COT 6936: Topics in Algorithms

Giri Narasimhan

ECS 254A / EC 2474; Phone x3748; Email: giri@cs.fiu.edu
HOMEPAGE: http://www.cs.fiu.edu/~giri
https://moodle.cis.fiu.edu/v2.1/course/view.php?id=612

Mar 27, 2014

Presentation Outline

Source

- Most of the material is from notes by Abhiram Ranade; http://www.cse.iitb.ac.in/~ranade/miscdocs/svd.pdf

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
- Search: Given a database of docs/images, find closest match to query

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
- Search: Given a database of docs/images, find closest match to query
- Clustering: Organize a database of docs/images into "clusters"

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
- Search: Given a database of docs/images, find closest match to query
- Clustering: Organize a database of docs/images into "clusters"
- Compression: useful for images

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
- Search: Given a database of docs/images, find closest match to query
- Clustering: Organize a database of docs/images into "clusters"
- Compression: useful for images
- Summarization: Find parts of document most representative of paragraph

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
- Search: Given a database of docs/images, find closest match to query
- Clustering: Organize a database of docs/images into "clusters"
- Compression: useful for images
- Summarization: Find parts of document most representative of paragraph
- Graph Partitioning: partition graph into dense subgraphs; useful in VLSI where densely connected parts will be laid out on a chip; also useful in divide-and-conquer algorithms.

Applications

- Many methods are based on Principal Component Analysis (PCA) and Singular Value Decomposition (SVD)
- Search: Given a database of docs/images, find closest match to query
- Clustering: Organize a database of docs/images into "clusters"
- Compression: useful for images
- Summarization: Find parts of document most representative of paragraph
- Graph Partitioning: partition graph into dense subgraphs; useful in VLSI where densely connected parts will be laid out on a chip; also useful in divide-and-conquer algorithms.
- Random Walks: Markov Chain Mixing, Google Page Rank
- Graph Connectivity, Coloring, . . .

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Examples:

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Examples:

- Images: a_{ij} = grayscale value of j-th pixel in i-th image

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Examples:

- Images: a_{ij} = grayscale value of j-th pixel in i-th image
- Documents: a_{ij} = does document i contain term j

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Examples:

- Images: a_{ij} = grayscale value of j-th pixel in i-th image
- Documents: a_{ij} = does document i contain term j
- Medical: a_{ij} = measured parameter j for subject i

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Examples:

- Images: a_{ij} = grayscale value of j-th pixel in i-th image
- Documents: a_{ij} = does document i contain term j
- Medical: a_{ij} = measured parameter j for subject i

Typical Solution: Rows (points) are in low-dimensional subspace (Rank r) plus some noise.

Common Theme

Given n points in m-dimensional space, typically given to us as an $n \times m$ matrix A, where the i-th row gives cooridinates of the i-th point.

Question: Disocver structure, shape, correlations, patterns, ...

Examples:

- Images: a_{ij} = grayscale value of j-th pixel in i-th image
- Documents: a_{ij} = does document i contain term j
- Medical: a_{ij} = measured parameter j for subject i

Typical Solution: Rows (points) are in low-dimensional subspace (Rank r) plus some noise. In other words,

$$A = PT,$$

where P is a $n \times r$ matrix and T is a $r \times m$ matrix.

Geometric Interpretations

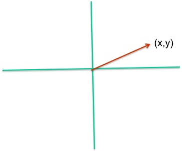
- 1-dimensional array with n items

Geometric Interpretations

- 1-dimensional array with n items
  - Point in space

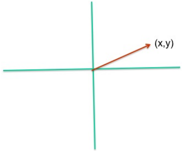
Geometric Interpretations

- 1-dimensional array with n items
  - Point in space



Geometric Interpretations

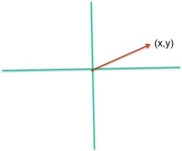
- 1-dimensional array with n items
  - Point in space



  - Vector
- Matrix with n rows and m columns

Geometric Interpretations

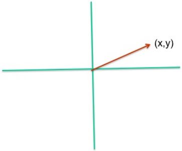
- 1-dimensional array with n items
  - Point in space



  - Vector
- Matrix with n rows and m columns
  - n points in m-dimensional space

Geometric Interpretations

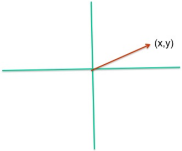
- 1-dimensional array with n items
  - Point in space



  - Vector
- Matrix with n rows and m columns
  - n points in m-dimensional space
  - More important interpretation . . .

Geometric Interpretations

- 1-dimensional array with n items
  - Point in space



  - Vector
- Matrix with n rows and m columns
  - n points in m-dimensional space
  - More important interpretation ...

Geometric Interpretations ... 2

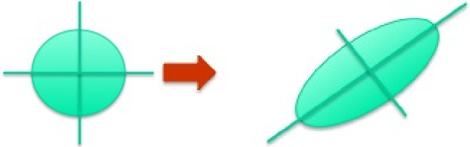
- Matrix with n rows and m columns

Geometric Interpretations ... 2

- Matrix with n rows and m columns
  - Linear transformations $\mathcal{R}^m \leftrightarrow \mathcal{R}^n$

Geometric Interpretations ... 2

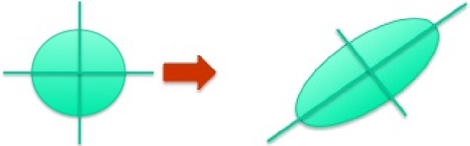
- Matrix with n rows and m columns
  - Linear transformations $\mathcal{R}^m \leftrightarrow \mathcal{R}^n$



- Eigenvalues and Eigenvectors

Geometric Interpretations ... 2

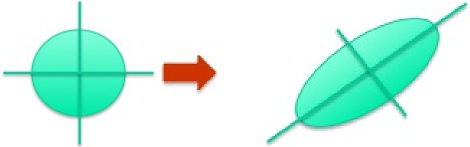
- Matrix with n rows and m columns
  - Linear transformations $\mathcal{R}^m \leftrightarrow \mathcal{R}^n$



- Eigenvalues and Eigenvectors
  - Solutions to equation $Ax = \lambda x$

Geometric Interpretations ... 2

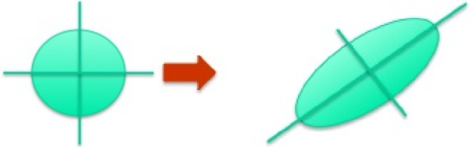
- Matrix with n rows and m columns
  - Linear transformations $\mathcal{R}^m \leftrightarrow \mathcal{R}^n$



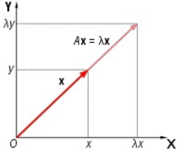
- Eigenvalues and Eigenvectors
  - Solutions to equation $Ax = \lambda x$
  - Under transformation A, an eigenvector does not change in direction, its magnitude changes by factor λ

Geometric Interpretations ... 2

- Matrix with n rows and m columns
  - Linear transformations $\mathcal{R}^m \leftrightarrow \mathcal{R}^n$

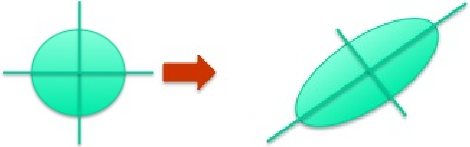


- Eigenvalues and Eigenvectors
  - Solutions to equation $Ax = \lambda x$
  - Under transformation A, an eigenvector does not change in direction, its magnitude changes by factor λ

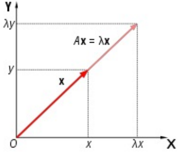


Geometric Interpretations ... 2

- Matrix with n rows and m columns
  - Linear transformations $\mathcal{R}^m \leftrightarrow \mathcal{R}^n$



- Eigenvalues and Eigenvectors
  - Solutions to equation $Ax = \lambda x$
  - Under transformation A, an eigenvector does not change in direction, its magnitude changes by factor λ



  - Decomposition: $A = Q \Lambda Q^{-1}$

Geometric Interpretations ... 3

- First Singular Value and Singular Vector

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2$$

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2$$

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

implying that Av_1 is in the same direction as u_1

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

    implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T
  - First singular value $\sigma_1 = \|Av_1\|_2 = \|u_1^T A\|_2$.
- Additional Singular vectors

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

    implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T
  - First singular value $\sigma_1 = \|Av_1\|_2 = \|u_1^T A\|_2$.
- Additional Singular vectors
  - $A_1 = Av_1 v_1^T = \sigma_1 u_1 v_1^T$ is of rank 1 and is the best rank 1 approximation to A

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

    $$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

    implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T
  - First singular value $\sigma_1 = \|Av_1\|_2 = \|u_1^T A\|_2$.
- Additional Singular vectors
  - $A_1 = Av_1 v_1^T = \sigma_1 u_1 v_1^T$ is of rank 1 and is the best rank 1 approximation to A
  - If $A' = A - A_1$, then computing u_2, v_2, σ_2 will give us the second Singular vector and value, . . .

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

    $$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

    implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T
  - First singular value $\sigma_1 = \|Av_1\|_2 = \|u_1^T A\|_2$.
- Additional Singular vectors
  - $A_1 = Av_1 v_1^T = \sigma_1 u_1 v_1^T$ is of rank 1 and is the best rank 1 approximation to A
  - If $A' = A - A_1$, then computing u_2, v_2, σ_2 will give us the second Singular vector and value, ...
  - k-th singular vector is orthogonal to all previous ones

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

    $$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

    implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T
  - First singular value $\sigma_1 = \|Av_1\|_2 = \|u_1^T A\|_2$.
- Additional Singular vectors
  - $A_1 = Av_1 v_1^T = \sigma_1 u_1 v_1^T$ is of rank 1 and is the best rank 1 approximation to A
  - If $A' = A - A_1$, then computing u_2, v_2, σ_2 will give us the second Singular vector and value, ...
  - k-th singular vector is orthogonal to all previous ones
  - Thus: $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$

Geometric Interpretations ... 3

- First Singular Value and Singular Vector
  - Under A, singular unit vector stretches the most

$$\max \|Av_1\|_2 \quad \text{or} \quad \max \|u_1^T A\|_2 \quad \text{or} \quad \max \|u_1^T A v_1\|_2,$$

    implying that Av_1 is in the same direction as u_1 OR that $u_1^T A$ is in the same direction as v_1^T
  - First singular value $\sigma_1 = \|Av_1\|_2 = \|u_1^T A\|_2$.
- Additional Singular vectors
  - $A_1 = Av_1 v_1^T = \sigma_1 u_1 v_1^T$ is of rank 1 and is the best rank 1 approximation to A
  - If $A' = A - A_1$, then computing u_2, v_2, σ_2 will give us the second Singular vector and value, ...
  - k-th singular vector is orthogonal to all previous ones
  - Thus: $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ and $A_r = A$, where $\text{rank}(A) = r$

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;
Let Σ_k be a diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along diagonal;

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;
Let Σ_k be a diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along diagonal;
Thus, $AV_r = U + r\Sigma_r$;

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;
Let Σ_k be a diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along diagonal;
Thus, $AV_r = U + r\Sigma_r$;
Since $V^T = V^{-1}$, we have the SVD as

$$A = U\Sigma V^T,$$

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;
Let Σ_k be a diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along diagonal;
Thus, $AV_r = U + r\Sigma_r$;
Since $V^T = V^{-1}$, we have the SVD as

$$A = U\Sigma V^T,$$

which can be computed in $O(mn^2 + m^2 n)$ time [Golub and van Loan, *Matrix Computations*, 1996]

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;
Let Σ_k be a diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along diagonal;
Thus, $AV_r = U + r\Sigma_r$;
Since $V^T = V^{-1}$, we have the SVD as

$$A = U\Sigma V^T,$$

which can be computed in $O(mn^2 + m^2 n)$ time [Golub and van Loan, *Matrix Computations*, 1996]
Also A_k is the best rank k approximation to A.

Singular Value Decomposition (SVD)

Let U_k be a matrix with columns $u_1, \ldots, u_k$;
Let V_k be a matrix with columns $v_1, \ldots, v_k$;
Let Σ_k be a diagonal matrix with $\sigma_1, \ldots, \sigma_k$ along diagonal;
Thus, $AV_r = U + r\Sigma_r$;
Since $V^T = V^{-1}$, we have the SVD as

$$A = U\Sigma V^T,$$

which can be computed in $O(mn^2 + m^2n)$ time [Golub and van Loan, *Matrix Computations*, 1996]
Also A_k is the best rank k approximation to A.
Furthermore, $\|A - A_k\|_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_r^2$.

Singular Values/Vectors vs Eigenvectors/values

- Singular vectors approximate directions of rows/columns of matrix;

Singular Values/Vectors vs Eigenvectors/values

- Singular vectors approximate directions of rows/columns of matrix;
- Singular values are always real; eigenvalues may be imaginary;

Singular Values/Vectors vs Eigenvectors/values

- Singular vectors approximate directions of rows/columns of matrix;
- Singular values are always real; eigenvalues may be imaginary;
- Left singular vectors = left eigenvectors of AA^T;

Singular Values/Vectors vs Eigenvectors/values

- Singular vectors approximate directions of rows/columns of matrix;
- Singular values are always real; eigenvalues may be imaginary;
- Left singular vectors = left eigenvectors of AA^T;
- Right singular vectors = right eigenvectors of A^TA;

Singular Values/Vectors vs Eigenvectors/values

- Singular vectors approximate directions of rows/columns of matrix;
- Singular values are always real; eigenvalues may be imaginary;
- Left singular vectors = left eigenvectors of AA^T;
- Right singular vectors = right eigenvectors of A^TA;
- $\lambda_i = \sigma_i^2$, $i = 1, \ldots, r$

Graph Bisection

- If A is the adjacency matrix, then the Laplacian,

$$\mathcal{L} = M - A,$$

where M is the diagonal matrix of vertex degrees.

Graph Bisection

- If A is the adjacency matrix, then the Laplacian,

$$\mathcal{L} = M - A,$$

  where M is the diagonal matrix of vertex degrees.
- $\mathcal{L}$ is positive semi-definite (PSD),

Graph Bisection

- If A is the adjacency matrix, then the Laplacian,

$$\mathcal{L} = M - A,$$

  where M is the diagonal matrix of vertex degrees.
- $\mathcal{L}$ is positive semi-definite (PSD), i.e., all eigenvalues are non-negative;

Graph Bisection

- If A is the adjacency matrix, then the Laplacian,

$$\mathcal{L} = M - A,$$

  where M is the diagonal matrix of vertex degrees.
- $\mathcal{L}$ is positive semi-definite (PSD), i.e., all eigenvalues are non-negative;
- $\mathcal{L}$ has smallest eigenvalue $= 0$

Graph Bisection

- If A is the adjacency matrix, then the Laplacian,

$$\mathcal{L} = M - A,$$

  where M is the diagonal matrix of vertex degrees.
- $\mathcal{L}$ is positive semi-definite (PSD), i.e., all eigenvalues are non-negative;
- $\mathcal{L}$ has smallest eigenvalue $= 0$
- Oddly enough, the eigenvector e_2 for the second smallest eigenvalue λ_2 provides info on bisection

Graph Bisection ... 2

- Compute the eigenvector for the second smallest eigenvalue, e_2

Graph Bisection ... 2

- Compute the eigenvector for the second smallest eigenvalue, e_2
- Use the signs of the vector to give a bisection

Graph Bisection ... 2

- Compute the eigenvector for the second smallest eigenvalue, e_2
- Use the signs of the vector to give a bisection
- Can be used to get bisections with $n/2$ vertices – by using the median value in e_2

Graph Bisection ... 2

- Compute the eigenvector for the second smallest eigenvalue, e_2
- Use the signs of the vector to give a bisection
- Can be used to get bisections with $n/2$ vertices – by using the median value in e_2
- Can be used to get k partitions

Graph Bisection ... 2

- Compute the eigenvector for the second smallest eigenvalue, e_2
- Use the signs of the vector to give a bisection
- Can be used to get bisections with $n/2$ vertices – by using the median value in e_2
- Can be used to get k partitions by performing bisections recursively or by using more eigenvectors

Spectral Clustering

- Let A be the adjacency matrix and $M =$ diagonal matrix of degrees

Spectral Clustering

- Let A be the adjacency matrix and M = diagonal matrix of degrees
- Construct the Laplacian (PSD)

Spectral Clustering

- Let A be the adjacency matrix and $M =$ diagonal matrix of degrees
- Construct the Laplacian (PSD)
  - Unnormalized: $\mathcal{L} = M - A$

Spectral Clustering

- Let A be the adjacency matrix and $M =$ diagonal matrix of degrees
- Construct the Laplacian (PSD)
  - Unnormalized: $\mathcal{L} = M - A$
  - Normalized, symmetric: $\mathcal{L} = D^{-1/2} L D^{1/2}$

Spectral Clustering

- Let A be the adjacency matrix and M = diagonal matrix of degrees
- Construct the Laplacian (PSD)
  - Unnormalized: $\mathcal{L} = M - A$
  - Normalized, symmetric: $\mathcal{L} = D^{-1/2} L D^{1/2}$
  - Random Walk: $\mathcal{L} = D^{-1} L$

Spectral Clustering

- Let A be the adjacency matrix and $M =$ diagonal matrix of degrees
- Construct the Laplacian (PSD)
  - Unnormalized: $\mathcal{L} = M - A$
  - Normalized, symmetric: $\mathcal{L} = D^{-1/2} L D^{1/2}$
  - Random Walk: $\mathcal{L} = D^{-1} L$
- Define $\mathcal{L}_k$ as the matrix with first k eigenvectors as its columns

Spectral Clustering

- Let A be the adjacency matrix and $M =$ diagonal matrix of degrees
- Construct the Laplacian (PSD)
  - Unnormalized: $\mathcal{L} = M - A$
  - Normalized, symmetric: $\mathcal{L} = D^{-1/2} L D^{1/2}$
  - Random Walk: $\mathcal{L} = D^{-1} L$
- Define $\mathcal{L}_k$ as the matrix with first k eigenvectors as its columns
- Cluster rows of L_k

Spectral Clustering

- Let A be the adjacency matrix and $M =$ diagonal matrix of degrees
- Construct the Laplacian (PSD)
  - Unnormalized: $\mathcal{L} = M - A$
  - Normalized, symmetric: $\mathcal{L} = D^{-1/2} L D^{1/2}$
  - Random Walk: $\mathcal{L} = D^{-1} L$
- Define $\mathcal{L}_k$ as the matrix with first k eigenvectors as its columns
- Cluster rows of L_k