بسم الله الرحمن الرحيم

STATISTICAL PATTERN RECOGNITION

M. R. Ahmadzadeh
Isfahan University of Technology
Ahmadzadeh@iut.ac.ir

Most of materials used in this course were taken and edited from the textbook *"Pattern Recognition"*
By Sergios Theodoridis and Konstantinos Koutroumbas

Textbooks

- **Pattern Recognition, 4th Ed., Theodoridis and Koutroumbas**
- **Pattern Classification (2nd ed.) by Richard O. Duda, Peter E. Hart and David G. Stork**
- **Pattern Recognition and Machine Learning, Bishop**
- The elements of statistical learning Data mining, inference, and prediction 2008-Trevor Hastie et al.
- Murphy, Machine Learning A Probabilistic Perspective
- Statistical Pattern Recognition, 3rd Ed. Andrew R. Webb And Keith D. Copsey
- Introduction to Statistical Pattern Recognition, 2nd Ed., Fukunaga
- A Statistical Approach to Neural Networks for Pattern Recognition, R. A. Dunne.

Grading Criteria

- Midterm Exam ≈ 25%±5%
- HW, Comp. Assignments and projects: ≈ 30%
- Final exam ≈ 45%±5%

- **Course Website**:
- http://yekta.iut.ac.ir or http://elearning.iut.ac.ir/
- **Email:** Ahmadzadeh@iut.ac.ir
- **Skype Name**: live:ahmadzadeh.m_2
- **Skype Group:** See course website
  How do I hand in homework? Hardcopy or electronic version (single file in pdf and LMS only- no email please). Losing 30% of the grade for every week of late submission.

PATTERN RECOGNITION

- Typical application areas
  - Machine vision
  - Character recognition (OCR)
  - Computer aided diagnosis
  - Speech recognition
  - Face recognition
  - Biometrics
  - Image Data Base retrieval
  - Data mining
  - Bioinformatics

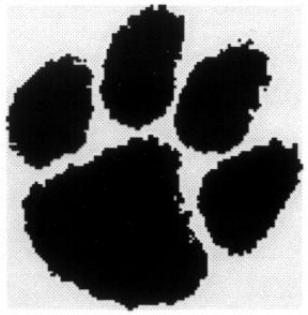
Statistical Pattern Recognition

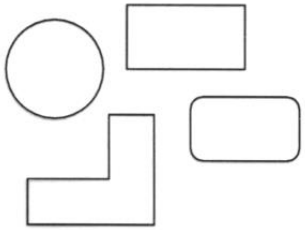
- 1. Introduction
- 2. Classifiers based on Bayes Decision
- 3. Linear Classifiers
- 4. Nonlinear Classifiers
- 5. Feature Selection
- 6. Feature Generation I: Data Transformation and Dimensionality Reduction
- 7. Feature Generation II
- 8. Template Matching

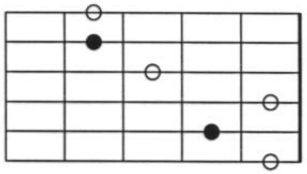
- 9. Context Dependent Clarification
- 10. Supervised Learning
- 11. Clustering: Basic Concepts
- 12. Clustering Algorithms I: Sequential
- 13. Clustering Algorithms II: Hierarchical
- 14. Clustering Algorithms III: Based on Function Optimization
- 15. Clustering Algorithms IV:
- 16. Cluster Validity

What is a Pattern?

- "A pattern is the opposite of a chaos; it is an entity vaguely defined, that could be given a name." (Watanabe)

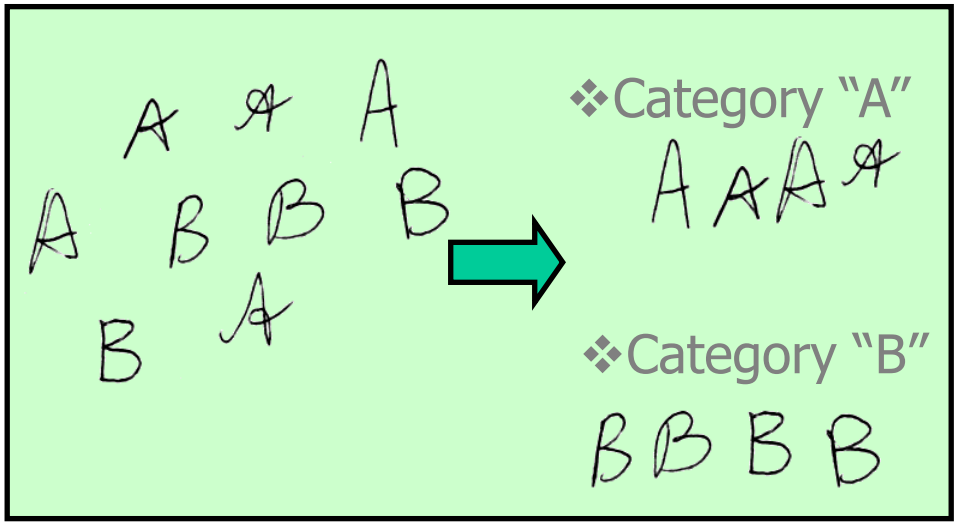




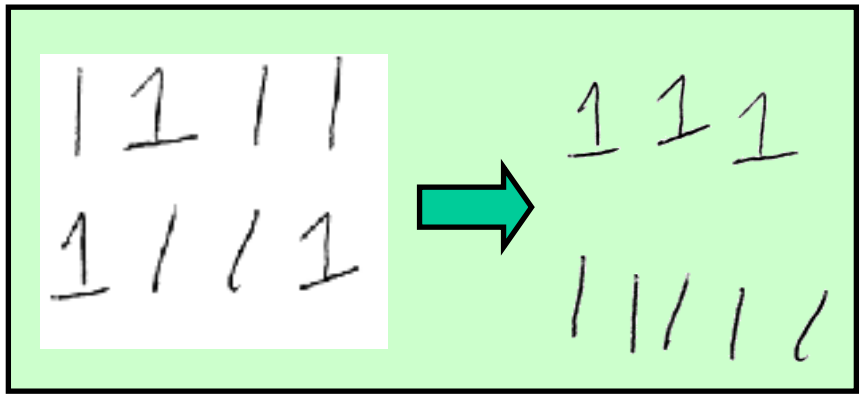


Recognition

- Identification of a pattern as a member of a category we already know, or we are familiar with
  - Classification (known categories)
  - Clustering (creation of new categories)
- The task: Assign unknown objects – patterns – into the correct class. This is known as classification.



Classification



Clustering

Pattern Recognition

- Given an input pattern, make a decision about the "category" or "class" of the pattern
- Pattern recognition is a very broad subject with many applications
- In this course we will study a variety of techniques to solve P.R. problems and discuss their relative strengths and weaknesses

Pattern Class

- A collection of "similar" (not necessarily identical) objects
- A class is defined by class samples (paradigms, exemplars, prototypes)
- Inter-class variability
- Intra-class variability

Pattern Class Model

❖ Different descriptions, which are typically mathematical in form for each class/population

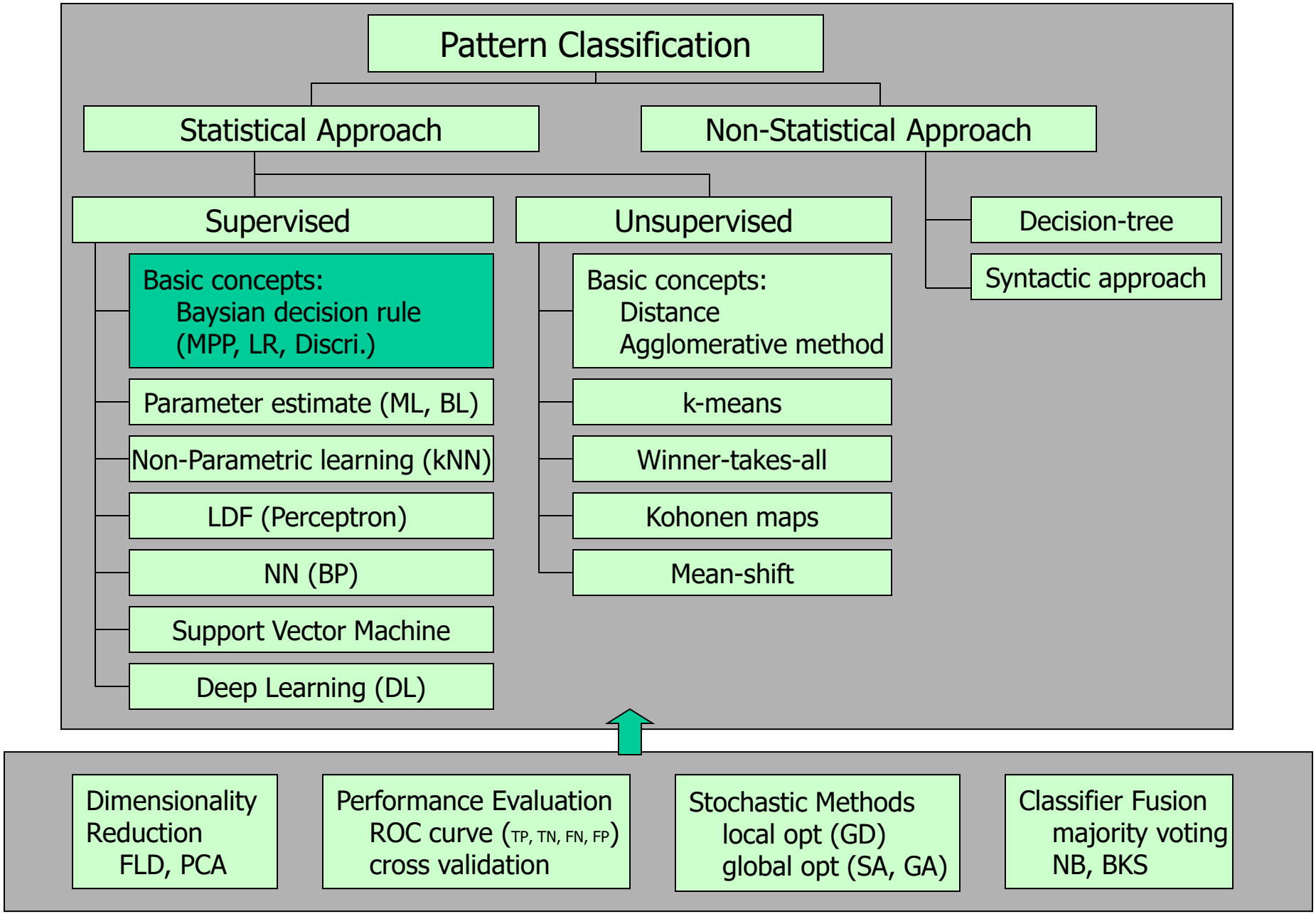
❖ Given a pattern, choose the best-fitting model for it and then assign it to class associated with the model

Pattern Recognition Applications

Problem	Input	Output
Speech recognition	Speech waveforms	Spoken words, speaker identity
Non-destructive testing	Ultrasound, eddy current, acoustic emission waveforms	Presence/absence of flaw, type of flaw
Detection and diagnosis of disease	EKG, EEG waveforms	Types of cardiac conditions, classes of brain conditions
Natural resource identification	Multispectral images	Terrain forms, vegetation cover
Aerial reconnaissance	Visual, infrared, radar images	Tanks, airfields
Character recognition (page readers, zip code, license plate)	Optical scanned image	Alphanumeric characters

Pattern Recognition Applications

Problem	Input	Output
Identification and counting of cells	Slides of blood samples, micro-sections of tissues	Type of cells
Inspection (PC boards, IC masks, textiles)	Scanned image (visible, infrared)	Acceptable/unacceptable
Manufacturing	3-D images (structured light, laser, stereo)	Identify objects, pose, assembly
Web search	Key words specified by a user	Text relevant to the user
Fingerprint identification	Input image from fingerprint sensors	Owner of the fingerprint, fingerprint classes
Online handwriting retrieval	Query word written by a user	Occurrence of the word in the database


Pattern Classification
Statistical Approach
Non-Statistical Approach
Supervised
Unsupervised
Decision-tree
Syntactic approach
Basic concepts:
Baysian decision rule
(MPP, LR, Discri.)
Parameter estimate (ML, BL)
Non-Parametric learning (kNN)
LDF (Perceptron)
NN (BP)
Support Vector Machine
Deep Learning (DL)
Basic concepts:
Distance
Agglomerative method
k-means
Winner-takes-all
Kohonen maps
Mean-shift
Dimensionality Reduction
FLD, PCA
Performance Evaluation
ROC curve (TP, TN, FN, FP)
cross validation
Stochastic Methods
local opt (GD)
global opt (SA, GA)
Classifier Fusion
majority voting
NB, BKS

An Example

"Sorting incoming Fish on a conveyor according to species using optical sensing"

Species

Sea bass

Salmon

❖Problem Analysis

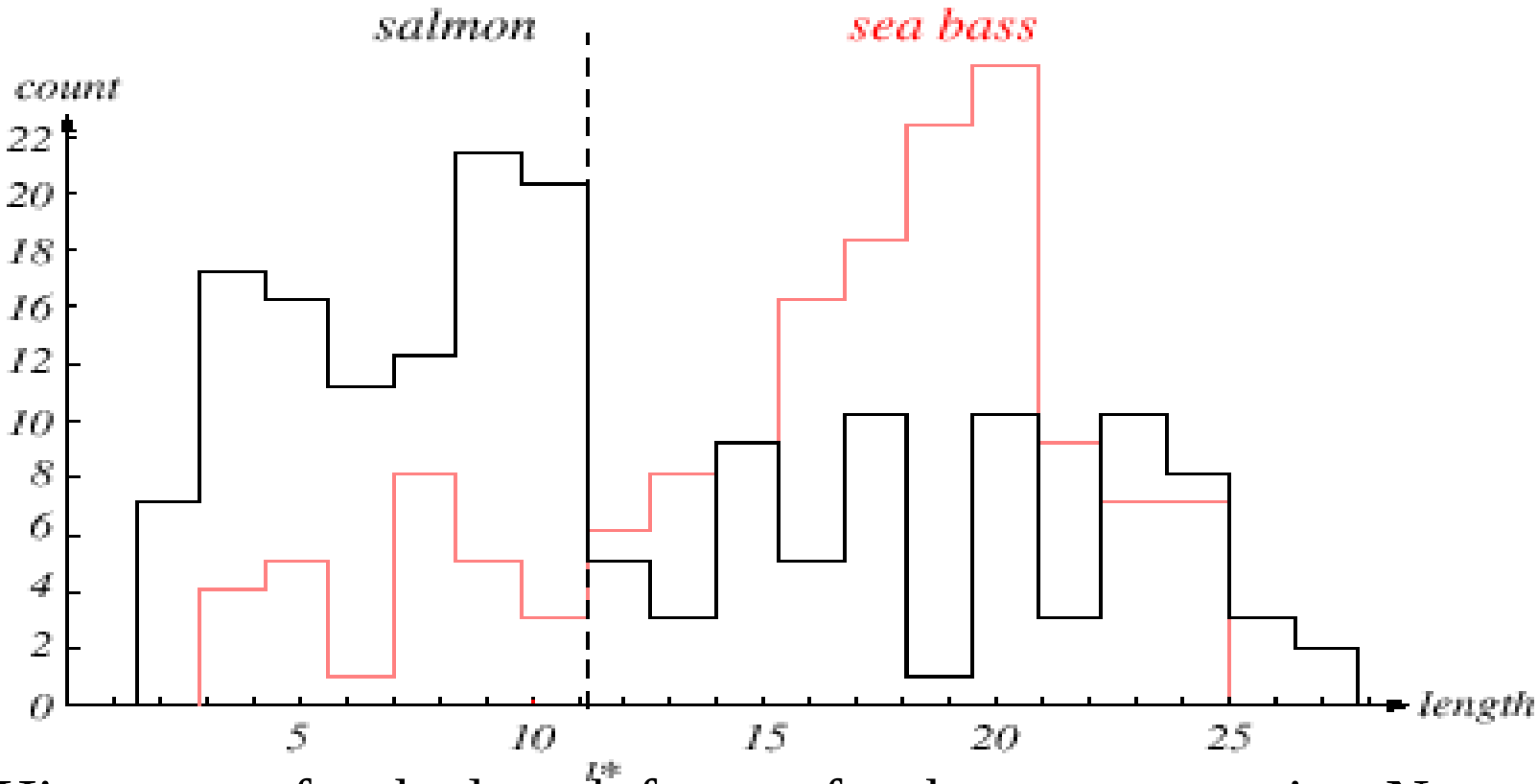
- ➢Set up a camera and take some sample images to extract features

  - Length
  - Lightness
  - Width
  - Number and shape of fins
  - Position of the mouth, etc…

  This is the set of all suggested features to explore for use in our classifier!

- Preprocessing
  - Use a segmentation operation to isolate fishes from one another and from the background
- Information from a single fish is sent to a feature extractor whose purpose is to reduce the data by measuring certain features
- The features are passed to a classifier

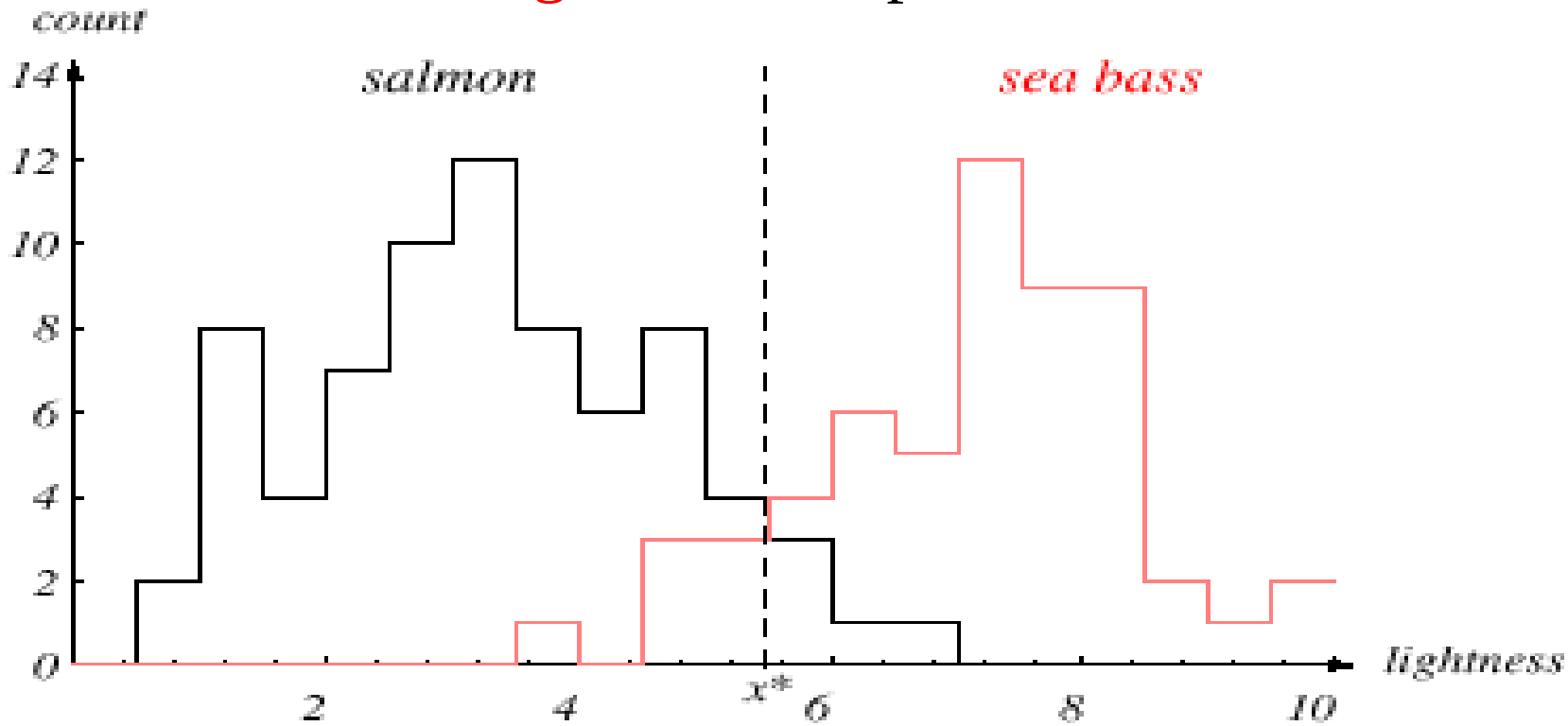
❖Classification: Select the length of the fish as a possible feature for discrimination



❖Histograms for the length feature for the two categories. No single threshold value l^* (decision boundary) will serve to unambiguously discriminate between the two categories; using length alone, we will have some errors. The value l^* marked will lead to the smallest number of errors, on average.

The length is a poor feature alone!

Select the lightness as a possible feature.



❖Histograms for the lightness feature for the two categories. No single threshold value x^* (decision boundary) will serve to unambiguously discriminate between the two categories; using lightness alone, we will have some errors. The value x^* marked will lead to the smallest number of errors, on average.

❖ Adopt the lightness and add the width of the fish

Fish → $\mathbf{x} = \underline{x} = [x_1, x_2]^T$

❖ **Lightness** ❖ **Width**

❖ We realize that the feature extractor has thus reduced the image of each fish to a point or *feature vector* **x** in a two-dimensional *feature space*.

❖ Features: These are measurable quantities obtained from the patterns, and the classification task is based on their respective values.

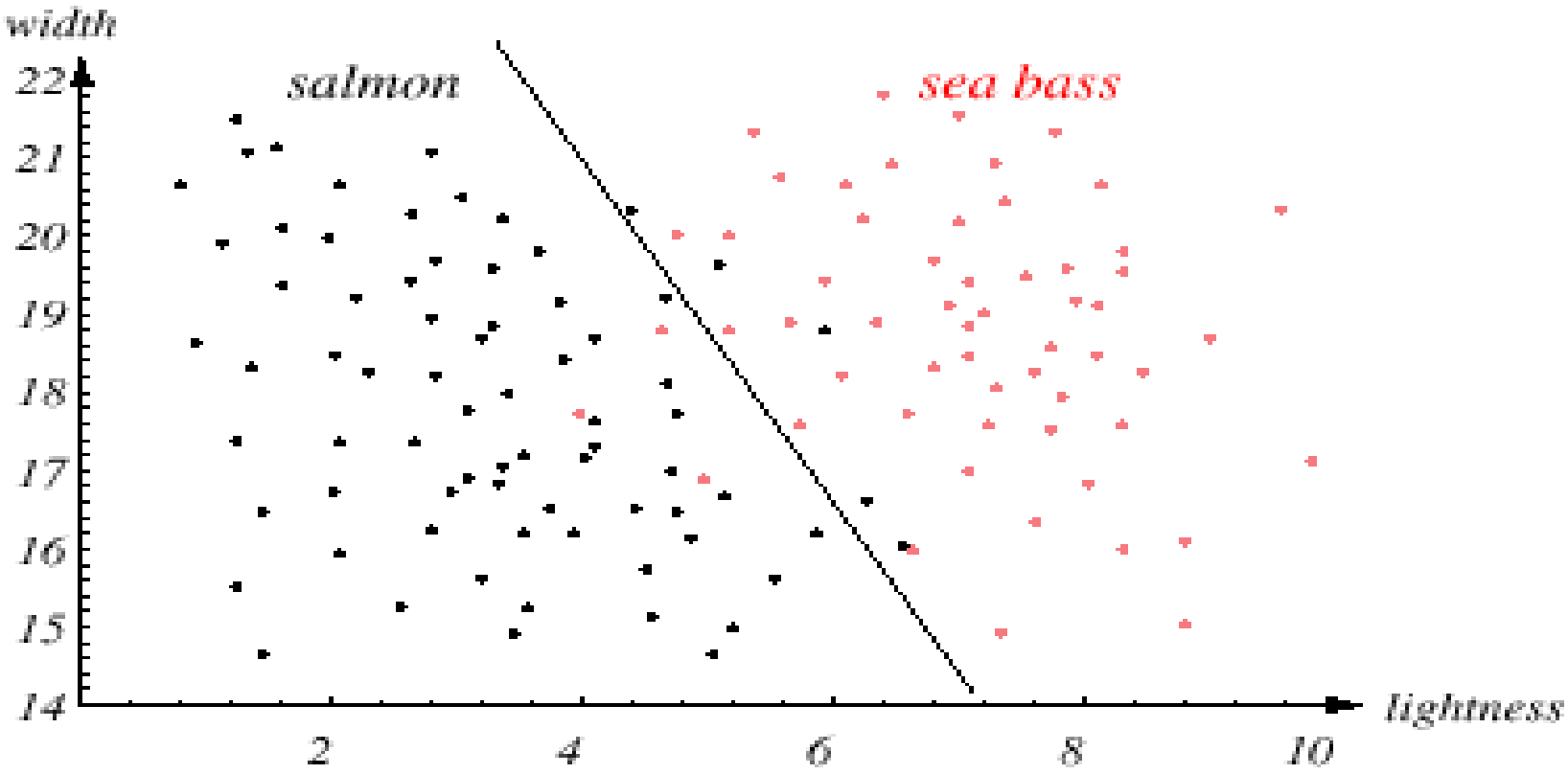
❖ Feature vectors: A number of features

$$x_1, \ldots, x_l$$

constitute the feature vector

$$\underline{x} = \left[x_1, \ldots, x_l\right]^T \in R^l$$

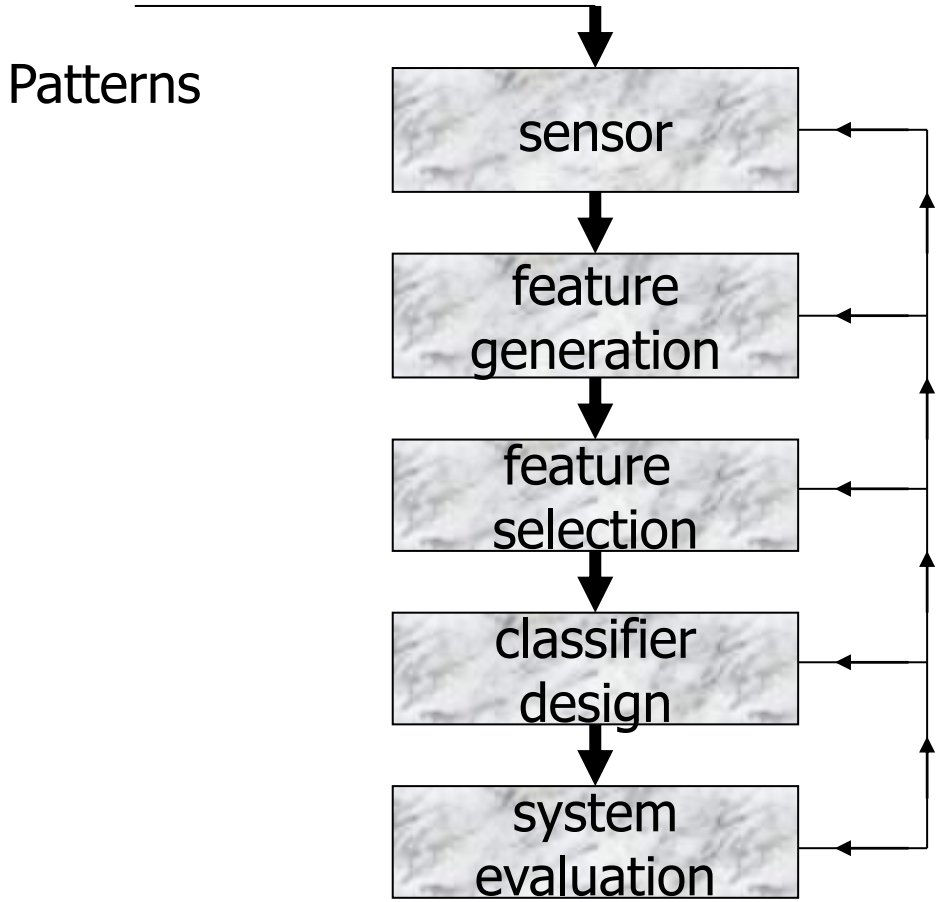
Feature vectors are treated as random vectors.



❖The two features of lightness and width for sea bass and salmon. The dark line might serve as a decision boundary of our classifier. Overall classification error on the data shown is lower than if we use only one feature as in Fig. 1.3, but there will still be some errors.

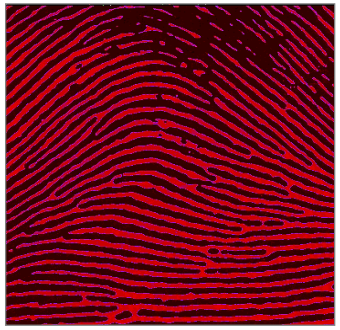
- The classifier consists of a set of functions, whose values, computed at $\underline{x}$, determine the class to which the corresponding pattern belongs

- Classification system overview



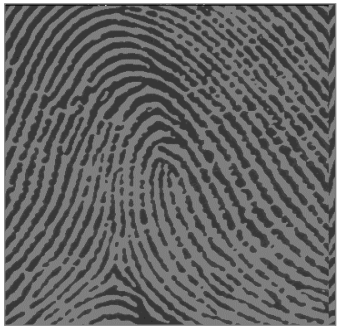
❖Fingerprint Classification

- ❖ Assign fingerprints into one of pre-specified types



❖Plain Arch

❖Tented Arch



❖Right Loop

❖Left Loop

❖Accidental

❖Pocket Whorl

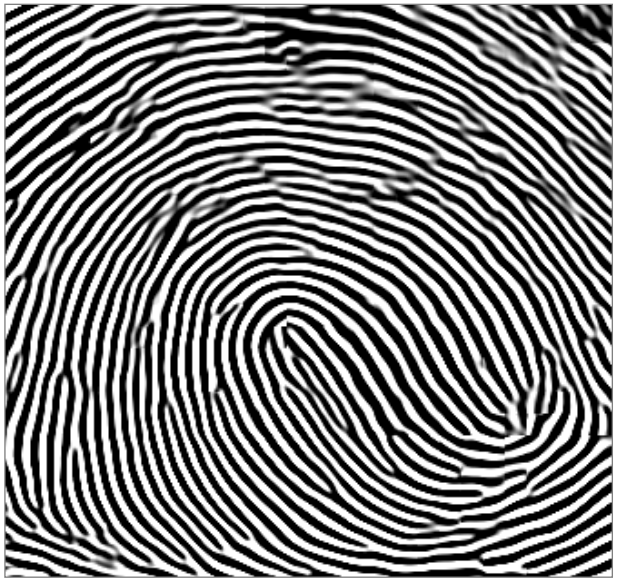
❖Plain Whorl

❖Double Loop

❖Fingerprint Enhancement

- To address the problem of poor quality fingerprints

❖Noisy image



❖Enhanced image

The Design Cycle

- Data collection
- Feature Choice
- Model Choice
- Training
- Evaluation
- Computational Complexity

- Supervised – unsupervised pattern recognition: The two major directions
  - **Supervised**: Patterns whose class is known a-priori are used for training.
  - **Unsupervised**: The number of classes is (in general) unknown and no training patterns are available.

  - **Reinforcement Learning**
    - In *reinforcement learning* or *learning with a critic*, no desired category signal is given; critic instead, the only teaching feedback is that the tentative category is right or wrong.
    - This is analogous to a critic who merely states that something is right or wrong, but does not say specifically *how* it is wrong.

CLASSIFIERS BASED ON BAYES DECISION THEORY

- Statistical nature of feature vectors

$$\underline{x} = [x_1, x_2, \ldots, x_l]^T$$

- Assign the pattern represented by feature vector $\underline{x}$ to the most probable of the available classes

$$\omega_1, \omega_2, \ldots, \omega_M$$

That is $\underline{x} \rightarrow \omega_i \quad : \quad \underset{\text{maximum}}{P(\omega_i \mid \underline{x})}$

- Computation of a-posteriori probabilities
  - Assume known
    - a-priori probabilities

$$P(\omega_1), P(\omega_2)..., P(\omega_M)$$

- $$p(\underline{x}|\omega_i), \quad i = 1, 2...M$$

This is also known as the likelihood of

$$\underline{x} \;\; w.r. \;\; to \;\; \omega_i.$$

- The Bayes rule (M=2)

$$p(\underline{x})P(\omega_i \,|\, \underline{x}) = p(\underline{x} \,|\, \omega_i)P(\omega_i) \Rightarrow$$

$$P(\omega_i \mid \underline{x}) = \frac{p(\underline{x} \,|\, \omega_i)P(\omega_i)}{p(\underline{x})} \leftrightarrow \text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

where

$$p(\underline{x}) = \sum_{i=1}^{2} p(\underline{x} \,|\, \omega_i)P(\omega_i)$$

❖ The Bayes classification rule (for two classes M=2)

- Given $\underline{x}$ classify it according to the rule

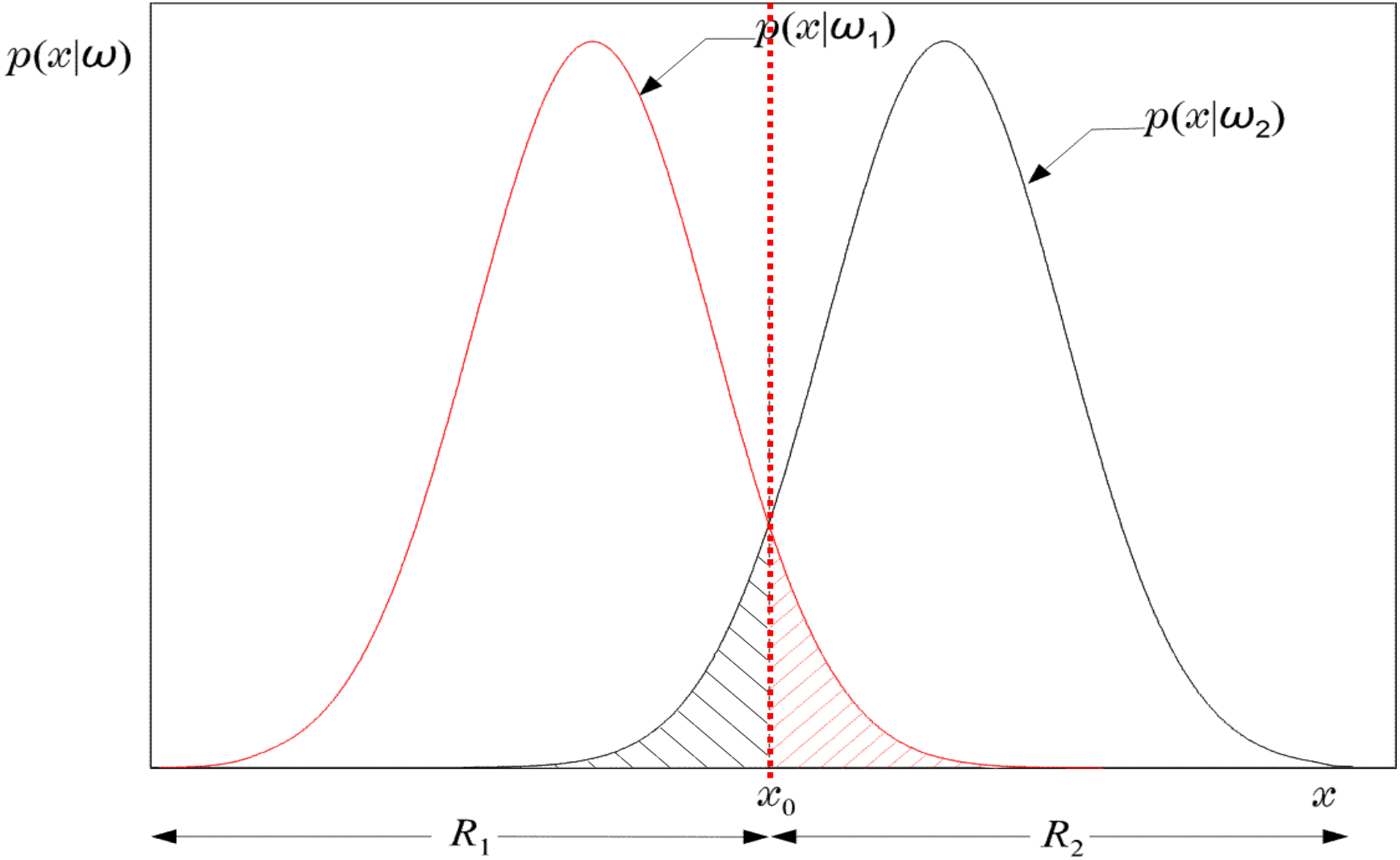
$$\boxed{\begin{aligned} &If\ P(\omega_1|\underline{x}) > P(\omega_2|\underline{x})\ \ \underline{x} \rightarrow \omega_1 \\ &If\ P(\omega_2|\underline{x}) > P(\omega_1|\underline{x})\ \ \underline{x} \rightarrow \omega_2 \end{aligned}}$$

- Equivalently: classify $\underline{x}$ according to the rule

$$\boxed{p(\underline{x}|\omega_1)P(\omega_1)(><)p(\underline{x}|\omega_2)P(\omega_2)}$$

- For equiprobable classes the test becomes

$$\boxed{p(\underline{x}|\omega_1)(><)P(\underline{x}|\omega_2)}$$



$$R_1(\rightarrow \omega_1) \ and \ R_2(\rightarrow \omega_2)$$

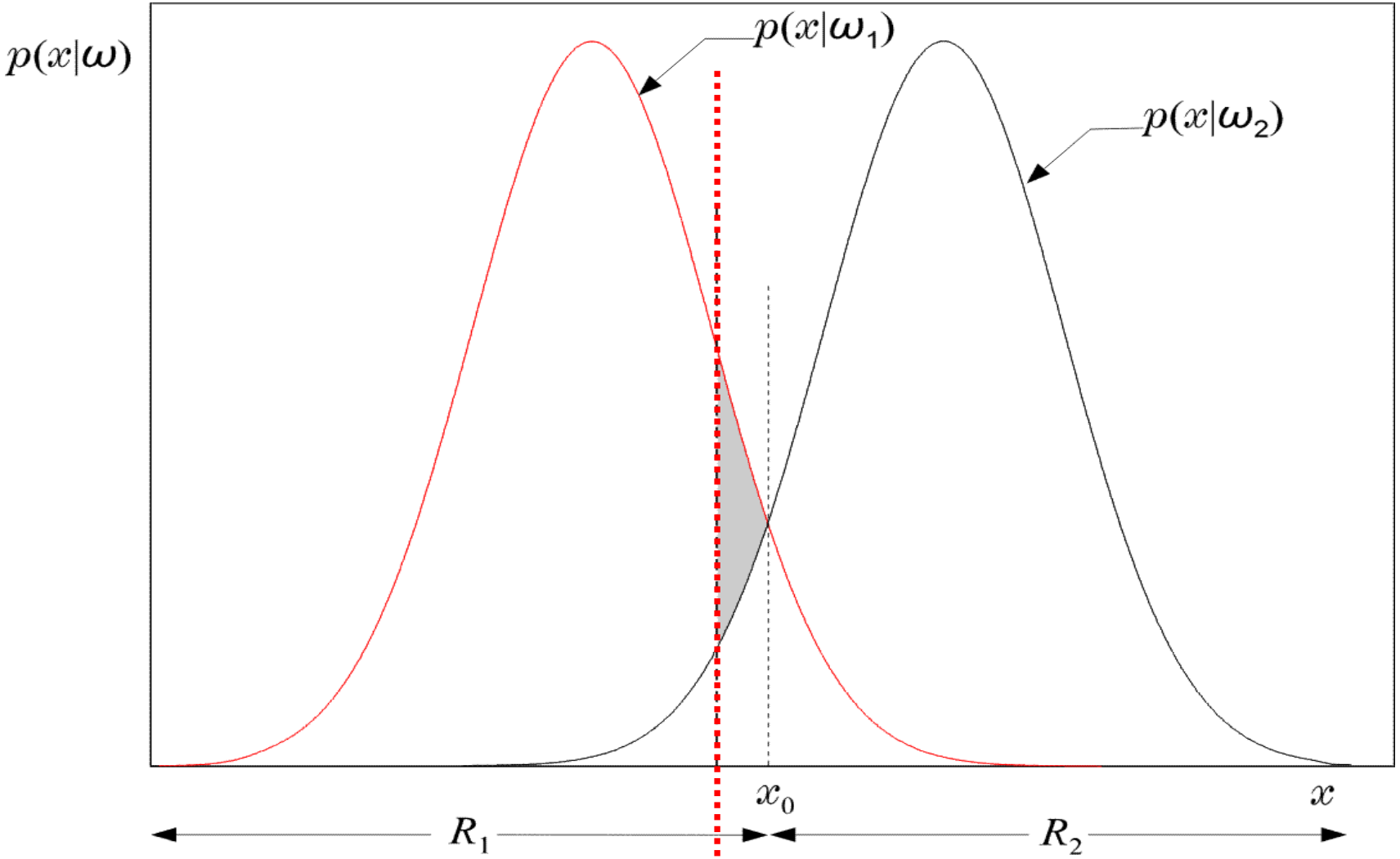
- Equivalently in words: Divide space in two regions

$$\boxed{\begin{array}{l} \text{If } \underline{x} \in R_1 \Rightarrow \underline{x} \text{ in } \omega_1 \\ \text{If } \underline{x} \in R_2 \Rightarrow \underline{x} \text{ in } \omega_2 \end{array}}$$

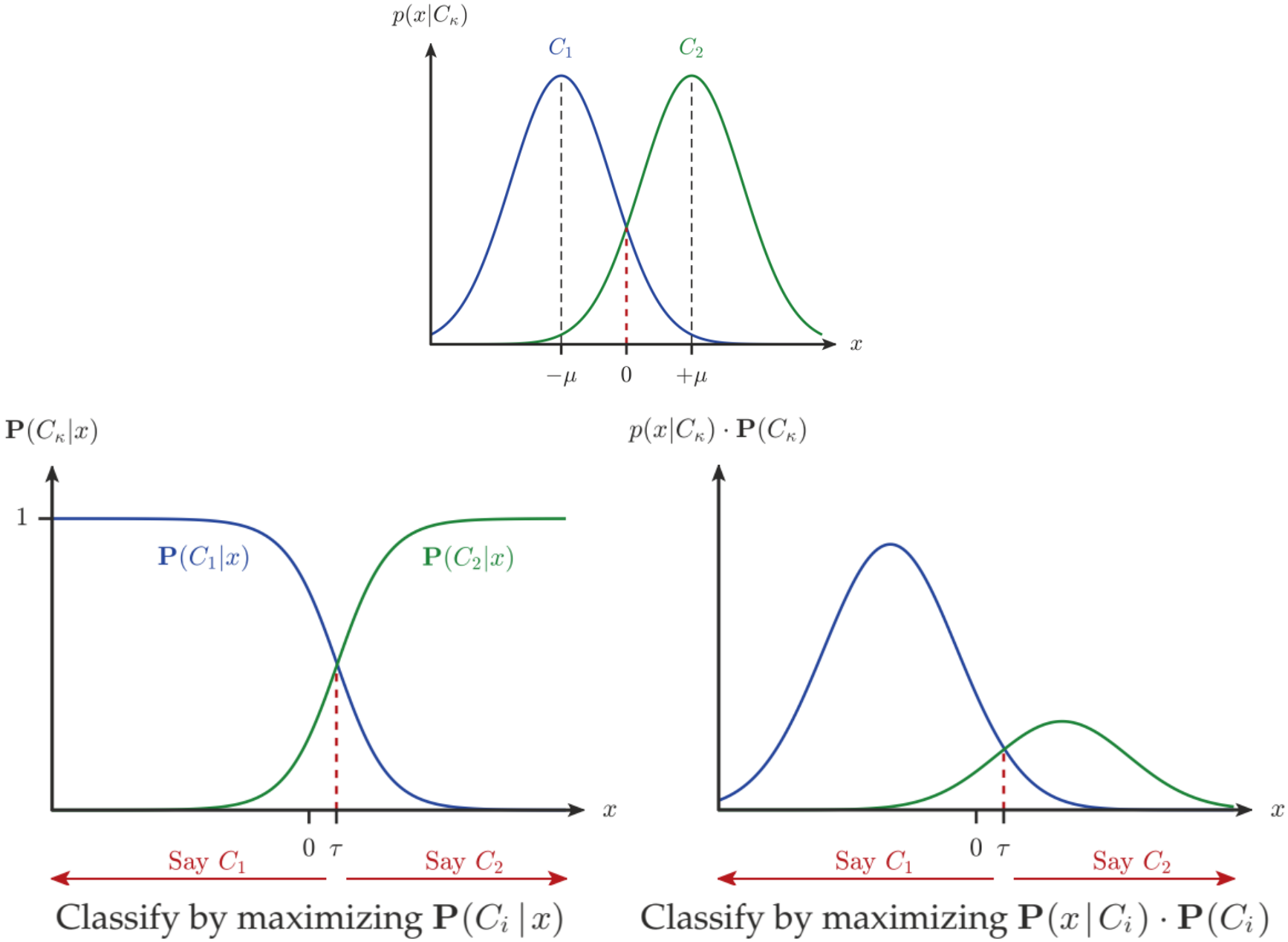
- Probability of error
  - Total shaded area
  - $$\begin{aligned} P(error) &= P(\mathbf{x} \in \mathcal{R}_2, \omega_1) + P(\mathbf{x} \in \mathcal{R}_1, \omega_2) \\ &= P(\mathbf{x} \in \mathcal{R}_2 | \omega_1) P(\omega_1) + P(\mathbf{x} \in \mathcal{R}_1 | \omega_2) P(\omega_2) \end{aligned}$$

$$P_e = P(\omega_1) \int_{x_0}^{+\infty} p(x|\omega_1) dx + P(\omega_2) \int_{-\infty}^{x_0} p(x|\omega_2) dx$$

- Bayesian classifier is OPTIMAL with respect to minimizing the classification error probability!!!!



- Indeed: Moving the threshold the total shaded area INCREASES by the extra "grey" area.



An Introduction to Pattern Recognition and Machine Learning-Paul Fieguth(2022)

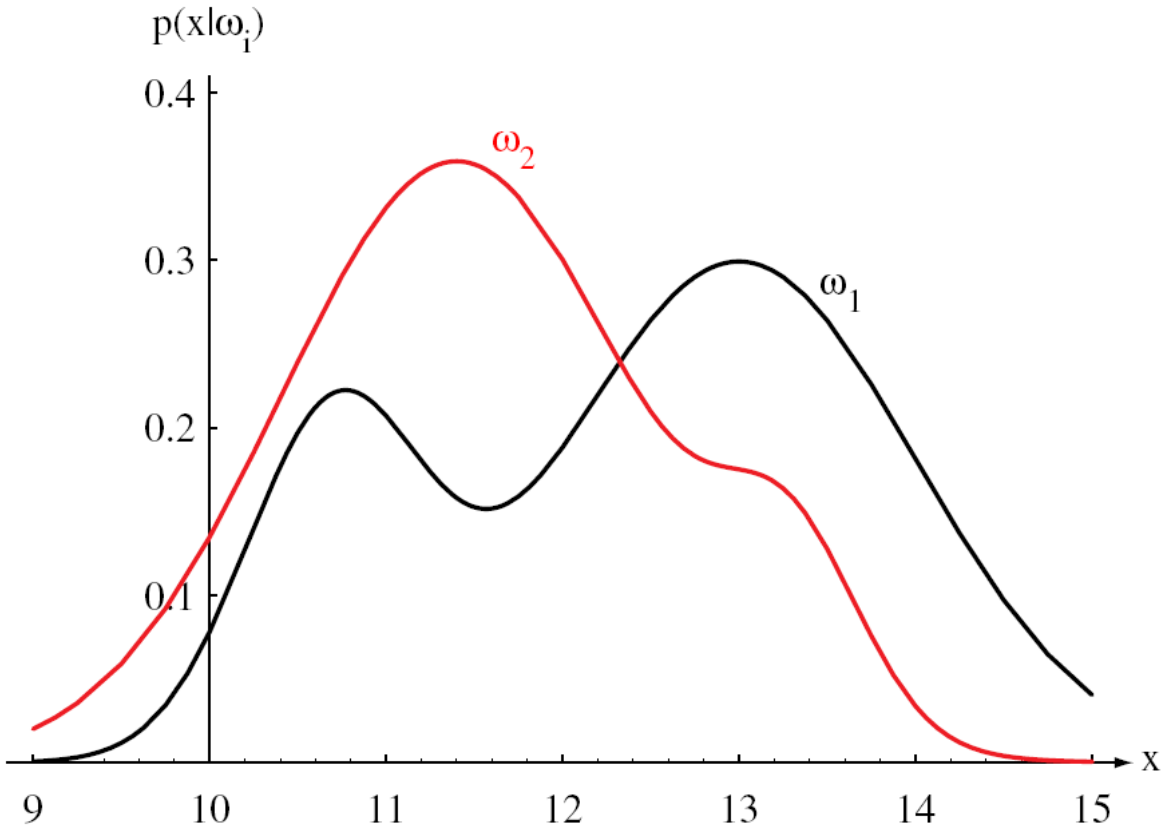


Figure 2.1: Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value *x* given the pattern is in category ω_i. If *x* represents the length of a fish, the two curves might describe the difference in length of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0.

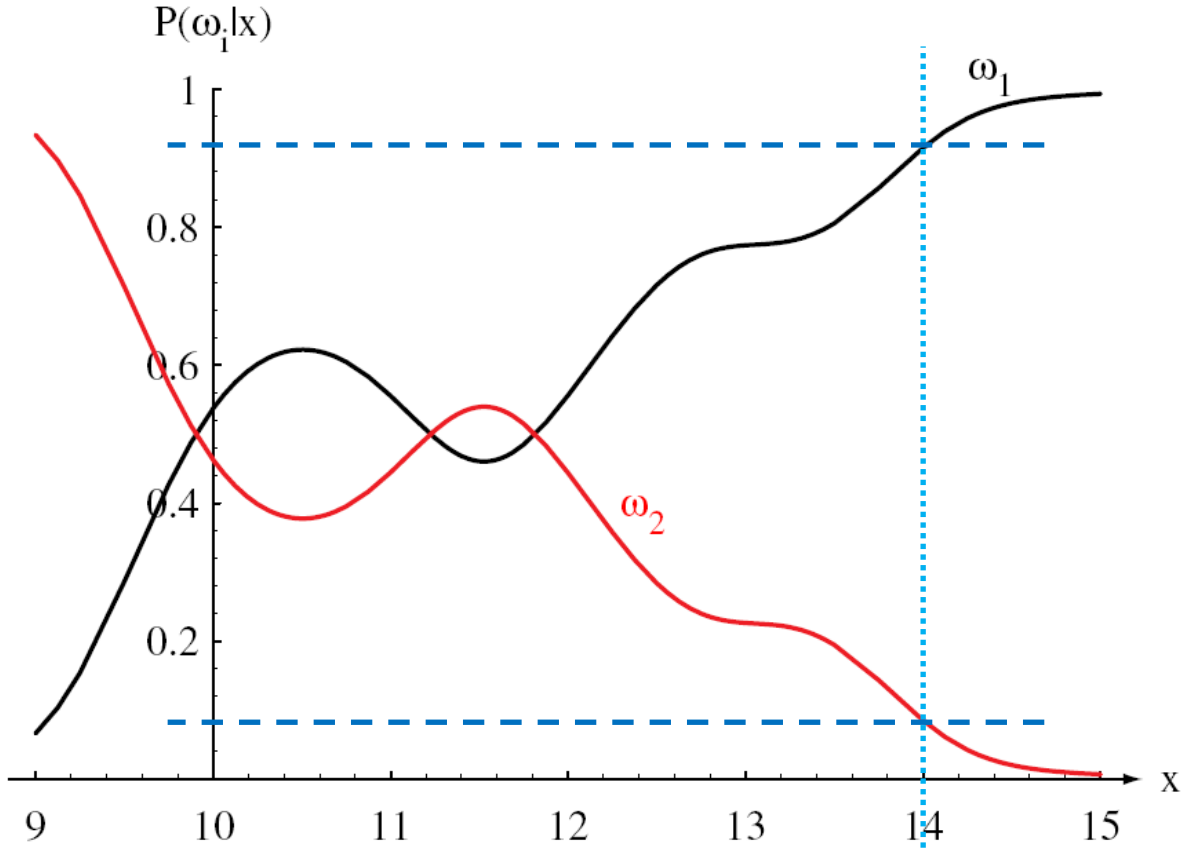


Figure 2.2: Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0

- The Bayes classification rule for many (M>2) classes:
  - Given $\underline{X}$ classify it to ω_i if:

$$\boxed{P(\omega_i|\underline{x}) > P(\omega_j|\underline{x}) \quad \forall j \neq i}$$

  - Such a choice also minimizes the classification error probability

- Minimizing the average risk
  - For each wrong decision, a penalty term is assigned since some decisions are more sensitive than others

- For M=2
  - Define the loss matrix

$$L = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix}$$

  - λ_{12} penalty term for deciding class ω_2, although the pattern belongs to ω_1, etc.

- Risk with respect to ω_1

$$r_1 = \lambda_{11} \int_{R_1} p(\underline{x}|\omega_1) d\underline{x} + \boxed{\lambda_{12} \int_{R_2} p(\underline{x}|\omega_1) d\underline{x}}$$

- Risk with respect to ω_2

$$r_2 = \boxed{\lambda_{21} \int_{R_1} p(\underline{x}|\omega_2) d\underline{x}} + \lambda_{22} \int_{R_2} p(\underline{x}|\omega_2) d\underline{x}$$

- $\boxed{} \Longrightarrow$ Probabilities of wrong decisions, weighted by the penalty terms

- Average risk $r = r_1 P(\omega_1) + r_2 P(\omega_2)$

Typically, $\lambda_{11} = \lambda_{22} = 0$

$$\text{Assign } \boldsymbol{x} \text{ to } \omega_1 \ (\omega_2) \text{ if}: \underbrace{\lambda_{12} P(\omega_1)}_{P'(\omega_1)} p(\boldsymbol{x}|\omega_1) > (<) \ \underbrace{\lambda_{21} P(\omega_2)}_{P'(\omega_2)} p(\boldsymbol{x}|\omega_2).$$

M-class problem

The risk or loss associated with ω_k is defined as

$$r_k = \sum_{i=1}^{M} \lambda_{ki} \int_{R_i} p(\underline{x} \mid \omega_k) d\,\underline{x}$$

the average risk

$$r = \sum_{k=1}^{M} r_k P(\omega_k) = \sum_{i=1}^{M} \int_{R_i} \left(\sum_{k=1}^{M} \lambda_{ki}\, p(\underline{x} \mid \omega_k) P(\omega_k) \right) d\,\underline{x}$$

is minimized. Then we have:

$$\underline{x} \in R_i \text{ if } l_i \equiv \sum_{k=1}^{M} \lambda_{ki}\, p(\underline{x} \mid \omega_k) P(\omega_k) < l_j \equiv \sum_{k=1}^{M} \lambda_{kj}\, p(\underline{x} \mid \omega_k) P(\omega_k) \quad \forall j \neq i$$

- If $\lambda_{ki} = 1 - \delta_{ki}$, where δ_{ki} is Kronecker's delta (0 if $k \neq i$ and 1 if $k = i$), then minimizing the average risk becomes equivalent to minimizing the classification error probability.

For M=2

- Choose R_1 and R_2 so that r is minimized
-
- Then assign $\underline{x}$ to ω_i if $l_1 < l_2$ or

$$\lambda_{11} p(\underline{x}|\omega_1)P(\omega_1) + \lambda_{21} p(\underline{x}|\omega_2)P(\omega_2) <$$

$$\lambda_{12} p(\underline{x}|\omega_1)P(\omega_1) + \lambda_{22} p(\underline{x}|\omega_2)P(\omega_2)$$

- Equivalently:
  assign $\underline{x}$ to $\omega_1(\omega_2)$ if

$$\boxed{l_{12} \equiv \frac{p(\underline{x}|\omega_1)}{p(\underline{x}|\omega_2)} > (<) \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}}$$

l_{12} :likelihood ratio

❖ If

$$P(\omega_1) = P(\omega_2) = \frac{1}{2} \text{ and } \lambda_{11} = \lambda_{22} = 0$$

$$\boxed{\begin{aligned} &\underline{x} \to \omega_1 \ \text{ if } \ p(\underline{x} \mid \omega_1) > p(\underline{x} \mid \omega_2) \frac{\lambda_{21}}{\lambda_{12}} \\ &\underline{x} \to \omega_2 \ \text{ if } \ p(\underline{x} \mid \omega_2) > p(\underline{x} \mid \omega_1) \frac{\lambda_{12}}{\lambda_{21}} \end{aligned}}$$

if $\lambda_{21} = \lambda_{12} \Rightarrow$ Minimum classification error probability

❖ An example:

- $p(x|\omega_1) = \frac{1}{\sqrt{\pi}} \exp(-x^2)$
- $p(x|\omega_2) = \frac{1}{\sqrt{\pi}} \exp(-(x-1)^2)$
- $P(\omega_1) = P(\omega_2) = \frac{1}{2}$
- $L = \begin{pmatrix} 0 & 0.5 \\ 1.0 & 0 \end{pmatrix}$

- Then the threshold value is:

$$x_0 \text{ for minimum } P_e :$$

$$x_0 : \exp(-x^2) = \exp(-(x-1)^2) \Rightarrow$$

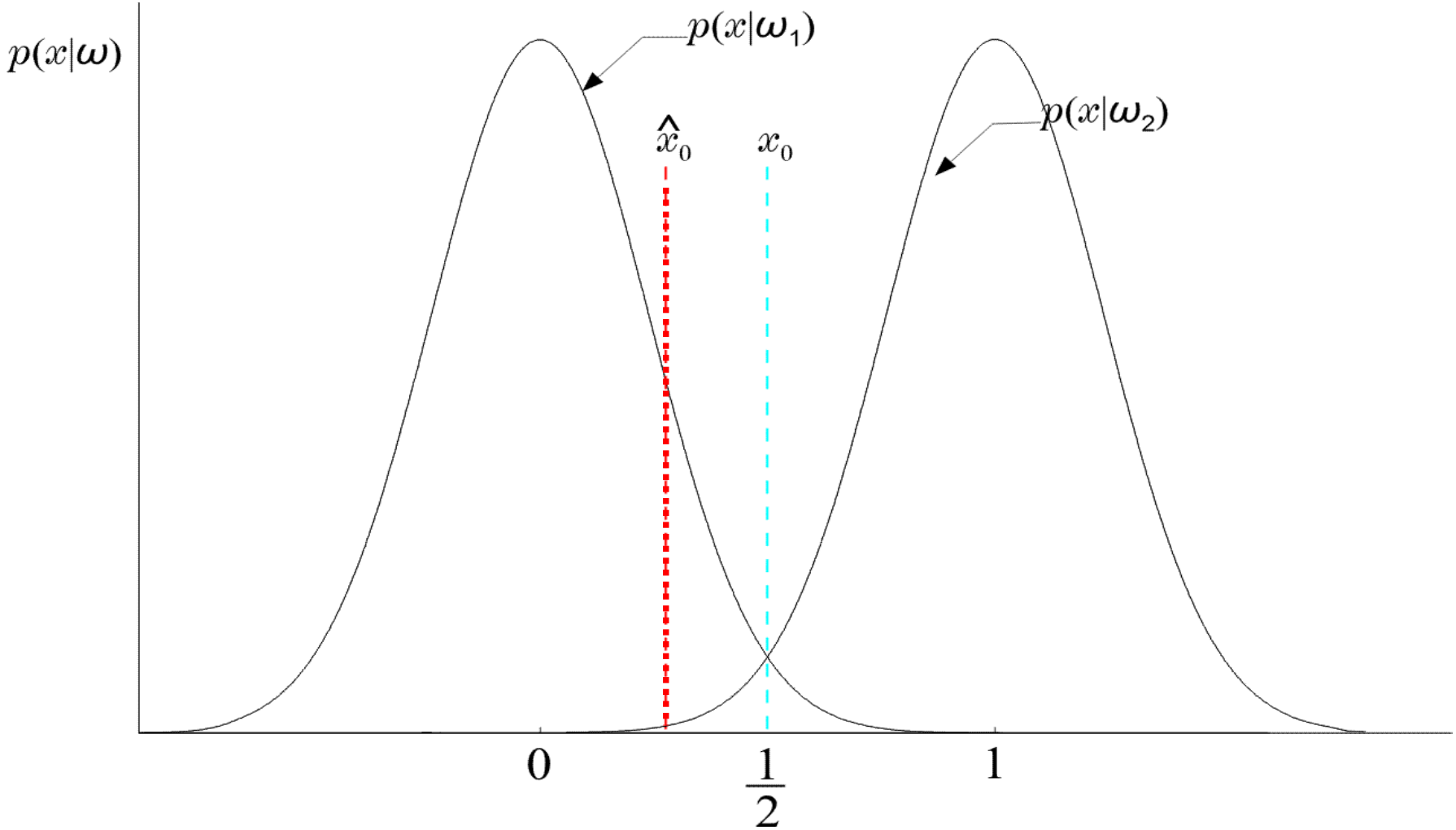
$$x_0 = \frac{1}{2}$$

- Threshold $\hat{x}_0$ for minimum r

$$\hat{x}_0 : \quad \exp(-x^2) = 2\exp(-(x-1)^2) \Rightarrow$$

$$\hat{x}_0 = \frac{(1-\ell n 2)}{2} = 0.1534 < \frac{1}{2}$$

Thus $\hat{x}_0$ moves to the left of $x_0 = \frac{1}{2}$ (WHY?)



Rejection

- Allowing actions other than classification primarily allows the possibility of rejection, i.e., of refusing to make a decision in close cases; this is a useful option if being indecisive is not too costly.

$$R = \left\{ \boldsymbol{x} | 1 - \max_i p(\omega_i | \boldsymbol{x}) > t \right\}$$

R, a *reject region*

$$A = \left\{ \boldsymbol{x} | 1 - \max_i p(\omega_i | \boldsymbol{x}) \leq t \right\}$$

A, an *acceptance* or *classification region*

where *t* is a threshold.

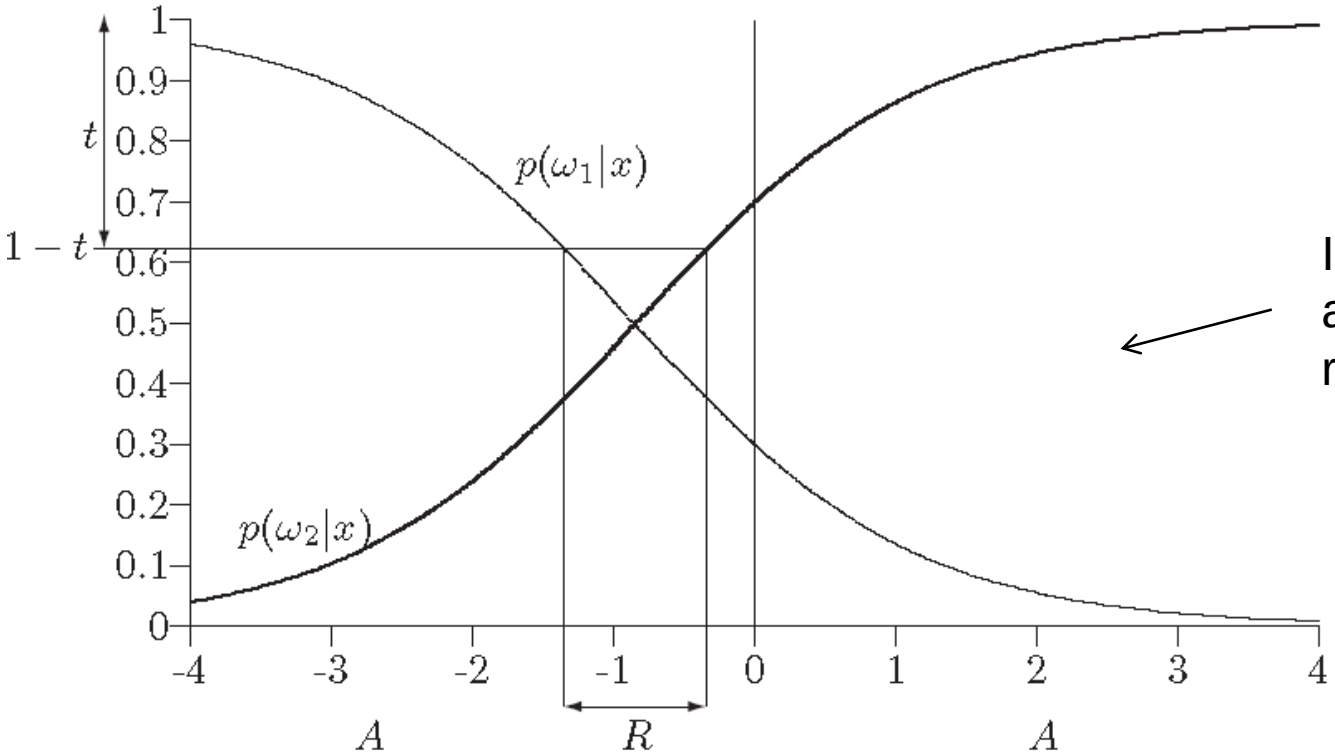


Illustration of acceptance and reject regions.

DISCRIMINANT FUNCTIONS
DECISION SURFACES

❖ If R_i, R_j are contiguous: $g(\underline{x}) \equiv P(\omega_i|\underline{x}) - P(\omega_j|\underline{x}) = 0$

$$R_i : P(\omega_i|\underline{x}) > P(\omega_j|\underline{x})$$

$+$

$$g(\underline{x}) = 0$$

$-$

$$R_j : P(\omega_j|\underline{x}) > P(\omega_i|\underline{x})$$

is the surface separating the regions. On one side is positive (+), on the other is negative (-). It is known as Decision Surface.

- If $f(.)$ monotonic, the rule remains the same if we use:

$$\underline{x} \rightarrow \omega_i \quad \text{if: } f(P(\omega_i|\underline{x})) > f(P(\omega_j|\underline{x})) \quad \forall i \neq j$$

- $g_i(\underline{x}) \equiv f(P(\omega_i|\underline{x}))$ is a **discriminant function**

- In general, discriminant functions can be defined independent of the Bayesian rule. They lead to suboptimal solutions, yet if chosen appropriately, can be computationally more tractable.

BAYESIAN CLASSIFIER FOR NORMAL DISTRIBUTIONS

❖ Univariate Gaussian pdf

$$p(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right],$$

$$\mu = E[x] \equiv \int_{-\infty}^{+\infty} x p(x) dx$$

$$\sigma^2 = E[(x-\mu)^2] \equiv \int_{-\infty}^{+\infty} (x-\mu)^2 p(x) dx$$

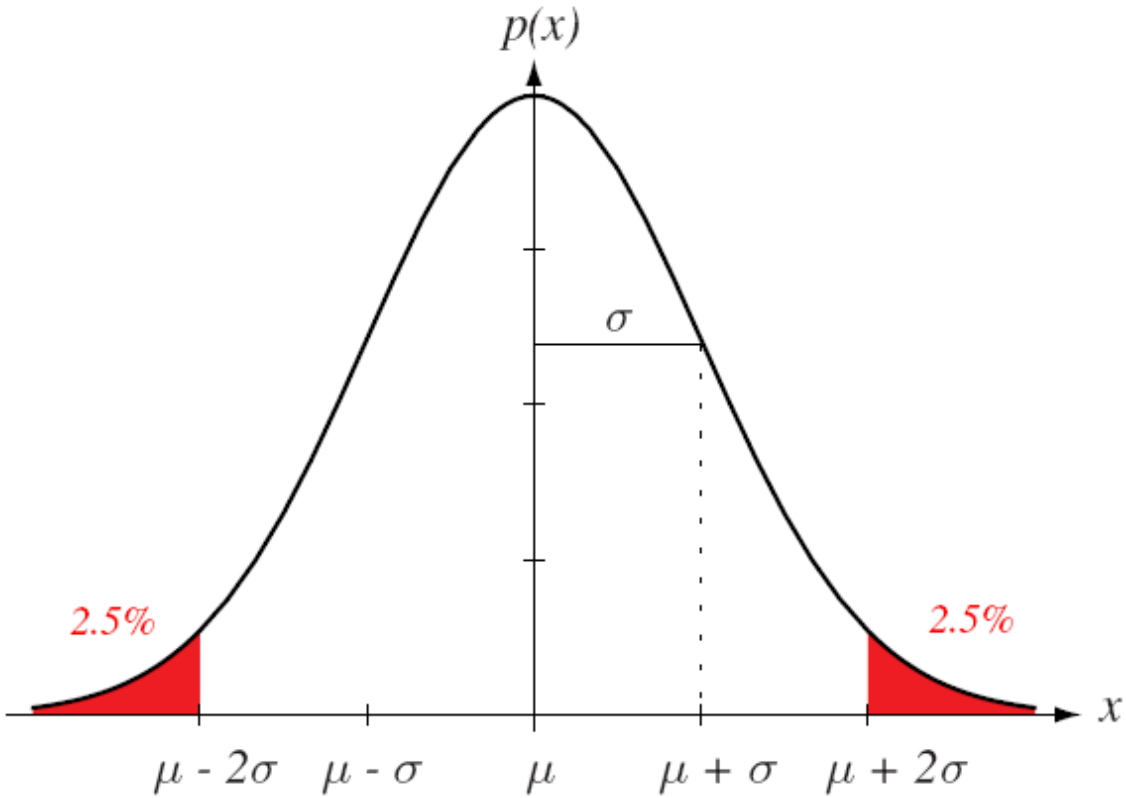


FIGURE 2.7. A univariate normal distribution has roughly 95% of its area in the range $|x - \mu| \leq 2\sigma$, as shown. The peak of the distribution has value $p(\mu) = 1/(\sqrt{2\pi}\,\sigma)$

BAYESIAN CLASSIFIER FOR NORMAL DISTRIBUTIONS

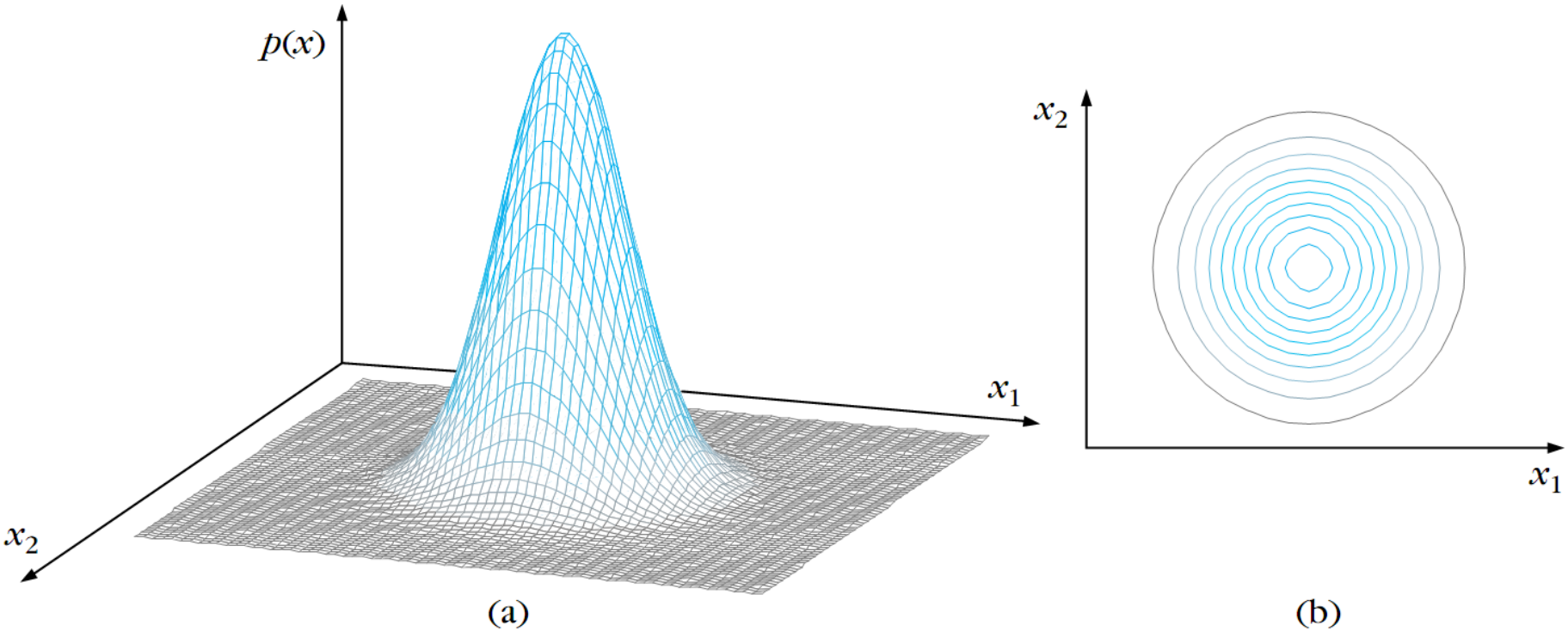
❖ Multivariate Gaussian pdf

$$p(\underline{x}|\omega_i) = \frac{1}{(2\pi)^{\frac{\ell}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\underline{x} - \underline{\mu}_i)^{\mathrm{T}} \Sigma_i^{-1} (\underline{x} - \underline{\mu}_i) \right)$$

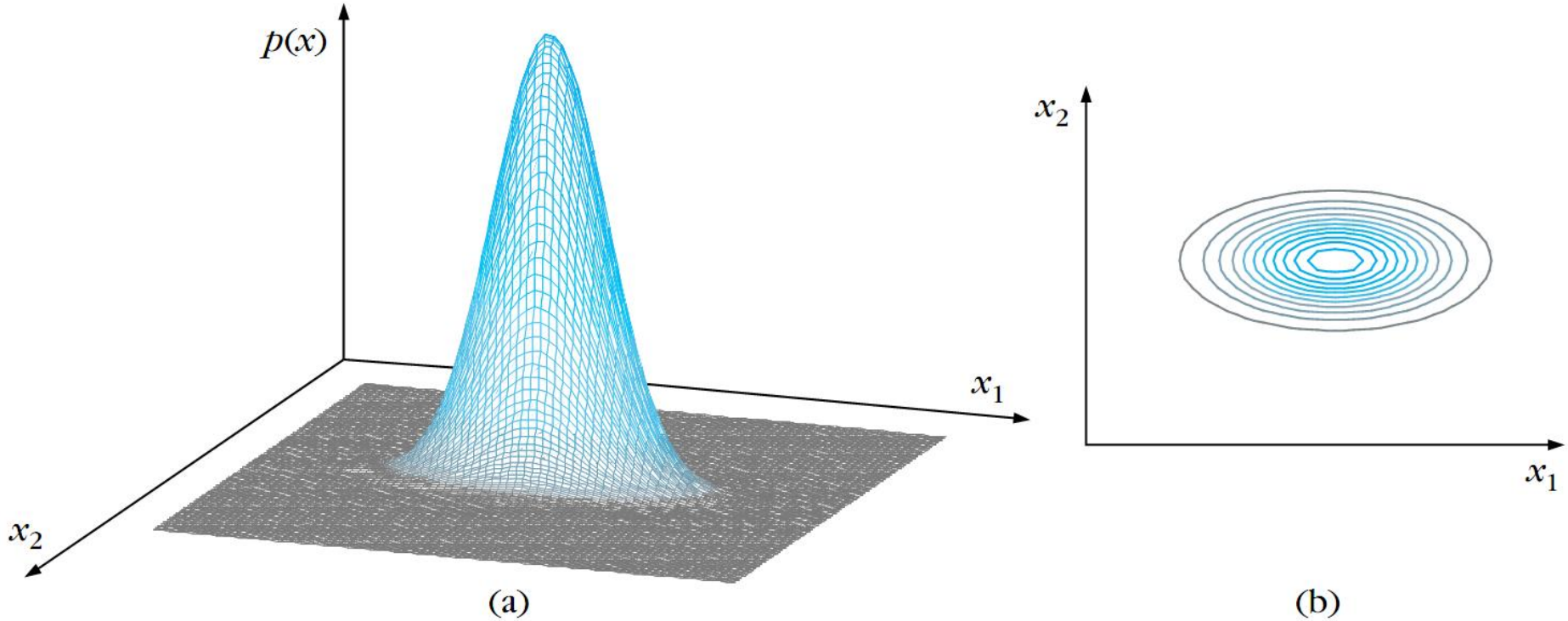
$\underline{\mu}_i = E\left[\underline{x}\right]$ $\ell \times 1$ mean vector in ω_i

$\Sigma_i = E\left[(\underline{x} - \underline{\mu}_i)(\underline{x} - \underline{\mu}_i)^{\mathrm{T}} \right]$ $\ell \times \ell$ matrix in ω_i

called covariance matrix

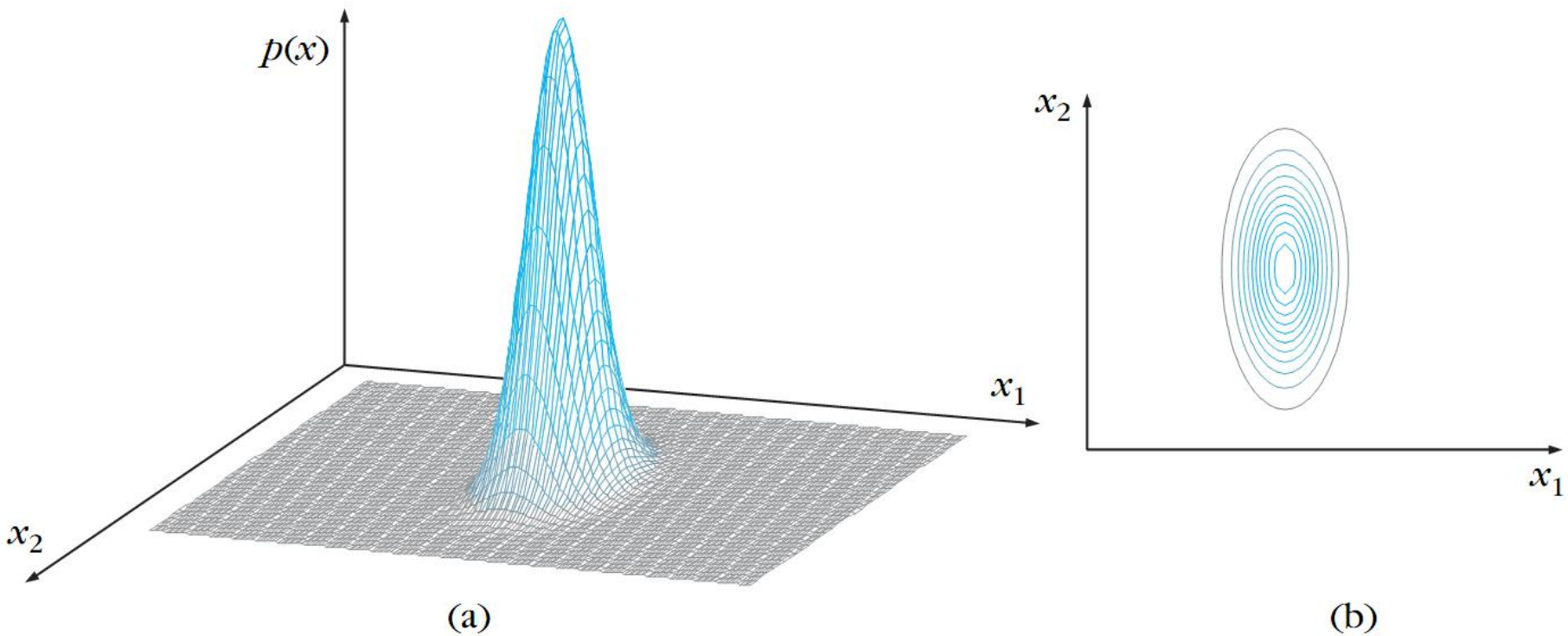


The graph of a two-dimensional Gaussian pdf and the corresponding isovalue curves for a diagonal $\mathbf{\Sigma}$ with $\sigma_1^2 = \sigma_2^2$. The graph has a spherical symmetry showing no preference in any direction.



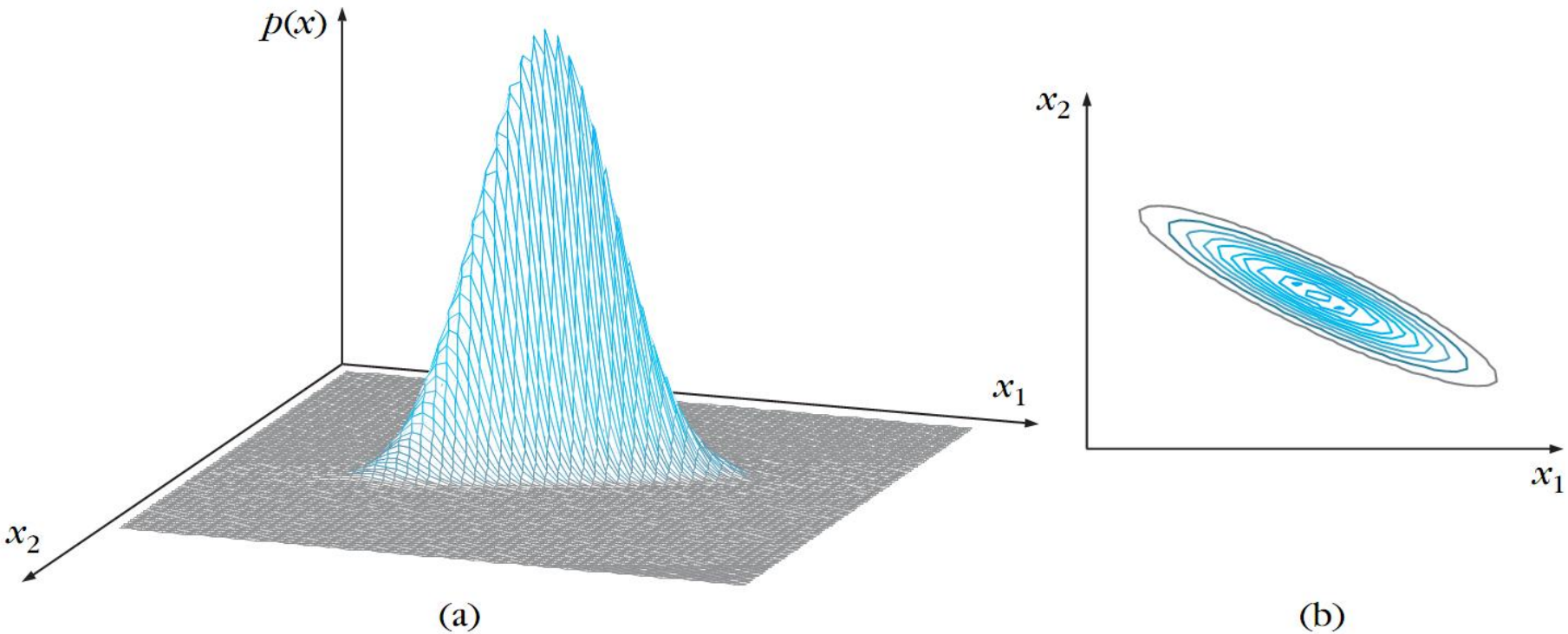
(a) The graph of a two-dimensional Gaussian pdf and

(b) the corresponding isovalue curves for a diagonal $\mathbf{\Sigma}$ with $\sigma_1^2 >> \sigma_2^2$. The graph is elongated along the x_1 direction.



(a) The graph of a two-dimensional Gaussian pdf and

(b) the corresponding isovalue curves for a diagonal $\mathbf{\Sigma}$ with $\sigma_1^2 << \sigma_2^2$. The graph is elongated along the x_2 direction.



(a) The graph of a two-dimensional Gaussian pdf and

(b) the corresponding isovalue curves for a case of a nondiagonal **Σ.** Playing with the values of the elements of **Σ** one can achieve different shapes and orientations

Properties

- The covariance matrix $\mathbf{\Sigma}$ is always symmetric and positive semidefinite.
- (A matrix $\mathbf{A}$ is pos. semidefinite if: $\mathbf{z}^t \mathbf{A}\mathbf{z} \geq 0$ for any $\mathbf{z}$.)
- In the case in which $\mathbf{\Sigma}$ is positive definite, the determinant of $\mathbf{\Sigma}$ is strictly positive.
- The diagonal elements σ_{ii} are the variances of the respective x_i (i.e., σ^2_i), and the off-diagonal elements σ_{ij} are the *covariances* of x_i and x_j .
- If x_i and x_j are *statistically independent*, $\sigma_{ij} = 0$.

- Linear combinations of jointly normally distributed random variables, independent or not, are normally distributed.
- If $\mathbf{A}$ is a l-by-k matrix and $\mathbf{y} = \mathbf{A}^t \mathbf{x}$ is a k-component vector, then $p(\mathbf{y}) \sim N(\mathbf{A}^t \boldsymbol{\mu}, \mathbf{A}^t \boldsymbol{\Sigma} \mathbf{A})$
- In the special case where $k = 1$ and $\mathbf{A}$ is a unit-length vector $\mathbf{a}$, $y = \mathbf{a}^t \mathbf{x}$ is a scalar that represents the projection of $\mathbf{x}$ onto a line in the direction of $\mathbf{a}$; in that case $\mathbf{a}^t \boldsymbol{\Sigma} \mathbf{a}$ is the variance of the projection of $\mathbf{x}$ onto $\mathbf{a}$.

$\Sigma_{\mathbf{x}} \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$, eigenvectors ($\boldsymbol{v}_i$) of distinct eigenvalues are orthognal

$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_l \end{bmatrix}$ $l \times l$ matrix consisting of l eigenvectors

$\mathbf{y} = \boldsymbol{\Phi}^{\mathrm{t}} \mathbf{x}$ use $\boldsymbol{\Phi}$ as the transformation matrix $\mathbf{A}$

$\Sigma_{\mathbf{y}} = \boldsymbol{\Phi}^{\mathrm{t}} \Sigma_{\mathbf{x}} \boldsymbol{\Phi} = \boldsymbol{\Lambda}$ covariance matrix of transformed vector

note: $(\boldsymbol{\Phi}^{\mathrm{t}})^t = \boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^{-1} = \boldsymbol{\Phi}^{\mathrm{t}}$

❖ *Whitening* Transformation

$$\mathbf{y} = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Phi}^{\mathrm{t}} \mathbf{x} = (\boldsymbol{\Phi} \boldsymbol{\Lambda}^{-1/2})^{\mathrm{t}} \mathbf{x}$$

use $\boldsymbol{\Phi} \boldsymbol{\Lambda}^{-1/2}$ as the transformation matrix $\mathbf{A}$

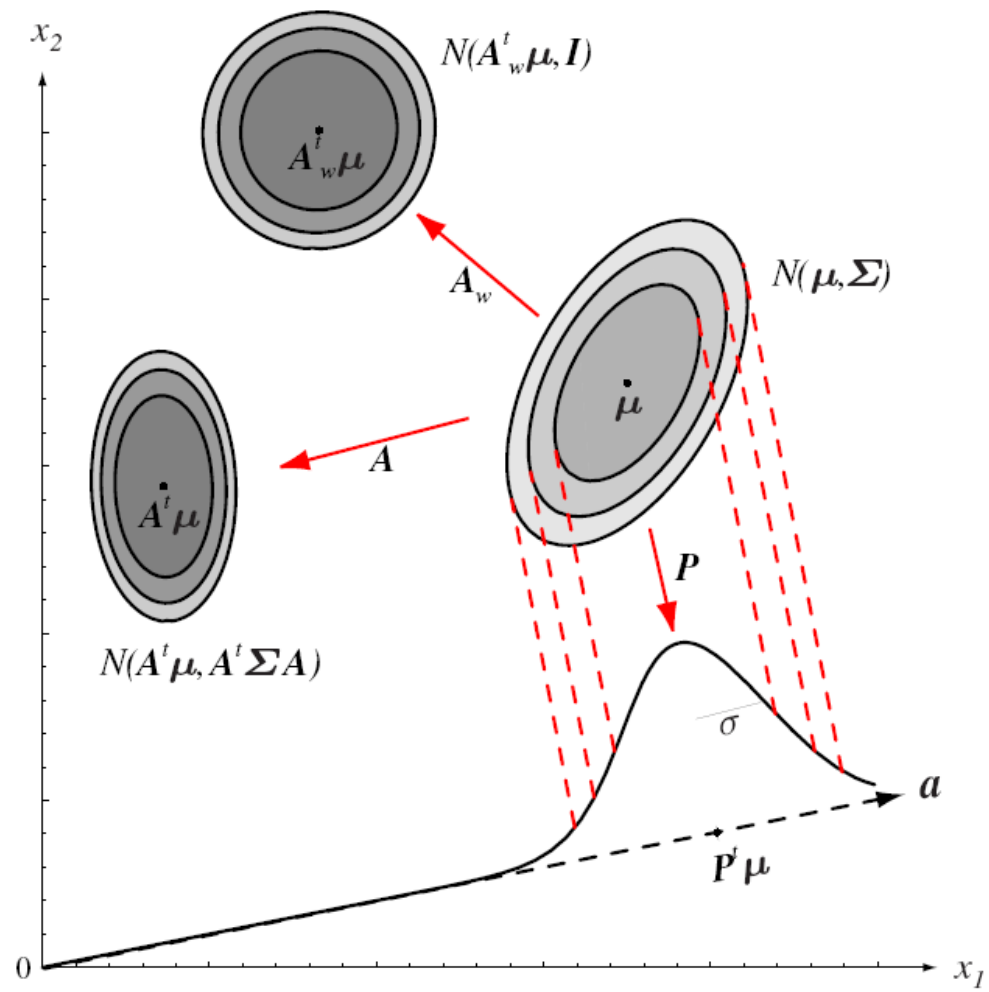
$$\Sigma_{\mathbf{y}} = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Phi}^{\mathrm{t}} \Sigma_{\mathbf{x}} \boldsymbol{\Phi} \boldsymbol{\Lambda}^{-1/2} = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Lambda} \boldsymbol{\Lambda}^{-1/2} = \mathbf{I}$$

covariance matrix of transformed vector is identity matrix

- If we define $\mathbf{\Phi}$ to be the matrix whose columns are the orthonormal eigenvectors of $\mathbf{\Sigma}$, and $\mathbf{\Lambda}$ the diagonal matrix of the corresponding eigenvalues, then the transformation $\mathbf{A}_w = \mathbf{\Phi\Lambda}^{-1/2}$ applied to the coordinates insures that the transformed distribution has covariance matrix equal to the identity matrix.
- In signal processing, the transform $\mathbf{A}_w$ is called a *whitening* transformation, since it makes the spectrum of eigenvectors of the transformed distribution uniform.

- The multivariate normal density is completely specified by $l + l(l + 1)/2$ parameters — the elements of the mean vector $\boldsymbol{\mu}$ and the independent elements of the covariance matrix $\boldsymbol{\Sigma}$.
- Samples drawn from a normal population tend to fall in a single cloud or cluster (Fig. 2.9); the center of the cluster is determined by the mean vector, and the shape of the cluster is determined by the covariance matrix.

FIG. 2.8. The action of a linear transformation on the feature space will convert an arbitrary normal distribution into another normal distribution. One transformation, **A**, takes the source distribution into distribution $N(\boldsymbol{A}^t\mu, \boldsymbol{A}^t\boldsymbol{\Sigma}\boldsymbol{A})$. Another linear transformation a projection **P** onto a line defined by vector **a** leads $N(\mu,\sigma^2)$ to measured along that line. While the transforms yield distributions in a different space, we show them superimposed on the original x_1-x_2 space. A whitening transform, $\boldsymbol{A}_w$, leads to a circularly symmetric Gaussian, here shown displaced.



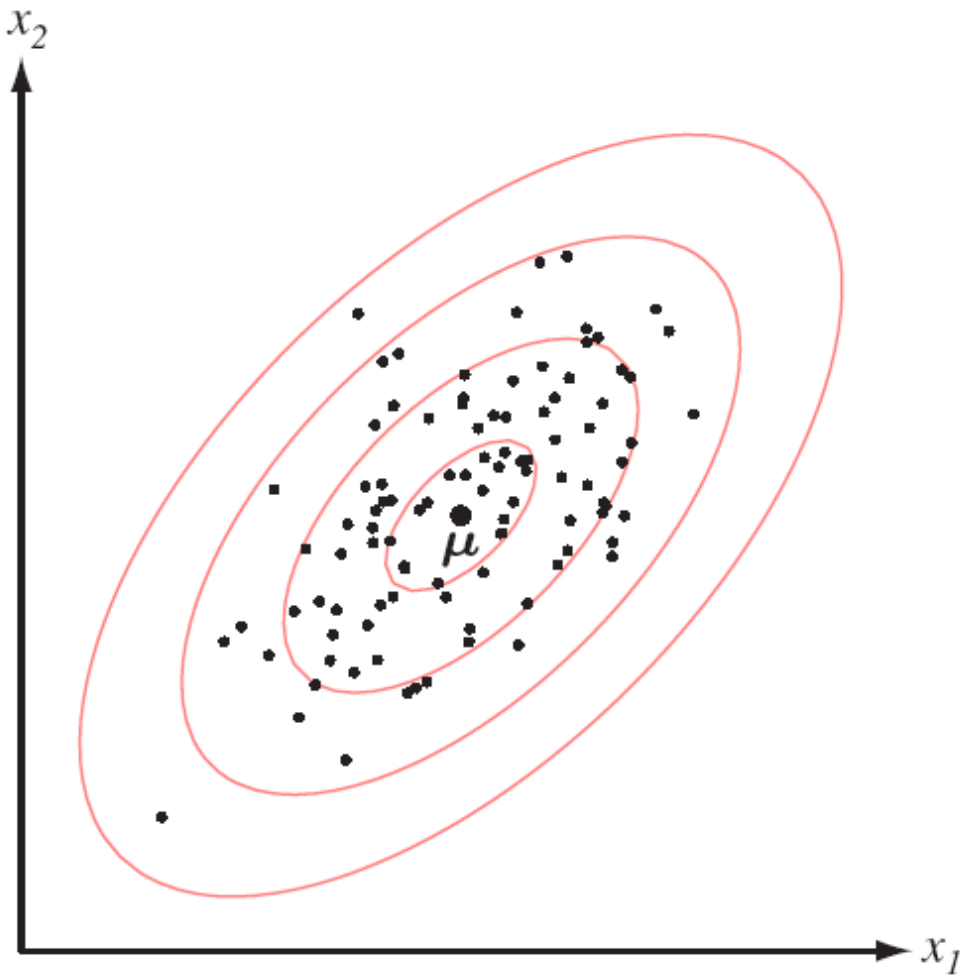
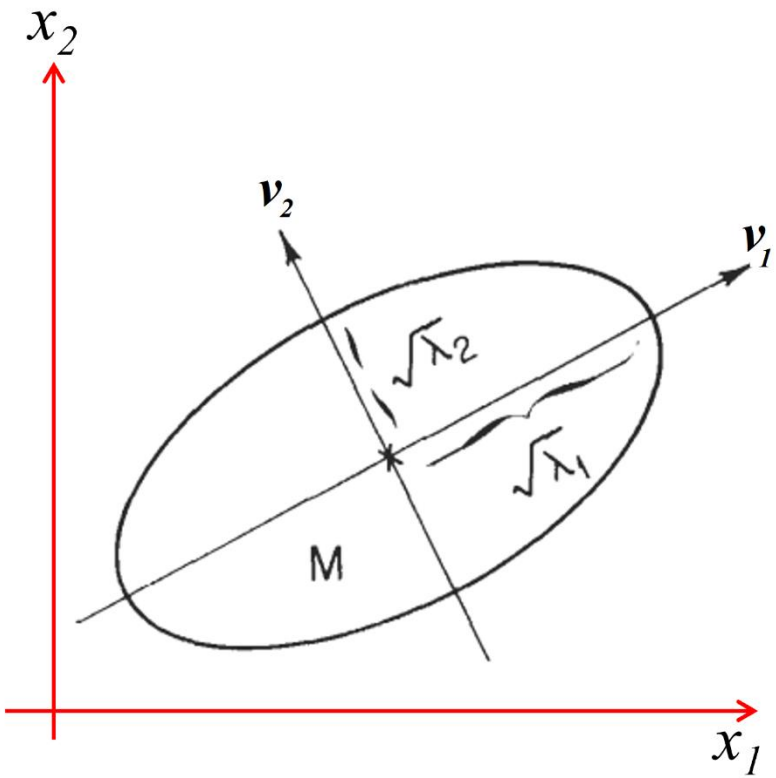


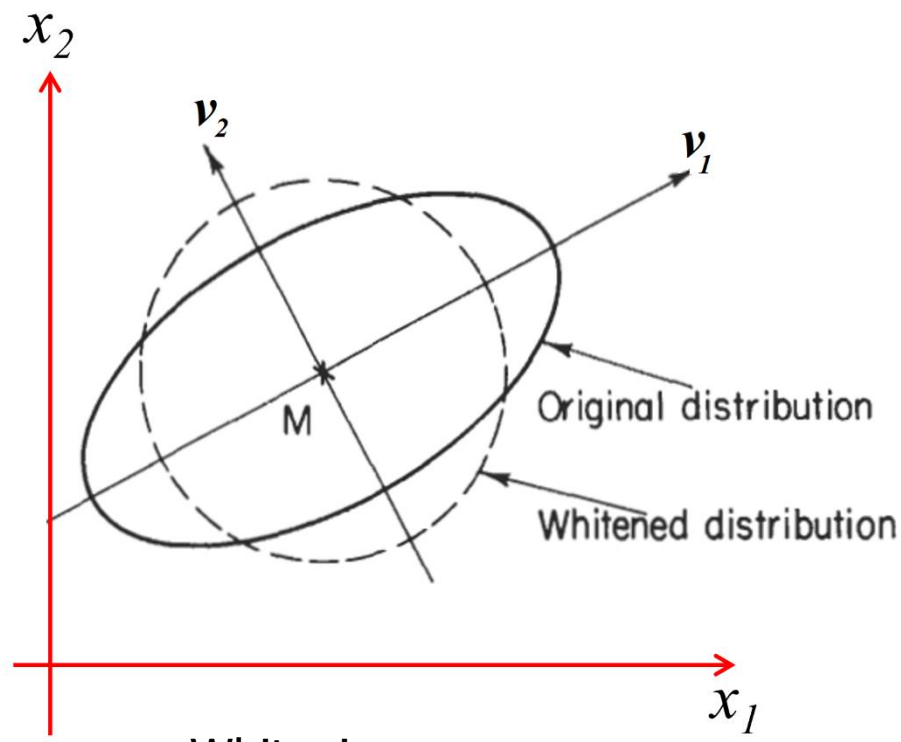
FIGURE 2.9. Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean. The ellipses show lines of equal probability density of the Gaussian.

❖eigenvectors $\boldsymbol{\varphi}_1 \; or \; \boldsymbol{v}_1$

$\boldsymbol{\varphi}_2 \; or \; \boldsymbol{v}_2$



Eigenvalues and eigenvectors of a distribution.



Whitening process

Sample generation

- To generate samples which are to be normally distributed according to a given expected vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
- From the given $\boldsymbol{\Sigma}$, find the whitening transformation of $\mathbf{y} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{\Phi}^t\mathbf{x}$. In the transformed space, $\boldsymbol{\Sigma}_\mathbf{y}=\mathbf{I}$.
- Generate N independent, normally distributed numbers for each y_i (i=1, . . . , l) with zero expected value and unit variance. Then, form N vectors $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N$.
- Transform back the generated samples to the $\mathbf{x}$-space by

$$\mathbf{x}_k = \boldsymbol{\Phi}\boldsymbol{\Lambda}^{1/2}\mathbf{y}_k \qquad (k = 1, \cdots, N).$$

- Add $\boldsymbol{\mu}$ to the samples in the $\mathbf{x}$-space as $\mathbf{x}_k + \boldsymbol{\mu}$ (k=1, ..., N).

BAYESIAN CLASSIFIER FOR NORMAL DISTRIBUTIONS

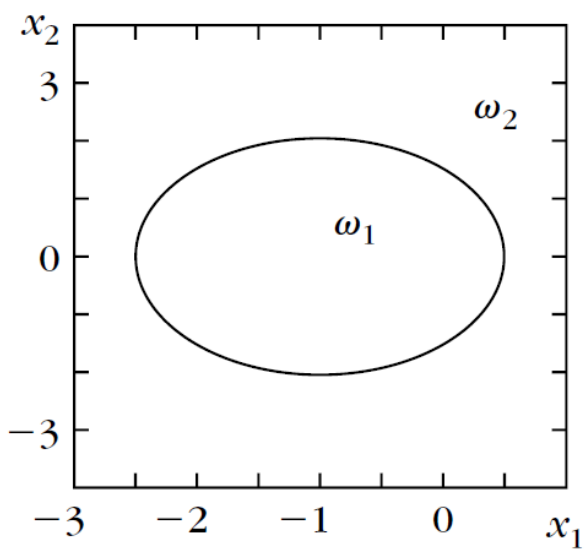
- $ln(.)$ is monotonic. Define:
-
  - $$g_i(\underline{x}) = \ln(p(\underline{x}|\omega_i)P(\omega_i)) =$$
    $$\ln p(\underline{x}|\omega_i) + \ln P(\omega_i)$$
  - $$g_i(\underline{x}) = -\frac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma_i^{-1}(\underline{x} - \underline{\mu}_i) + C_i + \ln P(\omega_i)$$
    $$C_i = -(\frac{\ell}{2})\ln 2\pi - (\frac{1}{2})\ln|\Sigma_i|$$
  - Example: $$\Sigma_i = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

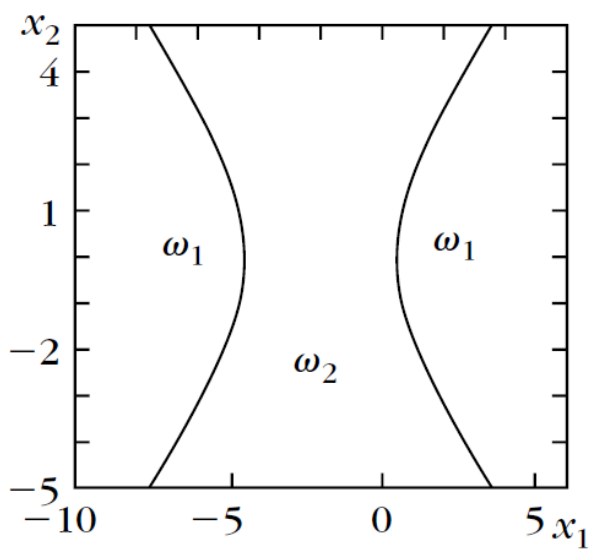
- $$g_i(\underline{x}) = -\frac{1}{2\sigma^2}(x_1^2 + x_2^2) + \frac{1}{\sigma^2}(\mu_{i1}x_1 + \mu_{i2}x_2)$$
$$-\frac{1}{2\sigma^2}(\mu_{i1}^2 + \mu_{i2}^2) + \ln P(\omega_i) + C_i$$

That is, $g_i(x)$ is quadratic and the surfaces

$$g_i(\underline{x}) - g_j(\underline{x}) = 0$$

quadrics, ellipsoids, parabolas, hyperbolas, pairs of lines. For example:





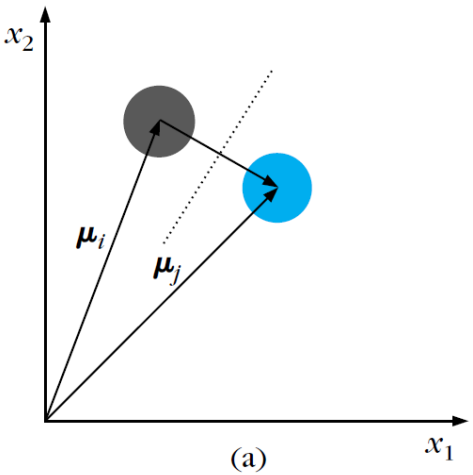
❖ Decision Hyperplanes

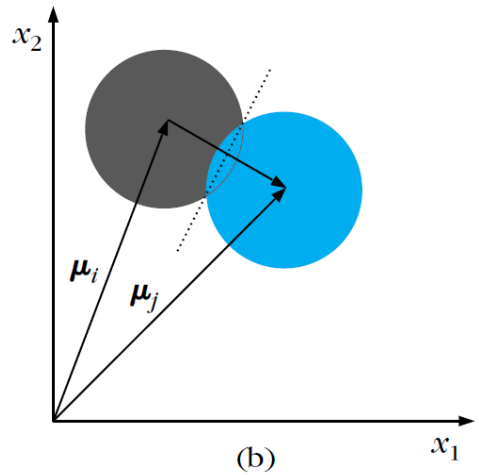
➢ Quadratic terms: $\underline{x}^T \Sigma_i^{-1} \underline{x}$

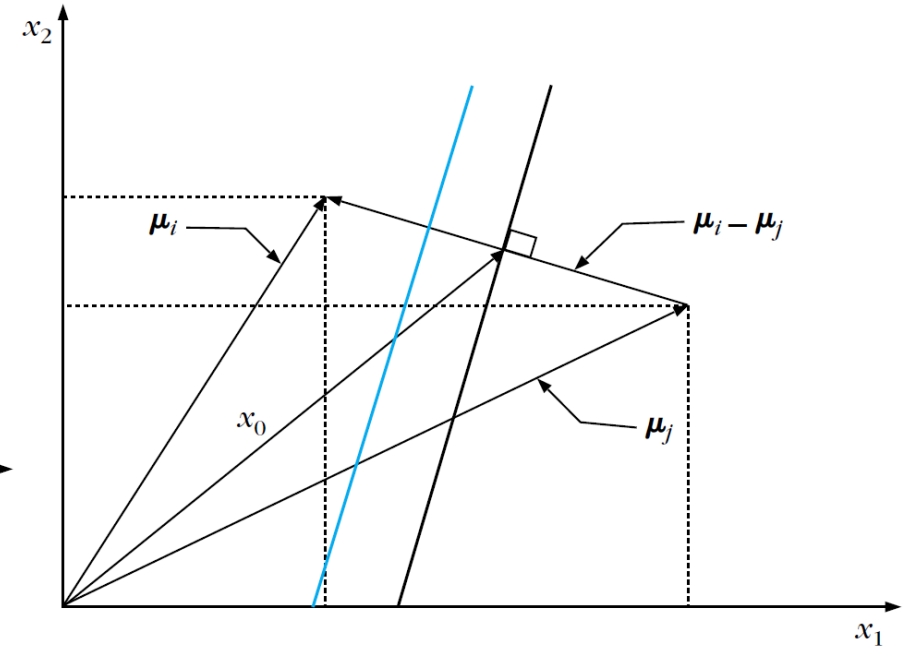
If ALL $\Sigma_i = \Sigma$ (the same) the quadratic terms are not of interest. They are not involved in comparisons. Then, equivalently, we can write:

$$\boxed{\begin{aligned} g_i(\underline{x}) &= \underline{w}_i^T \underline{x} + w_{io} \\ \underline{w}_i &= \Sigma^{-1} \underline{\mu}_i \\ w_{i0} &= \ln P(\omega_i) - \frac{1}{2} \underline{\mu}^T{}_i \Sigma^{-1} \underline{\mu}_i \end{aligned}}$$

Discriminant functions are LINEAR







Let in addition:

$$\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}. \text{ Then } \quad g_i(\underline{x}) = \frac{1}{\sigma^2} \underline{\mu}_i^T \underline{x} + w_{i0}$$

$$g_{ij}(\underline{x}) = g_i(\underline{x}) - g_j(\underline{x}) = 0 \; = \underline{w}^T (\underline{x} - \underline{x}_o), \qquad \underline{w} = \underline{\mu}_i - \underline{\mu}_j,$$

$$\underline{x}_o = \frac{1}{2}(\underline{\mu}_i + \underline{\mu}_j) - \sigma^2 \ln\left(\frac{P(\omega_i)}{P(\omega_j)}\right) \frac{\underline{\mu}_i - \underline{\mu}_j}{\left\| \underline{\mu}_i - \underline{\mu}_j \right\|^2}$$

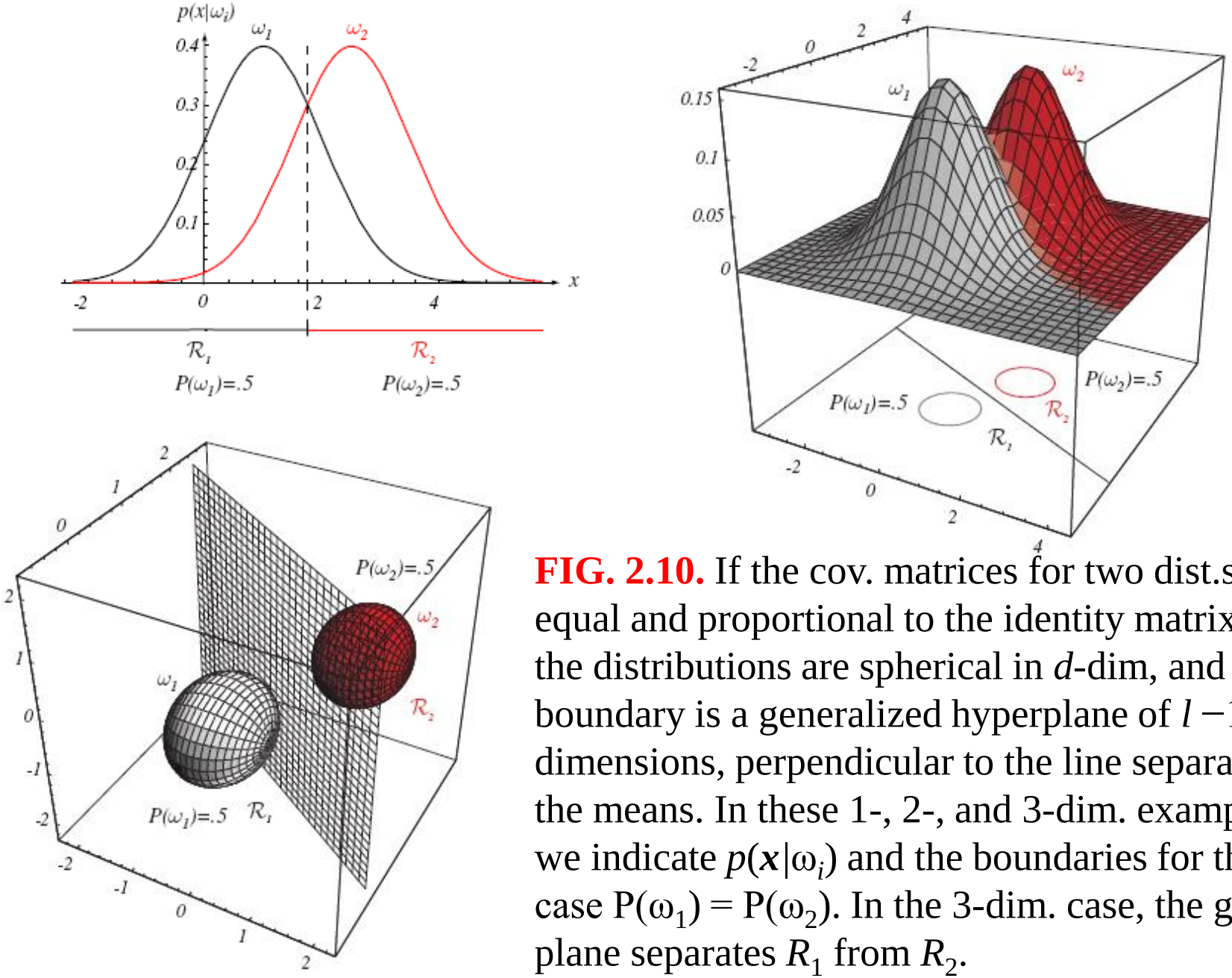


FIG. 2.10. If the cov. matrices for two dist.s are equal and proportional to the identity matrix, then the distributions are spherical in *d*-dim, and the boundary is a generalized hyperplane of $l-1$ dimensions, perpendicular to the line separating the means. In these 1-, 2-, and 3-dim. examples, we indicate $p(\boldsymbol{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the 3-dim. case, the grid plane separates R_1 from R_2.

- Nondiagonal: $\Sigma \neq \sigma^2 I$

  - $g_{ij}(\underline{x}) = \underline{w}^T(\underline{x} - \underline{x}_0) = 0$

  - $\underline{w} = \Sigma^{-1}(\underline{\mu}_i - \underline{\mu}_j)$

  - $$\underline{x}_0 = \frac{1}{2}(\underline{\mu}_i + \underline{\mu}_j) - \ell\mathrm{n}\left(\frac{P(\omega_i)}{P(\omega_j)}\right)\frac{\underline{\mu}_i - \underline{\mu}_j}{\left\|\underline{\mu}_i - \underline{\mu}_j\right\|^2_{\Sigma^{-1}}}$$

where $\|\underline{x}\|_{\Sigma^{-1}} \equiv (\underline{x}^T \Sigma^{-1} \underline{x})^{\frac{1}{2}}$

- Decision hyperplane
  - not normal to $\underline{\mu}_i - \underline{\mu}_j$
  - normal to $\Sigma^{-1}(\underline{\mu}_i - \underline{\mu}_j)$

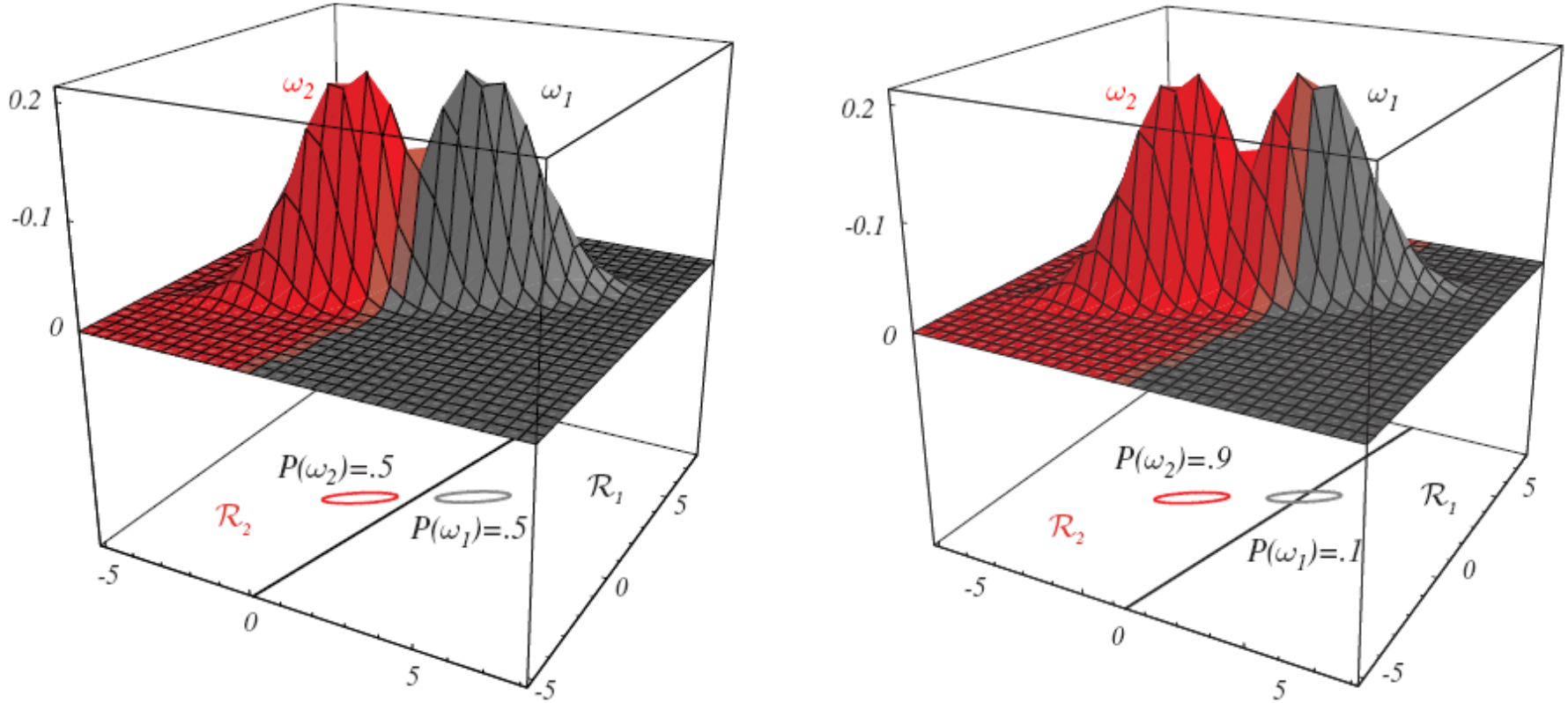


FIGURE 2.12. Probability densities (indicated by the surfaces in two dimensions and ellipsoidal surfaces in three dimensions) and decision regions for equal but asymmetric Gaussian distributions. The decision hyperplanes need not be perpendicular to the line connecting the means.

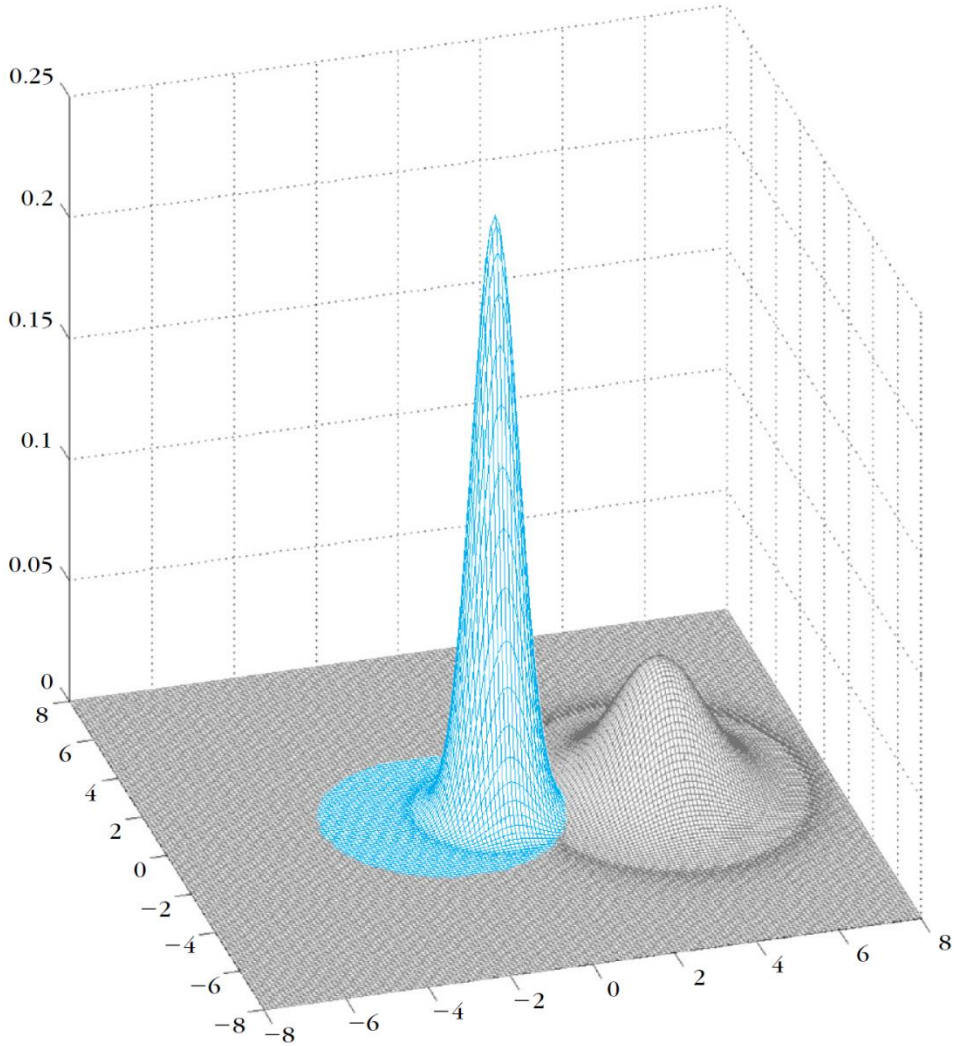


FIGURE 2.8 An example of the pdfs of two equiprobable classes in the two-dimensional space. The feature vectors in both classes are normally distributed with different covariance matrices. In this case, the decision curve is an ellipse and it is shown in Figure 2.7a. The coloring indicates the areas where the value of the respective pdf is larger

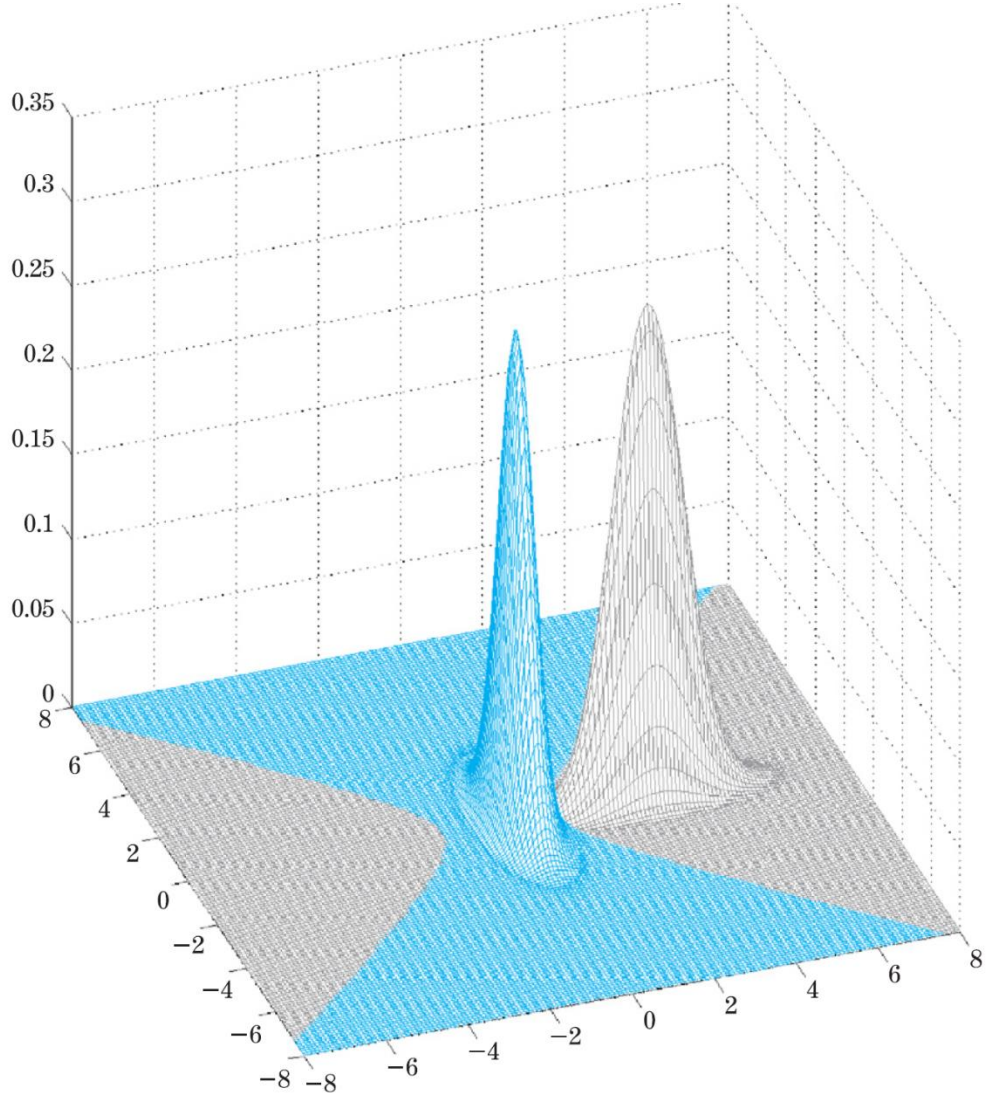


FIGURE 2.9 An example of the pdfs of two equiprobable classes in the 2D space. The feature vectors in both classes are normally distributed with different covariance matrices. In this case, the decision curve is a hyperbola and it is shown in Figure 2.7b.

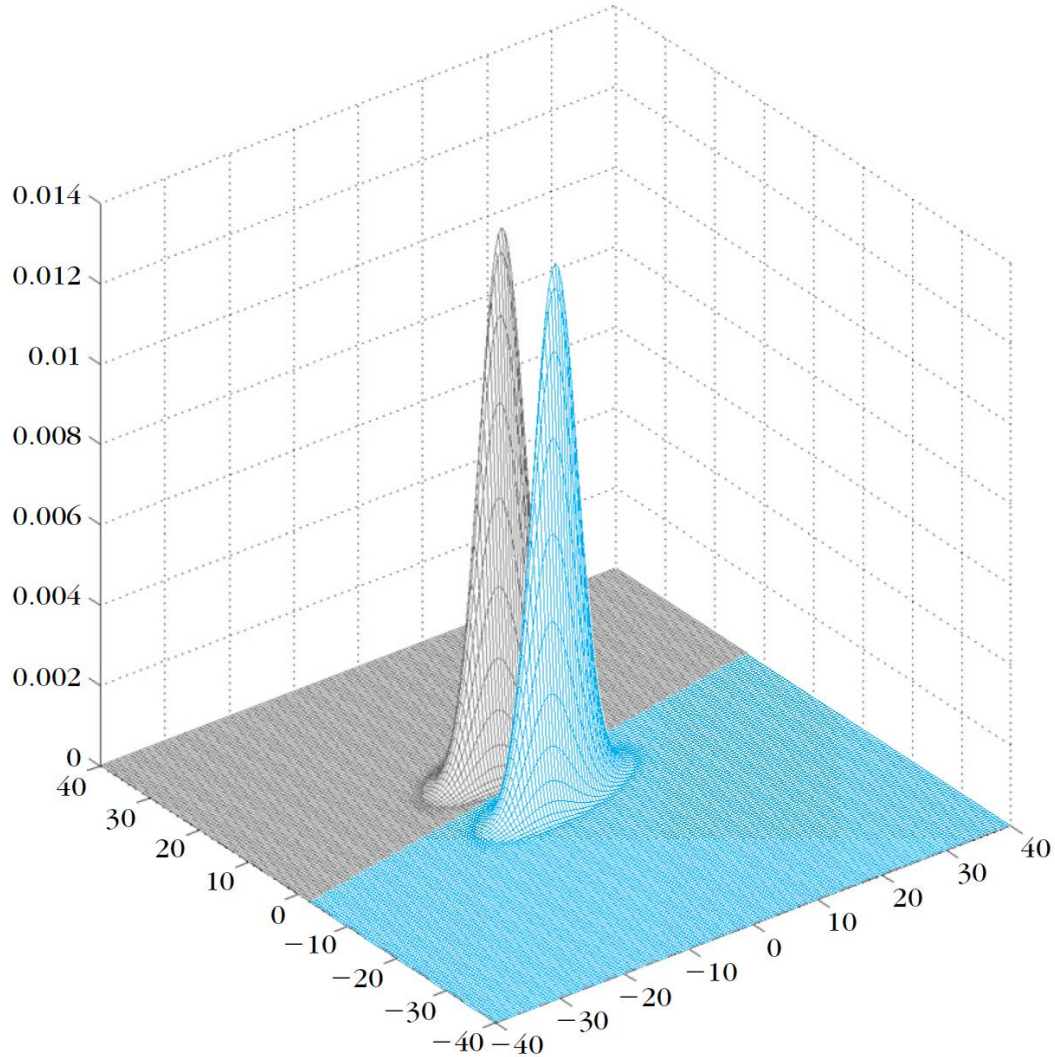


FIGURE 2.12 An example of two Gaussian pdfs with the same covariance matrix in the two-dimensional space. Each one of them is associated with one of two equiprobable classes. In this case, the decision curve is a straight line.

❖ Minimum Distance Classifiers

- $P(\omega_i) = \dfrac{1}{M}$ equiprobable
- $g_i(\underline{x}) = -\dfrac{1}{2}(\underline{x} - \underline{\mu}_i)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_i)$
- $\Sigma = \sigma^2 I$: Assign $\underline{x} \rightarrow \omega_i$:

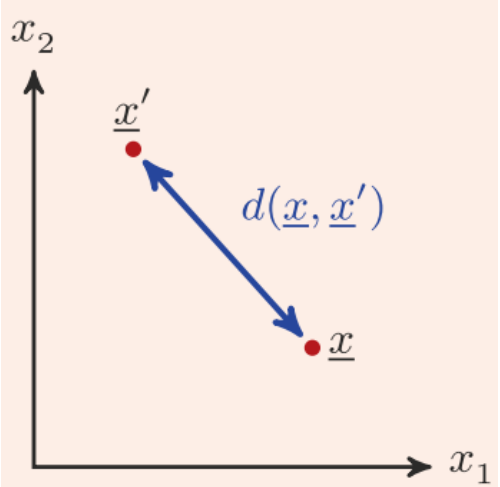
  Euclidean Distance: $d_E \equiv \left\| \underline{x} - \underline{\mu}_i \right\|$

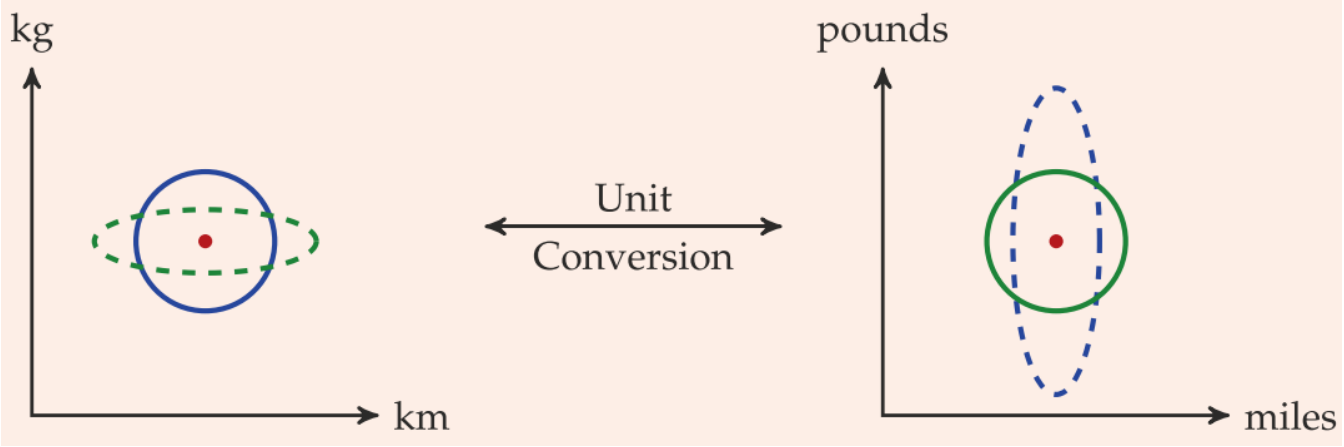
  smaller

- $\Sigma \neq \sigma^2 I$: Assign $\underline{x} \rightarrow \omega_i$:

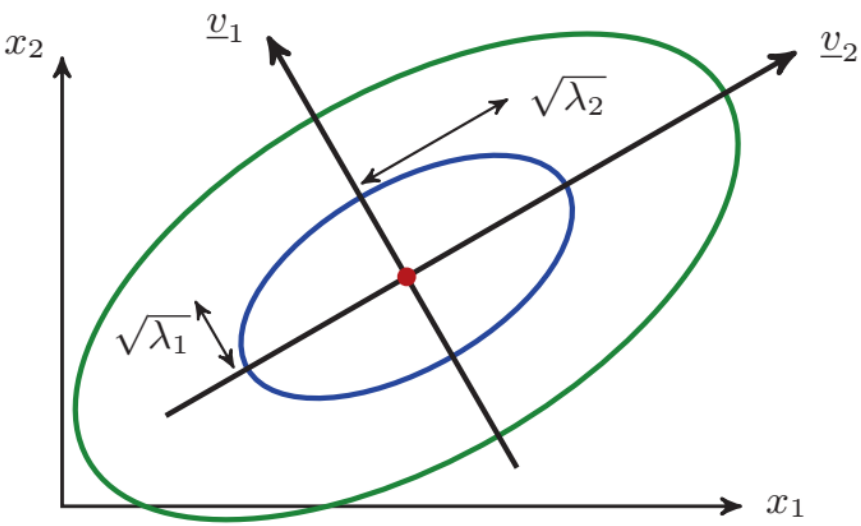
  Mahalanobis Distance: $d_m = \left((\underline{x} - \underline{\mu}_i)^T \Sigma^{-1} (\underline{x} - \underline{\mu}_i) \right)^{\frac{1}{2}}$

  smaller





Which distance is "correct"? . . . The Euclidean distance in *km-kg* space (blue), or the Euclidean distance in *mile-pound* space (green)?



The Mahalanobis Distance is based on fitting a hyper-ellipse to a class, such that the hyper-ellipse represents a distance of one standard deviation from the class mean (red dot). Two contours of constant distance from the mean are shown, with contours $\zeta = 1$ (blue) and $\zeta = 2$ (green).

$$\textit{Euclidean distance:} \quad d_\epsilon = \|\boldsymbol{x} - \boldsymbol{\mu}_i\|$$

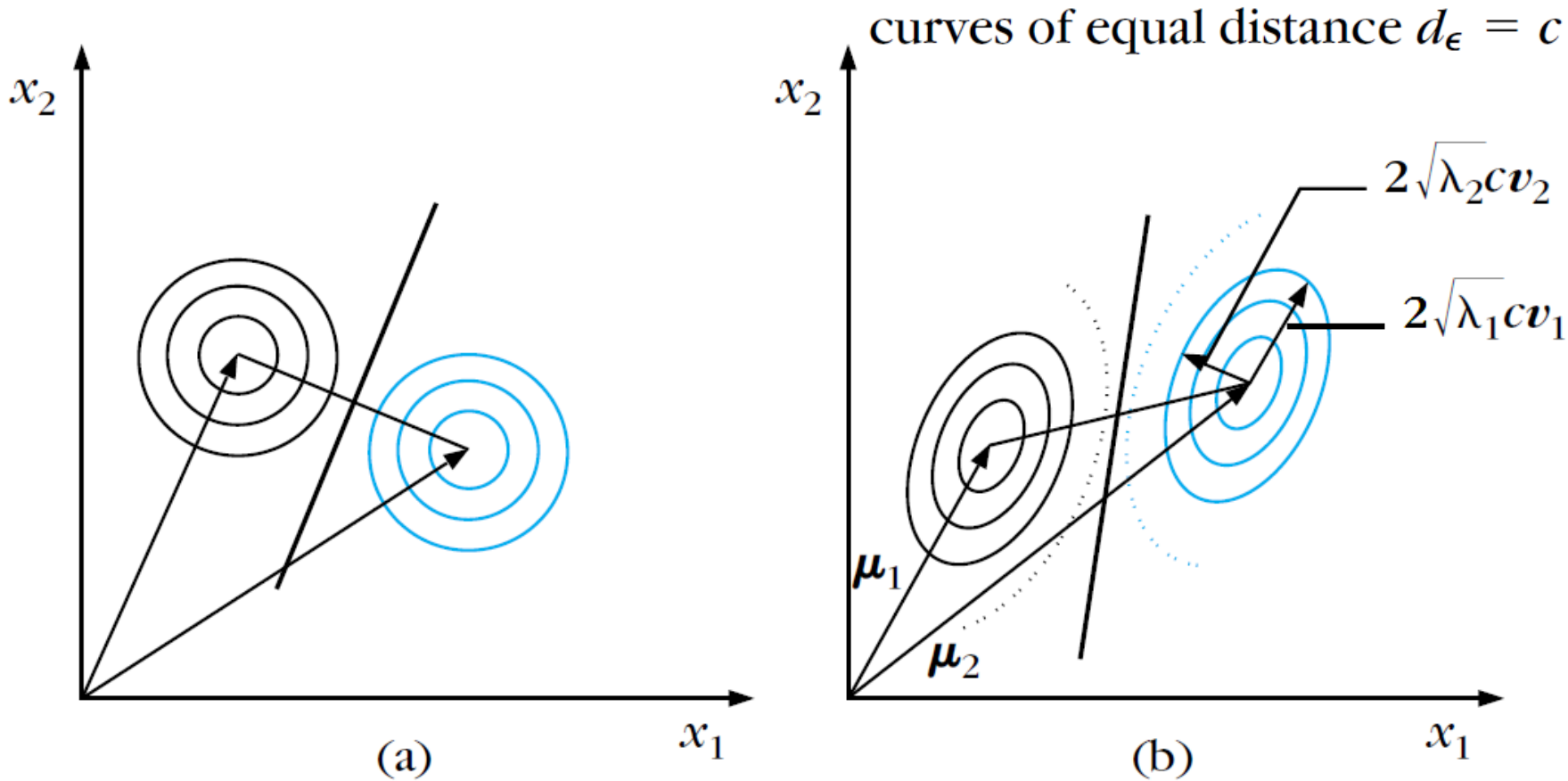


FIG 2.13 Curves of (a) equal Euclidean distance and (b) equal Mahalanobis distance from the mean points of each class. In the two-dimensional space, they are circles in the case of Euclidean distance and ellipses in the case of Mahalanobis distance. Observe that in the latter case the decision line is no longer orthogonal to the line segment joining the mean values. It turns according to the shape of the ellipses.

$$d_m^2 = r^2 = (\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

The contours of constant density are hyperellipsoids of constant Mahalanobis distance to $\boldsymbol{\mu}$ and the volume of these hyperellipsoids measures the scatter of the samples about the mean.

- The volume of the hyperellipsoid corresponding to a Mahalanobis distance r is given by

$$V = V_l \left|\Sigma\right|^{\frac{1}{2}} r^l$$

where V_l is the volume of a l-dimensional unit hypersphere:

$$V_l = \begin{cases} \pi^{l/2} / (l/2)! & l \text{ even} \\ 2^l \pi^{(l-1)/2} \left(\dfrac{l-1}{2}\right)! / (l)! & l \text{ odd.} \end{cases}$$

❖ Example:

Given ω_1, ω_2 : $P(\omega_1) = P(\omega_2)$ and $p(\underline{x}|\omega_1) = N(\underline{\mu}_1, \Sigma)$,

$$p(\underline{x}|\omega_2) = N(\underline{\mu}_2, \Sigma),\ \underline{\mu}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \underline{\mu}_2 = \begin{bmatrix} 3 \\ 3 \end{bmatrix},\ \Sigma = \begin{bmatrix} 1.1 & 0.3 \\ 0.3 & 1.9 \end{bmatrix}$$

classify the vector $\underline{x} = \begin{bmatrix} 1.0 \\ 2.2 \end{bmatrix}$ using Bayesian classification :

- $\Sigma^{-1} = \begin{bmatrix} 0.95 & -0.15 \\ -0.15 & 0.55 \end{bmatrix}$
- Compute Mahalanobis d_m from μ_1, μ_2 :

$$d^2_{m,1} = [1.0, \quad 2.2]\,\Sigma^{-1} \begin{bmatrix} 1.0 \\ 2.2 \end{bmatrix} = 2.952,$$

$$d^2_{m,2} = [-2.0, \quad -0.8]\,\Sigma^{-1} \begin{bmatrix} -2.0 \\ -0.8 \end{bmatrix} = 3.672$$

- Classify $\underline{x} \rightarrow \omega_1$. Observe that $d_{E,2} < d_{E,1}$

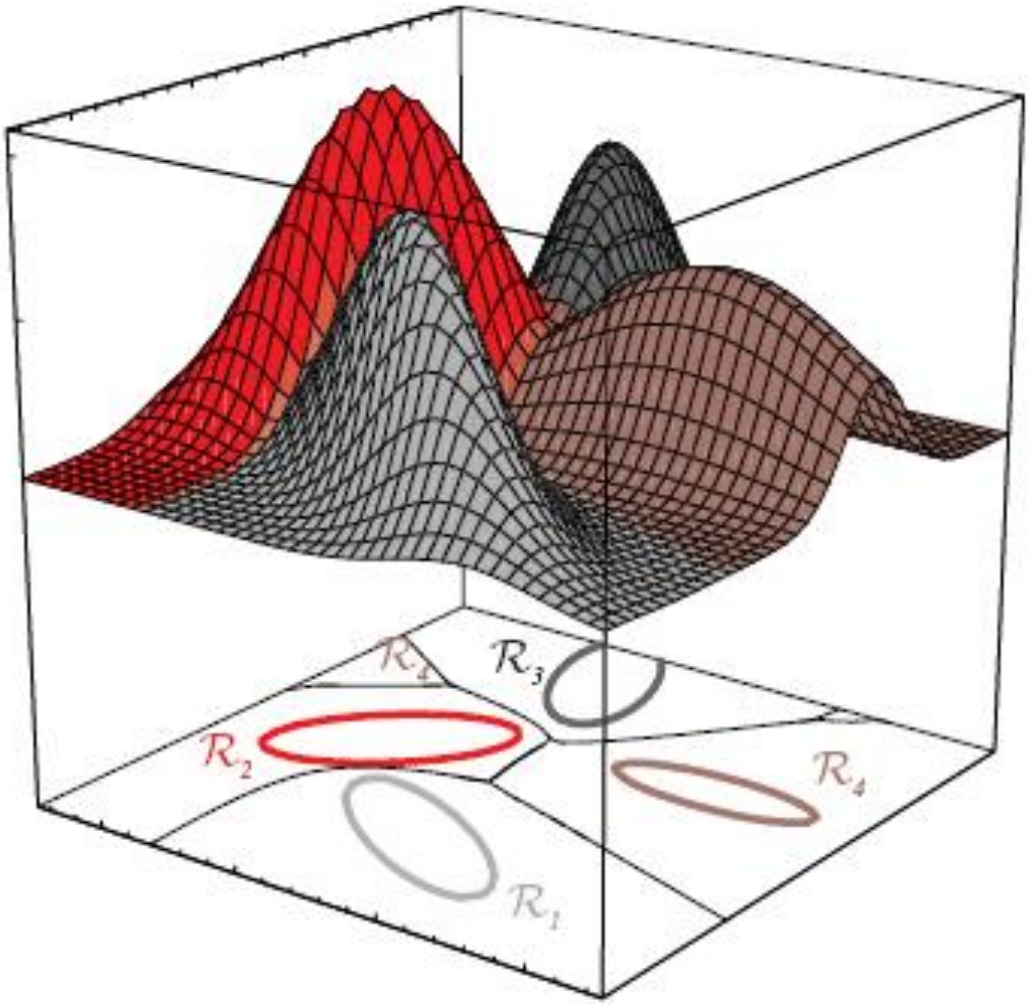
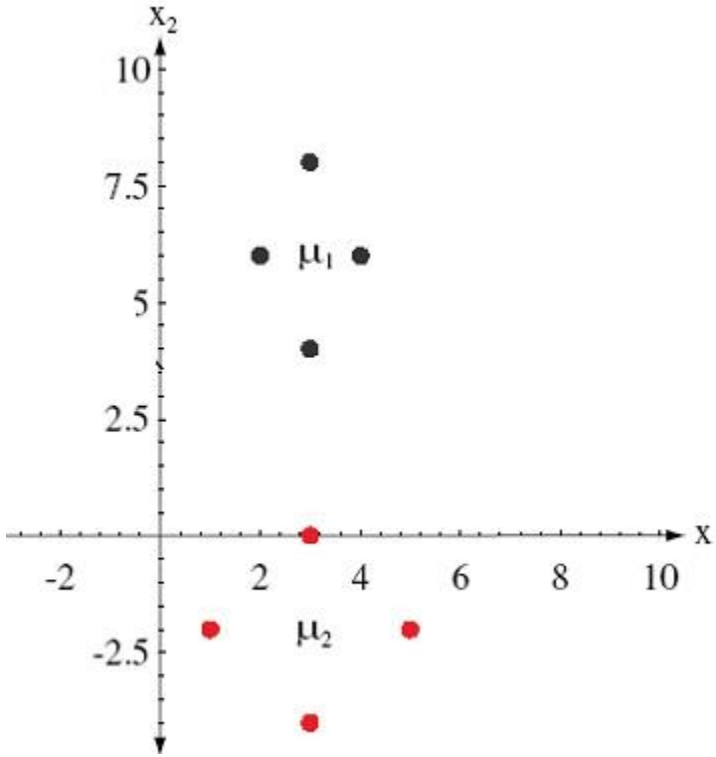


FIGURE 2.16. The decision regions for four normal distributions. Even with such a low number of categories, the shapes of the boundary regions can be rather complex.

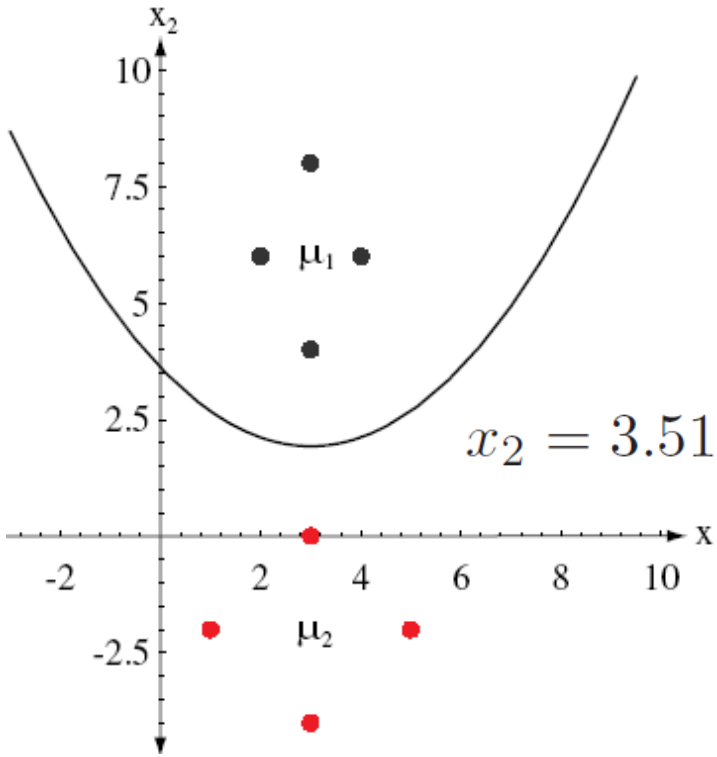
Example 1: Decision regions for two-dimensional Gaussian data



$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}; \qquad \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix} \quad \boldsymbol{\Sigma}_1^{-1} = \begin{pmatrix} 2 & 0 \\ 0 & 1/2 \end{pmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}; \qquad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad \boldsymbol{\Sigma}_2^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$$

Example 1: Decision regions for two-dimensional Gaussian data



$$x_2 = 3.514 - 1.125x_1 + 0.1875x_1^2.$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3 \\ 6 \end{bmatrix}; \qquad \boldsymbol{\Sigma}_1 = \begin{pmatrix} 1/2 & 0 \\ 0 & 2 \end{pmatrix} \quad \boldsymbol{\Sigma}_1^{-1} = \begin{pmatrix} 2 & 0 \\ 0 & 1/2 \end{pmatrix}$$

$$\boldsymbol{\mu}_2 = \begin{bmatrix} 3 \\ -2 \end{bmatrix}; \qquad \boldsymbol{\Sigma}_2 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \quad \boldsymbol{\Sigma}_2^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$$

اثبات رابطهٔ مرز تصمیم

$$g_1(\underline{x}) = g_2(\underline{x})$$

$$-\frac{1}{2}\underline{x}^T\Sigma_1^{-1}\underline{x} + (\Sigma_1^{-1}\underline{\mu}_1)^T\underline{x} - \frac{1}{2}\underline{\mu}_1^T\Sigma_1^{-1}\underline{\mu}_1 - \frac{1}{2}\ln|\Sigma_1| + \ln P(\omega_1) =$$

$$-\frac{1}{2}\underline{x}^T\Sigma_2^{-1}\underline{x} + (\Sigma_2^{-1}\underline{\mu}_2)^T\underline{x} - \frac{1}{2}\underline{\mu}_2^T\Sigma_2^{-1}\underline{\mu}_2 - \frac{1}{2}\ln|\Sigma_2| + \ln P(\omega_2)$$

$$\Rightarrow -\frac{1}{2}[x_1 \quad x_2]\begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \left(\begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} 3 \\ 6 \end{bmatrix}\right)^T\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} -$$

$$\frac{1}{2}[3 \quad 6]\begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} 3 \\ 6 \end{bmatrix} - \frac{1}{2}\ln\begin{vmatrix} \frac{1}{2} & 0 \\ 0 & 2 \end{vmatrix} + \ln 0.5 =$$

$$-\frac{1}{2}[x_1 \quad x_2]\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \left(\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} 3 \\ -2 \end{bmatrix}\right)^T\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} -$$

$$\frac{1}{2}[3 \quad -2]\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\begin{bmatrix} 3 \\ -2 \end{bmatrix} - \frac{1}{2}\ln\begin{vmatrix} 2 & 0 \\ 0 & 2 \end{vmatrix} + \ln 0.5$$

$$\Rightarrow -\frac{1}{2}[x_1 \quad x_2]\begin{bmatrix} 2x_1 \\ \frac{1}{2}x_2 \end{bmatrix} + [6 \quad 3]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \frac{1}{2}[3 \quad 6]\begin{bmatrix} 6 \\ 3 \end{bmatrix} - 0 + \ln 0.5 =$$

$$-\frac{1}{4}x_1^2 - \frac{1}{4}x_2^2 + [3/2 \quad -1]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \frac{1}{4}(9+4) - \frac{1}{2}\ln 4 + \ln 0.5$$

$$\Rightarrow -x_1^2 - \frac{1}{4}x_2^2 + 6x_1 + 3x_2 - 18 + \ln 0.5 =$$

$$-\frac{1}{4}x_1^2 - \frac{1}{4}x_2^2 + \frac{3}{2}x_1 - x_2 - \frac{13}{4} - \frac{1}{2}\ln 4 + \ln 0.5$$

$$\Rightarrow -\frac{3}{4}x_1^2 + 4.5x_1 + 4x_2 - 14.06 = 0 \Rightarrow \boxed{x_2 = 0.1875x_1^2 - 1.125x_1 + 3.515}$$

ESTIMATION OF UNKNOWN PROBABILITY DENSITY FUNCTIONS

(PARAMETRIC)

❖ 1- Maximum Likelihood

- Let $\underline{x}_1,\ \underline{x}_2,\ldots,\ \underline{x}_N$ known and independent
- Let $p(\underline{x})$ known within an unknown vector parameter $\underline{\theta}$: $p(\underline{x}) \equiv p(\underline{x};\underline{\theta})$
- $X = \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N$
- $p(X;\underline{\theta}) \equiv p(\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N;\underline{\theta})$

$$= \prod_{k=1}^{N} p(\underline{x}_k;\underline{\theta})$$

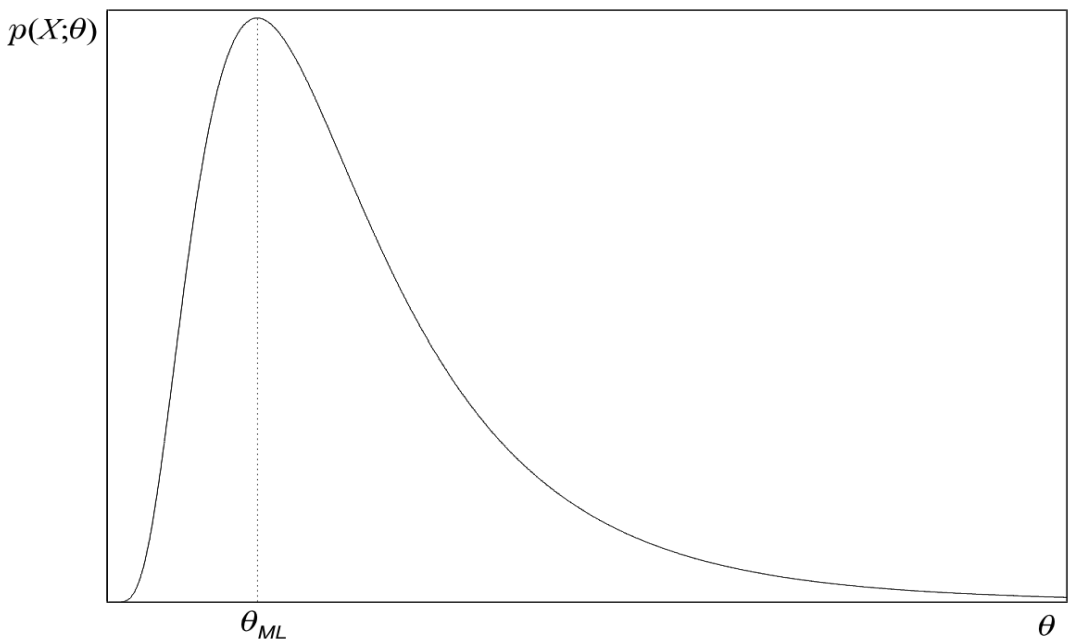
which is known as the Likelihood of $\underline{\theta}$ w.r. to X.

$$\boxed{\hat{\underline{\theta}}_{\mathrm{ML}} : \arg\max_{\underline{\theta}} \prod_{k=1}^{N} p(\underline{x}_k ; \underline{\theta}),} \quad L(\underline{\theta}) \equiv \ln p(X ; \underline{\theta}) = \sum_{k=1}^{N} \ln p(\underline{x}_k ; \underline{\theta})$$

$$\hat{\underline{\theta}}_{ML} : \frac{\partial L(\underline{\theta})}{\partial(\underline{\theta})} = \sum_{k=1}^{N} \frac{1}{p(\underline{x}_k ; \underline{\theta})} \frac{\partial p(\underline{x}_k ; \underline{\theta})}{\partial(\underline{\theta})}$$

$$= \sum_{k=1}^{N} \frac{1}{p(\underline{x}_k ; \underline{\theta})} \nabla_{\underline{\theta}} p(\underline{x}_k ; \underline{\theta}) = \underline{0}$$

$$\nabla_{\underline{\theta}} = \begin{bmatrix} \frac{\partial}{\partial \theta_1} & \frac{\partial}{\partial \theta_2} & \cdots & \frac{\partial}{\partial \theta_p} \end{bmatrix}^T$$



If, indeed, there is a $\underline{\theta}_0$ such that

$p(\underline{x}) = p(\underline{x};\underline{\theta}_0)$, then

$$\lim_{N\to\infty} E[\underline{\theta}_{ML}] = \underline{\theta}_0$$ Asymptotically unbiased

$$\lim_{N\to\infty} E\left\|\hat{\underline{\theta}}_{ML} - \underline{\theta}_0\right\|^2 = 0$$ consistent

The ML estimator is unbiased, is normally distributed, and has the minimum possible variance. However, all these nice properties are valid only for large values of N.

❖ Example:

$$p(\underline{x}) \sim N(\underline{\mu}, \mathbf{\Sigma}): \quad \underline{\mu} \text{ unknown}, \quad \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N \quad p(\underline{x}_k) \equiv p(\underline{x}_k; \underline{\mu})$$

$$p(\underline{x}_k; \underline{\mu}) = \frac{1}{(2\pi)^{\frac{l}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}} \exp(-\frac{1}{2}(\underline{x}_k - \underline{\mu})^T \mathbf{\Sigma}^{-1}(\underline{x}_k - \underline{\mu}))$$

$$L(\underline{\mu}) = \ln \prod_{k=1}^{N} p(\underline{x}_k; \underline{\mu}) = C - \frac{1}{2} \sum_{k=1}^{N} (\underline{x}_k - \underline{\mu})^T \mathbf{\Sigma}^{-1}(\underline{x}_k - \underline{\mu})$$

$$\frac{\partial L(\underline{\mu})}{\partial(\underline{\mu})} \equiv \begin{bmatrix} \frac{\partial L}{\partial \mu_1} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial L}{\partial \mu_l} \end{bmatrix} = \sum_{k=1}^{N} \mathbf{\Sigma}^{-1}(\underline{x}_k - \underline{\mu}) = \underline{0} \Rightarrow \underline{\mu}_{ML} = \frac{1}{N} \sum_{k=1}^{N} \underline{x}_k$$

Remember: if matrix $\mathbf{A} = \mathbf{A}^T \Rightarrow \dfrac{\partial(\underline{\alpha}^T \mathbf{A} \underline{\alpha})}{\partial \underline{\alpha}} = 2\mathbf{A}\underline{\alpha}$

❖ work example 2.3 textbook p 42

Example 2.3

Assume that N data points, $x_1, x_2, \ldots, x_N$, have been generated by a one-dimensional Gaussian pdf of known mean, μ, but of unknown variance. Derive the ML estimate of the variance.

The log-likelihood function for this case is given by

$$L(\sigma^2) = \ln \prod_{k=1}^{N} p(x_k; \sigma^2) = \ln \prod_{k=1}^{N} \frac{1}{\sqrt{2\pi}\sqrt{\sigma^2}} \exp\left(-\frac{(x_k - \mu)^2}{2\sigma^2}\right)$$

or

$$L(\sigma^2) = -\frac{N}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{k=1}^{N} (x_k - \mu)^2$$

Taking the derivative of the above with respect to σ^2 and equating to zero, we obtain

$$-\frac{N}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{k=1}^{N} (x_k - \mu)^2 = 0$$

and finally the ML estimate of σ^2 results as the solution of the above,

$$\boxed{\hat{\sigma}^2_{ML} = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)^2} \tag{2.63}$$

Observe that, for finite N, $\hat{\sigma}^2_{ML}$ in Eq. (2.63) is a biased estimate of the variance. Indeed,

$$E[\hat{\sigma}^2_{ML}] = \frac{1}{N}\sum_{k=1}^{N}E[(x_k - \mu)^2] = \frac{N-1}{N}\sigma^2$$

Wrong!!!

where σ^2 is the true variance of the Gaussian pdf. However, for large values of N, we have

$$E[\hat{\sigma}^2_{ML}] = (1 - \frac{1}{N})\sigma^2 \approx \sigma^2$$

which is in line with the theoretical result of asymptotic consistency of the ML estimator.

For known μ: $E[\hat{\sigma}^2] = \sigma^2$

For unknown μ: $E[\hat{\mu}] = \mu$ and $E[\hat{\sigma}^2] = \frac{N-1}{N}.\sigma^2$

ML Estimation:

Gaussian Case: unknown $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$

$\boldsymbol{\theta} = (\theta_1, \theta_2)^T = (\mu, \sigma^2)^T$ single point

$$l = \ln p(x_k \mid \boldsymbol{\theta}) = -\frac{1}{2}\ln 2\pi\theta_2 - \frac{1}{2\theta_2}(x_k - \theta_1)^2$$

$$\nabla_{\boldsymbol{\theta}} l = \nabla_{\boldsymbol{\theta}} \ln p(x_k \mid \boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial}{\partial \theta_1}(\ln p(x_k \mid \boldsymbol{\theta})) \\ \frac{\partial}{\partial \theta_2}(\ln p(x_k \mid \boldsymbol{\theta})) \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\theta_2}(x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

Summation (Applying above eq. to the full log-likelihood leads to the conditions):

$$\begin{cases} \sum_{k=1}^{N} \frac{1}{\hat{\theta}_2}(x_k - \hat{\theta}_1) = 0 & (1) \\ -\sum_{k=1}^{N} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{N} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

Combining (1) and (2), one obtains (By substituting $\hat{\mu} = \hat{\theta}_1, \ \ \hat{\sigma}^2 = \hat{\theta}_2$ and doing a little rearranging):

$$\boxed{\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N} x_k \quad ; \quad \hat{\sigma}^2 = \frac{1}{N}\sum_{k=1}^{N}(x_k - \hat{\mu})^2}$$

Derivation of Expectation of ML estimate for σ^2

$$E[\hat{\mu}] = \ldots = \mu$$

$$E[\hat{\sigma}^2] = E\left[\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2\right]$$

$$= E[\frac{1}{N}\sum_{i=1}^{N} x_i^2] - 2E[\frac{1}{N}\sum_{i=1}^{N} x_i \hat{\mu}] + E[\frac{1}{N}\sum_{i=1}^{N} \hat{\mu}^2]$$

$$= \mu^2 + \sigma^2 - 2(\mu^2 + \frac{\sigma^2}{N}) + (\mu^2 + \frac{\sigma^2}{N})$$

$$= \frac{N-1}{N}.\sigma^2 = \sigma^2 - \frac{\sigma^2}{N} \neq \sigma^2$$

The multivariate case

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k$$

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^T$$

The maximum likelihood estimate for the mean vector is the sample mean.

The maximum likelihood estimate for the covariance matrix is the arithmetic average of the N matrices

$$(\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^T.$$

❖ 2- Maximum Aposteriori Probability Estimation

- In ML method, $\underline{\theta}$ was considered as a parameter
- Here we shall look at $\underline{\theta}$ as a random vector described by a pdf $p(\underline{\theta})$, assumed to be known
- Given

$$X = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$$

Compute the maximum of

$$p(\underline{\theta} | X)$$

- From Bayes theorem

$$p(\underline{\theta})\, p(X | \underline{\theta}) = p(X)\, p(\underline{\theta} | X) \quad \text{or}$$

$$p(\underline{\theta} \,|\, X) = \frac{p(\underline{\theta})\, p(X | \underline{\theta})}{p(X)}$$

- The method:

$$\underline{\hat{\theta}}_{MAP} = \arg\max_{\underline{\theta}} p(\underline{\theta}|X) \text{ or}$$

$$\boxed{\underline{\hat{\theta}}_{MAP} : \frac{\partial}{\partial \underline{\theta}} (P(\underline{\theta}) p(X|\underline{\theta}))}$$

If $p(\underline{\theta})$ is uniform or broad enough $\underline{\hat{\theta}}_{MAP} \cong \underline{\theta}_{ML}$

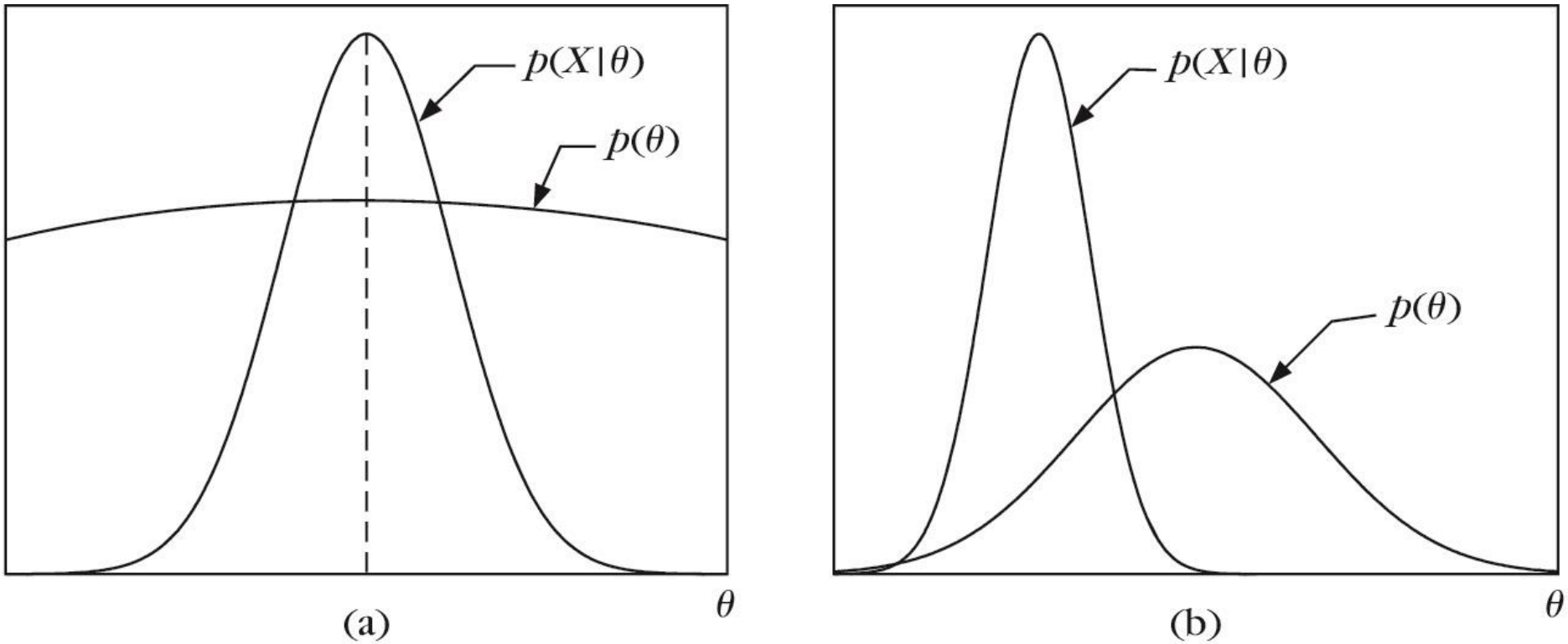


FIGURE 2.15 ML and MAP estimates of θ will be approximately the same in (a) and different in (b).

❖ Example:

$$p(\underline{x}):\ N(\underline{\mu},\Sigma),\quad \underline{\mu}\ \text{ unknown},\quad X=\{\underline{x}_1,\ldots,\underline{x}_N\}$$

$$p(\underline{\mu})=\frac{1}{(2\pi)^{\frac{l}{2}}\sigma_\mu^l}\exp\left(-\frac{\|\underline{\mu}-\underline{\mu}_0\|^2}{2\sigma_\mu^2}\right)\qquad \text{for } \Sigma=\sigma^2 I$$

$$\Downarrow$$

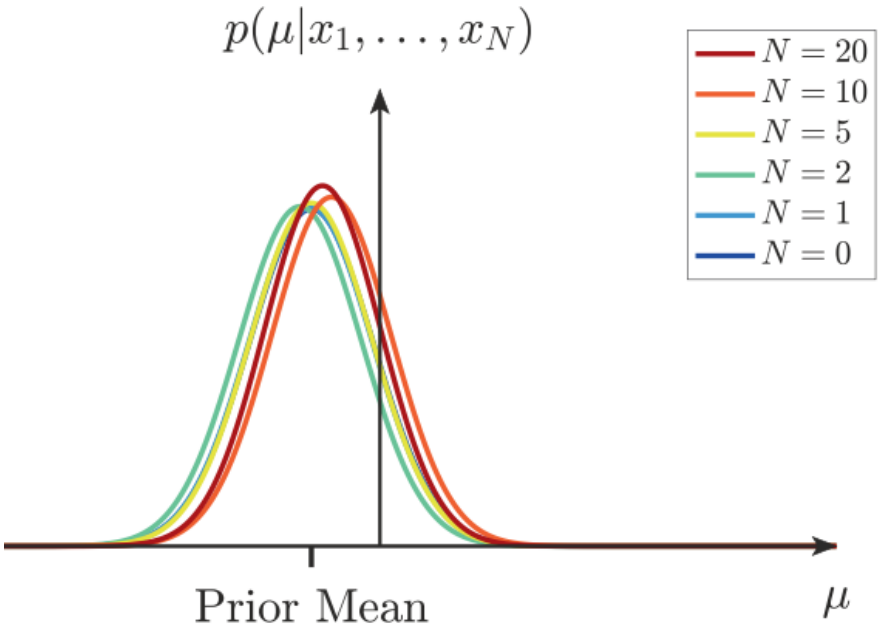
$$\hat{\underline{\theta}}_{MAP}:\quad \frac{\partial}{\partial\underline{\mu}}\ln\left(\prod_{k=1}^{N}p(\underline{x}_k|\underline{\mu})p(\underline{\mu})\right)=\underline{0}\quad\text{or}\quad \sum_{k=1}^{N}\frac{1}{\sigma^2}(\underline{x}_k-\underline{\mu})-\frac{1}{\sigma_\mu^2}(\underline{\mu}-\underline{\mu}_0)=\underline{0}$$

$$\Rightarrow\quad \hat{\underline{\mu}}_{MAP}=\frac{\underline{\mu}_0+\frac{\sigma_\mu^2}{\sigma^2}\sum_{k=1}^{N}\underline{x}_k}{1+\frac{\sigma_\mu^2}{\sigma^2}N}=\frac{1}{1+\frac{\sigma_\mu^2}{\sigma^2}N}\underline{\mu}_0+\frac{\frac{\sigma_\mu^2}{\sigma^2}N}{1+\frac{\sigma_\mu^2}{\sigma^2}N}\left(\frac{1}{N}\sum_{k=1}^{N}\underline{x}_k\right)$$

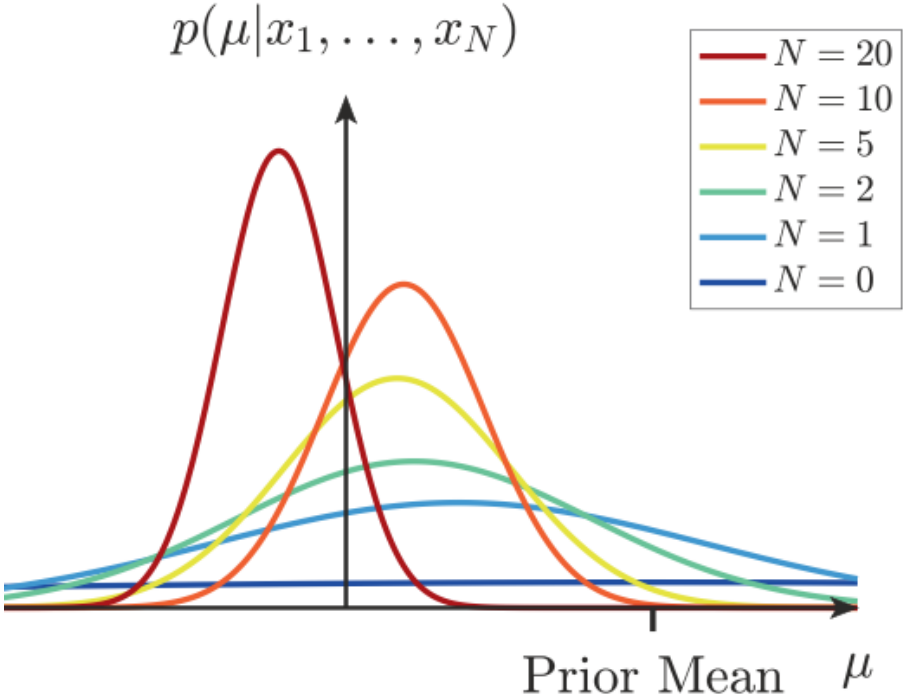
Relative certainty of prior

Relative certainty of measurements

$$For\quad \frac{\sigma_\mu^2}{\sigma^2}>>1,\ \text{or for } N\to\infty\qquad \hat{\underline{\mu}}_{\mathrm{MAP}}\cong\hat{\underline{\mu}}_{ML}=\frac{1}{N}\sum_{k=1}^{N}\underline{x}_k$$



Strong prior
Weak measurements



Weak prior
Strong measurements

❖ 3- Bayesian Inference

- $\text{ML}, \text{MAP} \Rightarrow$ a single estimate for $\underline{\theta}$.

  Here a different root is followed.

  $\text{Given}: \; X = \{\underline{x}_1, \ldots, \underline{x}_N\}, \; p(\underline{x}|\underline{\theta}) \text{ and } p(\underline{\theta})$

  $\text{The goal}: \; \text{estimate } \mathrm{p}(\underline{x}|X)$

  How??

$$p(\underline{x}|X) = \int p(\underline{x}|\underline{\theta})\, p(\underline{\theta}|X)\, d\underline{\theta}$$

$$p(\underline{\theta}|X) = \frac{p(X|\underline{\theta})\, p(\underline{\theta})}{p(X)} = \frac{p(X|\underline{\theta})\, p(\underline{\theta})}{\int p(X|\underline{\theta})\, p(\underline{\theta})\, d\underline{\theta}}$$

$$p(X|\underline{\theta}) = \prod_{k=1}^{N} p(\underline{x}_k|\underline{\theta})$$

A bit more insight via an example

- $Let\ p(x|\mu) \rightarrow N(\mu, \sigma^2)$
- $p(\mu) \rightarrow N(\mu_0, \sigma_0^2)$
- It turns out that: $p(\mu|X) \rightarrow N(\mu_N, \sigma_N^2)$

$$p(\mu|X) = \frac{p(X|\mu)\, p(\mu)}{p(X)} = \frac{1}{\beta} \prod_{k=1}^{N} p(x_k|\mu)\, p(\mu) = \alpha \prod_{k=1}^{N} p(x_k|\mu)\, p(\mu)$$

$$p(\mu \mid X) = \alpha \prod_{k=1}^{n} \left\{ \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{1}{2}(\frac{x_k - \mu}{\sigma})^2] \right\} \left\{ \frac{1}{\sqrt{2\pi}\sigma_0} \exp[-\frac{1}{2}(\frac{\mu - \mu_0}{\sigma_0})^2] \right\}$$

$$= \alpha' \exp\left[-\frac{1}{2}(\sum_{k=1}^{n}(\frac{x_k - \mu}{\sigma})^2 + (\frac{\mu - \mu_0}{\sigma_0})^2) \right]$$

$n \leftrightarrow N$

$$= \alpha' \exp\left[-\frac{1}{2}(\sum_{k=1}^{n}(\frac{x_k^2}{\sigma^2} - 2\frac{x_k \mu}{\sigma^2} + \frac{\mu^2}{\sigma^2}) + (\frac{\mu^2 - 2\mu\mu_0 + \mu_0^2}{\sigma_0^2})) \right]$$

$$= \alpha'' \exp\left[-\frac{1}{2}(\sum_{k=1}^{n}(-2\frac{x_k \mu}{\sigma^2} + \frac{\mu^2}{\sigma^2}) + (\frac{\mu^2}{\sigma_0^2} - \frac{2\mu\mu_0}{\sigma_0^2})) \right]$$

$$= \alpha'' \exp - \frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right) \mu^2 - 2(\frac{1}{\sigma^2}\sum_{k=1}^{n} x_k + \frac{\mu_0}{\sigma_0^2})\mu \right]$$

$$= \alpha'' \exp - \frac{1}{2}\left[\left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2} \right) \mu^2 - 2(\frac{1}{\sigma^2}(n\hat{\mu}_n) + \frac{\mu_0}{\sigma_0^2})\mu \right] \qquad (1)$$

$$where \quad \hat{\mu}_n = \frac{1}{n}\sum_{k=1}^{n} x_k$$

❖If we write $\mathrm{p}(\mu \mid X) \sim N(\mu_N, \sigma_N^2)$

$$\begin{cases} \dfrac{1}{\sigma_N^2} = \dfrac{N}{\sigma^2} + \dfrac{1}{\sigma_0^2} \\ \dfrac{\mu_N}{\sigma_N^2} = \dfrac{N}{\sigma^2}\bar{x}_N + \dfrac{\mu_0}{\sigma_0^2} \end{cases}$$

❖where $\bar{x}_n$ is the sample mean

$$\bar{x}_N = \hat{\mu}_N = \frac{1}{N}\sum_{k=1}^{N} x_k$$

We solve explicitly for μ_N and σ_N^2 and obtain

$$\mu_N = \frac{N\sigma_0^2\bar{x} + \sigma^2\mu_0}{N\sigma_0^2 + \sigma^2}, \quad \sigma_N^2 = \frac{\sigma^2\sigma_0^2}{N\sigma_0^2 + \sigma^2}, \quad \bar{x} = \frac{1}{N}\sum_{k=1}^{N} x_k$$

it can be shown

$$p(x \mid X) = \frac{1}{\sqrt{2\pi(\sigma^2 + \sigma_N^2)}} \exp\left(-\frac{1}{2}\frac{(x - \mu_N)^2}{(\sigma^2 + \sigma_N^2)}\right)$$

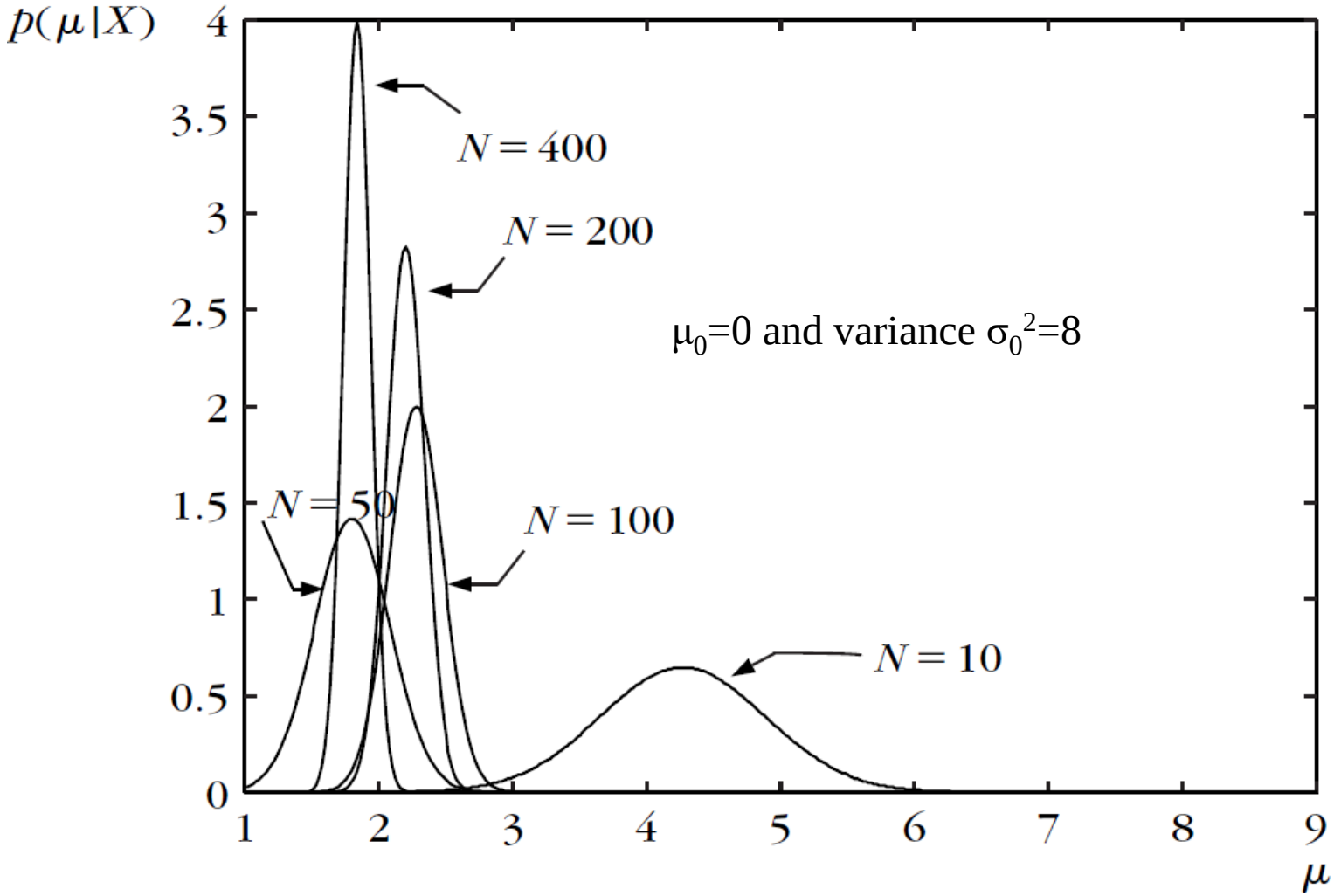


FIGURE 2.16 A sequence of the posterior pdf estimates (Eq. (2.73)), for the case of Example 2.6. As the number of training points increases, the posterior pdf becomes more spiky (the ambiguity decreases) and its center moves toward the true mean value of the data.

Data were generated using a pseudorandom number generator following a Gaussian pdf with mean value equal to $\mu=2$ and variance $\sigma^2=4$.

3- Bayesian Inference: The Multivariate Case

- The treatment of the multivariate case in which $\boldsymbol{\Sigma}$ is known but $\boldsymbol{\mu}$ is not, is a direct generalization of the univariate case.

$$p(\mathbf{x}|\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}, \Sigma) \quad \text{and} \quad p(\boldsymbol{\mu}) \sim N(\boldsymbol{\mu}_0, \Sigma_0)$$

- where $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_0$, and $\boldsymbol{\mu}_0$ are assumed to be known.
- After observing a set X of n independent samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we use Bayes' formula to obtain

$$p(\boldsymbol{\mu} \mid X) = \alpha \prod_{k=1}^{n} p(\mathbf{x}_k \mid \boldsymbol{\mu}) p(\boldsymbol{\mu})$$

$$n \leftrightarrow N$$

$$= \alpha' \exp\left[-\frac{1}{2} (\boldsymbol{\mu}^{\mathrm{t}} (n \Sigma^{-1} + \Sigma_0^{-1}) \boldsymbol{\mu} - 2 \boldsymbol{\mu}^{\mathrm{t}} (\Sigma^{-1} \sum_{i=1}^{n} \mathbf{x}_k + \Sigma_0^{-1} \boldsymbol{\mu}_0)) \right]$$

which has the form

$$p(\boldsymbol{\mu} \mid X) = \alpha'' \exp\left[-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\mu}_N)^T \Sigma_N^{-1}(\boldsymbol{\mu}-\boldsymbol{\mu}_N)\right]$$

Thus, $p(\boldsymbol{\mu}|X) \sim N(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N)$, and once again we have a reproducing density.

$$\begin{cases} \Sigma_N^{-1} = N\Sigma^{-1} + \Sigma_0^{-1} \\ \Sigma_N^{-1}\boldsymbol{\mu}_N = N\Sigma^{-1}\hat{\boldsymbol{\mu}}_N + \Sigma_0^{-1}\mu_0 \end{cases}$$

where $\hat{\boldsymbol{\mu}}_N = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k$

After a little manipulation

Linear combination of $\hat{\boldsymbol{\mu}}_n$ and $\boldsymbol{\mu}_0$

$$\begin{cases} \boldsymbol{\mu}_N = \Sigma_0(\Sigma_0 + \frac{1}{N}\Sigma)^{-1}\hat{\boldsymbol{\mu}}_N + \frac{1}{N}\Sigma(\Sigma_0 + \frac{1}{N}\Sigma)^{-1}\boldsymbol{\mu}_0 \\ \Sigma_N = \Sigma_0(\Sigma_0 + \frac{1}{N}\Sigma)^{-1}\frac{1}{N}\Sigma \end{cases}$$

$$p(\mathbf{x}|X) \sim N(\boldsymbol{\mu}_N, \boldsymbol{\Sigma}+\boldsymbol{\Sigma}_N)$$

❖ 4- Maximum Entropy

- Entropy is a measure of the uncertainty concerning an event and, from another view point, a measure of randomness of the messages (feature vectors in our case) occurring at the output of a system.

- If $p(\underline{x})$ is the density function, the associated entropy H is given by:

- $$H = -\int p(\underline{x}) \ln p(\underline{x}) d\underline{x}$$

$\hat{p}(\underline{x})$: Maximum H subject to the available constraints

According to the principle of maximum entropy, such an estimate corresponds to the distribution that exhibits the highest possible randomness, subject to the available constraints.

❖ Example: x is nonzero in the interval $x_1 \le x \le x_2$ and zero otherwise. Compute the ME pdf

- The constraint: $\int_{x_1}^{x_2} p(x)dx = 1$
- Lagrange Multipliers …

$$H_L = H + \lambda\left(\int_{x_1}^{x_2} p(x)dx - 1\right) = -\int_{x_1}^{x_2} p(x)\left(\ln p(x) - \lambda\right)dx$$

$$\frac{\partial H_L}{\partial p(x)} = -\int_{x_1}^{x_2} \left\{\left(\ln p(x) - \lambda\right) + 1\right\} dx = 0$$

$$\Rightarrow \quad \hat{p}(x) = \exp(\lambda - 1) \quad \Rightarrow \quad \hat{p}(x) = \begin{cases} \dfrac{1}{x_2 - x_1} & x_1 \le x \le x_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\int_{x_1}^{x_2} \exp(\lambda - 1)dx = 1$$

❖It can be shown that the normal distribution has the maximum entropy of all distributions having a given mean and variance.

❖ 5- Mixture Models

- $$p(\underline{x}) = \sum_{j=1}^{J} p(\underline{x} \mid j) P_j$$

J distributions contribute to the formation of $p(\boldsymbol{x})$.

$$\sum_{j=1}^{J} P_j = 1, \quad \int_{\underline{x}} p(\underline{x} \mid j) d\underline{x} = 1$$

- Assume parametric modeling, i.e., $p(\underline{x} \mid j; \underline{\theta})$

- The goal is to estimate $\underline{\theta}$ and $P_1, P_2, \ldots, P_J$

  given a set $X = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$

- Why not ML? As before?

$$\max_{\underline{\theta}, P_1, \ldots, P_J} \prod_{k=1}^{N} P(\underline{x}_k ; \underline{\theta}, P_1, \ldots, P_J)$$

- This is a nonlinear problem due to the missing label information. This is a typical problem with an incomplete data set.

- The Expectation-Maximization (EM) algorithm.

  - General formulation

$$\underline{y} \text{ the complete data set } \underline{y} \in Y \subseteq R^m, \text{ with } p_{\underline{y}}(\underline{y};\underline{\theta}),$$

which are not observed directly.

We observe

$$\underline{x} = g(\underline{y}) \in X_{ob} \subseteq R^l, l < m \text{ with } P_x(\underline{x};\underline{\theta}),$$

a many to one transformation

- Let $Y(\underline{x}) \subseteq Y$ all $\underline{y}'s \rightarrow$ to a specific $\underline{x}$

$$p_{\underline{x}}(\underline{x};\underline{\theta}) = \int_{Y(\underline{x})} p_{\underline{y}}(\underline{y};\underline{\theta})d\,\underline{y}$$

- What we need is to compute

$$\hat{\theta}_{ML} : \sum_k \frac{\partial \ln(p_{\underline{y}}(\underline{y}_k;\underline{\theta}))}{\partial \underline{\theta}} = \underline{0}$$

- But $y_k's$ are not observed. Here comes the EM. Maximize the expectation of the log-likelihood **conditioned** on the observed samples and the current iteration estimate of $\underline{\theta}$.

- The algorithm:
  - E-step: $$Q(\underline{\theta};\underline{\theta}(t)) = E\left[\sum_k \ln(p_{\underline{y}}(\underline{y}_k;\underline{\theta}\,|\,X;\underline{\theta}(t))\right]$$
  - M-step: $$\underline{\theta}(t+1): \frac{\partial Q(\underline{\theta};\underline{\theta}(t))}{\partial \underline{\theta}} = \underline{0}$$
- we start from an initial estimate $\underline{\theta}(0)$,and iterations are terminated if $\|\underline{\theta}(t+1)-\underline{\theta}(t)\| \le \varepsilon$ for an appropriately chosen vector norm and ε.

❖ Application to the mixture modeling problem

- Complete data $(\underline{x}_k, j_k), k = 1,2,\ldots,N$

j_k is an integer $\in [1,\ J]$ and it denotes the mixture from which $\underline{x}_k$ is generated

❖Observed data $\underline{x}_k,\ k = 1, 2, \ldots, N$

❖ $$p(\underline{x}_k, j_k; \underline{\theta}) = p(\underline{x}_k \,|\, j_k; \underline{\theta}) P_{j_k}$$

❖Assuming mutual independence among samples of the data set

$$L(\underline{\theta}) = \sum_{k=1}^{N} \ln(p(\underline{x}_k \,|\, j_k; \underline{\theta}) P_{j_k})$$

❖Unknown parameters

$$\underline{\Theta}^T = [\underline{\theta}^T, \underline{P}^T]^T, \underline{P} = [P_1, P_2, \ldots, P_J]^T$$

❖Taking the expectation over the unobserved data, conditioned on the training samples and the current estimates, $\underline{\Theta}(t)$, of the unknown parameters, we have:

❖ E-step

$$Q(\underline{\Theta};\underline{\Theta}(t)) = E\left[\sum_{k=1}^{N} \ln(p(\underline{x}_k \,|\, j_k;\underline{\theta})P_{j_k})\right] = \sum_{k=1}^{N} E\left[\ \ldots\ \right]$$

$$= \sum_{k=1}^{N} \sum_{j_k=1}^{J} \boxed{P\ (j_k \,|\, \underline{x}_k;\underline{\Theta}(t))} \ln(p(\underline{x}_k \,|\, j_k;\underline{\theta})P_{j_k}) \qquad (2\text{-}95)$$

❖ M-step

$$\frac{\partial Q}{\partial \underline{\theta}} = \underline{0} \qquad \frac{\partial Q}{\partial P_{jk}} = 0, \qquad j_k = 1,2,\ldots,J$$

Example: For $\mathbf{\Sigma}_j = \sigma_j^2 \mathbf{I}$

$$p(\underline{x}_k \mid j,\underline{\theta}) = \frac{1}{\left(2\pi\sigma_j^2\right)^{\frac{l}{2}}} \exp\left[-\frac{\left\|\underline{x}_k - \underline{\mu}_j\right\|^2}{2\sigma_j^2}\right],$$

E-step:

$$Q(\boldsymbol{\Theta};\boldsymbol{\Theta}(t)) = \sum_{k=1}^{N}\sum_{j=1}^{J} P(j|\boldsymbol{x}_k;\boldsymbol{\Theta}(t))\left(-\frac{l}{2}\ln\sigma_j^2 - \frac{1}{2\sigma_j^2}\|\boldsymbol{x}_k-\boldsymbol{\mu}_j\|^2 + \ln P_j\right)$$

M-step: Maximizing the above with respect to $\boldsymbol{\mu}_j$, σ_j^2, and P_j results in

$$\begin{cases} \boldsymbol{\mu}_j(t+1) = \dfrac{\sum_{k=1}^{N} P(j|\boldsymbol{x}_k;\boldsymbol{\Theta}(t))\boldsymbol{x}_k}{\sum_{k=1}^{N} P(j|\boldsymbol{x}_k;\boldsymbol{\Theta}(t))} \\[2ex] \sigma_j^2(t+1) = \dfrac{\sum_{k=1}^{N} P(j|\boldsymbol{x}_k;\boldsymbol{\Theta}(t))\|\boldsymbol{x}_k-\boldsymbol{\mu}_j(t+1)\|^2}{\sum_{k=1}^{N} P(j|\boldsymbol{x}_k;\boldsymbol{\Theta}(t))} \\[2ex] P_j(t+1) = \dfrac{1}{N}\sum_{k=1}^{N} P(j|\boldsymbol{x}_k;\boldsymbol{\Theta}(t)) \end{cases}$$

$$P(j\,|\,\underline{x}_k;\underline{\Theta}(t)) = \frac{p(\underline{x}_k\,|\,j;\underline{\theta}(t))P_j(t)}{p(\underline{x}_k;\underline{\Theta}(t))} \quad , p(\underline{x}_k;\underline{\Theta}(t)) = \sum_{j=1}^{J} p(\underline{x}_k\,|\,j;\underline{\theta}(t))P_j(t)$$

Gaussian Mixture Model (GMM)

- A Gaussian mixture model represents a **distribution** as

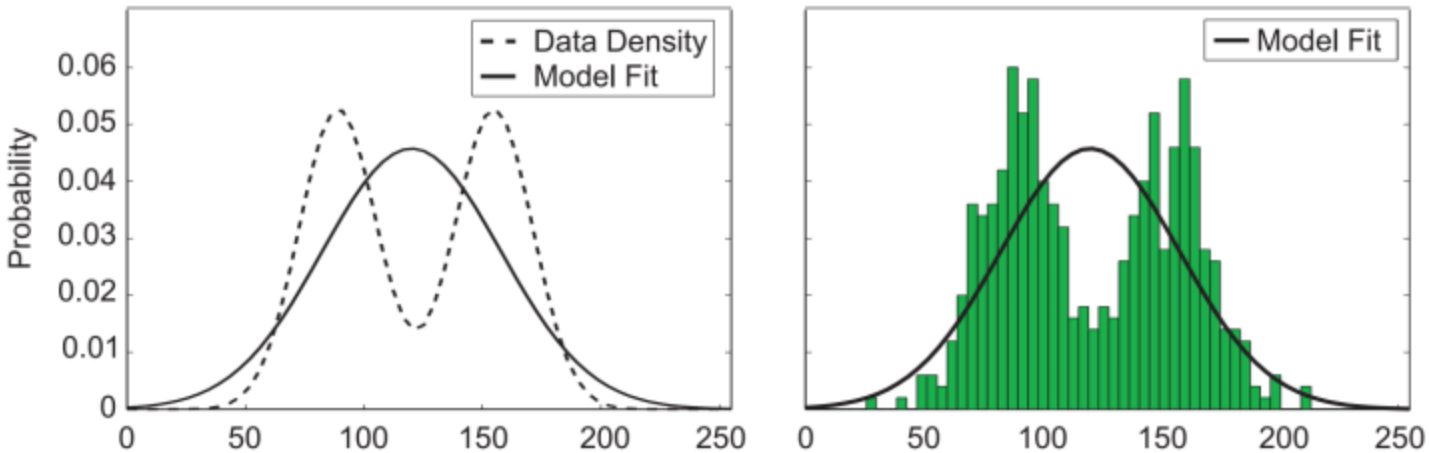
$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

with π_k the mixing coefficients, where:

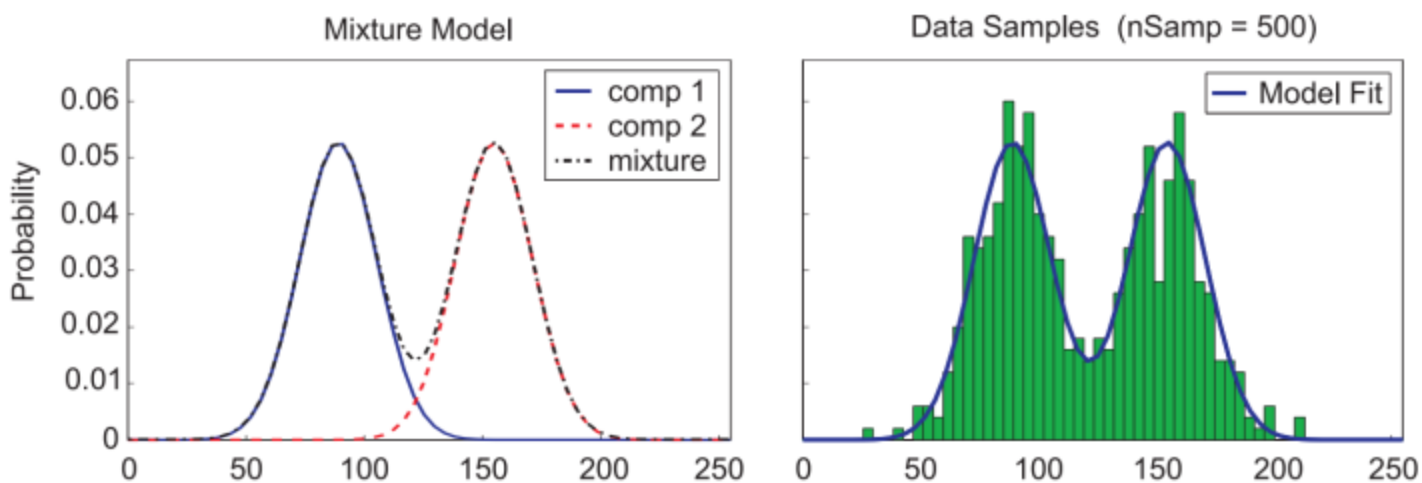
$$\sum_{k=1}^{K} \pi_k = 1 \quad \text{and} \quad \pi_k \geq 0 \quad \forall k$$

- GMM is a density estimator
- Where have we already used a density estimator?
- We know that neural nets are universal approximators of functions
- GMMs are **universal approximators of densities** (if you have enough Gaussians). Even diagonal GMMs are universal approximators.

- In the beginning of class, we tried to fit a Gaussian to data:



- Now, we are trying to fit a GMM (with $K = 2$ in this example):



Sampling from a Mixture Model

Generate u = uniform random number between 0 and 1

If $u < \pi_1$

  generate $x \sim N(x \mid \mu_1, \Sigma_1)$

elseif $u < \pi_1 + \pi_2$

  generate $x \sim N(x \mid \mu_2, \Sigma_2)$

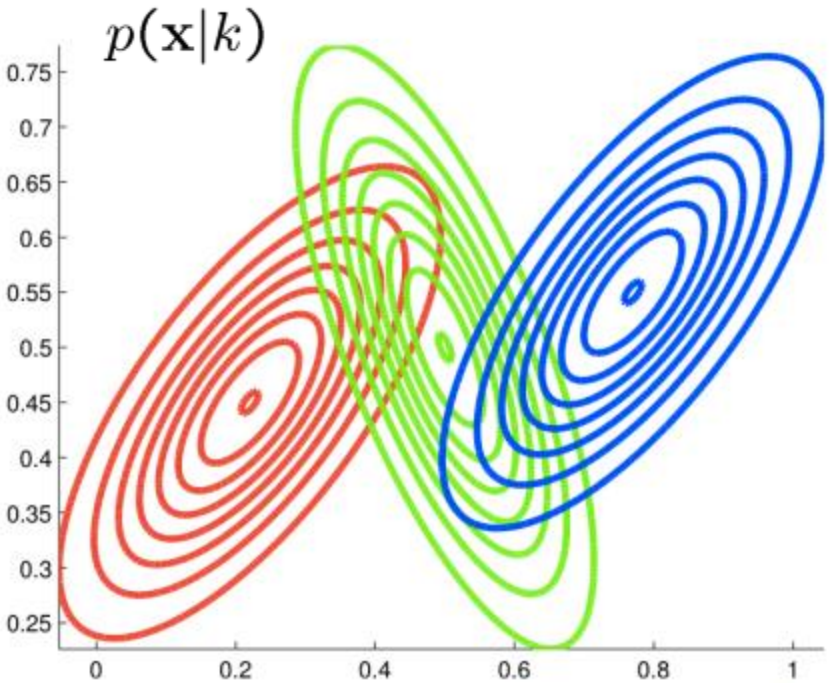
$\vdots$

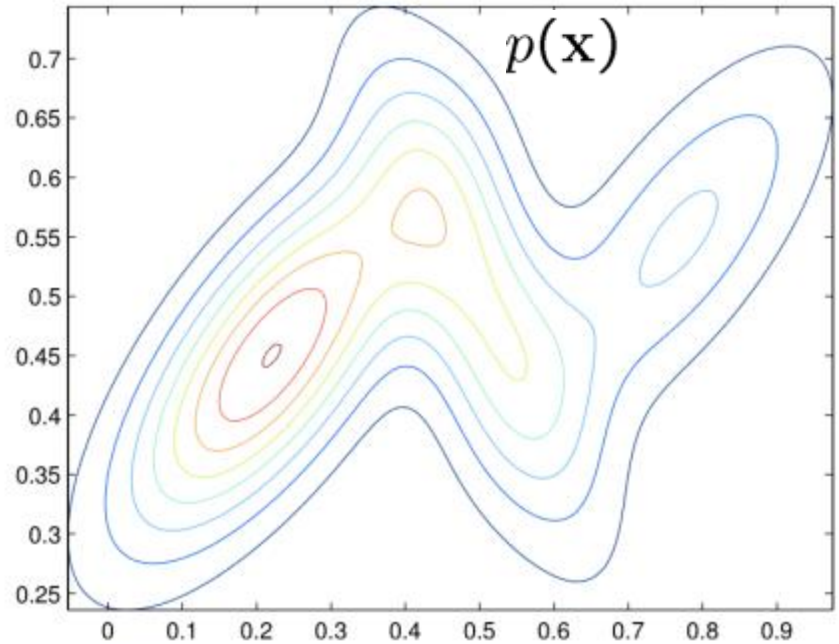
elseif $u < \pi_1 + \pi_2 + \ldots + \pi_{K-1}$

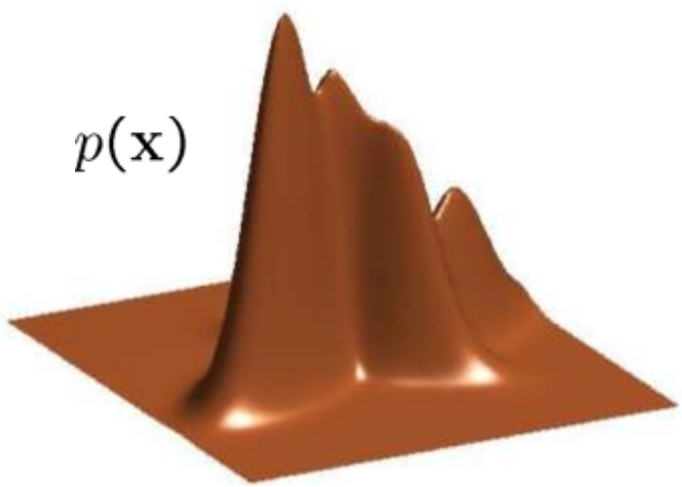
  generate $x \sim N(x \mid \mu_{K-1}, \Sigma_{K-1})$

else

  generate $x \sim N(x \mid \mu_K, \Sigma_K)$


$p(\mathbf{x}|k)$
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0
0.2
0.4
0.6
0.8
1


$p(\mathbf{x})$
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9


$p(\mathbf{x})$

- Maximum likelihood maximizes

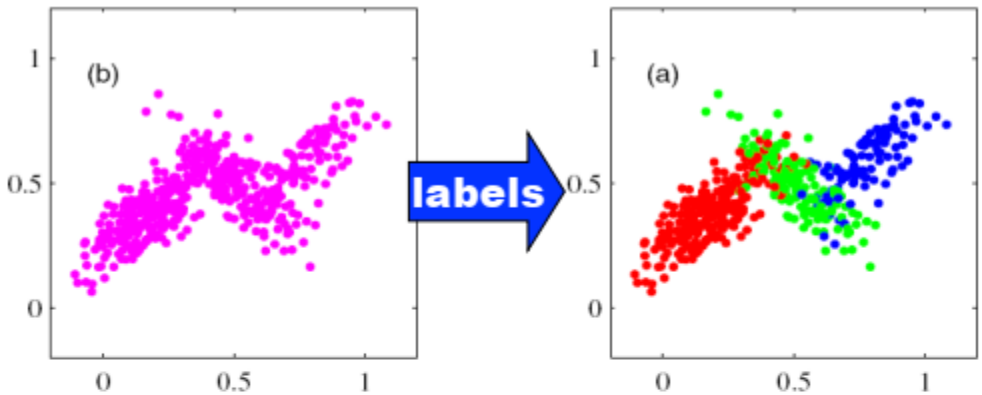
$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}^{(n)}|\mu_k, \Sigma_k) \right)$$

Uh-oh, log of a sum

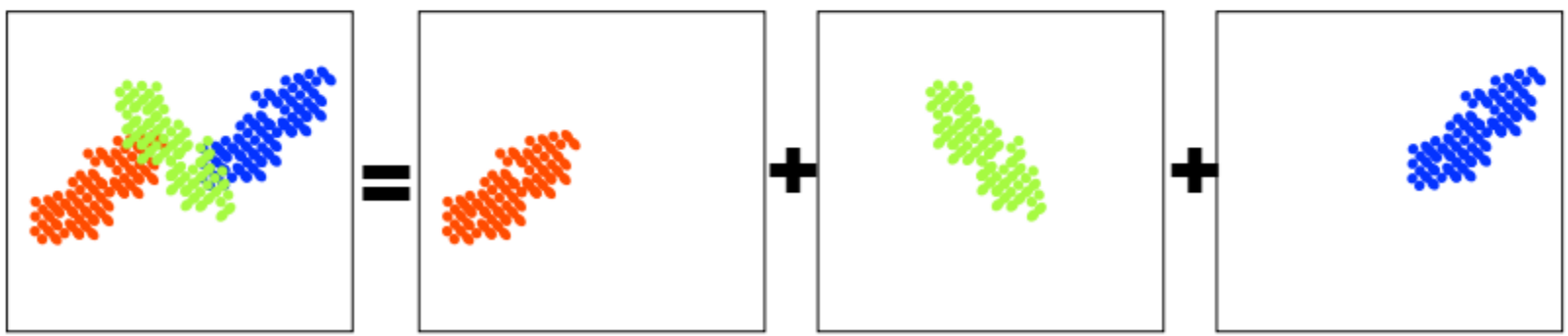
w.r.t $\Theta = \{\pi_k, \mu_k, \Sigma_k\}$

The maximum likelihood solution for the parameters no longer has a closed-form analytical solution.

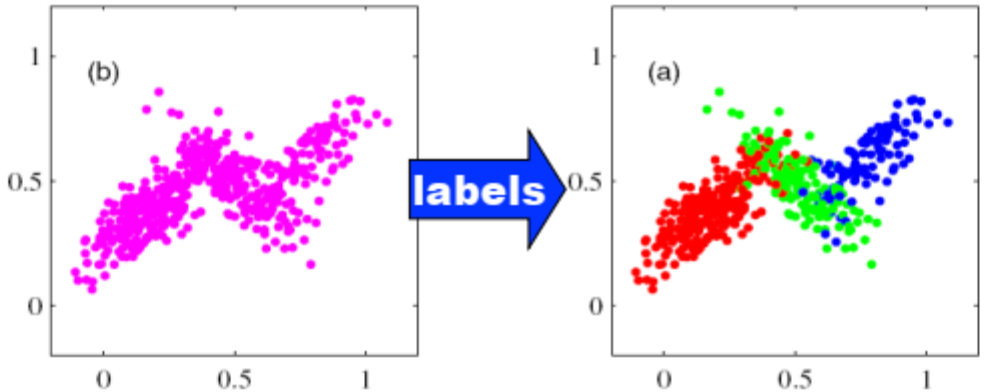
EM Algorithm



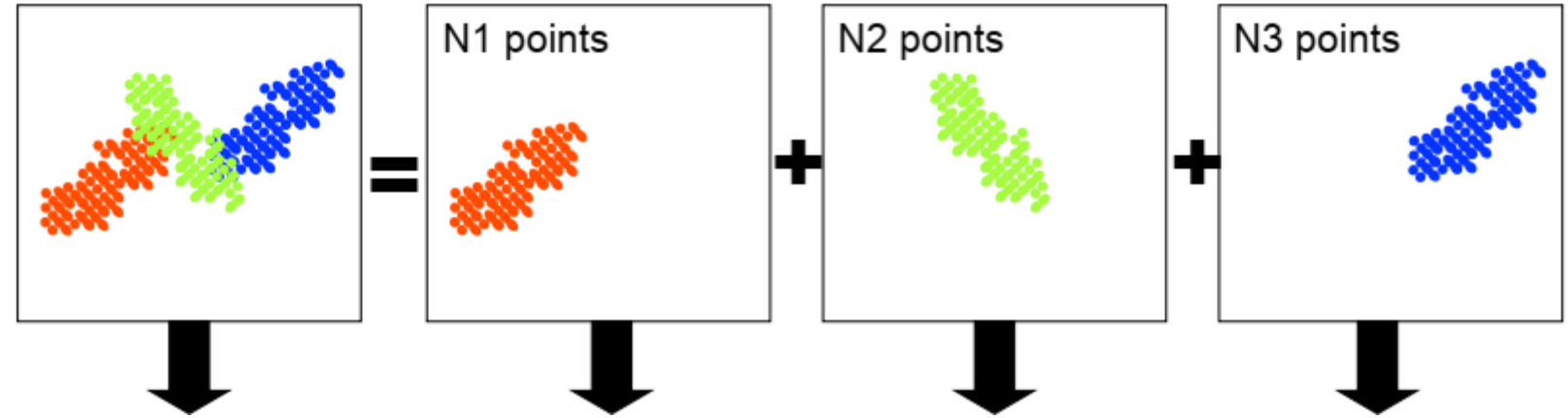
This lets us recover the underlying generating process decomposition:



EM Algorithm



And we can easily estimate each Gaussian, along with the mixture weights!



Estimate $\pi_k = N_k / N$ Estimate μ_1, σ_1 Estimate μ_2, σ_2 Estimate μ_3, σ_3

EM Algorithm

Remember that this was a problem...

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

If an oracle gave us the values of the latent variables (component that generated each point) we could work with the complete log likelihood

$$p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} N(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

and the log of that looks much better!

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

Latent Variable View

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

Note: for a given n, there are K of these latent variables, and only ONE of them is 1 (all the rest are 0)

This is thus equivalent to

$$\sum_{\substack{\text{all n for which}\\ z_{n,1}=1}} \ln\pi_1 + \ln\mathcal{N}(x_n|\mu_1,\Sigma_1)$$

$$+ \sum_{\substack{\text{all n for which}\\ z_{n,2}=1}} \ln\pi_2 + \ln\mathcal{N}(x_n|\mu_2,\Sigma_2) \quad + \quad \cdots \quad +$$

$$+ \sum_{\substack{\text{all n for which}\\ z_{n,K}=1}} \ln\pi_K + \ln\mathcal{N}(x_n|\mu_K,\Sigma_K)$$

Latent Variable View

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1)$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2)$$

$$+ \quad \cdots \quad +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K)$$

Latent Variable View

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1)}$$

can be estimated separately

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2)}$$

can be estimated separately

$$+ \quad \cdots \quad +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \boxed{\sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K)}$$

can be estimated separately

Latent Variable View

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n|\mu_1, \Sigma_1)$$

can be estimated separately

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n|\mu_2, \Sigma_2)$$

can be estimated separately

$$+ \cdots +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n|\mu_K, \Sigma_K)$$

can be estimated separately

These are coupled because the mixing weights all sum to 1, but it is no big deal to solve

Latent Variable View

- Unfortunately, oracles don't exist (or if they do, they won't talk to us)
- So we don't know values of the $z_{n,k}$ variables
- What EM proposes to do:
  - 1) compute $p(Z|X, \theta)$, the posterior distribution over $z_{n,k}$, given our current best guess at the values of θ
  - 2) compute the expected value of the log likelihood $\ln(p(X,Z|\theta))$ with respect to the distribution $p(Z|X, \theta)$
  - 3) find θ_{new} that maximizes that function. This is our new best guess at the values of θ.
  - 4) iterate...

Latent Variable View

- Since we don't know the latent variables, we instead take the expected value of the log likelihood with respect to their posterior distribution $P(Z|X, \theta)$. In the GMM case, this is equivalent to "softening" the binary latent variables to continuous ones (the expected values of the latent variables)

$$\ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} \underbrace{z_{nk}}_{\text{unknown discrete value 0 or 1}} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

$$\mathrm{E}_{\mathbf{z}}[\ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})] = \sum_{n=1}^{N} \sum_{k=1}^{K} \underbrace{\gamma_k(\mathbf{x}_n)}_{\text{known continuous value between 0 and 1}} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

Where $\gamma_j(\mathbf{x}_n)$ is $P(z_{nk} = 1)$

Latent Variable View

- So now, after replacing the binary latent variables with their continuous expected values:
- All points contribute to the estimation of all components
- Each point has unit mass to contribute, but splits it across the K components
- The amount of weight a point contributes to a component is proportional to the relative likelihood that the point was generated by that component

Latent Variable View (with an oracle)

$$\sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \pi_1 + \sum_{\substack{\text{all n for which} \\ z_{n,1}=1}} \ln \mathcal{N}(x_n|\mu_1, \Sigma_1)$$

can be estimated separately

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \pi_2 + \sum_{\substack{\text{all n for which} \\ z_{n,2}=1}} \ln \mathcal{N}(x_n|\mu_2, \Sigma_2)$$

can be estimated separately

$$+ \cdots +$$

$$+ \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \pi_K + \sum_{\substack{\text{all n for which} \\ z_{n,K}=1}} \ln \mathcal{N}(x_n|\mu_K, \Sigma_K)$$

can be estimated separately

these are coupled because the mixing weights all sum to 1, but it is no big deal to solve

Latent Variable View (with EM, $\gamma^{i}_{n,k}$ a constant at iteration i)

$$\sum_N \sum_K \gamma^{i}_{n,k} \ln \pi_1 + \sum_N \sum_K \gamma^{i}_{n,k} \ln \mathcal{N}(x_n | \mu_1, \Sigma_1)$$

can be estimated separately

$$+ \sum_N \sum_K \gamma^{i}_{n,k} \ln \pi_2 + \sum_N \sum_K \gamma^{i}_{n,k} \ln \mathcal{N}(x_n | \mu_2, \Sigma_2)$$

can be estimated separately

$$+ \cdots +$$

$$+ \sum_N \sum_K \gamma^{i}_{n,k} \ln \pi_K + \sum_N \sum_K \gamma^{i}_{n,k} \ln \mathcal{N}(x_n | \mu_K, \Sigma_K)$$

can be estimated separately

these are coupled because the mixing weights all sum to 1, but it is no big deal to solve

EM Algorithm for GMM

E

$$\gamma_j(\mathbf{x}_n) = \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

ownership weights

M

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

means

$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^\top}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

covariances

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

mixing probabilities

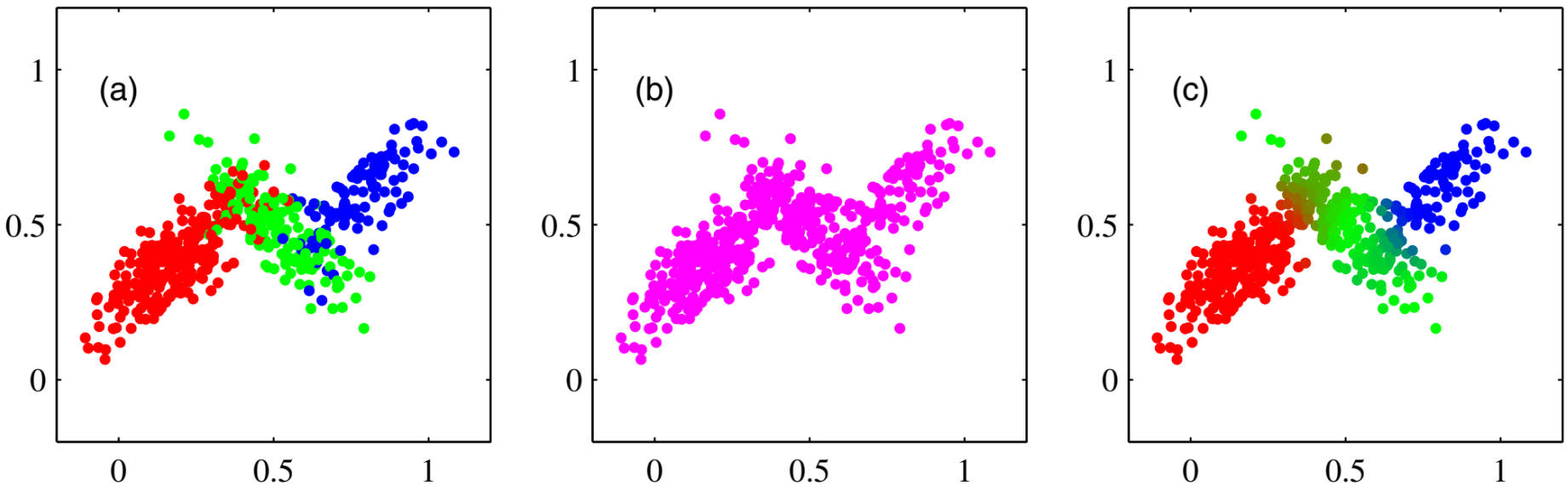
Another Approach

Posterior Probabilities

- We can think of the mixing coefficients as prior probabilities for the components
- For a given value of $\mathbf{x}$ we can evaluate the corresponding posterior probabilities, called *responsibilities*
- These are given from Bayes' theorem by

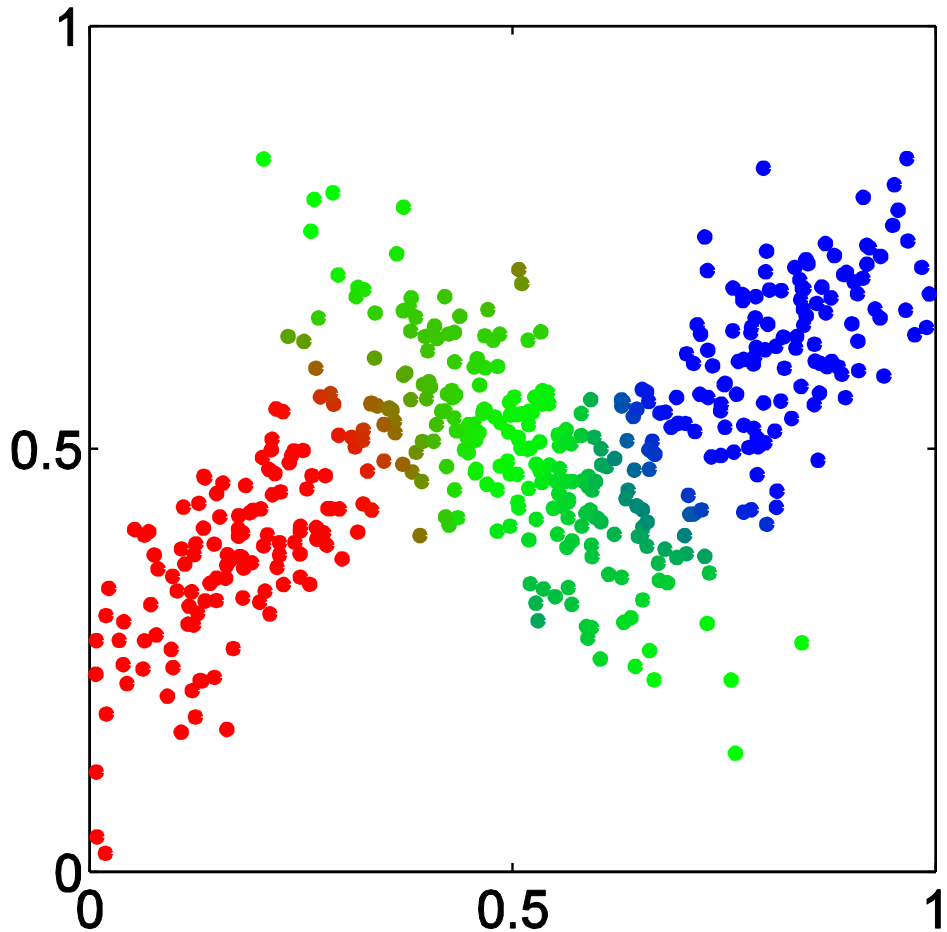
$$\gamma_k(\mathbf{x}) \equiv p(k|\mathbf{x}) = \frac{p(k)p(\mathbf{x}|k)}{p(\mathbf{x})}$$

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

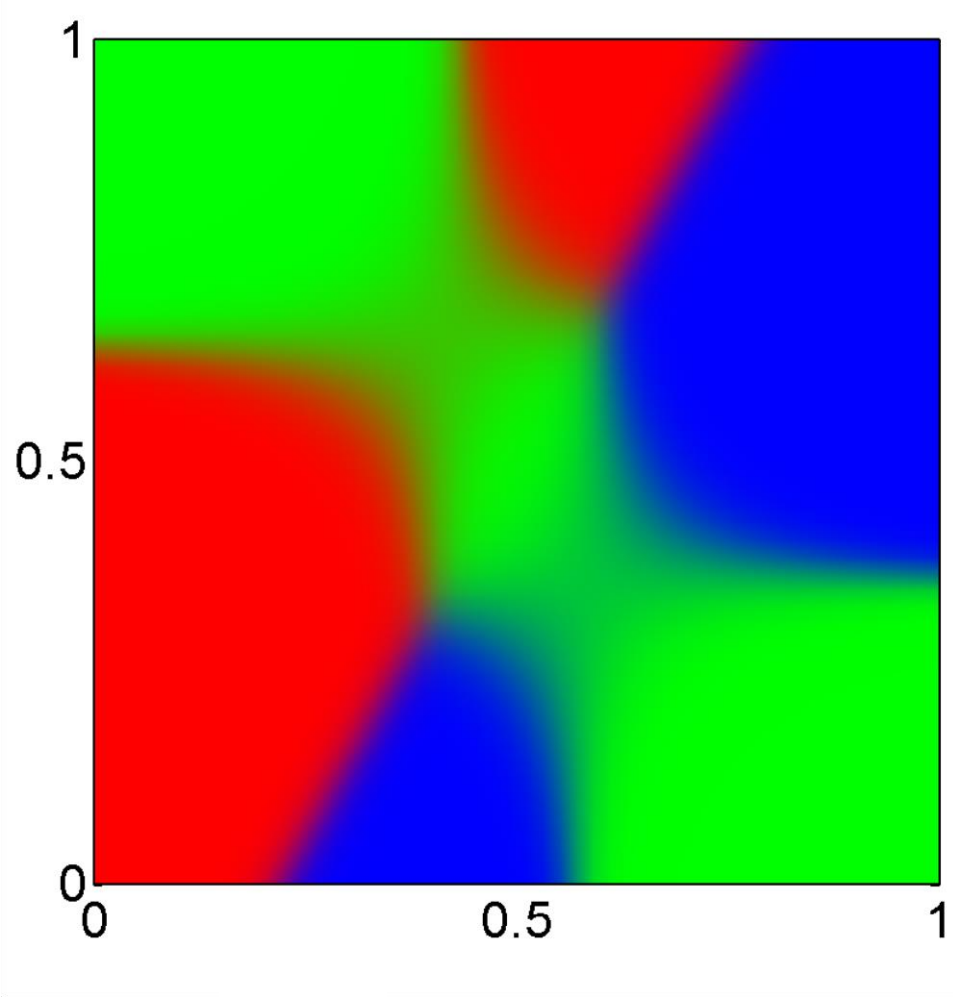


Example of 500 points drawn from the mixture of 3 Gaussians. (a) Samples from the joint distribution $p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$ in which the three states of $\mathbf{z}$, corresponding to the three components of the mixture, are depicted in red, green, and blue, and (b) the corresponding samples from the marginal distribution $p(\mathbf{x})$, which is obtained by simply ignoring the values of $\mathbf{z}$ and just plotting the $\mathbf{x}$ values. The data set in (a) is said to be **complete**, whereas that in (b) is **incomplete**. (c) The same samples in which the colours represent the value of the responsibilities $\gamma(z_{nk})$ associated with data point $\mathbf{x}_n$, obtained by plotting the corresponding point using proportions of red, blue, and green ink given by $\gamma(z_{nk})$ for $k = 1, 2, 3$, respectively.

Posterior Probabilities (colour coded)



Posterior Probability Map



Maximum Likelihood for the GMM

- Gaussian mixture distribution

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

- The log likelihood function takes the form

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

- Note: sum over components appears *inside* the log
- There is no closed form solution for maximum likelihood

Problems and Solutions

- How to maximize the log likelihood
  - solved by expectation-maximization (EM) algorithm
- How to avoid singularities in the likelihood function
  - solved by a Bayesian treatment
- How to choose number K of components
  - also solved by a Bayesian treatment

EM Algorithm – Informal Derivation

- Let us proceed by simply differentiating the log likelihood
- Setting derivative with respect to $\boldsymbol{\mu}_j$ equal to zero gives

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

giving

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)\mathbf{x}_n}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

which is simply the weighted mean of the data

EM Algorithm – Informal Derivation

- Similarly for the covariances

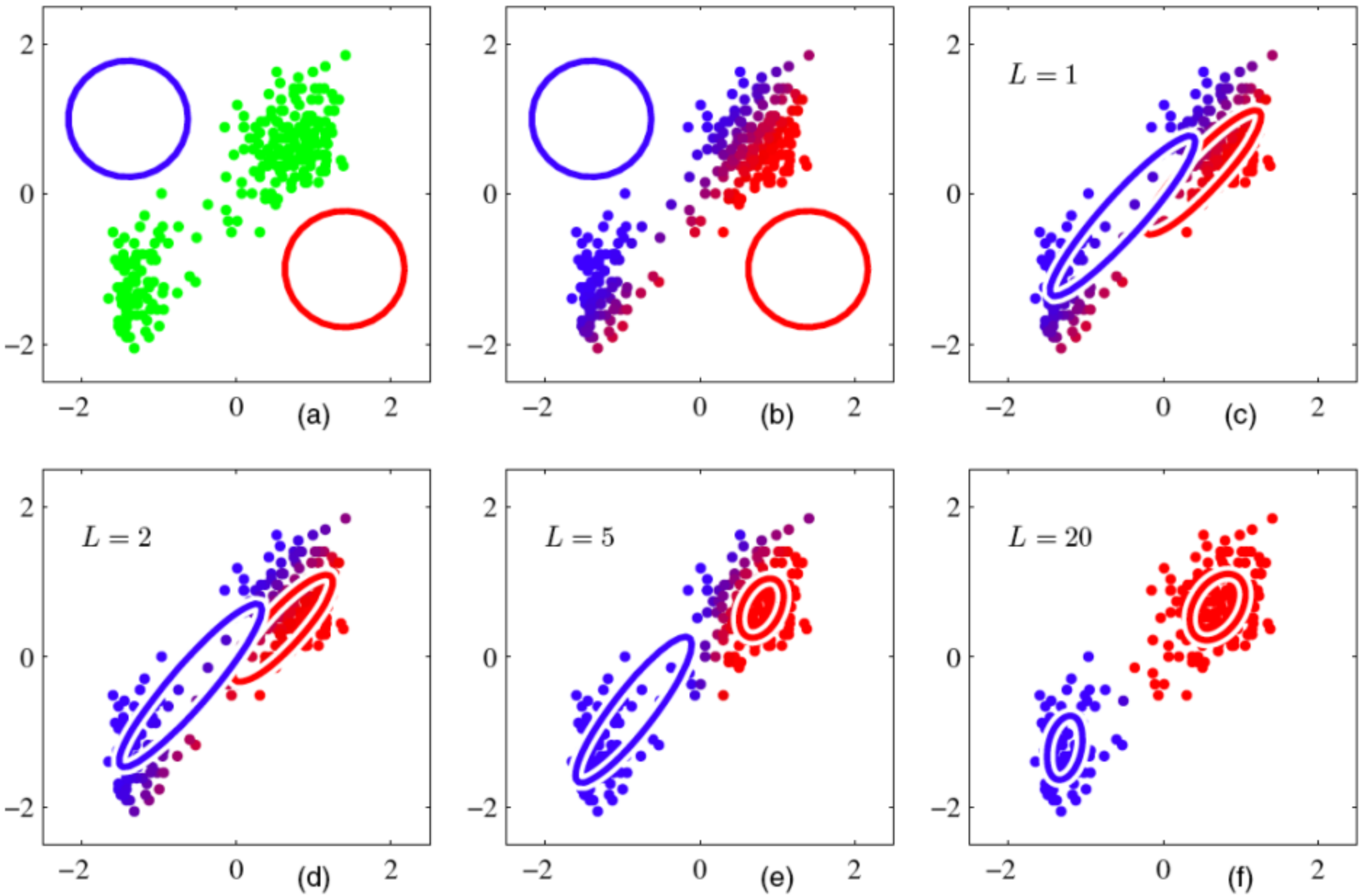
$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^{\mathsf{T}}}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

- For mixing coefficients use a Lagrange multiplier. After maximizing

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right) \Rightarrow \pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

EM Algorithm – Informal Derivation

- The solutions are not closed form since they are coupled
- Suggests an iterative scheme for solving them:
  - Make initial guesses for the parameters
  - Alternate between the following two stages:
    1. E-step: evaluate responsibilities
    2. M-step: update parameters using ML results

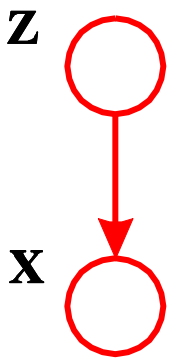


Slide from: Christopher M. Bishop

EM – Latent Variable Viewpoint

- Binary latent variables $\mathbf{z} = \{z_{kn}\}$ describing which component generated each data point
- Conditional distribution of observed variable

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



- Prior distribution of latent variables

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

- Marginalizing over the latent variables we obtain

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

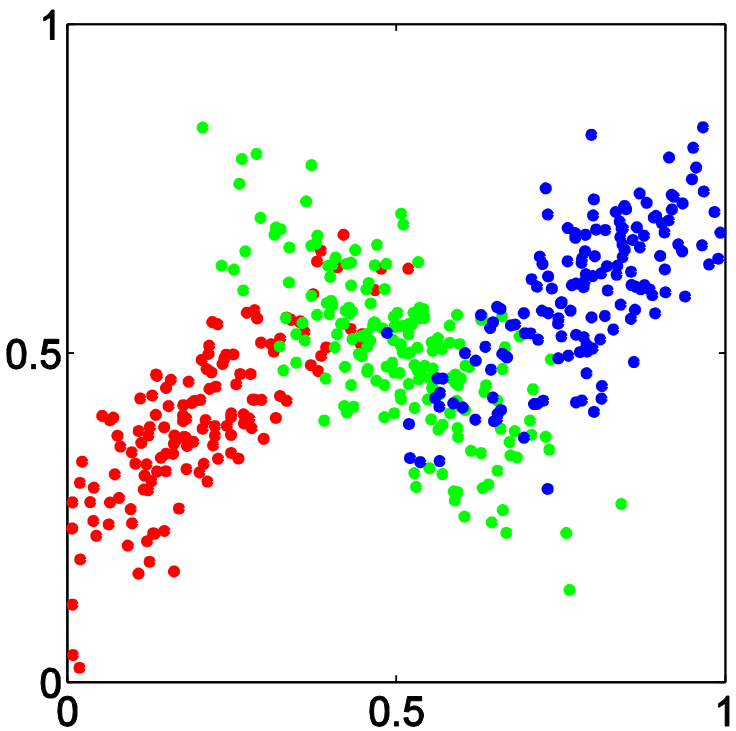
Slide from: Christopher M. Bishop

Expected Value of Latent Variable

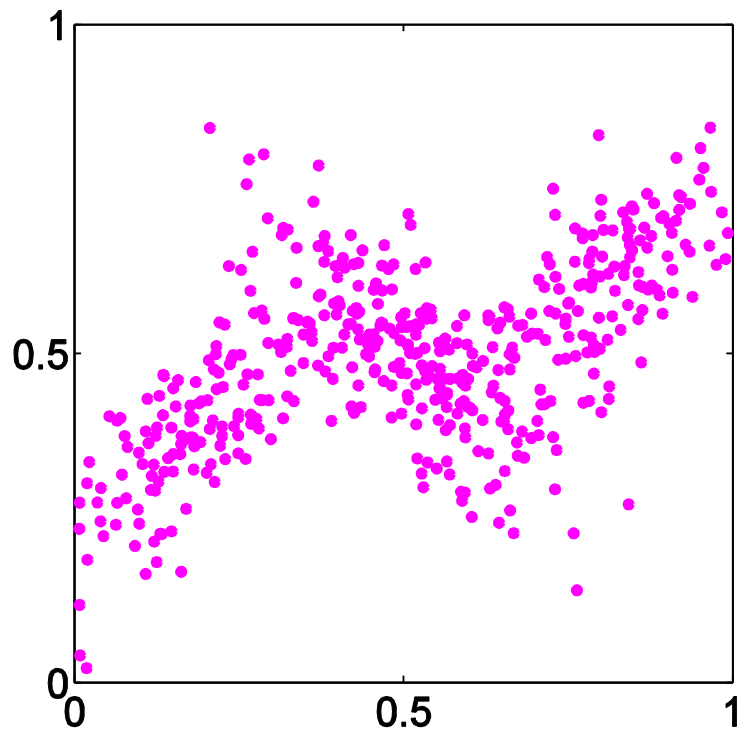
- From Bayes' theorem

$$
\begin{aligned}
\mathsf{E}[z_{ni}] &= \frac{\sum_{z_{ni}} z_{ni} \left[\pi_i p(\mathbf{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right]^{z_{ni}}}{\sum_{z_{ni}} \left[\pi_i p(\mathbf{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right]^{z_{ni}}} \\
&= \frac{\pi_i \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \\
&= \gamma_i(\mathbf{x}_n)
\end{aligned}
$$

Complete and Incomplete Data



complete



incomplete

Latent Variable View of EM

- If we knew the values for the latent variables, we would maximize the complete-data log likelihood

$$\ln p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}$$

  which gives a trivial closed-form solution (fit each component to the corresponding set of data points)
- We don't know the values of the latent variables
- However, for given parameter values we can compute the expected values of the latent variables

Expected Complete-Data Log Likelihood

- Suppose we make a guess $\boldsymbol{\theta}_{\text{old}}$ for the parameter values (means, covariances and mixing coefficients)
- Use these to evaluate the responsibilities
- Consider expected complete-data log likelihood

$$\mathsf{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})] = \sum_{n=1}^{N} \sum_{i=1}^{K} \gamma_i(\mathbf{x}_n) \left\{ \ln \pi_i + \ln \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right\}$$

  where responsibilities are computed using $\boldsymbol{\theta}_{\text{old}}$
- We are implicitly 'filling in' latent variables with best guess
- Keeping the responsibilities fixed and maximizing with respect to the parameters give the previous results

❖Textbook p 48

Example 2.8

Figure 2.17a shows $N = 100$ points in the two-dimensional space, which have been drawn from a multimodal distribution. The samples were generated using two Gaussian random generators $\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$, $\mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$, with

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix}$$

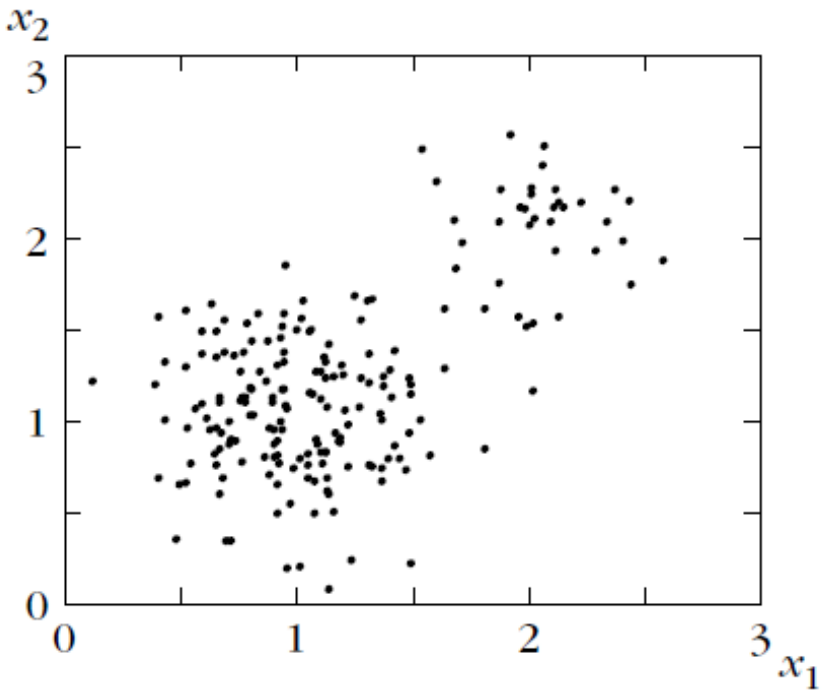
and covariance matrices

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.1 \end{bmatrix}$$

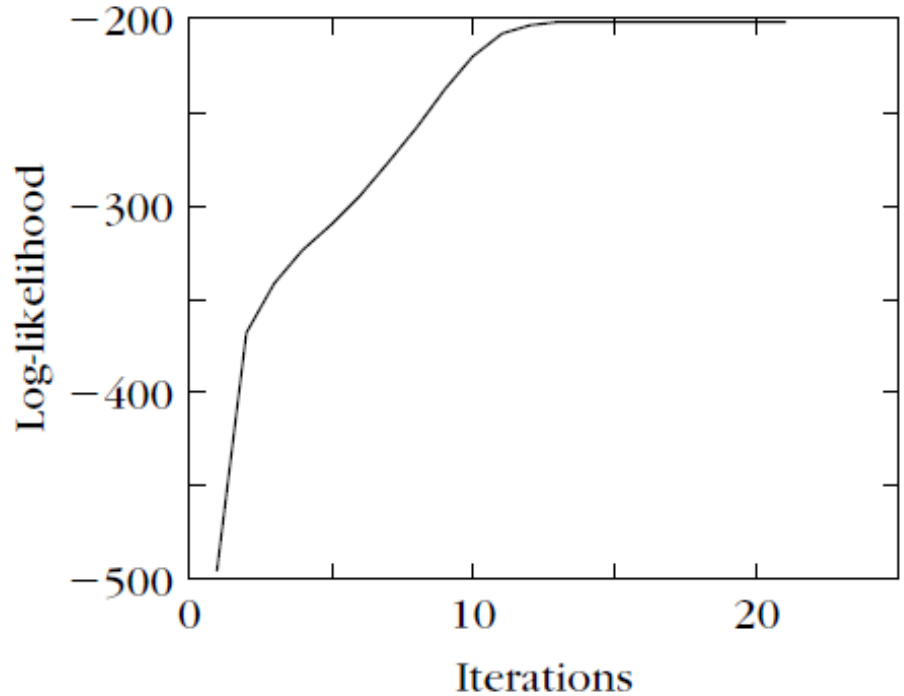
respectively. Each time a sample $\boldsymbol{x}_k$, $k = 1, 2, \ldots, N$, is to be generated a coin is tossed. The corresponding probabilities for heads or tails are $P(H) \equiv P = 0.8$, $P(T) = 1 - P = 0.2$, respectively. If the outcome of the coin flip is heads, the sample $\boldsymbol{x}_k$ is generated from $\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$. Otherwise, it is drawn from $\mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$. This is the reason that in Figure 2.17a the space around the point $[1.0, 1.0]^T$ is more densely populated. The pdf of the data set can obviously be written as

$$p(\boldsymbol{x}) = g(\boldsymbol{x}; \boldsymbol{\mu}_1, \sigma_1^2)P + g(\boldsymbol{x}; \boldsymbol{\mu}_2, \sigma_2^2)(1 - P) \tag{2.103}$$

where $g(\cdot; \boldsymbol{\mu}, \sigma^2)$ denotes the Gaussian pdf with parameters the mean value $\boldsymbol{\mu}$ and a diagonal covariance matrix, $\Sigma = \text{diag}\{\sigma^2\}$, having σ^2 across the diagonal and zeros



(a)



(b)

elsewhere. Equation (2.103) is a special case of the more general formulation given in (2.86). The goal is to compute the maximum likelihood estimate of the unknown parameters vector

$$\boldsymbol{\Theta}^T = [P, \boldsymbol{\mu}_1^T, \sigma_1^2, \boldsymbol{\mu}_2^T, \sigma_2^2]$$

based on the available $N = 100$ points. The full training data set consists of the sample pairs $(\boldsymbol{x}_k, j_k)$, $k = 1, 2, \ldots, N$, where $j_k \in \{1, 2\}$, and it indicates the origin of each observed sample. However, only the points $\boldsymbol{x}_k$ are at our disposal, with the "label" information being hidden from us. To understand this issue better and gain more insight into the rationale behind the EM methodology, it may be useful to arrive at Eq. (2.95) from a slightly different route. Each of the random vectors, $\boldsymbol{x}_k$, can be thought of as the result of a linear combination of two other random vectors; namely,

$$\boldsymbol{x}_k = \alpha_k \boldsymbol{x}_k^1 + (1 - \alpha_k)\boldsymbol{x}_k^2$$

where $\boldsymbol{x}_k^1$ is drawn from $\mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$ and $\boldsymbol{x}_k^2$ from $\mathcal{N}(\boldsymbol{\mu}_2, \Sigma_2)$. The binary coefficients $\alpha_k \in \{0, 1\}$ are randomly chosen with probabilities $P(1) = P = 0.8$, $P(0) = 0.2$. If the values of the α_ks, $k = 1, 2, \ldots, N$, were known to us, the log-likelihood function in (2.93) would be written as

$$L(\boldsymbol{\Theta};\alpha) = \sum_{k=1}^{N} \alpha_k \ln\left\{g(\boldsymbol{x}_k;\boldsymbol{\mu}_1,\sigma_1^2)P\right\} + \sum_{k=1}^{N}(1-\alpha_k)\ln\left\{g(\boldsymbol{x}_k;\boldsymbol{\mu}_2,\sigma_2^2)(1-P)\right\} \quad (2.104)$$

since we can split the summation in two parts, depending on the origin of each sample $\boldsymbol{x}_k$. However, this is just an "illusion" since the $\alpha_k s$ are unknown to us. Motivated by the spirit behind the EM algorithm, we substitute in (2.104) the respective mean values $E[\alpha_k|\boldsymbol{x}_k;\hat{\boldsymbol{\Theta}}]$, given an estimate, $\hat{\boldsymbol{\Theta}}$, of the unknown parameter vector. For the needs of our example we have

$$E[\alpha_k|\boldsymbol{x}_k;\hat{\boldsymbol{\Theta}}] = 1 \times P(1|\boldsymbol{x}_k;\hat{\boldsymbol{\Theta}}) + 0 \times (1 - P(1|\boldsymbol{x}_k;\hat{\boldsymbol{\Theta}})) = P(1|\boldsymbol{x}_k;\hat{\boldsymbol{\Theta}}) \quad (2.105)$$

Substitution of (2.105) into (2.104) results in (2.95) for the case of $J = 2$.

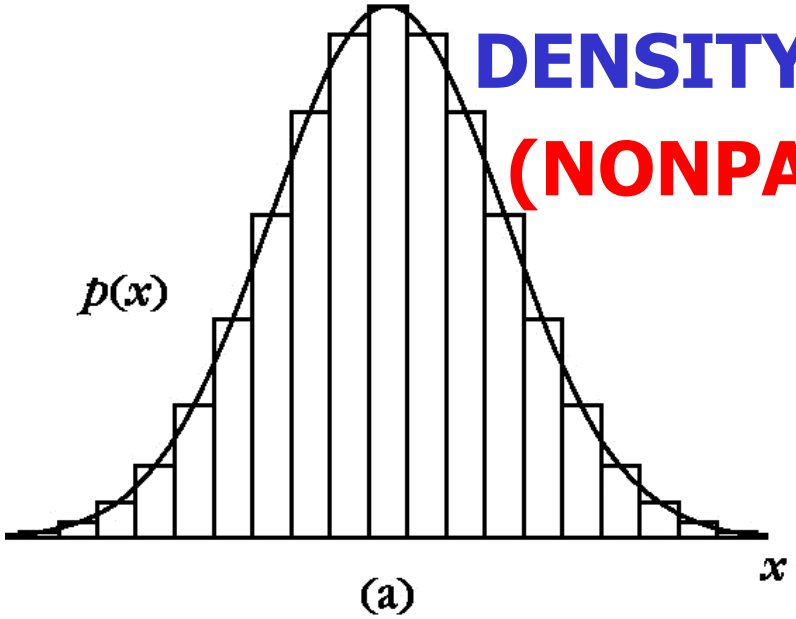
We are now ready to apply the EM algorithm [Eqs. (2.98)–(2.102)] to the needs of our example. The initial values were chosen to be

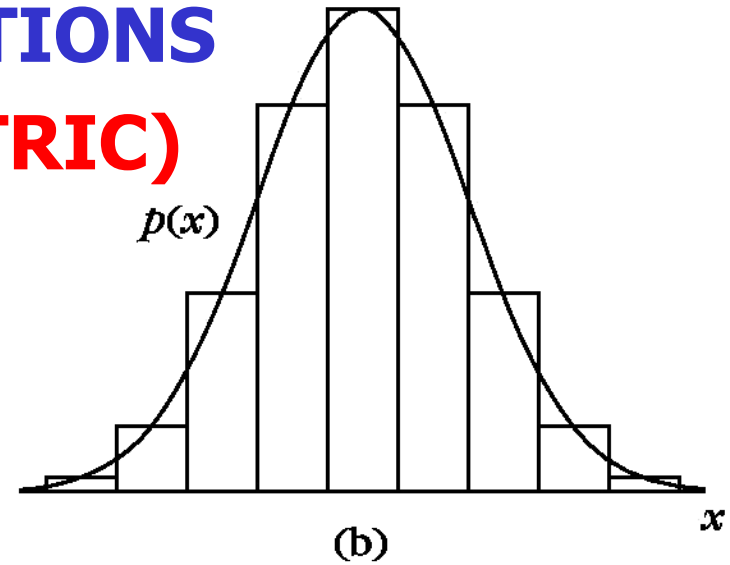
$$\boldsymbol{\mu}_1(0) = [1.37,\ 1.20]^T, \quad \boldsymbol{\mu}_2(0) = [1.81,\ 1.62]^T, \quad \sigma_1^2 = \sigma_2^2 = 0.44, \quad P = 0.5$$

Figure 2.17b shows the log-likelihood as a function of the number of iterations. After convergence, the obtained estimates for the unknown parameters are

$$\boldsymbol{\mu}_1 = [1.05,\ 1.03]^T,\ \boldsymbol{\mu}_2 = [1.90,\ 2.08]^T,\ \sigma_1^2 = 0.10,\ \sigma_2^2 = 0.06,\ P = 0.844 \quad (2.106)$$

ESTIMATION OF UNKNOWN PROBABILITY DENSITY FUNCTIONS (NONPARAMETRIC)





Probability that a vector **x** will fall in region R is:

$$\boxed{P = \int_{\Re} p(\mathbf{x}')d\mathbf{x}' \quad (1)}$$

P is a smoothed (or averaged) version of the density function $p(\mathbf{x})$ if we have a sample of size N; therefore, the probability that k_N points fall in R is then:

Prob. that the rest are not

$$P_k = BIN(k_N \mid N, P) = \binom{N}{k_N} P^{k_N}(1-P)^{N-k_N} = \frac{N!}{k_N!(N-k_N)!} P^{k_N}(1-P)^{N-k_N} \qquad (2)$$

No. of unique splits k vs N-k_N

Prob. that k_N of particular x-es are in R

and the expected and variance value for k_N is:

$$E(k_N) = NP \quad , Var(k_N)=NP(1-P) \qquad (3)$$

What is ML estimation of $P = \theta$?

$$\nabla_P \ln(P_k) = \nabla_P \left(\ln \binom{N}{k_N} + k_N \ln(P) + (N - k_N)\ln(1-P) \right) = \frac{k_N}{P} - \frac{N - k_N}{1-P} = 0$$

$\underset{\theta}{Max}(P_k \mid \boldsymbol{\theta})$ is reached for $\hat{\boldsymbol{\theta}} = \frac{k_N}{N} \cong P$ (4)

Therefore, the ratio k_N/N is a good estimate for the probability P and hence for the density function p.

- If $p(\mathbf{x})$ is continuous and that the region R is so small that p does not vary significantly within it, we can write:

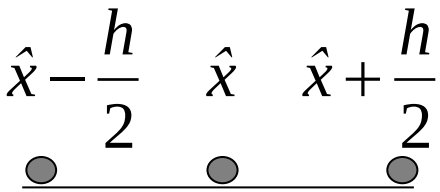
$$\boxed{\int_{\Re} p(\mathbf{x}')d\mathbf{x}' \cong p(\mathbf{x})\, V} \qquad (5)$$

where $\mathbf{x}$ is a point within R and V the volume enclosed by R.

Combining equation (1) , (4) and (5) yields:

$$\boxed{p(\mathbf{x}) \cong \frac{k_N / N}{V}}$$

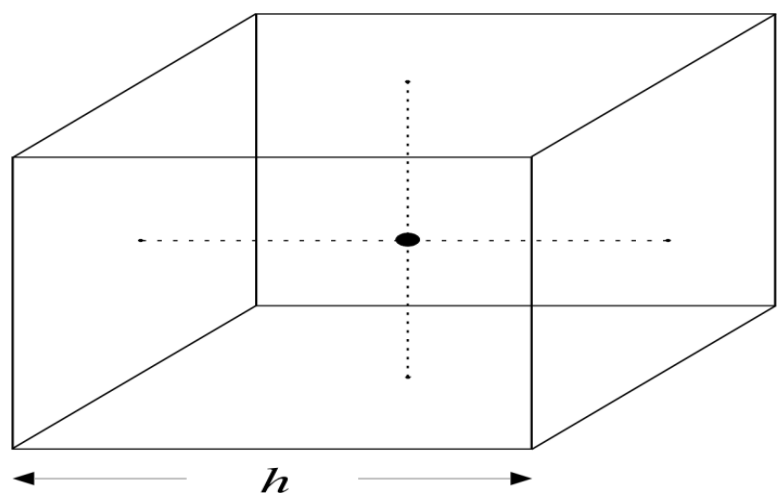
$$\hat{p}(x) \equiv \hat{p}(\hat{x}) = \frac{1}{h} \frac{k_N}{N}, \quad |x\text{-}\hat{x}| \le \frac{h}{2}$$



If $p(x)$ is continuous, $\hat{p}(x) \to p(x)$ as $N \to \infty$, if

$$h_N \to 0, \qquad k_N \to \infty, \qquad \frac{k_N}{N} \to 0$$

- Parzen Windows
  - Divide the multidimensional space in hypercubes



➢ Define

$$\varphi(\underline{x}_i) = \begin{cases} 1 & \text{for } |x_{ij}| \le 1/2 \\ 0 & \text{otherwise} \end{cases}$$

- That is, it is 1 inside a unit side hypercube centered at 0

$$\boxed{\hat{p}(\underline{x}) = \frac{1}{h^l}\left(\frac{1}{N}\sum_{i=1}^{N}\varphi\left(\frac{\underline{x}_i - \underline{x}}{h}\right)\right)}$$

a hypercube with length of side h centered at $\boldsymbol{x}$

$$\hat{p}(\underline{x}) = \frac{1}{volume} \times \frac{1}{N} \times$$

$$\times number\ of\ \ points\ inside\ \text{ an } h\text{-side hypercube centered at } \underline{x}$$

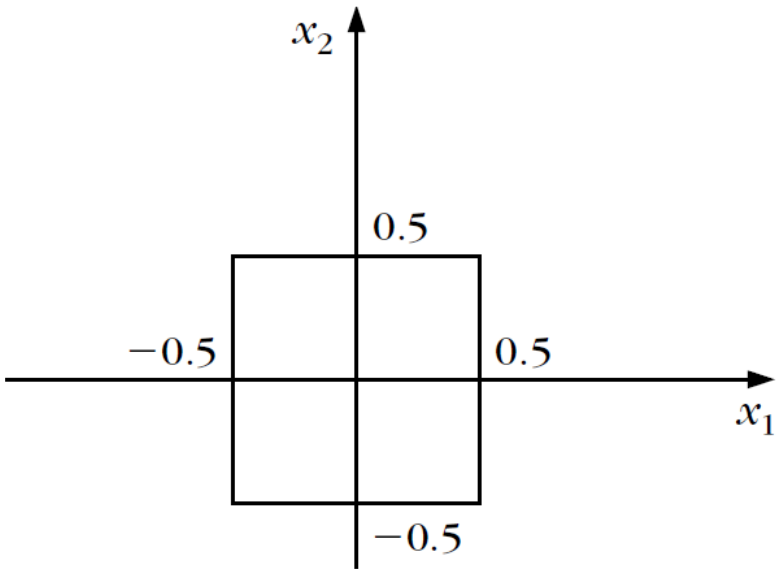
- The problem: $p(\underline{x})$ continuous

  $\varphi(.)$ discontinu ous

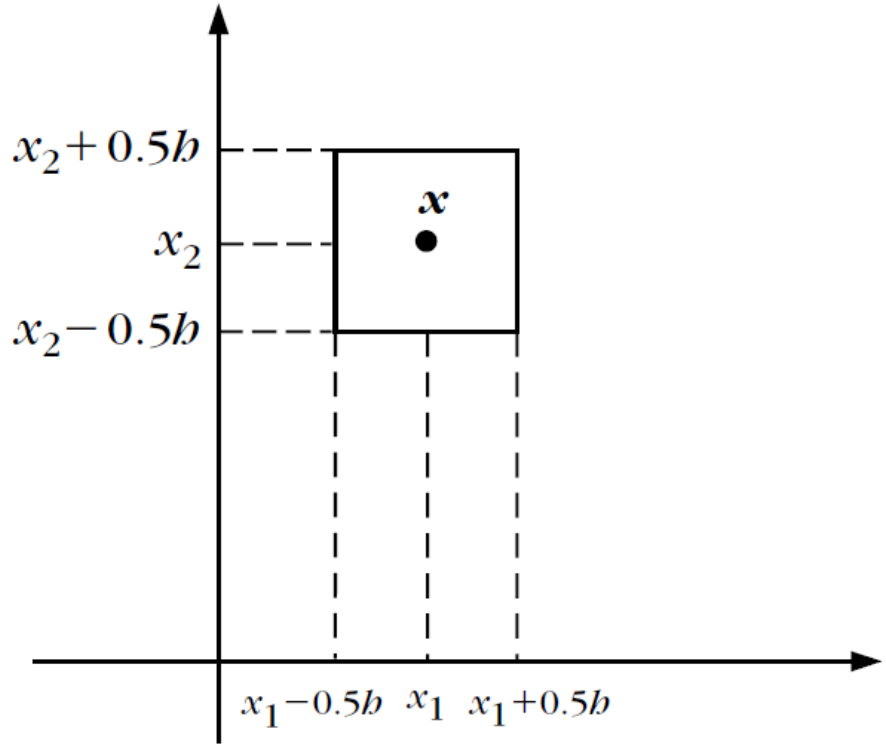
- Parzen windows-kernels-potential functions

$$\phi(\underline{x}) \text{ is smooth} \quad , \quad \phi(\underline{x}) \ge 0, \quad \int_{\underline{x}} \phi(\underline{x})d\,\underline{x} = 1$$

e.g. $=N(\mathbf{0}, \boldsymbol{I})$



The function $\phi(\boldsymbol{x}_i)$ is equal to one for every point, $\boldsymbol{x}_i$, inside the square of unit side length, centered at the origin and equal to zero for every point outside it.



The function $\phi(\frac{\underline{x}_i - \underline{x}}{h})$ is equal to unity for every point $\boldsymbol{x}_i$ inside the square with side length equal to h, centered at $\boldsymbol{x}$ and zero for all the other points.

➢ Mean value

$$E[\hat{p}(\underline{x})] = \frac{1}{h^l}\left(\frac{1}{N}\sum_{i=1}^{N} E[\phi(\frac{\underline{x}_i - \underline{x}}{h})]\right) = \int_{\underline{x}'} \frac{1}{h^l}\phi(\frac{\underline{x}' - \underline{x}}{h}) p(\underline{x}')d\underline{x}'$$

- $h \to 0,\ \frac{1}{h^l} \to \infty$
- $h \to 0$ the width of $\varphi(\frac{\underline{x}' - \underline{x}}{h}) \to 0$
- $\int \frac{1}{h^l}\phi(\frac{\underline{x}' - \underline{x}}{h})d\underline{x} = 1$
- $h \to 0 \qquad \frac{1}{h^l}\phi(\frac{\underline{x}}{h}) \to \delta(\underline{x})$

$$E[\hat{p}(x)] = \int_{\underline{x}'} \delta(\underline{x}' - \underline{x}) p(\underline{x}')d\underline{x}' = p(\underline{x})$$

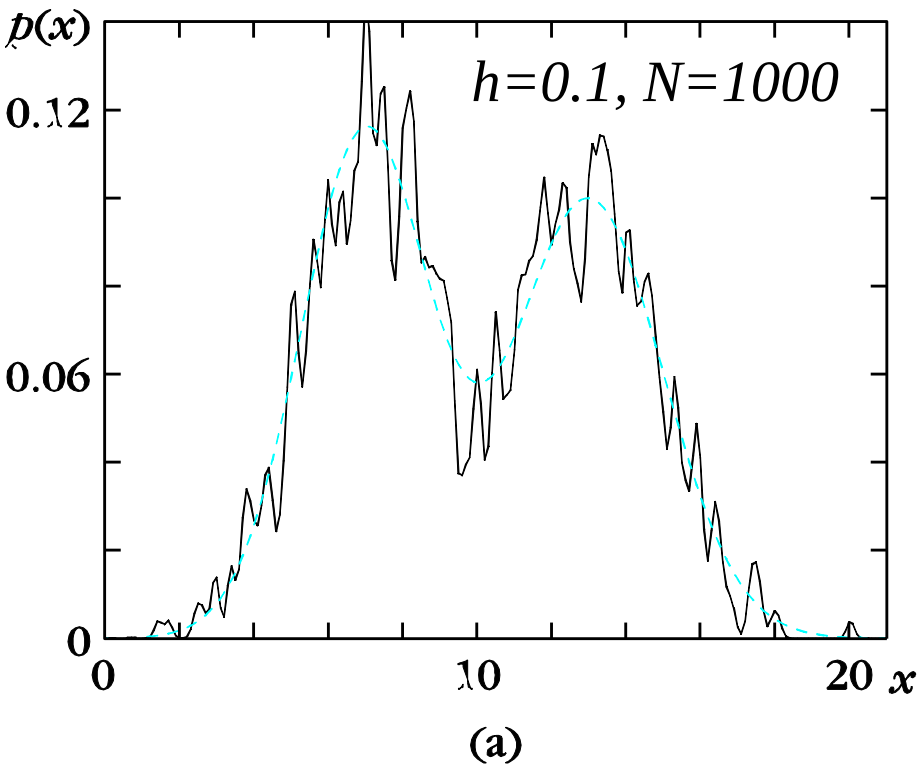
Hence $\boxed{\text{unbiased in the limit}}$

If:

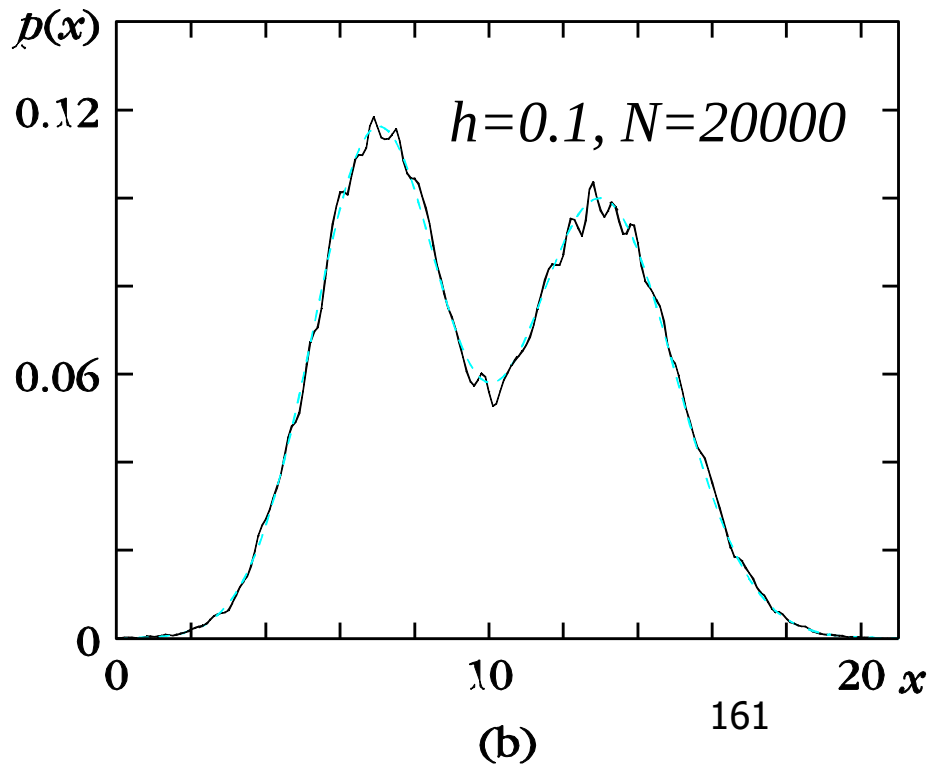
$$h \to 0, N \to \infty, h_N \to \infty$$

asymptotically unbiased

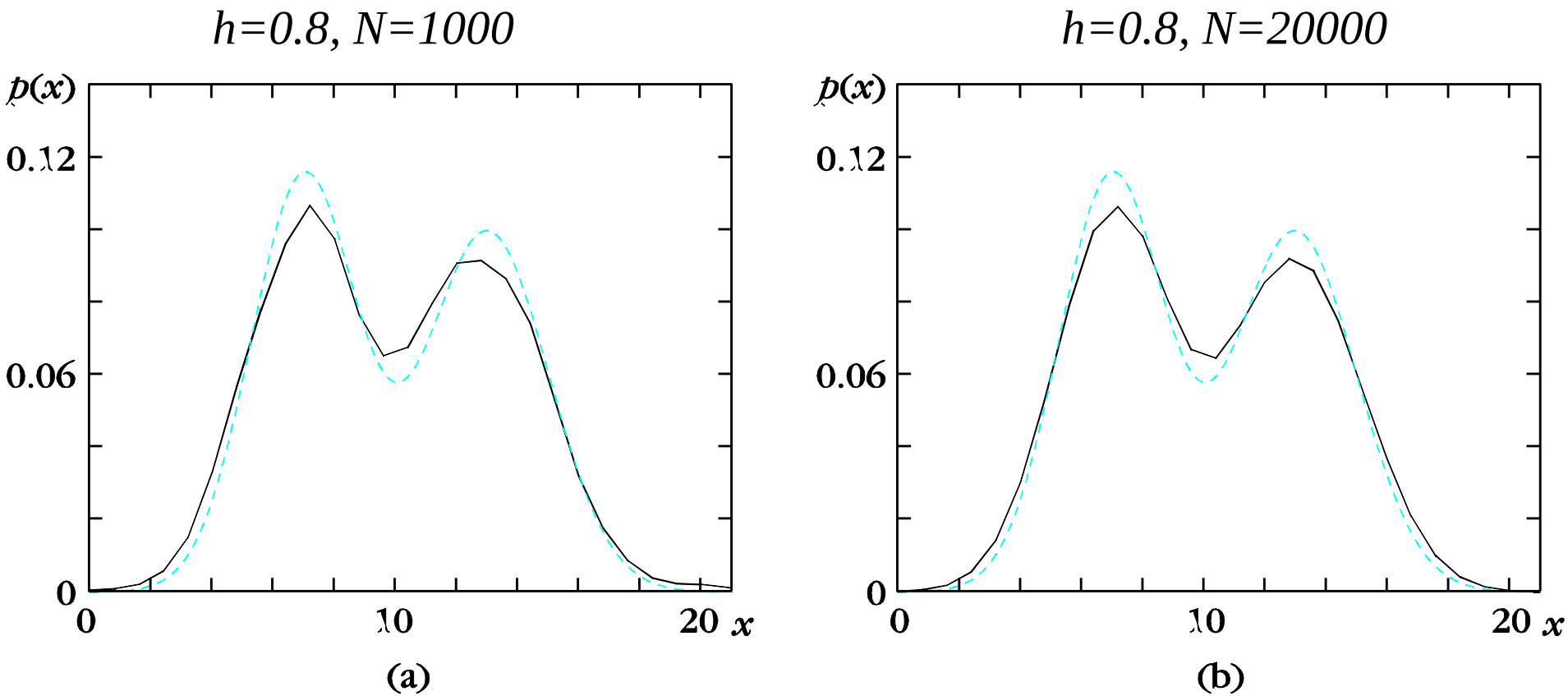
- Variance
  - The smaller the h the higher the variance



(a)



(b)



- The higher the N the better the accuracy

❖ Application to classification:

➢ The method

- Remember: l_{12} likelihood ratio

$$l_{12} = \frac{p(\underline{x} \mid \omega_1)}{p(\underline{x} \mid \omega_2)} > (<) \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}} = \theta$$

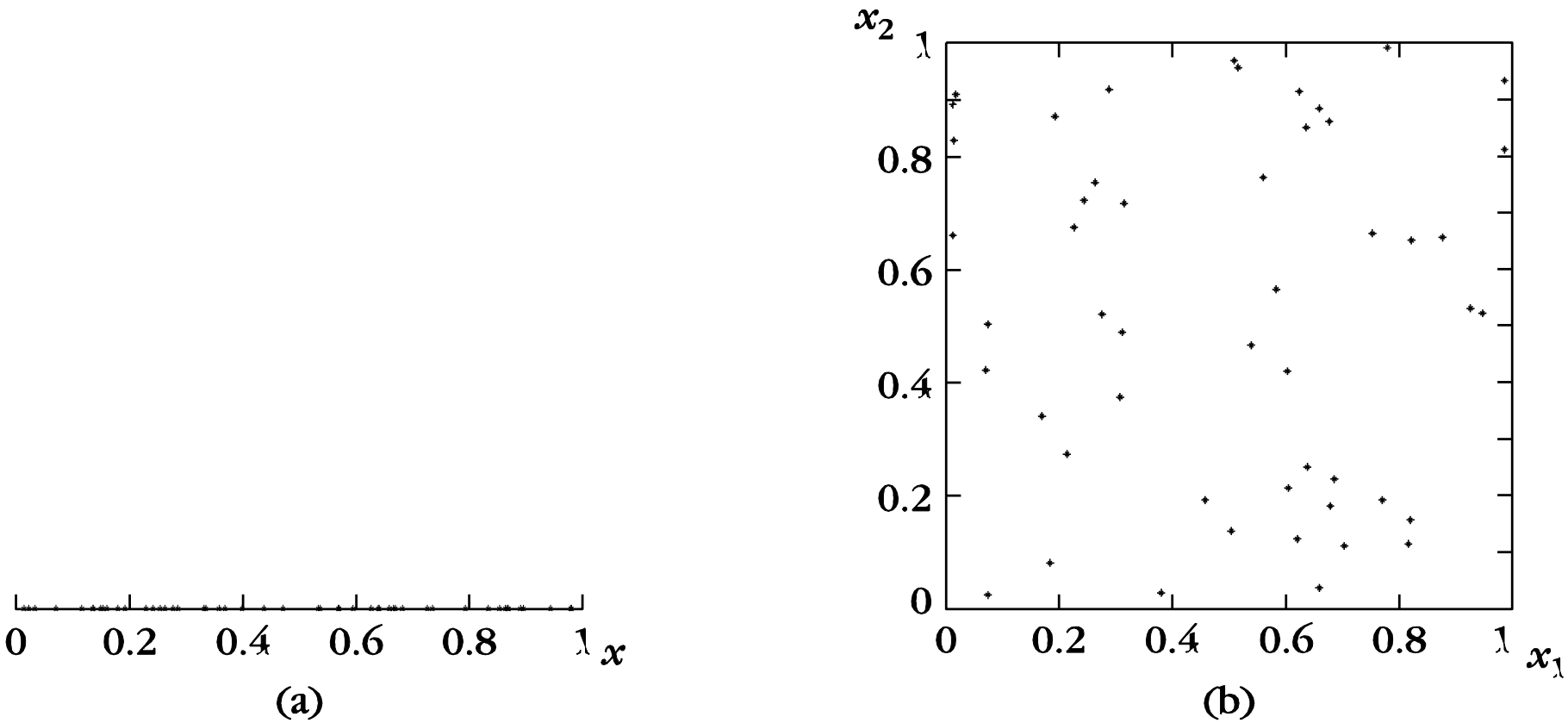
$$\boxed{\frac{\frac{1}{N_1 h^l} \sum_{i=1}^{N_1} \phi\left(\frac{\underline{x}_i - \underline{x}}{h}\right)}{\frac{1}{N_2 h^l} \sum_{i=1}^{N_2} \phi\left(\frac{\underline{x}_i - \underline{x}}{h}\right)} > (<) \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}}$$

❖ Remarks: The Curse Of Dimensionality

- In all the methods, so far, we saw that the highest the number of points, N, the better the resulting estimate.

- If in the one-dimensional space an interval, filled with N points, is adequately (for good estimation), in the two-dimensional space the corresponding square will require N^2 and in the ℓ-dimensional space the ℓ-dimensional cube will require N^ℓ points.

- The exponential increase in the number of necessary points in known as the curse of dimensionality. This is a major problem one is confronted with in high dimensional spaces.



Fifty points generated by a uniform distribution lying in the (a) one-dimensional unit-length segment and (b) the unit-length square. In the two-dimensional space the points are more spread compared to the same number of points in the one-dimensional space.

❖ K Nearest Neighbor Density Estimation

- In Parzen:
  - The volume is constant
  - The number of points in the volume is varying
- Now:
  - Keep the number of points $k_N = k$ constant
  - Leave the volume to be varying

$$\hat{p}(\underline{x}) = \frac{k}{NV(\underline{x})}$$

  - Again it can be shown [Fuku 90] that asymptotically

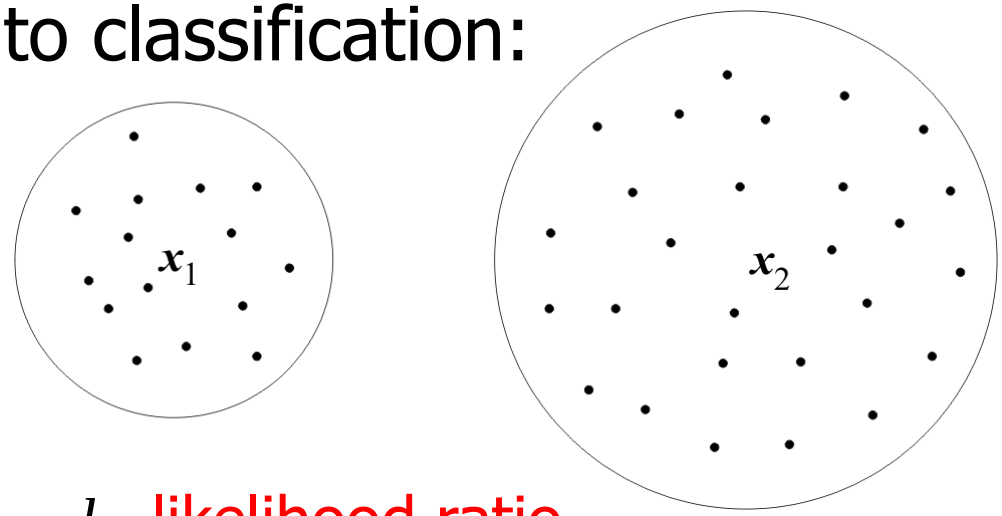
$(\lim k \to +\infty, \lim N \to +\infty, \lim (k/N) \to 0)$ this is an unbiased and consistent estimate of the true pdf, and it is known as the k Nearest Neighbor (k NN) density estimate.

Application to classification:

The volume of a hyperellipsoid $V = V_0 |\Sigma|^{\frac{1}{2}} r^l$

V_0 the volume of the hypersphere of unit radius

$$V_0 = \begin{cases} \pi^{\frac{l}{2}}/(l/2)!, & l \text{ even} \\ 2^l \pi^{\frac{l-1}{2}} (\frac{l-1}{2})!/l!, & l \text{ odd} \end{cases}$$

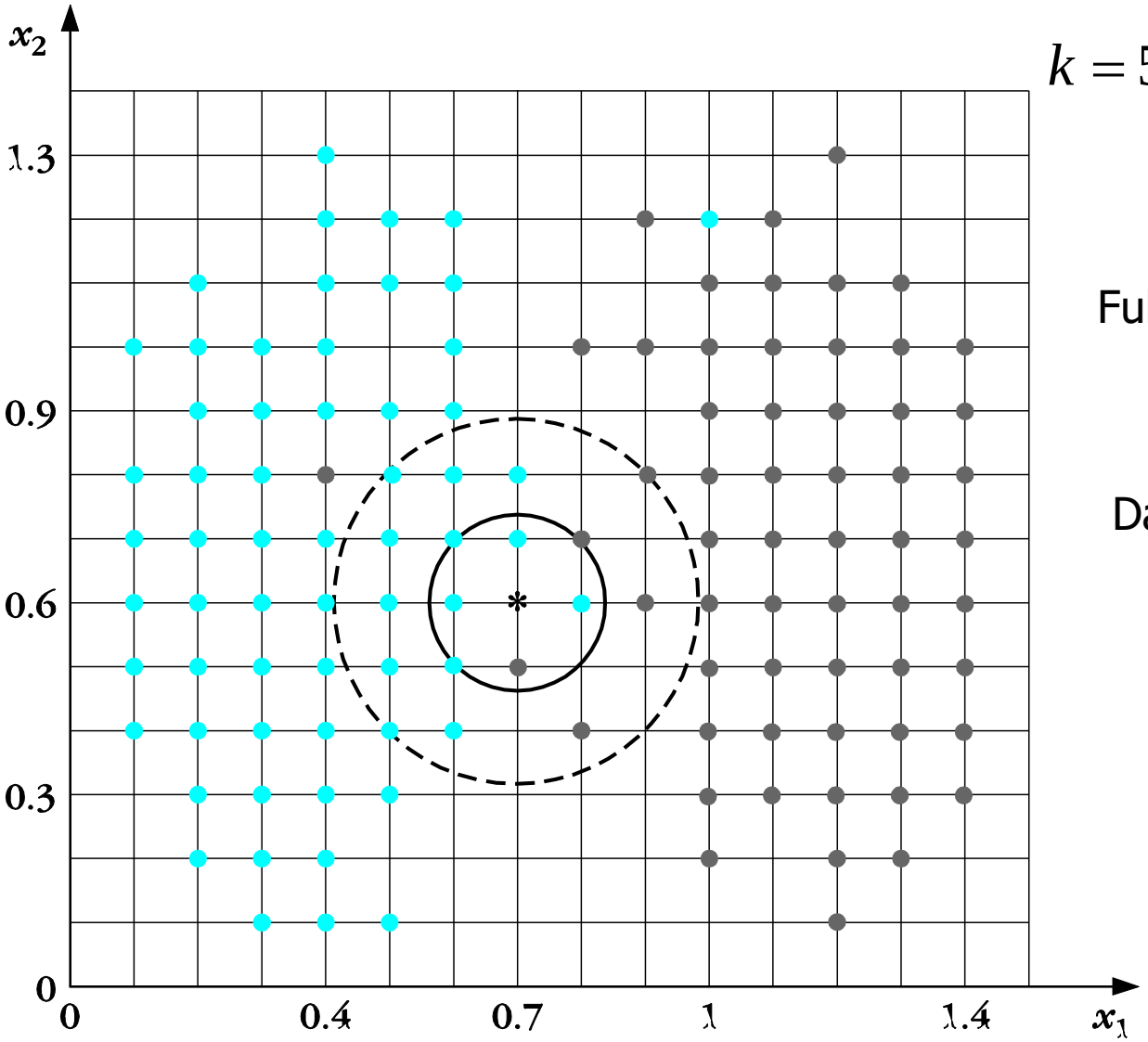


l_{12} likelihood ratio

$$\frac{\frac{k}{N_1 V_1}}{\frac{k}{N_2 V_2}} = \boxed{\frac{N_2 V_2}{N_1 V_1} > (<) \theta} \Rightarrow \frac{\frac{k}{N_1 V_1}}{\frac{k}{N_2 V_2}} = \frac{N_2 V_2}{N_1 V_1} > (<) \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}$$

assign $\underline{x}$ to $\omega_1 (\omega_2)$ if $l_{12} = \frac{N_2 V_2}{N_1 V_1} > (<) \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}$

$$\Rightarrow \frac{V_2}{V_1} > (<) \frac{N_1}{N_2} \frac{P(\omega_2)}{P(\omega_1)} \frac{\lambda_{21} - \lambda_{22}}{\lambda_{12} - \lambda_{11}}$$



$k = 5$

Black ω_1

Blue ω_2

Full line circle

$$\rho = \sqrt{0.1^2 + 0.1^2} = 0.1\sqrt{2}$$

Dash line circle

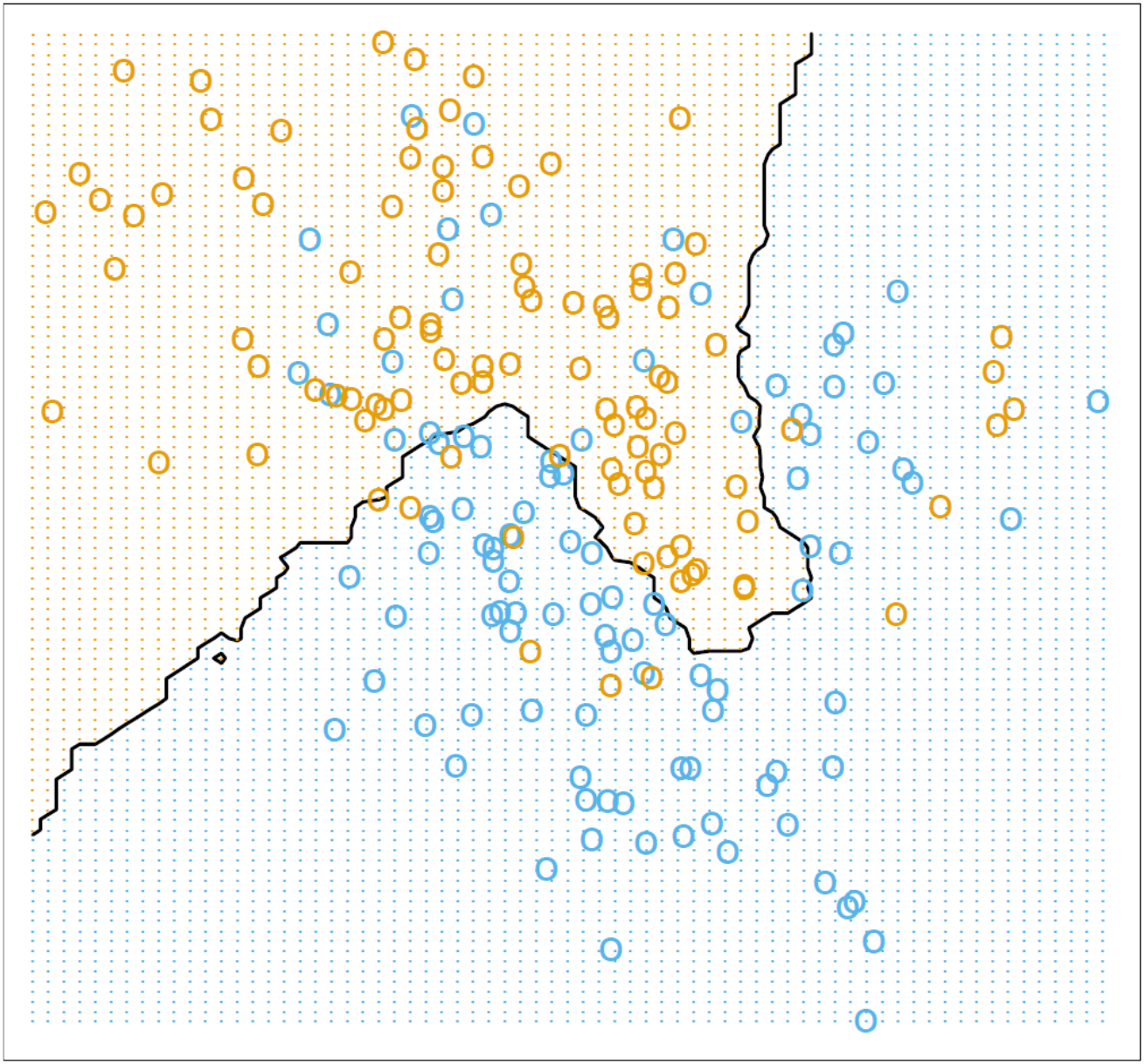
$$\sqrt{0.2^2 + 0.2^2} = 0.2\sqrt{2} = 2\rho$$

$N_1=59$ $N_2=61$

$$V_1 = 4\pi\rho^2, V_2 = \pi\rho^2,$$

$$\frac{V_2}{V_1} = \frac{\pi\rho^2}{4\pi\rho^2} = 0.25$$

✓ ω_2 ⇦ $0.25 < \frac{59}{61}$



15-nearest-neighbor

❖ NAIVE – BAYES CLASSIFIER

- Let $\underline{x} \in \Re^{\ell}$ and the goal is to estimate $p(\underline{x} \mid \omega_i)$ $i = 1, 2, \ldots, M$. For a "good" estimate of the pdf one would need, say, N^{ℓ} points.

- Assume $x_1, x_2, \ldots, x_{\ell}$ mutually independent. Then:

$$p(\underline{x} \mid \omega_i) = \prod_{j=1}^{\ell} p(x_j \mid \omega_i)$$

- In this case, one would require, roughly, N points for each pdf. Thus, a number of points of the order $N \cdot \ell$ would suffice.

- It turns out that the Naïve – Bayes classifier works reasonably well even in cases that violate the independence assumption.

❖ The Nearest Neighbor Rule

- Out of the N training vectors, identify the k nearest ones to $\underline{x}$ regardless of class label. k not to be a multiple of the number of classes M.

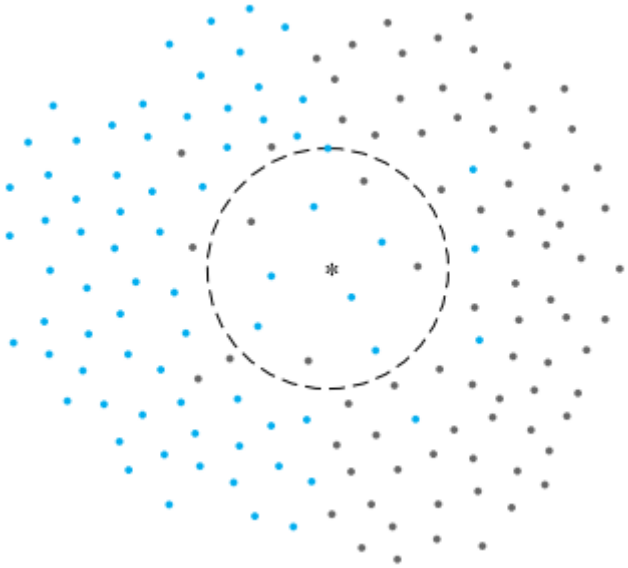
- Out of these k identify k_i that belong to class ω_i

$$\text{Assign } \underline{x} \rightarrow \omega_i : \; k_i > k_j \quad \forall i \neq j$$

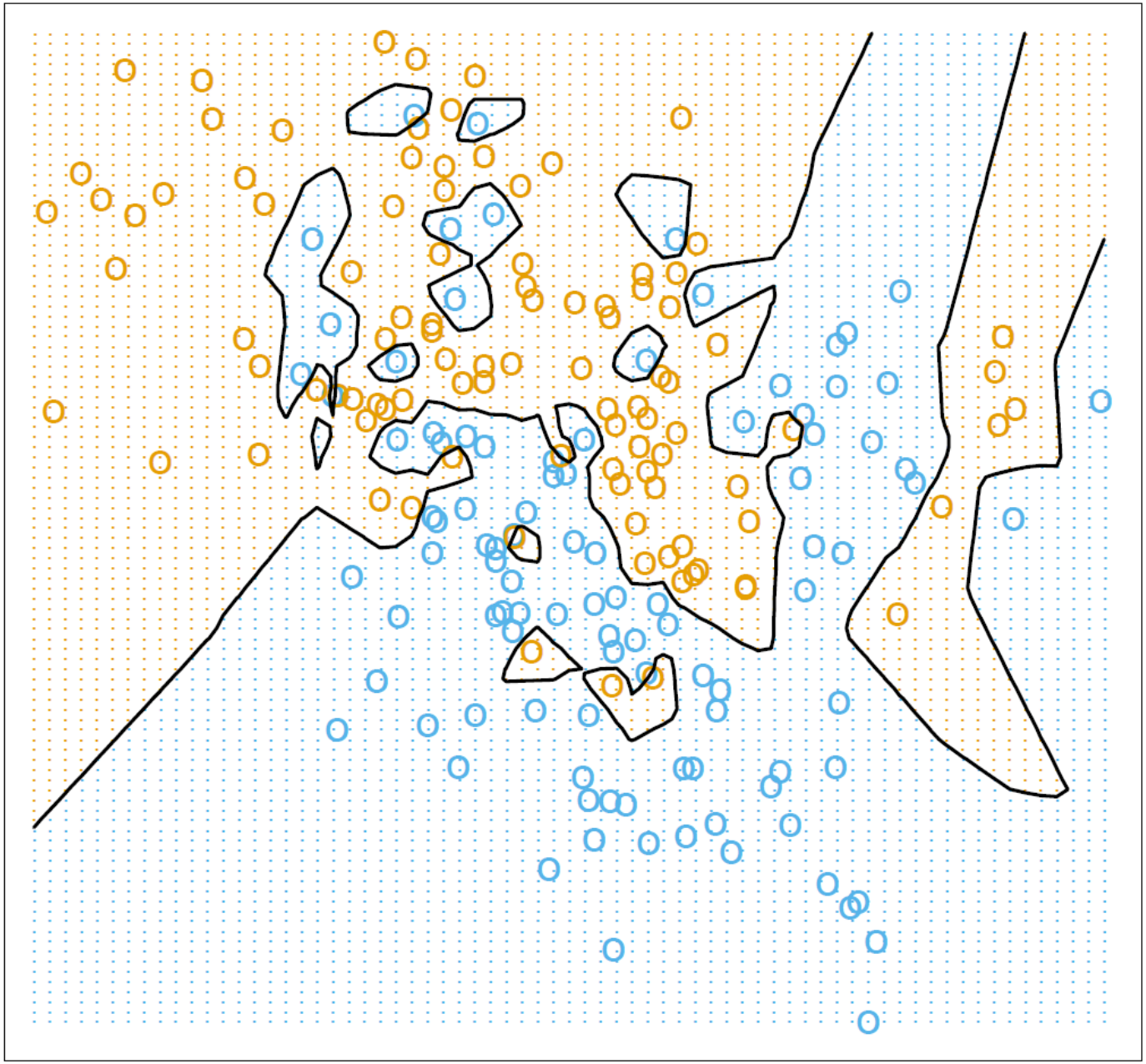
-

- The simplest version $\boxed{k=1 \text{ !!!}}$

An example:



Using the 11-NN rule



1-nearest-neighbor classication.

- For large N this is not bad. It can be shown that: if P_B is the optimal Bayesian error probability, then:

$$P_B \le P_{NN} \le P_B\left(2 - \frac{M}{M-1} P_B\right) \le 2P_B$$

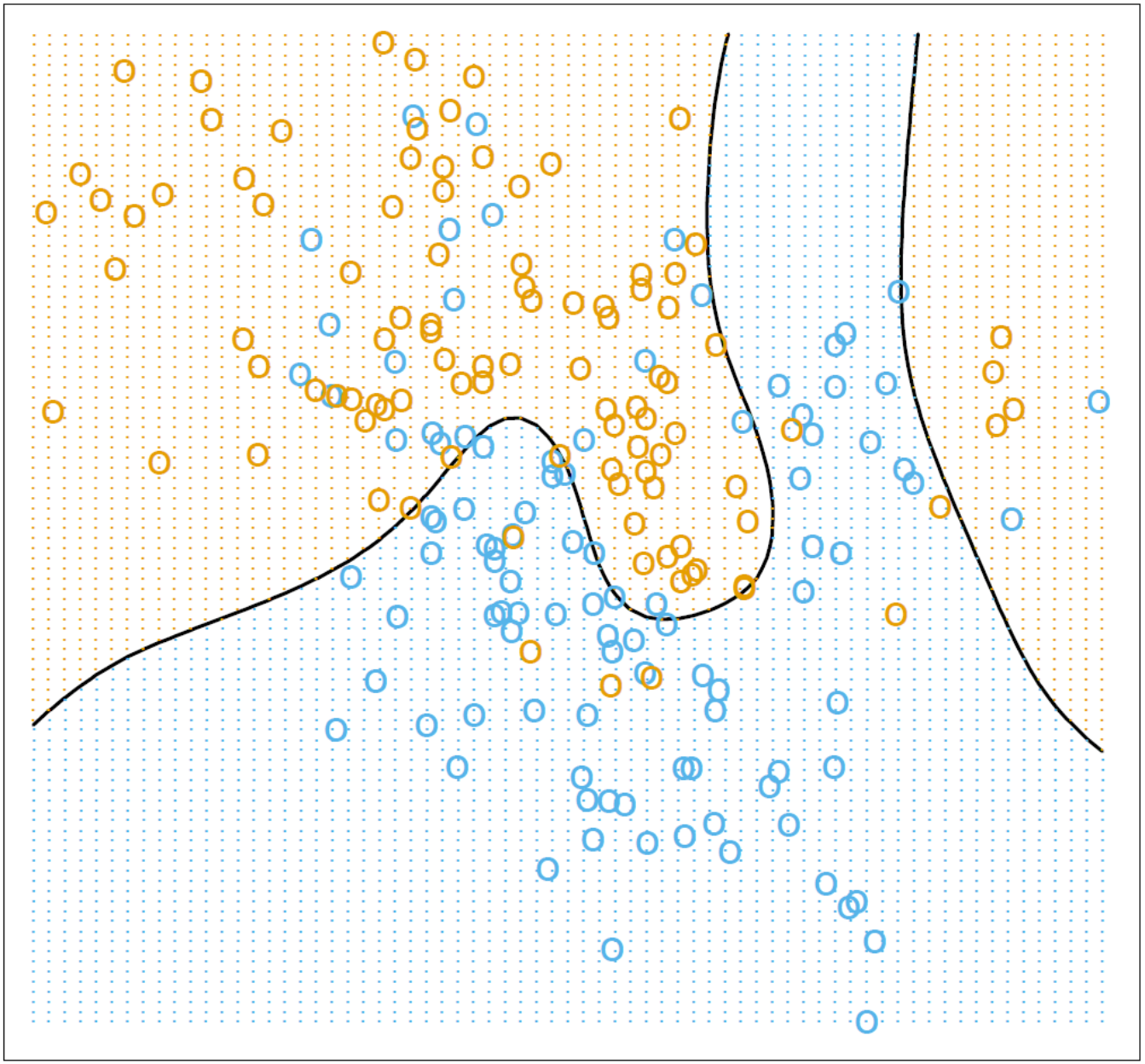
- $$P_B \le P_{kNN} \le P_B + \sqrt{\frac{2P_{NN}}{k}}$$

- $$\boxed{k \to \infty,\ P_{kNN} \to P_B}$$

- For small P_B:

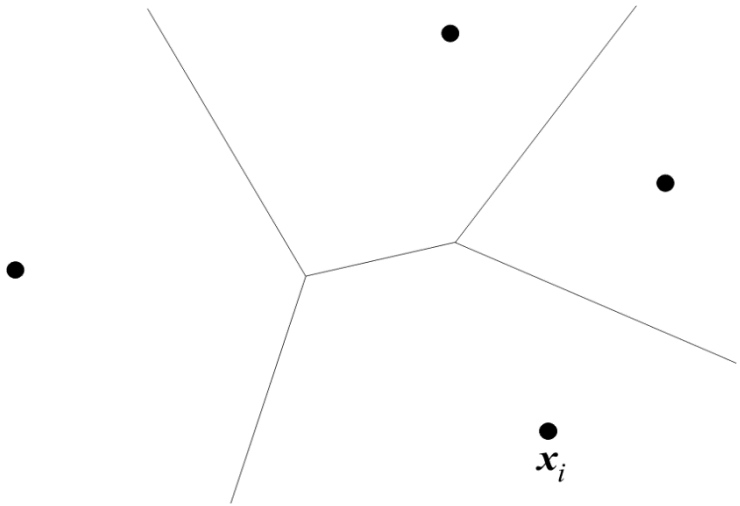
$$P_{NN} \cong 2P_B$$

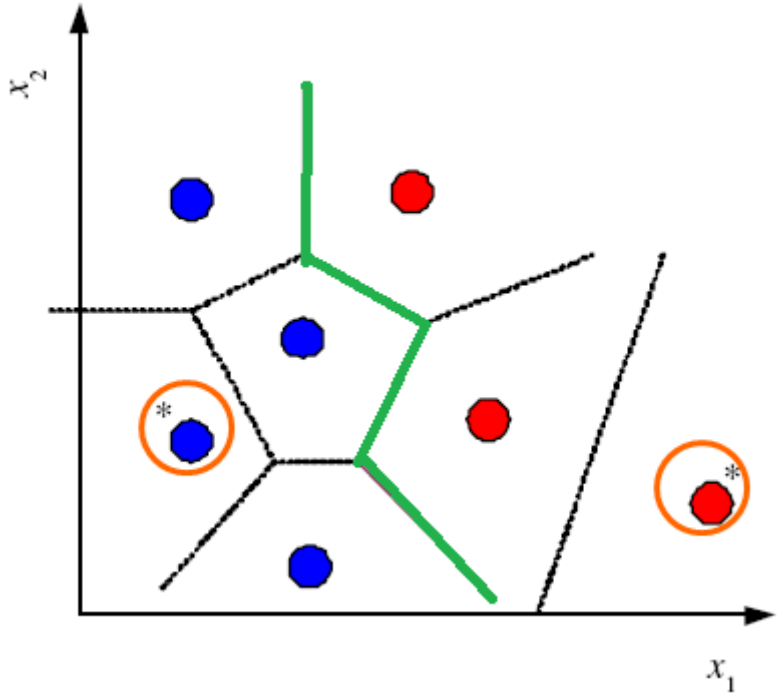
$$P_{3NN} \cong P_B + 3(P_B)^2$$



The optimal Bayes decision boundary

❖ Voronoi tesselation





$$R_i = \underline{x} : d(\underline{x}, \underline{x}_i) < d(\underline{x}, \underline{x}_j) \; i \neq j$$

BAYESIAN NETWORKS

- Bayes Probability Chain Rule

$$p(x_1, x_2, \ldots, x_\ell) = p(x_\ell \mid x_{\ell-1}, \ldots, x_1) \cdot p(x_{\ell-1} \mid x_{\ell-2}, \ldots, x_1) \cdot \ldots$$
$$\ldots \cdot p(x_2 \mid x_1) \cdot p(x_1)$$

  - Assume now that the conditional dependence for each x_i is limited to a subset of the features appearing in each of the product terms. That is:

$$p(x_1, x_2, \ldots, x_\ell) = p(x_1) \cdot \prod_{i=2}^{\ell} p(x_i \mid A_i)$$

where

$$A_i \subseteq \{x_{i-1}, x_{i-2}, \ldots, x_1\}$$

- For example, if ℓ=6, then we could assume:

$$p(x_6 \mid x_5,\ldots,x_1) = p(x_6 \mid x_5, x_4)$$

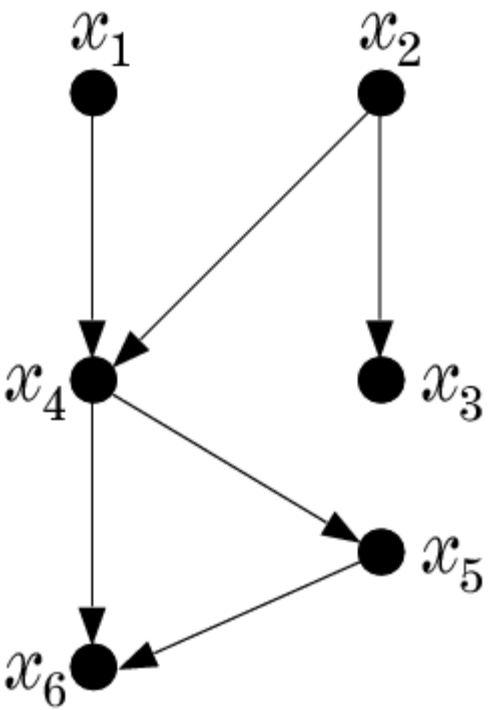
Then:

$$A_6 = \{x_5, x_4\} \subseteq \{x_5,\ldots,x_1\}$$

- The above is a generalization of the Naïve – Bayes. For the Naïve – Bayes the assumption is:

$$A_i = \emptyset, \text{ for } i = 1, 2, \ldots, l$$

- A graphical way to portray conditional dependencies is given below



- According to this figure we have that:
  - x_6 is conditionally dependent on x_4, x_5.
  - x_5 on x_4
  - x_4 on x_1, x_2
  - x_3 on x_2
  - x_1, x_2 are conditionally independent on other variables.

- For this case:

$$p(x_1, x_2, \ldots, x_6) =$$

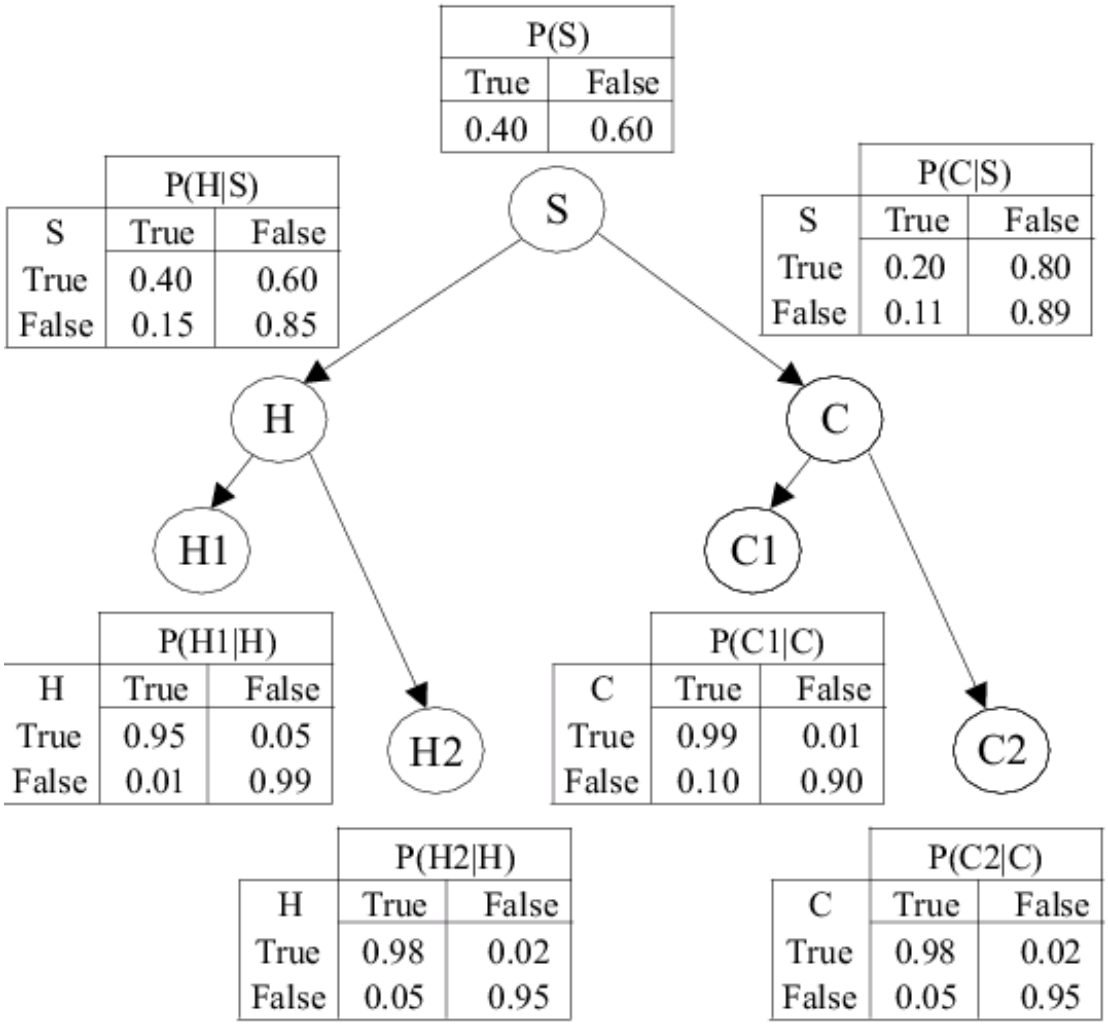
$$p(x_6 \mid x_5, x_4) \cdot p(x_5 \mid x_4) \cdot p(x_4 \mid x_2, x_1) \cdot p(x_3 \mid x_2) \cdot p(x_2) \cdot p(x_1)$$

❖ Bayesian Networks

- **Definition:** A Bayesian Network is a directed acyclic graph (DAG) where the nodes correspond to random variables. Each node is associated with a set of conditional probabilities (densities), $p(x_i|A_i)$, where x_i is the variable associated with the node and A_i is the set of its parents in the graph.

- A Bayesian Network is specified by:
  - The marginal probabilities of its root nodes.
  - The conditional probabilities of the non-root nodes, given their parents, for ALL possible combinations.

- The figure below is an example of a Bayesian Network corresponding to a paradigm from the medical applications field.



P(S)	
True	False
0.40	0.60

	P(H|S)	
S	True	False
True	0.40	0.60
False	0.15	0.85

	P(C|S)	
S	True	False
True	0.20	0.80
False	0.11	0.89

	P(H1|H)	
H	True	False
True	0.95	0.05
False	0.01	0.99

	P(C1|C)	
C	True	False
True	0.99	0.01
False	0.10	0.90

	P(H2|H)	
H	True	False
True	0.98	0.02
False	0.05	0.95

	P(C2|C)	
C	True	False
True	0.98	0.02
False	0.05	0.95

- This Bayesian network models conditional dependencies for an example concerning smokers (S), tendencies to develop cancer (C) and heart disease (H), together with variables corresponding to heart (H1, H2) and cancer (C1, C2) medical tests.

- BNs facilitate the description of a collection of beliefs by making explicit causality relations and conditional independence among beliefs
- BNs provide a more efficient way (than by using joint distribution tables) to update belief strengths when new evidence is observed
- Other names: Belief networks, Probabilistic networks, Causal networks.
- Causal networks can be used to follow how a change of certainty in one variable may change certainty of other variables.

- Once a DAG has been constructed, the joint probability can be obtained by multiplying the marginal* (root nodes) and the conditional (non-root nodes) probabilities.

- Training: Once a topology is given, probabilities are estimated via the training data set. There are also methods that learn the topology.

- Probability Inference: This is the most common task that Bayesian networks help us to solve efficiently. Given the values of some of the variables in the graph, known as evidence, the goal is to compute the conditional probabilities for some of the other variables, given the evidence.

* In the study of several random variables, the statistics of each are called marginal.

❖ Example: Consider the Bayesian network of figure 2.29:

To be provided

$P(x1)=0.60$	$P(y1\|x1)=0.40$	$P(z1\|y1)=0.25$	$P(w1\|z1)=0.45$
	$P(y1\|x0)=0.30$	$P(z1\|y0)=0.60$	$P(w1\|z0)=0.30$

$$X \rightarrow Y \rightarrow Z \rightarrow W$$

Can be derived

$P(x0)=0.40$	$P(y0\|x1)=0.60$	$P(z0\|y1)=0.75$	$P(w0\|z1)=0.55$
	$P(y0\|x0)=0.70$	$P(z0\|y0)=0.40$	$P(w0\|z0)=0.70$
	$P(y1)=0.36$	$P(z1)=0.47$	$P(w1)=0.37$
	$P(y0)=0.64$	$P(z0)=0.53$	$P(w0)=0.63$

Figure 2.29

a) If x is measured to be $x=1$ ($x1$), compute $P(w=0|x=1)$ [$P(w0|x1)$].

b) If w is measured to be $w=1$ ($w1$) compute $P(x=0|w=1)$ [$P(x0|w1)$].

- For a), a set of calculations are required that propagate from node X to node W. It turns out that $P(w0|x1) = 0.63$.

$$P(z1|x1) = P(z1|y1, x1)P(y1|x1) + P(z1|y0, x1)P(y0|x1)$$
$$= P(z1|y1)P(y1|x1) + P(z1|y0)P(y0|x1)$$
$$=(0.25)(0.4) + (0.6)(0.6) = 0.46$$

In a similar way, $P(z0|x1) = 1 - P(z1|x1) = 0.54$

$$P(w0|x1) = P(w0|z1, x1)P(z1|x1) + P(w0|z0, x1)P(z0|x1)$$

$$= P(w0|z1)P(z1|x1) + P(w0|z0)P(z0|x1)$$

$$= (0.55)(0.46) + (0.7)(0.54) = 0.63$$

- For b), the propagation is reversed in direction. It turns out that $P(x0|w1) = 0.4$.

$$P(z1|w1) = \frac{P(w1|z1)P(z1)}{P(w1)} = \frac{(0.45)(0.47)}{0.37} = 0.57$$

$$P(y1|w1) = \frac{P(w1|y1)P(y1)}{P(w1)}$$

$$P(w1|y1) = P(w1|z1, y1)P(z1|y1) + P(w1|z0, y1)P(z0|y1)$$

$$= P(w1|z1)P(z1|y1) + P(w1|z0)P(z0|y1)$$

$$= (0.45)(0.25) + (0.3)(0.75) = 0.34$$

In a similar way,

$$P(w1|y0) = 0.39$$

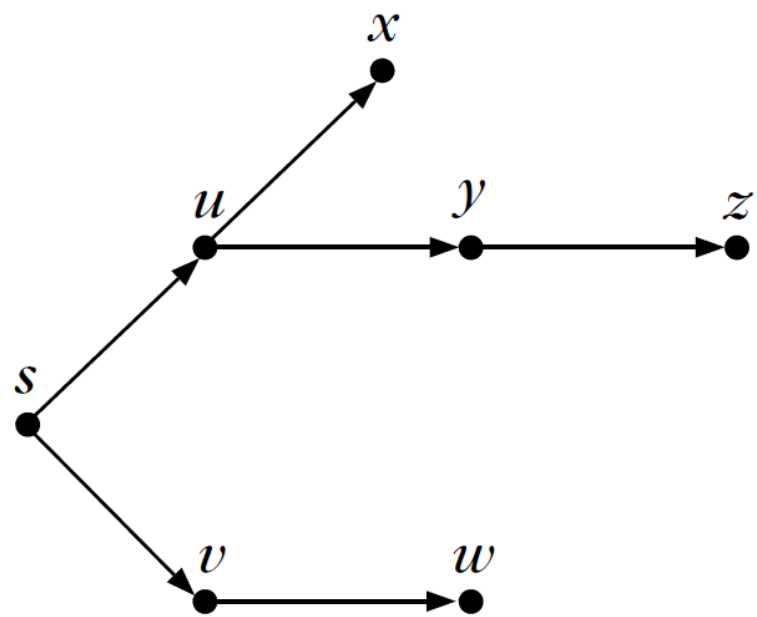
It is left as an exercise to show that $P(x0|w1) = 0.4$.

- In general, the required inference information is computed via a combined process of "message passing" among the nodes of the DAG.

- Complexity:
  - For singly connected graphs, message passing algorithms amount to a complexity linear in the number of nodes.

Bayesian networks with tree structure

- Compute the conditional probability $P(s|\mathbf{z} = z_0)$, where $\mathbf{z} = z_0$ is the evidence.



$$P(s|z = z_0) = \frac{P(s, z = z_0)}{P(z = z_0)} = \frac{P(s, z = z_0)}{\sum_s P(s, z = z_0)}$$

- Marginalization …

$$P(s, z = z_0) = \sum_{u,v,x,y,w} P(s, u, v, x, y, w, z = z_0)$$

If the discrete variables can take L values, the complexity of the previous computations amounts to L^5 operations.

We exploit the structure of the Bayesian network in order to reduce the computational burden.

$$\sum_{u,v,x,y,w} P(s,u,v,x,y,w,z=z_0) =$$

$$\sum_{u,v,x,y,w} P(s)P(u|s)P(v|s)P(w|v)P(x|u)P(y|u)P(z=z_0|y) =$$

$$P(s)\underbrace{\sum_{u,v} P(u|s)P(v|s)\underbrace{\sum_{w} P(w|v)}_{v}\underbrace{\sum_{x} P(x|u)}_{u}\underbrace{\sum_{y} P(y|u)P(z=z_0|y)}_{u}}_{s}$$

$$\sum_{u,v,x,y,w} P(s,u,v,x,y,w,z=z_0) = P(s)\sum_{u,v} P(u|s)P(v|s)\phi_1(v)\phi_2(u)\phi_3(u)$$

The order of L^2 , instead of the order of L^5 demanded for the brute-force computation