# Introduction to Machine Learning

# Lecture 11
# Unsupervised Learning II

Goker Erdogan
26 – 30 November 2018
Pontificia Universidad Javeriana

# Clustering

- Find subgroups (clusters) in data
  - What makes a cluster?
    - Within cluster similarity high
    - Across cluster similarity low

- Canonical application: market segmentation
  - Cluster customers into subgroups
  - Target each subgroup differently

- Many clustering techniques
  - We'll look at
    - K-means
    - Gaussian mixtures

# K-means algorithm

- Partition data into K distinct (non-overlapping) clusters
  - Objective: minimize within cluster variation

- Given N training samples $\{x\}_{n=1,2,...N}$
  - K clusters
    - Each cluster has a prototype vector $\boldsymbol{\mu}_k$
  - Assign each sample to a cluster, $r_{nk}=1$ if $x_n$ belongs to cluster k

  - Minimize

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

# K-means algorithm

- Solve

$$\min_{r,\boldsymbol{\mu}} \sum_n \sum_k r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Difficult
  - Too many possible partitions, $\sim K^N$
  - Find local minimum using an iterative procedure

    1) Fix r, optimize for $\boldsymbol{\mu}$

    2) Fix $\boldsymbol{\mu}$, optimize for r

    3) Repeat

# Solving K-means

- Fix $\boldsymbol{\mu}$, optimize for r
  - Assign each sample to closest prototype's cluster

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

- Fix r, optimize for $\boldsymbol{\mu}$
  - Set $\boldsymbol{\mu}_k$ to the center of all samples in cluster k

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}.$$

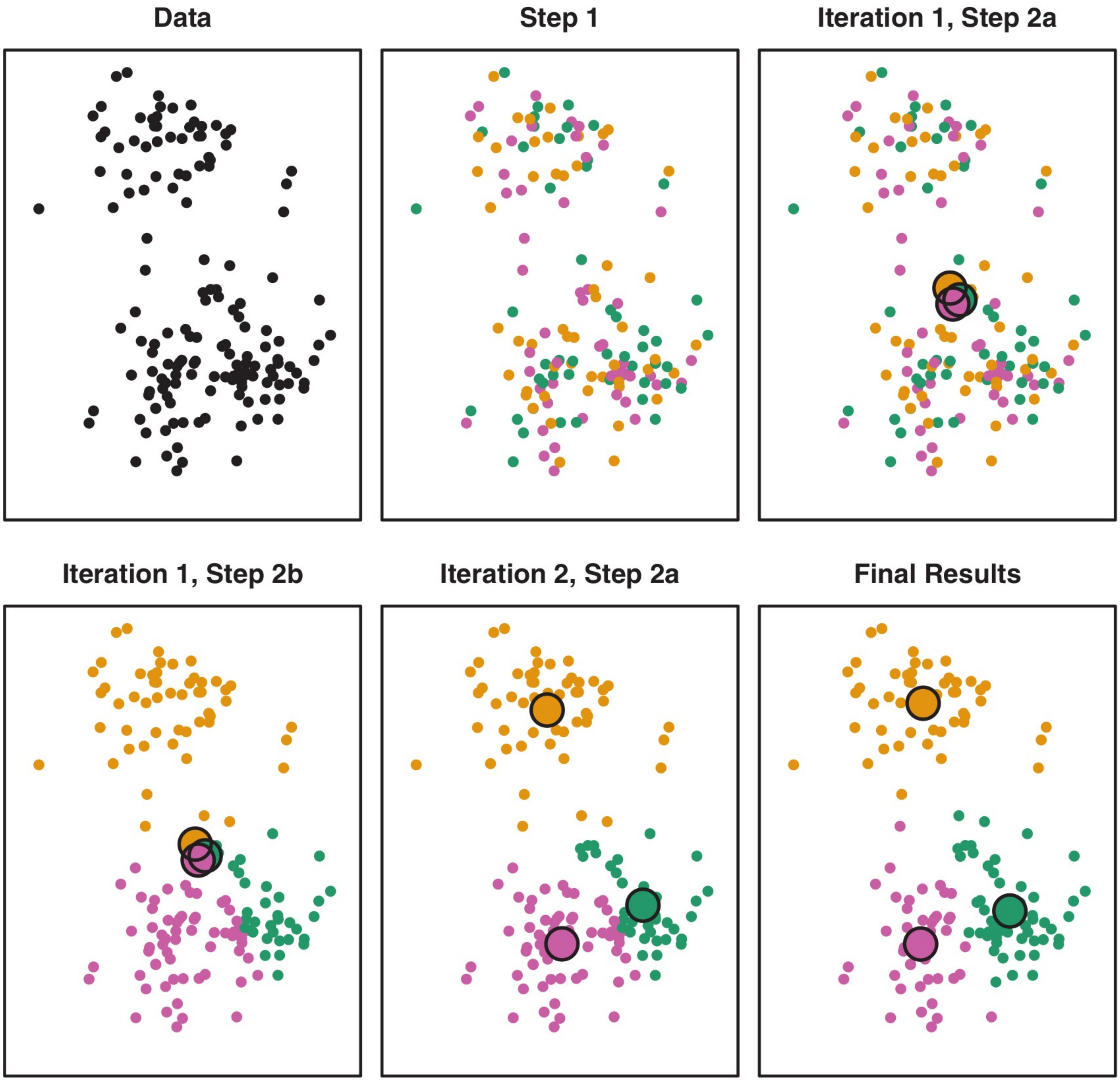
**Algorithm 10.1** *$K$-Means Clustering*

1. Randomly assign a number, from 1 to $K$, to each of the observations. These serve as initial cluster assignments for the observations.

2. Iterate until the cluster assignments stop changing:

   (a) For each of the $K$ clusters, compute the cluster *centroid*. The $k$th cluster centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.

   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

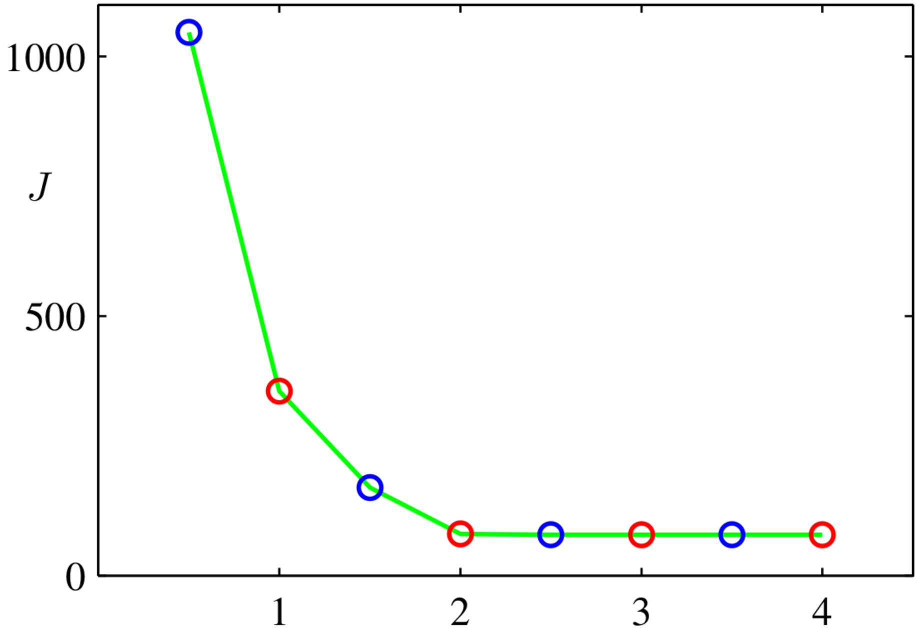

Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

# K-means algorithm

- Cost is guaranteed to decrease (or stay the same)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

  - Watch J and stop when it doesn't improve anymore
    - Cluster assignments don't change
  - Or after a maximum number of iterations



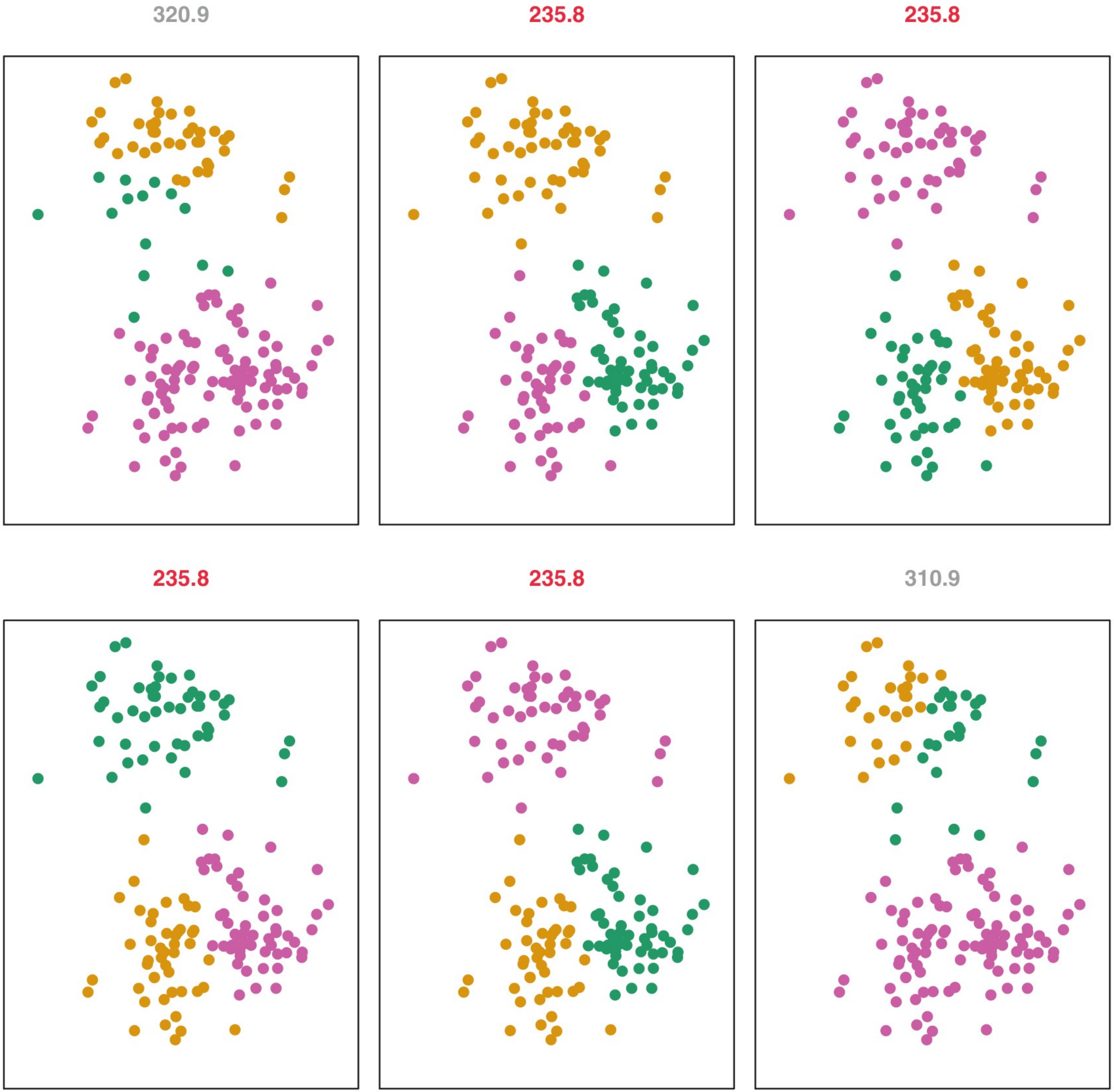
# K-means algorithm

- Cost is guaranteed to decrease (or stay the same)

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

  - Watch J and stop when it doesn't improve anymore
    - Cluster assignments don't change
  - Or after a maximum number of iterations

- Sensitive to initial cluster assignments
  - Finds local minima
  - Run k-means multiple times with different random initializations
    - Pick the best



320.9

235.8

235.8

235.8

235.8

310.9

# Potential issues

- Can be slow if N is large
  - Techniques to speed it up are available

- Hard to pick number of clusters K
  - Run it on a separate dataset and see if you get the same clusters
  - Can use bootstrap to sample datasets

- Not robust to outliers
  - Because of squared Euclidean distances

- K-means makes hard cluster assignments
  - A sample belongs to a cluster or not
  - Soft assignments may make more sense

# K-medoids algorithm

- Use a different distance (dissimilarity) function
  - e.g., $l_1$ norm (absolute value)

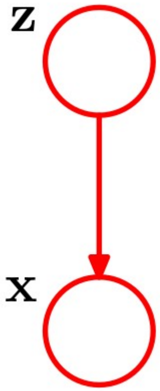
$$\widetilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

- Can be hard to optimize wrt to $\boldsymbol{\mu}$

- Constrain: prototype $\boldsymbol{\mu}_k$ must be one of $x_n$ in the cluster
  - Makes it easier to optimize
  - Look at each $x_n$ in the cluster one by one

# The probabilistic perspective

- Assume a latent variable z for cluster membership
  - $z_{nk} = 1$ if $x_n$ belongs to cluster k
  - z has a multinomial distribution



$$z \sim \mathrm{Mult}(\pi_1, \pi_2, \ldots, \pi_k)$$

$$0 \leqslant \pi_k \leqslant 1 \qquad \sum_{k=1}^{K} \pi_k = 1$$

- Assume x|z has a Gaussian distribution

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

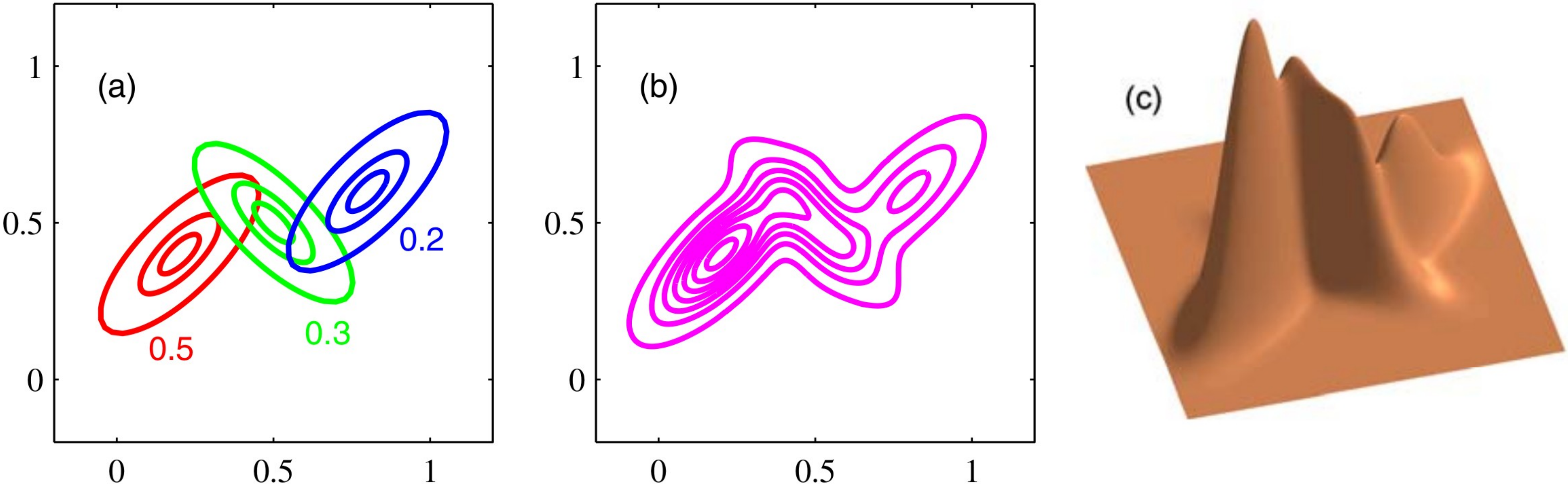


**Figure 2.23** Illustration of a mixture of 3 Gaussians in a two-dimensional space. (a) Contours of constant density for each of the mixture components, in which the 3 components are denoted red, blue and green, and the values of the mixing coefficients are shown below each component. (b) Contours of the marginal probability density $p(\mathbf{x})$ of the mixture distribution. (c) A surface plot of the distribution $p(\mathbf{x})$.

# Mixture of Gaussians

- Given a dataset of N samples {x}$_{n=1,2,...N}$
  - Find parameters $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$

- Write the likelihood of parameters given data
  - Maximize it (maximum likelihood)

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

$$p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}.$$

Mixture of Gaussians model

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

# Maximum likelihood for MoG

- Log likelihood of parameters given data

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}.$$

- Maximize using an iterative procedure
  - Fix $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ optimize for $\boldsymbol{\pi}$
  - Fix $\boldsymbol{\pi}$ optimize for $\boldsymbol{\mu}, \boldsymbol{\Sigma}$

- This is expectation-maximization (EM) applied to mixture of Gaussians

## EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients $\pi_k$, and evaluate the initial value of the log likelihood.
2. **E step**. Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{9.23}$$

3. **M step**. Re-estimate the parameters using the current responsibilities

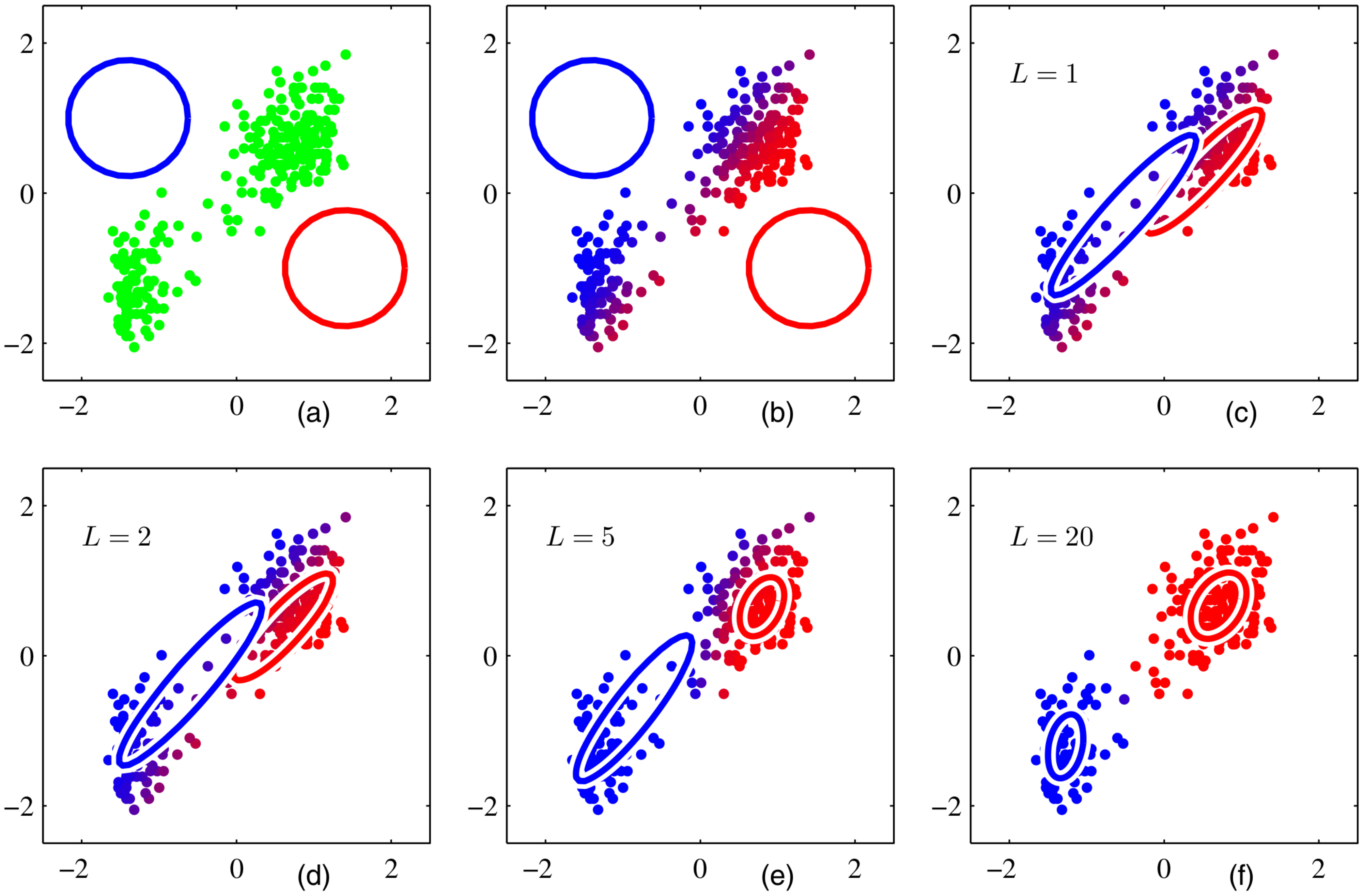
$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \tag{9.24}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \tag{9.25}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \tag{9.26}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}). \tag{9.27}$$


2
0
−2
−2
0
(a)
2
2
0
−2
−2
0
(b)
2
2
L = 1
0
−2
−2
0
(c)
2
2
L = 2
0
−2
−2
0
(d)
2
2
L = 5
0
−2
−2
0
(e)
2
2
L = 20
0
−2
−2
0
(f)
2

# Why do all of this?

- Allows you to use any distribution
  - K-means: spherical clusters
  - Mixture of Bernoullis

- A principled approach to pick K (number of clusters)
  - Find the K that maximizes posterior
  - Better: Marginalize out K (average over all possible K)

- Allows you to generalize it
  - Dirichlet mixture models
    - Infinite mixture of Gaussian models
    - No need to specify K
    - Adjusted based on data

# Summary

- Clustering
- K-means
  - Algorithm
  - K-medoids
- Probabilistic perspective
  - Mixture of Gaussians
  - Maximum likelihood for MoG
- Exercises
  - Show that for the K-means algorithm, the cost J never increases from one iteration to next
  - Do lab 10.5.1 in ISLR

# References

[1] James, Witten, Hastie, and Tibshirani. An Introduction to Statistical Learning with Applications in R. Chapter 10.

[2] Hastie, Tibshirani, and Friedman. The Elements of Statistical Learning. Chapter 14.

[3] Bishop, C. Pattern Recognition and Machine Learning. Chapter 9.