Universität
Konstanz

Lecture 10 Clustering

Dr. Hanhe Lin
Dept. of Computer and Information Science
University of Konstanz

Unsupervised learning

- In unsupervised learning, the training data consists of a set of input vectors without any corresponding target values

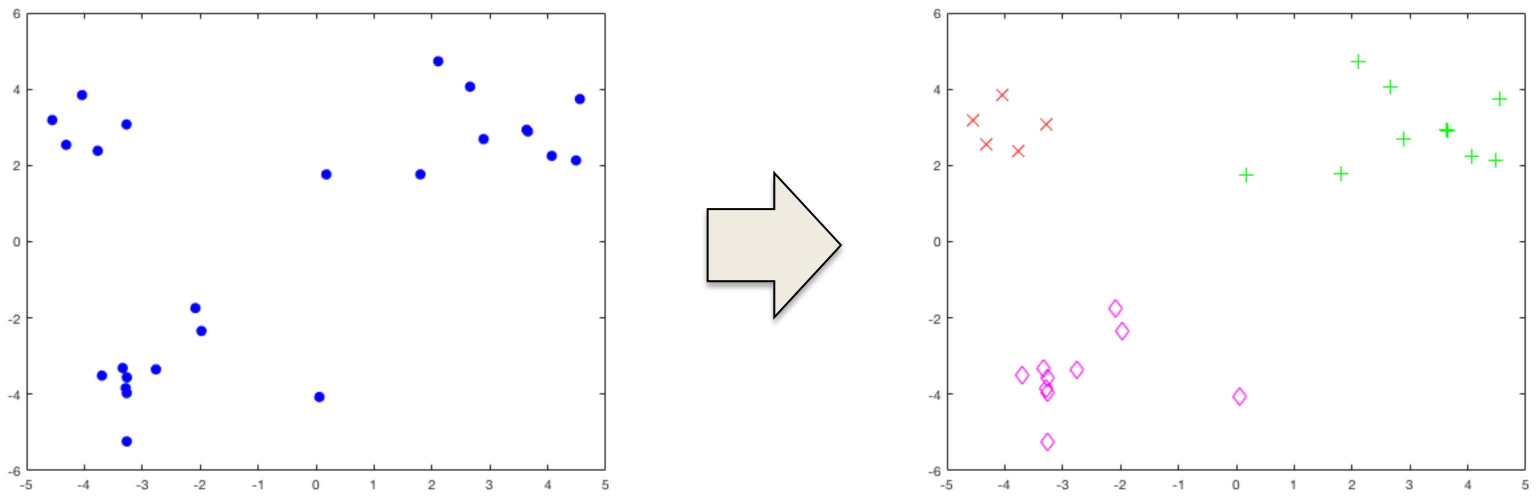
- Goals:
  - Clustering: discover groups of similar examples within the data
  - Density estimation: determine the distribution of data within the input space
  - Data visualization: project the data from a high-dimensional space down to two or three dimensions

Unsupervised learning

- Unsupervised learning was largely ignored by machine learning community
  - It is hard to say what the aim of unsupervised learning is

- Unsupervised learning is the future of machine learning
  - Large scale of unlabeled data
  - Very expensive and infeasible to annotate large data

Clustering - intuition



Clustering

- Goal: partition the data into different groups (clusters), where the data in each group are similar to each other other than to those in other groups

- Clustering is one of the most commonly researched unsupervised learning topic

- The only information clustering uses is the similarity between examples

- Clustering groups examples based of their mutual similarities

- A good cluster algorithm is:
  - High within-cluster similarity
  - Low inter-cluster similarity

Clustering: when and why?

- Useful for:
  - Automatically organizing data
  - Understanding hidden structure in some data
  - Representing high-dimensional data in a low-dimensional space

- Examples:
  - Image segmentation
  - Customers according to purchase histories
  - Genes according to expression profile
  - Search results according to topic
  - Social network analysis

Clustering notation

Our data are

$$D = \left\{ x^{(1)}, \ldots, x^{(m)} \right\}$$

Each data point is n-dimensional, i.e.,

$$x = \left\{ x_1, \ldots, x_n \right\}$$

Define a distance function between data $x^{(i)}$ and $x^{(j)}$, $d(x^{(i)}, x^{(j)})$

Goal: segment the data into k groups

Clustering division

- Hierarchical clustering
  - Agglomerative: bottom-up
  - Divisive: top-down
  - …
- Centroid-based clustering
  - K-means clustering
  - Kernel k-means
  - Fuzz c-means
  - …
- Distribution-based clustering
  - Gaussian mixture model (GMM)
  - …
- Density-based clustering
  - Mean-shift
  - …

Distance measure

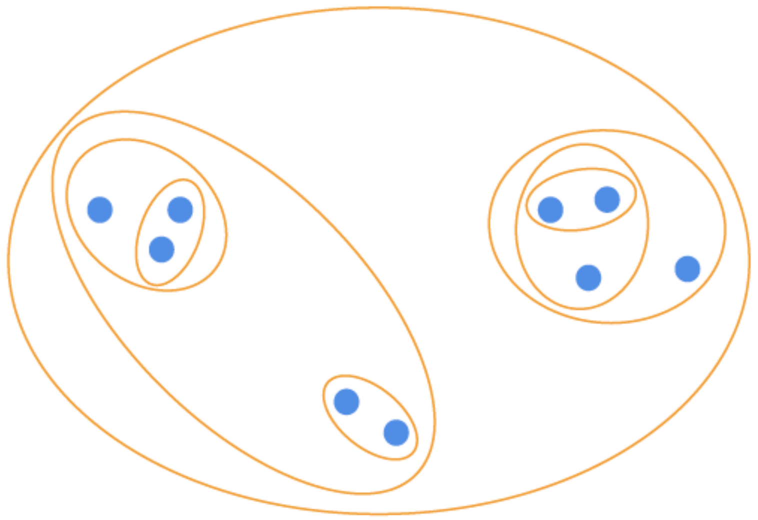
- Distance is inversely related to similarity, namely, large distance, low similarity; small distance, high similarity
- Choice of the distance measure is very important for clustering
- Some commonly used distance metrics:
  - Euclidean distance (L2 norm): $d\left(x^{(i)}, x^{(j)}\right) = \left\|x^{(i)} - x^{(j)}\right\|_2$
  - Manhattan distance (L1 norm): $d\left(x^{(i)}, x^{(j)}\right) = \left\|x^{(i)} - x^{(j)}\right\|_1$
  - Mahalanobis distance: $d\left(x^{(i)}, x^{(j)}\right) = \sqrt{(x^{(i)} - x^{(j)})^T S^{-1} (x^{(i)} - x^{(j)})}$ where S is the covariance matrix
- Similarity is subjective and hard to define
- Different similarity criteria can lead to different clusters

The property of a distance measure

- Non-negative: $d(x^{(i)}, x^{(j)}) \geq 0$
- Symmetry: $d\left(x^{(i)}, x^{(j)}\right) = d(x^{(j)}, x^{(i)})$
- Self-similarity: $d\left(x^{(i)}, x^{(j)}\right) = 0$ if $x^{(i)} = x^{(j)}$, we should not conclude A looks like B, but B does not look like A
- Triangle inequality: $d\left(x^{(i)}, x^{(k)}\right) + d\left(x^{(k)}, x^{(j)}\right) \geq d\left(x^{(i)}, x^{(j)}\right)$

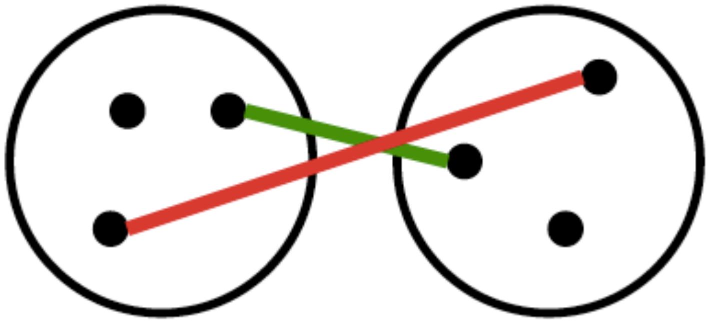
Agglomerative clustering

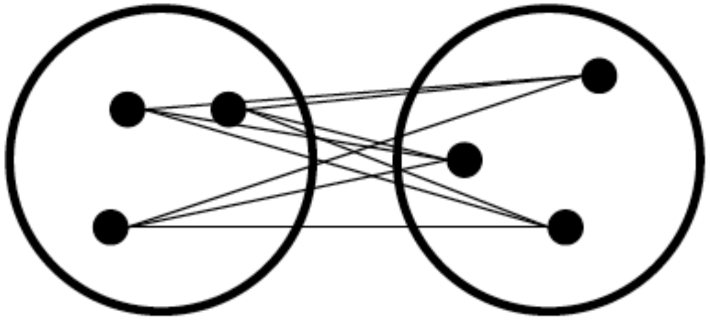
- Initially, each point is treated as a cluster
- Repeat:
  - Compute distance between all clusters in terms of the defined distance measure (store for efficiency)
  - Merge two nearest clusters into a new cluster
  - Stop when there is only on cluster left
- Save both clusters and sequence of cluster operations by "Dendrogram"



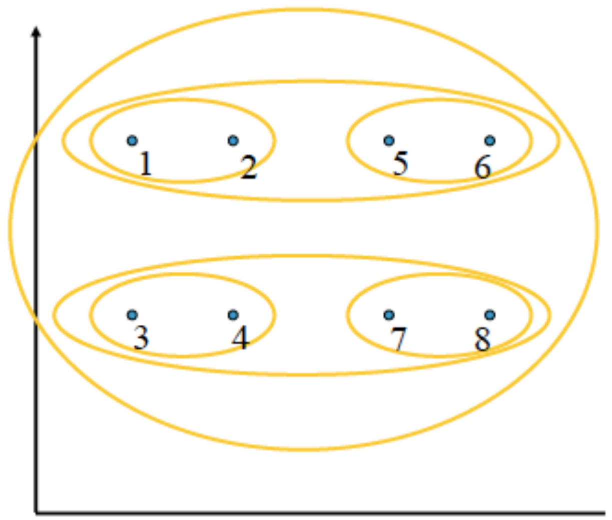
Agglomerative clustering

- How should we define "closest" for clustering with multiple elements?
- Many options:
  - Closest pair (single-link clustering)
  - Farthest pair (complete-link clustering)
  - Average of all pairs
- Different choices create different clustering behaviors

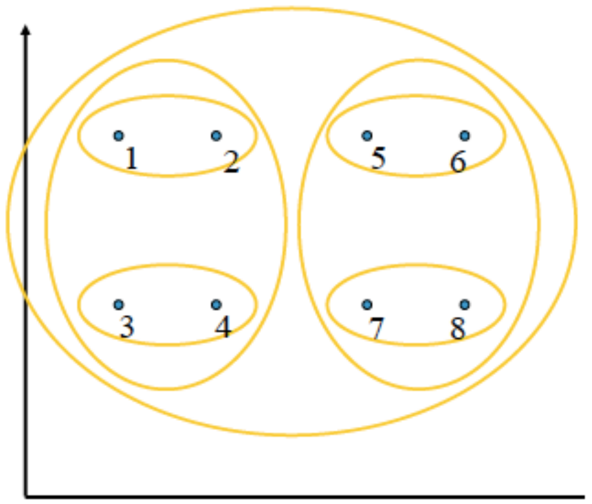




Agglomerative clustering

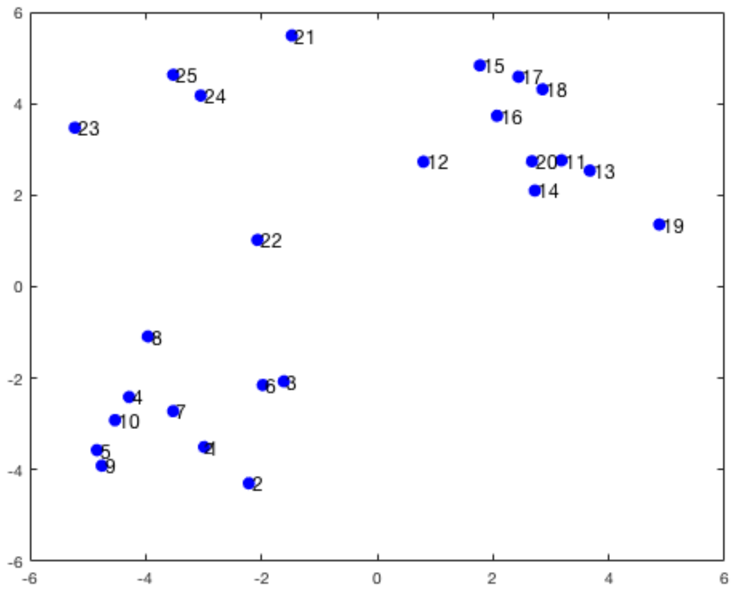


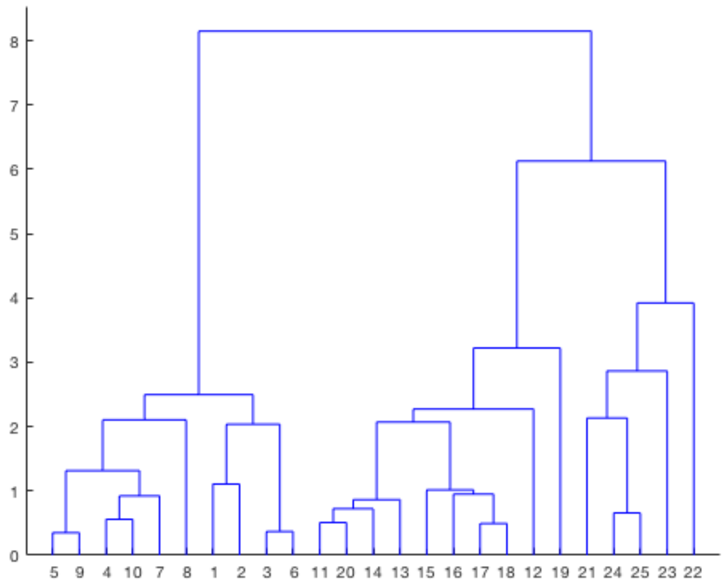
Closest pair



Farthest pair

Example 1: agglomerative clustering

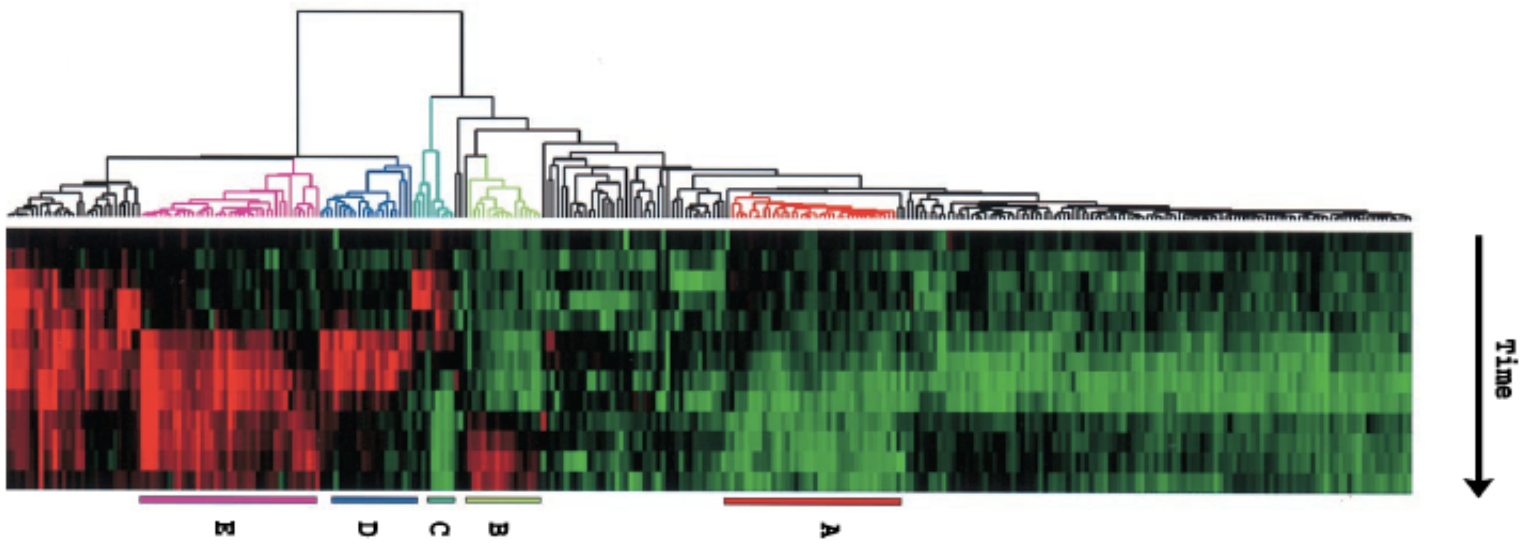




Dendrogram

Example 2: agglomerative clustering

- Clustering gene expression data [Eisen et al 1998].



Overview: agglomerative clustering

- Summary
  - Choose a cluster distance/dissimilarity method
  - Successively merge closet pair of clusters until only one cluster left

- Pros and cons
  - Easy to understand and implement
  - Don’t need to define the number of clusters
  - Possible to view partitions at different levels of granularities
  - Computational extensive, complexity is $O(m^2 \log m)$

k-means clustering

- The basic idea is to describe each cluster by its mean value.
- Note: this works only for distances such that a mean is well-defined.
- The goal of k-means is to assign data to clusters and define these clusters with their means.

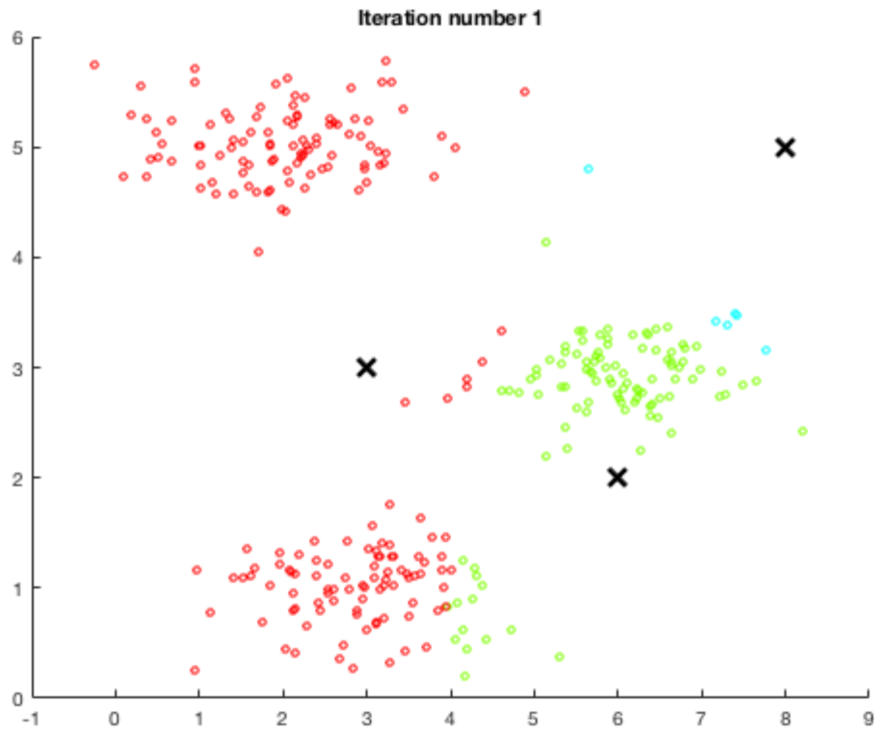
k-means clustering

- Input:
  - Number of clusters k
  - Data set $\{x^{(1)}, \cdots, x^{(m)}\}$

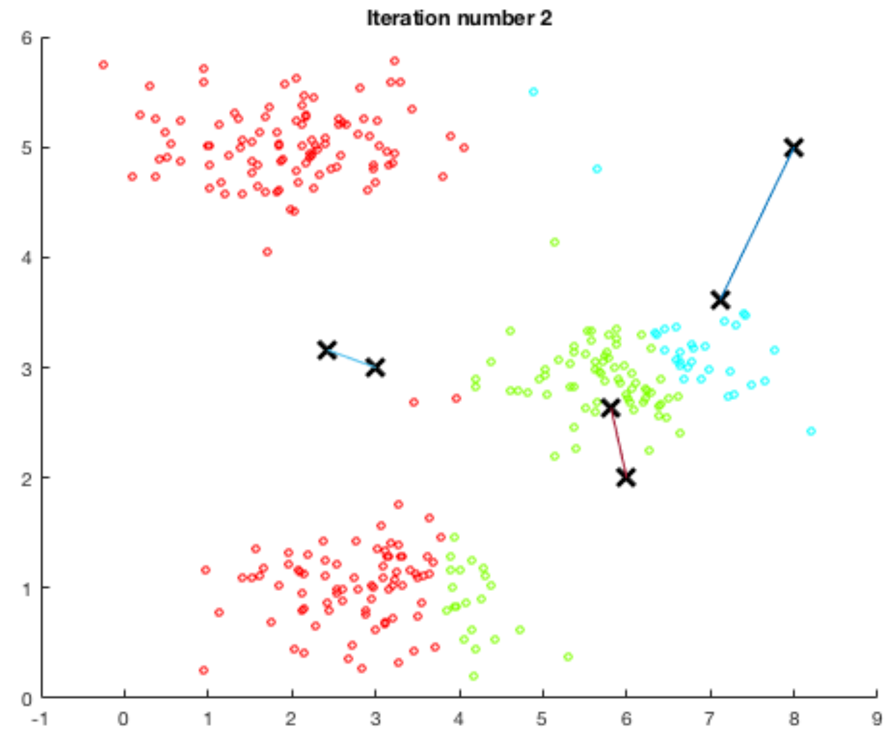
- Steps:
  - Randomly initialize k cluster centroids, i.e., mean/average
  - Repeat until converge
    - Assign each data to its closest centroid
    - Change the cluster center to the average of its assigned points

- Output:
  - Centroids of each cluster
  - The cluster label which is assigned to each data

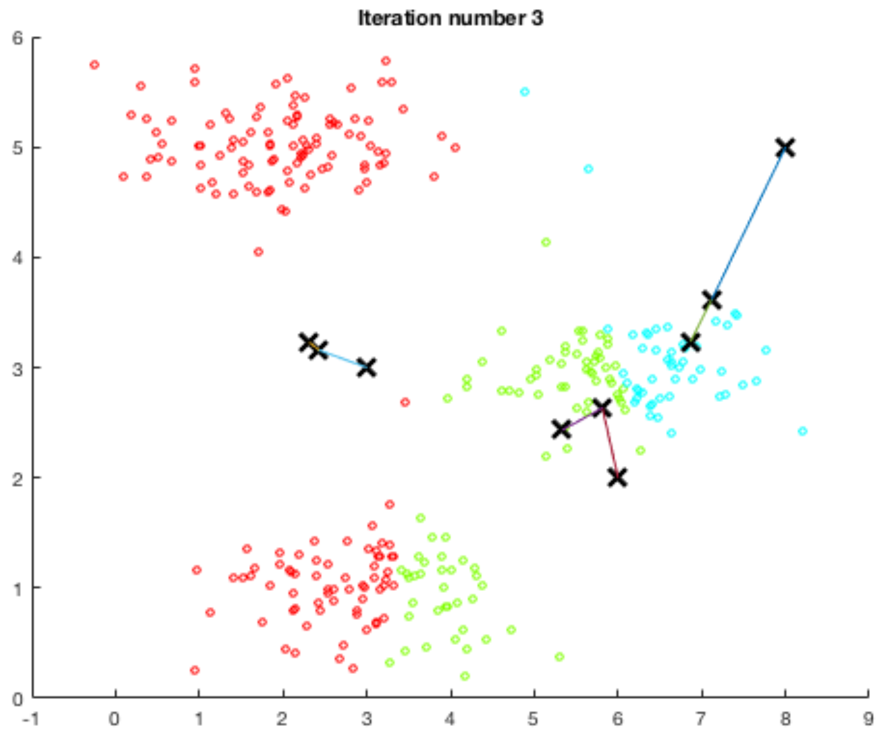
k-means clustering



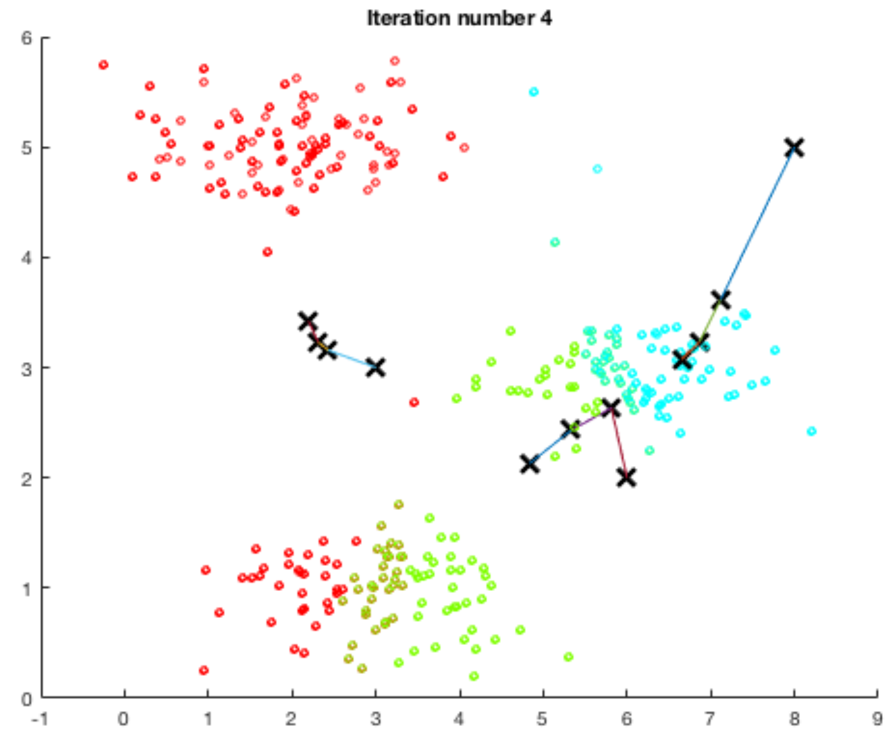
k-means clustering



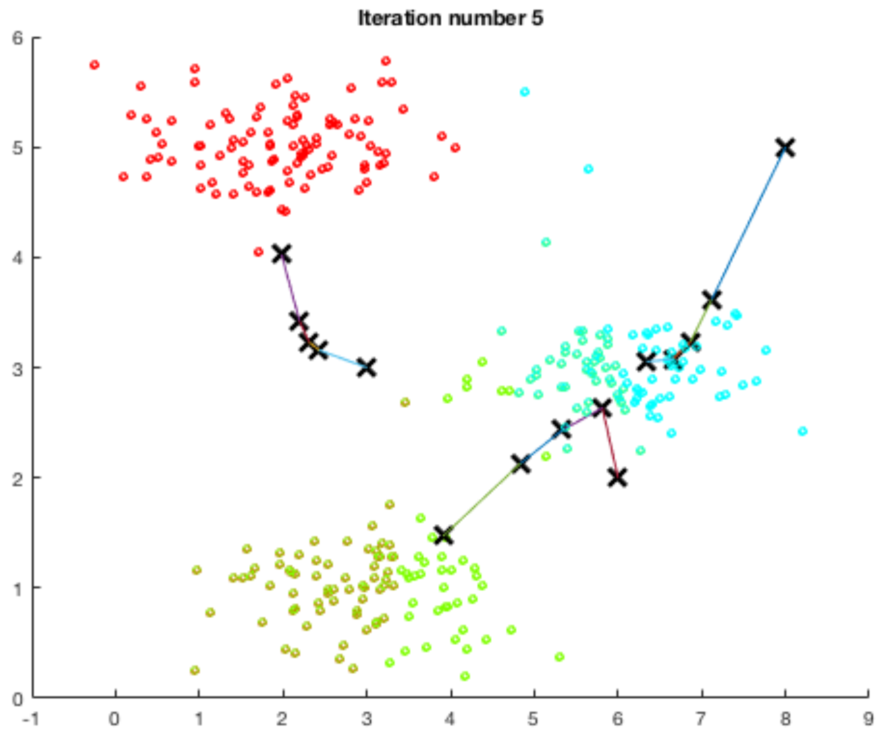
k-means clustering



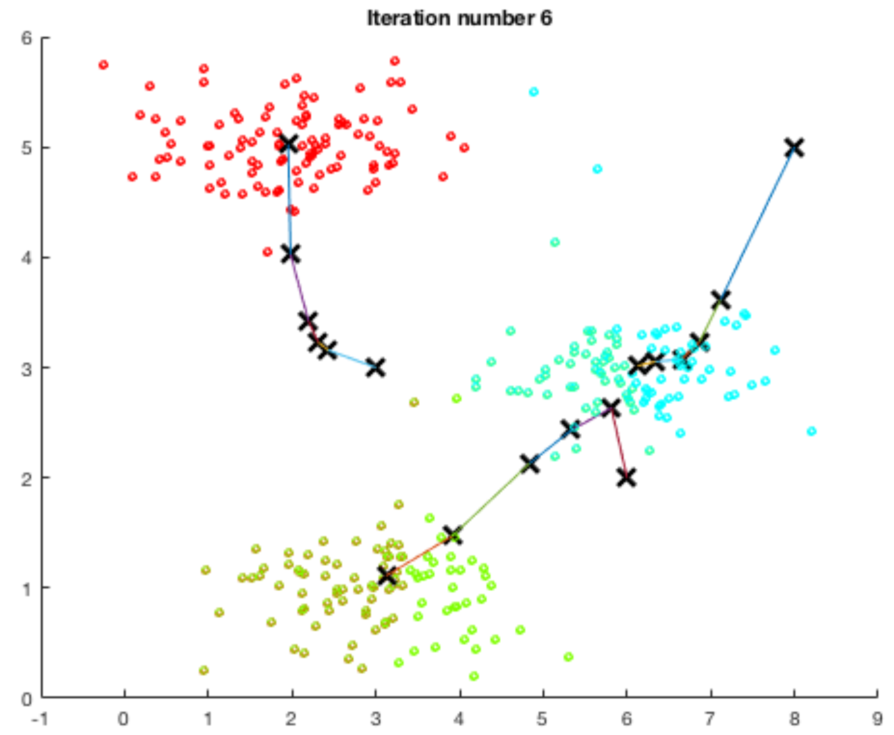
k-means clustering



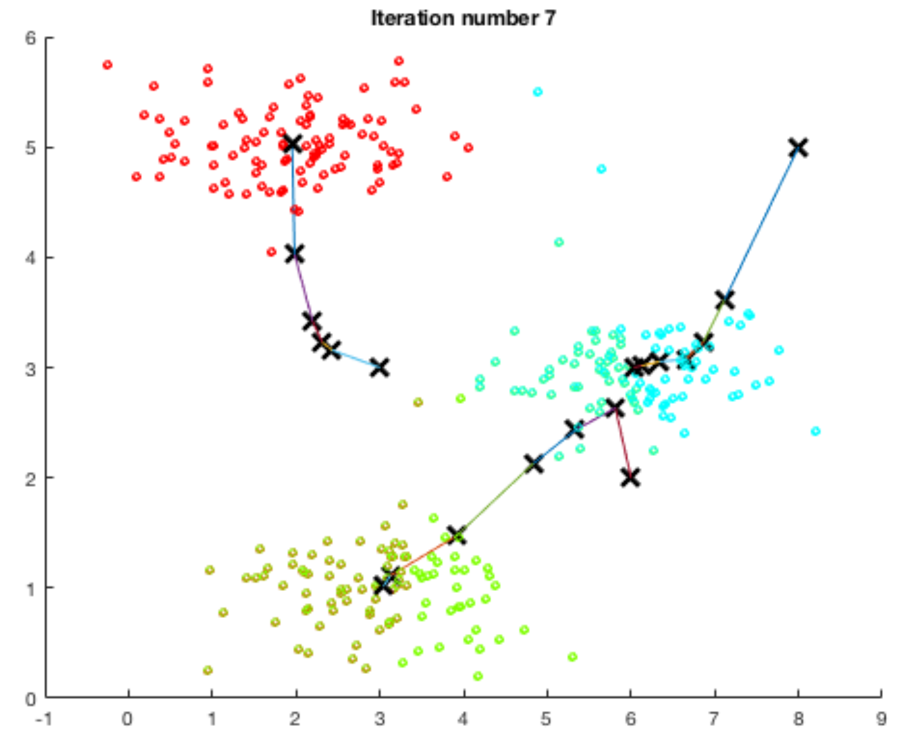
k-means clustering



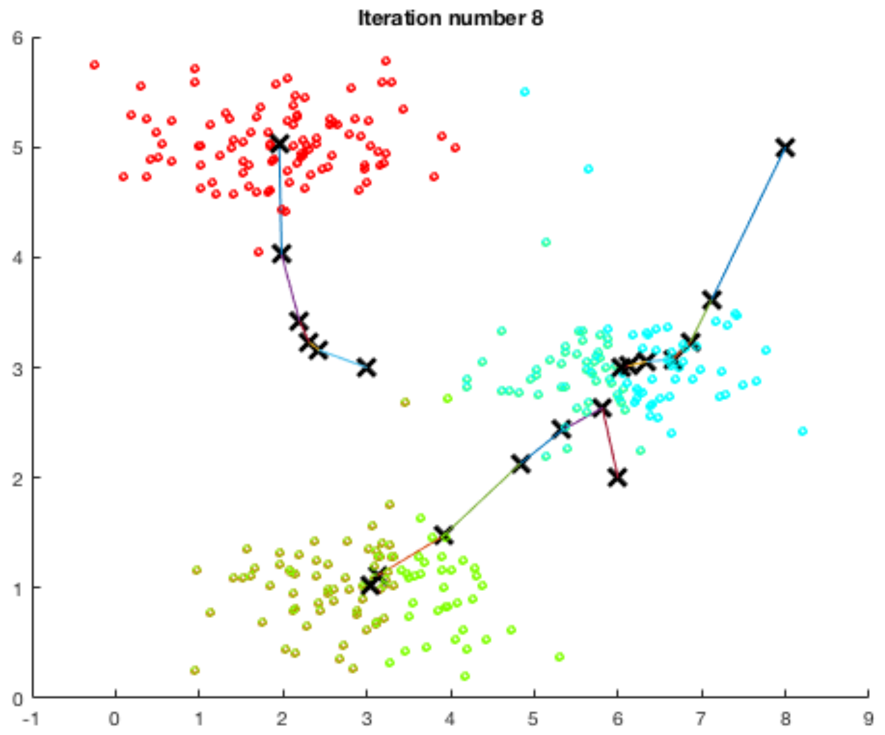
k-means clustering



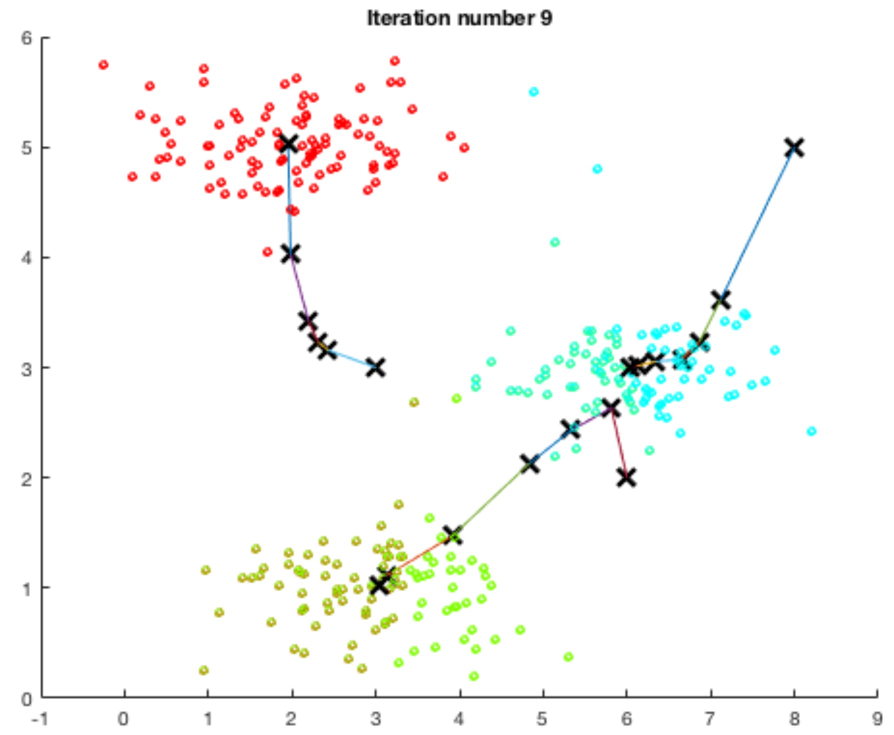
k-means clustering



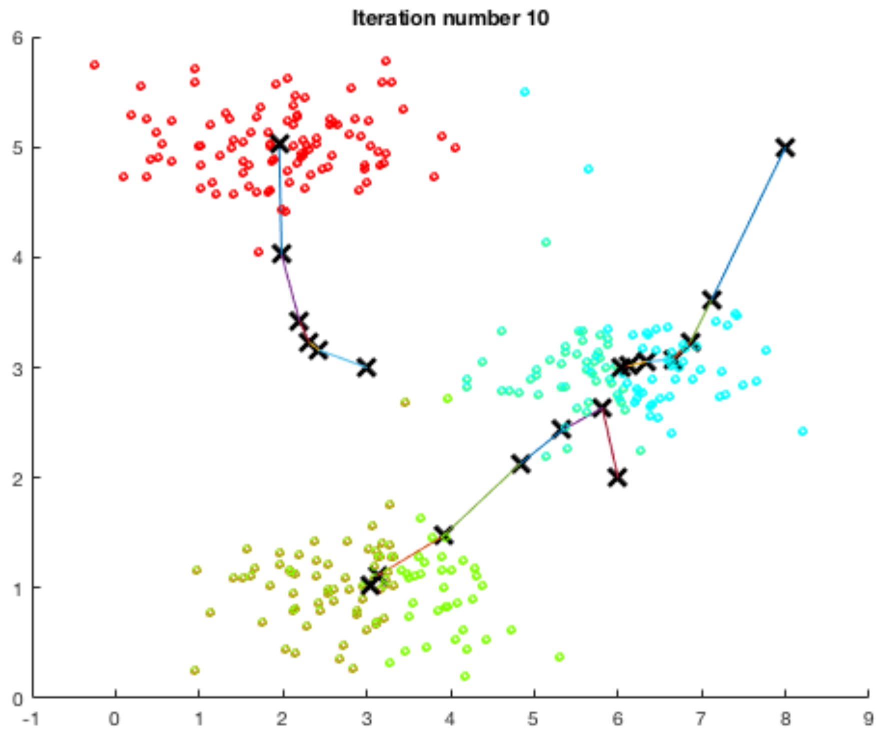
k-means clustering



k-means clustering



k-means clustering



k-means clustering

Let be $\mu_1, \ldots, \mu_K$ the centriods of clusters

Optimization objective:

$$\min_{\mu_k, b_k^{(i)}} \sum_{i=1}^{m} \sum_{k=1}^{K} b_k^{(i)} \left\| x^{(i)} - \mu_k \right\|^2$$

where $b_k^{(i)} = \begin{cases} 1 \quad if \quad \left\| x^{(i)} - \mu_k \right\| = \min \left\| x^{(i)} - \mu_j \right\| \quad j = 1, \ldots, K \\ 0 \quad \text{otherwise} \end{cases}$

$b_k^{(i)}$ is the indicator to represent if the data i is assigned to cluster k

Exact optimization of the k-means objective is NP-hard

k-means clustering

The solution for k-means algorithm is heuristic

Assign step: fix centroids $\mu_1, \ldots, \mu_K$, optimize $b_k^{(i)}$

Mean relocation step: fix $b_k^{(i)}$, optimize centroids $\mu_1, \ldots, \mu_K$

Each step is guaranteed to decrease the objective, thus guaranteed to converge to a local minimum.

k-means is a coordinate descent algorithm.

k-means clustering

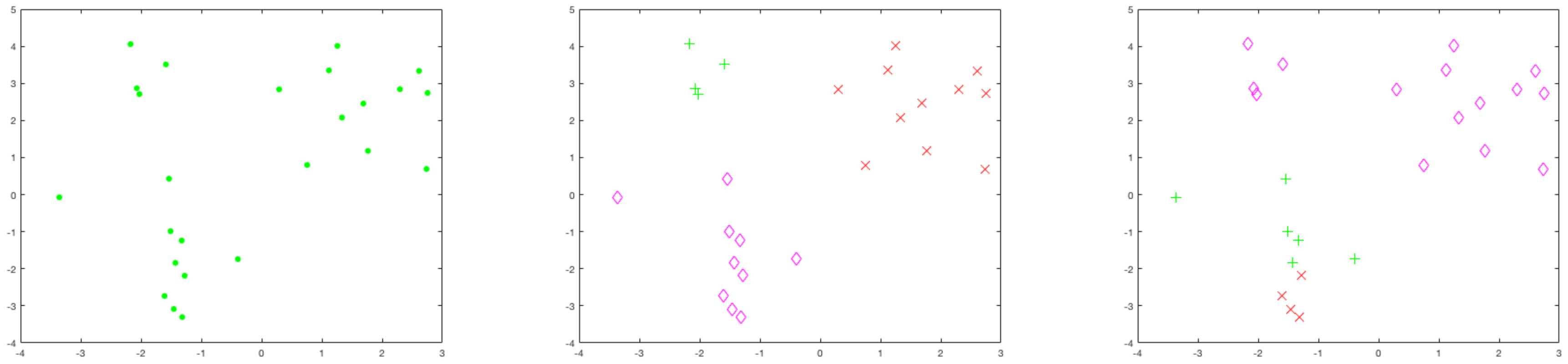
Algorithm:

- Randomly initialize k cluster centroids
- Repeat:
  - For 1 to m
    - Compute its indicator
  - For 1 to k
    - Compute the new cluster centroids

Random initialization

- k-means algorithm converges to a local minimum, so it is extremely sensitive to cluster center initialization.

- Bad initialization may lead to:
  - Poor convergence speed
  - Bad overall clustering



Handle bad initialization

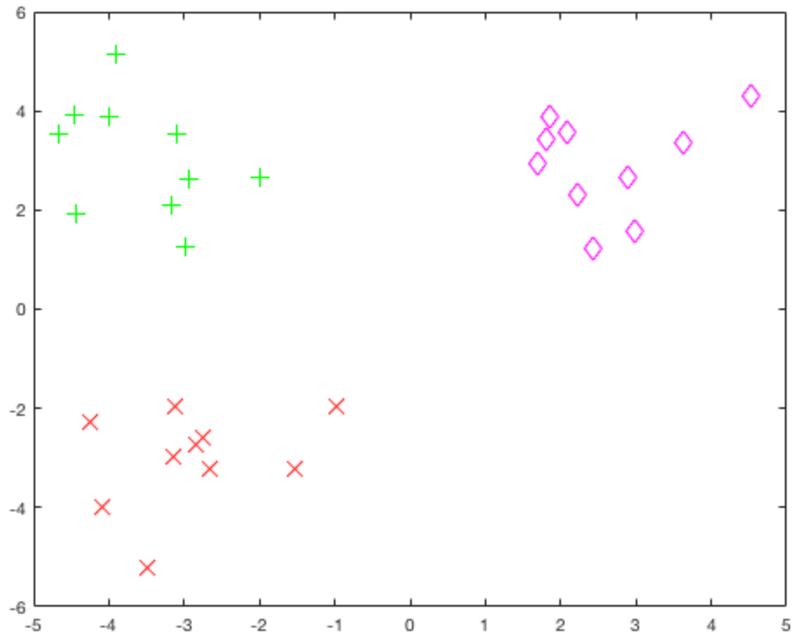
- Make initialized centroids evenly distribute in training set:
  - Choose first center as one of the examples, second which is the farthest from the first, third which is the farthest from both, and so on

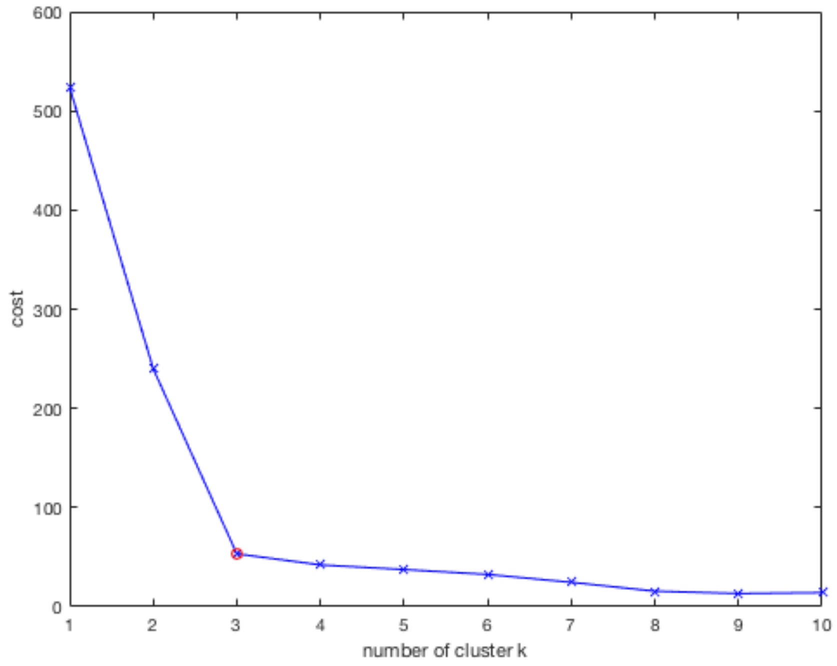
- Try-and-see, namely, try multiple initializations and choose the best result:
  - For $i = 1$ to 100
    - Run k-means
    - Compute cost function
  - Pick the result that gave the lowest loss

$$L = \sum_{i=1}^{m} \sum_{k=1}^{K} b_k^{(i)} \left\| x^{(i)} - \mu_k \right\|^2$$

Choosing the number of cluster *k*

- Elbow method: try different values of *k*, plot the k-means loss function *L*, and find out the "elbow-point" in the figure





Example: image segmentation by k-means

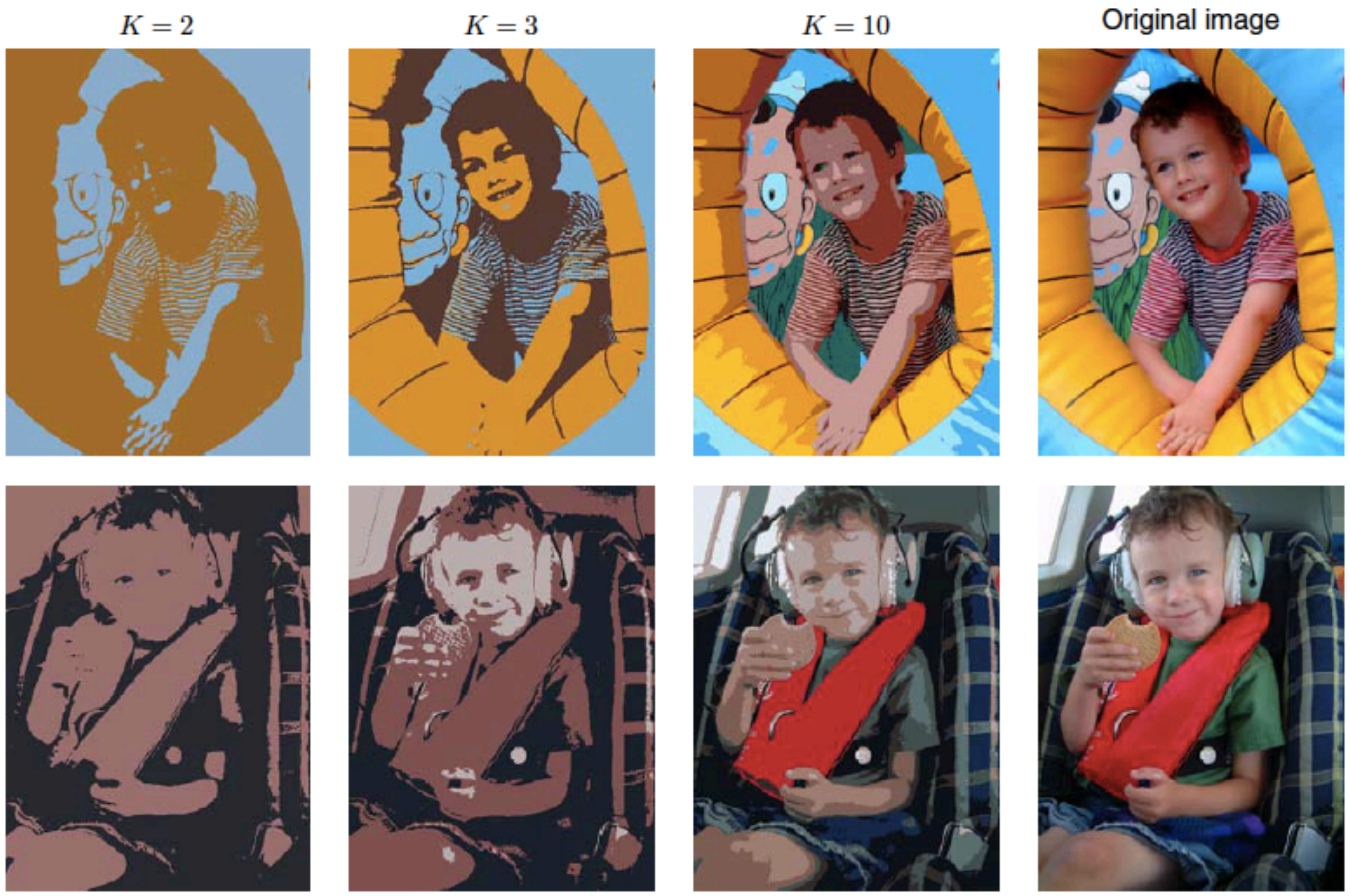
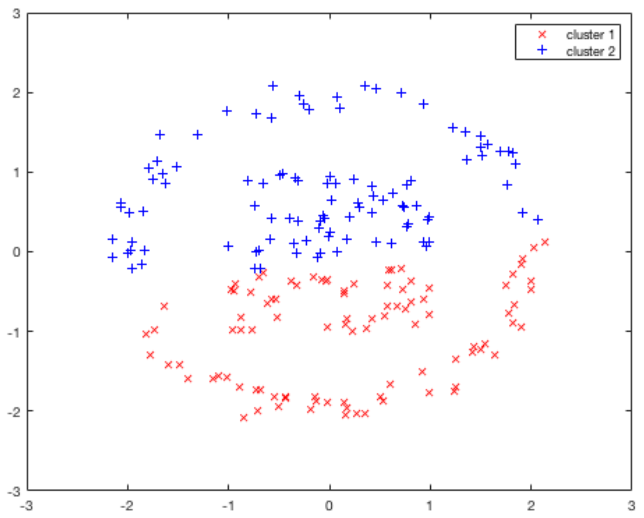


Figure 9.3 Two examples of the application of the K-means clustering algorithm to image segmentation showing the initial images together with their K-means segmentations obtained using various values of K. This also illustrates of the use of vector quantization for data compression, in which smaller values of K give higher compression at the expense of poorer image quality.

k-means clustering: summary

- k-means is the most popular and widely used clustering algorithm
- Converge to a local minimum
- Computational complexity in each iteration:
  - Assign data points to closest cluster center $O(mk)$
  - Change the cluster center to the average of its assigned points $O(m)$
- "Hard assignment", either 1 or 0 to each cluster
- Limitations:
  - Sensitive to outliers
  - Works well only for round shaped, and of roughly equal sizes/density clusters
- Do not work well on non-linear clusters



k-medoids

- In many practical settings, Euclidean distance is not appropriate
- For example:
  - Discrete multivariate data, such as purchase histories
  - Positive data, such as time spent on a web-page
- k-medoids is an algorithm that only requires knowing distances between data points $x^{(i)}$ and $x^{(j)}$, $d(x^{(i)}, x^{(j)})$
- No need to define the mean
- Each of the clusters is associated with its most typical example

k-medoids clustering

Algorithm:

- Randomly choose *k* data as initial cluster center
- Repeat:
  - Assign each data point to its closest center
  - For each cluster, find the data point in that cluster that is closest to the other points in that cluster
  - Set each cluster center equal to their closest data points

Agglomerative vs. k-means

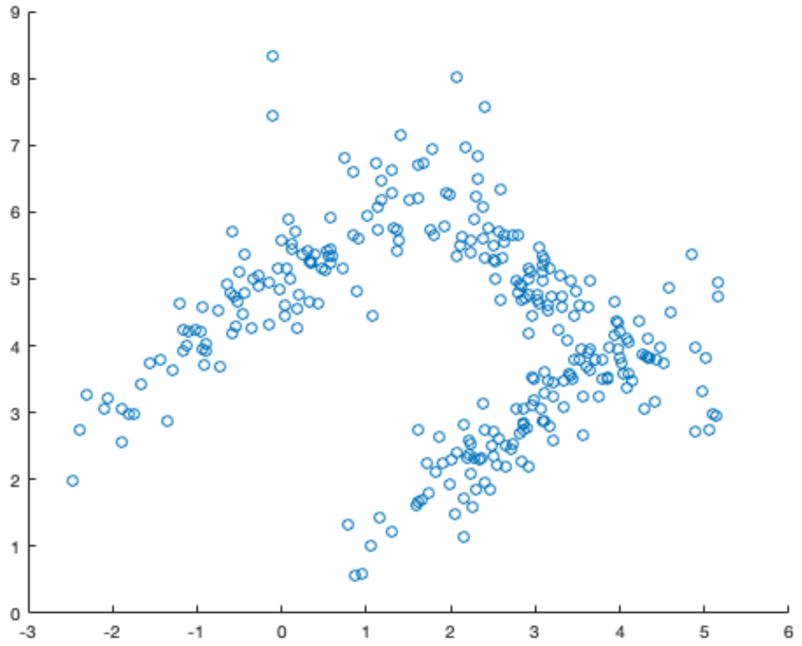
Agglomerative:

- Gives different partitioning depending on the level-of-resolution we are looking at
- Doesn't need the number of clusters to be specified
- May be slow as it has to compute distance many times $O(m^2 \log m)$

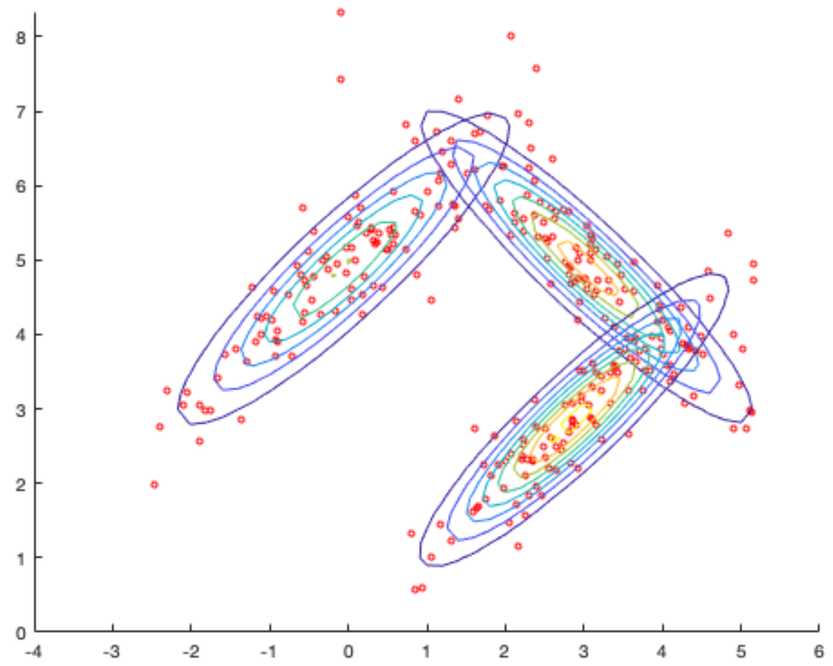
k-means:

- Produces a single partitioning
- Needs the number of clusters to be specified
- More efficient $O(m(k+1)i)$, where i is the number of iteration

GMM



GMM

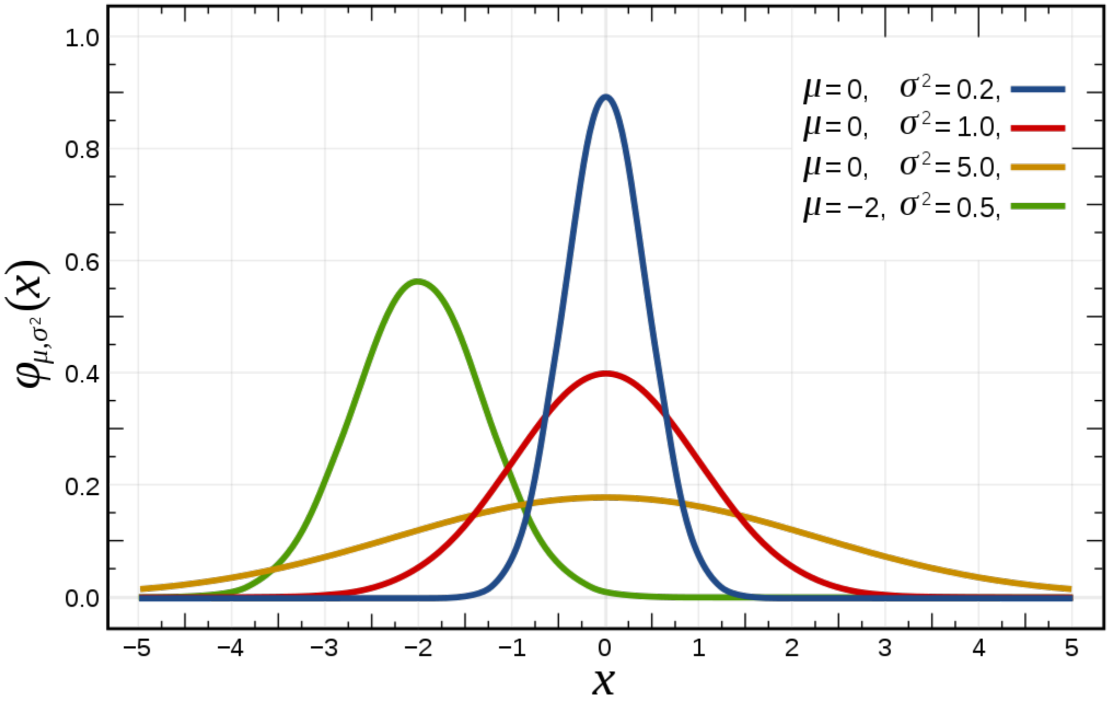


Gaussian distribution

The probability density of the normal (Gaussian) distribution is

$$p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean, and σ^2 is the variance



Multivariate Gaussian distribution

The probability density function of multivariate (n-dim) Gaussian distribution is given by:

$$p(x|\mu,\Sigma\,) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}\,e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

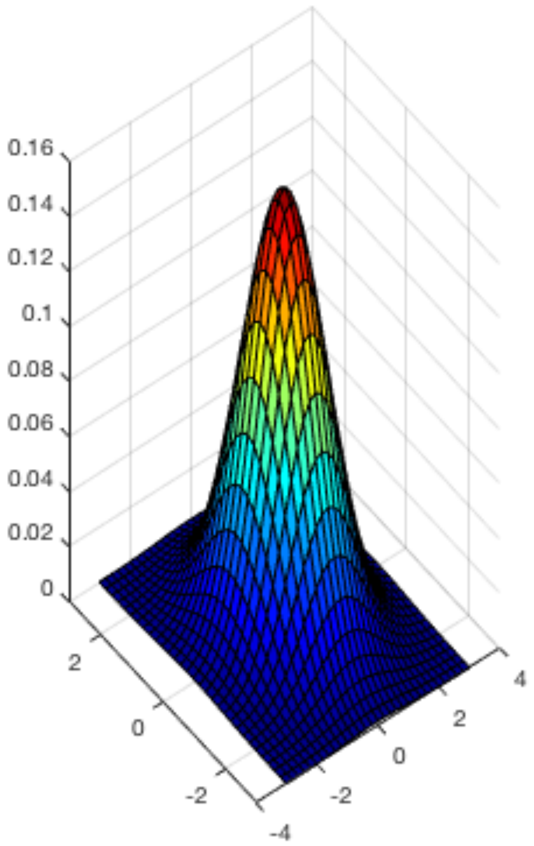
where μ is the mean, Σ is the covariance matrix, and $|\Sigma|$ denotes the dterminant of the covariance matrix

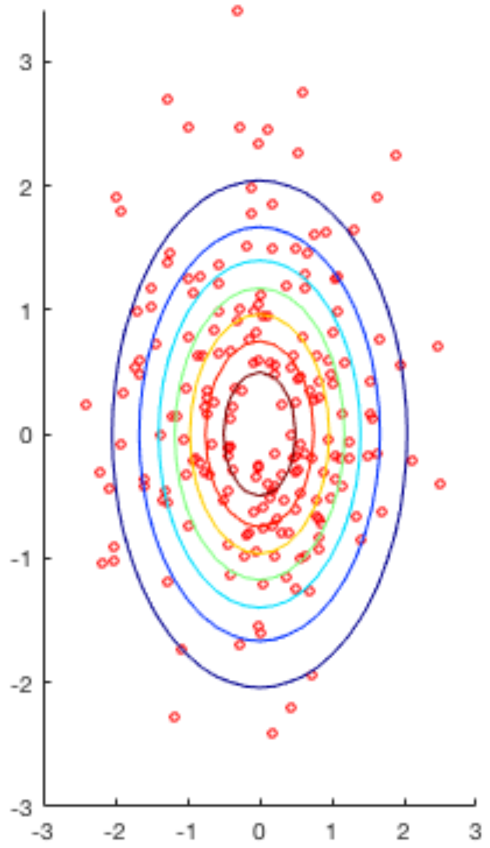
Patrial derivative to mean and covariance:

$$\frac{\partial}{\partial\mu}p(x|\mu,\Sigma\,) = p(x|\mu,\Sigma\,)\Sigma^{-1}(x-\mu)$$

$$\frac{\partial}{\partial\Sigma}p(x|\mu,\Sigma\,) = -\frac{1}{2}p(x|\mu,\Sigma\,)(\Sigma^{-1}-\Sigma^{-1}(x-\mu)(x-\mu)^T\Sigma^{-1})$$

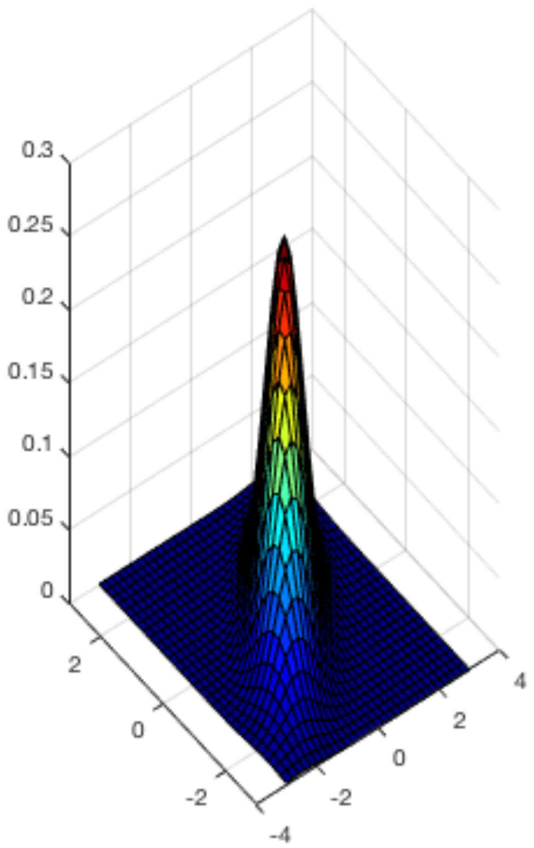
Multivariate Gaussian distribution

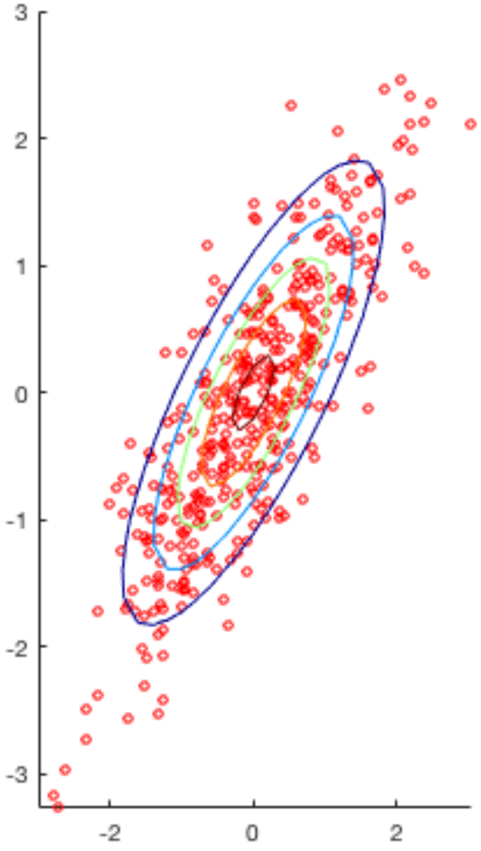




$$\mu = \begin{bmatrix} 0 & 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

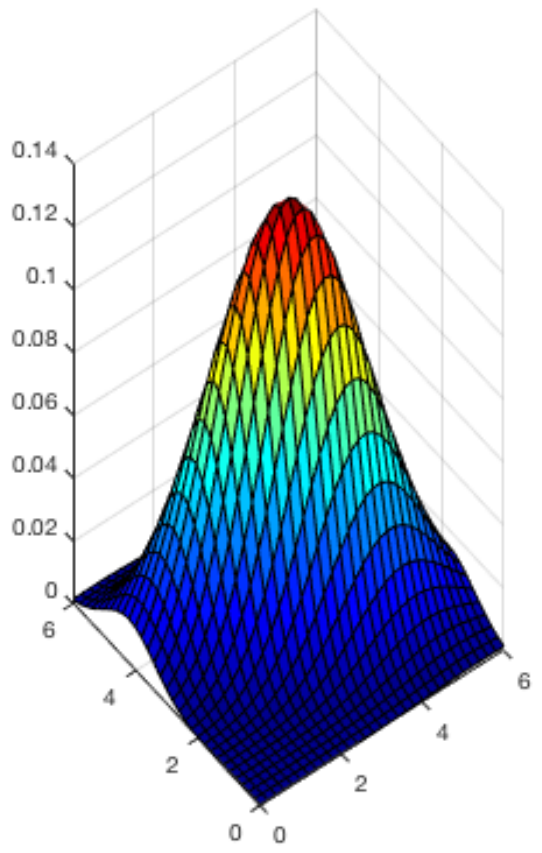
Multivariate Gaussian distribution

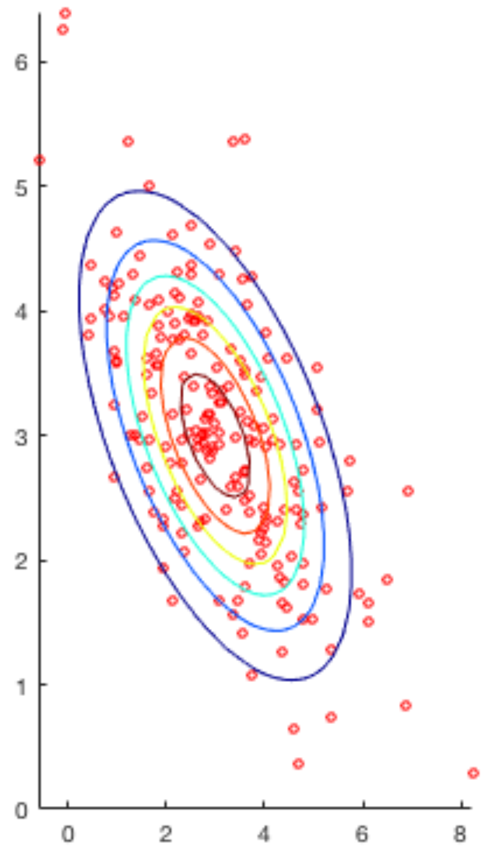




$$\mu = \begin{bmatrix} 0 & 0 \end{bmatrix} \Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Multivariate Gaussian distribution





$$\mu = \begin{bmatrix} 3 & 3 \end{bmatrix} \; \Sigma = \begin{bmatrix} 2 & -0.8 \\ -0.8 & 1 \end{bmatrix}$$

GMM – model representation

- Let $z^{(i)}$ be the latent variable, and $z^{(i)} \in \{1,2,\cdots,k\}$. $p(z^{(i)} = j)$ denotes the probability that $x^{(i)}$ generated from jth Gaussian model (k models in total)

- Gaussian mixture density is given by:

$$p\left(x^{(i)}\right) = \sum_{j=1}^{k} p\left(z^{(i)} = j\right) p(x^{(i)} | \mu^{(j)}, \Sigma^{(j)})$$

- Assume $z^{(i)} \sim Multinomial(\alpha)$, where $\alpha_j \geq 0, \sum_{j=1}^{k} \alpha_j = 1$. Correspondingly,

$$p\left(z^{(i)} = j\right) = \alpha_j$$

- Gaussian mixture density is rewritten as:

$$p\left(x^{(i)}\right) = \sum_{j=1}^{k} \alpha_j p(x^{(i)} | \mu^{(j)}, \Sigma^{(j)})$$

where α_j is mixture coefficient

GMM

- The parameters of GMM are thus $\alpha_j, \mu^{(j)}, \Sigma^{(j)}\ \forall j = 1,2,\cdots,k$

- Given the training data $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$, Maximum Likelihood Estimation (MLE) maximizes the following log likelihood:

$$\begin{aligned} L &= \log\left(\prod_{i=1}^{m} p\left(x^{(i)}\right)\right) \\ &= \sum_{i=1}^{m} \log\left(p\left(x^{(i)}\right)\right) \\ &= \sum_{i=1}^{m} \log\left(\sum_{j=1}^{k} \alpha_j p(x^{(i)} | \mu^{(j)}, \Sigma^{(j)})\right) \end{aligned}$$

GMM

If parameters $\alpha_j, \mu^{(j)}, \Sigma^{(j)}\ \forall j = 1, 2, \ldots, k$ maximize log likelihood, then we have:

$$\frac{\partial L}{\partial \mu^{(j)}} = \sum_{i=1}^{m} \frac{\alpha_j}{\sum_{l=1}^{k} \alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)})} \frac{\partial}{\partial \mu^{(j)}} p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)}) = 0$$

$$\frac{\partial L}{\partial \Sigma^{(j)}} = \sum_{i=1}^{m} \frac{\alpha_j}{\sum_{l=1}^{k} \alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)})} \frac{\partial}{\partial \Sigma^{(j)}} p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)}) = 0$$

Combined with

$$\frac{\partial}{\partial \mu} p(x \mid \mu, \Sigma) = p(x \mid \mu, \Sigma) \Sigma^{-1} (x - \mu)$$

$$\frac{\partial}{\partial \Sigma} p(x \mid \mu, \Sigma) = -\frac{1}{2} p(x \mid \mu, \Sigma)(\Sigma^{-1} - \Sigma^{-1}(x - \mu)(x - \mu)^T \Sigma^{-1})$$

GMM

Bayes rule: $p(A \mid B)p(B) = p(B \mid A)p(A)$

Applied to GMM: $p(z^{(i)} = j \mid x^{(i)})p(x^{(i)}) = p(z^{(i)} = j)p(x^{(i)} \mid z^{(i)} = j)$

$$\Rightarrow p(z^{(i)} = j \mid x^{(i)}) \sum_{l=1}^{k} \alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)}) = \alpha_j p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)})$$

Posterior probability:

$$\underbrace{p(z^{(i)} = j \mid x^{(i)})}_{\gamma_{ij}} = \frac{\alpha_j p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)})}{\sum_{l=1}^{k} \alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)})}$$

The mean and variance are computed as:

$$\mu^{(j)} = \frac{\sum_{i=1}^{m} \gamma_{ij} x^{(i)}}{\sum_{i=1}^{m} \gamma_{ij}} \qquad \Sigma^{(j)} = \frac{\sum_{i=1}^{m} \gamma_{ij} (x^{(i)} - \mu^{(j)})(x^{(i)} - \mu^{(j)})^T}{\sum_{i=1}^{m} \gamma_{ij}}$$

GMM

Bayes rule: $p(A \mid B)p(B) = p(B \mid A)p(A)$

Applied to GMM: $p(z^{(i)} = j \mid x^{(i)})p(x^{(i)}) = p(z^{(i)} = j)p(x^{(i)} \mid z^{(i)} = j)$

$$\Rightarrow p(z^{(i)} = j \mid x^{(i)})\sum_{l=1}^{k}\alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)}) = \alpha_j p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)})$$

Posterior probability:

$$\underbrace{p(z^{(i)} = j \mid x^{(i)})}_{\gamma_{ij}} = \frac{\alpha_j p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)})}{\sum_{l=1}^{k}\alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)})}$$

Soft assignment

The mean and variance are computed as:

$$\mu^{(j)} = \frac{\sum_{i=1}^{m}\gamma_{ij}x^{(i)}}{\sum_{i=1}^{m}\gamma_{ij}} \qquad \Sigma^{(j)} = \frac{\sum_{i=1}^{m}\gamma_{ij}(x^{(i)} - \mu^{(j)})(x^{(i)} - \mu^{(j)})^T}{\sum_{i=1}^{m}\gamma_{ij}}$$

GMM

We still have an additional constraint: $\alpha_j \geq 0, \sum_{j=1}^{k} \alpha_j = 1$

Introduce the Lagrange multiplier:

$$J = L + \lambda(\sum_{j=1}^{k} \alpha_j - 1)$$

Take derivative, we have

$$\frac{\partial J}{\partial \alpha_j} = 0 \Rightarrow \alpha_j = \frac{1}{m} \sum_{i=1}^{m} \gamma_{ij}$$

GMM algorithm

Randomly initialize parameters $\alpha_j, \mu^{(j)}, \Sigma^{(j)} \ \forall j = 1, 2, \ldots, k$

Repeat until converge:

E-step. Compute posterior probability γ_{ij}:

$$\gamma_{ij} = \frac{\alpha_j p(x^{(i)} \mid \mu^{(j)}, \Sigma^{(j)})}{\sum_{l=1}^{k} \alpha_l p(x^{(i)} \mid \mu^{(l)}, \Sigma^{(l)})}$$

M-step. Update the parameters using the current γ_{ij}:

$$\alpha_j = \frac{1}{m} \sum_{i=1}^{m} \gamma_{ij} \qquad \mu^{(j)} = \frac{\sum_{i=1}^{m} \gamma_{ij} x^{(i)}}{\sum_{i=1}^{m} \gamma_{ij}} \qquad \Sigma^{(j)} = \frac{\sum_{i=1}^{m} \gamma_{ij} (x^{(i)} - \mu^{(j)})(x^{(i)} - \mu^{(j)})^T}{\sum_{i=1}^{m} \gamma_{ij}}$$

The General EM algorithm

Given a joint distribution $p(X,Z|\theta)$ the over observed variables X and the latent variables Z, governed by parameters θ, the goal is to maximize the likelihood function $p(X|\theta)$ with respect to θ.

- Choose an initial setting for the parameters θ
- Repeat until converge:

  E-step: evaluate $p(X,Z|\theta)$ using current θ

  M-step: evaluate θ given by Z

GMM vs. k-means

GMM (soft assignment)

- Randomly initialize parameters of *k* Gaussian distribution and mixture coefficients
- Repeat until converge:
  - E-step. Compute soft membership, i.e., posterior probability
  - M-step. Update the parameters using the current soft membership

k-means (hard assignment)

- Randomly initialize *k* cluster centroids
- Repeat until converge
  - Assign each data to its closest centroid
  - Update the cluster center using its current assigned points