CS 559 Machine Learning

Lecture 9: Clustering and EM

Ping Wang
Department of Computer Science
Stevens Institute of Technology

Supervised and Unsupervised Learning

- **Supervised learning** approaches that have covered:
    - Linear Model: linear regression, linear classification (logistic regression, perceptron algorithm, Fisher Discriminant Analysis, etc)
    - Non-parametric: Parzen window, KNN.
    - Non-linear: SVM, Decision Trees, Ensemble Methods, etc.

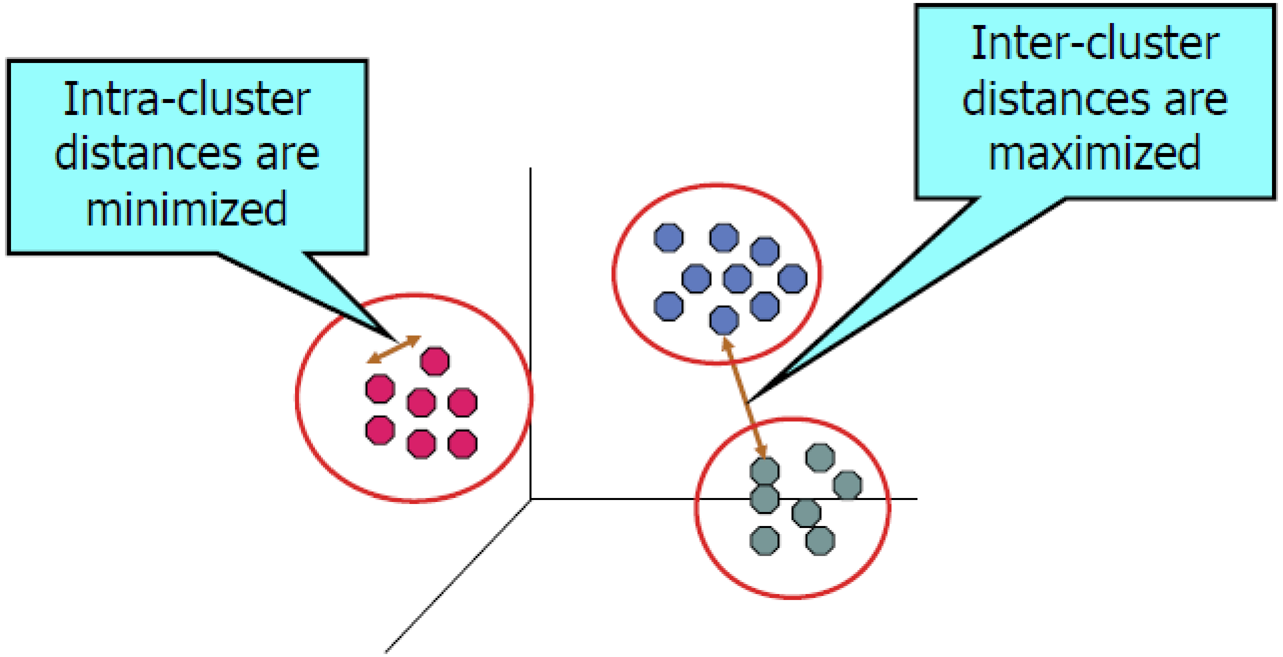
- **Unsupervised learning** approaches that have been discussed:
    - Density estimation: Parzen window, KNN.
    - Clustering: K-means, GMM (this lecture).

Today’s Lecture

- Clustering
- K-means Algorithm
- Gaussian Mixture Model (GMM)
- Expectation-Maximization (EM) algorithm for GMM

What is Clustering?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.



Intra-cluster distances are minimized
Inter-cluster distances are maximized

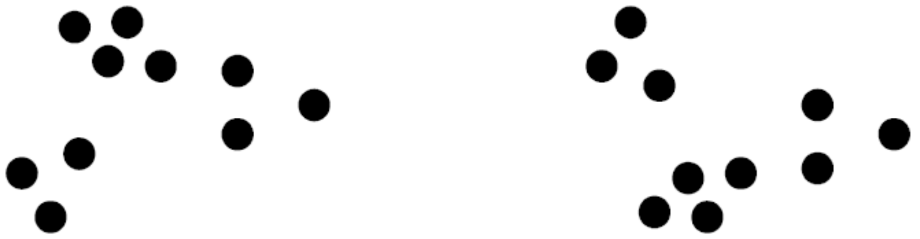
Applications of Clustering

- Group related documents for browsing
- Group genes and proteins that have similar functionality
- Group stocks with similar price fluctuations
- Group the regions with similar temperature
- And many others…

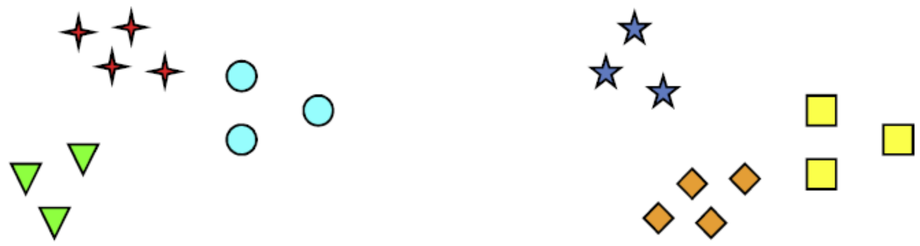
What is Not Clustering?

- **Simple segmentation**
  - Dividing students into different registration groups alphabetically, by last name.
- **Results of a query**
  - Groupings are a result of an external specific query conditions.
  - Clustering is a grouping of objects based on the data.
- **Supervised classification**
  - Have class label information

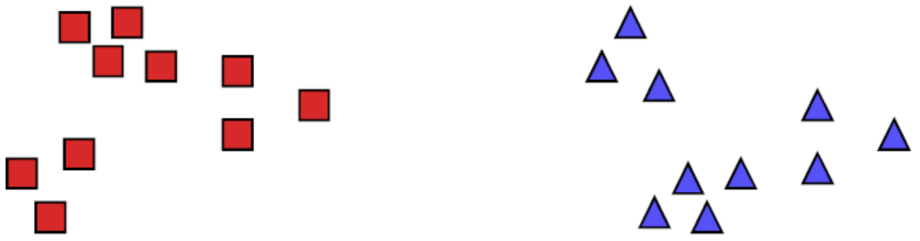
Notion of a Cluster can be Ambiguous



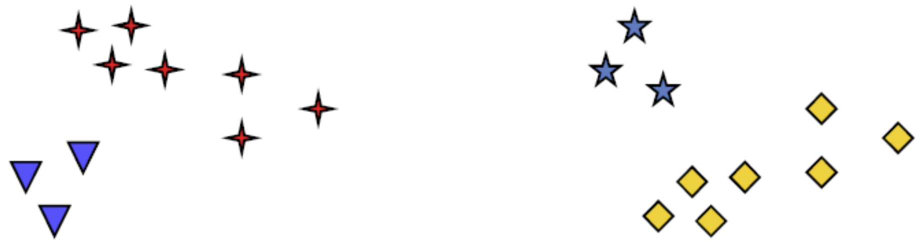
How many clusters?



Six Clusters



Two Clusters



Four Clusters

Distance Based Clustering

- Fundamental to all clustering techniques is the choice of distance or dissimilarity measure between two objects.
- **What is Similarity?:** The quality or state of being similar; likeness; resemblance; as, a similarity of features - Websters' dictionary.
- Similarity is hard to define, but we know it when we see it.

Dissimilarities Based on Features

- Assuming $x^i = \left(x_1^i, x_2^i, \dots, x_d^i\right)^{\mathrm{T}} \in \mathbb{R}^d, i = 1, \dots, N$
- Distance (dissimilarity): $D\left(x^i, x^j\right) = \sum_{k=1}^{d} d_k(x_k^i, x_k^j)$
- Squared Euclidean distance:

$$d_k\left(x_k^i, x_k^j\right) = \left(x_k^i - x_k^j\right)^2$$

$$\Rightarrow D\left(x^i, x^j\right) = \sum_{k=1}^{d} \left(x_k^i - x_k^j\right)^2$$

Non-Probabilistic Algorithms

- These algorithms work directly on the observed data, without relying on a probability model about the data.
- Commonly used in data mining, since often the prior knowledge about the data generation is not available.

Non-Probabilistic Algorithms

- Assuming $x_i \in \mathbb{R}^d, i = 1, \dots, N$
- Pre-specified number of cluster $K, k \in \{1, \dots, K\}$
- Each data point x_i is assigned to **one, and only one cluster**.
- **Goal**: Partition the data into K clusters that achieves a required objective, defined by a dissimilarity function $D\left(x^i, x^j\right)$.
- Usually, the assignment of data to clusters is done by minimizing a “loss” function that measures the degrees to which the clustering goal is not met.

Non-probabilistic Algorithms

- Since the goal is to assign close points to the same cluster, a natural loss function would be within cluster scatter. Need to define the following two notations.
- Notations:
  - Binary indicator r_{nk}: (describe which of the K clusters the data point x_n is assigned to)

    $$r_{nk} = \begin{cases} 1, & if\ x_n\ is\ assigned\ to\ cluster\ k \\ 0, & otherwise \end{cases}$$

    E.g., if $K = 3$, x_n is assigned to cluster 2, then $r_{n1} = 0, r_{n2} = 1, r_{n3} = 0$. Known as 1-of-K coding scheme.
  - Prototype μ_k (centroid) for each cluster: $\mu_k \in \mathcal{R}^d$ and can be considered as centres of clusters.

Objective Function *J*

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} D(x_n, \mu_k)$$

- Also called distortion measure.
- Represents distances of each data point to the centre of its assigned cluster μ_k.
- Need to select $D(.,.)$, find r_{nk} and μ_k.

Objective of K-means Clustering

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} D(x_n, \mu_k)$$

- K-means is one of the most popular iterative descent clustering methods.
- Use squared Euclidean distance as $D(.,.)$:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

  - Represents the sum of square distance of each data point to its assigned vector μ_k
  - Features: quantitative type.
  - Goal is to find values for the r_{nk} and the μ_k that can minimize J.
  - Through iterative procedure...

K-means Algorithm

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

- Minimize J in an iterative procedure to find out r_{nk} and μ_k:
- First phase: cluster assignment step
  - Choose some initial values for the μ_k. Then minimize J with respect to the r_{nk}, keeping the μ_k fixed.
- Second phase: centroid update step
  - We minimize J with respect to the μ_k, keeping r_{nk} fixed.

K-means Algorithm: Cluster Assignment Step

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

- **First phase:** minimize J with respect to the r_{nk}, keeping the μ_k fixed.
    - When μ_k fixed, J becomes linear function of r_{nk}.
    - Terms involving different n are independent, optimize for each n separately.
    - Choosing r_{nk} to be 1 for the value of k that gives the minimum value of $\|x_n - \mu_k\|^2$ (i.e., assign x_n to the closest μ_k).

$$r_{nk} = \begin{cases} 1, & if\ k = \arg\min_j \left\|x_n - \mu_j\right\|^2 \\ 0, & otherwise \end{cases}$$

K-means Algorithm: Centroid Update Step

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|x_n - \mu_k\|^2$$

- **Second phase:** We minimize J with respect to the μ_k, keeping r_{nk} fixed.
  - J is quadratic function of μ_k.
  - Let $\frac{\partial J}{\partial \mu_k} = 0$:

$$2 \sum_{n=1}^{N} r_{nk}(x_n - \mu_k) = 0$$

$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

  - Mean of all of the data points x_n assigned to cluster k.

K-means Algorithm

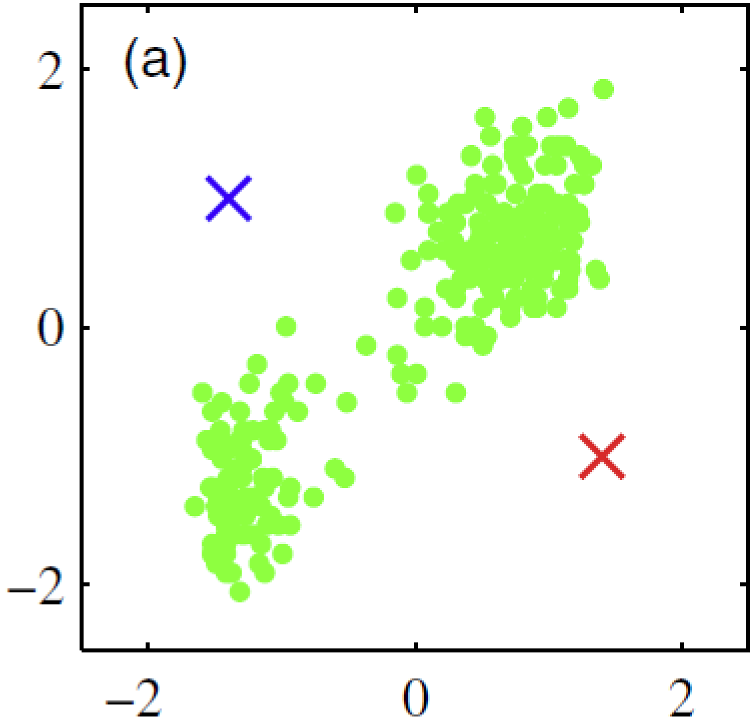
- Summary: initialize the μ_k, then iterate the following two steps until convergence.
  - Cluster Assignment Step: Minimize J with respect to the r_{nk}, keeping the μ_k fixed.

$$r_{nk} = \begin{cases} 1, & if\ k = \arg\min_j \left\| x_n - \mu_j \right\|^2 \\ 0, & otherwise \end{cases}$$

  - Centroid update step: Minimize J with respect to the μ_k, keeping r_{nk} fixed.

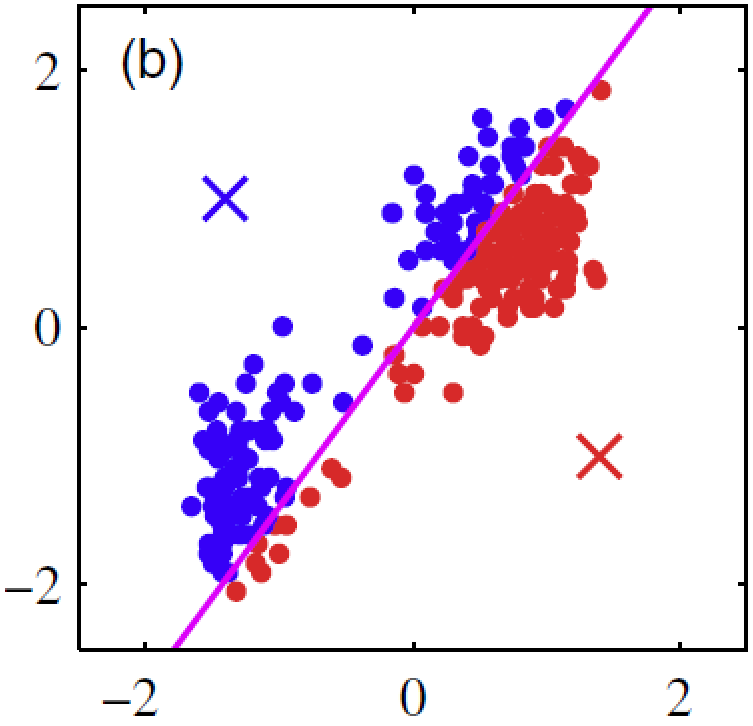
$$\mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$$

K-means: Example



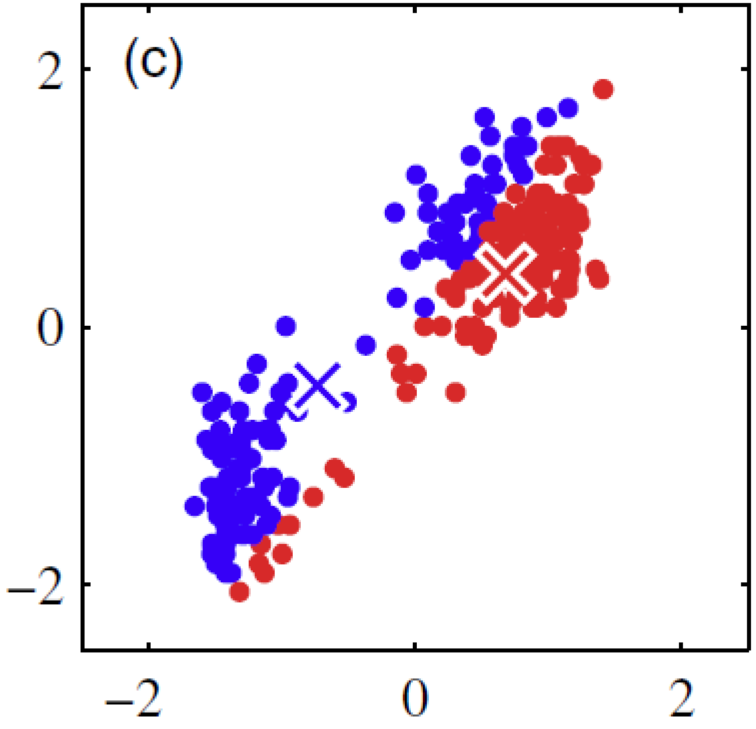
Initialization: two randomly selected centroid

K-means: Example



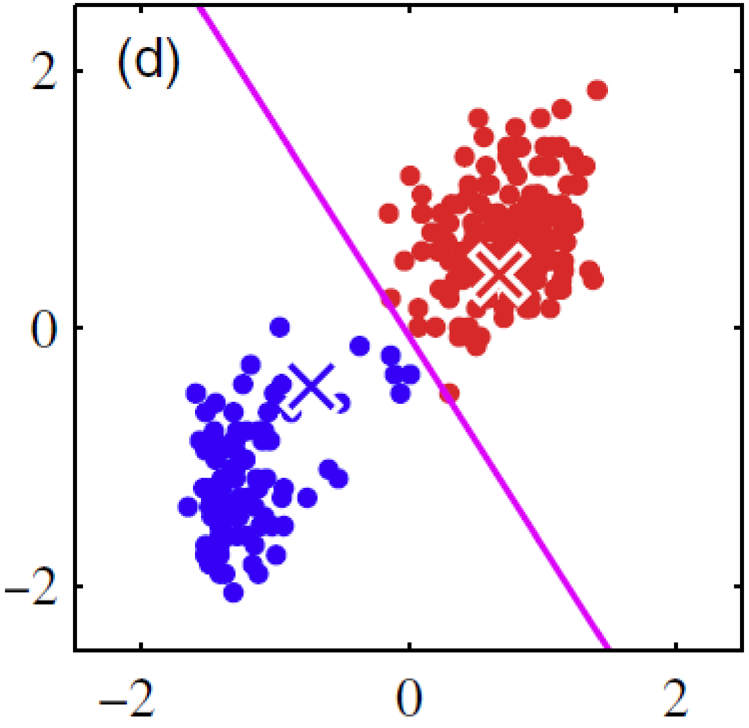
Step 1: assign clusters, determine r_{nk}.

K-means: Example



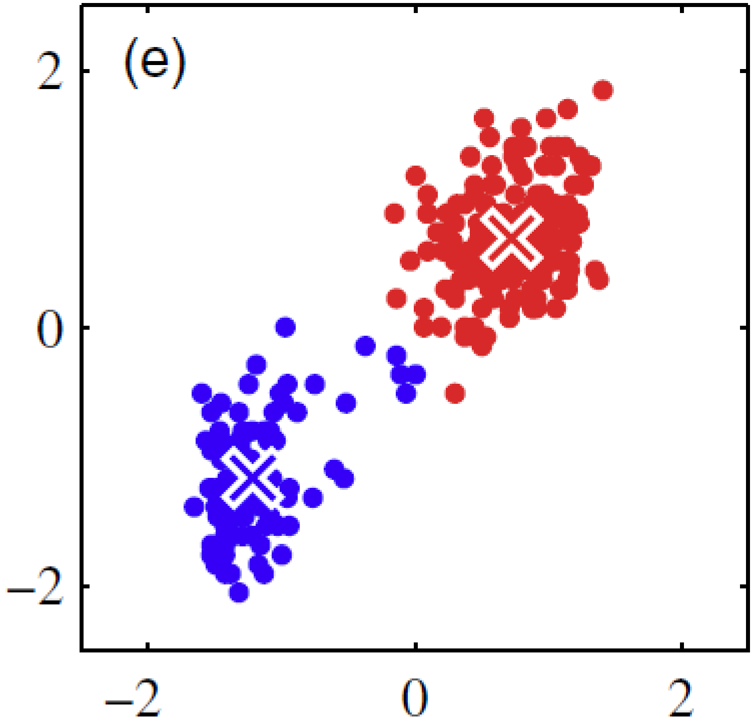
Step 1: recalculate means of clusters, determine μ_k.

K-means: Example



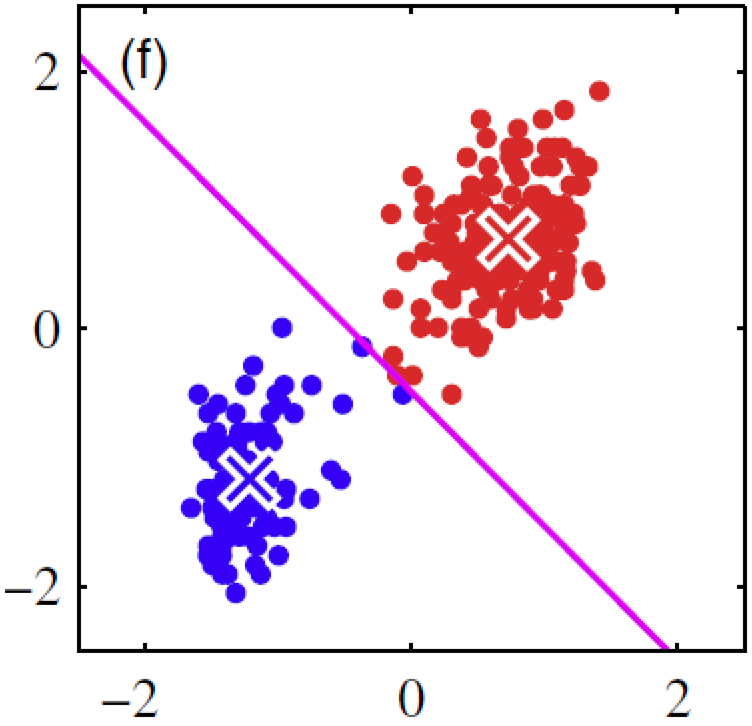
Step 2: assign clusters to new means, determine r_{nk}.

K-means: Example



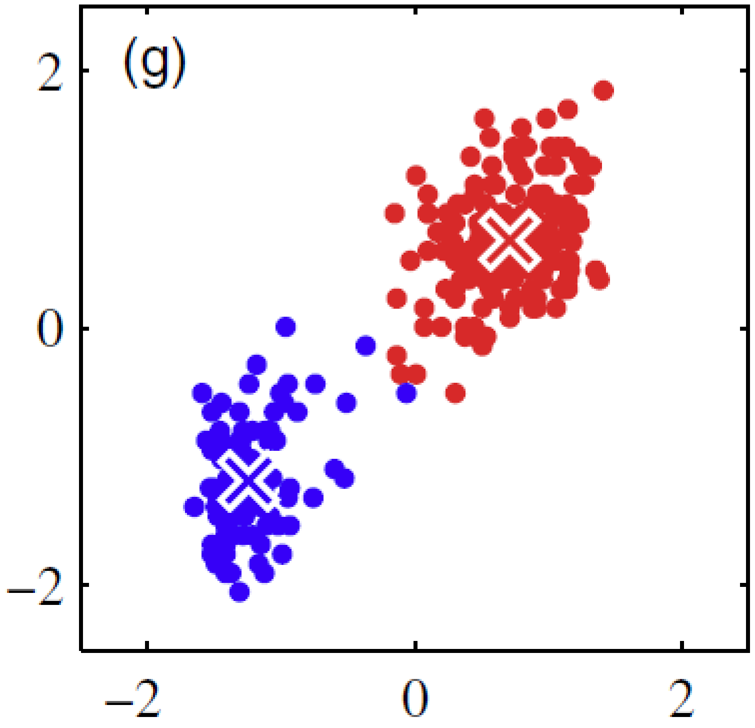
Step 2: recalculate means of clusters, determine μ_k.

K-means: Example



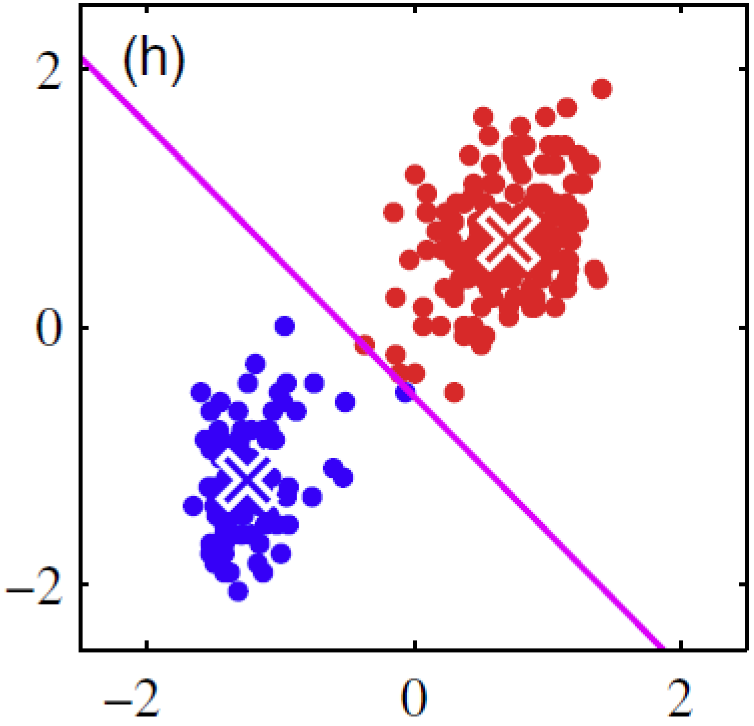
Step 3: assign clusters to new means, determine r_{nk}.

K-means: Example



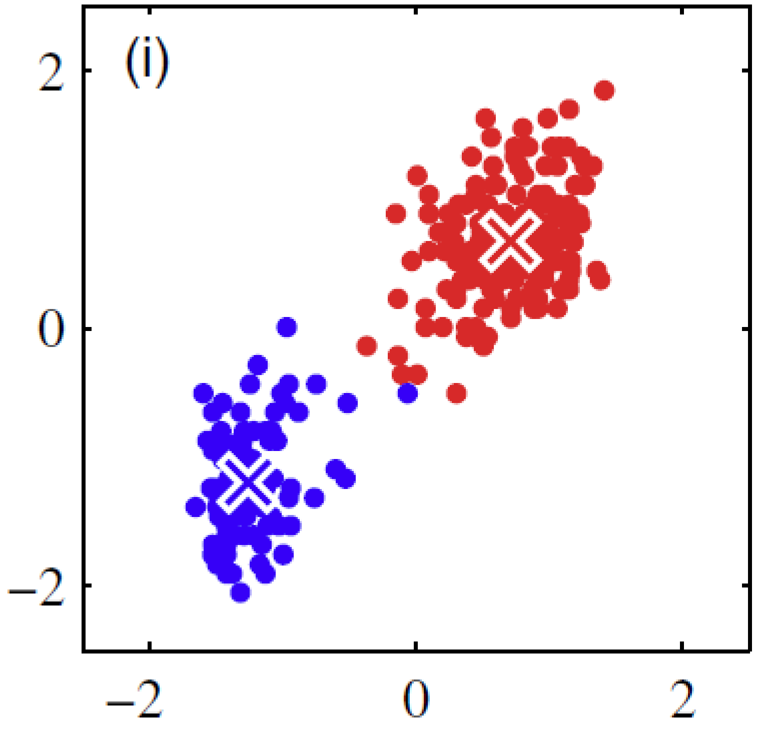
Step 3: recalculate means of clusters, determine μ_k.

K-means: Example



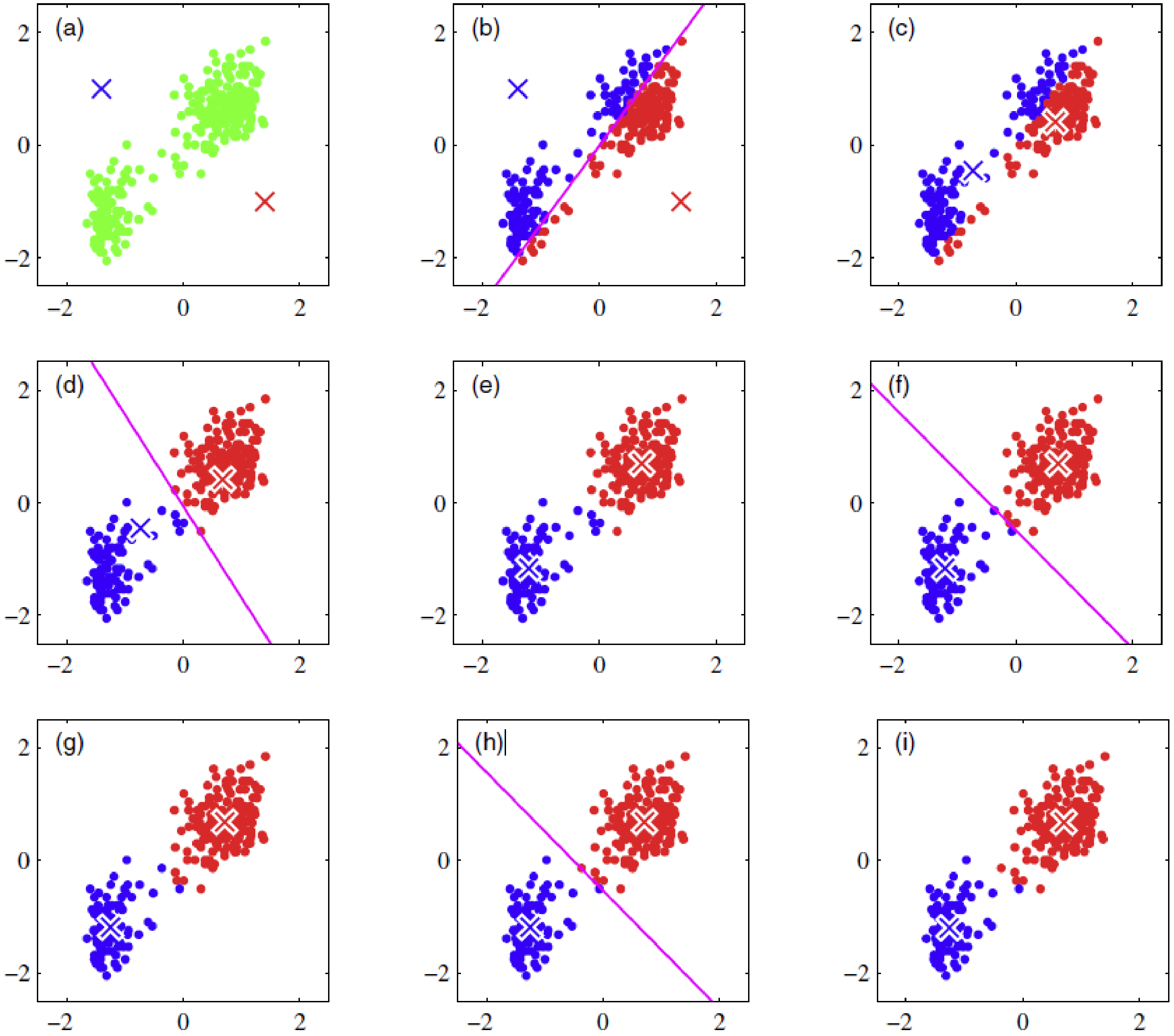
Step 4: assign clusters to new means, determine r_{nk}.

K-means: Example

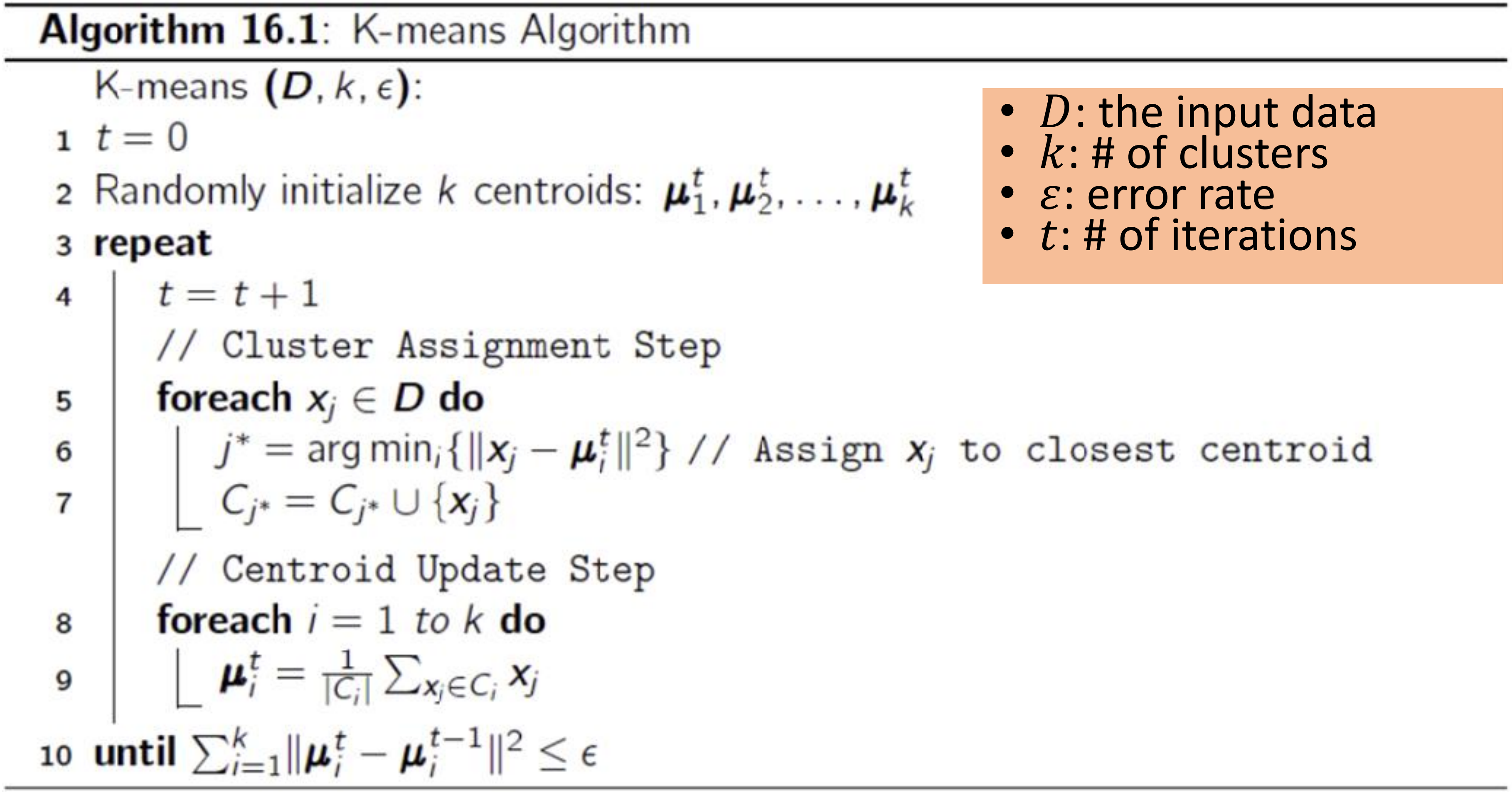


Step 4: recalculate means of clusters, determine μ_k. Converged.

K-means: Example



K-means Clustering: Pseudocode



Algorithm 16.1: K-means Algorithm

```
K-means $(\mathbf{D}, k, \epsilon)$:
$t = 0$
Randomly initialize $k$ centroids: $\boldsymbol{\mu}_1^t, \boldsymbol{\mu}_2^t, \ldots, \boldsymbol{\mu}_k^t$
repeat
    $t = t + 1$
    // Cluster Assignment Step
    foreach $\mathbf{x}_j \in \mathbf{D}$ do
        $j^* = \arg\min_i \{\|\mathbf{x}_j - \boldsymbol{\mu}_i^t\|^2\}$ // Assign $\mathbf{x}_j$ to closest centroid
        $C_{j^*} = C_{j^*} \cup \{\mathbf{x}_j\}$
    // Centroid Update Step
    foreach $i = 1$ to $k$ do
        $\boldsymbol{\mu}_i^t = \frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j$
until $\sum_{i=1}^{k} \|\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_i^{t-1}\|^2 \leq \epsilon$
```

- D: the input data
- k: # of clusters
- ε: error rate
- t: # of iterations

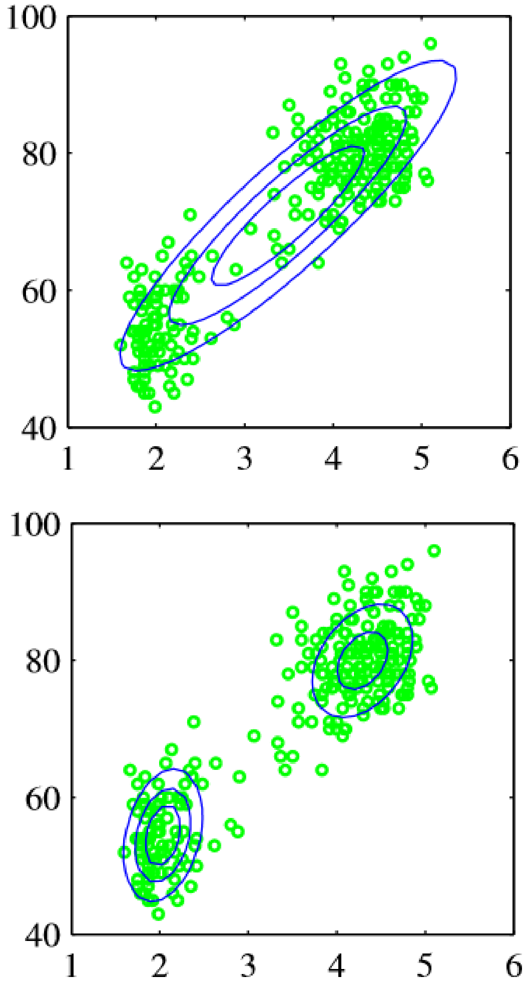
K-means: Properties and Limitations

- The algorithm converges to a local minimum.
- The solution depends on the initial values.
- One should start the algorithm with many different random choices for the initial means, and choose the solution having smallest value of the objective function
- The number of clusters K need to be pre-specified, but can be estimated from the data.
- The algorithm is sensitive to outliers.
- Every data point is assigned to one, and only one of the clusters. (hard assignment, no probability involved to reflect the level of uncertainty)
- How about probabilistic approach that obtain the soft assignment of data points?

Gaussian Mixture Model and EM

Motivation of Mixture of Gaussians

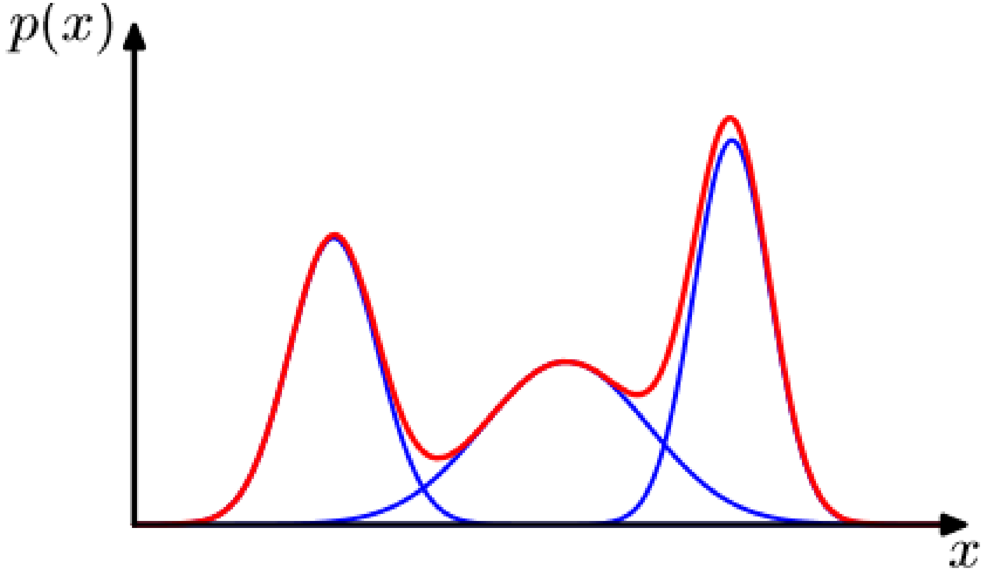
- Gaussian distribution has some important analytical properties.
- It also suffers from significant limitations and single Gaussian may not be enough.
- A single Gaussian distribution fails to capture the two clumps in the data and indeed places much of its probability mass in the central region between the clumps where the data are relatively sparse.
- A linear combination of two Gaussians gives a better representation of the data.



The blue curves show contours of constant probability density. Figure 2.21 in PRML.

Motivation of Mixture of Gaussians

- Mixture of Gaussian distributions (Linear combination of Gaussians) can represent very complex densities.



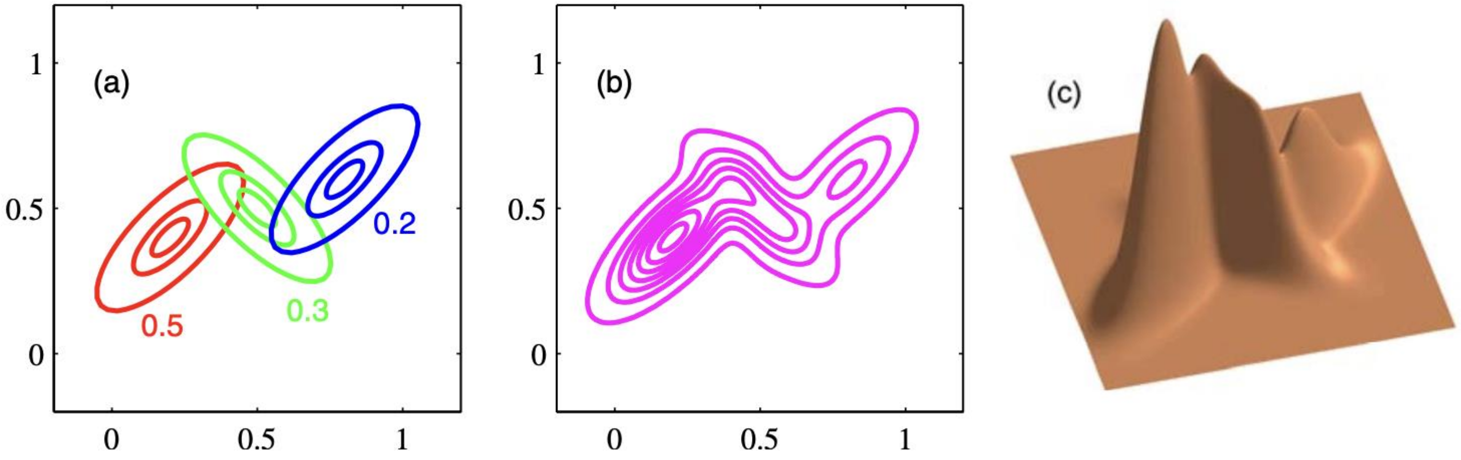
Example of a Gaussian mixture in one dimension showing 3 Gaussians (in blue) and their sum in red

Mixture of Gaussians

- We therefore consider a superposition of K Gaussian densities of the following form:

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

- Each Gaussian density $\mathcal{N}(x|\mu_k, \Sigma_k)$ is called a component of the mixture and has its own mean μ_k and covariance Σ_k
- Mixing coefficients: $0 \leq \pi_k \leq 1, \sum_{k=1}^{K} \pi_k = 1$



a) Contours of constant density for each of the mixture components
b) Contours of the marginal probability density p(x) of the mixture distribution
c) A surface plot of the distribution p(x).

Mixture of Gaussians

- Given N data points, our goal is to estimate the parameters that maximizes the likelihood function.
- Maximize the log of the likelihood function:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)\}$$

- Difficulty:
  - No closed-form solution
  - The likelihood is highly non-linear

Chicken-and-Egg Problem

- **Problem:**
  - If we know an estimation for the parameters, then we can estimate the cluster of x.
  - If we know the cluster that the data point belongs to, then we can estimate the parameters.
  - We don’t know either of the above.
- **Solution:**
  - Introducing latent variables about the cluster assignments.

Mixture of Gaussians: Introducing Latent Variables

- We introduce a K-dimensional binary random variable z having a 1-of-K representation in which a particular element $z_k = 1$ and all other elements are equal to 0.
- Thus, $z_k \in \{0,1\}, \sum_k z_k = 1$.
- The marginal distribution over z is specified in terms of the mixing coefficients:

$$p(z_k = 1) = \pi_k$$

- Because z uses a 1-of-K representation, we can write the distribution in this form:

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Mixtures of Gaussians: Conditional Distribution

- Similarly, the conditional distribution of x given a particular value for z is a Gaussian:

$$p(x|z_k = 1) = \mathcal{N}(x|\mu_k, \Sigma_k)$$

which can be written as:

$$p(x|z) = \prod_{k=1}^{K} \mathcal{N}(x|\mu_k, \Sigma_k)^{z_k}$$

- The marginal distribution of x is then obtained by summing the joint distribution over all possible values of z:

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

Mixtures of Gaussians

- The conditional probability of z given x_n can be obtained using Bayes’ theorem:

$$\gamma(z_{nk}) = p(z_k = 1|x_n) = \frac{p(z_k = 1)p(x_n|z_k = 1)}{\sum_j p\big(z_j = 1\big)p(x_n|z_j = 1)} = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}\big(x_n\big|\mu_j, \Sigma_j\big)}$$

- We consider π_k as the prior probability of $z_k = 1$.
- The quantity $\gamma(z_{nk})$ will be the corresponding posterior probability once we have observed x_n.

Maximum Likelihood

- Gaussian Mixture Distribution:

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

- The parameters: π, μ, Σ
- Suppose we have a data set of observations $X = \{x_1, x_2, \dots, x_N\} \in \mathcal{R}^{N\times D}$, model the data using a mixture of Gaussians.
- Maximize the log of the likelihood function:

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)\}$$

EM for Mixtures of Gaussians

- An elegant and powerful method for finding maximum likelihood solutions for Gaussian Mixture Model is called the expectation-maximization algorithm (EM).
- Find π_k, μ_k and Σ_k.
- Consider the gradients of log-likelihood function w.r.t these parameters.

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)\}$$

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

EM for Mixtures of Gaussians : μ_k

- Setting the derivatives of $\ln p(X|\pi, \mu, \Sigma)$ w.r.t μ_k to 0:

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}\left(x|\mu_j, \Sigma_j\right)}}_{r(z_{nk})} \Sigma_k (x_n - \mu_k)$$

- We get:

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} r(z_{nk}) x_n$$

where $N_k = \sum_{n=1}^{N} r(z_{nk})$

EM for Mixtures of Gaussians: μ_k

- We have:

$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk})x_n$$

$$N_k = \sum_{n=1}^{N} r(z_{nk})$$

- $r(z_{nk})$: posterior probability $p(k|z_{nk})$, responsibility that component k takes for “explaining” the observation x_n.
- N_k as the effective number of points assigned to cluster k.
- μ_k for the k-th Gaussian component is obtained by taking a weighted mean of **ALL** of the points in the data set, in which the weighting factor for data point x_n is given by the posterior probability $r(z_{nk})$ that component k was responsible for generating x_n.

EM for Mixtures of Gaussians: Σ_k

- Setting the derivatives of $\ln p(X|\pi, \mu, \Sigma)$ w.r.t Σ_k to 0:
- We get:

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$N_k = \sum_{n=1}^{N} r(z_{nk})$$

- Each data point weighted by the corresponding posterior probability. Denominator given by the effective number of points associated with the corresponding component.

EM for Mixtures of Gaussians: π_k

- Finally, we maximize $\ln p(X|\pi, \mu, \Sigma)$ w.r.t π_k:
- We know that $\sum_k \pi_k = 1$:
- Using a Lagrange multiplier and maximizing the following quantity:

$$\ln p(X|\pi, \mu, \Sigma) + \lambda(\textstyle\sum_{k=1}^{K} \pi_k - 1)$$

- It gives:

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_j (x|\mu_j, \Sigma_j)} + \lambda$$

- Multiply both sides by π_k and sum over k, make use of $\sum_k \pi_k = 1$, we find $\lambda = -N$ and we get:

$$\pi_k = \frac{N_k}{N} = \frac{\sum_{n=1}^{N} r(z_{nk})}{N}$$

EM Algorithm: Initialization

- Initialize the means μ_k, covariances Σ_k and mixing coefficient π_k, and evaluate the initial value of the log likelihood.

EM algorithm: E step

- **E step**. Evaluate the responsibilities using the current parameter values:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}\left(x_n|\mu_j, \Sigma_j\right)}$$

EM algorithm: M Step

- **M step**. Re-estimate the parameters using the current responsibilities (where $N_k = \sum_{n=1}^{N} r(z_{nk})$)

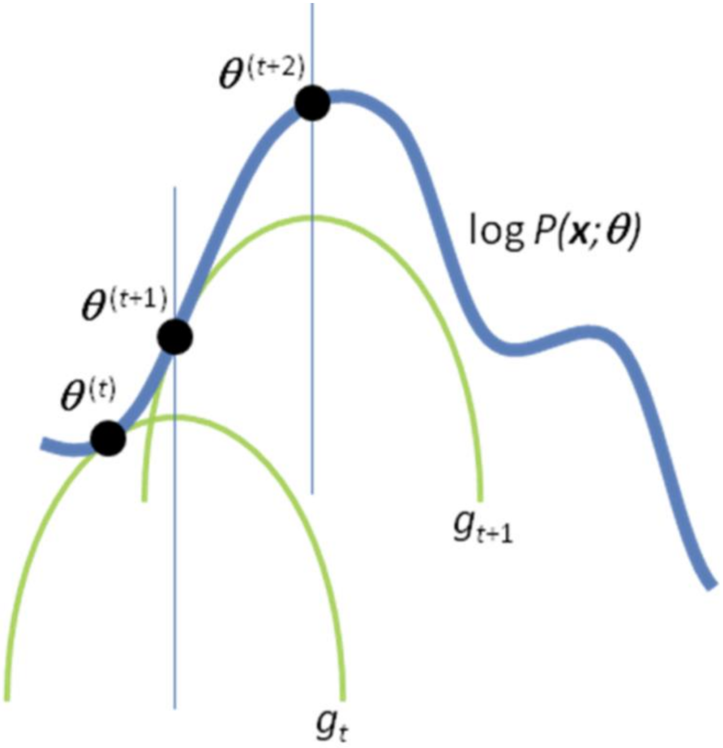
$$\mu_k = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk})x_n$$

$$\Sigma_k = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

Convergence of EM Algorithm

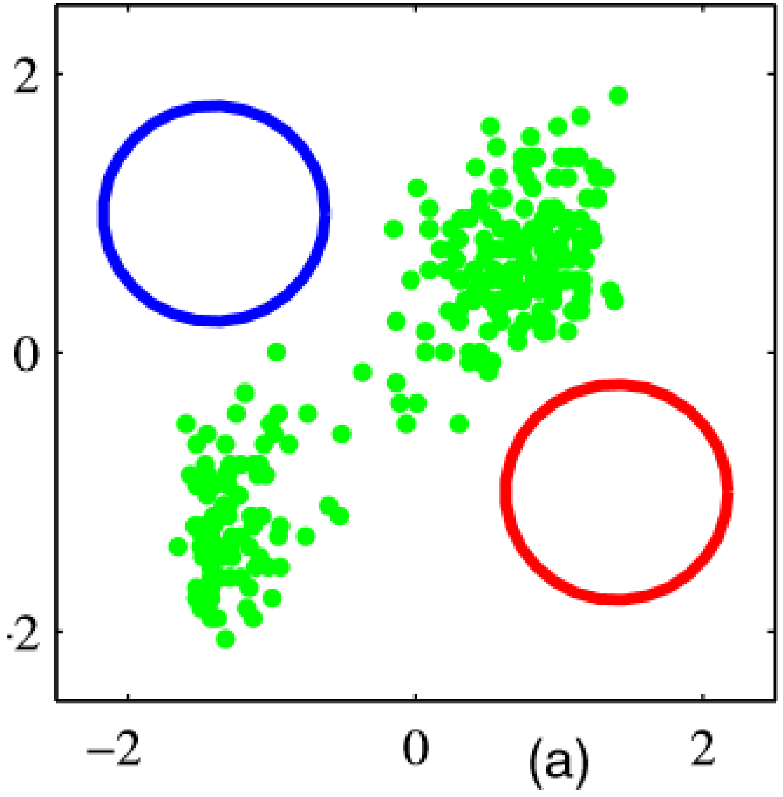
- Evaluate the log likelihood by checking the convergence of either the parameters or the log likelihood.



Supplementary Figure 1 Convergence of the EM algorithm. Starting from initial parameters $\boldsymbol{\theta}^{(t)}$, the E-step of the EM algorithm constructs a function g_t that lower-bounds the objective function $\log P(\boldsymbol{x};\boldsymbol{\theta})$. In the M-step, $\boldsymbol{\theta}^{(t+1)}$ is computed as the maximum of g_t. In the next E-step, a new lower-bound g_{t+1} is constructed; maximization of g_{t+1} in the next M-step gives $\boldsymbol{\theta}^{(t+2)}$, etc.

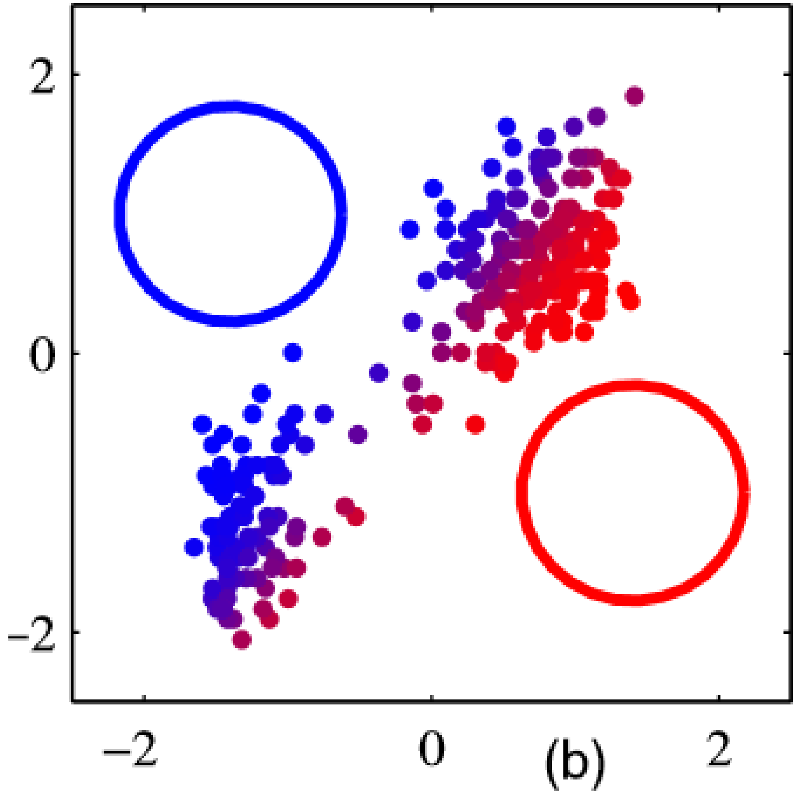
Figure source: https://people.duke.edu/~ccc14/sta-663/EMAlgorithm.html

EM for GMM: Example



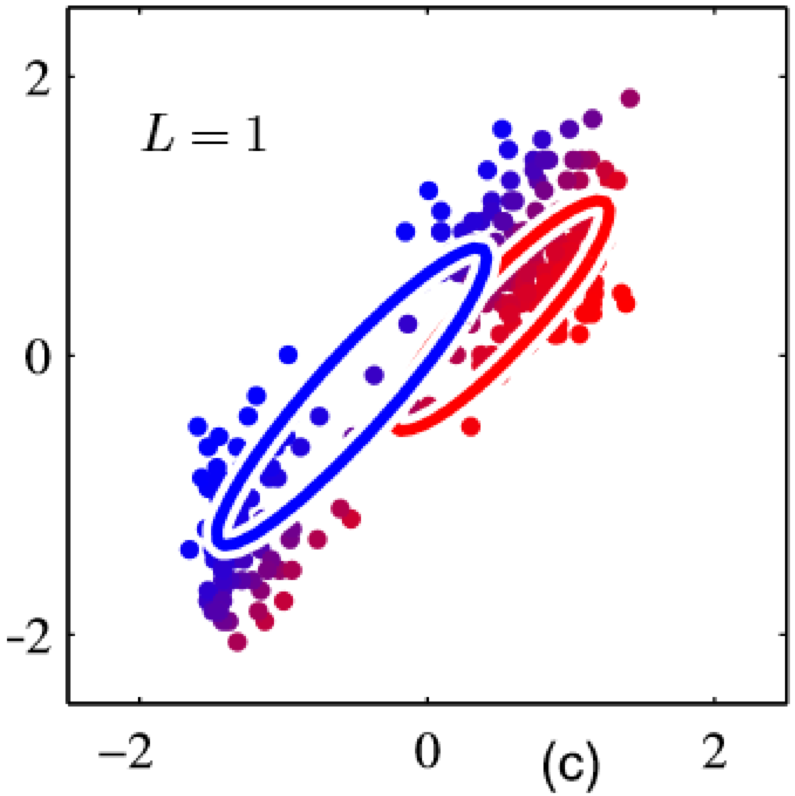
Initialization: initialize the means, covariance and mixing coefficient [C. Bishop, PRML]

EM for GMM: Example



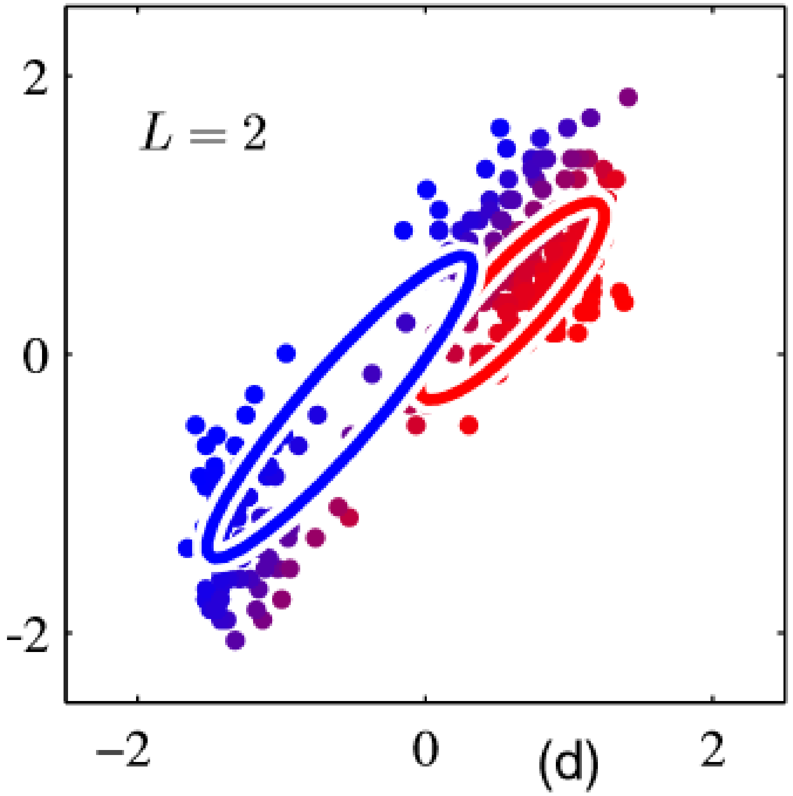
E step: use the current values for the parameters to evaluate the posterior probabilities, or responsibilities $r(z_{nk})$. [C. Bishop, PRML]

EM for GMM: Example



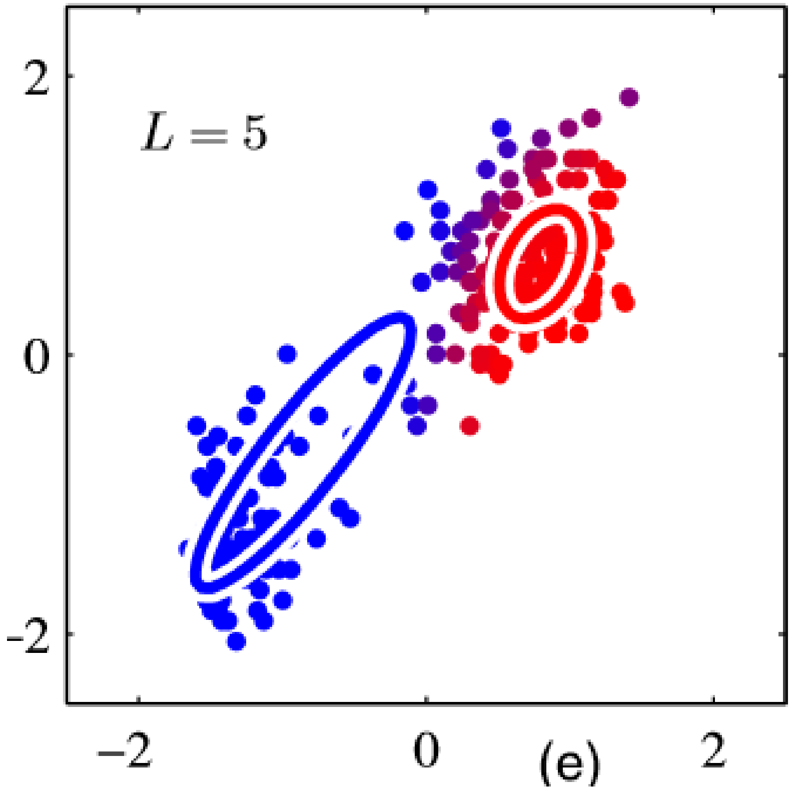
M step: use the posterior $r(z_{nk})$ to re-estimate the means, covariances, and mixing coefficients [C. Bishop, PRML]

EM for GMM: Example



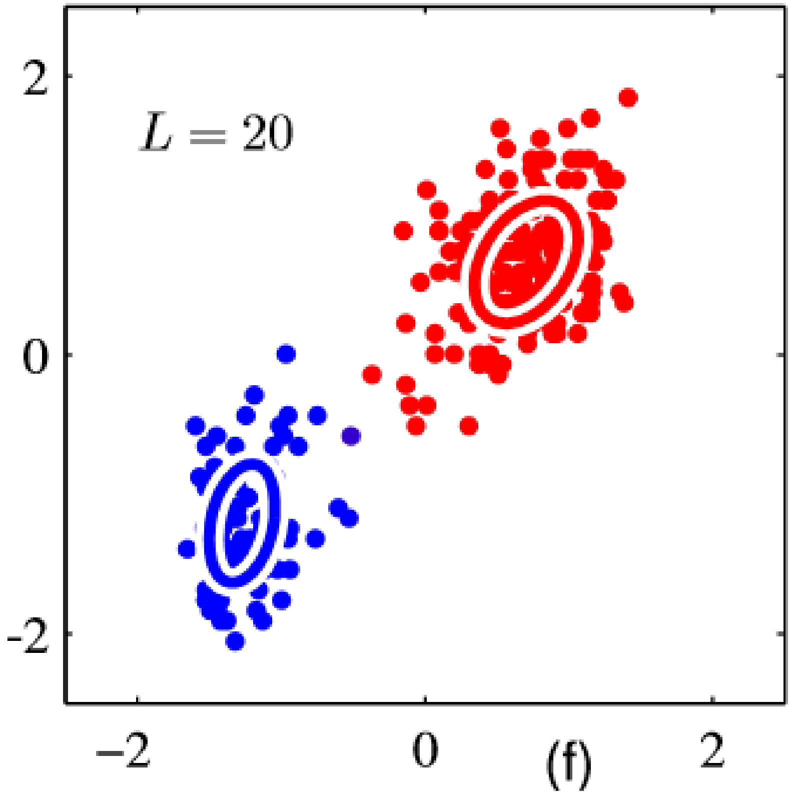
After 2 complete cycle of EM. [C. Bishop, PRML]

EM for GMM: Example



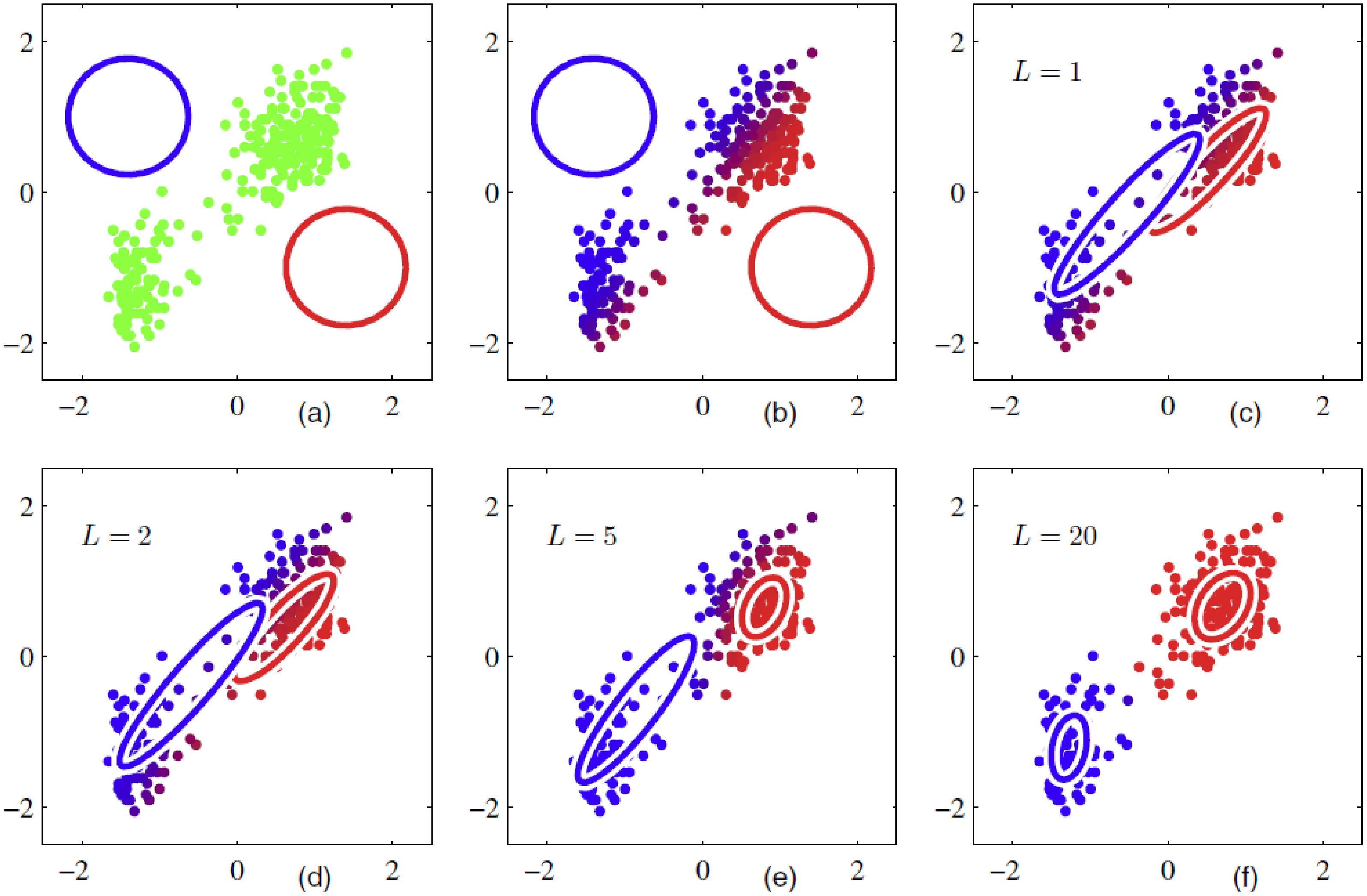
After 5 complete cycle of EM. [C. Bishop, PRML]

EM for GMM: Example



After 20 complete cycle of EM. [C. Bishop, PRML]

EM for GMM: Example



K-means vs GMM

- K-means: hard assignment, each data point is associated uniquely with one cluster.
- GMM: soft assignment, based on the posterior probabilities.
  - K-means as special case: consider same and infinitely small variance for each Gaussian component.
  - K-means results can be used as the initialization for EM algorithm.

Summary of Today’s Lecture

- Clustering
    - K-means Algorithm
- Gaussian Mixture Model (GMM)
- Expectation-Maximization (EM) algorithm for GMM