CS5489
Lecture 6.2: Clustering

Kede Ma

City University of Hong Kong (Dongguan)

Slide template by courtesy of Benjamin M. Marlin

Outline

1 Introduction

2 Exhaustive Clustering

3 K-Means Clustering

4 Gaussian Mixture Models

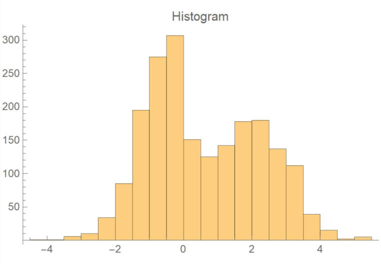
Supervised vs Unsupervised Learning

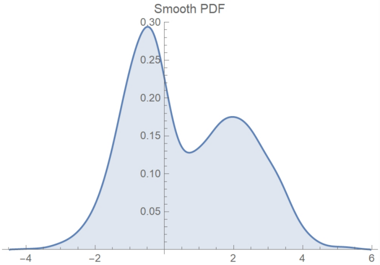
- Supervised learning considers input-output pairs $(\mathbf{x}, y)$
  - Learn a mapping f from input to output
  - Classification: output $y \in \{-1, 1\}$
  - Regression: output $y \in \mathbb{R}$
  - “Supervised” here means that the algorithm is learning the mapping that we want

- Unsupervised learning only considers the input data $\mathbf{x}$
  - There is no output value
  - **Goal**: try to discover inherent properties in the data
    - Density estimation
    - Clustering
    - Dimensionality reduction
    - Manifold embedding

Density Estimation

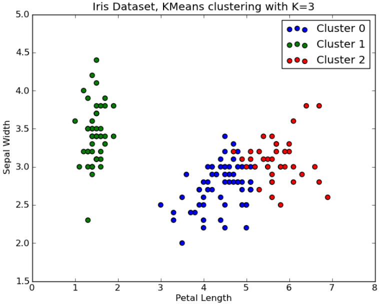
- From $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$, estimate a probability distribution $p(\mathbf{x})$
    - Mother of all unsupervised learning problems
    - Key technique underpinning AIGC
    - Can be conditional, i.e., to estimate $p(\mathbf{x}|y)$. (Here, we don't learn a mapping to predict y from $\mathbf{x}$, we just use y as conditioning)





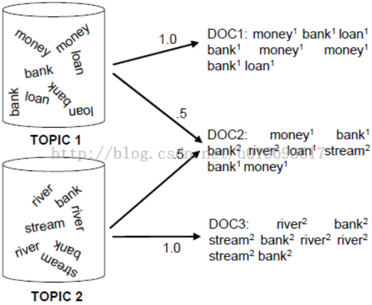
Clustering

- Find clusters of similar items in the data
- Find a representative item that “summarizes” all items in the cluster
- For example: group iris flowers by their measurements (sepal width and petal length)



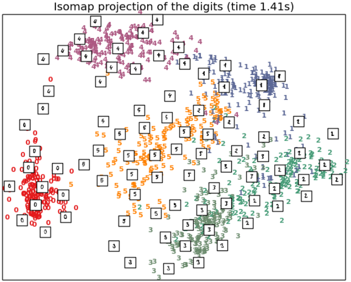
Dimensionality Reduction

- Transform high-dimensional vectors into low-dimensional vectors
  - Dimensions in the low-dim data may have semantic meaning
- For example: document analysis
  - High-dim: bag-of-words vectors of documents
  - Low-dim: each dimension represents similarity to a topic


money
money
loan
bank
bank
loan
bank
loan
TOPIC 1
1.0
.5
DOC1: money^1 bank^1 loan^1 bank^1 money^1 money^1 bank^1 loan^1
DOC2: money^1 bank^1 bank^2 river^2 loan^1 stream^2 bank^1 money^1
.5
river
bank
river
stream
river
bank
stream
TOPIC 2
1.0
DOC3: river^2 bank^2 stream^2 bank^2 river^2 river^2 stream^2 bank^2

Manifold Embedding

- Project high-dimensional vectors into 2- or 3-dimensional space for visualization
  - Points in the low-dim space have similar pair-wise distances as in the high-dim space
- For example: visualize a collection of hand-written digits (images)



Outline

1 Introduction

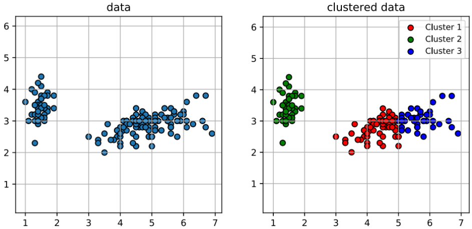
2 Exhaustive Clustering

3 K-Means Clustering

4 Gaussian Mixture Models

Defining a Clustering

- Suppose we have M data points $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^N$
- A clustering of the M points into K clusters is a partitioning of $\mathcal{D}$ into K mutually disjoint groups $\mathcal{C} = \{C_1, \ldots, C_K\}$ such that $C_1 \cup \ldots \cup C_K = \mathcal{D}$
  - Groups are also called clusters
  - K is the number of clusters
  - Each data point is assigned with a cluster index ($y \in \{1, \ldots, K\}$)



Exhaustive Clustering

- Suppose we have a function $f(\mathcal{C})$ that takes a clustering $\mathcal{C}$ of the data set $\mathcal{D}$ as input, and returns a score with lower scores indicating better clustering
- The optimal clustering according to f is simply given by

$$\arg\min_{\mathcal{C}} \; f(\mathcal{C})$$

- **Question:** What is the complexity of exhaustive clustering?

Number of Clusterings

- The total number of clusterings of a data set with M elements is the Bell number B_M, where $B_0 = 1$ and $B_{M+1} = \sum_{k=0}^{M} \binom{M}{k} B_k$
- The first few Bell numbers are: 1, 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975, 678570, 4213597, 27644437, 190899322, ...
- The complexity of exhaustive clustering scales with B_M and is thus computationally totally intractable for general scoring functions
- We will need either approximation algorithms or scoring functions with special properties

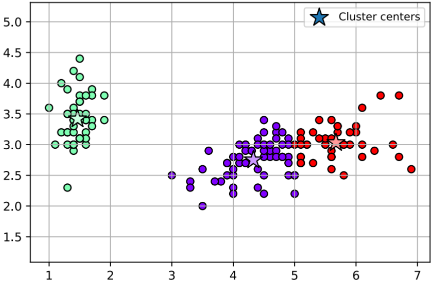
`https://oeis.org/A000110`

Outline

1. Introduction
2. Exhaustive Clustering
3. K-Means Clustering
4. Gaussian Mixture Models

K-Means Clustering

- **Idea**:
  - Assume K clusters
  - Each cluster is represented by a **cluster center**
    - $\mathbf{c}_j \in \mathbb{R}^N, j \in \{1, \ldots, K\}$
  - Assign each data point to the closest cluster center
    - According to Euclidean distance $\|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2$



K-Means Clustering Problem

- How to pick the cluster centers?
  - Assume there are K clusters
  - Pick the cluster centers that minimize the squared distance to all its cluster members

$$\min_{\mathbf{c}_1,\ldots,\mathbf{c}_K} \sum_{i=1}^{M} \|\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}\|_2^2,$$

  where $z^{(i)}$ is the index of the closest cluster center to $\mathbf{x}^{(i)}$
    - $z^{(i)} = \text{argmin}_{j=\{1,\ldots,K\}} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2$
    - I.e., the assignment of point to its closest cluster
- Solution:
  - If the assignments $\{z^{(i)}\}$ are known...
    - Let C_j be the set of points assigned to Cluster j: $C_j = \{\mathbf{x}^{(i)}|z^{(i)} = j\}$
    - Cluster center is the mean of the points in that cluster: $\mathbf{c}_j = \frac{1}{|C_j|}\sum_{\mathbf{x}^{(i)}\in C_j} \mathbf{x}^{(i)}$

Chicken and Egg Problem

- Cluster assignment of each point depends on the cluster centers
- Location of cluster center depends on which points are assigned to it
- **Question:** How to resolve this issue?

K-Means Algorithm

- Pick initial cluster centers
- Repeat:
  1. **Assignment step**: calculate assignment $z^{(i)}$ for each point $\mathbf{x}^{(i)}$: closest cluster center using Euclidean distance

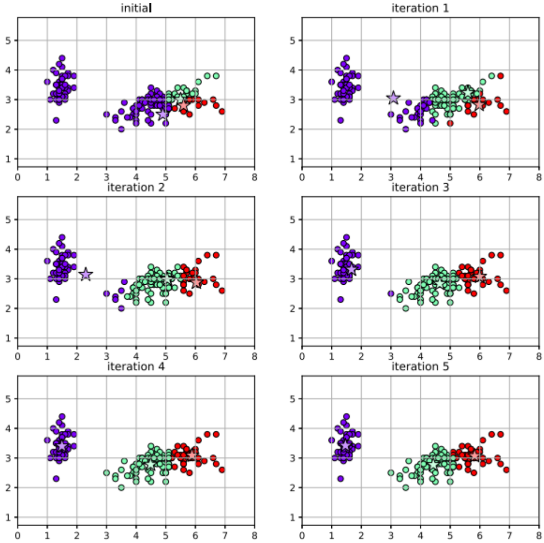
$$z^{(i)} = \underset{j=\{1,\ldots,K\}}{\arg\min} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2$$

  2. **Update step**: Calculate cluster center as average of points assigned to Cluster j

$$\mathbf{c}_j = \frac{\sum_{i=1}^{M} \mathbb{I}[z^{(i)} = j]\mathbf{x}^{(i)}}{\sum_{i=1}^{M} \mathbb{I}[z^{(i)} = j]}$$

- This procedure will converge eventually

Example: Iris Dataset



The K-Means Objective

- K-means attempts to minimize the sum of within-cluster variation over all clusters (also called the within-cluster sum of squares):

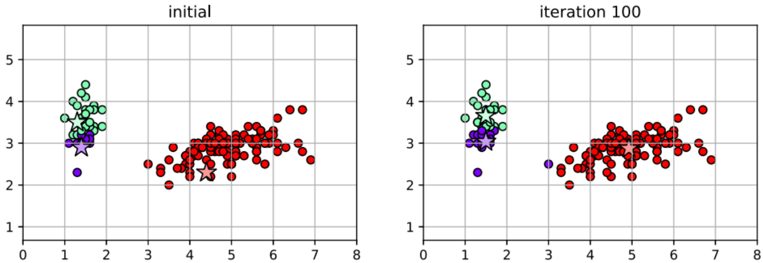
$$\min \ell(\mathbf{z}, \{\mathbf{c}_j\}_{j=1}^K) = \min_{\mathbf{z}, \{\mathbf{c}_j\}_{j=1}^K} \sum_{i=1}^{M} \|\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}\|_2^2,$$

where $\mathbf{z} = [z^{(1)}, z^{(2)}, \ldots, z^{(M)}]^T$

- K-means is exactly **coordinate descent** on ℓ, where Assignment step minimizes ℓ w.r.t. $\mathbf{z}$ while holding $\{\mathbf{c}_j\}$ fixed, and Update step minimizes ℓ w.r.t. $\{\mathbf{c}_j\}$ while holding $\mathbf{z}$ fixed
- Thus, ℓ is monotonically decreasing. As ℓ is also lower bounded by 0, the value of ℓ must converge

Important Note

- Note that K-means has many local optima in general, each corresponding to a different clustering of the data. Finding the global optimum is not computationally tractable
- Thus, the final results can be highly sensitive to initialization
  - Some bad initial cluster centers will yield poor clustering results!

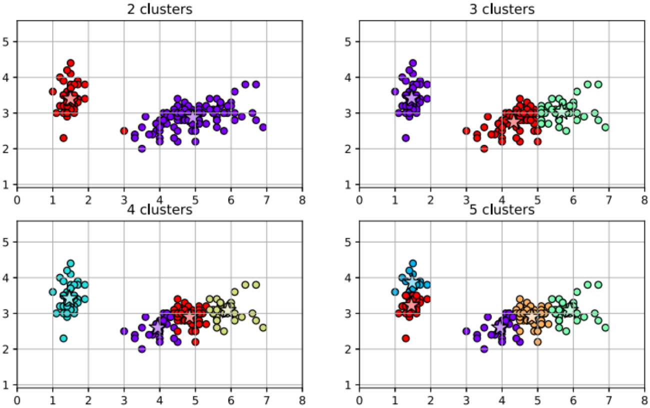


Solution to Initialization

- It is common to perform multiple random re-starts of the algorithm, and take the clustering with the best result
- Common initializations include 1) setting the initial centers to be randomly selected data points, 2) setting the initial partition to a random partition, and 3) selecting centers using a "furthest first"-style heuristic (more formally known as K-means++)
- It often helps to initially to run with $K\log(K)$ clusters, then merge clusters to get down to K and run the algorithm from that initialization

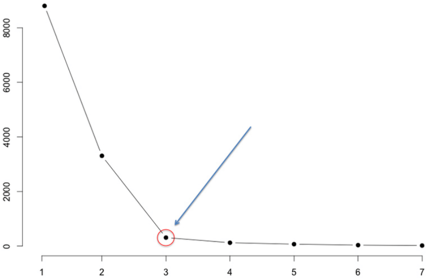
For Different *K*

- We need to choose the appropriate *K*



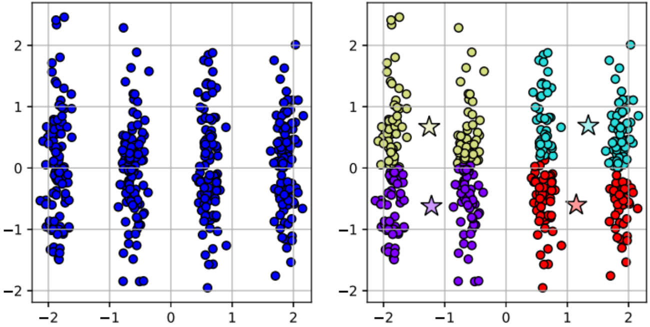
Choosing K

- Clustering results depend on the number of clusters K used
- We don't typically know this information beforehand
- The elbow method
    - Simple, only requires one fit per value of K



Circular Clusters

- One problem with K-means is that it assumes that each cluster has a circular shape
    - Based on Euclidean distance to each center
    - K-means cannot handle skewed (elliptical) clusters



Outline

1. Introduction
2. Exhaustive Clustering
3. K-Means Clustering
4. Gaussian Mixture Models

Gaussian Mixture Model (GMM)

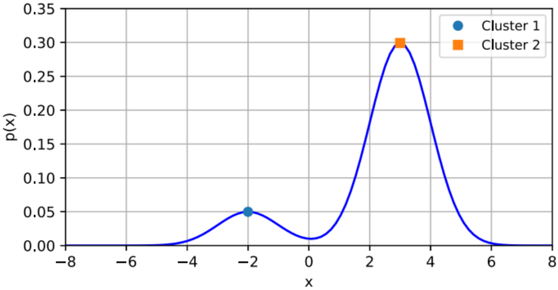
- A multivariate Gaussian can model a cluster with an elliptical shape
  - The ellipse shape is controlled by the covariance matrix of the Gaussian
  - The location of the cluster is controlled by the mean
- Gaussian mixture model is a weighted sum of Gaussians

$$p(\mathbf{x}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

  - Each Gaussian represents one elliptical cluster
    - $\boldsymbol{\mu}_j$ is mean of the j-th Gaussian (the location)
    - $\boldsymbol{\Sigma}_j$ is covariance matrix of the j-th Gaussian (the ellipse shape)
    - π_j is prior weight of the j-th Gaussian (how likely is this cluster)

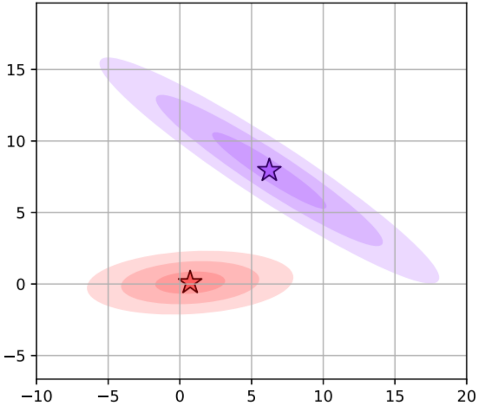
1-D Example of GMM

- Each Gaussian is a “mountain”



2-D Example of GMM

- Each Gaussian defines a “mountain”
  - Contours are ellipses



Clustering with GMMs

- Given a data set $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$, learn a GMM using maximum likelihood estimation:

$$\max_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}} \sum_{i=1}^{M} \log \sum_{j=1}^{K} \pi_j N(\mathbf{x}^{(i)}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

- While we can do this directly using gradient-based optimization, it's often faster to use a special algorithm called **Expectation Maximization**

Expectation Maximization (EM) for GMM

- EM results in an algorithm similar to *K*-means
  - **E-Step**: Calculate cluster membership with "soft" assignment - a data point can have a fractional contribution to different clusters
    - Contribution of Point *i* to Cluster *j* is defined by the posterior probability that $\mathbf{x}^{(i)}$ belongs to Cluster *j* using the Bayes' rule

$$
\begin{aligned}
z_j^{(i)} = p(z^{(i)} = j | \mathbf{x}^{(i)}) &= \frac{p(\mathbf{x}^{(i)}, z^{(i)} = j)}{p(\mathbf{x}^{(i)})} \\
&= \frac{p(\mathbf{x}^{(i)} | z^{(i)} = j) p(z^{(i)} = j)}{\sum_{k=1}^{K} p(\mathbf{x}^{(i)} | z^{(i)} = k) p(z^{(i)} = k)} \\
&= \frac{\pi_j \mathcal{N}(\mathbf{x}^{(i)}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(i)}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}
\end{aligned}
$$

Expectation Maximization (EM) for GMM

- EM results in an algorithm similar to *K*-means
  - **M-Step**: Update each Gaussian cluster (mean, covariance, and weight) using “soft” weighting
    - “Soft” count of points in Cluster *j*:

$$M_j = \sum_{i=1}^{M} z_j^{(i)}$$

    - Weight:

$$\pi_j = \frac{M_j}{M}$$

    - Mean:

$$\boldsymbol{\mu}_j = \frac{1}{M_j} \sum_{i=1}^{M} z_j^{(i)} \mathbf{x}^{(i)}$$

    - Covariance:

$$\boldsymbol{\Sigma}_j = \frac{1}{M_j} \sum_{i=1}^{M} z_j^{(i)} (\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)(\mathbf{x}^{(i)} - \boldsymbol{\mu}_j)^T$$

GMM: A Special Case

- Suppose we fix $\pi_j = 1/K$ and $\mathbf{\Sigma}_j = \mathbf{I}$. In this case we have

$$p(\mathbf{x}^{(i)}|z=j) = \mathcal{N}(\mathbf{x}^{(i)}; \boldsymbol{\mu}_j, \mathbf{I}) = \frac{1}{\sqrt{(2\pi)^N}} \exp(-\frac{1}{2}||\mathbf{x}^{(i)} - \boldsymbol{\mu}_j||_2^2)$$

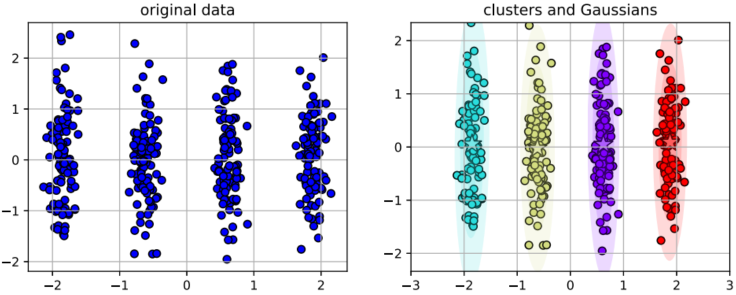
- We obtain a special case of the EM algorithm for GMM:

$$z_j^{(i)} = \frac{\exp(-\frac{1}{2}||\mathbf{x}^{(i)} - \boldsymbol{\mu}_j||_2^2)}{\sum_{k=1}^{K} \exp(-\frac{1}{2}||\mathbf{x}^{(i)} - \boldsymbol{\mu}_k||_2^2)},$$

$$\boldsymbol{\mu}_j = \frac{1}{M_j} \sum_{i=1}^{M} z_j^{(i)} \mathbf{x}^{(i)}$$

- This is often referred to as **soft** K-means

GMM Clustering Example



Covariance Matrix

- The covariance matrix is an $N \times N$ matrix

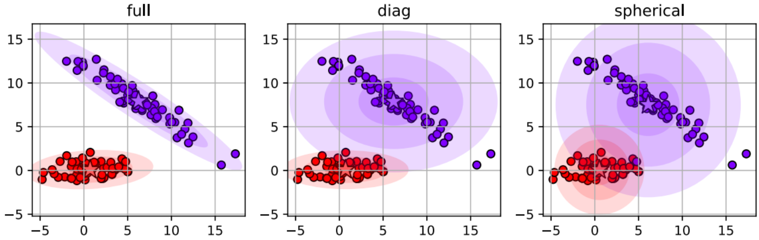
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

- For high-dimensional data, it can be very large
  - Requires a lot of data to learn effectively
- Solution:
  - Use **diagonal** covariance matrices (N parameters) or **spherical** covariance matrices (1 parameter)

$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \qquad \begin{bmatrix} a & 0 & 0 \\ 0 & a & 0 \\ 0 & 0 & a \end{bmatrix}$$

    - Diagonal: axes of ellipses will be aligned with the coordinate axes
    - Spherical: clusters will be circular (similar to K-means)

GMM Clustering Example



Trade-Offs

- The original K-means algorithm performs hard assignments during clustering, and implicitly assumes all clusters will have an equal number of points assigned as well as a unit covariance matrix
- GMM for clustering relaxes all of these assumptions. The objective still has multiple local optima
- EM can also be used with any component densities/distributions to customize the model to a given data set
- As with K-means, initialization is important for GMM