THE UNIVERSITY of ADELAIDE

CRICOS PROVIDER 00123M

Unsupervised Learning: Clustering

Lingqiao Liu
University of Adelaide

Outlines

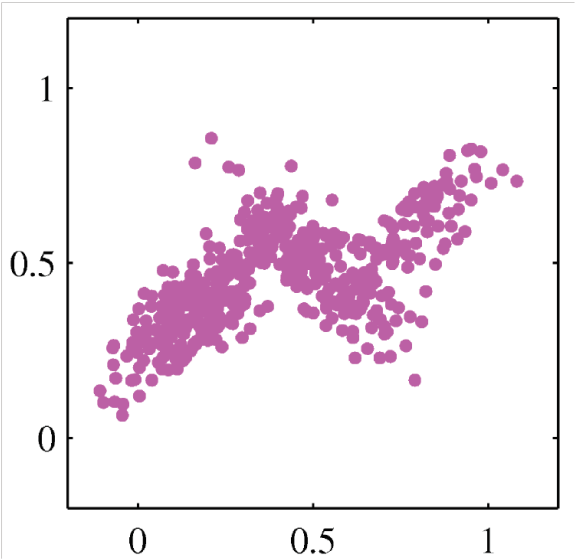
- Overview of Unsupervised Learning and Clustering
- K-means clustering
- Gaussian mixture models (GMM)
  - Distribution modeling
  - GMM
  - Latent variable
  - EM algorithm

Unsupervised learning

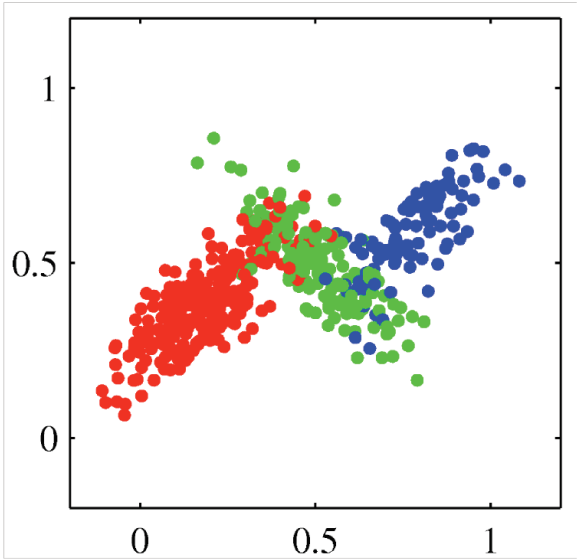
- Learning without supervision
  - Find the distribution of data
  - Learning to generate samples
  - Clustering
  - Anomaly detection
  - Feature learning
- Clustering
  - One of the most important unsupervised learning tasks
  - Clustering is the process of identifying groups, or clusters, of data points in a (usually) multidimensional space.
  - Connection to distribution modeling: related to mixture model

Clustering

Discover groups such that samples within a group are more similar to each other than samples across groups.



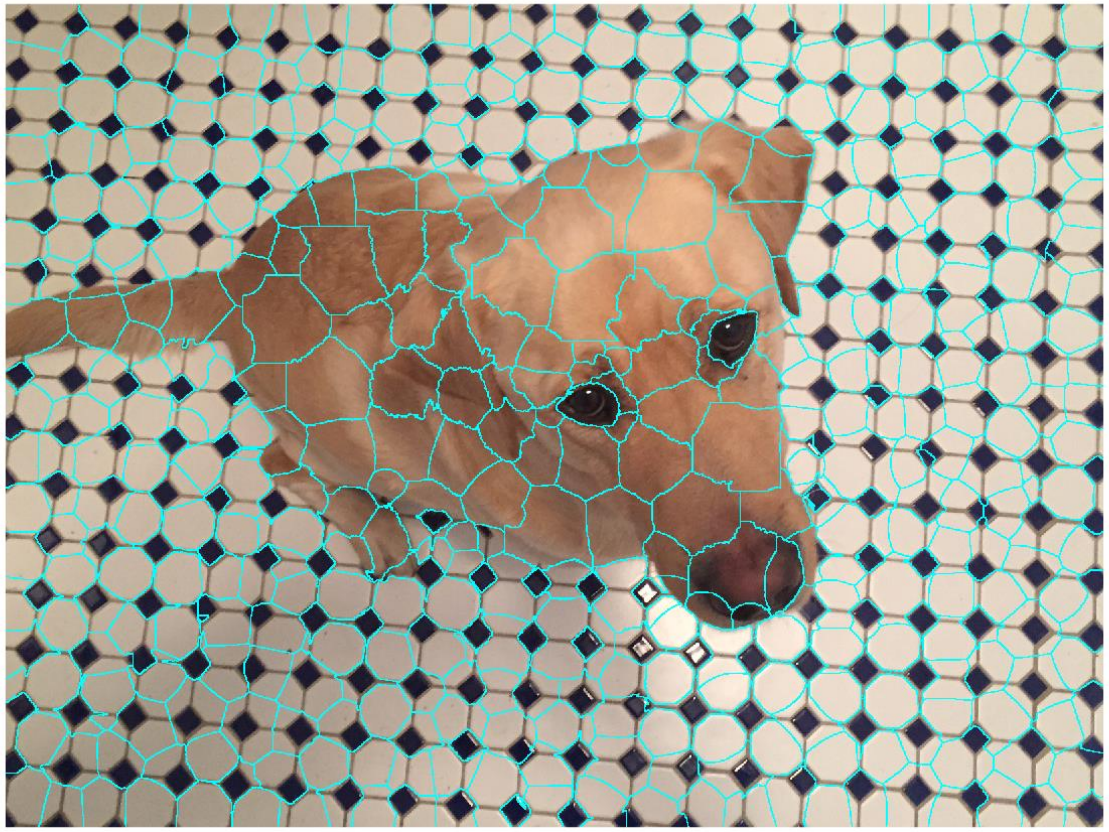
→



Application of clustering: Segmentation

http://people.cs.uchicago.edu/ pff/segment

Superpixel

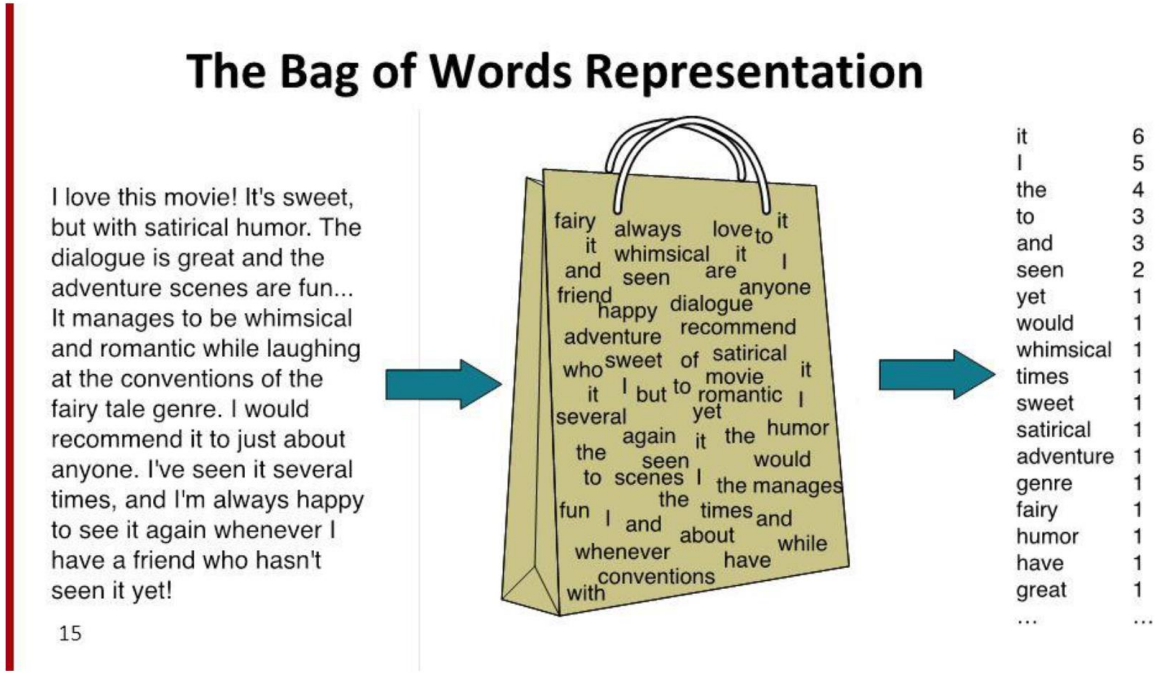


X = (r,g,b,x,y)

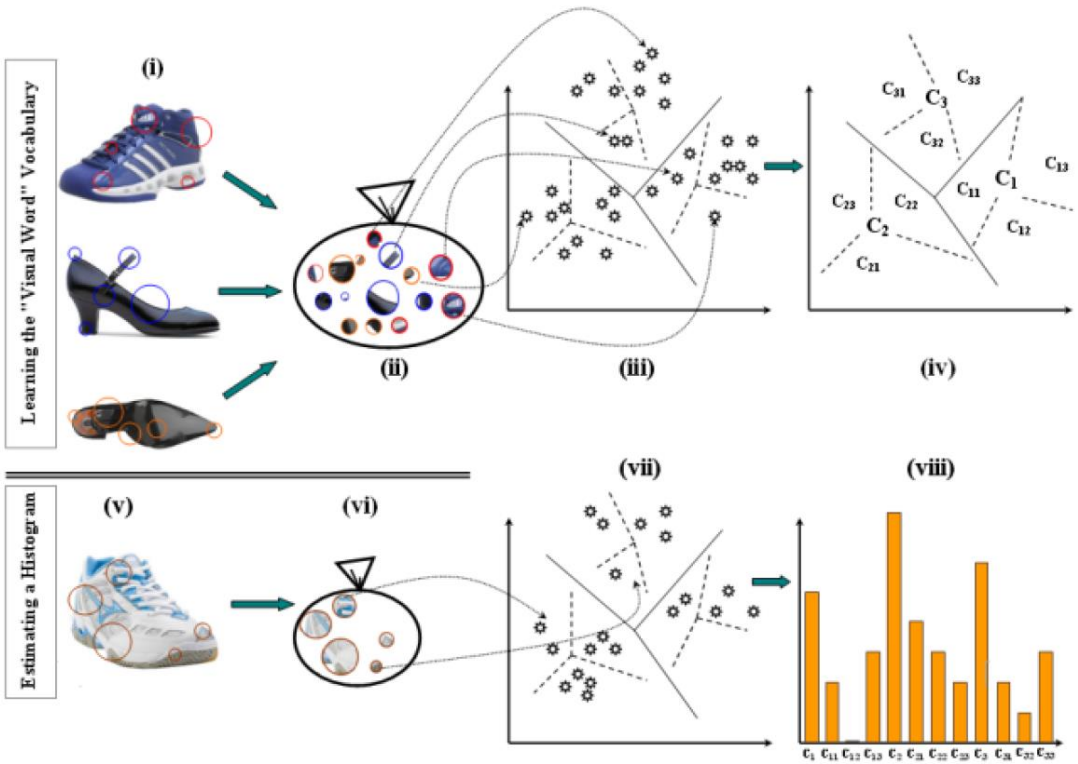
Application of clustering: Vector quantization in Bag-of-feature model

- Bag-of-features model
  - Extended from bag-of-words model

Yuo cna porbalby raed tihs esaliy desptie teh msispeillgns.



Application of clustering:



Vector Quantization: Build a dictionary with k centres.
For a vector, find its closet centre and use its ID as its "visual word"

Ingredient for Clustering

- A dissimilarity function between samples.
- A loss function to evaluate clusters.
- Algorithm that optimizes this loss function.

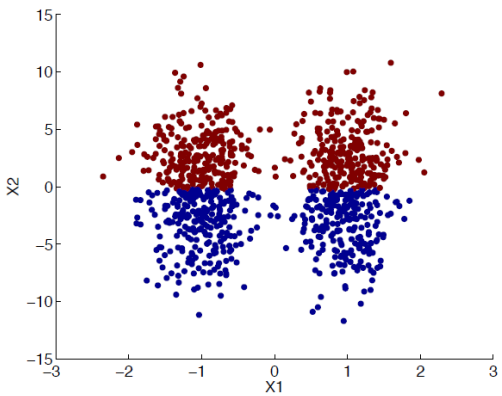
Dissimilar function

- Data point x_i has features $x_{ij}, j = 1, \ldots, p$.
- One choice of dissimilarity function is the Euclidean distance

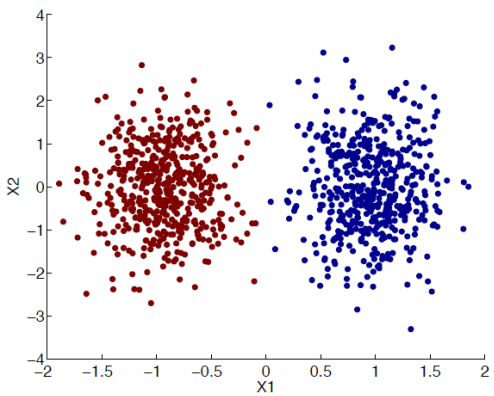
$$D(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{i'j})^2}$$

- Resulting clusters invariant to rotation and translation of features but not to scaling.
- If the features have different scales, standardize the data.

Standardization

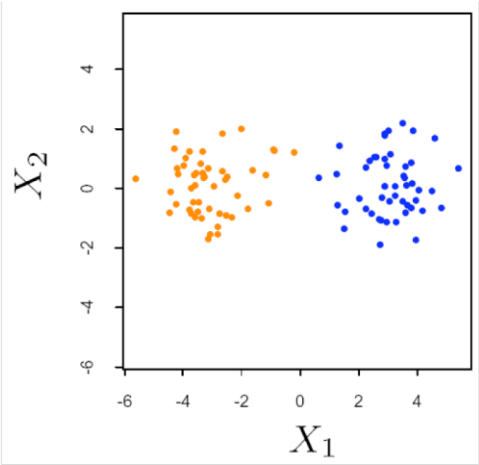


Without standardization

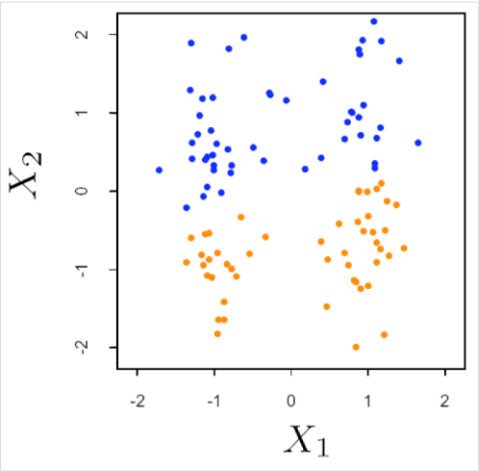


With standardization

Standardization is not always helpful



Without standardization



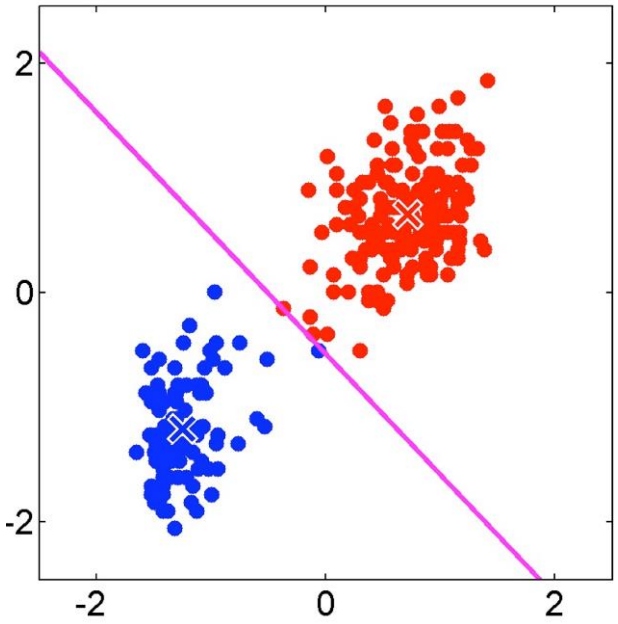
With standardization

Outlines

- Overview of Unsupervised Learning and Clustering
- K-means clustering
- Gaussian mixture models (GMM)
  - Distribution modeling
  - GMM
  - Latent variable
  - EM algorithm

K-means clustering

- One of the most commonly used clustering methods
- Input: data points and the number of clusters, k
- Basic idea
  - Each sample can only fall into one of the k groups
  - From each group, we can calculate the mean vectors, that is, the average of samples falling into a group
  - Any sample should be closest to the mean vector of its own group than the mean vectors of other groups



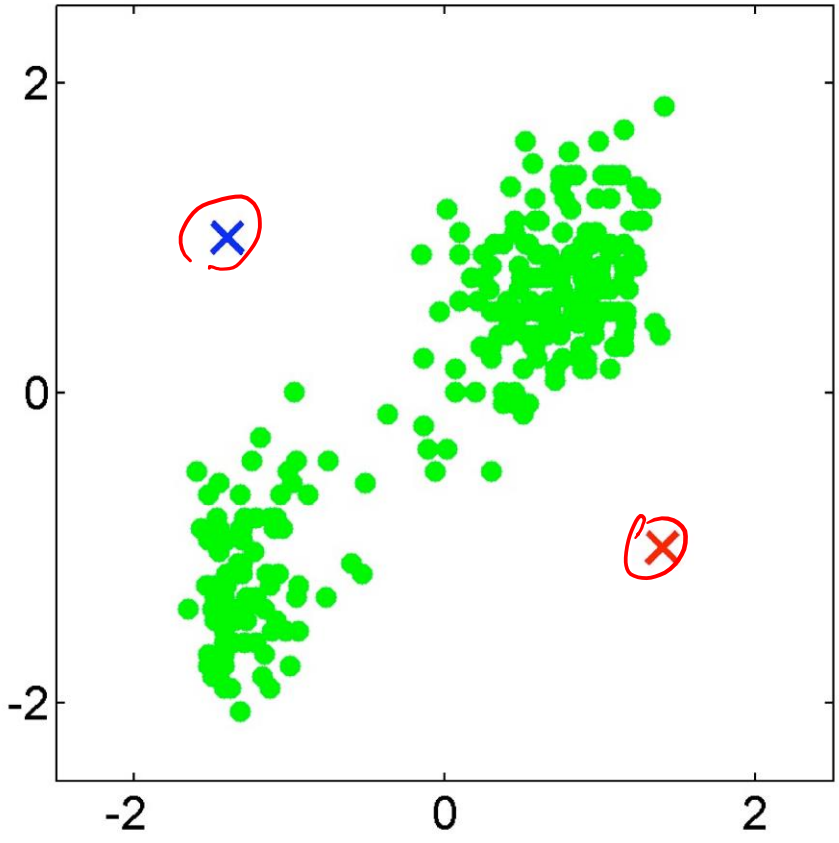
K-means clustering

- How could we find the assignment and mean vectors of each group?
- It is a chicken egg problems
  - If we know the assignment, we can easily calculate the mean vectors
  - If we know the mean vectors, we can easily calculate the assignment
- K-means algorithm takes an iterative approach for solving the problem

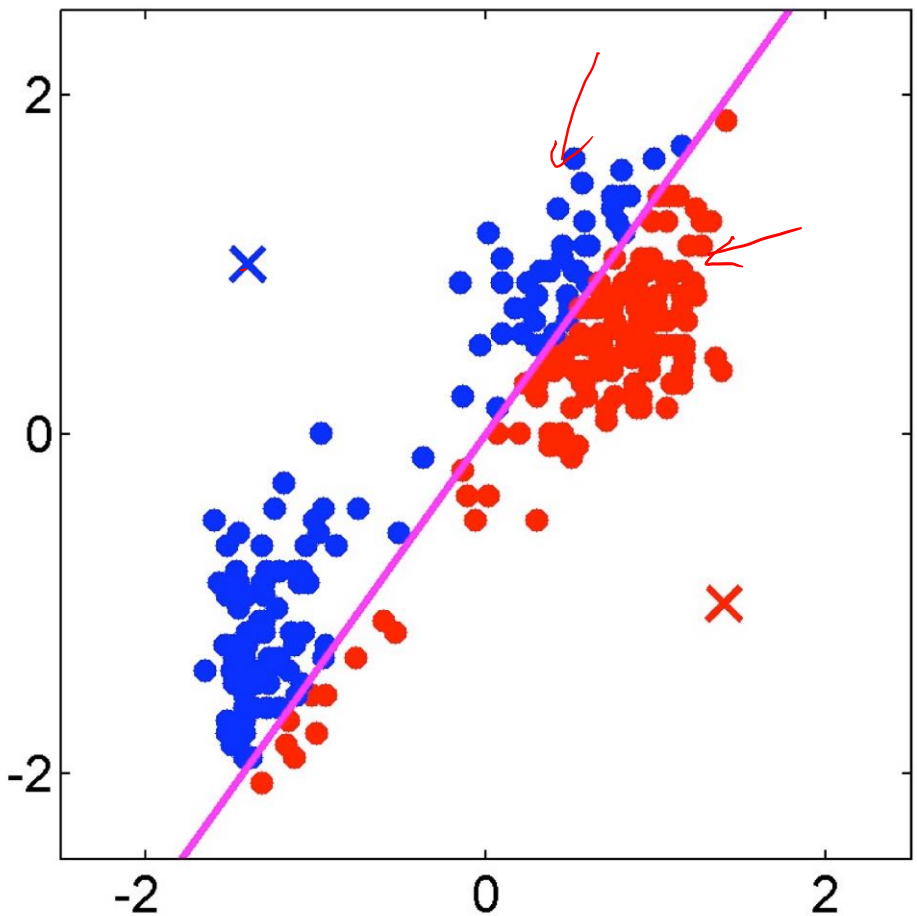
K means algorithm

- E-step: fixed the current mean vectors of each group. If a sample is closest to the i-th mean vector, then the sample is assigned to the i-th group
- M-step: fixed the assignment, calculate the mean vector of each group
- Iterate E step and M step, until converge

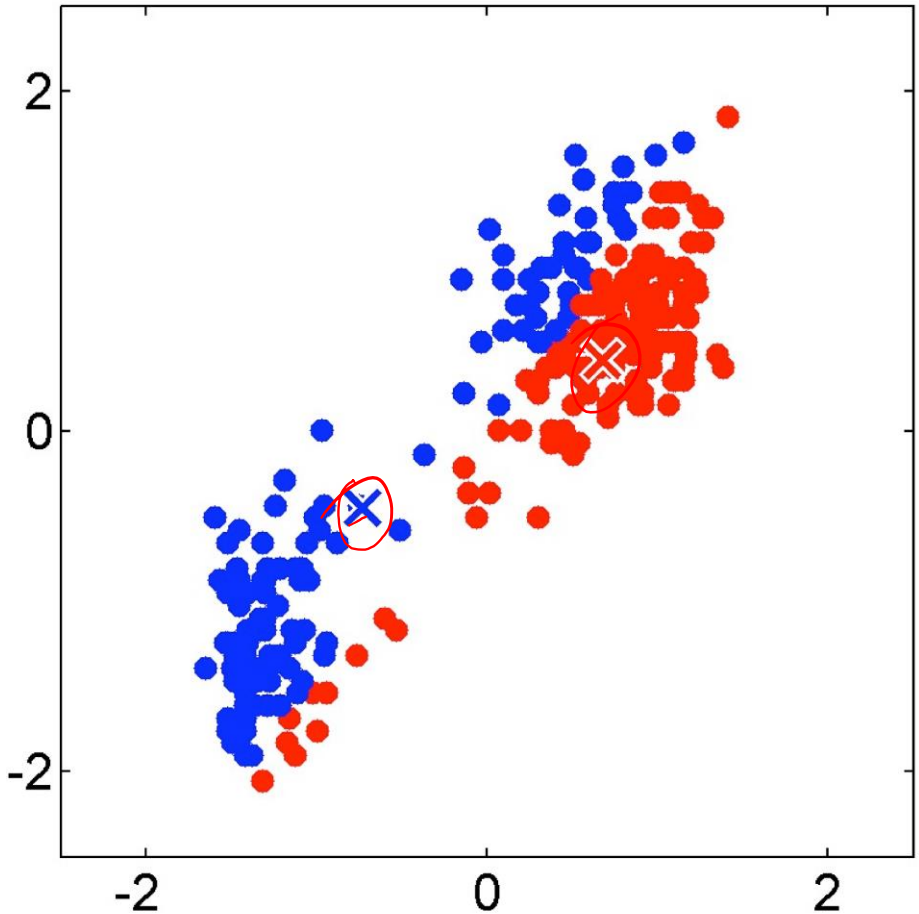
K-means example



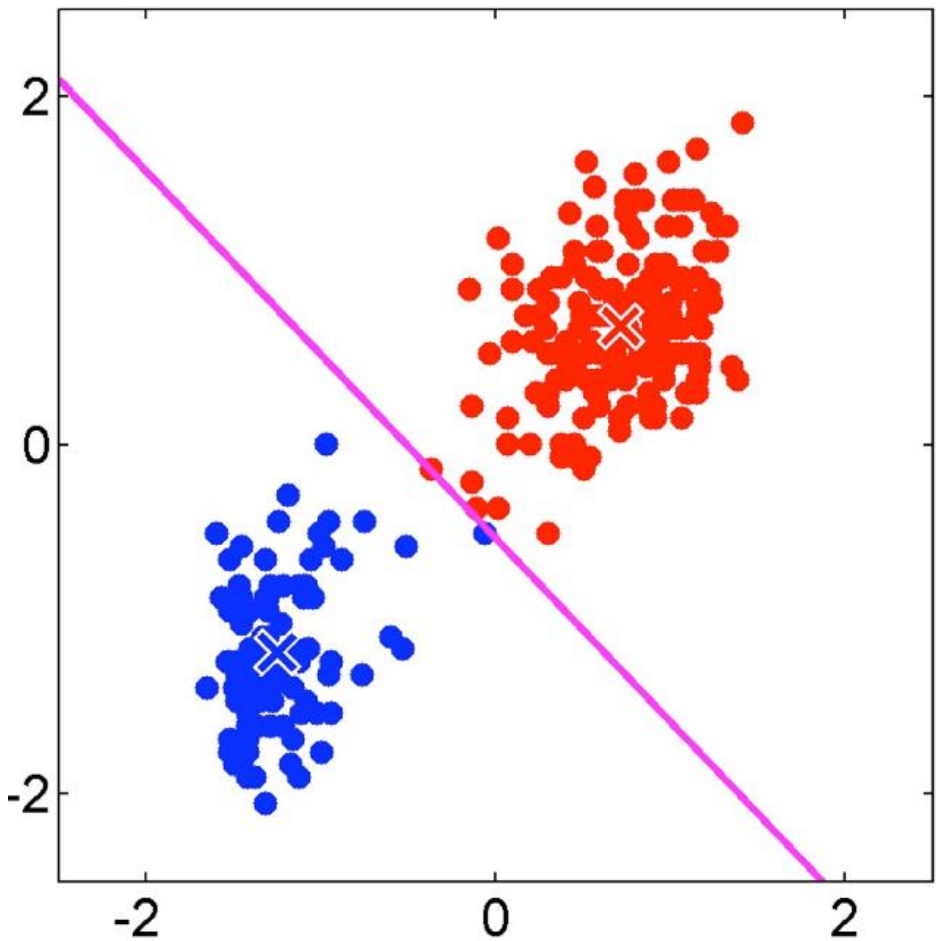
K-means example



K-means example



K-means example



Questions

- Why using those two steps can lead to converged result?
- Why always calculate mean vectors?
- What is the objective function of k-means clustering algorithm?
  - Objective function will give a measurement of the goodness of clustering results

K-means algorithm: revisit

- K clusters each summarized by a prototype μ_k.
- Assignment of data x_i to a cluster represented by responsibilities $r_{ik} \in \{0, 1\}$ with $\sum_{k=1}^{K} r_{ik} = 1$.
- An example with 4 data points and 3 clusters.

$$(r_{ik}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Loss function $J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2$.

K-means loss function

- Loss function $J = \sum_{i=1}^{n} \boxed{\sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2}.$

Variables: r_{ik} and μ_k

r_{ik} : Each point will be assigned to one of the prototype, the distance between the point to the prototype is the cost of assignment. We are seeking the optimal assignment that can minimize the cost

- Loss function $J = \boxed{\sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2}.$

μ_k : We are also seeking the optimal prototypes that can minimize the total cost

K-means algorithm

- E-step: Fix μ_k, minimize J w.r.t. r_{ik}.
  - Assign each data point to its nearest prototype.
- M-step: Fix r_{ik}, minimize J w.r.t. μ_k.
  - Set each prototype to the mean of the points in that cluster, i.e., $\mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$.
- This procedure is guaranteed to converge.
- Converges to a local minimum.

- Loss function $J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2$.

E-step: Fix μ_k solve r_{ik}

Equivalent to solve $\min_{r_{ik}} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$

$$r_{ik} = \begin{cases} 1 & \text{if } \|\mathbf{x}_i - \mu_k\|_2^2 \text{ is the smallest} \\ 0 & \text{otherwise} \end{cases}$$

We also know that

$$J_E^{t+1} \leq J^t$$

After the E-step, the objective value will decrease (at least not increase)

- Loss function $J = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2$.

M-step: Fix r_{ik} solve μ_k

Equivalent to solve $\min_{\mu_k} \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|\mathbf{x}_i - \mu_k\|_2^2$

Solve $\frac{\partial J}{\mu_k} = \sum_{i=1}^{N} r_{ik}(\mathbf{x}_i - \mu_k) = 0$

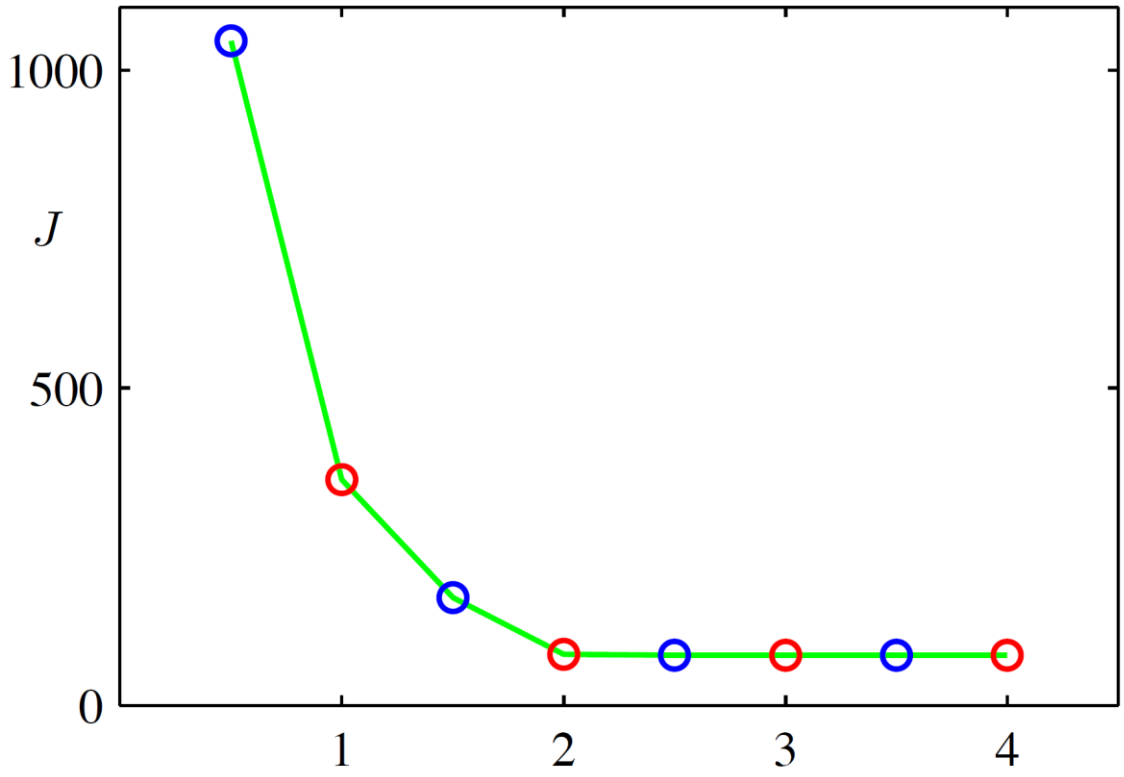
$$\mu_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}$$

Thus, we also know that

$$J^{t+1} \leq J_E^{t+1} \leq J^t$$

After the M-step, the objective value will decrease (at least not increase)

K-means example: loss function



The loss function will monotonically decrease

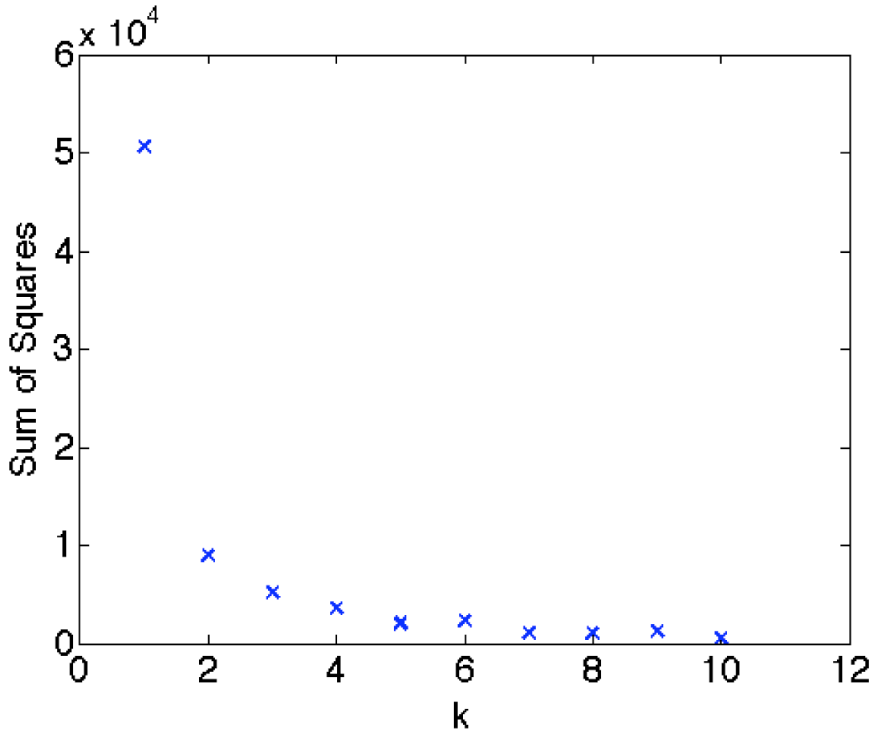
K-means algorithm

- Converges to a local minimum.
  - Use different initializations and pick the best solution.
  - May still be insufficient for large search spaces.
  - Other ways include a split-merge approach.

- Some heuristics
  - Randomly pick K data points as prototypes.
  - Pick prototype $i+1$ to be farthest from prototypes $\{1, \ldots, i\}$.

How to choose K?

- The loss function J generally decreases with K.



How to choose K?

- Unlike in supervised learning, we can evaluate validation accuracy, for unsupervised learning we can use different ways of evaluate the learning quality
- Cross-validation: Partition data into two sets. Estimate prototypes on one and use these to compute the loss function on the other.
  - Choose K leads to the smallest loss value in the validation set

Limitations of k-means

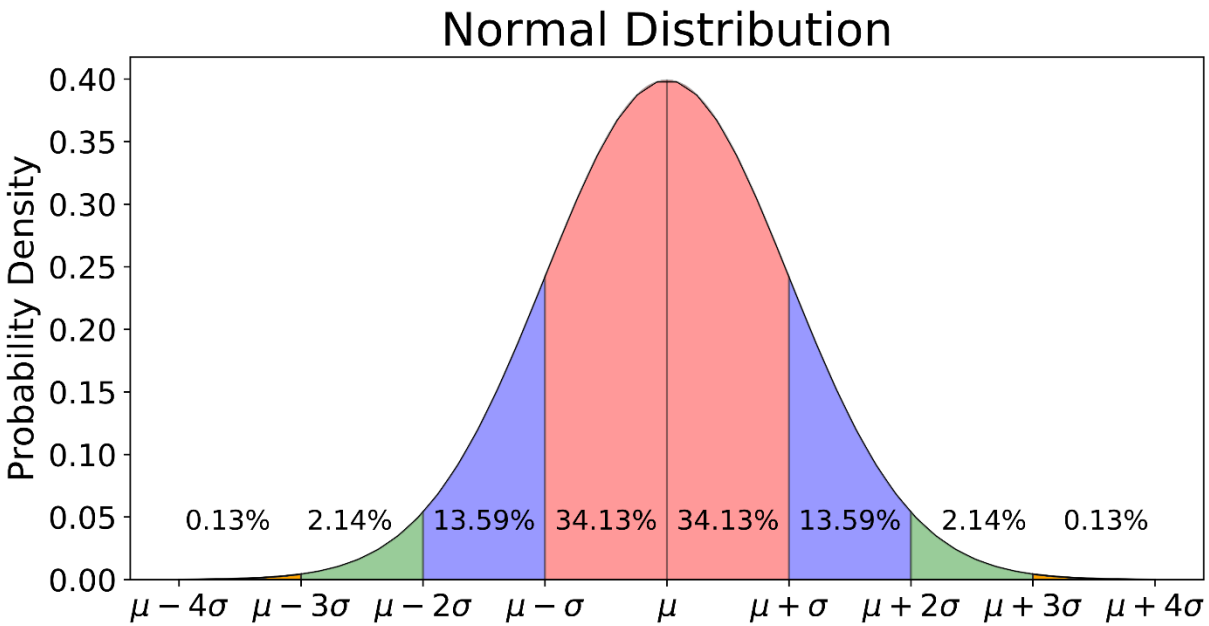
- Hard assignments of data points to clusters can cause a small perturbation to a data point to flip it to another cluster
- Assumes spherical clusters and equal probabilities for each cluster
- Those limitations can be solved by GMM based clustering

Outlines

- Overview of Unsupervised Learning and Clustering
- K-means clustering
- Gaussian mixture models (GMM)
  - Distribution modeling
  - GMM
  - Latent variable
  - EM algorithm

Distribution Modeling

- Distribution Modeling
  - Find a way to characterize the distribution of data
  - Modelling $P(x|\lambda)$
  - Example:



Distribution Modeling

- What is the form of $P(x|\lambda)$?
  - The unknow model parameters
- The function family
  - Based on assumptions
  - Based on prior knowledge about the data
  - Model it as a general function: parametric or nonparametric
- Once the function form is known, distribution modeling boils down to a parameter estimation problem
  - MLE: maximal likelihood estimation

$$\lambda = argmax_\lambda \log P(X|\lambda) = argmax_\lambda \sum_i \log P(\mathbf{x}_i|\lambda)$$

Assume samples are independently identical distributed (i.i.d.)

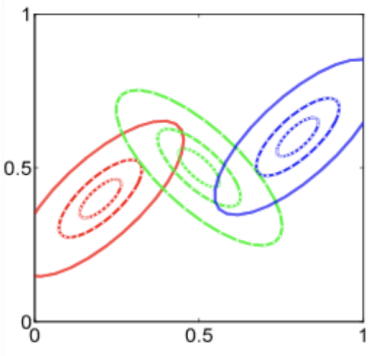
Commonly used assumption

- Multivariate Gaussian distribution

$$P(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) = \frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)\right)}{\sqrt{(2\pi)^k |\Sigma|}}$$

- Gaussian Mixture model

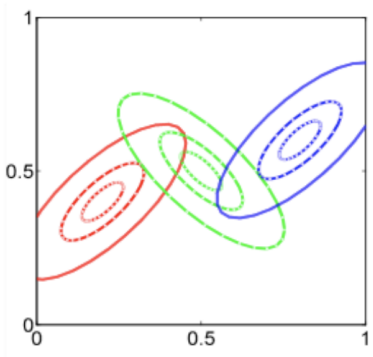
$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$



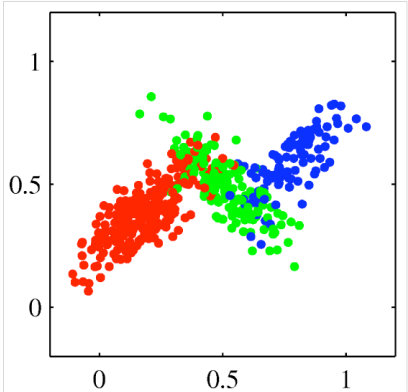
Gaussian Mixture Model (GMM)

- Likelihood $\Pr(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$ where

$$\sum_{k=1}^{K} \pi_k = 1, 0 \leq \pi_k \leq 1.$$



→



GMM can characterize the distribution that contains K clusters

What is relationship between GMM PDF and clusters of data?

Warmup practice

- Two bags:
  - Bag A: 60 black balls, 40 white balls
  - Bag B: 50 black balls, 50 white balls
  - Chance to select Bag A: 0.6 Bag B: 0.4
  - What's the probability of selecting a black ball (and a white ball) from this process?

Gaussian Mixture Model (GMM)

- Generative process:

$\{\pi_k, \mu_k, \Sigma_k\}$

1. Randomly select the k-th Gaussian distribution according to π_k
2. Randomly sample x from the selected Gaussian distribution

Imagine this process as a piece of program. Once written, it can generate a sample when we run it. The output x will be a random variable.

Gaussian Mixture Model (GMM)

- Generative process:

The distribution of x?

1. Randomly select the k-th Gaussian distribution according to π_k
2. Randomly sample x from the selected Gaussian distribution

$$P(\mathbf{x}) = P(\mathbf{x}|r=1)P(r=1) + P(\mathbf{x}|r=2)P(r=2) + \cdots$$

Let

$$P(r=k) = \pi_k$$
$$P(\mathbf{x}|r=k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$\Rightarrow \quad P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

Gaussian Mixture Model (GMM)

- Generative process:

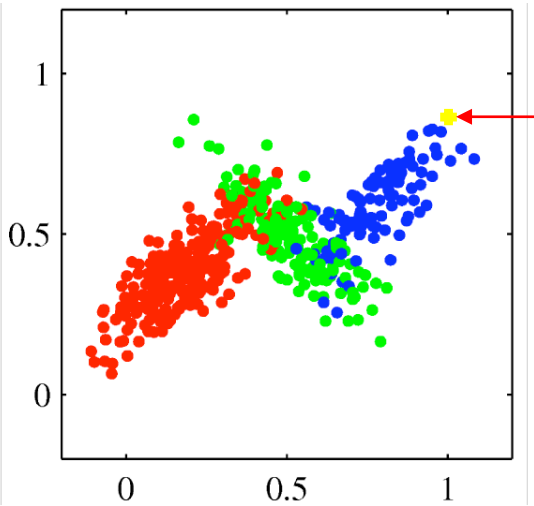
$\{\pi_k, \mu_k, \Sigma_k\}$

1. Randomly select the k-th Gaussian distribution according to π_k
2. Randomly sample x from the selected Gaussian distribution

The selection made inside the generative process, r is an internal variable, hidden from us. We call it latent variable

Latent variable

- Intermediate results inside a generative process
- Each sample is associated with a latent variable
- We do not know its exact value
- But we can infer how likely it can be
- Example



We may know this point is more likely to belong to the blue cluster

Latent variable and Inference

- Given an observation, we can estimate the likelihood of the latent variable. In the case of GMM, we try to estimate $P(r|\mathbf{x})$
- Because the latent variable in GMM indicates the membership of a sample belonging to a cluster. $P(r|\mathbf{x})$ can be seen as a soft-membership (soft-clustering)

Latent variable and Inference

- Given an observation, we can estimate the likelihood of the latent variable. In the case of GMM, we try to estimate $P(r|\mathbf{x})$

- The difference between $P(r|\mathbf{x})$ and $P(r) = \pi_k$

  $P(r|\mathbf{x})$ Posterior probability. The likelihood about c after x is observed

  $P(r)$ Prior probability. The likelihood before any observation

- The process of calculate $P(c|\mathbf{x})$ or the most likely r is called Inference

Inference in GMM

- Inference can be done by using Bayes theory

$$P(r|\mathbf{x}) = \frac{P(\mathbf{x}|r)P(r)}{P(\mathbf{x})}$$

$$P(r = k) = \pi_k$$

$$P(\mathbf{x}|r = k) = \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$P(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

$$P(r = k|\mathbf{x}) = \frac{\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)\pi_j}$$

Parameter estimation for GMM

- Use MLE (maximal likelihood estimation)

$$
\begin{aligned}
\mathcal{L} &= -\log P(X) \\
&= -\sum_{i=1}^{N} \log P(x_i) \\
&= -\sum_{i=1}^{N} \log\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)\}
\end{aligned}
$$

- Unfortunately, it is difficult to solve $\frac{\partial \mathcal{L}}{\partial \theta} = 0$

EM Algorithm solution

- What If we know the latent variable for each sample, say, r_i?
  - Let's use one hot encoding $\mathbf{r}_i$ (same as the case in k-means) to represent the selection result, e.g.,$r_{ik} = 1$, if select k-th Gaussian for i-th sample
  - Then the parameter estimation problem becomes easy because we know which sample is assigned to which Guassian
  - Solution:

$$\pi_k = \frac{\sum_i r_{ik}}{N},\ \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}},\ \Sigma_k = \frac{\sum_i r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r_{ik}}$$

EM Algorithm solution

- However, we do not know the assignment but only know its probability

$$P(r_{ik} = 1|\mathbf{x}) = \frac{\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{x}|\mu_j, \Sigma_j)\pi_j}$$

- We can actually use the expectation of r_{ik} instead

$$E(r_{ik}|\mathbf{x}_i) = 1 \times P(r_{ik} = 1|\mathbf{x}_i) + 0 \times P(r_{ik} = 0|\mathbf{x}_i) = P(r_{ik} = 1|\mathbf{x}_i)$$

EM algorithm solution to GMM

- E-step: calculate the posterior probability about the latent variable:

$$r'_{ik} = P(r_{ik} = 1|\mathbf{x}_i) = \frac{\mathcal{N}(\mathbf{x}_i|\mu_k, \Sigma_k)\pi_k}{\sum_{j=1}^{K} \mathcal{N}(\mathbf{x}_i|\mu_j, \Sigma_j)\pi_j}$$

- M-step: estimate parameters based the expectation of latent variables

$$\pi_k = \frac{\sum_i r'_{ik}}{N}, \ \mu_k = \frac{\sum_i r'_{ik} x_i}{\sum_i r'_{ik}}, \ \Sigma_k = \frac{\sum_i r'_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r'_{ik}}$$

EM algorithm (More general case)

- The above solution represents a special case of a more general method called EM-algorithm
- It is widely used for learning the probabilistic models with latent variable inside its generative process.
- It iterates between a E-step and a M-step.
  - We can theoretically prove that each step will lead to a non-increase of loss function
  - The iterative process will converge to a local minimum

EM algorithm in more general form

EM algorithm iterates between the following two steps:

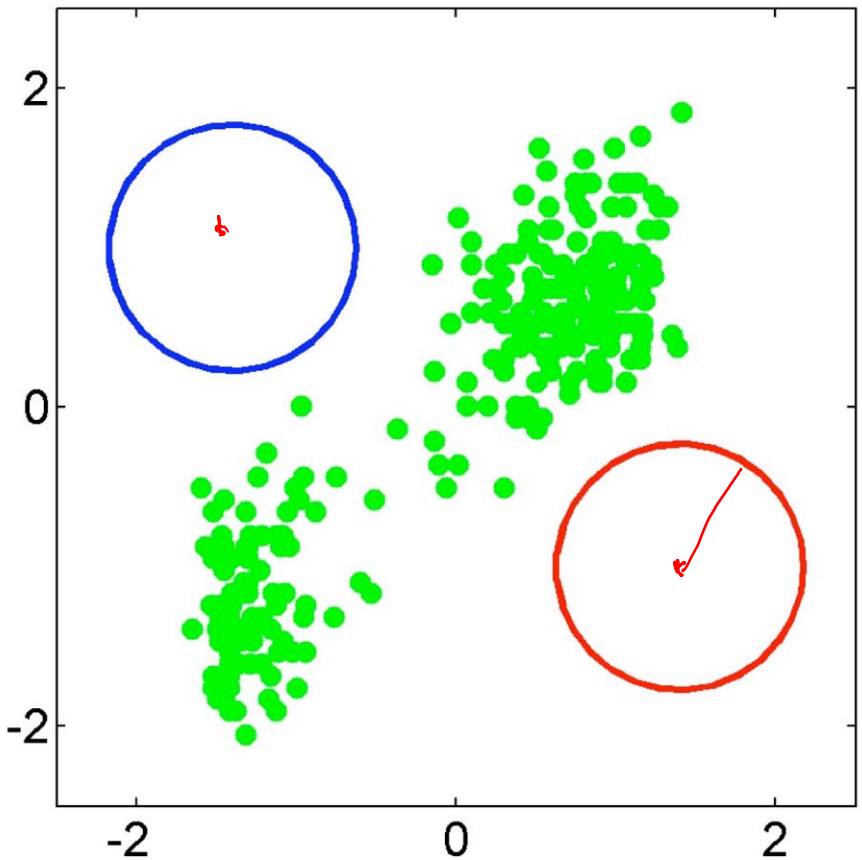
- **E-step:** calculate the posterior probability of latent variable $P(z|x, \theta^t)$ given current model parameter θ^t.
- **M-step:** Update the model parameter based on the expected log likelihood. This is done by maximizing

  $\theta^{t+1} = argmax_\theta \, E_{z|x,\theta} \log P(x, z|\theta) = \int P(z|x, \theta^t) \log(P(x|\theta, z)P(z|\theta))dz$

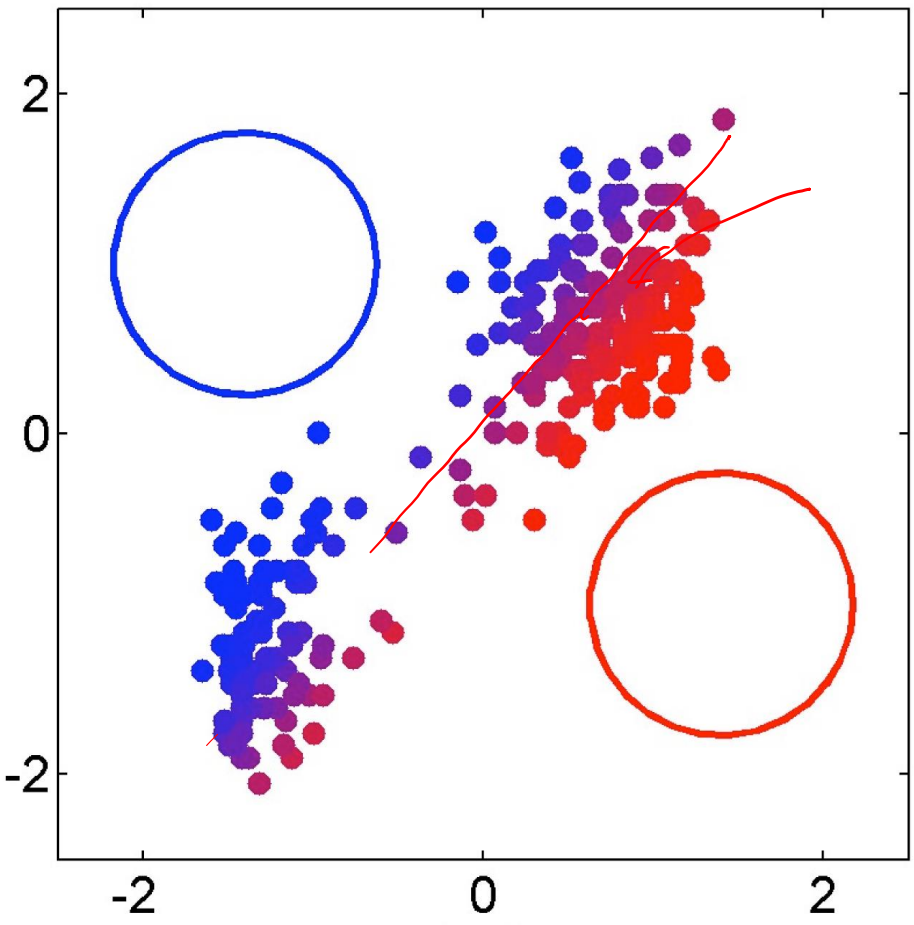
Connection to K-means

- E-step in GMM a soft version of K-means. $r_{ik} \in [0, 1]$ instead of $\{0, 1\}$.
- M-step in GMM estimates the probabilities and the covariance matrix of each cluster in addition to the means.
- All π_k are equal. $\Sigma_k = \delta^2 I$. As $\delta^2 \to 0$, $r_{ik} \to \{0, 1\}$, and the two methods coincide.

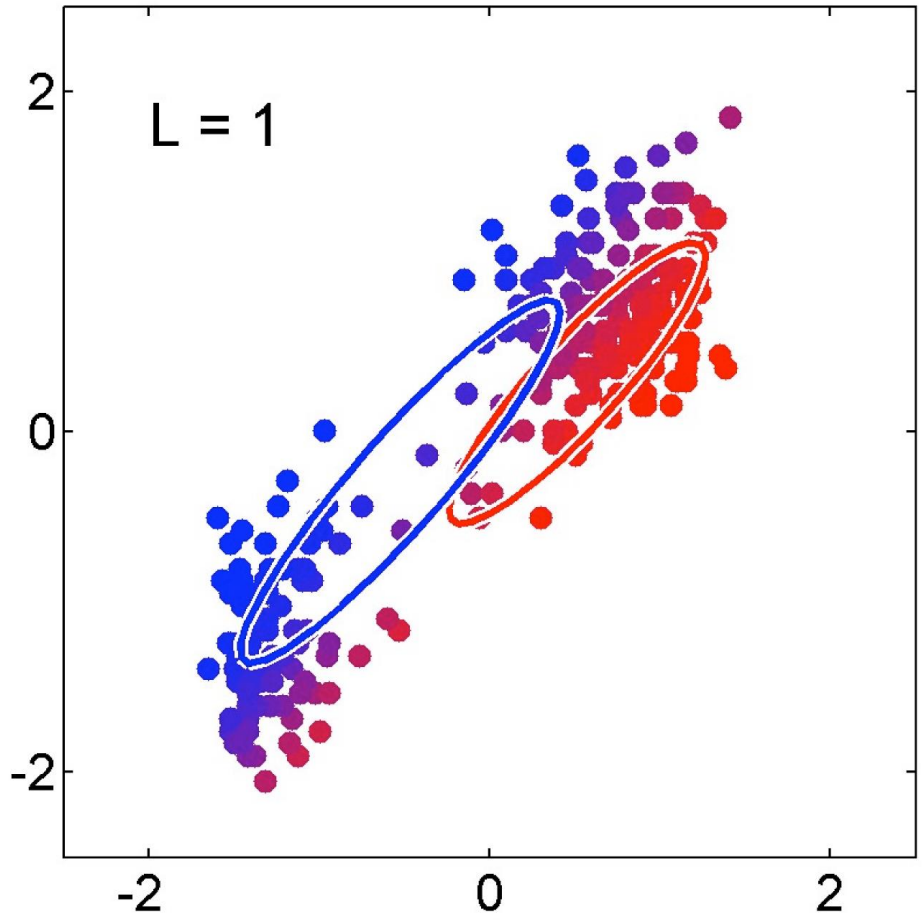
GMM example



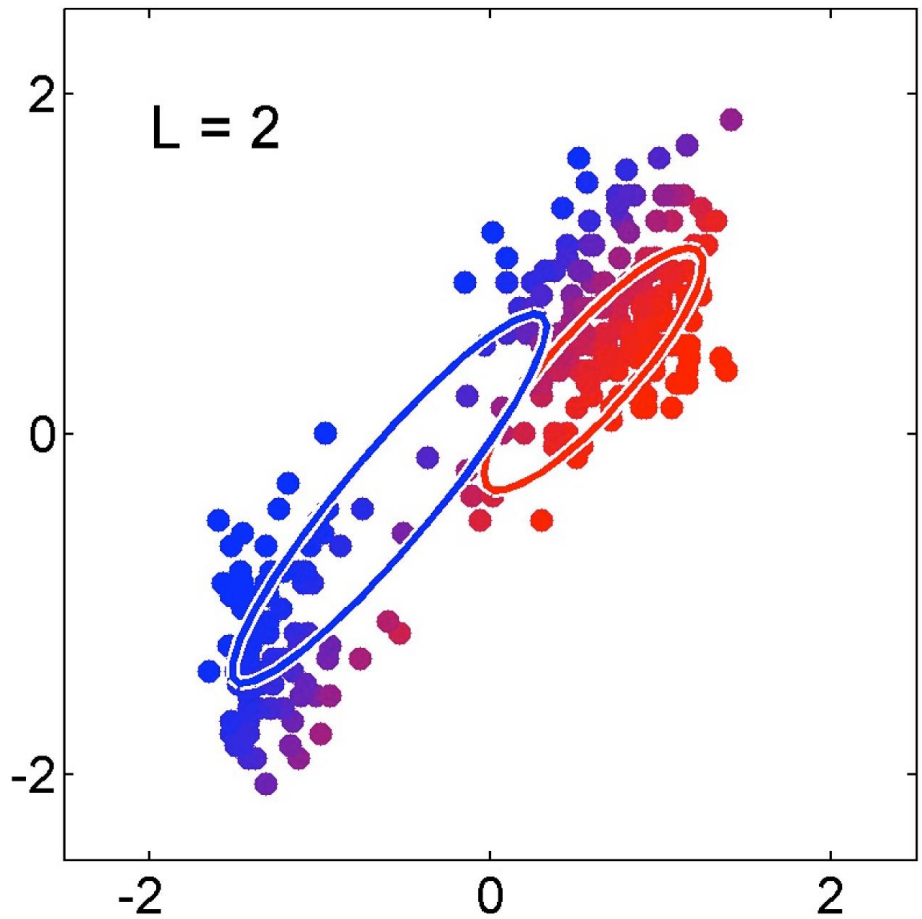
GMM example



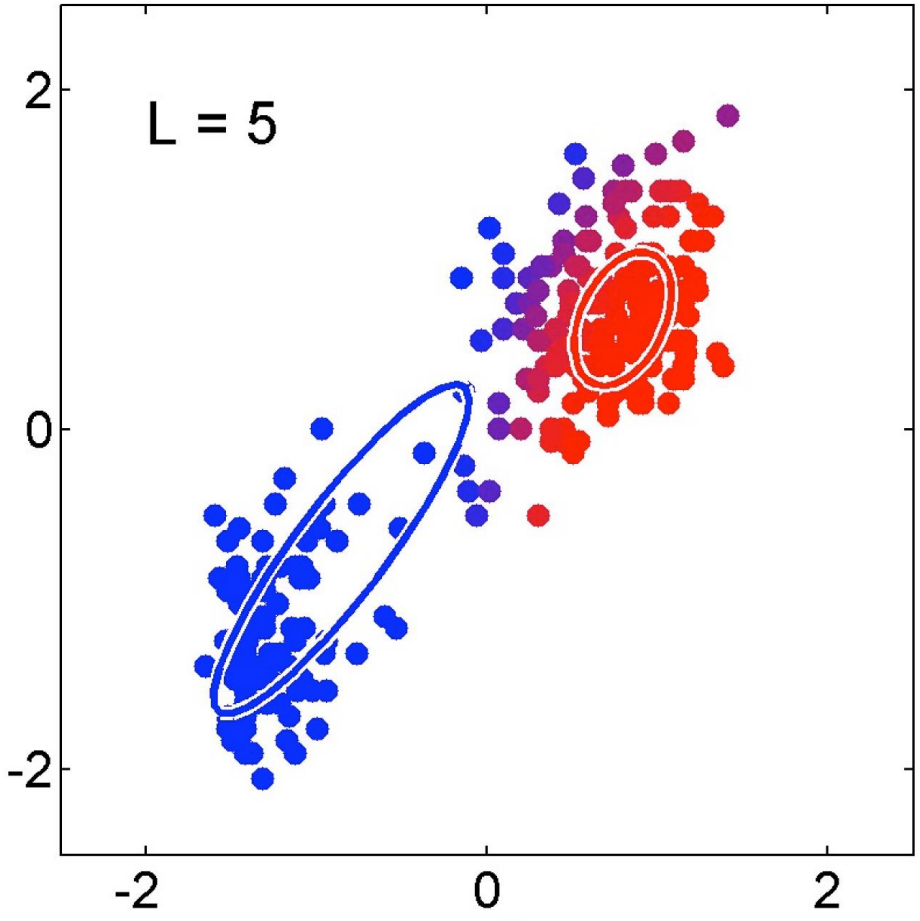
GMM example



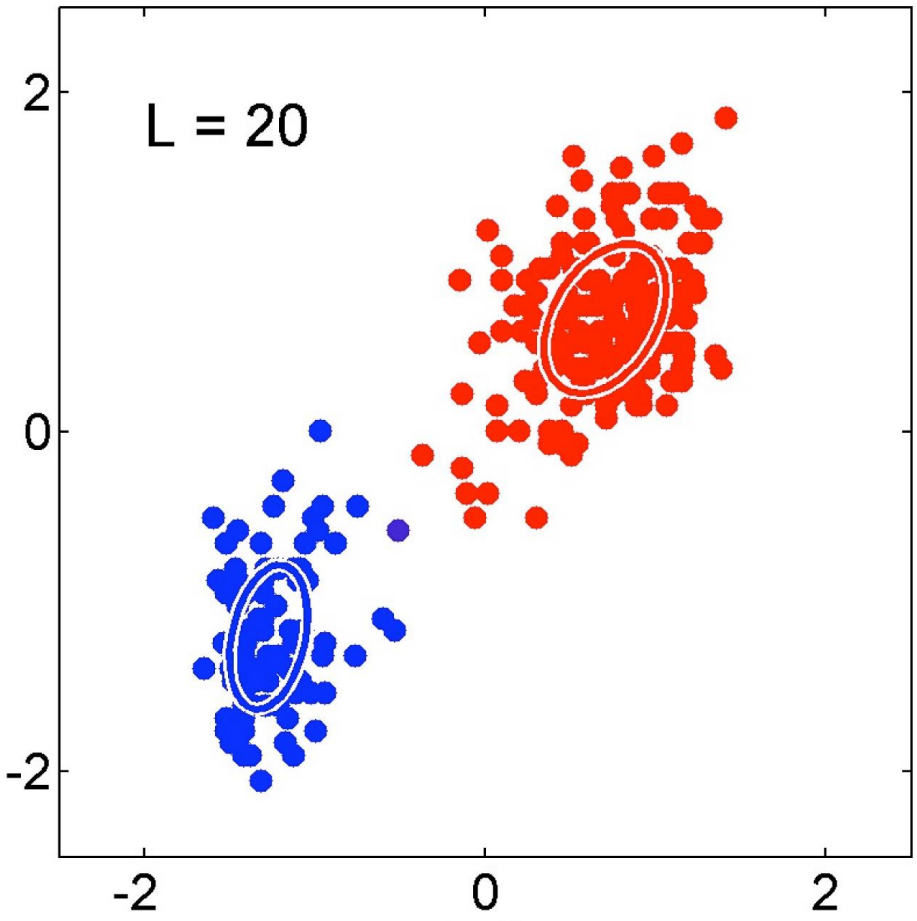
GMM example



GMM example



GMM example



Summary

- Unsupervised learning and clustering
- Clustering
  - Grouping data
  - Estimating a mixture model
- K-means clustering
  - How to do that?
  - The limitation
- GMM model
  - Connection between a distribution and group assignment: a generative process
  - Parameter estimation: E-M algorithm