherwise.
We will use the following notation/conventions:

- $G = (V, E)$ a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$.
- Each edge carries a weight $w_{ij} \geq 0$.
- The adjacency matrix of $G$ is $W = W_G = (w_{ij})_{i,j=1}^n$. We will assume $W$ is symmetric (undirected graphs).
We will use the following notation/conventions:

- $G = (V, E)$ a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$.
- Each edge carries a weight $w_{ij} \geq 0$.
- The adjacency matrix of $G$ is $W = W_G = (w_{ij})_{i,j=1}^n$. We will assume $W$ is symmetric (undirected graphs).
- The degree of $v_i$ is

$$d_i := \sum_{j=1}^n w_{ij}.$$
We will use the following notation/conventions:

- $G = (V, E)$ a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$.
- Each edge carries a weight $w_{ij} \geq 0$.
- The adjacency matrix of $G$ is $W = W_G = (w_{ij})_{i,j=1}^n$. We will assume $W$ is symmetric (undirected graphs).
- The degree of $v_i$ is
  $$d_i := \sum_{j=1}^n w_{ij}.$$
- The degree matrix of $G$ is $D := \text{diag}(d_1, \ldots, d_n)$. 
We will use the following notation/conventions:

- $G = (V, E)$ a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$.
- Each edge carries a weight $w_{ij} \geq 0$.
- The adjacency matrix of $G$ is $W = W_G = (w_{ij})_{i,j=1}^n$. We will assume $W$ is symmetric (undirected graphs).
- The degree of $v_i$ is
  \[ d_i := \sum_{j=1}^n w_{ij}. \]
- The degree matrix of $G$ is $D := \text{diag}(d_1, \ldots, d_n)$.
- We denote the complement of $A \subset V$ by $\overline{A}$.
We will use the following notation/conventions:

- **$G = (V, E)$** a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subset V \times V$.
- Each edge carries a *weight* $w_{ij} \geq 0$.
- The adjacency matrix of $G$ is $W = W_G = (w_{ij})_{i,j=1}^n$. We will assume $W$ is symmetric (undirected graphs).
- The *degree* of $v_i$ is
  \[ d_i := \sum_{j=1}^n w_{ij}. \]
- The *degree matrix* of $G$ is $D := \text{diag}(d_1, \ldots, d_n)$.
- We denote the complement of $A \subset V$ by $\overline{A}$.
- If $A \subset V$, then we let $\mathbbm{1}_A = (f_1, \ldots, f_n)^T \in \mathbb{R}^n$, where $f_i = 1$ if $v_i \in A$ and 0 otherwise.
We assume we are given a measure of similarity $s$ between data points $x_1, \ldots, x_n \in \mathcal{X}$:

$$s : \mathcal{X} \times \mathcal{X} \to [0, \infty).$$
We assume we are given a measure of similarity \( s \) between data points \( x_1, \ldots, x_n \in \mathcal{X} \):

\[
s : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty).
\]

We denote by \( s_{ij} := s(x_i, x_j) \) the \textit{measure of similarity} between \( x_i \) and \( x_j \).
We assume we are given a measure of similarity $s$ between data points $x_1, \ldots, x_n \in \mathcal{X}$:

$$s : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty).$$

We denote by $s_{ij} := s(x_i, x_j)$ the measure of similarity between $x_i$ and $x_j$.

Equivalently, we may assume we have a measure of distance between data points (e.g. $(\mathcal{X}, d)$ is a metric space).
Similary graphs

- We assume we are given a measure of similarity $s$ between data points $x_1, \ldots, x_n \in \mathcal{X}$:

  $$s : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty).$$

- We denote by $s_{ij} := s(x_i, x_j)$ the measure of similarity between $x_i$ and $x_j$.

- Equivalently, we may assume we have a measure of distance between data points (e.g. $(\mathcal{X}, d)$ is a metric space).

- Let $d_{ij} := d(x_i, x_j)$, the distance between $x_i$ and $x_j$. 


We assume we are given a measure of similarity $s$ between data points $x_1, \ldots, x_n \in \mathcal{X}$:

$$s : \mathcal{X} \times \mathcal{X} \to [0, \infty).$$

We denote by $s_{ij} := s(x_i, x_j)$ the measure of similarity between $x_i$ and $x_j$.

Equivalently, we may assume we have a measure of distance between data points (e.g. $(\mathcal{X}, d)$ is a metric space).

Let $d_{ij} := d(x_i, x_j)$, the distance between $x_i$ and $x_j$.

From $d_{ij}$ (or $s_{ij}$), we naturally build a similarity graph.
We assume we are given a measure of similarity $s$ between data points $x_1, \ldots, x_n \in \mathcal{X}$:

$$s : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty).$$

We denote by $s_{ij} := s(x_i, x_j)$ the measure of similarity between $x_i$ and $x_j$.

Equivalently, we may assume we have a measure of distance between data points (e.g. $(\mathcal{X}, d)$ is a metric space).

Let $d_{ij} := d(x_i, x_j)$, the distance between $x_i$ and $x_j$.

From $d_{ij}$ (or $s_{ij}$), we naturally build a similarity graph.

We will discuss 3 popular ways of building a similarity graph.
Vertex set $= \{v_1, \ldots, v_n\}$ where $n$ is the number of data points.
Vertex set = \( \{v_1, \ldots, v_n\} \) where \( n \) is the number of data points.

1. **The \( \epsilon \)-neighborhood graph**: Connect all points whose pairwise distances are smaller than some \( \epsilon > 0 \). We usually don’t weight the edges. The graph is thus a simple graph (unweighted, undirected graph containing no loops or multiple edges).

2. **The \( k \)-nearest neighbor graph**: The goal is to connect \( v_i \) to \( v_j \) if \( x_j \) is among the \( k \) nearest neighbors of \( x_i \). However, this leads to a directed graph. We therefore define the **mutual \( k \)-nearest neighbor graph**: \( v_i \) is adjacent to \( v_j \) if \( x_j \) is among the \( k \) nearest neighbors of \( x_i \) **and** \( x_i \) is among the \( k \) nearest neighbors of \( x_j \). We weight the edges by the similarity of their endpoints.
Vertex set = \( \{v_1, \ldots, v_n\} \) where \( n \) is the number of data points.

1. **The \( \epsilon \)-neighborhood graph**: Connect all points whose pairwise distances are smaller than some \( \epsilon > 0 \). We usually don’t weight the edges. The graph is thus a simple graph (unweighted, undirected graph containing no loops or multiple edges).

2. **The \( k \)-nearest neighbor graph**: The goal is to connect \( v_i \) to \( v_j \) if \( x_j \) is among the \( k \) nearest neighbors of \( x_i \). However, this leads to a directed graph. We therefore define:
Vertex set = \( \{v_1, \ldots, v_n\} \) where \( n \) is the number of data points.

1. **The \( \epsilon \)-neighborhood graph:** Connect all points whose pairwise distances are smaller than some \( \epsilon > 0 \). We usually don’t weight the edges. The graph is thus a simple graph (unweighted, undirected graph containing no loops or multiple edges).

2. **The \( k \)-nearest neighbor graph:** The goal is to connect \( v_i \) to \( v_j \) if \( x_j \) is among the \( k \) nearest neighbors of \( x_i \). However, this leads to a directed graph. We therefore define:
   - the \( k \)-nearest neighbor graph: \( v_i \) is adjacent to \( v_j \) iff \( x_j \) is among the \( k \) nearest neighbors of \( x_i \) **OR** \( x_i \) is among the \( k \) nearest neighbors of \( x_j \).
Vertex set = \{v_1, \ldots, v_n\} where \( n \) is the number of data points.

1. **The \( \epsilon \)-neighborhood graph**: Connect all points whose pairwise distances are smaller than some \( \epsilon > 0 \). We usually don’t weight the edges. The graph is thus a simple graph (unweighted, undirected graph containing no loops or multiple edges).

2. **The \( k \)-nearest neighbor graph**: The goal is to connect \( v_i \) to \( v_j \) if \( x_j \) is among the \( k \) nearest neighbors of \( x_i \). However, this leads to a directed graph. We therefore define:
   - the \( k \)-nearest neighbor graph: \( v_i \) is adjacent to \( v_j \) iff \( x_j \) is among the \( k \) nearest neighbors of \( x_i \) OR \( x_i \) is among the \( k \) nearest neighbors of \( x_j \).
   - the mutual \( k \)-nearest neighbor graph: \( v_i \) is adjacent to \( v_j \) iff \( x_j \) is among the \( k \) nearest neighbors of \( x_i \) AND \( x_i \) is among the \( k \) nearest neighbors of \( x_j \).
Vertex set = \{v_1, \ldots, v_n\} where \(n\) is the number of data points.

1. **The \(\epsilon\)-neighborhood graph:** Connect all points whose pairwise distances are smaller than some \(\epsilon > 0\). We usually don’t weight the edges. The graph is thus a simple graph (unweighted, undirected graph containing no loops or multiple edges).

2. **The \(k\)-nearest neighbor graph:** The goal is to connect \(v_i\) to \(v_j\) if \(x_j\) is among the \(k\) nearest neighbors of \(x_i\). However, this leads to a directed graph. We therefore define:
   - the \(k\)-nearest neighbor graph: \(v_i\) is adjacent to \(v_j\) iff \(x_j\) is among the \(k\) nearest neighbors of \(x_i\) OR \(x_i\) is among the \(k\) nearest neighbors of \(x_j\).
   - the mutual \(k\)-nearest neighbor graph: \(v_i\) is adjacent to \(v_j\) iff \(x_j\) is among the \(k\) nearest neighbors of \(x_i\) AND \(x_i\) is among the \(k\) nearest neighbors of \(x_j\).

We weight the edges by the similarity of their endpoints.
The fully connected graph: Connect all points with edge weights $s_{ij}$. 

Note: $\sigma^2$ controls the width of the neighborhoods. 

All graphs mentioned above are regularly used in spectral clustering.
The fully connected graph: Connect all points with edge weights $s_{ij}$. For example, one could use the *Gaussian similarity function* to represent a local neighborhood relationships:

$$s_{ij} = s(x_i, x_j) = \exp\left(-\|x_i - x_j\|^2/(2\sigma^2)\right) \quad (\sigma^2 > 0).$$

Note: $\sigma^2$ controls the width of the neighborhoods.
The fully connected graph: Connect all points with edge weights $s_{ij}$. For example, one could use the Gaussian similarity function to represent a local neighborhood relationships:

$$s_{ij} = s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2\sigma^2)) \quad (\sigma^2 > 0).$$

Note: $\sigma^2$ controls the width of the neighborhoods.

All graphs mentioned above are regularly used in spectral clustering.
There are three commonly used definitions of the graph Laplacian:

1. **The unnormalized Laplacian** is

   \[ L := D - W. \]
There are three commonly used definitions of the graph Laplacian:

1. **The unnormalized Laplacian** is

   \[ L := D - W. \]

2. **The normalized symmetric Laplacian** is

   \[ L_{\text{sym}} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}. \]
There are three commonly used definitions of the graph Laplacian:

1. **The unnormalized Laplacian** is
   \[
   L := D - W.
   \]

2. **The normalized symmetric Laplacian** is
   \[
   L_{\text{sym}} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2}.
   \]

3. **The normalized “random walk” Laplacian** is
   \[
   L_{\text{rw}} := D^{-1}L = I - D^{-1}W.
   \]
There are three commonly used definitions of the graph Laplacian:

1. **The unnormalized Laplacian** is
   \[
   L := D - W.
   \]

2. **The normalized symmetric Laplacian** is
   \[
   L_{\text{sym}} := D^{-1/2} LD^{-1/2} = I - D^{-1/2} WD^{-1/2}.
   \]

3. **The normalized “random walk” Laplacian** is
   \[
   L_{\text{rw}} := D^{-1} L = I - D^{-1} W.
   \]

We begin by studying properties of the **unnormalized Laplacian**.
Proposition: The matrix $L$ satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:
   $$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$

2. $L$ is symmetric and positive semidefinite.

3. 0 is an eigenvalue of $L$ with associated constant eigenvector 1.

Proof:
To prove (1),
$$f^T L f = f^T D f - f^T W f = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j = \frac{1}{2} \left( \sum_{i=1}^{n} d_i f_i^2 - 2 \sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2 \right) = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$ (2) follows from (1). (3) is easy.
The unnormalized Laplacian

**Proposition:** The matrix $L$ satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:

   $$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$
Proposition: The matrix $L$ satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:
   \[
   f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.
   \]

2. $L$ is symmetric and positive semidefinite.
Proposition: The matrix $L$ satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:
   \[ f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2. \]

2. $L$ is symmetric and positive semidefinite.
3. 0 is an eigenvalue of $L$ with associated constant eigenvector $\mathbf{1}$. 
Proposition: The matrix $L$ satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:
   
   $$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$ 

2. $L$ is symmetric and positive semidefinite.
3. 0 is an eigenvalue of $L$ with associated constant eigenvector $\mathbb{1}$.

Proof:
**Proposition:** The matrix $L$ satisfies the following properties:

1. For any $f \in \mathbb{R}^n$:
   \[
   f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.
   \]

2. $L$ is symmetric and positive semidefinite.

3. 0 is an eigenvalue of $L$ with associated constant eigenvector $\mathbb{1}$.

**Proof:** To prove (1),

\[
\begin{align*}
    f^T L f &= f^T D f - f^T W f \\
    &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j \\
    &= \frac{1}{2} \left( \sum_{i=1}^{n} d_i f_i^2 - 2 \sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2 \right) \\
    &= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.
\end{align*}
\]

(2) follows from (1). (3) is easy.
Proposition: Let $G$ be an undirected graph with non-negative weights. Then:

1. The multiplicity $k$ of the eigenvalue $0$ of $L$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. The eigenspace of eigenvalue $0$ is spanned by the indicator vectors $1_{A_1}, \ldots, 1_{A_k}$ of those components.

Proof: If $f$ is an eigenvector associated to $\lambda = 0$, then

$$0 = f^T L f = \sum_{i,j} w_{ij} (f_i - f_j)^2.$$

It follows that $f_i = f_j$ whenever $w_{ij} > 0$. Thus $f$ is constant on the connected components of $G$. We conclude that the eigenspace of $0$ is contained in $\text{span}(1_{A_1}, \ldots, 1_{A_k})$. Conversely, it is not hard to see that each $1_{A_i}$ is an eigenvector associated to $0$ (write $L$ in block diagonal form).
**Proposition**: Let $G$ be an undirected graph with non-negative weights. Then:

1. The multiplicity $k$ of the eigenvalue $0$ of $L$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.
Proposition: Let $G$ be an undirected graph with non-negative weights. Then:

1. The multiplicity $k$ of the eigenvalue $0$ of $L$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. The eigenspace of eigenvalue $0$ is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.
Proposition: Let $G$ be an undirected graph with non-negative weights. Then:

1. The multiplicity $k$ of the eigenvalue 0 of $L$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.

2. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

Proof:
Proposition: Let $G$ be an undirected graph with non-negative weights. Then:

1. The multiplicity $k$ of the eigenvalue $0$ of $L$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.
2. The eigenspace of eigenvalue $0$ is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

Proof: If $f$ is an eigenvector associate to $\lambda = 0$, then

$$0 = f^T L f = \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$

It follows that $f_i = f_j$ whenever $w_{ij} > 0$. Thus $f$ is constant on the connected components of $G$. We conclude that the eigenspace of $0$ is contained in $\text{span}(\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k})$. Conversely, it is not hard to see that each $\mathbb{1}_{A_i}$ is an eigenvector associated to $0$ (write $L$ in block diagonal form). \[ \square \]
Proposition: The normalized Laplacians satisfy the following properties:

1. For every $f \in \mathbb{R}^n$, we have
   \[ f^T L_{\text{sym}} f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left( \frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2. \]

2. $\lambda$ is an eigenvalue of $L_{\text{rw}}$ with eigenvector $u$ if and only if $\lambda$ is an eigenvalue of $L_{\text{sym}}$ with eigenvector $w = D^{1/2}u$.

3. $\lambda$ is an eigenvalue of $L_{\text{rw}}$ with eigenvector $u$ if and only if $\lambda$ and $u$ solve the generalized eigenproblem $Lu = \lambda Du$.

Proof: The proof of (1) is similar to the proof of the analogous result for the unnormalized Laplacian. (2) and (3) follow easily by using appropriate rescalings.
Proposition: Let $G$ be an undirected graph with non-negative weights. Then:

1. The multiplicity $k$ of the eigenvalue 0 of both $L_{\text{sym}}$ and $L_{\text{rw}}$ equals the number of connected components $A_1, \ldots, A_k$ in the graph.

2. For $L_{\text{rw}}$, the eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_i}, i = 1, \ldots, k$.

3. For $L_{\text{sym}}$, the eigenspace of eigenvalue 0 is spanned by the vectors $D^{1/2} \mathbb{1}_{A_i}, i = 1, \ldots, k$.

Proof: Similar to the proof of the analogous result for the unnormalized Laplacian.