Clustering

Mixture Model Clustering
Spectral Clustering

Huiping Cao

Different types of clusters

- Prototype-based
  - K-Means
  - **Mixture Model Clustering**
- Graph-based
  - Agglomerative hierarchical clustering
  - Divisive hierarchical clustering: Minimum spanning tree clustering
  - **Spectral Clustering**

Clustering Using Mixture Models

- Idea is to model the set of data points as arising from a mixture of distributions
  - Typically, normal (Gaussian) distribution is used
  - But other distributions have been very profitably used
- Generative process
  - Repeat m times
    - From given several distributions, randomly select a distribution
    - Generate an object from it

Mixture Models

- K distributions and m objects $X = \{x_1, \cdots, x_m\}$
- Set of all parameters: $\Theta = \{\theta_1, \cdots, \theta_K\}$
- $p(x_i|\theta_j)$: probability of the ith object if it comes from the jth distribution
- w_j: the probability that the jth distribution is chosen, $\sum_{j=1}^{K} w_j = 1$
- The probability of an object x is given by:

$$p(x|\Theta) = \sum_{j=1}^{K} w_j p_j(x|\theta_j)$$

- If objects are generated independently, the probability of the set of objects

$$p(X|\Theta) = \prod_{i=1}^{m} p(x_i|\Theta) = \prod_{i=1}^{m} \sum_{j=1}^{K} w_j p_j(x_i|\theta_j)$$

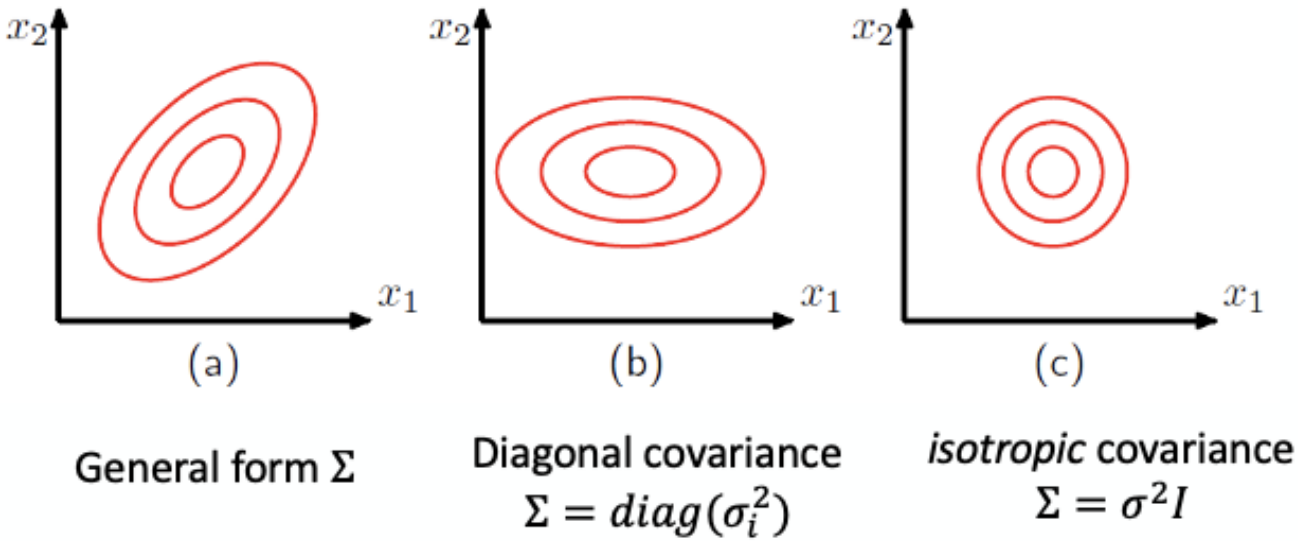
Clustering Using Mixture Models

- Each distribution describes a different cluster
- Clusters are found by estimating the parameters of the statistical distributions
- These parameters describe the distributions (clusters)
- Can identify which objects belong to which cluster (the probabilities)

Multivariate Normal Distribution

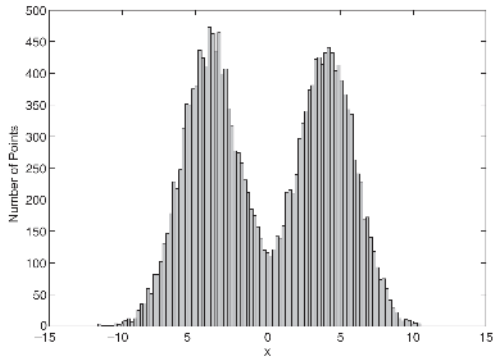
$$X \sim N(\mu, \Sigma)$$

$$f_x(x_1, \cdots, x_k) = \frac{exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))}{\sqrt{(2\pi)^k |\Sigma|}}$$



Probabilistic Clustering: Example

- Informal example: consider modeling the points that generate the following histogram.



- Looks like a combination of two normal (Gaussian) distributions

- Suppose we can estimate the mean and standard deviation of each normal distribution.
  - This completely describes the two clusters
  - We can compute the probabilities with which each point belongs to each cluster
  - Can assign each point to the cluster (distribution) for which it is most probable. $prob(x_i|\Theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

Probabilistic Clustering: Expectation-Maximization Algorithm (EM Algorithm)

- Initialize the parameters
- **Repeat**
  - For each point, compute its probability under each distribution
  - Using these probabilities, find the new estimates of parameters of each distribution that maximize the expected likelihood
- **Until** there is no change

Probabilistic Clustering: Updating Parameters (S.S.)

- Update formula for means assuming an estimate for statistical parameters
- m is the total number of points in the dataset, $\mathbf{x}_i$ is a data point, C_j is a cluster, and $\mathbf{c}_j$ is the centroid of cluster C_j.
  - mean update

$$\mathbf{c}_j = \frac{1}{\sum_{i=1}^{m} p(C_j|\mathbf{x}_i)} \sum_{i=1}^{m} \mathbf{x}_i p(C_j|\mathbf{x}_i)$$

  - covariance update

$$\Sigma_i = \frac{1}{\sum_{i=1}^{m} p(C_j|\mathbf{x}_i)} \sum_{i=1}^{m} (\mathbf{x}_i - \mathbf{c}_j)(\mathbf{x}_i - \mathbf{c}_j)^T p(C_j|\mathbf{x}_i)$$

  -

$$w_j = \frac{\sum_{i=1}^{m} p(C_j|\mathbf{x}_i)}{m}$$

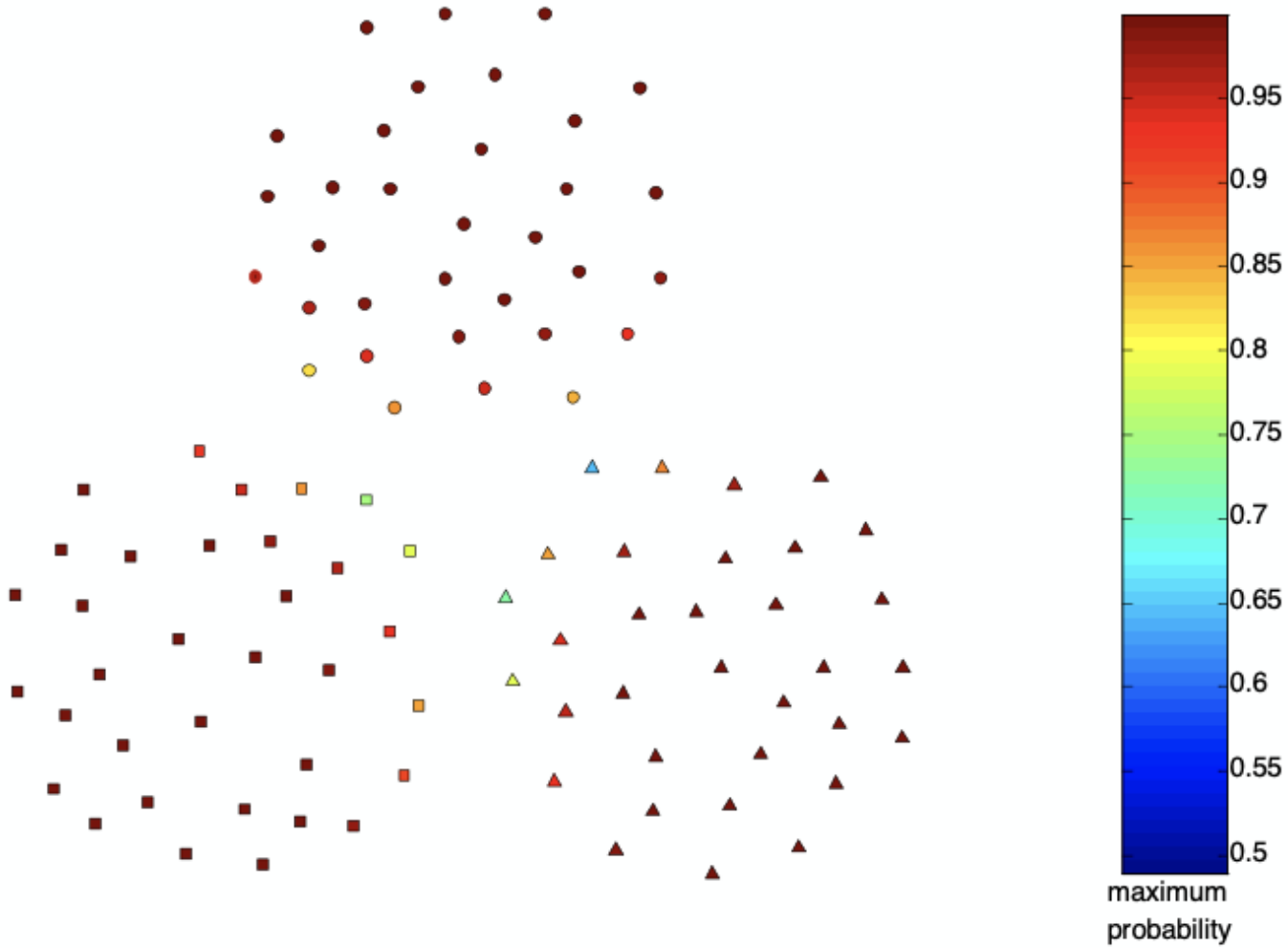
  - Note: detailed derivation can be found from Chapter 9, Pattern Recognition and Machine Learning, by Christopher M. Bishop.

More Detailed EM Algorithm

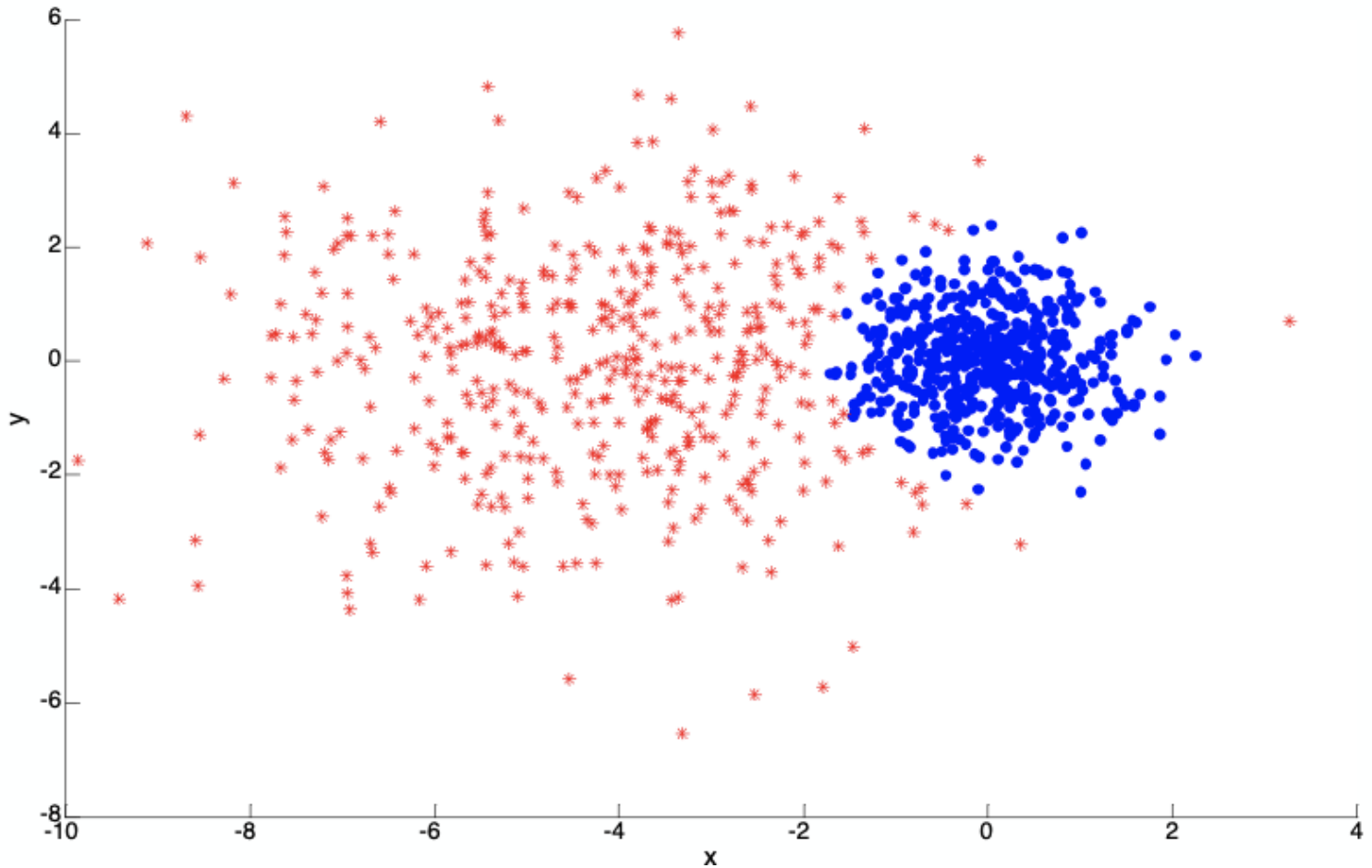
Algorithm 9.2 EM algorithm.

1: Select an initial set of model parameters.
(As with K-means, this can be done randomly or in a variety of ways.)
2: **repeat**
3: **Expectation Step** For each object, calculate the probability that each object belongs to each distribution, i.e., calculate $prob(distribution\ j|\mathbf{x}_i, \Theta)$.
4: **Maximization Step** Given the probabilities from the expectation step, find the new estimates of the parameters that maximize the expected likelihood.
5: **until** The parameters do not change.
(Alternatively, stop if the change in the parameters is below a specified threshold.)

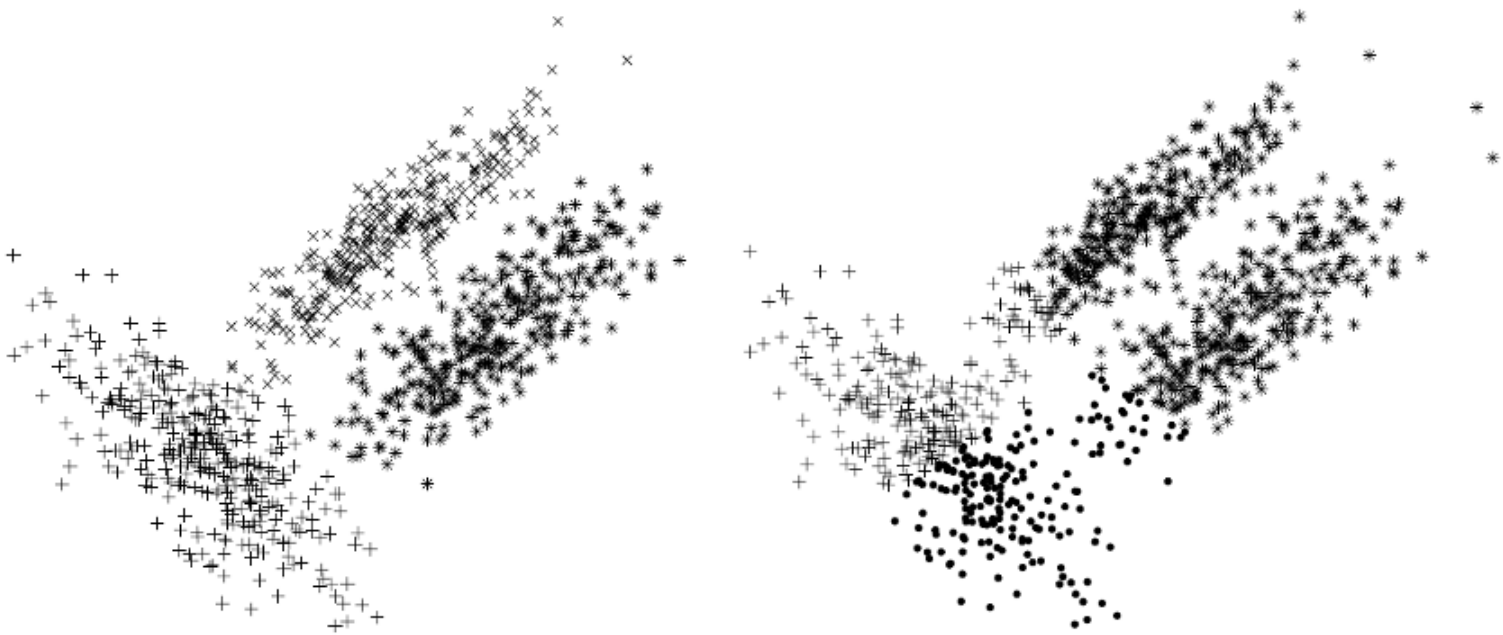
Probabilistic Clustering Applied to Sample Data



Probabilistic Clustering: Dense and Sparse Clusters



Mixture model clustering vs. K-means



Mixture Model Clustering

K-means Clustering

Problems with EM

- Convergence can be slow
- Only guarantees finding local maxima
- Makes some significant statistical assumptions
- Number of parameters for Gaussian distribution grows as $O(d^2)$, d the number of dimensions
  - Parameters associated with covariance matrix
  - K-means only estimates cluster means, which grow as $O(d)$

Alternatives to EM

- Method of moments / Spectral methods
  - ICML 2014 workshop bibliography
    `https://sites.google.com/site/momentsicml2014/bibliography`
- Markov chain Monte Carlo (MCMC)
- Other approaches

Graph-Based Clustering

- Graph-Based clustering uses the proximity graph
  - Start with the proximity matrix
  - Consider each point as a node in a graph
  - Each edge between two nodes has a weight which is the proximity between the two points
  - Initially the proximity graph is fully connected
  - MIN (single-link) and MAX (complete-link) can be viewed as starting with this graph
- In the simplest case, clusters are connected components in the graph.

Graph-Based Clustering: Sparsification

- The amount of data that needs to be processed is drastically reduced
  - Sparsification can eliminate more than 99% of the entries in a proximity matrix
  - The amount of time required to cluster the data is drastically reduced
  - The size of the problems that can be handled is increased

Graph-Based Clustering: Sparsification ...

- Clustering may work better
  - Sparsification techniques keep the connections to the most similar (nearest) neighbors of a point while breaking the connections to less similar points.
  - The nearest neighbors of a point tend to belong to the same class as the point itself.
  - This reduces the impact of noise and outliers and sharpens the distinction between clusters.
- Sparsification facilitates the use of graph partitioning algorithms (or algorithms based on graph partitioning algorithms)
  - Chameleon, spectral clustering

Sparsification in the Clustering Process

Spectral Clustering

- Use the graphs spectrum: **eigenvalues** and **eigenvectors** to identify the clusters.

Eigenvalues and Eigenvectors

- The eigenvalues and eigenvectors of an n by n matrix $\mathbf{A}$ are, respectively, the scalar values λ and the vectors $\mathbf{u}$ that are solutions to the following equation.

$$\mathbf{Au} = \lambda\mathbf{u}$$

- In other words, eigenvectors are the vectors that are unchanged, except for magnitude, when multiplied by $\mathbf{A}$.
- Eigen decomposition

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$$

$\mathbf{A}$: an $n \times n$ matrix. $\mathbf{Q}$: a square $n \times n$ matrix whose ith column is the eigenvector q_i of $\mathbf{A}$. $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\Lambda_{ii} = \lambda_i$.

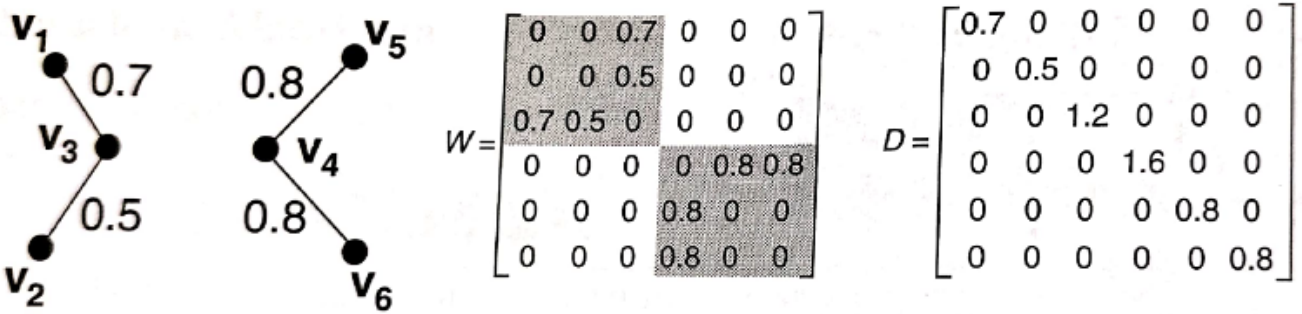
Example

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -2 & -3 \end{pmatrix}$$

$$\lambda_1 = -1, \lambda_2 = -2$$

$$\mathbf{u}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \mathbf{u}_2 = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

Similarity Graph with two connected components



- **W**: weighted adjacency matrix
- **D**:

$$D_{ij} = \begin{cases} \sum_k w_{ik} & \textit{if } \text{i=j} \\ 0 & \textit{otherwise} \end{cases}$$

Block Structure

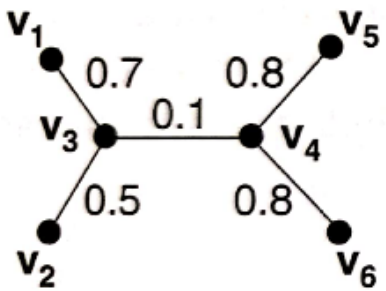
- Order rows and columns of $\mathbf{W}$ in such a way that nodes belonging to the same connected component are next to each other.

$$\mathbf{A} = \begin{pmatrix} W_1 & \mathbf{0} \\ \mathbf{0} & W_2 \end{pmatrix}$$

- k connected components:

$$\mathbf{A} = \begin{pmatrix} W_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & W_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & W_k \end{pmatrix}$$

Similarity Graph with one connected component



$$W = \begin{bmatrix} 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.7 & 0.5 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0.8 & 0.8 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix}$$

- **W**: weighted adjacency matrix
- **D**:

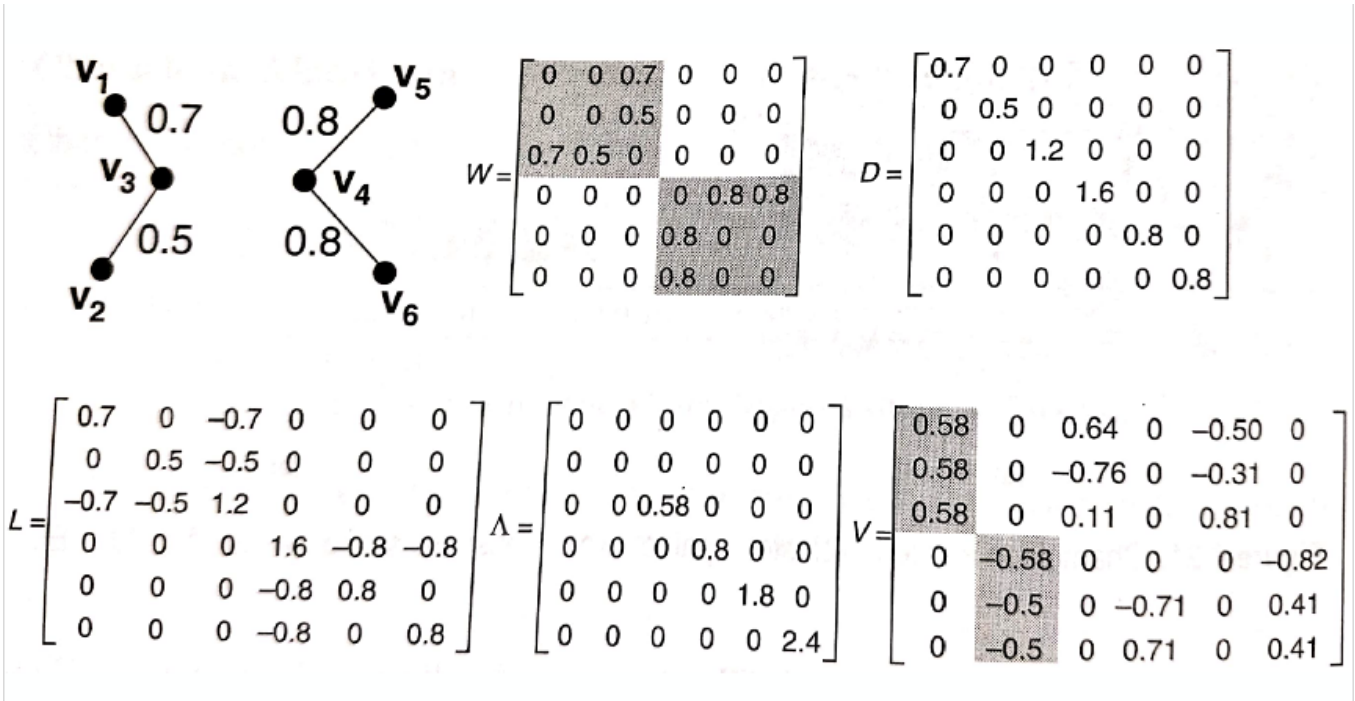
$$D_{ij} = \begin{cases} \sum_k w_{ik} & \text{if i=j} \\ 0 & \text{otherwise} \end{cases}$$

The Graph Laplacian Matrix

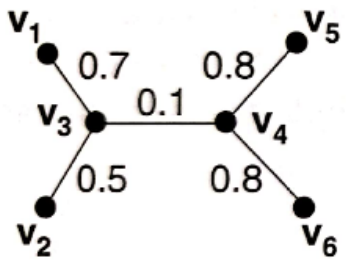
$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

- Symmetric
- All eigenvalues of $\mathbf{L}$ are non-negative

Similarity Graph with two connected components



Similarity Graph with one connected components



$$W = \begin{bmatrix} 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.7 & 0.5 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0.8 & 0.8 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix}$$

$$L = \begin{bmatrix} 0.7 & 0 & -0.7 & 0 & 0 & 0 \\ 0 & 0.5 & -0.5 & 0 & 0 & 0 \\ -0.7 & -0.5 & 1.3 & -0.1 & 0 & 0 \\ 0 & 0 & -0.1 & 1.7 & -0.8 & -0.8 \\ 0 & 0 & 0 & -0.8 & 0.8 & 0 \\ 0 & 0 & 0 & -0.8 & 0 & 0.8 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.06 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.58 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.80 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.88 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2.47 \end{bmatrix} \quad V = \begin{bmatrix} 0.41 & -0.41 & 0.65 & 0 & 0.48 & -0.04 \\ 0.41 & -0.43 & -0.75 & 0 & 0.29 & 0.03 \\ 0.41 & -0.38 & 0.11 & 0 & -0.81 & 0.10 \\ 0.41 & 0.38 & 0 & 0 & 0.08 & -0.82 \\ 0.41 & 0.42 & 0 & -0.71 & 0.06 & 0.39 \\ 0.41 & 0.42 & 0 & 0.71 & 0.06 & 0.39 \end{bmatrix}$$

The Graph Laplacian Matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

- Symmetric
- All eigenvalues of $\mathbf{L}$ are non-negative
- The smallest eigenvalue of $\mathbf{L}$ is zero, with the corresponding eigenvector $\mathbf{e}$ (a vector of 1s)

$$\mathbf{We} = \mathbf{De} \leftrightarrow (\mathbf{D} - \mathbf{W})\mathbf{e} = \mathbf{0} \leftrightarrow \mathbf{Le} = \mathbf{0e}$$

The Graph Laplacian Matrix

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

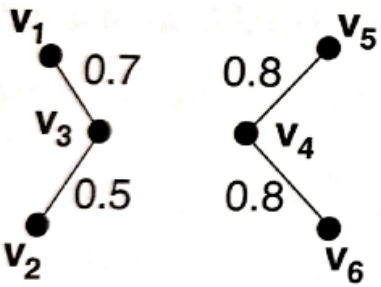
- For a graph with k connected components, $\mathbf{L}$ also has a block structure

$$\mathbf{L} = \begin{pmatrix} L_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & L_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & L_k \end{pmatrix}$$

- In addition, $\mathbf{L}$ has k eigenvalues of zeros, with the corresponding eigenvectors

$$\begin{pmatrix} \mathbf{e}_1 \\ \mathbf{0} \\ \cdots \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} \\ \mathbf{e}_2 \\ \cdots \\ \mathbf{0} \end{pmatrix}, \cdots, \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \cdots \\ \mathbf{e}_k \end{pmatrix},$$

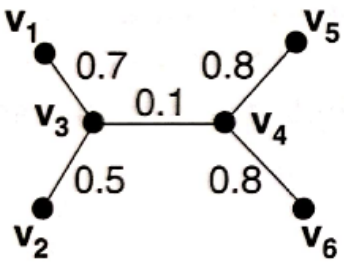
Similarity Graph with two connected components



$$W = \begin{bmatrix} 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.7 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0.8 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix}$$

$$V = \begin{matrix} V_1 \rightarrow \\ V_2 \rightarrow \\ V_3 \rightarrow \\ V_4 \rightarrow \\ V_5 \rightarrow \\ V_6 \rightarrow \end{matrix} \begin{bmatrix} 0.58 & 0 \\ 0.58 & 0 \\ 0.58 & 0 \\ 0 & -0.58 \\ 0 & -0.5 \\ 0 & -0.5 \end{bmatrix}$$

Similarity Graph with one connected components



$$W = \begin{bmatrix} 0 & 0 & 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 & 0 \\ 0.7 & 0.5 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0.8 & 0.8 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.8 \end{bmatrix}$$

$$V' = \begin{matrix} V_1 \rightarrow \\ V_2 \rightarrow \\ V_3 \rightarrow \\ V_4 \rightarrow \\ V_5 \rightarrow \\ V_6 \rightarrow \end{matrix} \begin{bmatrix} 0.41 & -0.41 \\ 0.41 & -0.43 \\ 0.41 & -0.38 \\ 0.41 & 0.38 \\ 0.41 & 0.42 \\ 0.41 & 0.42 \end{bmatrix}$$

Spectral Clustering Algorithm

- Create a sparsified similarity graph $\mathbf{W}$.
- Compute the graph Laplacian for $\mathbf{W}$, $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- Create a matrix $\mathbf{V}$ from the first k eigenvectors of $\mathbf{L}$
- Apply K-means clustering on $\mathbf{V}$ to obtain the k clusters

References

- Chapter 8: Introduction to Data Mining (2nd Edition) by Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar
- Chapter 9, Pattern Recognition and Machine Learning, by Christopher M. Bishop
- Appendices, Introduction to Data Mining (2nd Edition) by Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar
  `https://www-users.cs.umn.edu/~kumar001/dmbook/appendices_2ed.pdf`
- Gaussian mixture: `https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html`
- Spectral clustering: `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html`