Pattern Recognition

Clustering

林彥宇 教授
Yen-Yu Lin, Professor

國立陽明交通大學 資訊工程學系
Computer Science, National Yang Ming Chiao Tung University

Outline

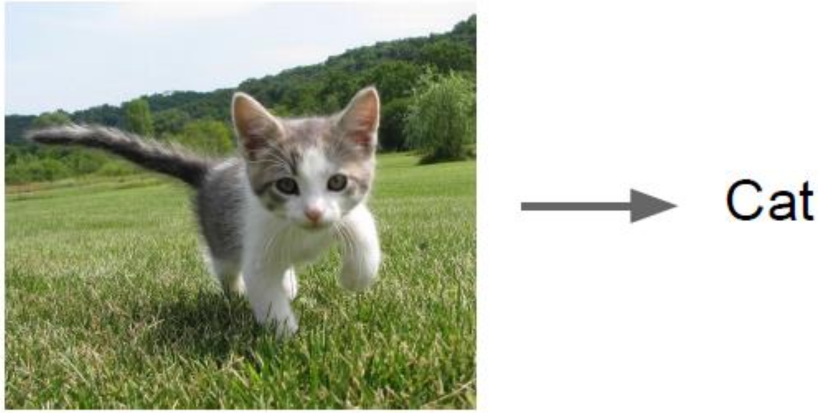
- Supervised learning vs. Unsupervised learning
- k-means clustering
- Mixtures of Gaussians

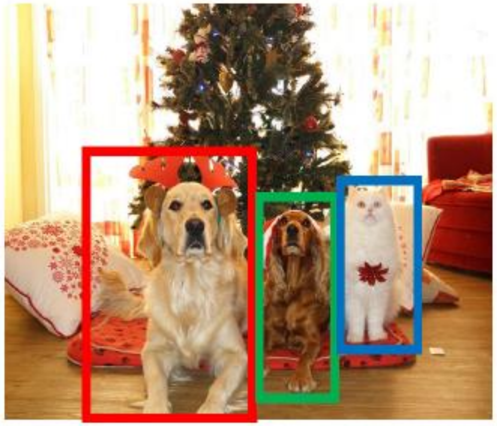
Outline

- Supervised learning vs. Unsupervised learning
- k-means clustering
- Mixtures of Gaussians

Supervised vs. unsupervised learning

- Supervised learning
- Labeled training data: $\{(\mathbf{x}_n, t_n)\}_{n=1}^{N}$
  - $\mathbf{x}_n$ is the *n*-th data point and t_n is its target label/value
- Goal: Learn a function to map $\mathbf{x}$ (data point) to t (label/value)
- Classification: FLD, neural networks, AdaBoost, SVMs, …
- Regression: curve fitting, Bayesian regression, neural networks, SVR, …





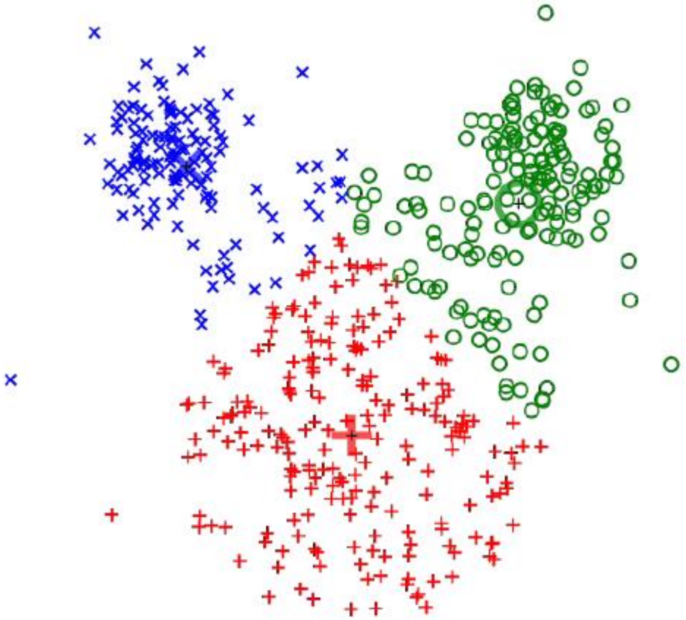
DOG, DOG, CAT

Supervised vs. unsupervised learning

- Unsupervised learning
- Unlabeled training data: $\{\mathbf{x}_n\}_{n=1}^{N}$
  - $\mathbf{x}_n$ is the *n*-th data point and no label is available
- Goal: Learn some underlying hidden structure of the data
- Some unsupervised learning tasks:
  - Clustering
  - Dimensionality reduction
  - Density estimation
  - Data reconstruction
  - Data generation

Clustering

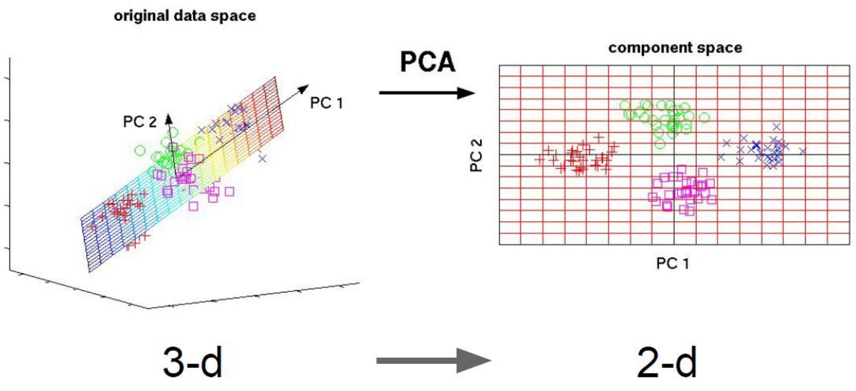
- Given a data set, partition the data set into clusters
  - Data points in the same cluster are more similar to each other than to those in other clusters



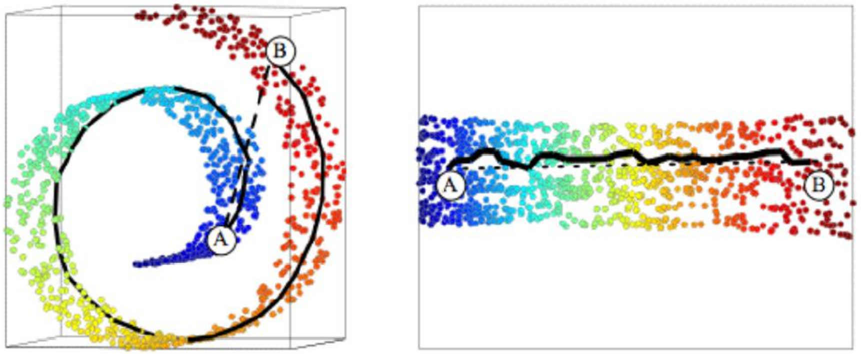
k-means clustering

Dimensionality reduction

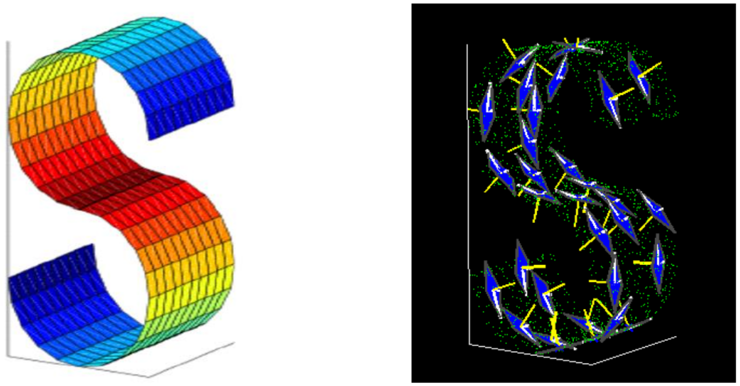
- Projecting data from a high-dimensional space to a low-dimensional one based on some criterion

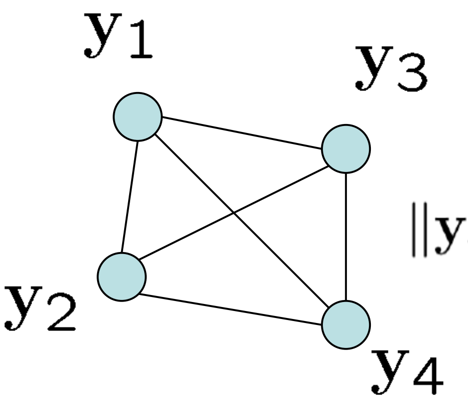


PCA



Isomap





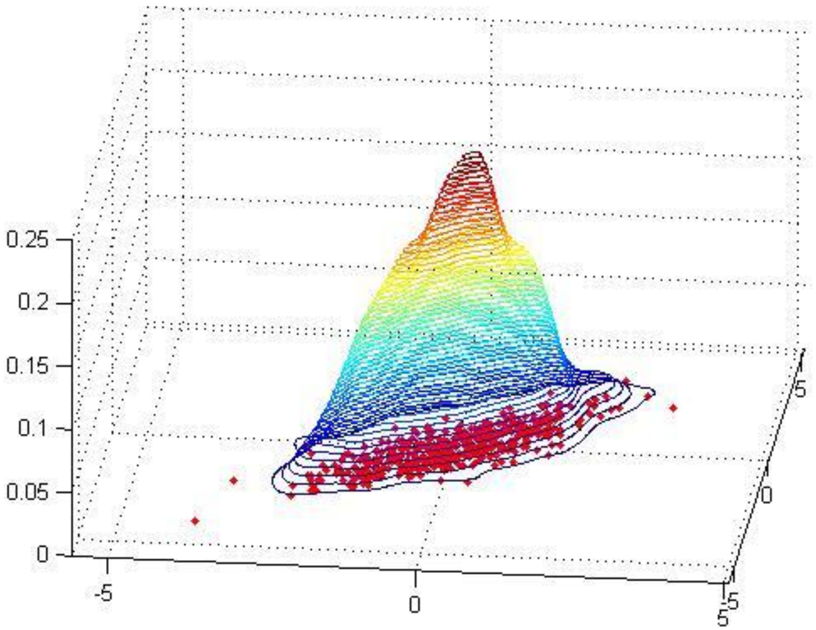
$$\|\mathbf{y}_i - \mathbf{y}_j\| \approx \Delta_{ij}$$

LLE

MDS

Density estimation

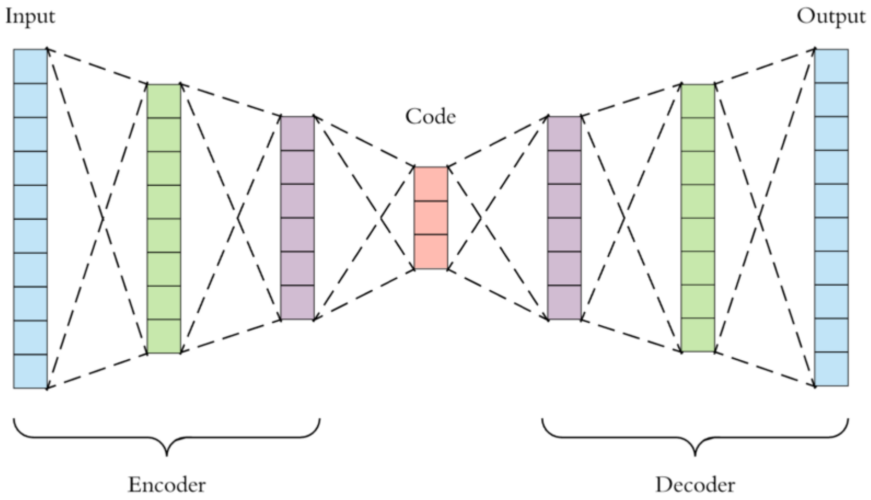
- Given a set of data points, estimate the underlying probability density function



kernel density estimation

Data reconstruction

- Autoencoder: a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The codes can be used to reconstruct the original data
  - Encoder: Decrease data dimensionality for codes generation
  - Decoder: Increase data dimensionality for reconstruction

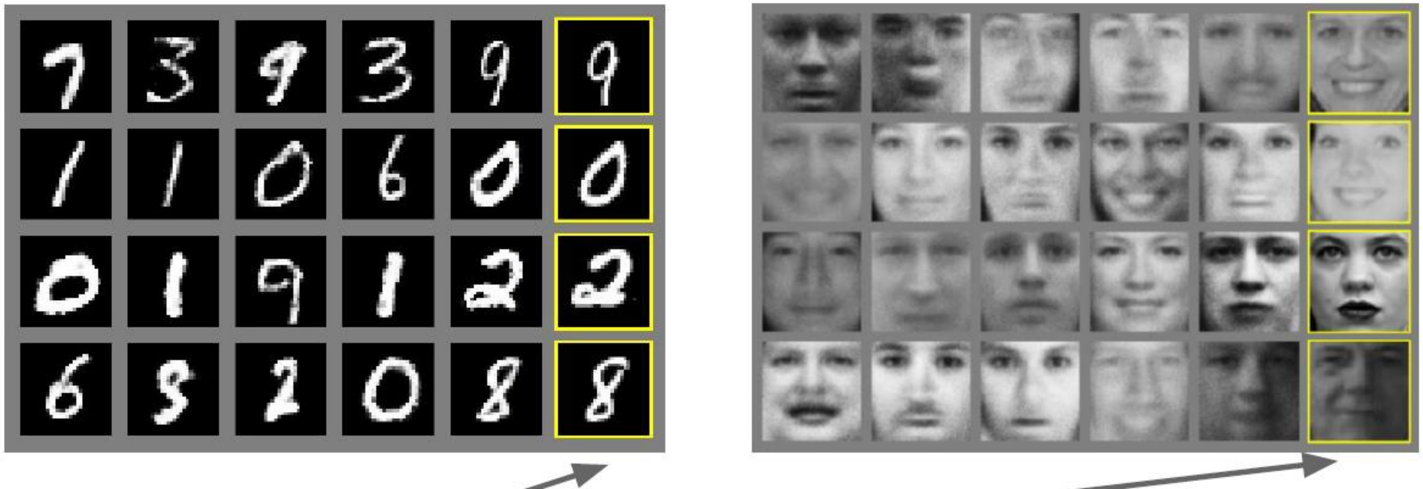


https://www.edureka.co/blog/autoencoders-tutorial/

Data generation

- Generate new data (images) from a complex, high-dimensional distribution
  - ➢ This distribution may be implicitly specified by a set of data

Training data ~ $p_{data}(x)$ Generated samples ~ $p_{model}(x)$



Outline

- Supervised learning vs. Unsupervised learning
- k-means clustering
- Mixtures of Gaussians

Clustering

- Given a data set, identify groups, or clusters, of these data points
- We have a set of data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ in a random D-dimensional space. Our goal is to partition the data set into K clusters
  - N observations, D-dimensional space, and K clusters
- A cluster is a subset of these data points whose inter-point distances are small compared with the distances to points outside of the cluster
  - K clusters in a D-dimensional space: Introduce a set of D-dimensional vectors $\boldsymbol{\mu}_k$, where $k = 1, 2, \ldots, K$, in which $\boldsymbol{\mu}_k$ is a prototype associated with the k-th cluster

k-means clustering formulation

- In k-means clustering, we consider $\boldsymbol{\mu}_k$ as the center of the data points belonging to cluster k
- k-means clustering aims to find an assignment of data points to clusters, as well as a set of vectors $\{\boldsymbol{\mu}_k\}$, such that the sum of the squared distances of each data point to its closest vector $\boldsymbol{\mu}_k$ is a minimum
- For each data point $\mathbf{x}_n$, we introduce a corresponding set of binary indicator variables $r_{nk} \in \{0, 1\}$, where $k = 1, \ldots, K$
  - $r_{nk} = 1$: Data point $\mathbf{x}_n$ is assigned to the k-th cluster
  - $r_{nk} = 0$: otherwise
  - Each data point is assigned to exactly one cluster

k-means clustering formulation

- The objective function of k-means clustering is, sometimes called a distortion measure, given by

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

  - It represents the sum of the squared distances of each data point to its assigned cluster $\boldsymbol{\mu}_k$
  - Our goal is to find the optimal values for the assignment $\{r_{nk}\}_{n=1,k=1}^{N,K}$ and clusters $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$

k-means clustering algorithm

- 1. Choose some initial values for clusters $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$
- 2. An iterative, two-stage process
  - Keep clusters $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ fixed and minimize J with respect to assignment $\{r_{nk}\}_{n=1,k=1}^{N,K}$ (E step)
  - Keep assignment $\{r_{nk}\}_{n=1,k=1}^{N,K}$ fixed and minimize J with respect to clusters $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ (M step)
- 3. Repeat step 2 until convergence
- Remark: Convergence of the k-means algorithm is assured. However, it may converge to a local minimum of J

EM algorithms

- EM (expectation-maximization) algorithms are PR and ML techniques for finding maximum likelihood estimators in latent variable models
  - An EM algorithm contains E (expectation) and M (maximization) steps

- k-means clustering is an EM algorithm
  - E step: Optimize the assignment by fixing the clusters
  - M step: Optimize the clusters by fixing the assignment

E-step in k-means clustering

- Objective function

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

- J is a linear function with respect to assignment $\{r_{nk}\}_{n=1,k=1}^{N,K}$
- The terms for different data points are independent
  - We can optimize for each data point $\mathbf{x}_n$ separately
- Since each point point $\mathbf{x}_n$ belongs to one cluster, we choose the cluster with minimal squared difference

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

M-step in k-means clustering

- Objective function

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

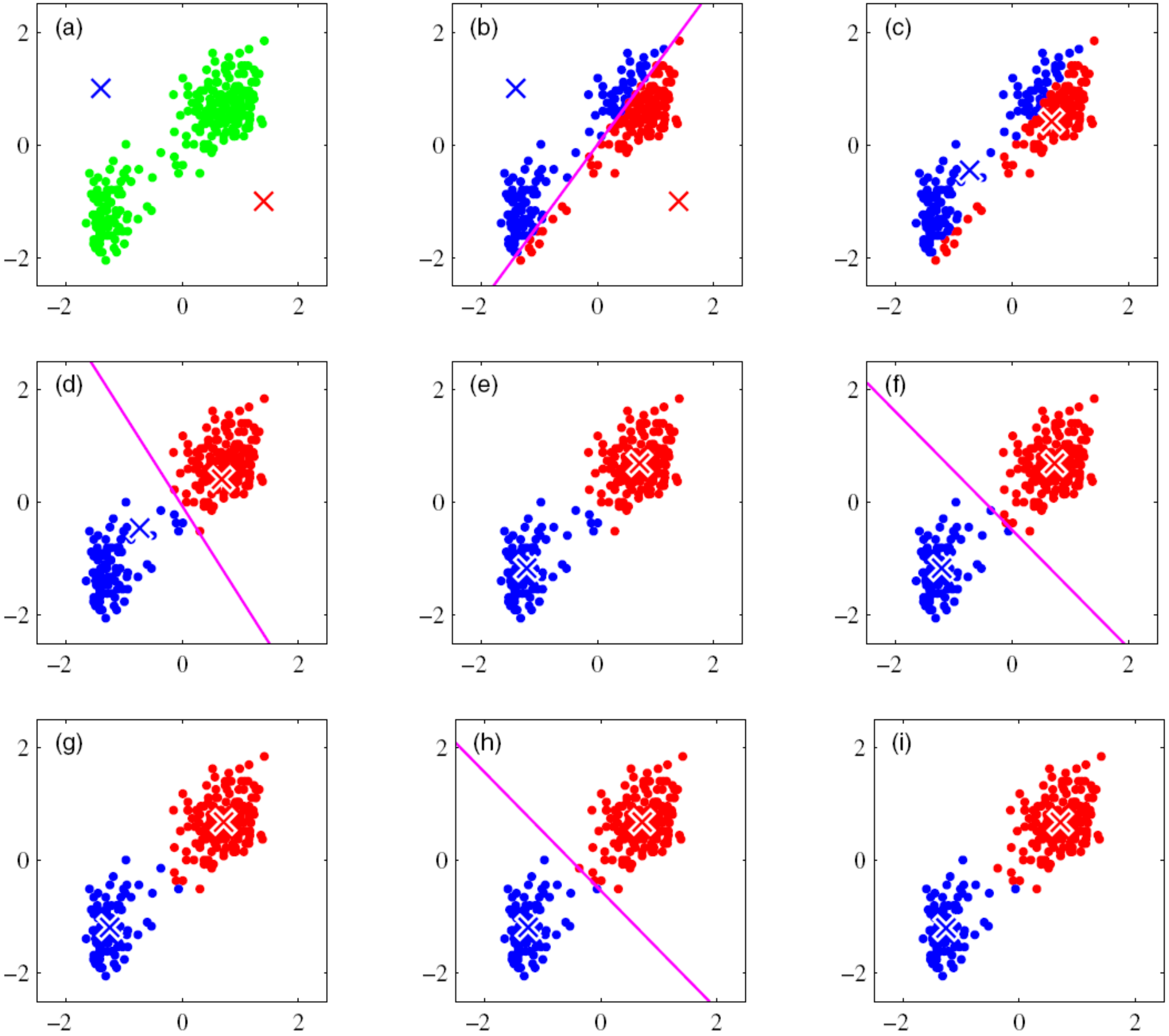
- J is a quadratic function with respect to each cluster $\boldsymbol{\mu}_k$
- Setting the derivative of J w.r.t. $\boldsymbol{\mu}_k$ to zero, we get

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

- $\boldsymbol{\mu}_k$ is the mean of data points assigned to cluster k
- That is the reason why this algorithm is called k-means clustering

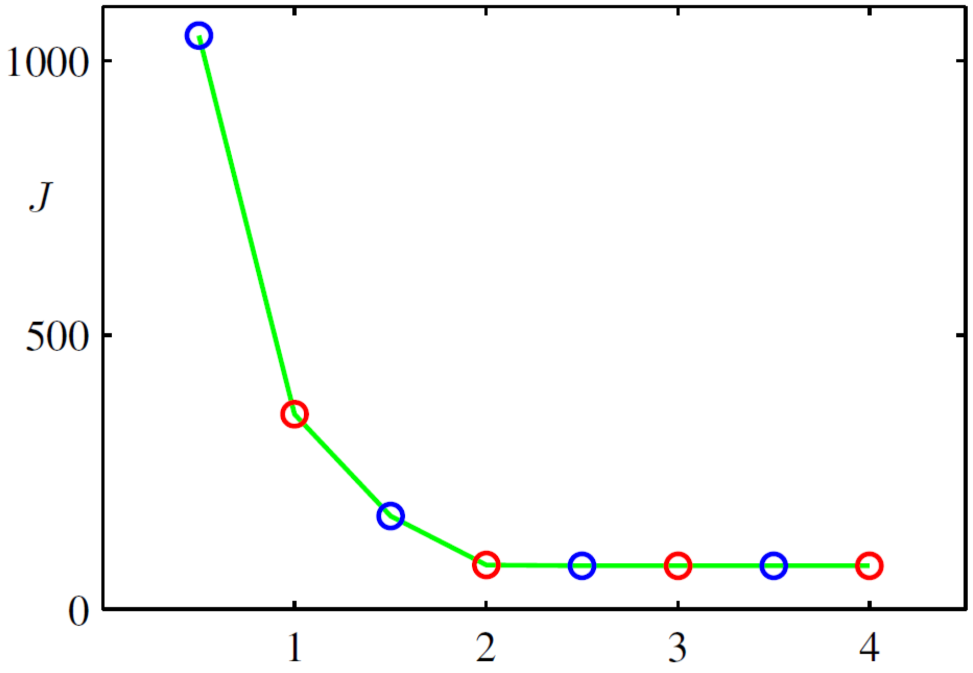
An example

- Green points
  - Data points
- Blue/Red crosses
  - $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$
- Purple line
  - Cluster partition



Convergence

- The plot of cost function J along the optimization process
- Blue point (E step), Red point (M step)
- The algorithm converges after three iterations



On-line k-means clustering

- Batch version of k-means clustering: The whole data are used together to update the clusters

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

- In on-line k-means clustering, we consider a data point $\mathbf{x}_n$ at a time. We update the nearest cluster center $\boldsymbol{\mu}_k$ via

$$\boldsymbol{\mu}_k^{\text{new}} = \boldsymbol{\mu}_k^{\text{old}} + \eta_n (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{old}})$$

  - where η_n is the learning rate

k-medoids clustering

- k-medoids clustering is the same as k-means clustering except for two differences
- 1. The objective function is changed to

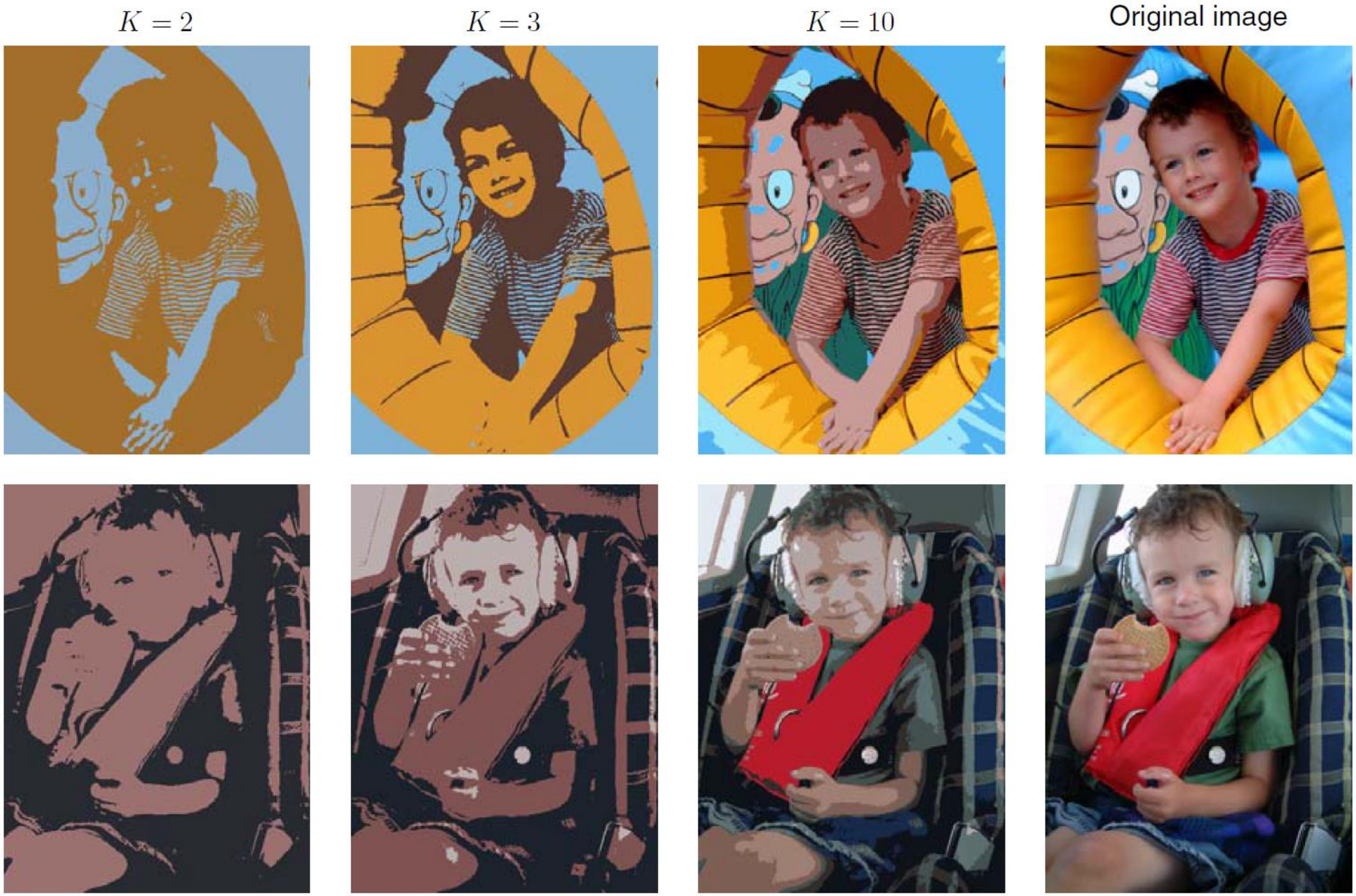
$$\widetilde{J} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \mathcal{V}(\mathbf{x}_n, \boldsymbol{\mu}_k)$$

  - A distance function $\mathcal{V}(\cdot, \cdot)$ for measuring the dissimilarity between a pair of data points
- 2. In M-step, each cluster $\boldsymbol{\mu}_k$ must be a data point
  - For data belonging to cluster k, we set $\boldsymbol{\mu}_k$ to the data point with the shortest average distance to all other points of this cluster

Image segmentation using k-means clustering

- Image segmentation is to partition an image into homogeneous regions
  - Pixels in each region have similar visual appearance
- Given an image, each pixel in this image is a point in a 3-dimensional space, i.e., the intensities in the R, G, and B channels
- k-means clustering is applied to pixels of the images, and we can get the cluster centers $\{\boldsymbol{\mu}_k\}$

Image segmentation using k-means clustering



$K = 2$ $K = 3$ $K = 10$ Original image

Image compression via k-means clustering

- Data before compression: $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ in a random D-dimensional space

- Data after compression: $\{\boldsymbol{\mu}_k\}_{k=1}^{K}$ and $\{r_{nk}\}_{n=1,k=1}^{N,K}$

Outline

- Supervised learning vs. Unsupervised learning
- k-means clustering
- Mixtures of Gaussians

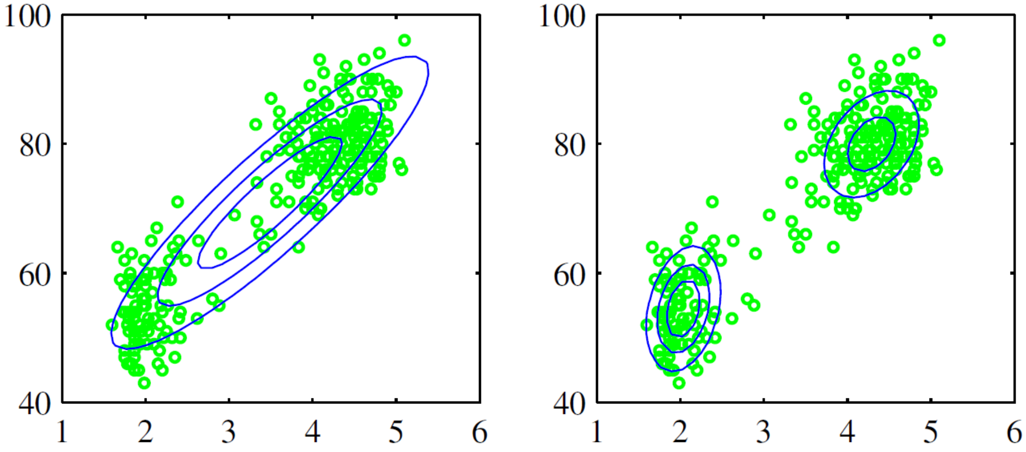
Gaussian distribution

- The Gaussian distribution defined over a D-dimensional vector $\mathbf{x}$ of continuous variables:

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

  where $\boldsymbol{\mu}$ is the mean while Σ is the co-variance matrix

- Gaussian distribution is widely used, but some data distributions cannot be well fit by a Gaussian distribution



Mixture of Gaussians

- A mixture of Gaussians

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

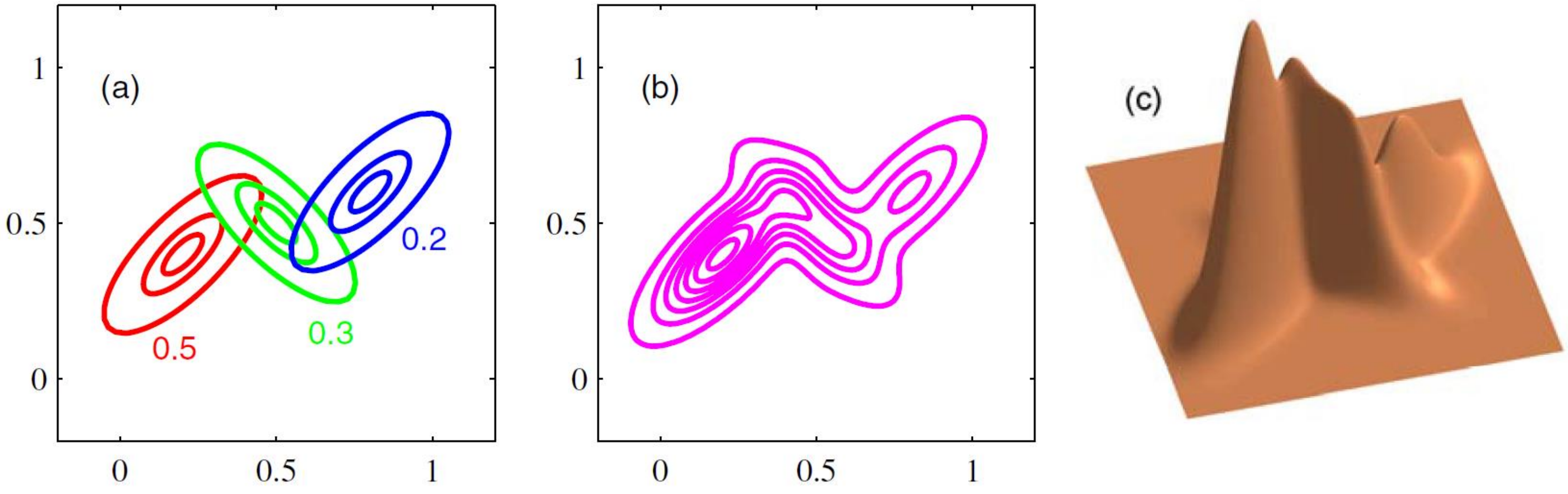
  - A distribution that is composed of multiple (K here) Gaussian distributions
  - Each Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is called a component of the mixture, and has its own mean $\boldsymbol{\mu}_k$ and covariance Σ_k
  - π_k is the mixing coefficient of the k-th component with

$$0 \leqslant \pi_k \leqslant 1$$

$$\sum_{k=1}^{K} \pi_k = 1$$

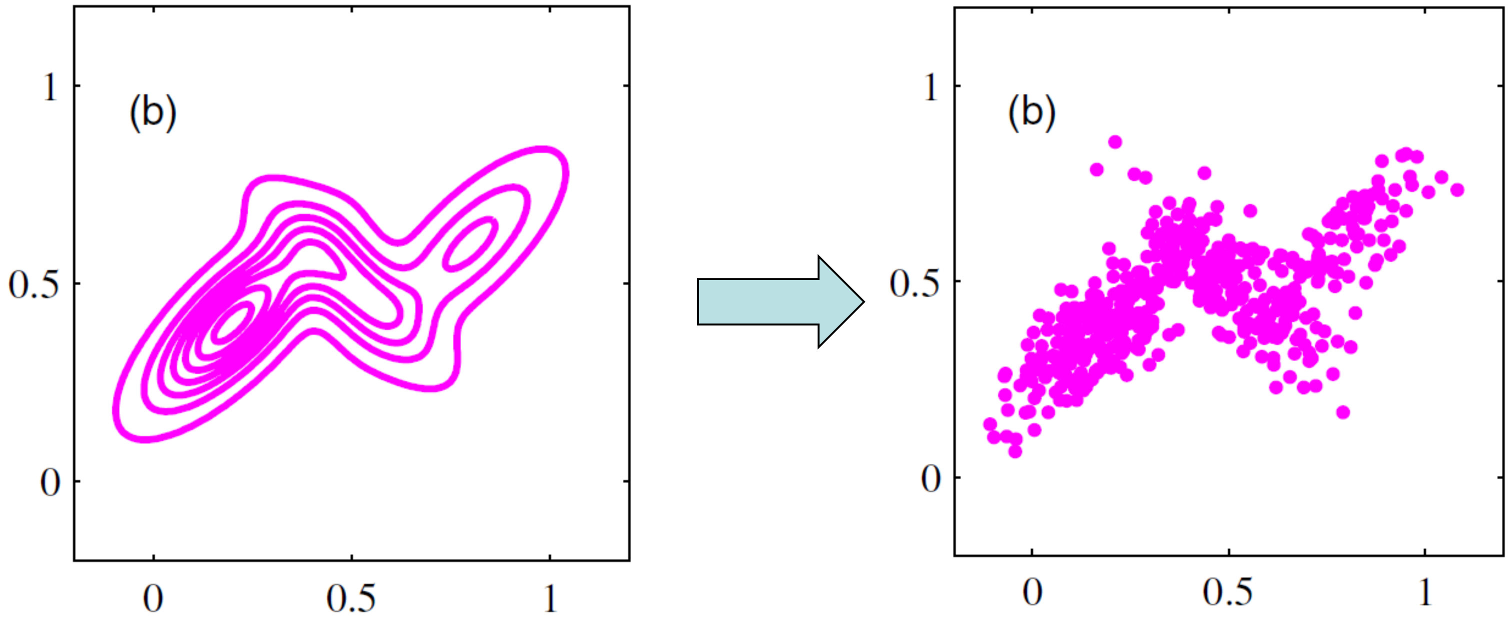
An example

- A mixture of three Gaussians
  - (a) Three components are denoted by R, G, and B respectively with their mixing coefficients
  - (b) The distribution specified by the mixture of Gaussians
  - (c) The surface plot of this distribution



How to fit a mixture of Gaussians

- Given a set of data points $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ sampled from an unknown distribution, how to use these data to fit a mixture of Gaussians with a specific value of K



Latent variable

- A mixture of Gaussians

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

- We associate each data point $\mathbf{x}$ with a random variable $\mathbf{z} = [z_1, z_2, \dots, z_K]$ having a 1-of-K representation
  - $z_k \in \{0, 1\}$ and $\sum_k z_k = 1$

- In a Gaussian mixture model (GMM), we model a conditional distribution $p(\mathbf{x}|\mathbf{z})$ and a marginal distribution $p(\mathbf{z})$ to estimate the data distribution $p(\mathbf{x})$, i.e.,

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$$

Conditional and marginal distributions

- The marginal distribution is defined as follows

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

  - ➢ The marginal distribution is correlated with mixing coefficients

- The conditional distribution of $\mathbf{x}$ given some particular Gaussian component k

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- Jointly considering multiple components, we have

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

  - ➢ The conditional distribution is correlated with the parameters of Gaussians

Data distribution via conditional and marginal distributions

- Marginal and conditional distributions

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

- We model the data distribution via the marginal and conditional distributions

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

- Since we have a set of data points $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, we associate a random variable $\boldsymbol{z}_n$ with each data point $\mathbf{x}_n$

Responsibility

- Another important conditional probability of $\mathbf{z}$ given $\mathbf{x}$

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

- $\gamma(z_k)$ can be considered as the responsibility that component k takes for explaining the data point $\mathbf{x}$

- GMM for clustering: $\gamma(z_k)$ is the probability of assigning $\mathbf{x}$ to cluster k (soft assignment)

Likelihood

- The data distribution

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

- The log-likelihood of the whole data

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

- Fit a mixture of Gaussian by maximizing log-likelihood

Fitting GMM

- Data log-likelihood

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

- 1. Setting the derivative of log-likelihood w.r.t. $\boldsymbol{\mu}_k$ to zero gives

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

- After rearrangement, we have

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

Fitting GMM

- 2. Setting the derivative of log-likelihood w.r.t. Σ_k to zero gives

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

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

  under the constraint $\sum_{k=1}^{K}\pi_k = 1$

  - Solving this constrained optimization problem by Lagrangian, we have

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

Iterative process

- E step: Estimate the responsibility

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

- M step: Update the GMM parameters, including mixing coefficients, means, and covariance matrices

$$\pi_k = \frac{N_k}{N} \qquad \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

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

Initialization

- Run k-means clustering
- 1. The means of Gaussian components are set to the obtained cluster centers
- 2. The covariance of each component is set to the covariance matrix obtained by using data belonging to this component
- 3. The mixing coefficient of each component is set to the fraction of data assigned to this component

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\Sigma}_k$ and mixing coefficients π_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

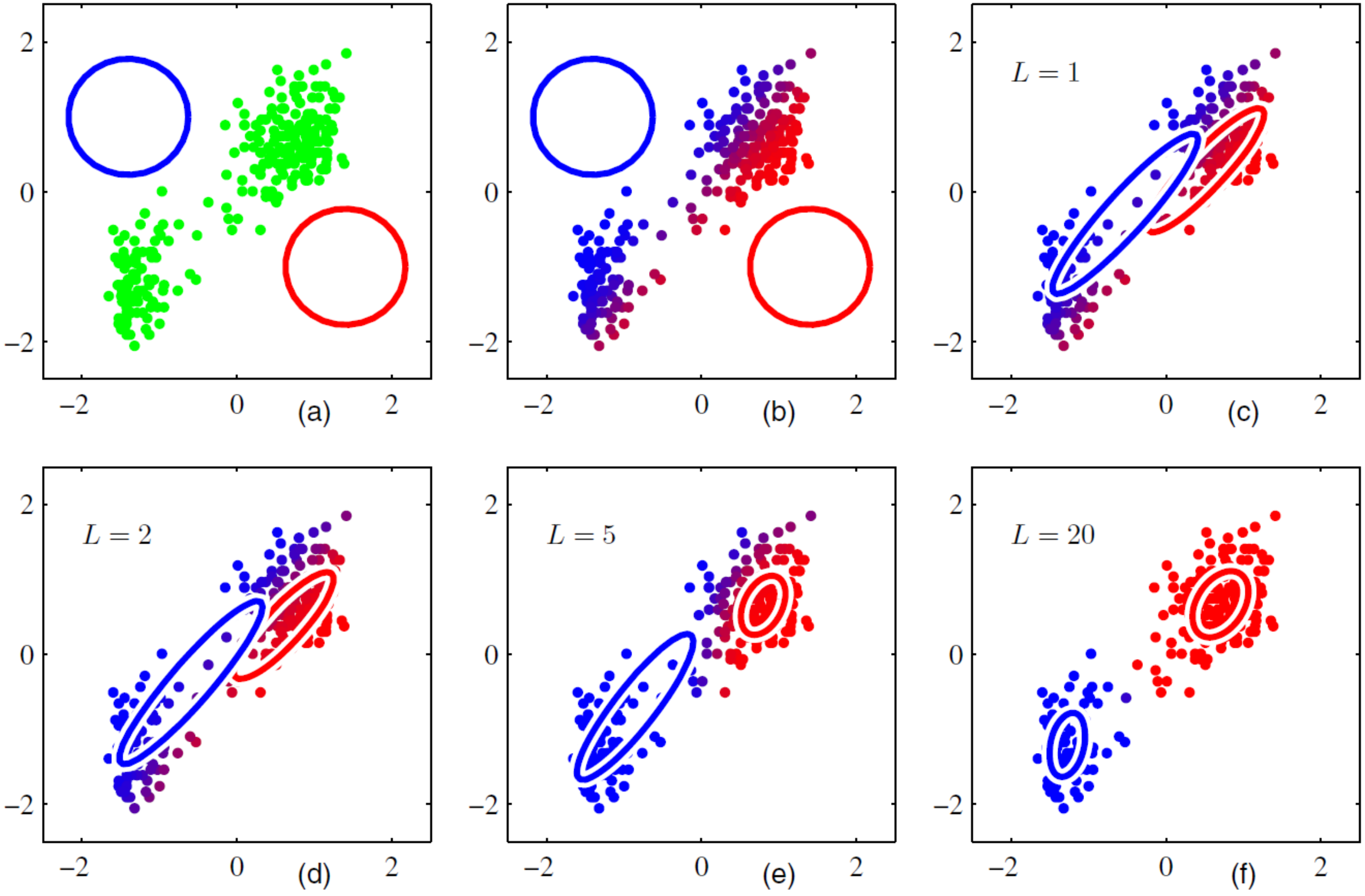
$$N_k = \sum_{n=1}^{N}\gamma(z_{nk}). \tag{9.27}$$

4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N}\ln\left\{\sum_{k=1}^{K}\pi_k\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right\} \tag{9.28}$$

and check for convergence of either the parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

An example



References

- k-means clustering
  - Chapter 9.1 in the PRML textbook
- Mixtures of Gaussians
  - Chapter 9.2 in the PRML textbook

Thank You for Your Attention!

Yen-Yu Lin (林彥宇)

Email: lin@cs.nctu.edu.tw
URL: https://www.cs.nctu.edu.tw/members/detail/lin