

Chapter 2 (Part 1): Bayesian Decision Theory (Sections 2.1-2.2)



- Introduction
- Bayesian Decision Theory—Continuous Features

Introduction

- The sea bass/salmon example

- State of nature, prior

- State of nature is a random variable
 - The catch of salmon and sea bass is equiprobable

- $P(\omega_1) = P(\omega_2)$ (uniform priors)
- $P(\omega_1) + P(\omega_2) = 1$ (exclusivity and exhaustivity)



- Decision rule with only the prior information
 - Decide ω_1 if $P(\omega_1) > P(\omega_2)$
 - otherwise decide ω_2



- Use More Information: the class – conditional information
- $p(x | \omega_1)$ and $p(x | \omega_2)$ describe the difference in lightness between populations of sea and salmon



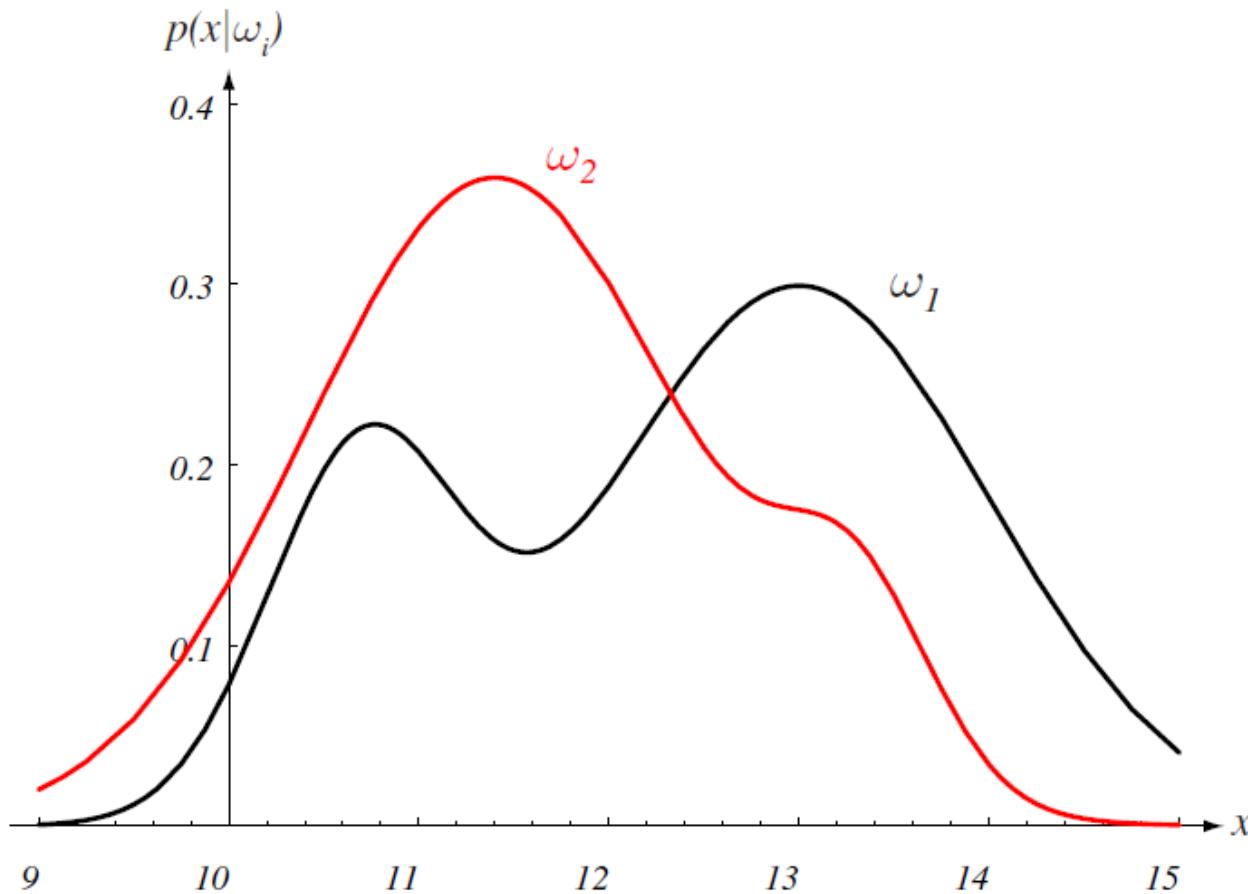


FIGURE 2.1. Hypothetical class-conditional probability density functions show the probability density of measuring a particular feature value x given the pattern is in category ω_i . If x represents the lightness of a fish, the two curves might describe the difference in lightness of populations of two types of fish. Density functions are normalized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



- Posterior, likelihood, evidence

- $P(\omega_j | x) = p(x | \omega_j) \cdot P(\omega_j) / p(x)$

- Where in case of two categories

$$p(x) = \sum_{j=1}^{j=2} p(x | \omega_j)P(\omega_j)$$

- Posterior = (Likelihood. Prior) / Evidence
 - Evidence can be viewed as a scale factor

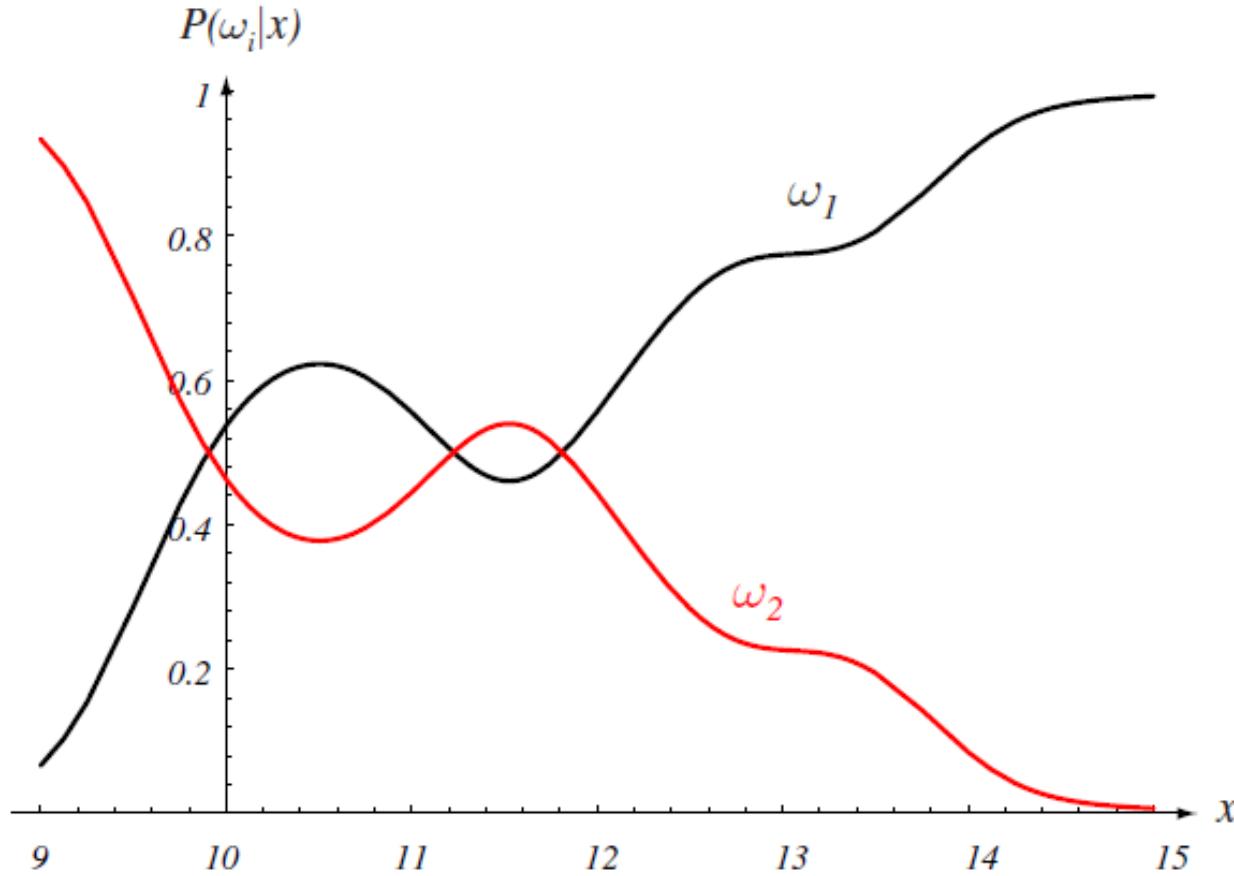
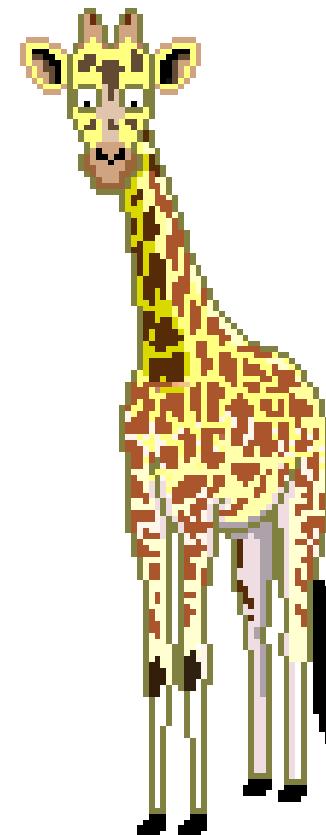
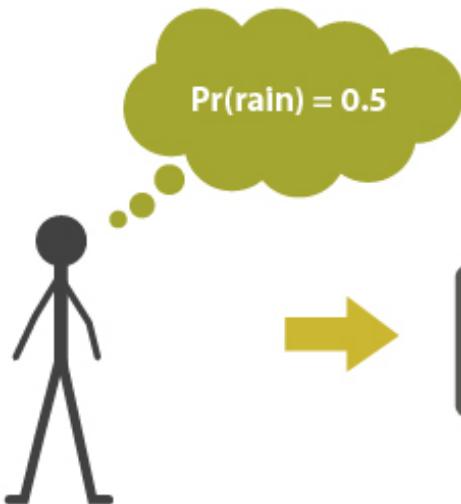


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x , the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Posterior is a modification of prior

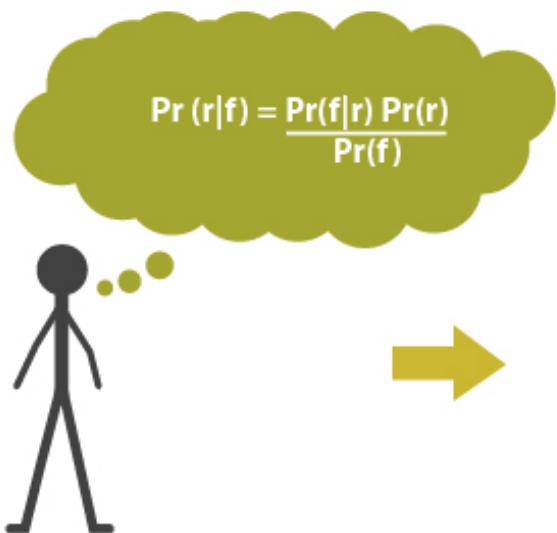
- The modification is caused by the likelihood

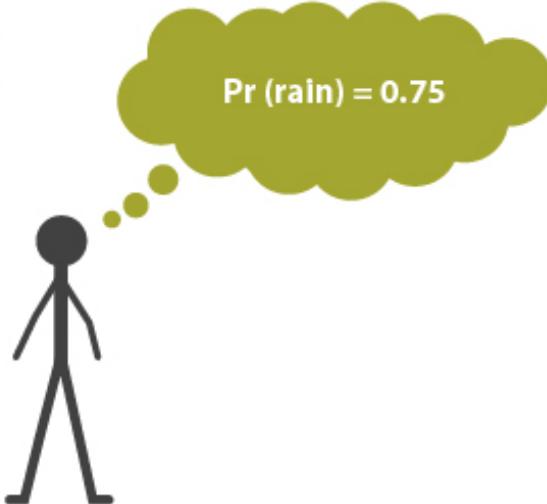




$\Pr(\text{rain}) = 0.5$

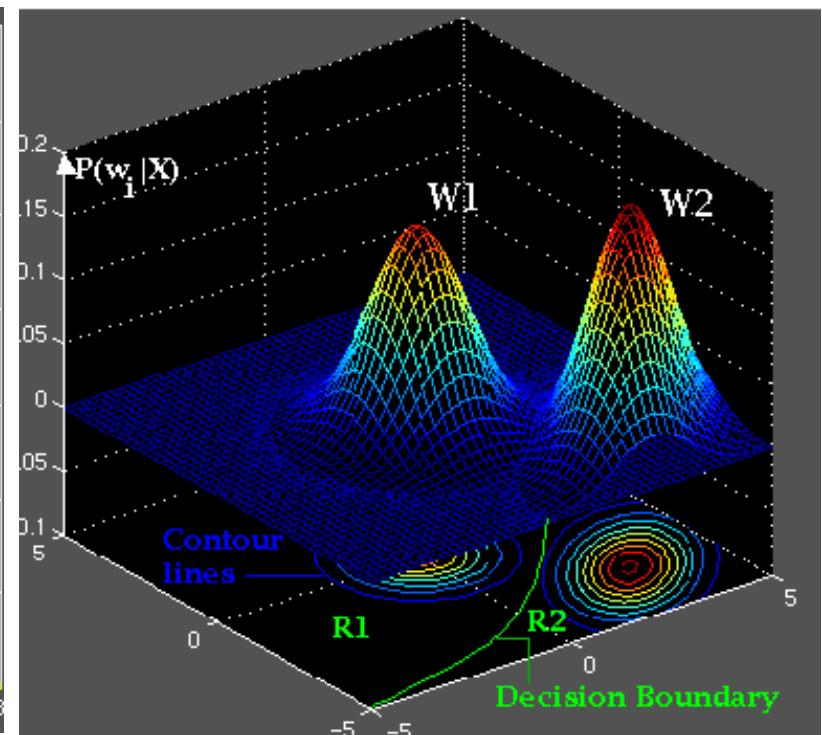
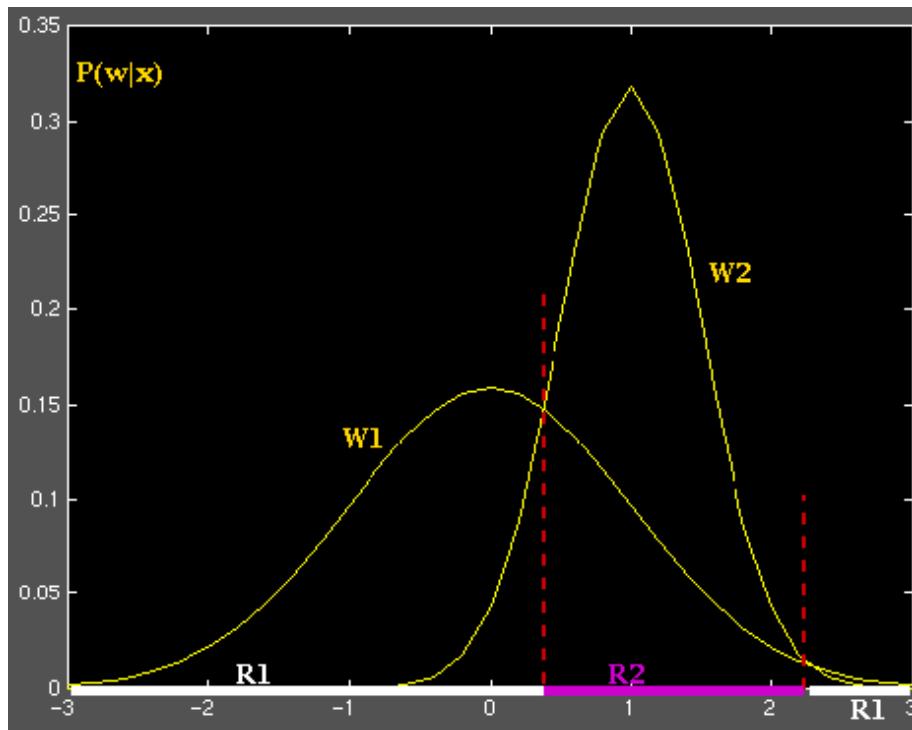



$$\Pr(r|f) = \frac{\Pr(f|r) \Pr(r)}{\Pr(f)}$$



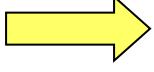
$\Pr(\text{rain}) = 0.75$

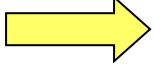
Decision region



- Decision given the posterior probabilities

X is an observation for which:

if $P(\omega_1 | x) > P(\omega_2 | x)$  True state of nature = ω_1

if $P(\omega_1 | x) < P(\omega_2 | x)$  True state of nature = ω_2

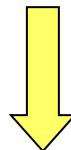
Multi-class ?

Therefore,

whenever we observe a particular x , the probability of error is :

$P(\text{error} | x) = P(\omega_1 | x) \text{ if we decide } \omega_2$

$P(\text{error} | x) = P(\omega_2 | x) \text{ if we decide } \omega_1$



$$P(\text{error} | x) = \min(P(\omega_1 | x), P(\omega_2 | x))$$

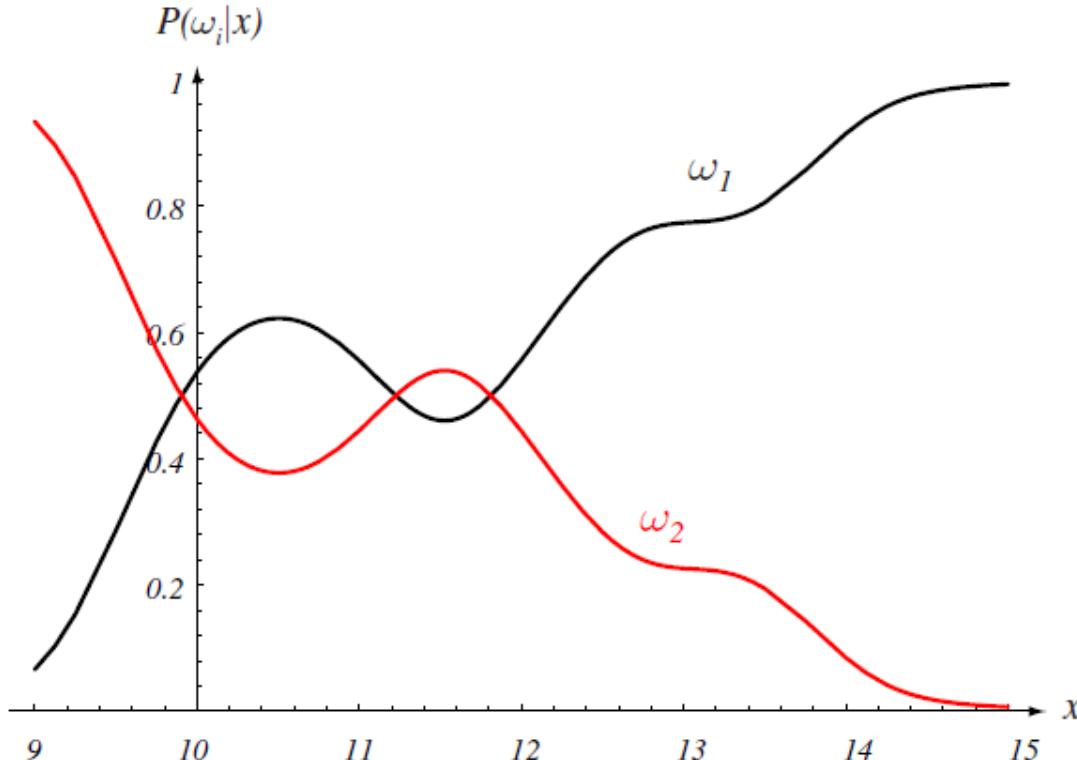


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x , the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Multi-class

if $P(\omega_j | x) > P(\omega_i | x)$

Then the true state of
nature = ω_j



- Minimizing the probability of error
- Decide ω_1 if $P(\omega_1 | x) > P(\omega_2 | x)$;
otherwise decide ω_2

Therefore:

$$P(\text{error} | x) = \min [P(\omega_1 | x), P(\omega_2 | x)]$$

(Bayes decision)

Decide ω_1 if $p(x | \omega_1) P(\omega_1) > p(x | \omega_2) P(\omega_2)$

- Special Case:

$$p(x | \omega_1) = p(x | \omega_2)$$

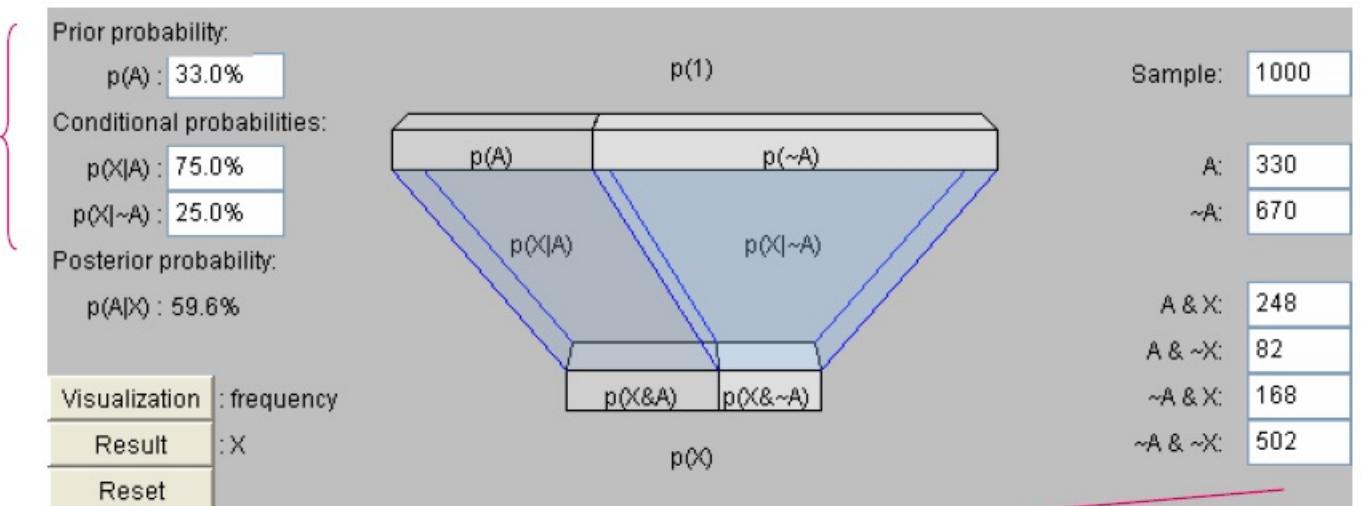
$$P(\omega_1) = P(\omega_2)$$

Interesting video

- <http://weike.enetedu.com/play.asp?vodid=148661>
- <http://weike.enetedu.com/play.asp?vodid=141126>

One example

Known



Data

By Conditional Probability Rule,

$$p(X/A) = \frac{p(X \& A)}{p(A)}$$

$$= \frac{.248}{.330} = 0.7515$$

$$p(X/\sim A) = \frac{p(X \& \sim A)}{p(\sim A)}$$

$$= \frac{.168}{.670} = 0.2507$$

By Bayes Rule, $P(A/X) = \frac{P(X/A)p(A)}{P(X)}$

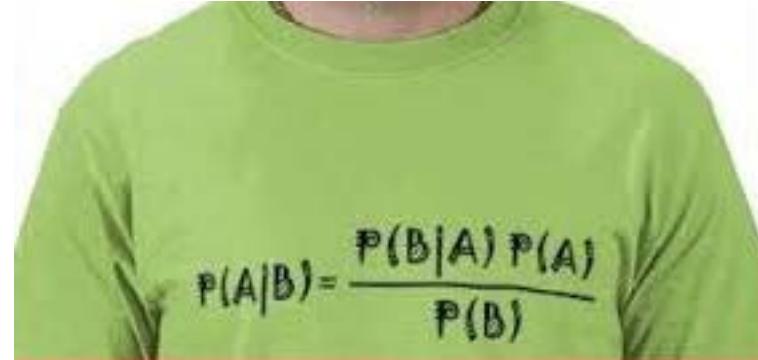
$$= \frac{P(X/A)p(A)}{P(X \& A) + P(X \& \sim A)}$$

$$= \frac{P(X/A)p(A)}{P(X/A)p(A) + P(X/\sim A)p(\sim A)}$$

$$= \frac{0.75 \times 0.33}{0.75 \times 0.33 + 0.25 \times 0.67}$$

$$= \frac{.2475}{.2475 + .1675} = \frac{.2475}{.415} = 0.596$$
2

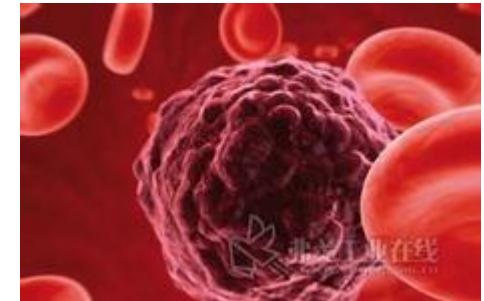
Exercise



- **Problem:**
- A patient takes a lab test and the result is positive. The test returns a correct positive result in 98% of the cases in which the cancer disease is actually present, and a correct negative result in 97% of the cases in which the disease is not present. Furthermore, 0.8% of the entire population have this cancer disease. Does the patient suffers from the cancer?

- **Solution:**

- Given: $P(+|\text{cancer})=0.98$
 $P(-|\text{no cancer})=0.97$



$$P(\text{cancer})=0.008$$

$$P(-\text{cancer})=0.992$$

- Compute:

$$P(\text{no cancer} | +), P(\text{cancer} | +),$$

我不想活了~



- $P(\text{cancer} | +) = P(+|\text{cancer}) * P(\text{cancer}) / p(+)$
- $P(\text{cancer} | +) = 0.98 \times 0.008 / p(+)$
- $P(\text{cancer} | +) = 0.00784 / p(+)$

$$P(\text{no cancer} | +) = P(+ | \text{no cancer}) * P(\text{no cancer}) / P(+)$$

$$P(\text{no cancer} | +) = (1 - 0.97) * (1 - 0.008) / P(+)$$

- $P(\text{no cancer} | +) = 0.02976 / P(+)$
- Since $P(\text{no cancer} | +) > P(\text{cancer} | +)$, we decide that the patient does not have cancer
- (Bayesian decision rule)



Exercise

- **Another Problem:**
- A person takes a lab test of nuclear radiation and the result is positive. The test returns a correct positive result in 99% of the cases in which the nuclear radiation is actually present, and a correct negative result in 95% of the cases in which the nuclear radiation is not present. Furthermore, 30% of the entire population are radioactively contaminated. Is this person contaminated ?

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayesian Decision Theory – Continuous Features

- Generalization of the preceding ideas
 - Use of more than one feature
 - Use more than two states of nature
 - Allowing actions and not only decide on the state of nature
 - Introduce a loss of function which is more general than the probability of error

Shortcoming of simple Bayesian decision

It have to let

$$X \rightarrow \omega_i$$



If the sample **does not belong to any class**, it will still be assigned to a class



- Allowing actions other than classification primarily allows the possibility of rejection
- Refusing to make a decision in close or bad cases!



- The loss function states how costly each action taken is



Examples of classification with rejection

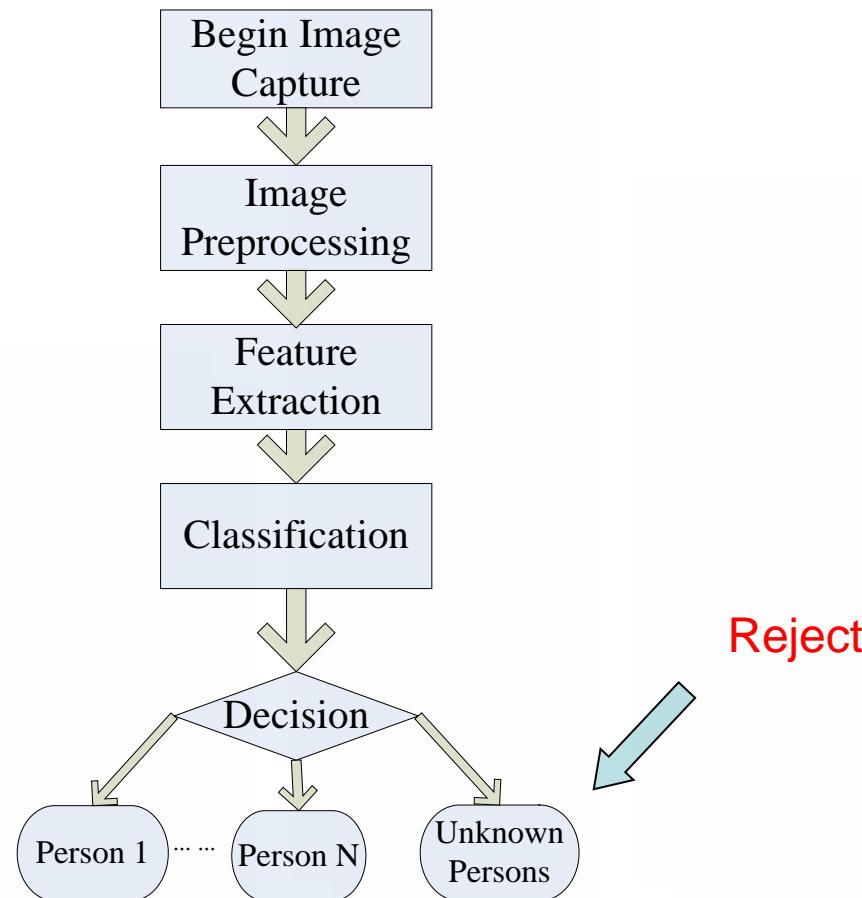


Consequence of no rejection: if a person (the user) **is not one of the registered users**, he will be also erroneously recognized as a registered user!

Consequently this user will be **erroneously allowed to pass the system!!**

Personal identification & rejection

Face recognition flowchart



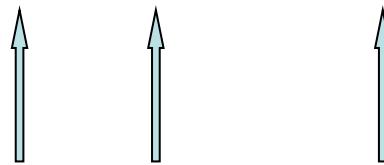
Let $\{\omega_1, \omega_2, \dots, \omega_c\}$ be the set of c states of nature
(or “categories”)

Let $\{\alpha_1, \alpha_2, \dots, \alpha_a\}$ be the set of possible actions

Let $\lambda(\alpha_i | \omega_j)$ be the loss incurred for taking action α_i when the state of nature is ω_j

A simple case

- $\omega_1, \omega_2, \dots, \omega_c$: C classes



- $\alpha_1, \alpha_2, \dots, \alpha_c$: C actions



α_{c+1} : do not assign the sample into any class--- *reject*



Overall risk

$R = \text{Sum of all } R(\alpha_i | x) \text{ for } i = 1, \dots, a$

$\underbrace{}$

Conditional risk

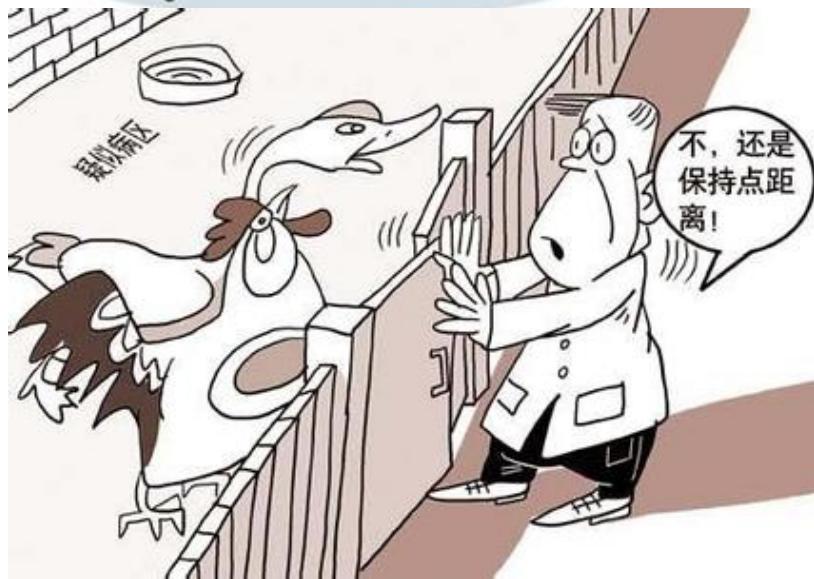
Minimizing R \longleftrightarrow Minimizing $R(\alpha_i | x)$ for $i = 1, \dots, a$

$$R(\alpha_i | x) = \sum_{j=1}^{j=c} \lambda(\alpha_i | \omega_j) P(\omega_j | x)$$

for $i = 1, \dots, a$



Fail to declare and error declaration



Select the action α_i for which $R(\alpha_i | x)$ is minimum

→ R is minimum and R in this case is called the Bayes risk = best reasonable result that can be achieved!

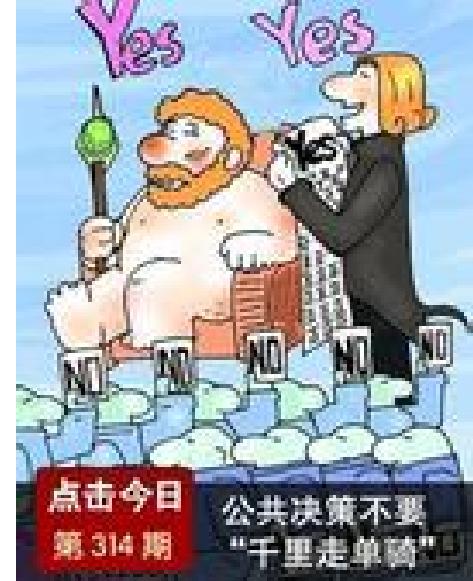


- Two-category classification

α_1 : deciding ω_1

α_2 : deciding ω_2

$$\lambda_{ij} = \lambda(\alpha_i | \omega_j)$$



λ_{ij} : loss incurred for deciding ω_i when the true state of nature is ω_j

Conditional risk:

$$R(\alpha_1 | x) = \lambda_{11}P(\omega_1 | x) + \lambda_{12}P(\omega_2 | x)$$

$$R(\alpha_2 | x) = \lambda_{21}P(\omega_1 | x) + \lambda_{22}P(\omega_2 | x)$$

Our rule is the following:

$$\text{if } R(\alpha_1 | x) < R(\alpha_2 | x)$$

action α_1 : “decide ω_1 ” is taken

This results in the equivalent rule :

decide ω_1 if:

$$(\lambda_{21} - \lambda_{11}) P(\omega_1 | x) > (\lambda_{12} - \lambda_{22}) P(\omega_2 | x)$$

- and decide ω_2 otherwise



$$(\lambda_{21} - \lambda_{11}) P(\omega_1 | x) > (\lambda_{12} - \lambda_{22}) P(\omega_2 | x)$$

- is equal to

$$(\lambda_{21} - \lambda_{11}) P(x|\omega_1) P(\omega_1) > (\lambda_{12} - \lambda_{22}) P(x|\omega_2) P(\omega_2)$$

Likelihood ratio:

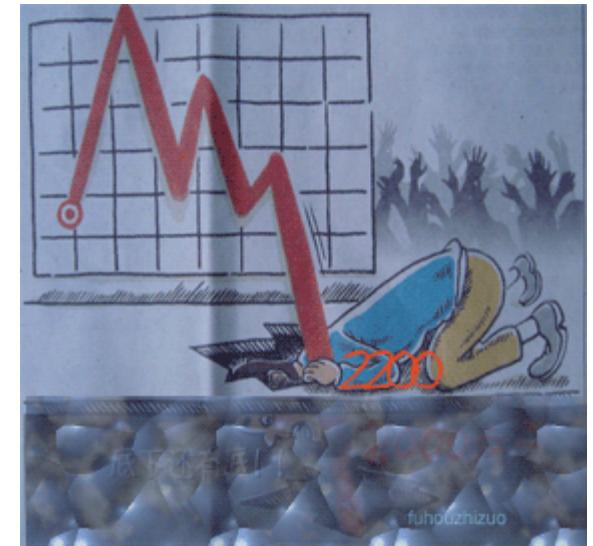
The preceding rule is equivalent to the following rule

$$if \frac{p(x | \omega_1)}{p(x | \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)}$$

Then take action α_1 (decide ω_1). Otherwise take action α_2 (decide ω_2)

Optimal decision property

“If the likelihood ratio exceeds a threshold value independent of the input pattern x , we can take optimal actions”





Bayesian Decision Theory

Loss Function

- $\lambda(\alpha_i | \omega_j)$: cost incurred for taking action α_i (i.e., classification or rejection) when the state of nature is ω_j
- Example

- x : financial characteristics of firms applying for a bank loan
- ω_0 – company did not go bankrupt
- ω_1 – company failed
- $P(\omega_1|x)$ – predicted probability of bankruptcy
- Confusion matrix:

	Algorithm: ω_0	Algorithm: ω_1
Truth: ω_0	TN	FP
Truth: ω_1	FN	TP

- FN are 10 times as costly as FP
- $$\Rightarrow \lambda(\alpha_0 | \omega_1) = \lambda_{01} = 10 \times \lambda(\alpha_1 | \omega_0) = 10 \times \lambda_{10}$$



- Simplest λ_{ij}
- $\lambda_{ij}=1, i \neq j$
- $\lambda_{ij}=0$
- Then minimum risk Bayesian decision **will be equivalent to** Minimum error Bayesian decision



Exercise

Select the optimal decision where:

$$= \{\omega_1, \omega_2\}$$

$$\begin{array}{lll} p(+ | \omega_1) & \xrightarrow{\text{yellow arrow}} & 0.9 \\ p(+ | \omega_2) & \xrightarrow{\text{yellow arrow}} & 0.001 \end{array}$$

$$\begin{aligned} P(\omega_1) &= 0.01 \\ P(\omega_2) &= 0.99 \end{aligned}$$

$$\lambda = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$$

$$\lambda = \begin{bmatrix} 1 & 4 \\ 2 & 3 \end{bmatrix}$$

means $\lambda_{11}=1$, $\lambda_{12}=4$, $\lambda_{21}=2$, $\lambda_{22}=3$



Example: earthquake forecast; typhoon forecast

Example

- 例：已知正常细胞先验概率为 $P(\omega_1) = 0.9$, 异常为 $P(\omega_2) = 0.1$,
从类条件概率密度分布曲线上查的 $P(x/\omega_1) = 0.2, P(x/\omega_2) = 0.4$,
 $\lambda_{11} = 0, \lambda_{12} = 6, \lambda_{21} = 1, \lambda_{22} = 0$
由上例中计算出的后验概率： $P(\omega_1/x) = 0.818, P(\omega_2/x) = 0.182$

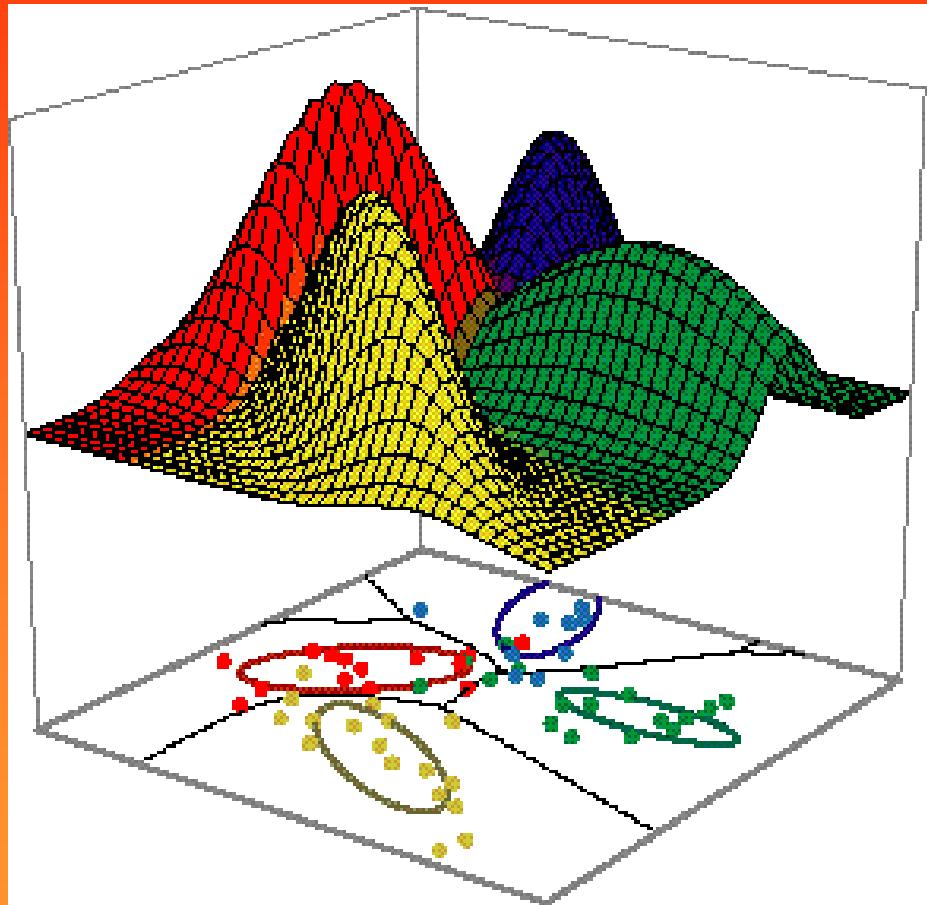
条件风险： $R(\alpha_1/x) = \sum_{j=1}^2 \lambda_{1j} P(\omega_j/x) = \lambda_{12} P(\omega_2/x) = 1.092$

$R(\alpha_2/x) = \lambda_{21} P(\omega_1/x) = 0.818$

因为 $R(\alpha_1/x) > R(\alpha_2/x) \therefore x \in$ 异常细胞, 因决策 ω_1 类风险大。
因 $\lambda_{12}=6$ 较大， 决策损失起决定作用。



Pattern Classification



All materials in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors and the publisher

Chapter 2

Bayesian Decision Theory (Sections 2.3-2.5)

- Minimum-Error-Rate Classification
- Classifiers, Discriminant Functions and Decision Surfaces
- The Normal Density

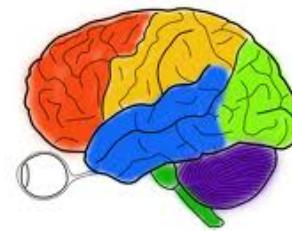
- Definitions of actions
- Action α_i : assign the test sample to the i-th class
- Seek a decision rule that minimizes the *probability of error* which is the *error rate*

2.3 Minimum-Error-Rate Classification

- Actions are decisions on classes
If action α_i is taken and the true state of nature is ω_j then:
the decision is correct if $i = j$ and in error if $i \neq j$



Decision Making



- The loss function for above case is the zero-one loss function:

$$\lambda(\alpha_i, \omega_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \quad i, j = 1, \dots, c$$

Therefore, the conditional risk is:

$$\begin{aligned} R(\alpha_i | x) &= \sum_{j=1}^{j=c} \lambda(\alpha_i | \omega_j) P(\omega_j | x) \\ &= \sum_{j \neq i} P(\omega_j | x) = 1 - P(\omega_i | x) \end{aligned}$$

Minimum-Error
Bayesian Decision !

"The risk corresponding to this loss function is the average probability error"

- Minimize the risk requires to maximize $P(\omega_i | x)$ (since $R(\alpha_i | x) = 1 - P(\omega_i | x)$)

- For Minimum error rate

- Decide ω_i if $P(\omega_i | x) > P(\omega_j | x) \ \forall j \neq i$

Minimum-Error and Maximum income in expectation may be not optimal

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



"我们这次推荐的策略比较冒险，要是您能在离开前就把钱付了我们会万分感谢的。"



How to minimize the risk ?



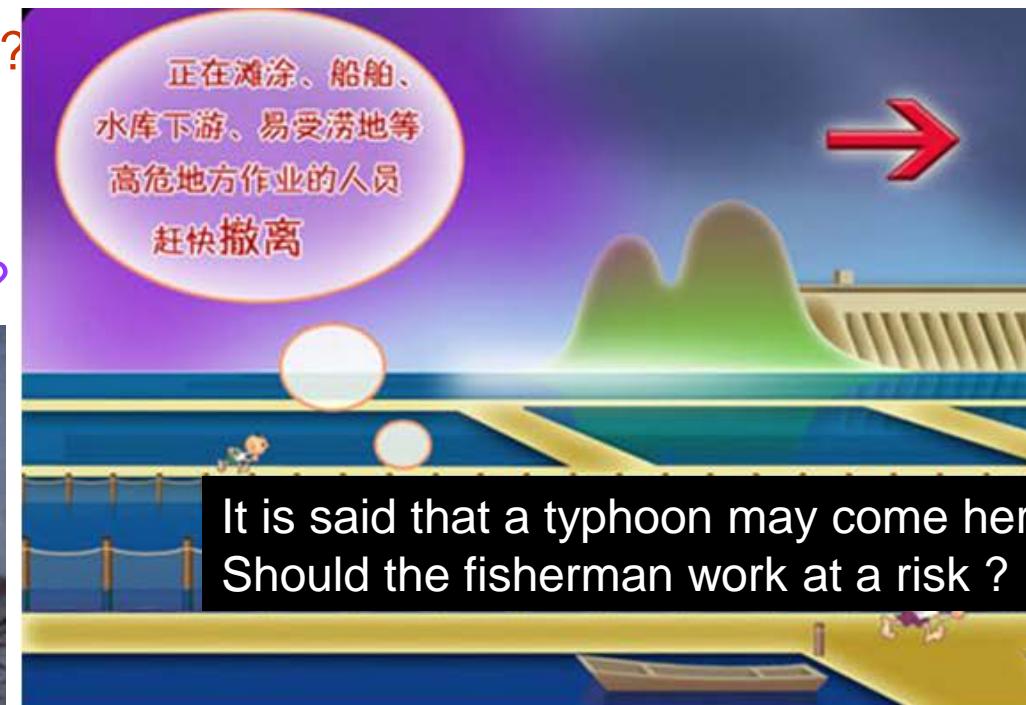
What should we do ?

How to define the risk (loss function) ?

How to obtain the minimum-risk
and make the decision safe enough ?



A earthquake may occur at a probability of 50%.
Should we issue the earthquake forecast ?



It is said that a seaquake may come at a probability of

Minimum-risk decision

The conditional risk

$$R(\alpha_i | x) = \sum_{j=1}^{j=c} \lambda(\alpha_i | \omega_j) P(\omega_j | x)$$



$\lambda(\alpha_i | \omega_j)$ can be defined in accordance with the real applications.

$\lambda(\alpha_i | \omega_j) \equiv \lambda_{ij}$:The loss (**cost**) in the case where **the class label is j but action α_i is adopted** (the test sample is classified into the i-th class).

For two-class problem



$$R(\alpha_1 | x) = \lambda(\alpha_1 | \omega_1)P(\omega_1 | x) + \lambda(\alpha_1 | \omega_2)P(\omega_2 | x)$$

$$R(\alpha_2 | x) = \lambda(\alpha_2 | \omega_1)P(\omega_1 | x) + \lambda(\alpha_2 | \omega_2)P(\omega_2 | x)$$

If $R(\alpha_1 | x) < R(\alpha_2 | x)$

Then the test sample is classified into the first class ω_1

Minimum-risk decision

$$R(\alpha_1 \mid x) < R(\alpha_2 \mid x) \longrightarrow \lambda_{21} \frac{P(x \mid \omega_1)P(\omega_1)}{P(x)} + \lambda_{22} \frac{P(x \mid \omega_2)P(\omega_2)}{P(x)} > \lambda_{11} \frac{P(x \mid \omega_1)P(\omega_1)}{P(x)} + \lambda_{12} \frac{P(x \mid \omega_2)P(\omega_2)}{P(x)}$$



$$\frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} > \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)}$$

- Regions of decision and zero-one loss function, therefore:

Let $\frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} \cdot \frac{P(\omega_2)}{P(\omega_1)} = \theta_\lambda$ then decide ω_1 if : $\frac{P(x | \omega_1)}{P(x | \omega_2)} > \theta_\lambda$

- If λ is the zero-one loss function which means:

$$\lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\text{then } \theta_\lambda = \frac{P(\omega_2)}{P(\omega_1)} = \theta_a$$

$$\text{if } \lambda = \begin{pmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} \text{ then } \theta_\lambda = \frac{2P(\omega_2)}{P(\omega_1)} = \theta_b$$

A high λ_{ij} means a high loss (**cost**).

λ_{ii} may be or be not zero.

An example: cancer diagnosis (+: 49%; - : 51%. How to reduce the risk ?)

An example: earthquake forecast



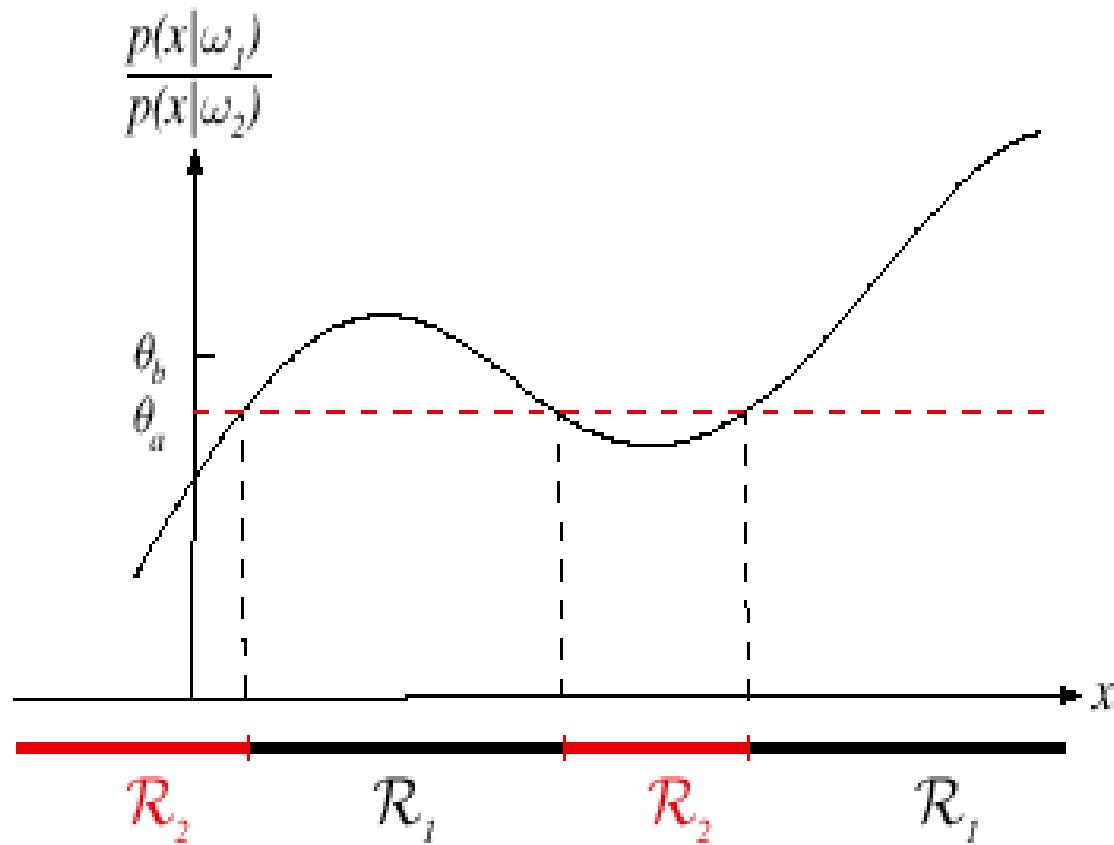


FIGURE 2.3. The likelihood ratio $p(x|\omega_1)/p(x|\omega_2)$ for the distributions shown in Fig. 2.1. If we employ a zero-one or classification loss, our decision boundaries are determined by the threshold θ_a . If our loss function penalizes miscategorizing ω_2 as ω_1 patterns more than the converse, we get the larger threshold θ_b , and hence \mathcal{R}_1 becomes smaller. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

■ Example of Minimum-Error-Rate Classification

- The **error-rate** is the sole criterion of classification



例题：

地震预报是比较困难的一个课题，可以根据地震与生物异常反应之间的联系来进行研究。根据历史记录的统计，地震前一周内出现生物异常反应的概率为 50%，而一周内没有发生地震但也出现了生物异常反应的概率为 10%。假设某一个地区属于地震高发区，发生地震的概率为 20%。问：

如果某日观察到明显的生物异常反应现象，是否应当预报一周内将发生地震？

解：

把地震是否发生设成两个类别：发生地震为 ω_1 ，不发生地震为 ω_2 ；

则两个类别出现的先验概率 $P_1=0.2$, $P_2=1-0.2=0.8$ ；

设地震前一周是否出现生物异常反应这一事件设为 x ，当 $x=1$ 时表示出现了， $x=0$ 时表示没出现；

则根据历史记录统计可得，: $p(x=1|\omega_1)=0.5$, $p(x=1|\omega_2)=0.1$

设地震前一周是否出现生物异常反应这一事件设为 x , 当 $x=1$ 时表示出现了, $x=0$ 时表示没出现;

则根据历史记录统计可得,; $p(x=1|\omega_1)=0.5$, $p(x=1|\omega_2)=0.1$

所以, 某日观察到明显的生物异常反应现象, 此时可以得到将发生地震的概率为:

$$\begin{aligned} p(\omega_1|x=1) &= (P_1 \times p(x=1|\omega_1)) / (P_1 \times p(x=1|\omega_1) + P_2 \times p(x=1|\omega_2)) \\ &= (0.2 \times 0.5) / (0.2 \times 0.5 + 0.8 \times 0.1) = 5/9 \end{aligned}$$

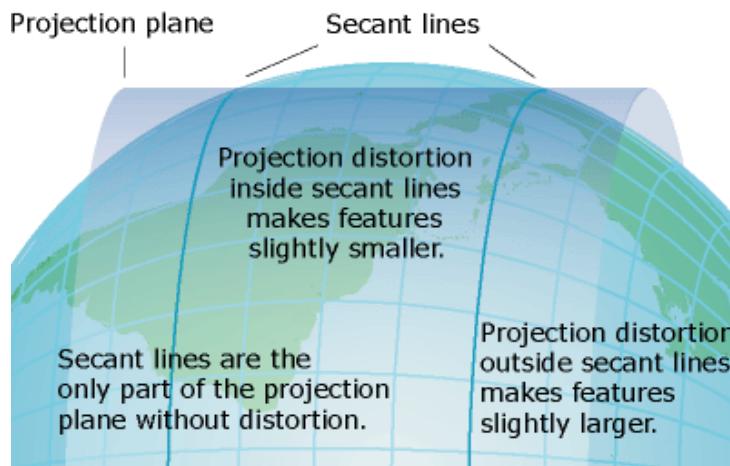
而不发生地震的概率为:

$$\begin{aligned} p(\omega_2|x=1) &= (P_2 \times p(x=1|\omega_2)) / (P_1 \times p(x=1|\omega_1) + P_2 \times p(x=1|\omega_2)) \\ &= (0.8 \times 0.1) / (0.2 \times 0.5 + 0.8 \times 0.1) = 4/9 \end{aligned}$$

因为 $p(\omega_1|x=1) > p(\omega_2|x=1)$, 所以在观察到明显的生物异常反应现象时, 发生地震的概率更高, 所以应当预报一周内将发生地震。

Disadvantage of Minimum-Error-Rate Classification

Minimum-Error might be not optimal. The cost is lower than the tunnel



Minimum-risk is better !!

Minimum-risk

例题：

对于上例中的地震预报问题，假设预报一周内发生地震，可以预先组织抗震救灾，由此带来的防灾成本会有 2500 万元，而当地震确实发生时，由于地震造成的直接损失会有 1000 万元；假设不预报将发生地震而地震又发生了，造成的损失会达到 5000 万元。请问在观察到明显的生物异常反应后，是否应当预报一周内将发生地震？

解：

设决策 1 为发布地震预报，决策 2 为不发布地震预报，则



Minimum-risk

发生了地震，而提前发布了地震预报，此时的损失为 $\lambda_{11}=2500+1000=3500$ 万元；

发生了地震，而没有提前发布地震预报，此时的损失为 $\lambda_{21}=5000$ 万元；

没有发生地震，而提前发布了地震预报，此时的损失为 $\lambda_{12}=2500$ 万元；

没有发生地震，而没有提前发布地震预报，此时的损失为 $\lambda_{22}=0$ 元；

则在观察到明显的生物异常反应现象时，发布地震预报的条件风险为：

$$R(\text{发布地震预报}|x=1) = \lambda_{11} \times p(\omega_1|x=1) + \lambda_{12} \times p(\omega_2|x=1) = 3500 \times 5/9 + 2500 \times 4/9 = 3056 \text{ 万元；}$$

而不发布地震预报带来的综合损失为：

$$R(\text{不发布地震预报}|x=1) = \lambda_{21} \times p(\omega_1|x=1) + \lambda_{22} \times p(\omega_2|x=1) = 5000 \times 5/9 = 2778 \text{ 万元；}$$

因为 $R(\text{发布地震预报}|x=1) > R(\text{不发布地震预报}|x=1)$

所以，发布地震预报风险更大，不应该发布地震预报。



2.4 Classifiers, Discriminant Functions and Decision Surfaces

- The multi-category case
 - Set of discriminant functions $g_i(x)$, $i = 1, \dots, c$
 - The classifier assigns a feature vector x to class ω_i
if: $g_i(x) > g_j(x) \quad \forall j \neq i$

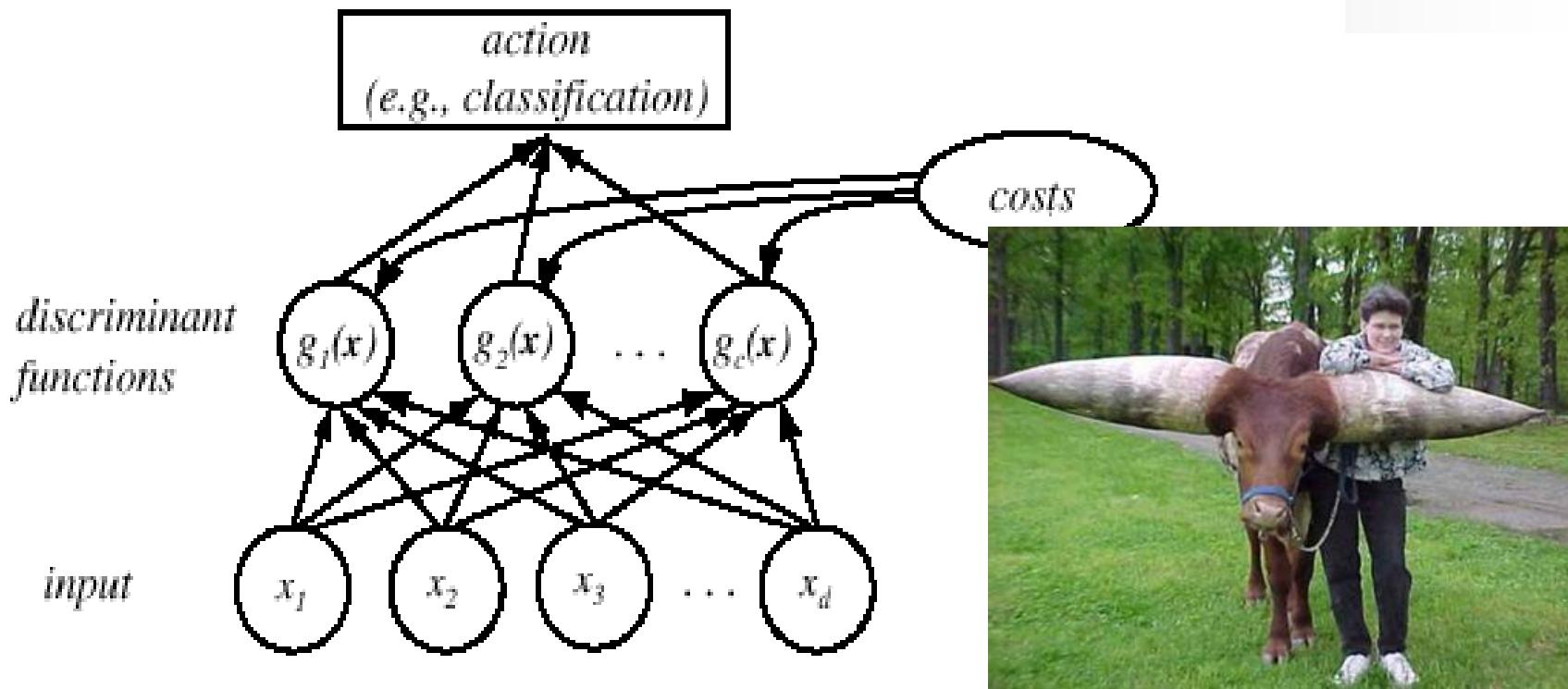


FIGURE 2.5. The functional structure of a general statistical pattern classifier which includes d inputs and c discriminant functions $g_i(\mathbf{x})$. A subsequent step determines which of the discriminant values is the maximum, and categorizes the input pattern accordingly. The arrows show the direction of the flow of information, though frequently the arrows are omitted when the direction of flow is self-evident. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- For the minimum *risk* case

Let $g_i(x) = -R(\alpha_i | x)$

(max. discriminant corresponds to min. risk!)

- For the minimum error rate case, we take

$g_i(x) = P(\omega_i | x)$

(max. discrimination corresponds to max. posterior!)

$$g_i(x) \equiv p(x | \omega_i) P(\omega_i)$$

$$g_i(x) = \ln p(x | \omega_i) + \ln P(\omega_i)$$

(\ln : natural logarithm!)



- Feature space divided into c decision regions

if $g_i(x) > g_j(x) \forall j \neq i$ then x is in R_i

(R_i means to assign x to ω_i)

- The two-category case

- A classifier is a “dichotomizer” that has two discriminant functions g_1 and g_2

Let $g(x) = g_1(x) - g_2(x)$

Decide ω_1 if $g(x) > 0$; Otherwise decide ω_2



□ The computation of $g(x)$

$$g(x) = P(\omega_1 | x) - P(\omega_2 | x)$$

$$g(x) = \ln \frac{p(x | \omega_1)}{p(x | \omega_2)} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$



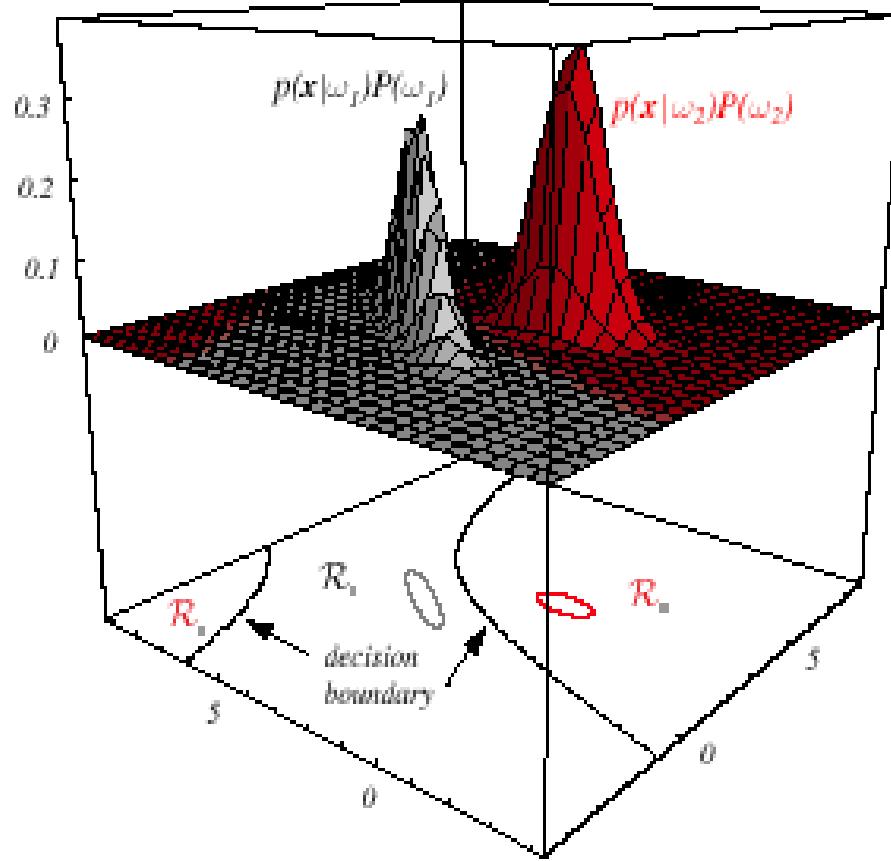
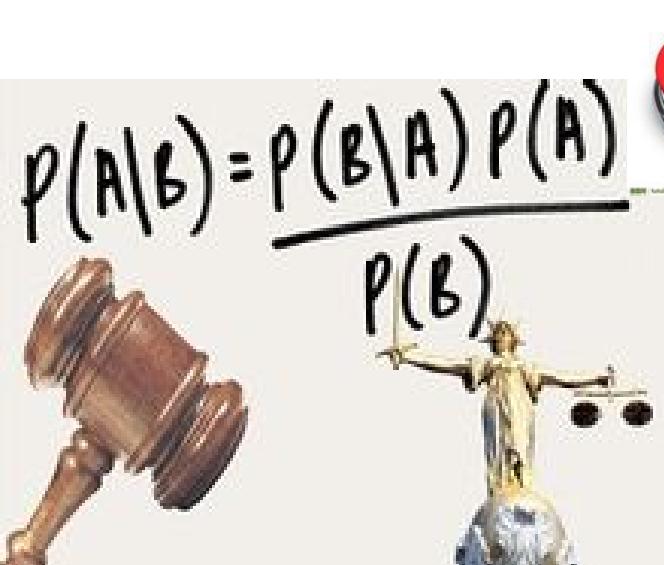


FIGURE 2.6. In this two-dimensional two-category classifier, the probability densities are Gaussian, the decision boundary consists of two hyperbolas, and thus the decision region \mathcal{R}_2 is not simply connected. The ellipses mark where the density is $1/e$ times that at the peak of the distribution. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

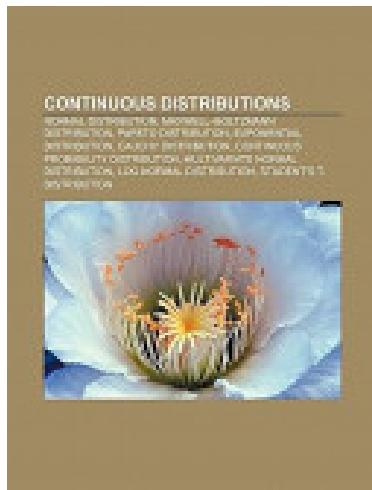
Difficulties of Bayesian decision

- It's not easy to obtain the probability (prior and conditional probability)





- To assume that the conditional probability satisfies the normal distributions is a basic way.
- The real-world verifies that the normal distributions is really consistent with the true distribution



2.5 The Normal Density

■ Univariate density

- Continuous density
- A lot of processes are asymptotically Gaussian
- Handwritten characters, speech sounds are ideal or prototype corrupted by random process (central limit theorem)

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

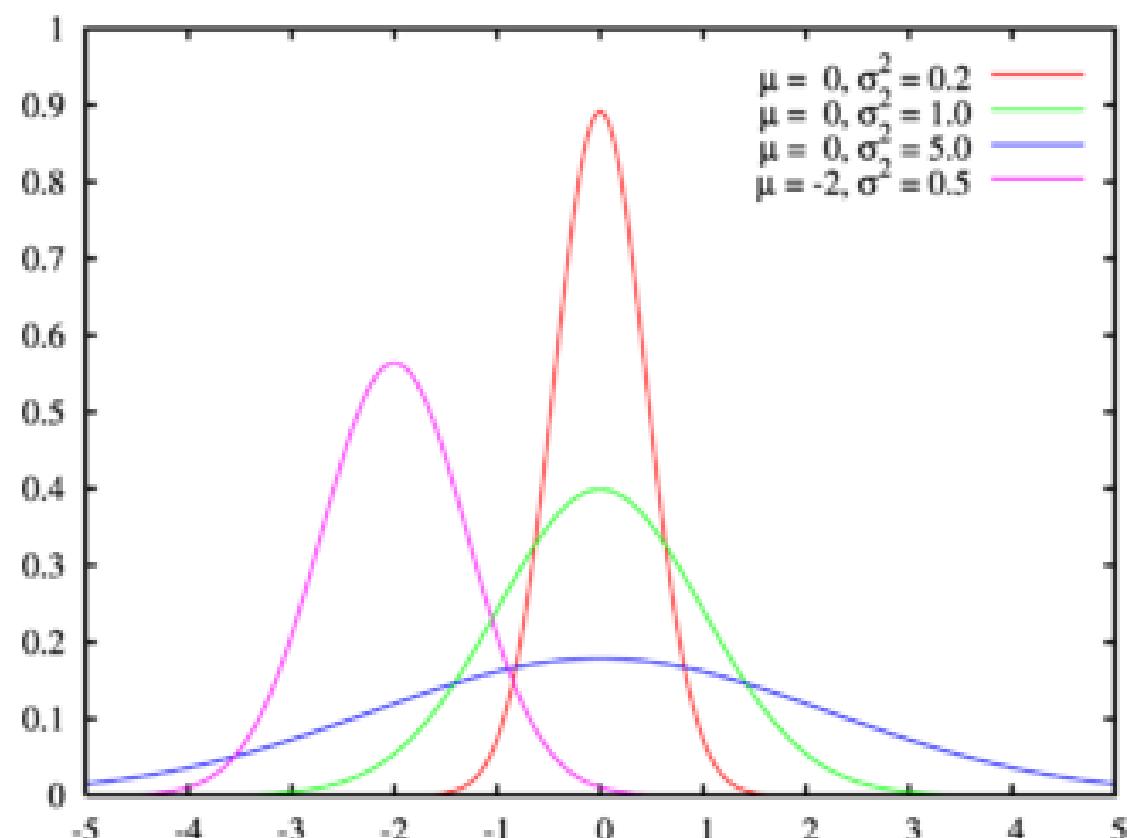
Where:

μ = mean (or expected value) of x

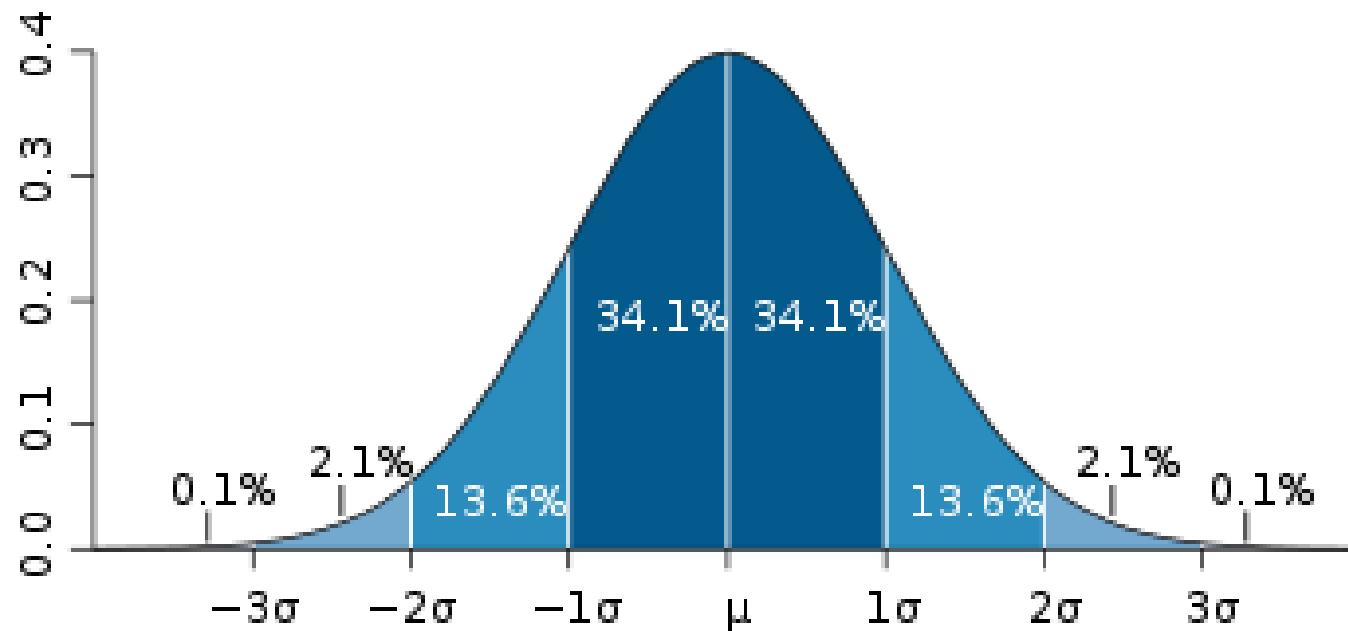
σ^2 = expected squared deviation or variance



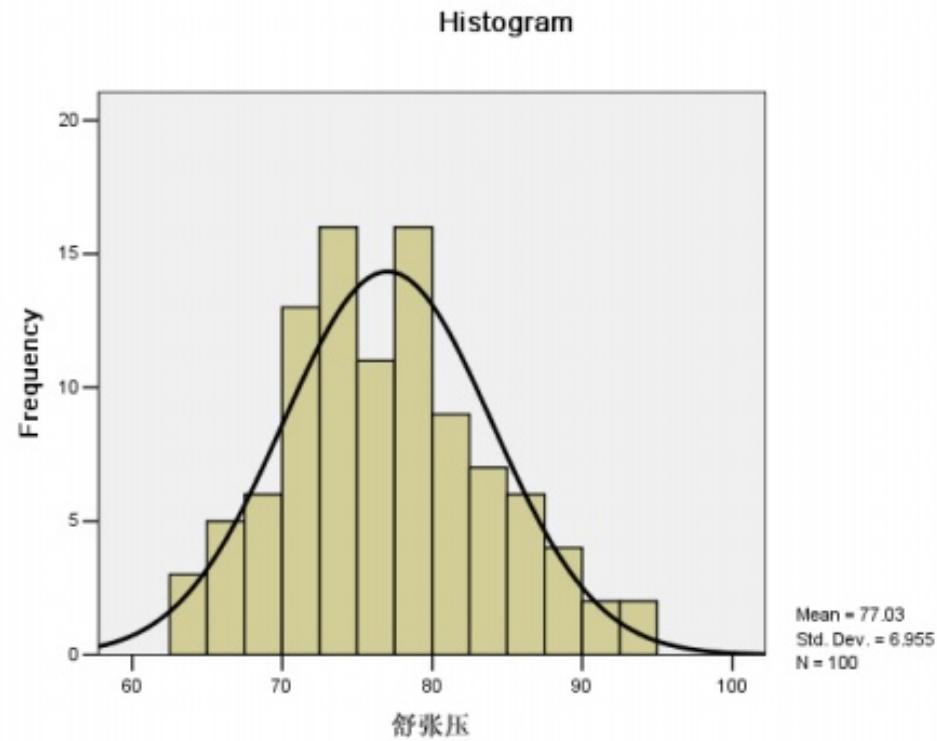
Examples of normal distributions



The 68-95-99.7 rule



Examples in the real world



Multivariate density

- Multivariate normal density in d dimensions is:

$$P(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu)^t \Sigma^{-1} (x - \mu)\right]$$

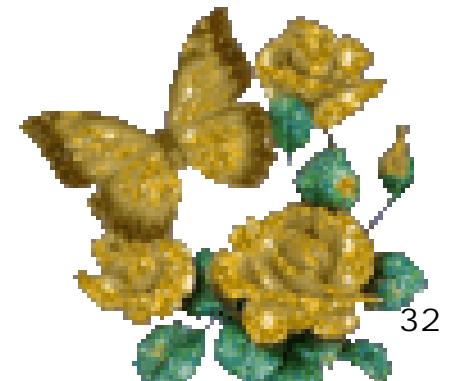
where:

$x = (x_1, x_2, \dots, x_d)^t$ (t stands for the transpose vector form)

$\mu = (\mu_1, \mu_2, \dots, \mu_d)^t$ mean vector

$\Sigma = d*d$ covariance matrix

$|\Sigma|$ and Σ^{-1} are determinant and inverse respectively





$$\mu = \mathcal{E}[x] = \int x p(x) dx$$

$$\Sigma = \mathcal{E}[(x - \mu)(x - \mu)^t] = \int (x - \mu)(x - \mu)^t p(x) dx$$

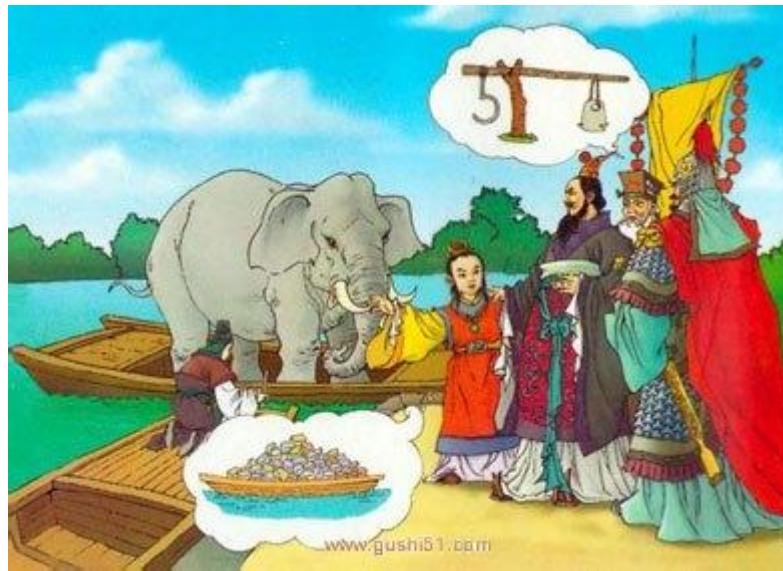
$$\mu_i = \mathcal{E}[x_i]$$

$$\sigma_{ij} = \mathcal{E}[(x_i - \mu_i)(x_j - \mu_j)]$$

■ Mahalanobis distance from x to μ

$$r^2 = (x - \mu)^t \Sigma^{-1} (x - \mu)$$

Mahalanobis distance is widely used in pattern recognition



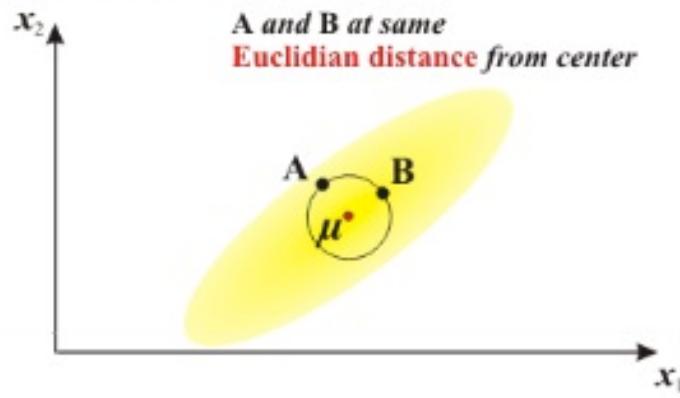
◆ 我们熟悉的欧氏距离虽然很有用，但在解决多元数据的分析问题时，就显示出了它的不足之处。一是它没有考虑到总体的变异对“距离”远近的影响，显然一个变异程度大的总体可能与更多样品近些，即使它们的欧几里得距离不一定最近；另外，欧几里得距离受变量的量纲影响，这对多元数据的处理是不利的。

马氏距离优点

它不受量纲的影响，两点之间的马氏距离与原始数据的测量单位无关；由标准化数据和中心化数据(即原始数据与均值之差)计算出的二点之间的马氏距离相同。马氏距离还可以排除变量之间的相关性的干扰。

十一 欧氏距离与马氏距离的区别与联系

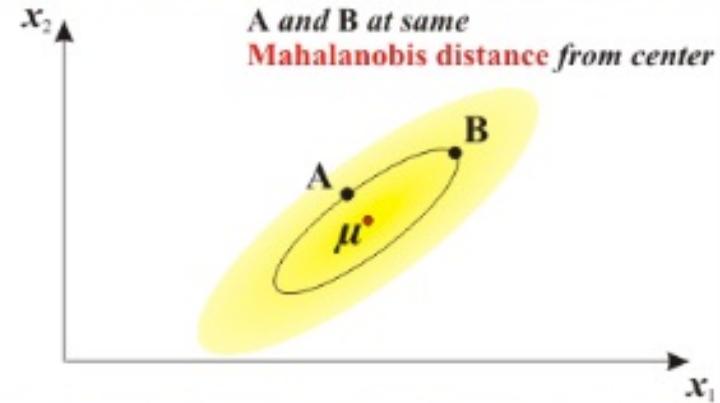
AI ACCESS



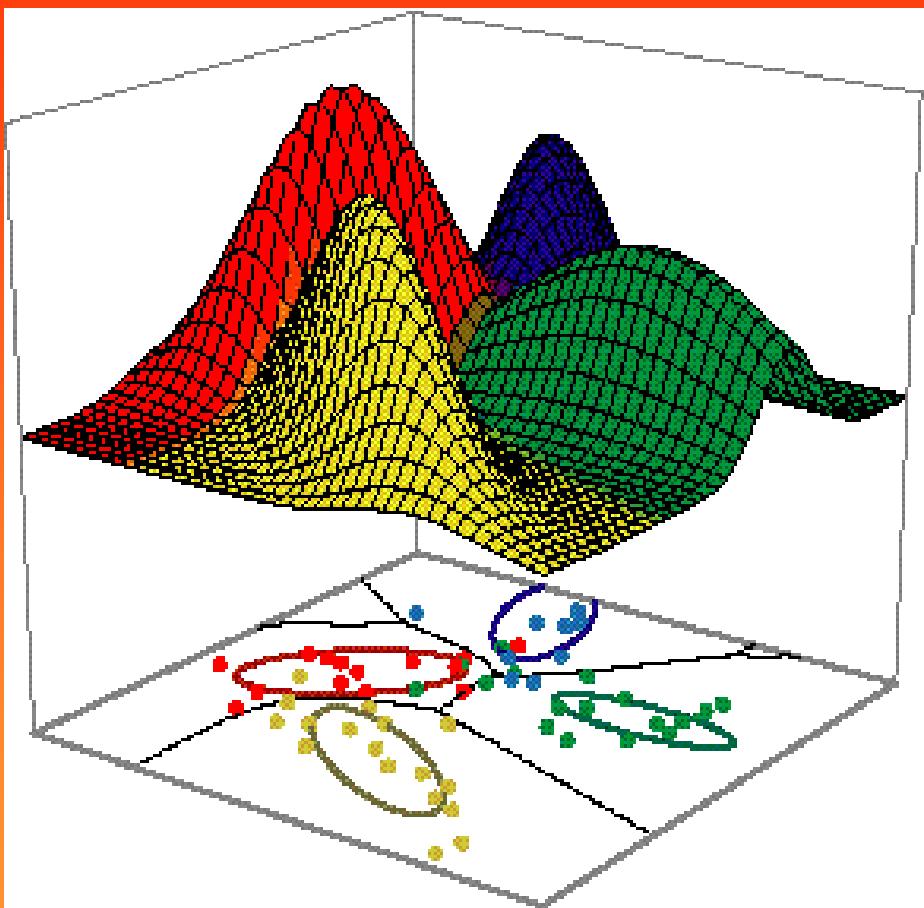
❖ 欧式距离

❖ 马氏距离

AI ACCESS



Pattern Classification



All materials in these slides were taken from
Pattern Classification (2nd ed) by R.
O. Duda, P. E. Hart and D. G. Stork,
John Wiley & Sons, 2000
with the permission of the authors
and the publisher

Chapter 2 Bayesian Decision Theory (Sections 2-6, 2-9)

- Discriminant Functions for the Normal Density
- Bayes Decision Theory – Discrete Features

2.6 Discriminant Functions for the Normal Density

- We saw that the minimum error-rate classification can be achieved by the discriminant function

$$g_i(x) = \ln p(x | \omega_i) + \ln P(\omega_i)$$

- Case of multivariate normal

$$p(x | \omega_i) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i)\right]$$



$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t (\sum_i)^{-1} (x - \mu_i) - \frac{d}{2} \ln 2\pi - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

Case $\Sigma_i = \sigma^2.I$ (I stands for the identity matrix)

$$g_i(x) = -\frac{\|X - \mu_i\|^2}{2\sigma^2} + \ln P(\omega_i)$$



■ Case $\Sigma_i = \sigma^2 I$ (I stands for the identity matrix)

$g_i(x) = w_i^t x + w_{i0}$ (*linear discriminant function*)

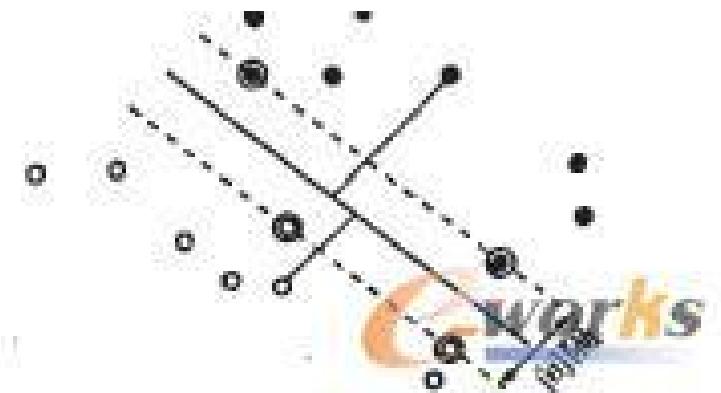
where :

$$w_i = \frac{\mu_i}{\sigma^2}; \quad w_{i0} = -\frac{1}{2\sigma^2} \mu_i^t \mu_i + \ln P(\omega_i)$$

(ω_{i0} is called the threshold for the i th category!)

- A classifier that uses linear discriminant functions is called “a linear machine”
- The decision surfaces for a linear machine are pieces of hyperplanes defined by:

$$g_i(x) = g_j(x)$$



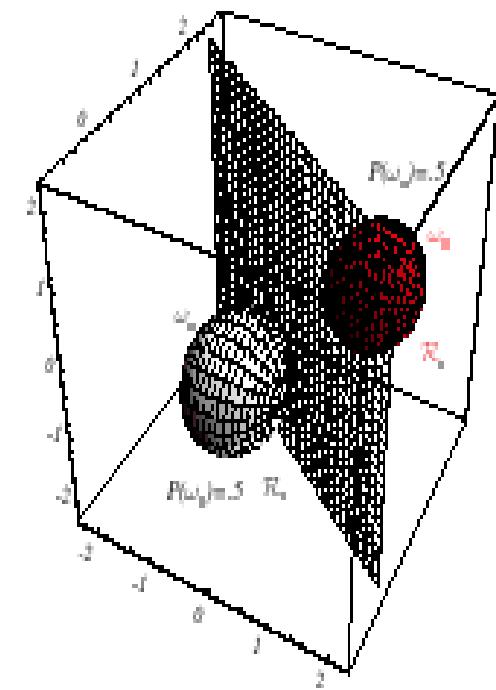
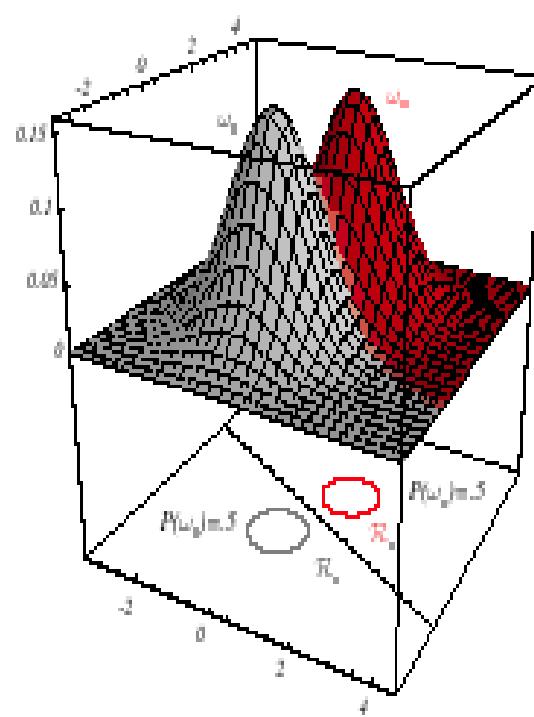
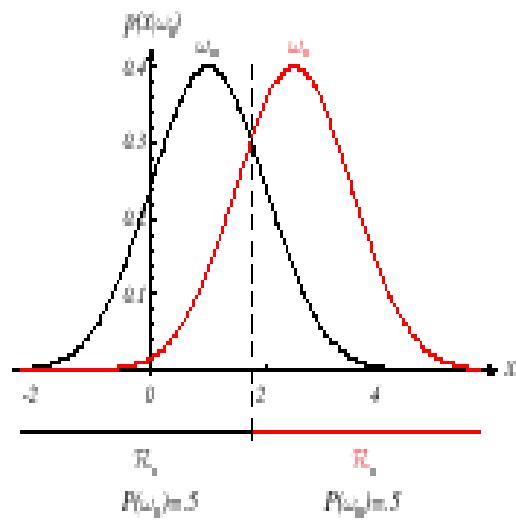


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d - 1$ dimensions, perpendicular to the line separating the means. In these one-, two-, and three-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the three-dimensional case, the grid plane separates \mathcal{R}_1 from \mathcal{R}_2 . From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



- The hyperplane separating \mathcal{R}_i and \mathcal{R}_j
is always orthogonal to the line linking the means!

$$g_i(x) = g_j(x)$$

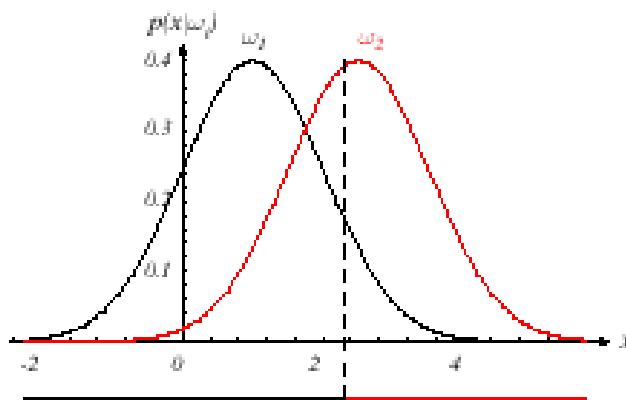


$$\mathbf{w}^t(\mathbf{x} - \mathbf{x}_0) = 0$$

$$\mathbf{w} = \boldsymbol{\mu}_i - \boldsymbol{\mu}_j$$

$$x_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j) - \frac{\sigma^2}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$$

$$\text{if } P(\omega_i) = P(\omega_j) \text{ then } x_0 = \frac{1}{2}(\boldsymbol{\mu}_i + \boldsymbol{\mu}_j)$$

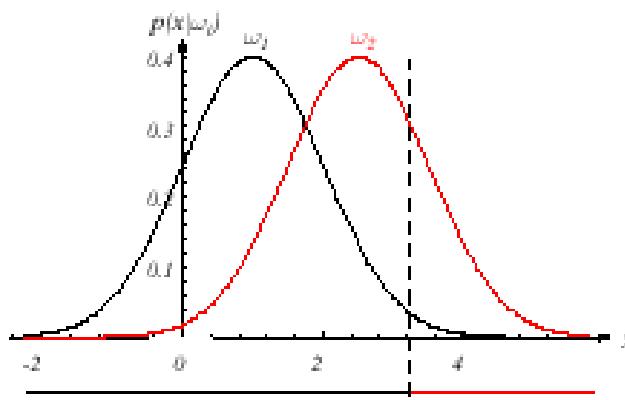


$$\mathcal{R}_1$$

$P(\omega_1) = .7$

$$\mathcal{R}_2$$

$P(\omega_2) = .3$

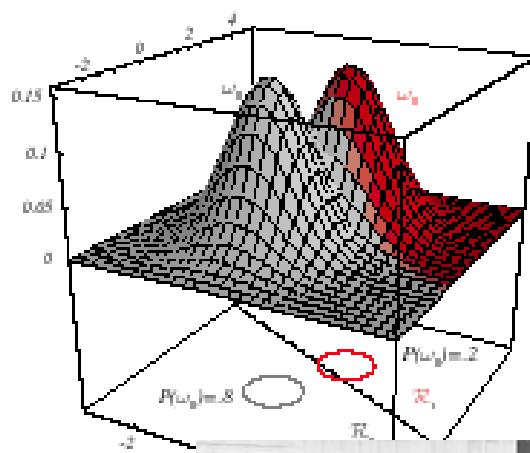


$$\mathcal{R}_1$$

$P(\omega_1) = .9$

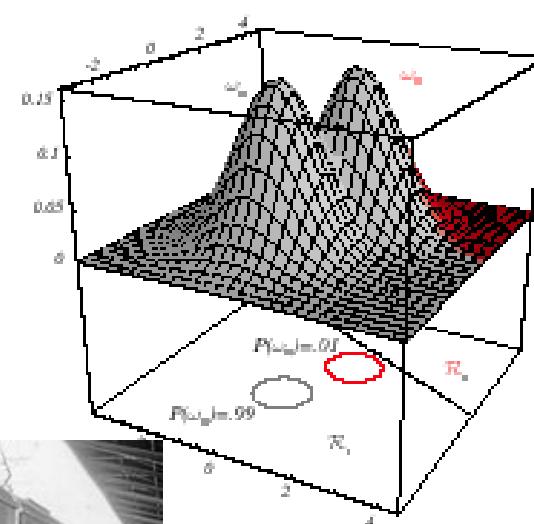
$$\mathcal{R}_2$$

$P(\omega_2) = .1$



part 3)

8



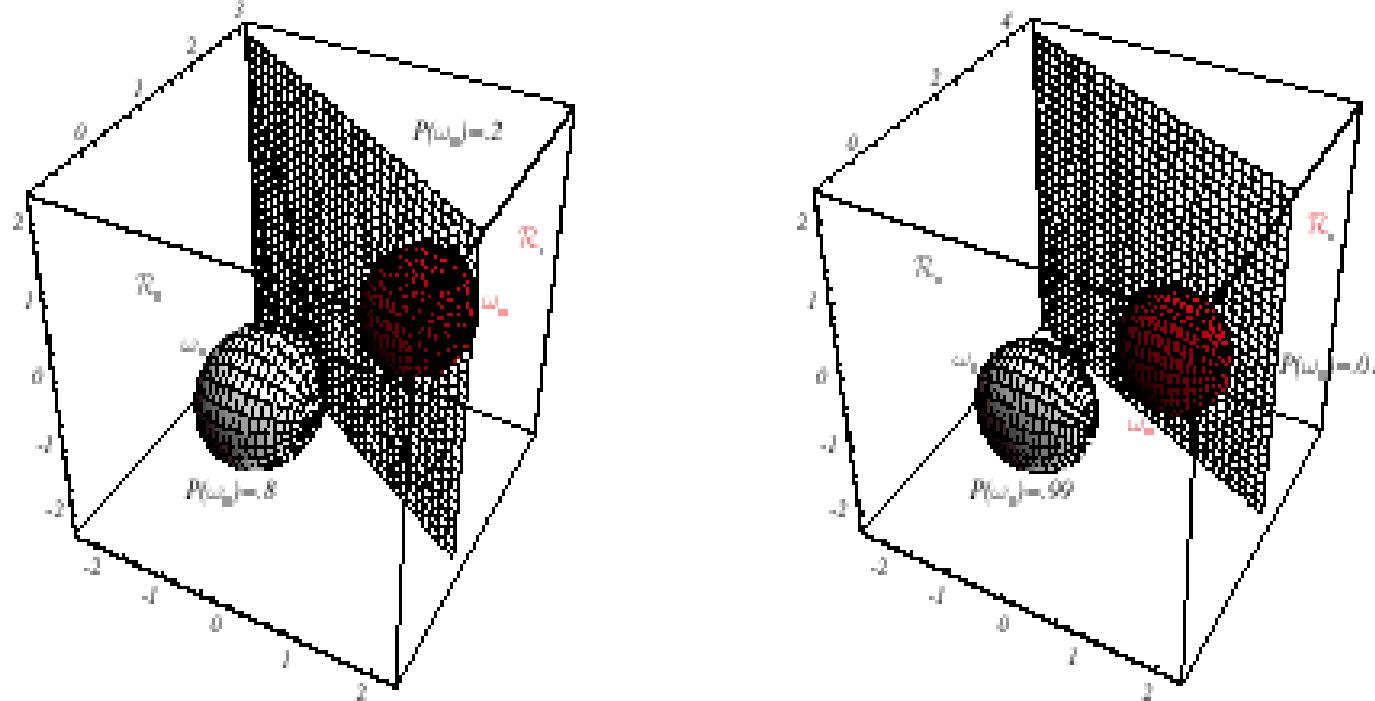


FIGURE 2.11. As the priors are changed, the decision boundary shifts; for sufficiently disparate priors the boundary will not lie between the means of these one-, two- and three-dimensional spherical Gaussian distributions. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



- Case $\Sigma_i = \Sigma$ (covariance of all classes are identical but arbitrary!)



$$g_i(x) = -\frac{1}{2}(X - \mu_i)^T \Sigma^{-1}(X - \mu_i) + \ln P(\omega_i)$$

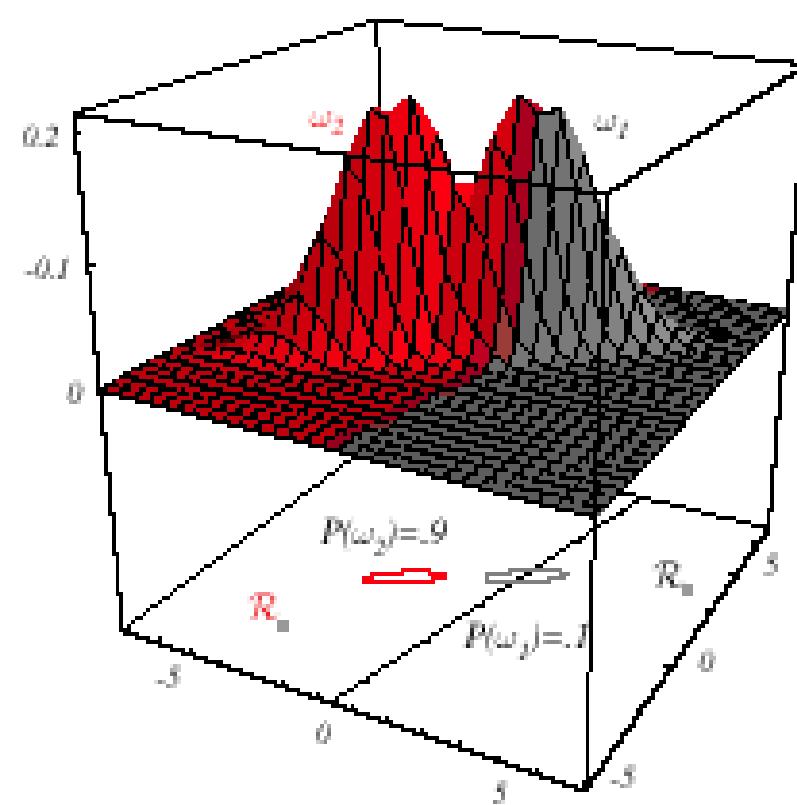
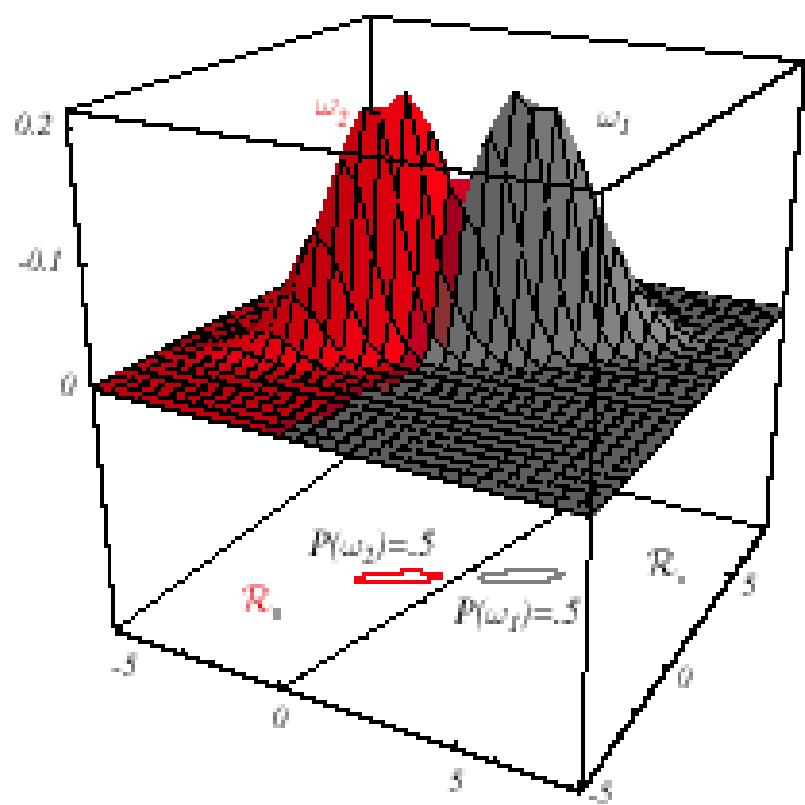
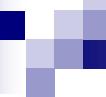
- Case $\Sigma_i = \Sigma$ (covariance of all classes are identical but arbitrary!)
 - Hyperplane separating \mathcal{R}_i and \mathcal{R}_j

$$w^t(x - x_0) = 0$$

$$w = \Sigma^{-1}(\mu_i - \mu_j)$$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\ln[P(\omega_i)/P(\omega_j)]}{(\mu_i - \mu_j)^t \Sigma^{-1}(\mu_i - \mu_j)} \cdot (\mu_i - \mu_j)$$

(the hyperplane separating \mathcal{R}_i and \mathcal{R}_j is generally not orthogonal to the line between the means!)



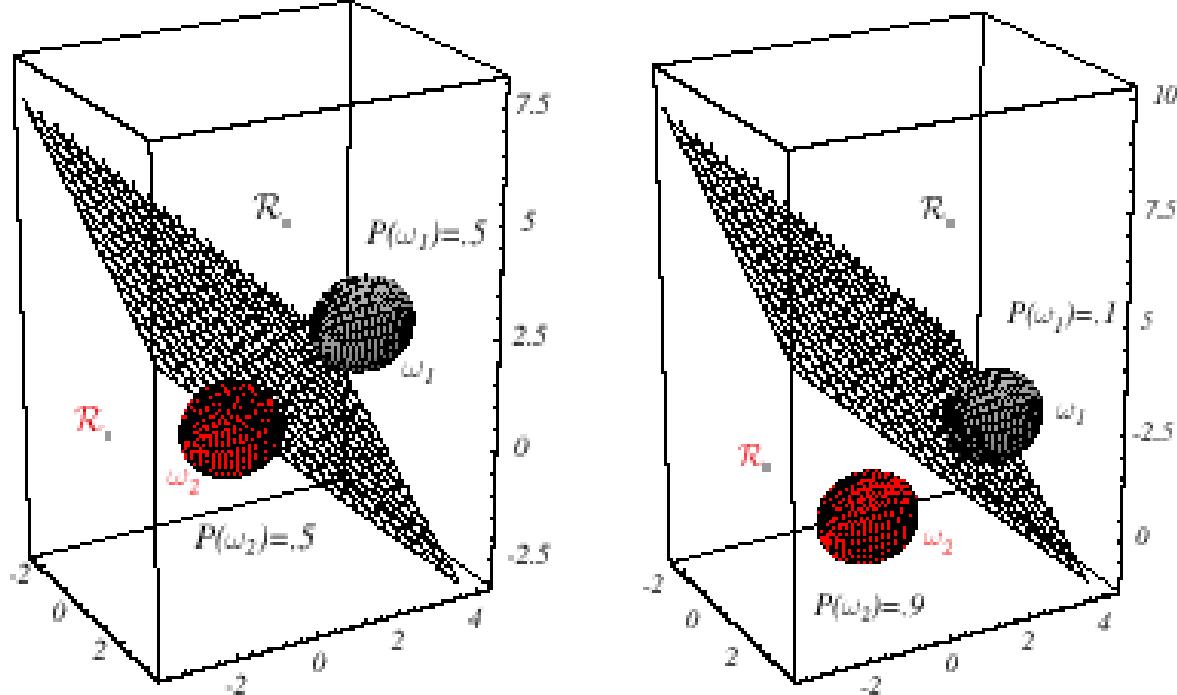


FIGURE 2.12. Probability densities (indicated by the surfaces in two dimensions and ellipsoidal surfaces in three dimensions) and decision regions for equal but asymmetric Gaussian distributions. The decision hyperplanes need not be perpendicular to the line connecting the means. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Case $\Sigma_i = \text{arbitrary}$

- The covariance matrices are different for each category

$$g_i(x) = x^t W_i x + w_i^t x + w_{i0}$$

where :

$$W_i = -\frac{1}{2} \Sigma_i^{-1}$$

$$w_i = \Sigma_i^{-1} \mu_i$$

$$w_{i0} = -\frac{1}{2} \mu_i^t \Sigma_i^{-1} \mu_i - \frac{1}{2} \ln |\Sigma_i| + \ln P(\omega_i)$$

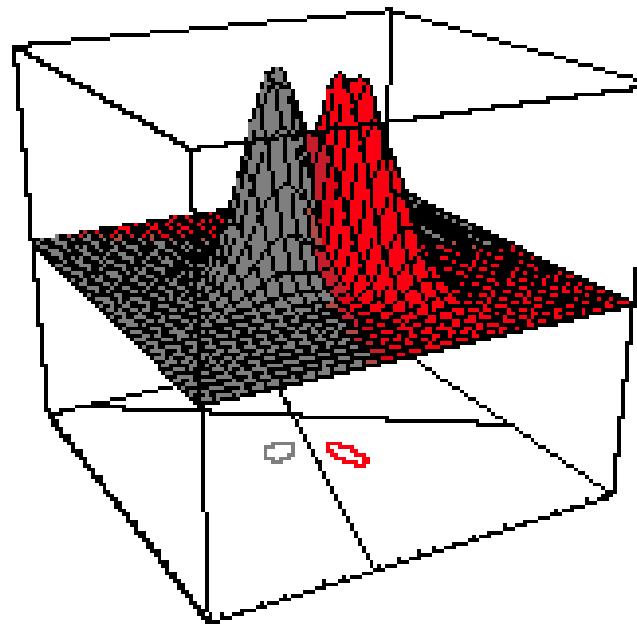
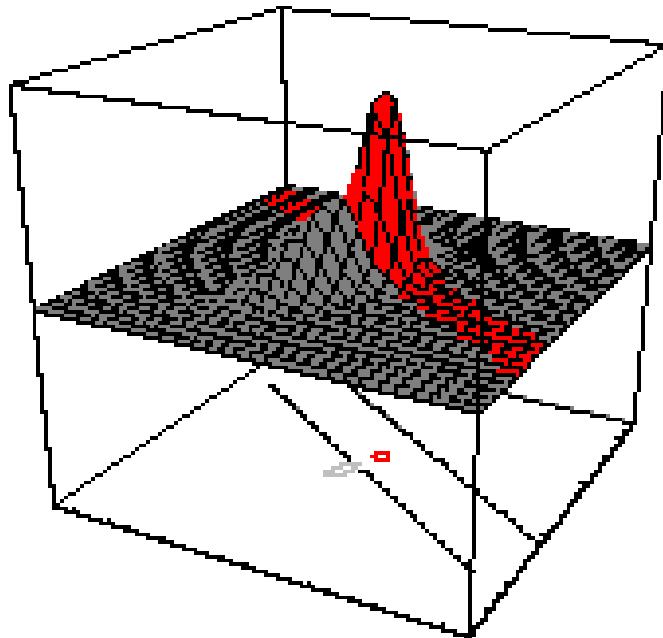
(Hyperquadrics which are: hyperplanes, pairs of hyperplanes, hyperspheres, hyperellipsoids, hyperparaboloids, hyperboloids)

Case Σ_i = arbitrary

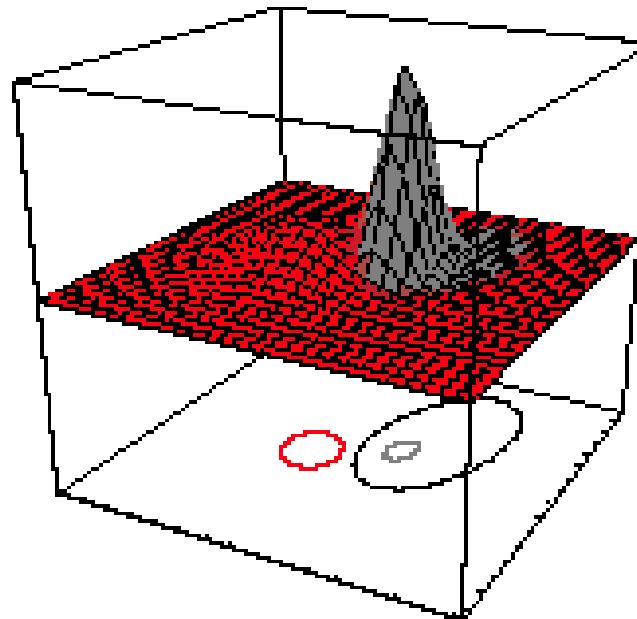
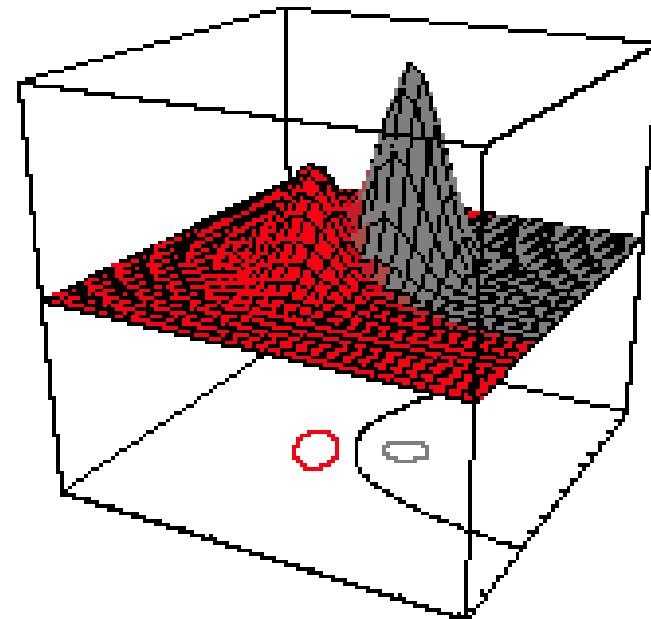
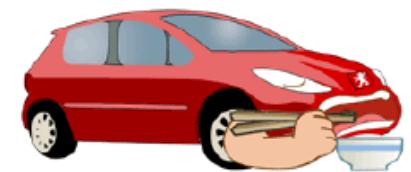
The covariance matrices are
different for each category

$$g_i(x) = -\frac{1}{2}(x - \mu_i)^t(\Sigma_i)^{-1}(x - \mu_i) - \frac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$





吃饭



Pattern Classification Chapter 2(Part 3)

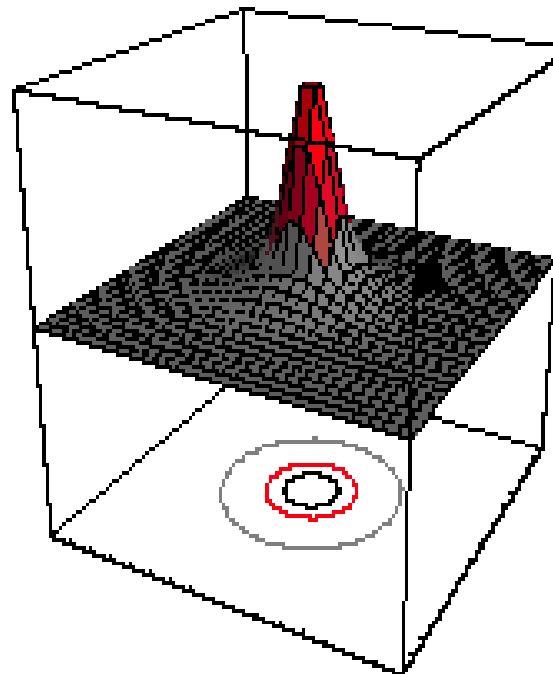
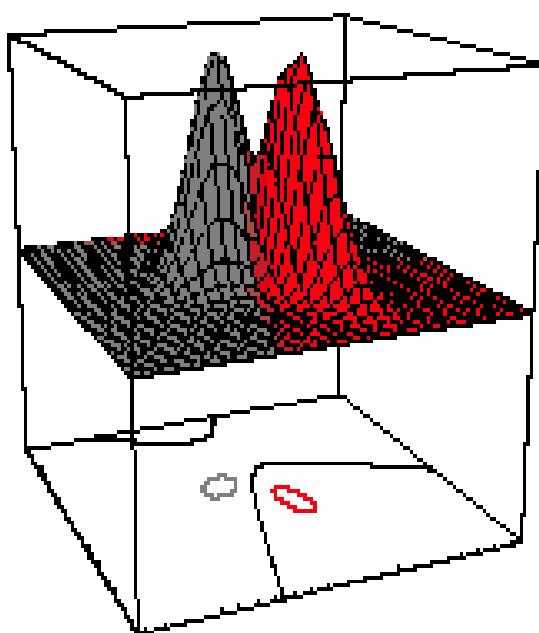


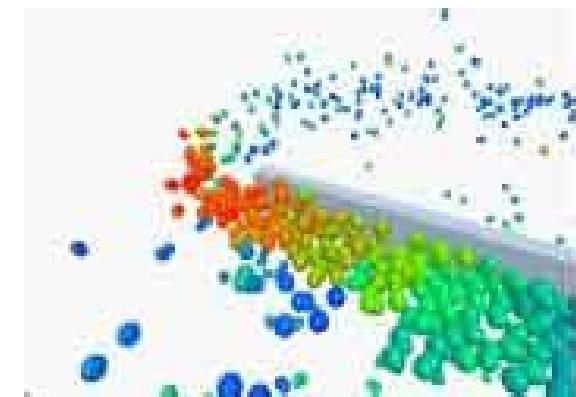
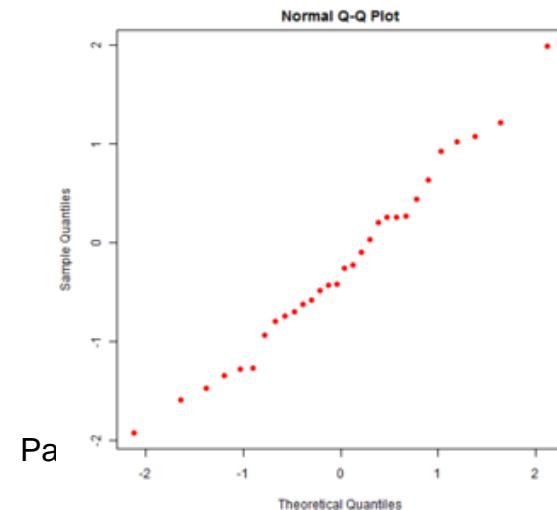
FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability density. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

2.9 Bayes Decision Theory – Discrete Features

- Components of x are binary or integer valued, x can take only one of m discrete values

$$v_1, v_2, \dots, v_m$$

$$P(\omega_j | x) = \frac{P(x | \omega_j)P(\omega_j)}{P(x)}$$



It's hard to evaluate the probability of discrete high-dimensional sample

- $x=[\text{he and she first went shopping.....}]$
- How to evaluate the value of $p(x)$?



Case of independent binary features in 2 category problem

Let $x = [x_1, x_2, \dots, x_d]^t$ where each x_i is either 0 or 1, with probabilities:

$$p_i = P(x_i = 1 | \omega_1) \quad (P(x_i = 0 | \omega_1) = 1 - p_i)$$

$$q_i = P(x_i = 1 | \omega_2) \quad (P(x_i = 0 | \omega_2) = 1 - q_i)$$

$$P(X / \omega_1) = \prod_{i=1}^d p_i^{x_i} (1 - p_i)^{1-x_i}$$

$$P(X / \omega_2) = \prod_{i=1}^d q_i^{x_i} (1 - q_i)^{1-x_i}$$

$$\frac{P(X / \omega_1)}{P(X / \omega_2)} = \prod_{i=1}^d \left(\frac{p_i}{q_i} \right)^{x_i} \left(\frac{1-p_i}{1-q_i} \right)^{1-x_i}$$



■ The discriminant function in this case is:

$$g(x) = \ln \frac{p(x | \omega_1)}{p(x | \omega_2)} + \ln \frac{p(\omega_1)}{p(\omega_2)}$$

$$g(x) = \sum_{i=1}^d w_i x_i + w_0$$

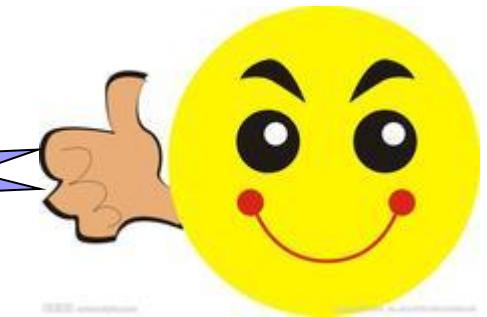
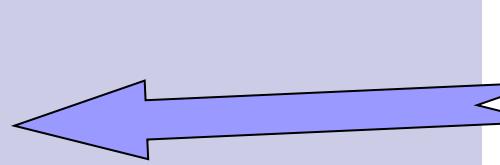
where :

$$w_i = \ln \frac{p_i(1 - q_i)}{q_i(1 - p_i)} \quad i = 1, \dots, d$$

and :

$$w_0 = \sum_{i=1}^d \ln \frac{1 - p_i}{1 - q_i} + \ln \frac{P(\omega_1)}{P(\omega_2)}$$

decide ω_1 if $g(x) > 0$ and ω_2 if $g(x) \leq 0$

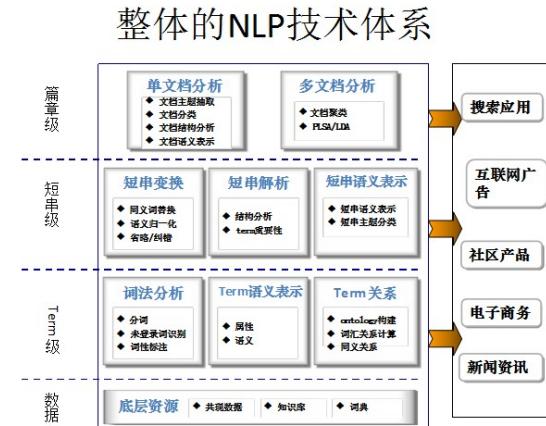


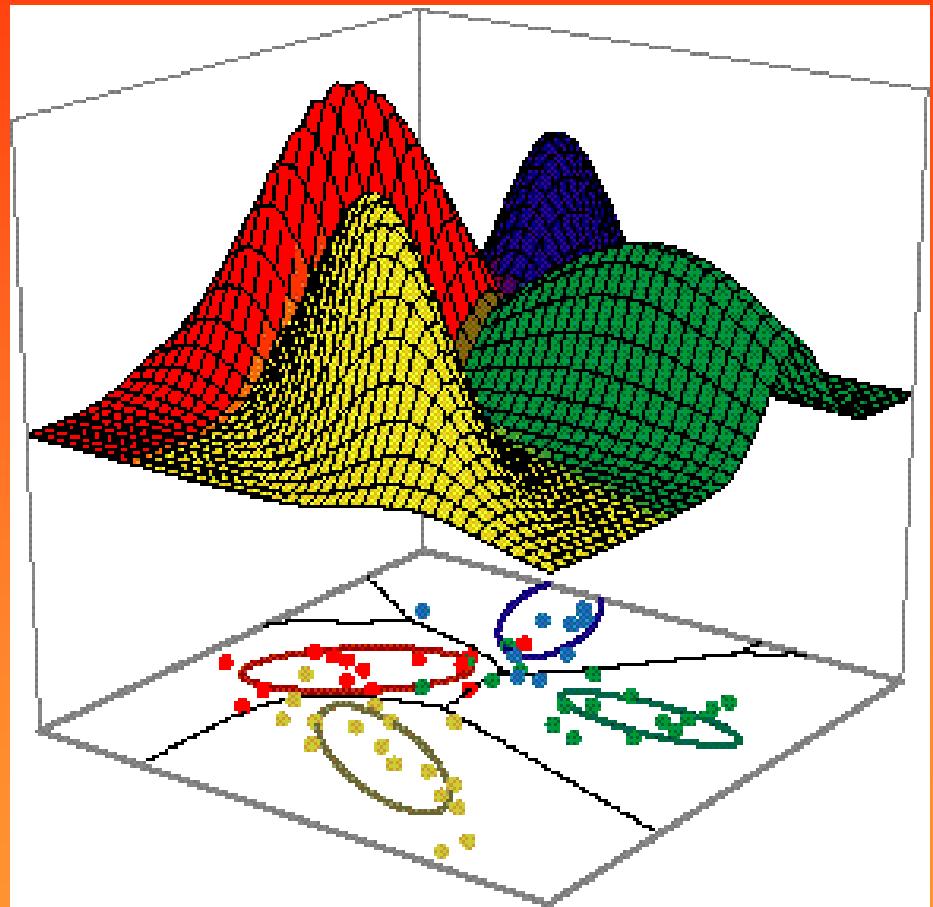
Assumption

- $V = [v_1, v_2, \dots, v_m]$
- $P(v|w_i) = P(v_1|w_i) \dots P(v_m|w_i)$
- *For natural language processing, v_1, v_2, \dots, v_m denote different words*



- Bayes Decision Theory on discrete features have been widely used in natural language processing, speech recognition, text classification, OCR et al.





Pattern Classification

Chapter 3: Maximum-Likelihood & Bayesian Parameter Estimation (3.1,3.2)

- ▶ Introduction
- ▶ Maximum-Likelihood Estimation
 - The Gaussian Case 1:unknown μ
 - The Gaussian Case 2: unknown μ and σ
 - Bias
- ▶

3.1 Introduction

► Data availability in a Bayesian framework

- We could design an optimal classifier if we knew:
 - ▶ $P(\omega_i)$ (priors)
 - ▶ $P(x | \omega_i)$ (class-conditional densities)—**Unknown parameters**
 - ▶ Unfortunately, we rarely have both complete information!

► Design a classifier from a training sample

- No problem with the estimation of prior probabilities
- Samples are often too few for the estimation of class-conditional densities
- Complexity for large dimension of feature space

- To simplify above problem
 - ▶ Normality of $P(x | \omega_i)$
 - ▶ $P(x | \omega_i) \sim N(\mu_i, \Sigma_i)$: Characterized by 2 parameters
 - ▶ The problem is changed from estimating $P(x | \omega_i)$ to estimating μ_i, Σ_i

► Estimation techniques

- Maximum-Likelihood (ML) and the Bayesian estimations
- Results are nearly identical, but the approaches are conceptually different



- Parameters in ML estimation are fixed but unknown!

Best parameters are obtained by maximizing the probability of obtaining the samples observed



- Bayesian methods view the parameters as random variables having some known prior distribution. Training data allow us to convert a distribution on this variable into a posterior probability density

In either approach, we use $P(\omega_i | x)$ for our classification rule!



3.2 Maximum-Likelihood Estimation

► M-L Estimation

- Has good convergence properties as the sample size increases
- Simpler than any other alternative techniques

► General principle

- Assume we have c classes and
 $p(x | \omega_j) \sim N(\mu_j, \Sigma_j)$
 $p(x | \omega_j) \equiv p(x | \omega_j, \theta_j)$ where:

$$\theta_j = (\mu_j, \Sigma_j) = (\mu_j^1, \mu_j^2, \dots, \sigma_j^{11}, \sigma_j^{22} \dots)$$



- Use the information provided by the training samples $D = (D_1, D_2, \dots, D_c)$ to estimate

$\theta = (\theta_1, \theta_2, \dots, \theta_c)$, each θ_i ($i = 1, 2, \dots, c$) is associated with each category.

assume D_i give no information about θ_j if $i \neq j$

So we use each class separately to simplify our notation

Suppose that D contains n samples, x_1, x_2, \dots, x_n

$$p(D | \theta) = \prod_{k=1}^{n} p(x_k | w, \theta) = \prod_{k=1}^{n} p(x_k | \theta) = F(\theta)$$

$p(D | \theta)$ is called the likelihood of θ



- ML estimation of θ is, by definition, the value $\hat{\theta}$ that maximizes $p(D | \theta)$

“It is the value of θ that best agrees with the actually observed training sample”

► ML Problem Statement

- Let $D = \{x_1, x_2, \dots, x_n\}$

$$p(x_1, \dots, x_n | \theta) = \prod_{k=1}^n P(x_k | \theta); |D| = n$$

Our goal is to determine $\hat{\theta}$ (value of θ that makes this sample the most representative!)



$$|D| = n$$

$$N(\mu_j, \Sigma_j) = P(x_j | \omega_c)$$

$x_1 \quad \cdot \quad \cdot \quad \cdot \quad x_2$
 $\cdot \quad x_n \quad \cdot \quad \cdot \quad \cdot$

$$P(x_j | \omega_1)$$

$x_{10} \quad x_{11}$
 x_{20}

D_k

$x_8 \quad \cdot \quad \cdot$
 $x_1 \quad x_9 \quad \cdot \quad \cdot$

D_1



$$\theta = (\theta_1, \theta_2, \dots, \theta_c)$$

Problem: find $\hat{\theta}$ such that:

$$\underset{\theta}{\text{Max P(D | \theta)} = \text{MaxP}(x_1, \dots, x_n | \theta)}$$

$$= \text{Max} \prod_{k=1}^n P(x_k | \theta)$$



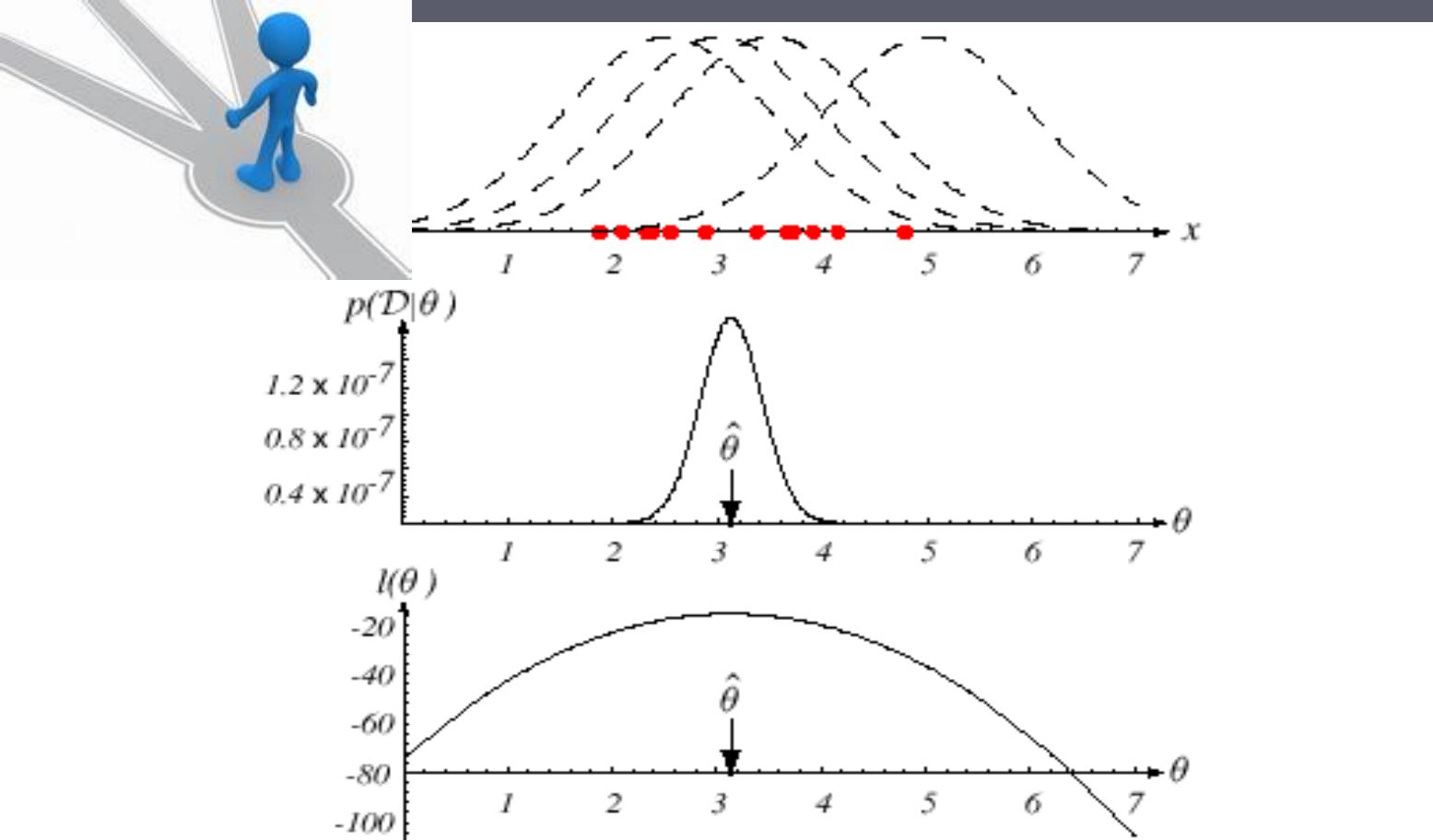


FIGURE 3.1. The top graph shows several training points in one dimension, known or assumed to be drawn from a Gaussian of a particular variance, but unknown mean. Four of the infinite number of candidate source distributions are shown in dashed lines. The middle figure shows the likelihood $p(\mathcal{D}|\theta)$ as a function of the mean. If we had a very large number of training points, this likelihood would be very narrow. The value that maximizes the likelihood is marked $\hat{\theta}$; it also maximizes the logarithm of the likelihood—that is, the log-likelihood $I(\theta)$, shown at the bottom. Note that even though they look similar, the likelihood $p(\mathcal{D}|\theta)$ is shown as a function of θ whereas the conditional density $p(x|\theta)$ is shown as a function of x . Furthermore, as a function of θ , the likelihood $p(\mathcal{D}|\theta)$ is not a probability density function and its area has no significance. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

■ Optimal estimation

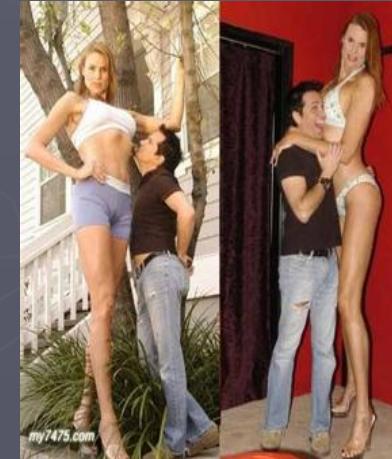
- ▶ Let $\theta = (\theta_1, \theta_2, \dots, \theta_p)^t$ and let ∇_θ be the gradient operator

$$\nabla_\theta = \left[\frac{\partial}{\partial \theta_1}, \frac{\partial}{\partial \theta_2}, \dots, \frac{\partial}{\partial \theta_p} \right]^t$$

- ▶ We define $l(\theta)$ as the log-likelihood function

$$l(\theta) = \ln p(D | \theta)$$

- ▶ New problem statement:
determine θ that maximizes the log-likelihood



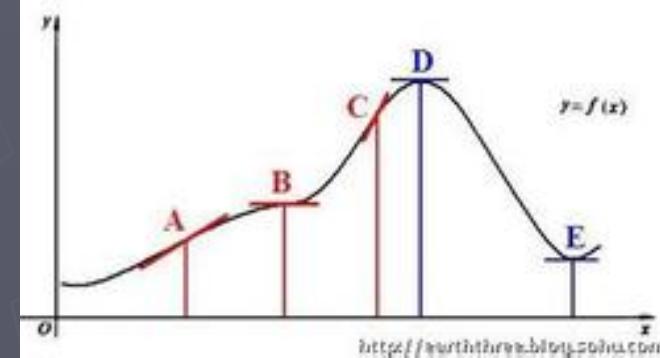
$$\hat{\theta} = \arg \max_{\theta} l(\theta)$$



- Set of necessary conditions for an optimum is:

$$(\nabla_{\theta} l = \sum_{k=1}^{k=n} \nabla_{\theta} \ln P(x_k | \theta))$$

$$\nabla_{\theta} l = 0$$



- Global maximum, local maximum or minimum, inflection point
- For reference: MAP estimators (Max a posteriori)

$$l(\theta)p(\theta)$$

► Example of a specific case 1: unknown μ

- $p(x_i | \mu) \sim N(\mu, \Sigma)$

(Samples are drawn from a multivariate normal population)

$$\ln p(x_k | \mu) = -\frac{1}{2} \ln [(2\pi)^d |\Sigma|] - \frac{1}{2} (x_k - \mu)^t (\Sigma)^{-1} (x_k - \mu)$$

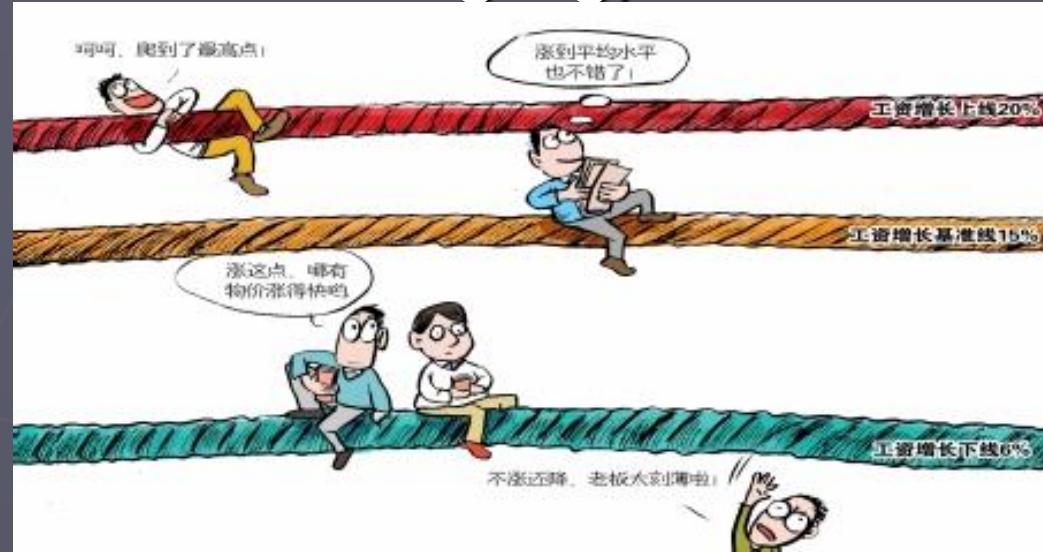
and $\nabla_{\mu} \ln p(x_k | \mu) = (\Sigma)^{-1} (x_k - \mu)$

- The ML estimate for μ must satisfy:

$$\sum_{k=1}^{n_k} \Sigma^{-1} (x_k - \hat{\mu}) = 0$$

- Multiplying by Σ and rearranging, we obtain:

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



Just the arithmetic average of the samples of the training samples!

Conclusion:

If $P(x_k | \omega_j)$ ($j = 1, 2, \dots, c$) is supposed to be Gaussian in a d -dimensional feature space; then we can estimate the vector

$\theta = (\theta_1, \theta_2, \dots, \theta_c)^t$ and perform an optimal classification!

► Example of a specific case 2

- Gaussian Case: *unknown μ and σ*

$$\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$$

$$l = \ln P(x_k | \theta) = -\frac{1}{2} \ln 2\pi\theta_2 - \frac{1}{2\theta_2} (x_k - \theta_1)^2$$

$$\nabla_{\theta} l = \begin{pmatrix} \frac{\partial}{\partial \theta_1} (\ln P(x_k | \theta)) \\ \frac{\partial}{\partial \theta_2} (\ln P(x_k | \theta)) \end{pmatrix} = 0$$

$$\begin{cases} \frac{1}{\theta_2} (x_k - \theta_1) = 0 \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} = 0 \end{cases}$$

Summation:

$$\left\{ \begin{array}{l} \sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} (x_k - \hat{\theta}_1) = 0 \\ - \sum_{k=1}^{k=n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 \end{array} \right. \quad (1)$$

Combining (1) and (2), one obtains:

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



► Bias

- ML estimate for σ^2 is biased

$$E[\hat{\sigma}^2] = E\left[\frac{1}{n}\sum(x_i - \bar{x})^2\right] = \frac{n-1}{n} \cdot \sigma^2 \neq \sigma^2$$

- An elementary unbiased estimator for Σ is:

$$C = \underbrace{\frac{1}{n-1} \sum_{k=1}^{k=n} (x_k - \hat{\mu})(x_k - \hat{\mu})^t}_{\text{Sample covariance matrix}}$$

- ML estimate for Σ is biased

$$\hat{\Sigma} = \frac{n-1}{n} C$$



- Absolutely unbiased, asymptotically unbiased
- Prove ML estimate for σ^2 is biased

$$E[x^2] = D[x] + E[x]^2$$

$$E\left[\sum_{i=1}^n x_i^2\right] = n(\sigma^2 + \mu^2)$$

$$E[\bar{x}^2] = D[\bar{x}] + E(\bar{x})^2 = \frac{1}{n}\sigma^2 + \mu^2$$

$$\begin{aligned} E\left[\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2\right] &= \frac{1}{n} E\left[\sum_{i=1}^n x_i^2 - n\bar{x}^2\right] \\ &= \frac{1}{n} [n(\sigma^2 + \mu^2) - n(\frac{1}{n}\sigma^2 + \mu^2)] = \frac{n-1}{n}\sigma^2 \end{aligned}$$

Chapter 3

Maximum-Likelihood and

Bayesian Parameter Estimation

(3,4,5)

- Bayesian Estimation (BE)
- Bayesian Parameter Estimation: Gaussian Case
- Bayesian Parameter Estimation: General Estimation

3.3 Bayesian Estimation

- ▶ In MLE θ was supposed to be a fixed value
- ▶ In BE θ is a random variable
- ▶ The computation of posterior probabilities $P(\omega_i | x)$ lies at the heart of Bayesian classification
- ▶ Goal: compute $P(\omega_i | x, D)$

Given the sample D , Bayes formula can be written

$$P(\omega_i | x, D) = \frac{p(x | \omega_i, D).P(\omega_i | D)}{\sum_{j=1}^c p(x | \omega_j, D).P(\omega_j | D)}$$

Sample D  likelihood (conditional probability)

 posterior probabilities

- To demonstrate the preceding equation, we use:

$$D = D_1 \cup D_2 \dots \cup D_c \quad x \in D_i \rightarrow x \text{ is } \omega_i$$

D_i has no influence on $p(x | \omega_j, D_j)$ if $i \neq j$

$P(\omega_i) = P(\omega_i | D)$ (Training sample provides this!)

Thus :

$$P(\omega_i | x, D) = \frac{P(x | \omega_i, D_i).P(\omega_i)}{\sum_{j=1}^c P(x | \omega_j, D_j).P(\omega_j)}$$

Further illustration

likelihood (conditional probability)

$$\begin{aligned} p(x | \omega_i, D) &= p(x) \cong p(x | D) \\ &= \int p(x, \theta | D) d\theta \\ &= \int p(x | \theta) p(\theta | D) d\theta \end{aligned}$$

posterior $p(\theta | D)$ Key

Unknown θ and known prior density $p(\theta)$

Description of the above illustration

► Parameter Distribution

- $p(x)$ is unknown, we assume it has a known parametric form $p(x|\theta)$, and value of parameter θ is unknown

- Known prior density $p(\theta)$
- Training data convert $p(\theta)$ to a posterior $p(\theta | D)$

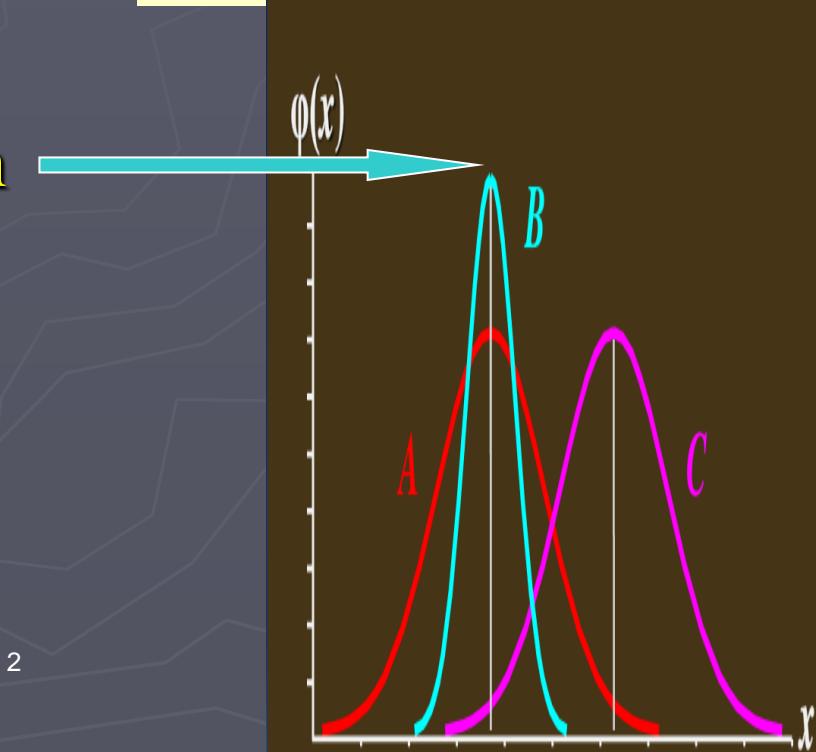
■ Our path:

$$\begin{aligned} p(x | \omega_i, D) &= p(x) \cong p(x | D) \\ &= \int p(x, \theta | D) d\theta \\ &= \int p(x | \theta) p(\theta | D) d\theta \end{aligned}$$

► If $p(\theta | D)$ peaks very sharply about parameter $\hat{\theta}$ and $p(x | \theta)$ is smooth, and the tails of the integral are not important, then

$$p(x | D) = \int p(x | \theta) p(\theta | D) d\theta \rightarrow p(x | D) \approx p(x | \hat{\theta})$$

e.g. the green
curve



3.4 Bayesian Parameter Estimation: Gaussian Case

- ▶ Goal: Estimate θ using the a-posteriori density $P(\theta | D)$
- ▶ The univariate case: $P(\mu | D)$
 μ is the only unknown parameter



$$P(x | \mu) \sim N(\mu, \sigma^2)$$

$$P(\mu) \sim N(\mu_0, \sigma_0^2)$$

(μ_0 and σ_0 are known!)

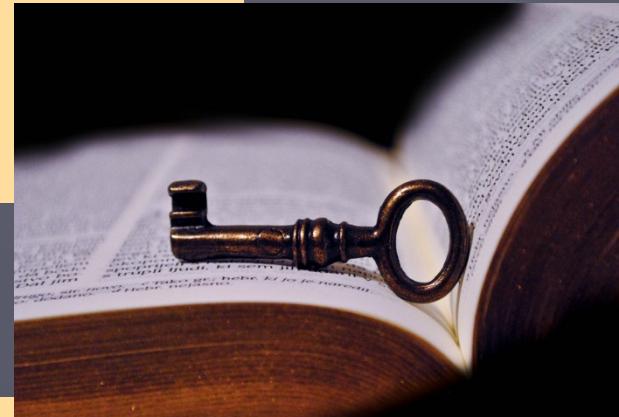


$$P(\mu | D) = \frac{P(D | \mu) \cdot P(\mu)}{\int P(D | \mu) \cdot P(\mu) d\mu} \quad (1)$$

$$= \alpha \prod_{k=1}^{n_k} P(x_k | \mu) \cdot P(\mu)$$

- Reproducing density

$$P(\mu | D) \sim N(\mu_n, \sigma_n^2)$$



(2)

Identifying (1) and (2)
yields:

$$\mu_n = \left(\frac{n\sigma_0^2}{n\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} \cdot \mu_0$$

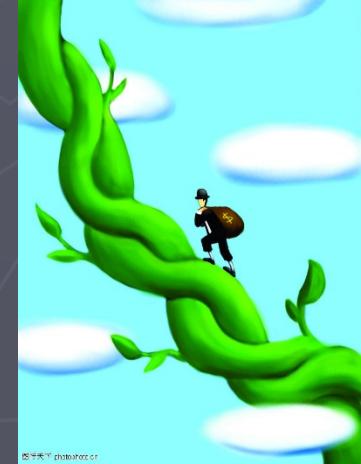
$$\text{and } \sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2} \quad \hat{\mu}_n = \frac{1}{n} (x_1 + \dots + x_n)$$



► Understanding

- μ_n represents our best guess for μ after observing n samples
- σ_n^2 measures our uncertainty about this guess
- Add samples to decrease the uncertainty
- Bayse Learning: as n increase, $p(\mu | D)$ becomes more and more sharply peaked, approaching a Dirac delta function as n approaches infinity

$$\frac{\sigma^2}{\sigma_0^2}$$



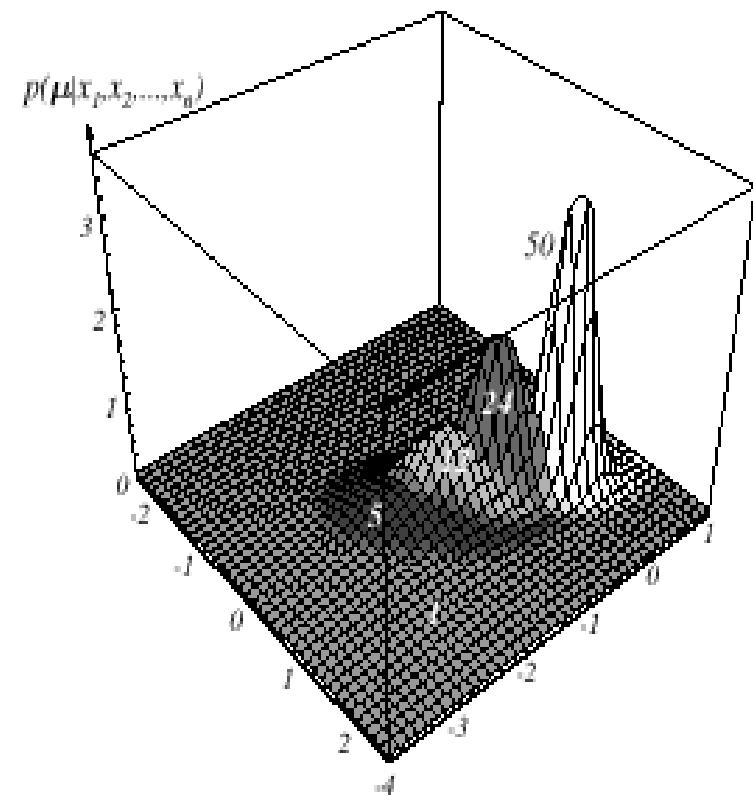
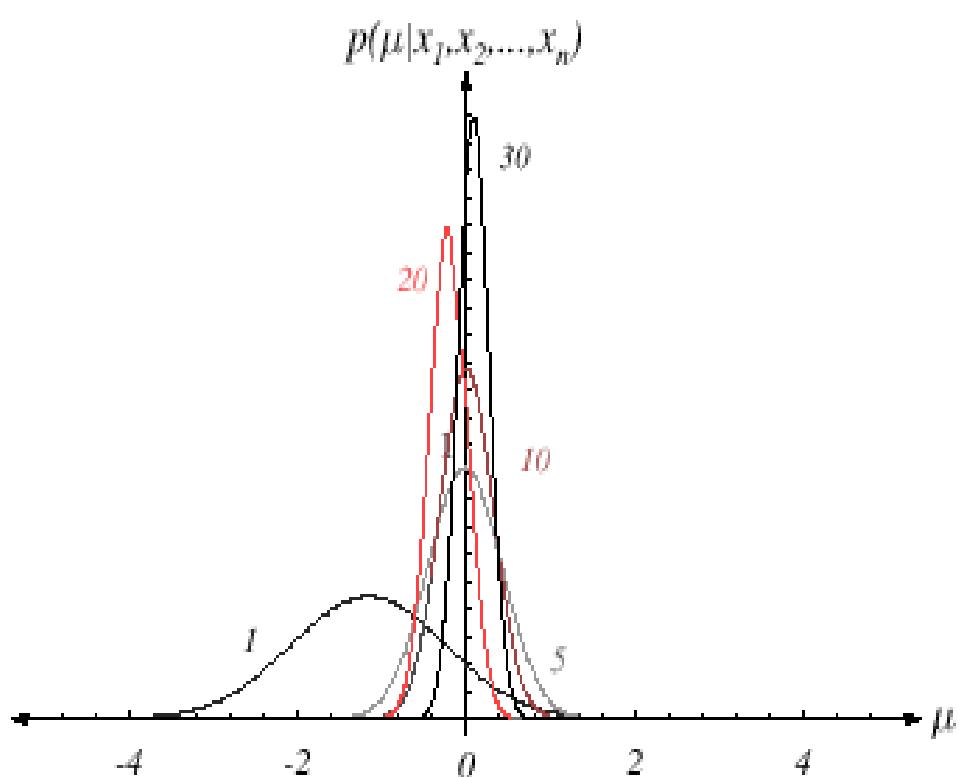


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



► The univariate case $P(x | D)$

- $P(\mu | D)$ computed as above
- $P(x | D)$ remains to be computed!

$P(x | D) = \int P(x | \mu).P(\mu | D) d\mu$ is Gaussian

- It provides:

$$P(x | D) \sim N(\mu_n, \sigma^2 + \sigma_n^2)$$

(Desired class-conditional density $P(x | D_j, \omega_j)$)

Therefore: $P(x | D_j, \omega_j)$ together with $P(\omega_j)$ are known

And using Bayes formula, we obtain the Bayesian classification rule:

$$\underset{\omega_j}{\operatorname{Max}} [P(\omega_j | x, D)] = \underset{\omega_j}{\operatorname{Max}} [P(x | \omega_j, D_j).P(\omega_j)]$$



3.5 Bayesian Parameter Estimation: General Theory

► $P(x | D)$ computation can be applied to any situation in which the unknown density can be parametrized. the basic assumptions are:

- The form of $P(x | \theta)$ is assumed known, but the value of θ is not known exactly
- Our knowledge about θ is assumed to be contained in a known prior density $P(\theta)$
- The rest of our knowledge θ is contained in a set D of n random variables x_1, x_2, \dots, x_n that follows unknown $P(x)$



► The basic problem is:

“Compute the posterior density $P(\theta | D)$ ”

then “Derive $p(x | D) = \int p(x | \theta)p(\theta | D)d\theta$ ”

Using Bayes formula, we have:

$$P(\theta | D) = \frac{P(D | \theta) \cdot P(\theta)}{\int P(D | \theta) \cdot P(\theta) d\theta},$$

And by independence assumption:

$$P(D | \theta) = \prod_{k=1}^{k=n} P(x_k | \theta)$$



► Bayse incremental learning

$$D^n = \{x_1, \dots, x_n\}$$

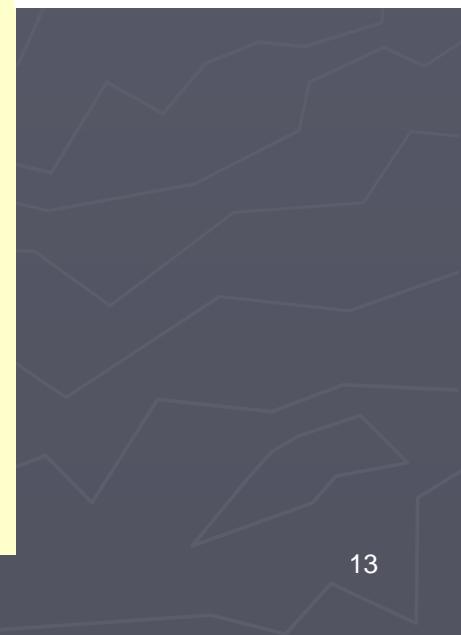
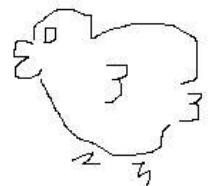
$$p(D^n | \theta) = p(x_n | \theta) p(D^{n-1} | \theta)$$

$$p(\theta | D^n) = \frac{p(D^n | \theta) p(\theta)}{\int p(D^n | \theta) p(\theta) d\theta} = \frac{p(x_n | \theta) p(D^{n-1} | \theta) p(\theta)}{\int p(x_n | \theta) p(D^{n-1} | \theta) p(\theta) d\theta}$$

$$= \frac{p(x_n | \theta) \frac{p(D^{n-1} | \theta) p(\theta)}{p(D^{n-1})}}{\int p(x_n | \theta) \frac{p(D^{n-1} | \theta) p(\theta)}{p(D^{n-1})} d\theta}$$

$$= \frac{p(x_n | \theta) p(\theta | D^{n-1})}{\int p(x_n | \theta) p(\theta | D^{n-1}) d\theta}$$

$$p(\theta | D^0) = p(\theta)$$



Maximum Likelihood vs Bayse Estimation

► Which is better ?

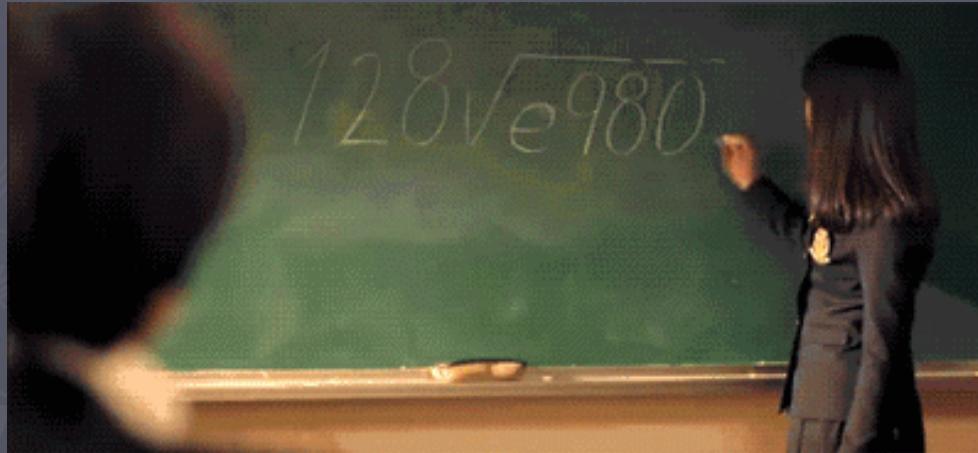
► Maximum Likelihood vs Bayse Estimation

- Computational complexity: ML
- Interpretability : ML
- Confidence in prior information

► Source of classification error

- Bayes Error
- Model Error
- Estimation Error

- Drawbacks of ML estimation
 - ▶ Some observations may be not consistent with the fact



Chapter 3

Maximum-Likelihood and

Bayesian Parameter Estimation

(7,10)

- Problems of Dimensionality
- Computational Complexity
- Hidden Markov Models

3.7 Problems of Dimensionality

- ▶ Features of entries of the data (samples) are statistically independent.
 - ▶ Classification accuracy depends upon the dimensionality and the amount of training data
 - ▶ Case of two classes: the likelihood function is multivariate normal with the same covariance
 - ▶ The two classes have a same prior.

$$P(error) = \frac{1}{\sqrt{2\pi}} \int_{r/2}^{\infty} e^{\frac{-u^2}{2}} du$$

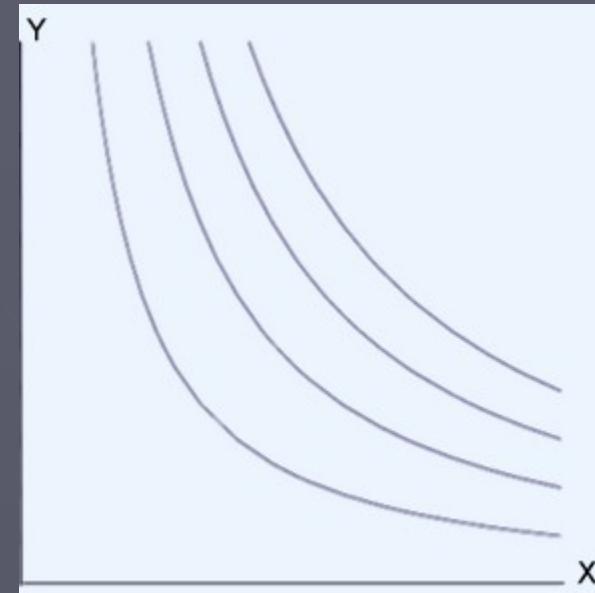
where: $r^2 = (\mu_1 - \mu_2)^t \Sigma^{-1} (\mu_1 - \mu_2)$

$$\lim_{r \rightarrow \infty} P(error) = 0$$

- If features are independent then:

$$\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_d^2)$$

$$r^2 = \sum_{i=1}^{d-1} \left(\frac{\mu_{i1} - \mu_{i2}}{\sigma_i} \right)^2$$



- Most useful features are the ones for which the difference between the means is large relative to the standard deviation
- It has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better performance: this is owing to complex factors.

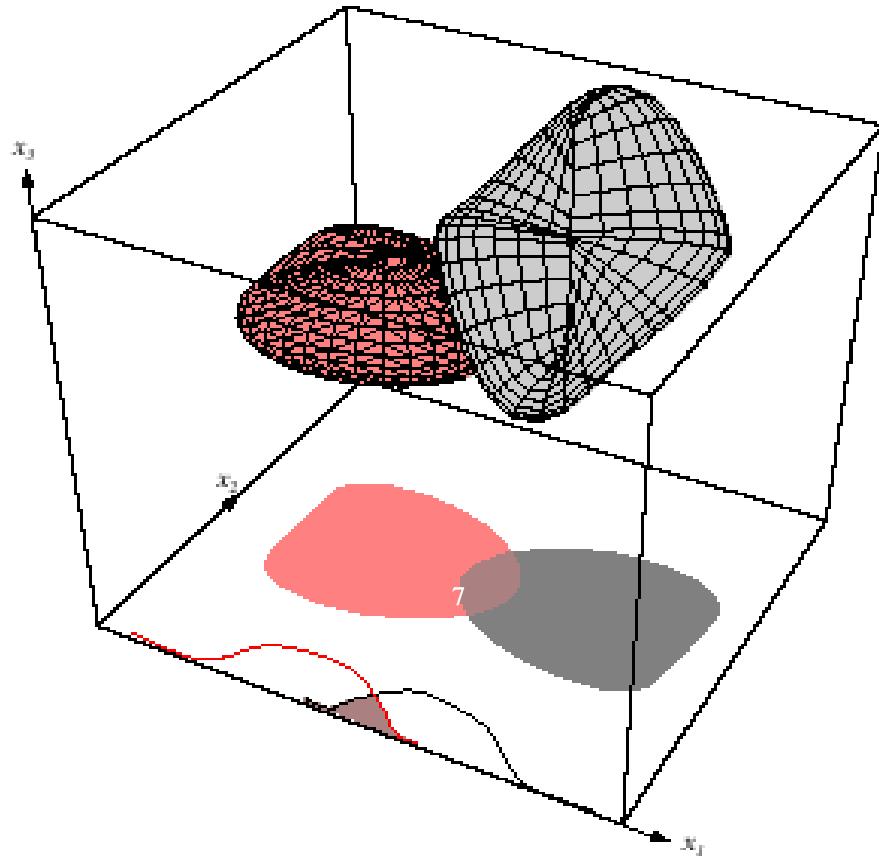
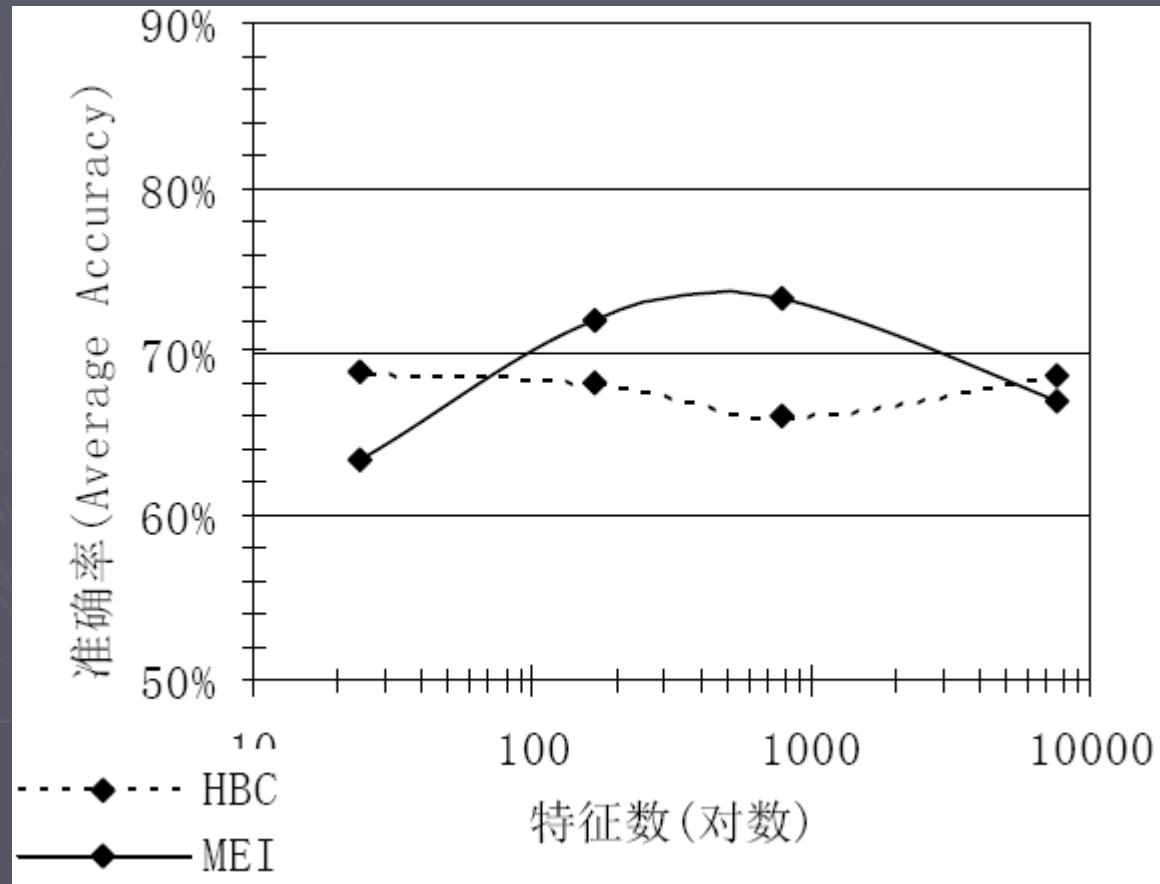
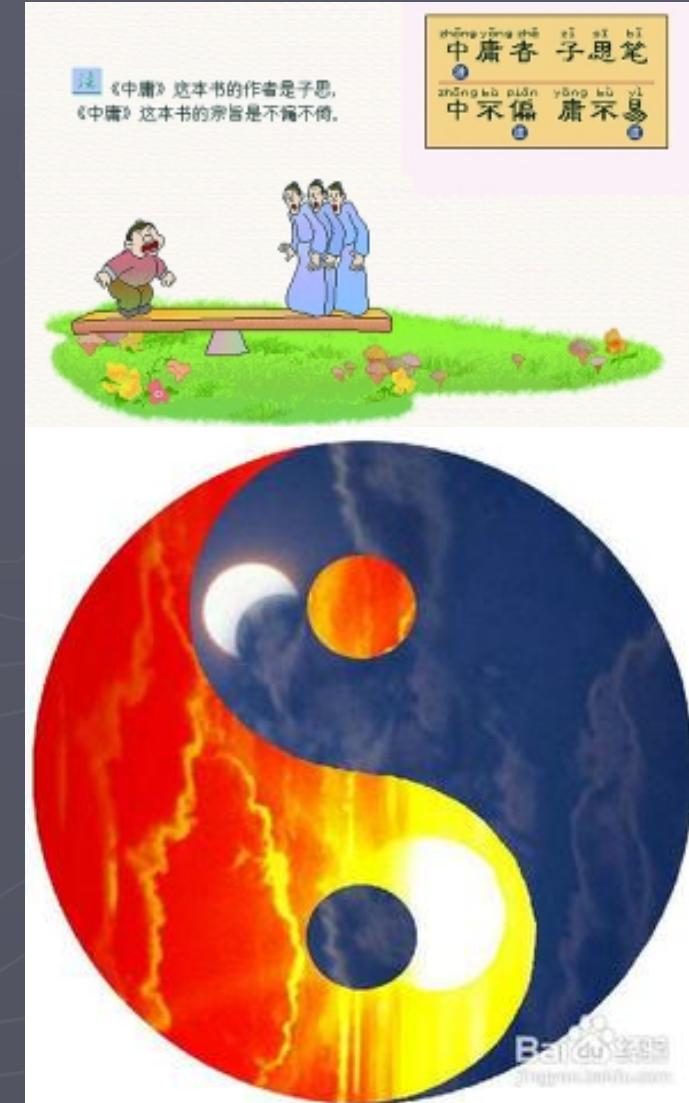


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, and hence greater Bayes error. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

One example of document clustering



Patter Classification chapter 3 part 3



Too many features may be harmful



Way to reduce features: feature selection



"Feature Selection"

学术搜索

找到约 255,000 条结果 (用时0.16秒)

时间不限
2013以来
2012以来
2009以来
自定义范围...

按相关性排序
按日期排序

搜索所有网页
中文网页
简体中文网页

包括专利
 包含引用

 创建快讯

小提示：只搜索中文(简体)结果，可在 学术搜索设置 指定搜索语言

[PDF] [A comparative study on feature selection in text categorization](#)

Y Yang, JO Pedersen - ICML, 1997 - faculty.cs.byu.edu

Abstract This paper is a comparative study of **feature selection** methods in statistical learning of text categorization. The focus is on aggressive dimensionality reduction. Five methods were evaluated, including term selection based on document frequency (DF), information ...
被引用次数: 4066 相关文章 所有 32 个版本 引用 更多▼

[An introduction to variable and feature selection](#)

I Guyon, A Elisseeff - The Journal of Machine Learning Research, 2003 - dl.acm.org

Abstract Variable and **feature selection** have become the focus of much research in areas of application for which datasets with tens or hundreds of thousands of variables are available. These areas include text processing of internet documents, gene expression array ...
被引用次数: 4749 相关文章 所有 119 个版本 引用

[The feature selection problem: Traditional methods and a new algorithm](#)

K Kira, LA Rendell - AAAI, 1992 - aaai.org

For real-world concept learning problems, **feature selection** is important to speed up learning and to improve concept quality. We review and analyze past approaches to **feature selection** and note their strengths and weaknesses. We then introduce and theoretically ...
被引用次数: 928 相关文章 所有 2 个版本 引用

[Floating search methods in feature selection](#)

P Pudil, J Novovičová, J Kittler - Pattern recognition letters, 1994 - Elsevier

Abstract Sequential search methods characterized by a dynamically changing number of features included or eliminated at each step, henceforth "floating" methods, are presented. They are shown to give very good results and to be computationally more effective than



► Computational Complexity

- Our design methodology is affected by the computational difficulty

- “big oh” notation

$f(x) = O(h(x))$ “big oh of $h(x)$ ”

If:

$$\exists(c_0, x_0) \in \Re^2; |f(x)| \leq c_0 |h(x)| \text{ for all } x > x_0$$

$$f(x) = 2+3x+4x^2$$

$$g(x) = x^2$$

$$f(x) = O(x^2)$$

► “big oh” is not unique!

$$f(x) = O(x^2); f(x) = O(x^3); f(x) = O(x^4)$$

► “big theta” notation

$$f(x) = \theta(h(x))$$

If:

$$\exists(x_0, c_1, c_2) \in \Re^3; \forall x > x_0$$

$$0 \leq c_1 h(x) \leq f(x) \leq c_2 h(x)$$



$$f(x) = \theta(x^2) \text{ but } f(x) \neq \theta(x^3)$$

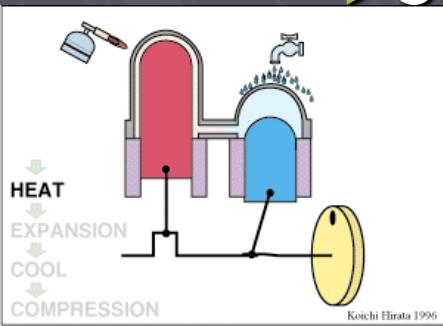
■ Complexity of the ML Estimation

- ▶ Gaussian priors in d dimensions classifier with n training samples for each of c classes
- ▶ For each category, we have to compute the discriminant function

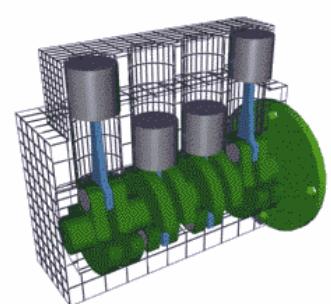
$$g(x) = -\frac{1}{2} \left(x - \hat{\mu} \right)^t \Sigma^{-1} \left(x - \hat{\mu} \right) - \underbrace{\frac{d}{2} \ln 2\pi}_{O(d^2)} - \underbrace{\frac{1}{2} \ln |\hat{\Sigma}| + \ln P(\omega)}_{O(n)}$$

Total = $O(d^2 \cdot n)$

- ▶ Total for c classes = $O(cd^2 \cdot n) \approx O(d^2 \cdot n)$
- ▶ Cost increase when d and n are large!



Patter Classification chapter 3 part 3



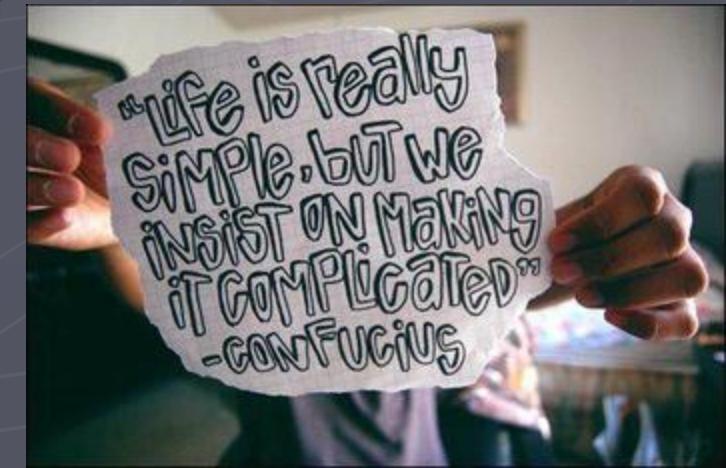
► Overfitting

- Samples are inadequate
 - ▶ Reduce dimensionality (select a subset of features or combine features)
 - ▶ All c classes share the same covariance matrix (**can better evaluate the covariance**)
 - ▶ Look for a better estimate for covariance matrix

Pseudo-Bayesian estimation:

$$\lambda \Sigma_0 + (1 - \lambda) \hat{\Sigma}$$

- ▶ An extreme case: statistical independence
- How to get better performance if statistical dependence
 - ▶ Sufficient data
 - ▶ Prevent overfitting



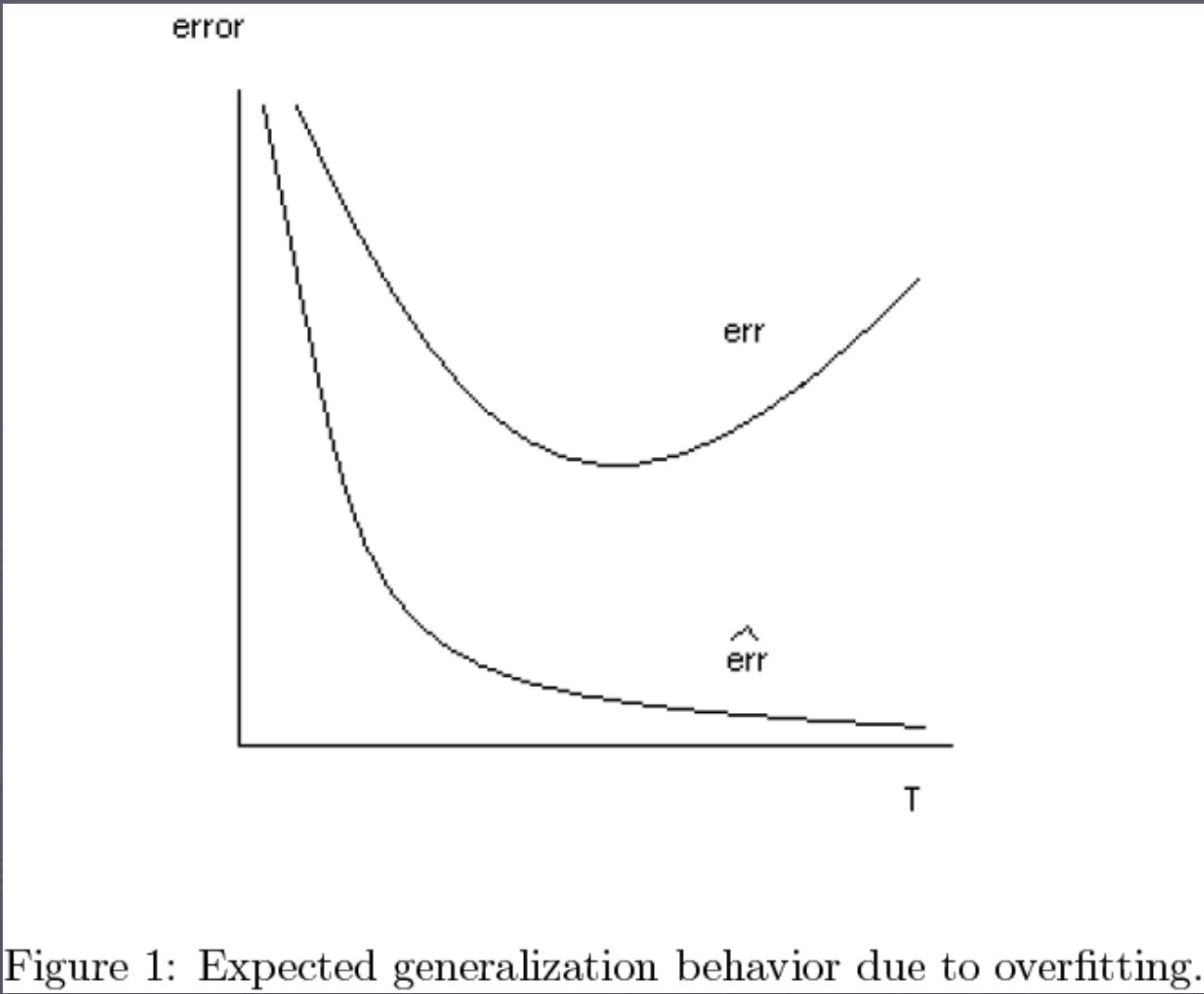
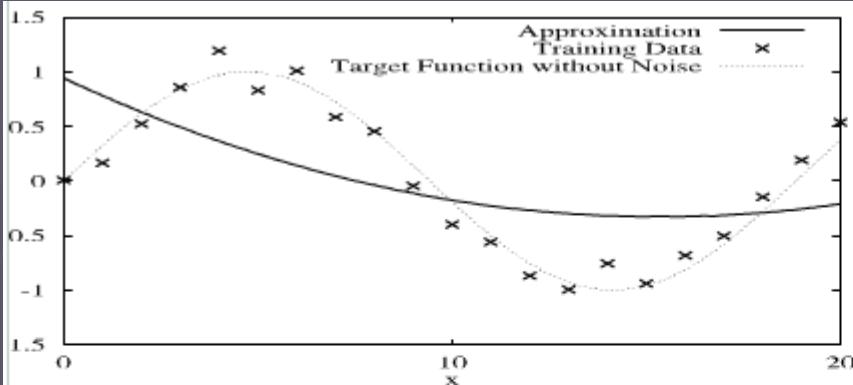
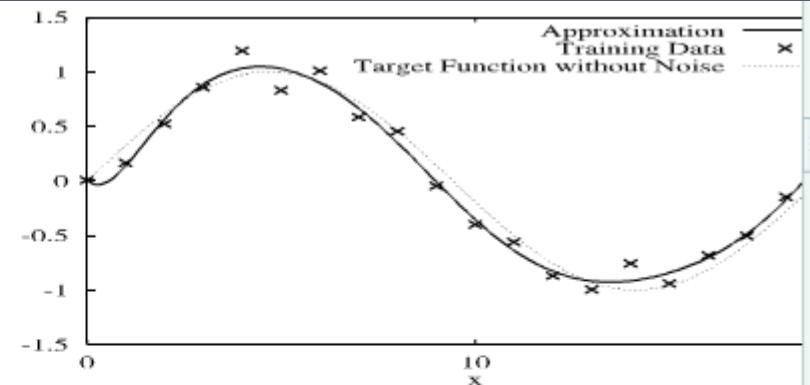


Figure 1: Expected generalization behavior due to overfitting.

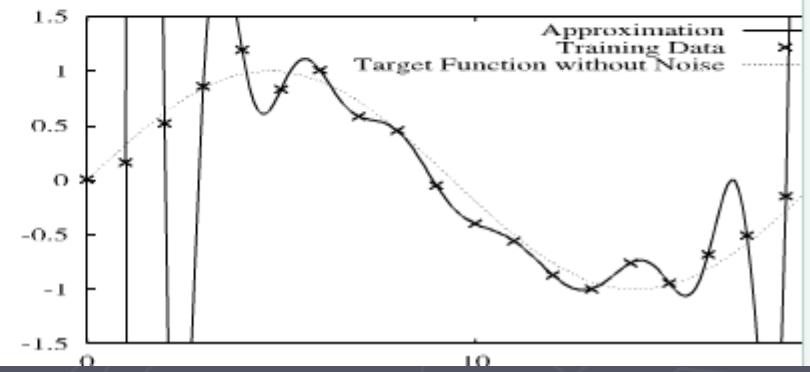
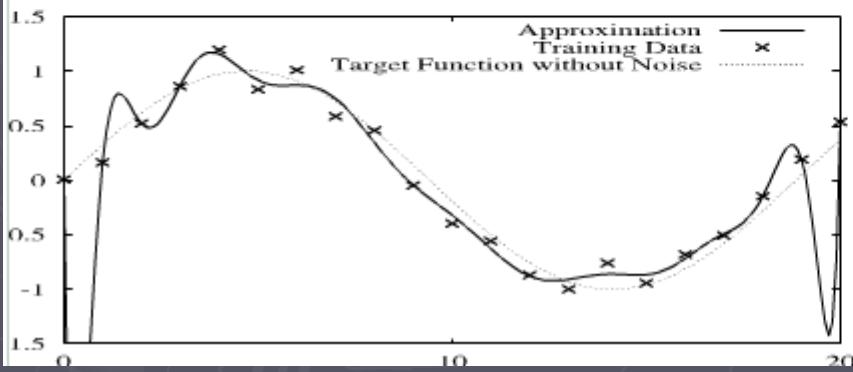
Possible overfitting in neural networks



Order 2

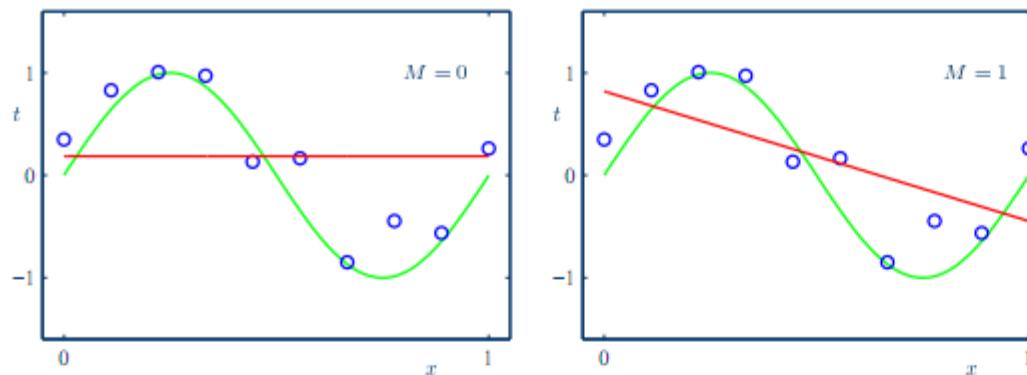


Order 10

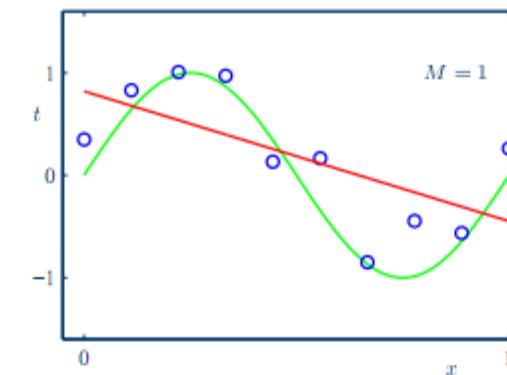


► Another example: Page 93 (In Chinese edition)

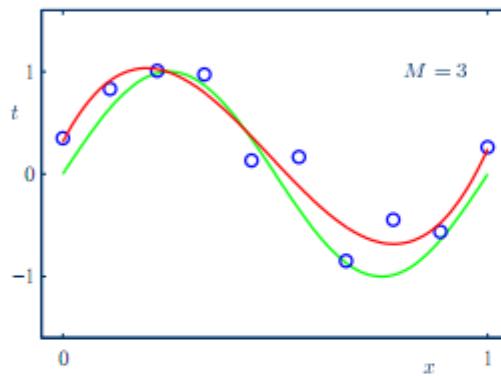
Classifiers with proper low complexity is favored



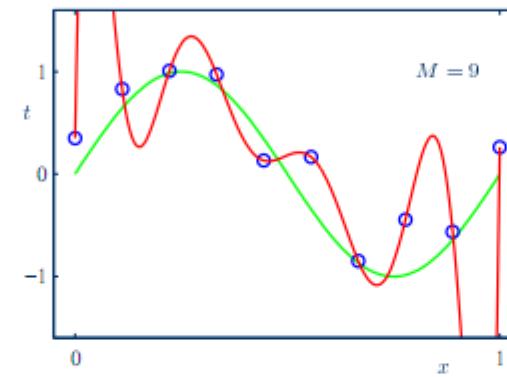
(a) 0'th order polynomial



(b) 1'st order polynomial



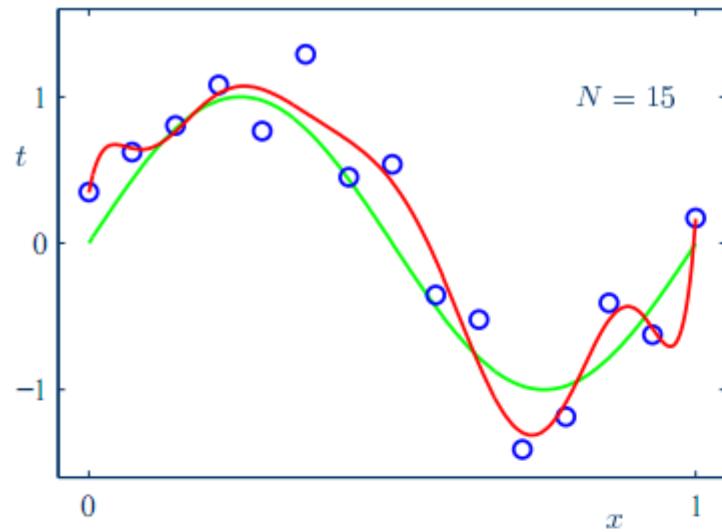
(c) 3'rd order polynomial



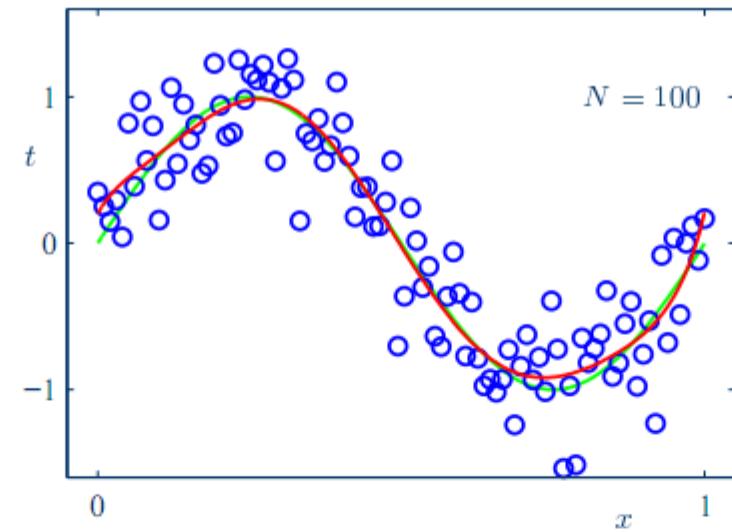
(d) 9'th order polynomial

Figure 19: Polynomial curve fitting: plots of polynomials having various orders, shown as red curves, fitted to the set of 10 sample points.

More training samples are better



(a) 15 sample points



(b) 100 sample points

Figure 20: Polynomial curve fitting: plots of 9'th order polynomials fitted to 15 and 100 sample points.

3.10 Hidden Markov Models

► Goal: make a sequence of decisions

- A process that unfold in time, states at time t are influenced by a state at time t-1
- Applications: speech recognition, gesture recognition, parts of speech tagging and DNA sequencing



GaiTu.com



Patter Classification chapter 3 part 3

L7 L7.
. YOBBBBBBBB;
. BBBHZMMHOMMBB.
iBFMSSGu:,,:r7..LB.
BBqkFqq 7B
:BBFMkkOU .7.. ::B
BM jkSSOY ;MY. 2BBr
NBi:7UqPiirZB. .UFB
iFBBEX7 . ZM
PBMq, viiz8
:i. MBBEUBMBBBE.
GBOMB5.rBSBv MBBD
rB. 7BBM8BBiBr
,F1::r5F

► First-order Markov models

- A sequence of states at successive times
$$\omega^T = \{\omega(1), \omega(2), \omega(3), \dots, \omega(T)\}$$
We might have $\omega^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$
- The system can revisit a state at different steps and not every state need to be visited
- Our productions of any sequence is described by the transition probabilities-time independent



$$P(\omega_j(t+1) | \omega_i(t)) = a_{ij}$$

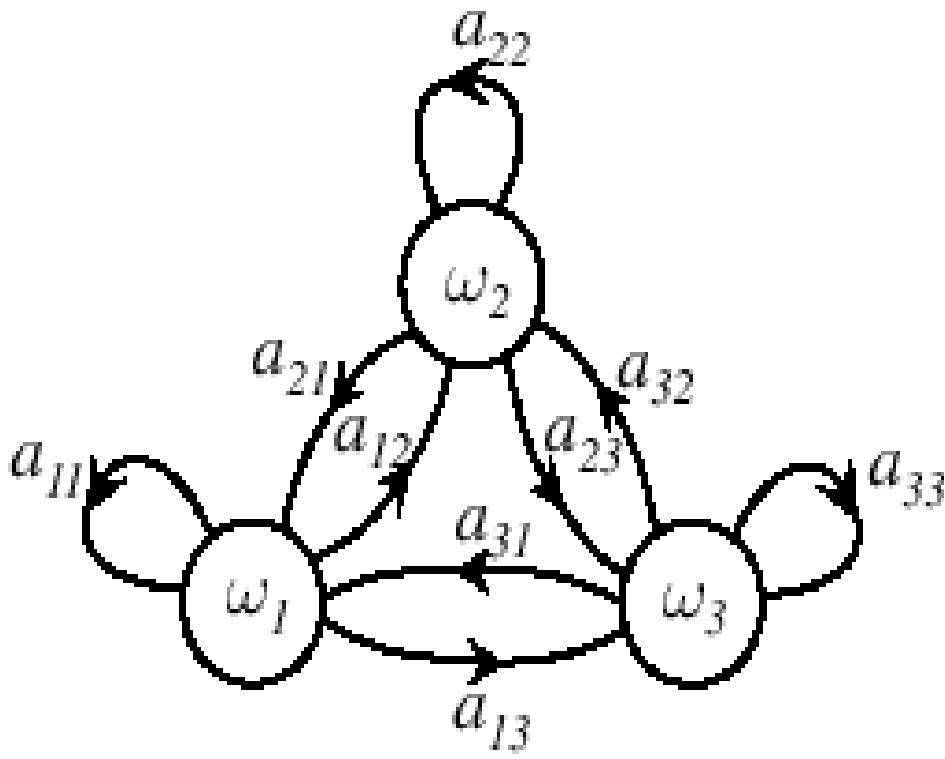


FIGURE 3.8. The discrete states, ω_i , in a basic Markov model are represented by nodes, and the transition probabilities, a_{ij} , are represented by links. In a first-order discrete-time Markov model, at any step t the full system is in a particular state $\omega(t)$. The state at step $t + 1$ is a random function that depends solely on the state at step t and the transition probabilities. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- A model $\theta = (a_{ij}, \omega^\top)$
 $\omega^\top = \omega^6 = \{\omega_1, \omega_4, \omega_2, \omega_2, \omega_1, \omega_4\}$
 $P(\omega^\top | \theta) = a_{14} \cdot a_{42} \cdot a_{22} \cdot a_{21} \cdot a_{14} \cdot$

$$P(\omega(1) = \omega_i)$$

- First-order Markov Model: the probability at t+1 depends only on the states at t
- Example: speech recognition
 “production of spoken words”
 Production of the word: “pattern” represented by phonemes

/p/ /a/ /tt/ /er/ /n/ // (// = silent state)

Transitions from /p/ to /a/, /a/ to /tt/, /tt/ to er/, /er/ to /n/ and /n/ to a silent state



Patter Classification chapter 3 part 3



► First-Order Hidden Markov Models (HMM)

- A state $\omega(t)$ emits some visible symbol $v(t)$, the sequence of such visible symbol is :
 $V^T = \{v(1), v(2), v(3), \dots, v(T)\}$
- In any state $\omega_j(t)$ we have a probability of emitting a particular visible state $v_k(t)$, ω_j are unobservable, such a full model is HMM
- In HMM a_{ij} is the transition probabilities among hidden states and b_{jk} is the probability of the emission of a visible state.

$$a_{ij} = P(\omega_j(t+1) | \omega_i(t))$$

$$\sum a_{ij} = 1 \text{ for all } i$$

$$b_{jk} = P(v_k(t) | \omega_j(t)).$$

$$\sum b_{jk} = 1 \text{ for all } j$$



张学良将军在一次接受记者采访的时候,讲起他小时候学英语的一段趣事:我父亲很想给我请英文教师,英文教师是外交署一个英文科长,这个人我很想念他,他是香港新约书院的。他是广东人,说广东国语。我跟你说个笑话,nine,就是九,他说九(狗),我听说是狗,他说九,我当说狗。那时候

► Three problems are associated with HMM

- The evaluation problem

a_{ij}, b_{jk} → probability of \mathcal{V}^T

- The decoding problem

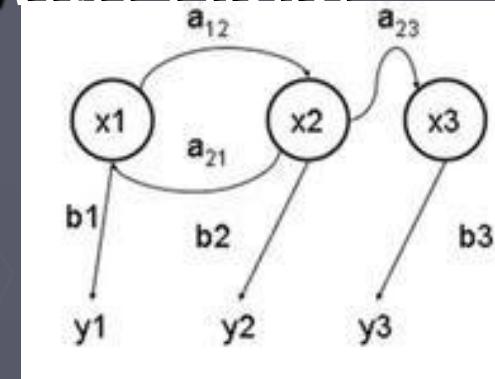
\mathcal{V}^T → ω^T (analogy: listen and write)

P 108

- The learning problem

\mathcal{V}^T → a_{ij}, b_{jk}

1. 一只狗 (九) 在狂吠
2. 那儿有九 (狗) 个人

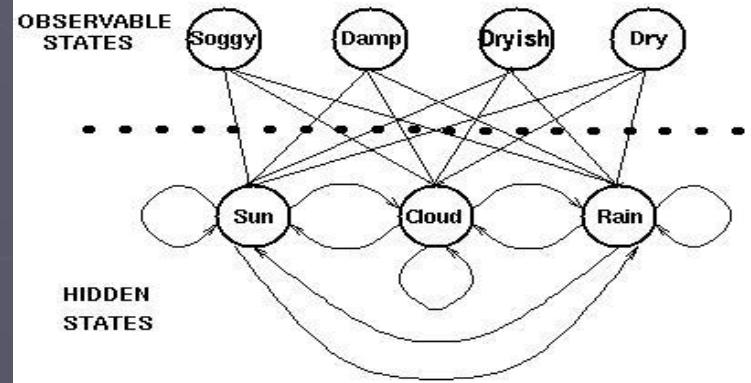


Patter Classification chart

► The evaluation problem

It is the probability that the model produces a sequence V^T of visible states. It is:

$$P(V^T) = \sum_{r=1}^{r_{max}} P(V^T | \omega_r^T) P(\omega_r^T)$$



where each r indexes a particular sequence

$$\omega_r^T = \{\omega_r(1), \omega_r(2), \dots, \omega_r(T)\}$$

hidden states.

$$(1) \quad P(V^T | \omega_r^T) = \prod_{t=1}^{t=T} P(v(t) | \omega_r(t))$$

$$(2) \quad P(\omega_r^T) = \prod_{t=1}^{t=T} P(\omega_r(t) | \omega_r(t-1))$$

Using equations (1) and (2), we can write:

$$P(V^T) = \sum_{r=1}^{r_{\max}} \prod_{t=1}^{t=T} P(v(t) | \omega_r(t)) P(\omega_r(t) | \omega_r(t-1))$$

Interpretation: The probability that we observe the particular sequence of T visible states V^T is equal to the sum over all r_{\max} possible sequences of hidden states of the conditional probability that the system has made a particular transition multiplied by the probability that it then emitted the visible symbol in our target sequence.

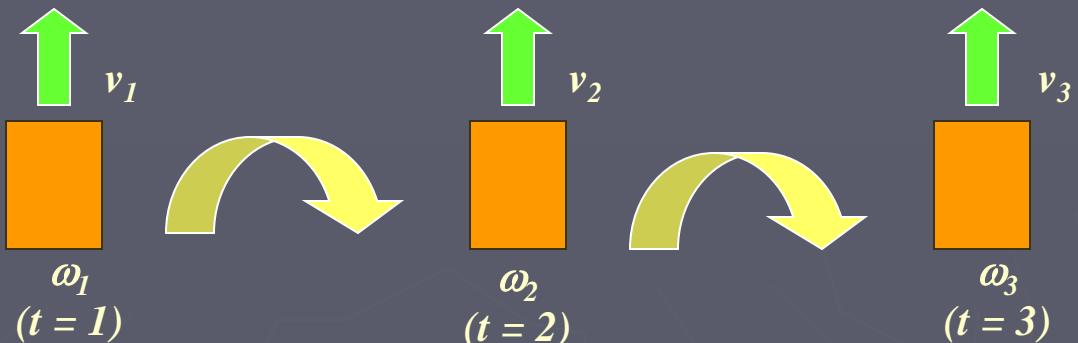
Example: Let $\omega_1, \omega_2, \omega_3$ be the hidden states; v_1, v_2, v_3 be the visible states

and $V^3 = \{v_1, v_2, v_3\}$ is the sequence of visible states

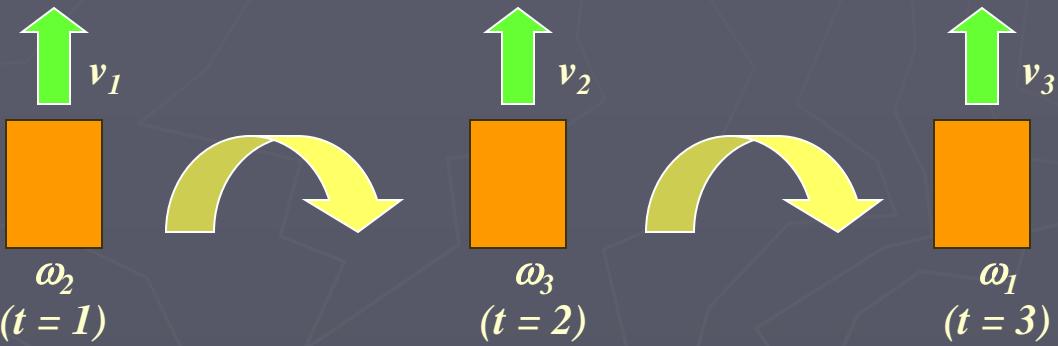
$$P(\{v_1, v_2, v_3\}) = P(\omega_1).P(v_1 / \omega_1).P(\omega_2 / \omega_1).P(v_2 / \omega_2).P(\omega_3 / \omega_2).P(v_3 / \omega_3)$$

+ ... + (possible terms in the sum= all possible $(3^3= 27)$ cases !)

First case:



Second case :



$$P(\{v_1, v_2, v_3\}) = P(\omega_1).P(v_1 / \omega_1).P(\omega_2 / \omega_1).P(v_2 / \omega_2).P(\omega_3 / \omega_2).P(v_3 / \omega_3) +$$

$$P(\omega_2).P(v_1 / \omega_2).P(\omega_3 / \omega_2).P(v_2 / \omega_3).P(\omega_1 / \omega_3).P(v_3 / \omega_1) + \dots +$$

Therefore:

$$P(\{v_1, v_2, v_3\}) = \sum_{\substack{\text{possible sequence} \\ r \text{ of hidden states}}} \prod_{t=1}^{t=3} P(v(t) | \omega_r(t)).P(\omega_r(t) | \omega_r(t-1))$$

- HMM Forward algorithm



$$\alpha_j(t) = \begin{cases} 0 & t = 0 \text{ and } j \neq \text{initial state} \\ 1 & t = 0 \text{ and } j = \text{initial state} \\ [\sum_i \alpha_i(t-1) a_{ij}] b_{jk}(v(t)) & \text{otherwise} \end{cases}$$

$b_{jk}(v(t))$: emission probability b_{jk} selected by the visible state at time t with visible state $v(t)$.

$\alpha_j(t)$: probability that HMM is in hidden state ω_j at step t having generated the first t elements of V

- HMM Forward algorithm

Initialize $t=0$, a_{ij} , b_{jk} , V^T , $\alpha_j(0)$
for $t=t+1$

$$\alpha_j(t) \leftarrow b_{jk} v(t) \sum_{i=1}^c \alpha_i(t-1) a_{ij}$$

Until $t=T$

return $P(V^T) \leftarrow \alpha_0(T)$ for the final state
end

- A left-to-right HMM

► The decoding problem (optimal state sequence)

Given a sequence of visible states V^T , the decoding problem is to find the most probable sequence of hidden states.

This problem can be expressed mathematically as:
find the single “best” state sequence (hidden states)

$\hat{\omega}(1), \hat{\omega}(2), \dots, \hat{\omega}(T)$ such that :

$$\hat{\omega}(1), \hat{\omega}(2), \dots, \hat{\omega}(T) = \arg \max_{\omega(1), \omega(2), \dots, \omega(T)} P[\omega(1), \omega(2), \dots, \omega(T), v(1), v(2), \dots, V(T)] / \lambda$$

Note that the summation disappeared, since we want to find
Only one unique best case !

Where: $\lambda = [\pi, A, B]$

$\pi = P(\omega(1) = \omega)$ (*initial state probability*)

$A = a_{ij} = P(\omega(t+1) = j / \omega(t) = i)$

$B = b_{jk} = P(v(t) = k / \omega(t) = j)$

In the preceding example, this computation corresponds to the selection of the best path amongst:

$\{\omega_1(t = 1), \omega_2(t = 2), \omega_3(t = 3)\}, \{\omega_2(t = 1), \omega_3(t = 2), \omega_1(t = 3)\}$

$\{\omega_3(t = 1), \omega_1(t = 2), \omega_2(t = 3)\}, \{\omega_3(t = 1), \omega_2(t = 2), \omega_1(t = 3)\}$

$\{\omega_2(t = 1), \omega_1(t = 2), \omega_3(t = 3)\}.....$

► HMM decoding algorithm

begin initialize Path $\leftarrow \{\}\right.$, t $\leftarrow 0$

for t $\leftarrow t + 1$

j $\leftarrow 1$

for j $\leftarrow j + 1$

$$\alpha_j(t) \leftarrow b_{jk} v(t) \sum_{i=1}^c \alpha_i(t-1) a_{ij}$$

until j = c

j' $\leftarrow \arg \max_j \alpha_j$

Append $\omega_{j'}$ to Path

until t = T

return Path

end

► The learning problem (parameter estimation)

This third problem consists of determining a method to adjust the model parameters $\lambda = [\pi, A, B]$ to satisfy a certain optimization criterion. We need to find the best model

$$\hat{\lambda} = [\hat{\pi}, \hat{A}, \hat{B}]$$

Such that to maximize the probability of the observation sequence:

$$\underset{\lambda}{\text{Max}} P(V^T / \lambda)$$

We use an iterative procedure such as Baum-Welch or Gradient to find this local optimum

► Parameter Updates: Forward-Backward Algorithm

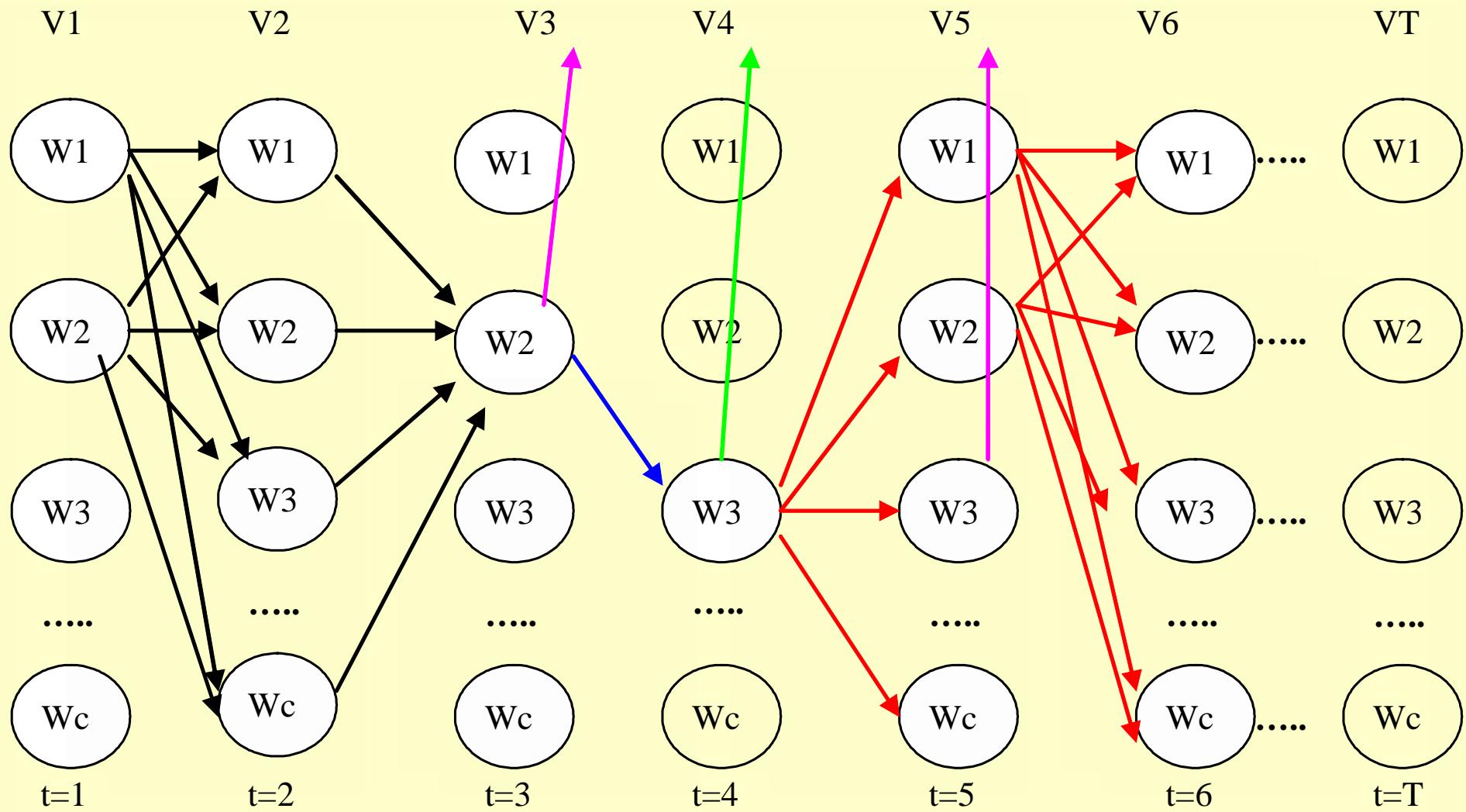
$$\beta_i(t) = \begin{cases} 0 & t = T \text{ and } \omega_i(t) \neq \omega_0 \\ 1 & t = T \text{ and } \omega_i(t) = \omega_0 \\ [\sum_j \beta_j(t+1) a_{ij}] b_{jk} v(t+1) & \text{otherwise} \end{cases}$$

$$\gamma_{ij}(t) = \frac{\alpha_i(t-1) a_{ij} b_{jk} \beta_j(t)}{P(V^T | \Theta)}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \gamma_{ij}(t)}{\sum_{t=1}^T \sum_k \gamma_{ik}(t)}$$

$$\hat{b}_{jk} = \frac{\sum_{\substack{t=1 \\ v(t)=v_k}}^T \sum_l \gamma_{jl}(t)}{\sum_{t=1}^T \sum_l \gamma_{jl}(t)}$$

- $\alpha_i(t) = P(\text{model generates visible sequence up to step } t \text{ given hidden state } \omega_i(t))$
- $\beta_i(t) = P(\text{model will generate the sequence from } t+1 \text{ to } T \text{ given } \omega_i(t))$
- $\gamma_{ij}(t)$ is the probability from $\omega_i(t-1)$ to $\omega_j(t)$



$$\gamma_{23}(4) = \frac{\alpha_2(3)a_{23}b_{34}\beta_3(4)}{p(V^T | \theta)}$$

► Parameters Learning Algorithm

Begin initialize

a_{ij} , b_{jk} , training sequence V^T ,
convergence criterion (cc), $z=0$

Do $z=z+1$

compute $\hat{a}(z)$ from $a(z-1)$ and $b(z-1)$

compute $\hat{b}(z)$ from $a(z-1)$ and $b(z-1)$

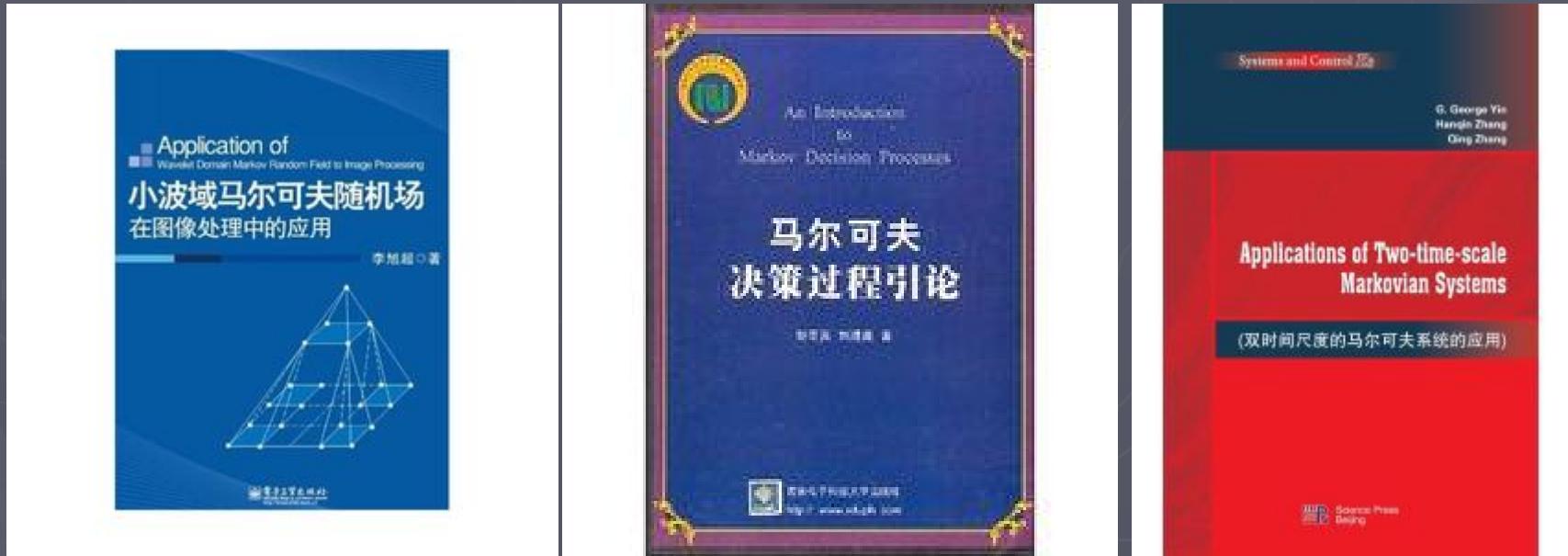
$$a_{ij}(z) = \hat{a}_{ij}(z-1)$$

$$b_{jk}(z) = \hat{b}_{jk}(z-1)$$

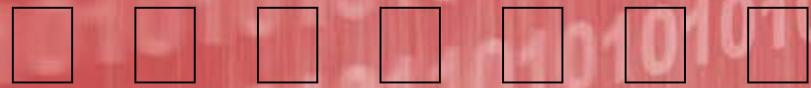
Until $\max\{a_{ij}(z) - a_{ij}(z-1), b_{jk}(z) - b_{jk}(z-1)\} < cc$

Return $a_{ij}=a_{ij}(z)$; $b_{jk}=b_{jk}(z)$

End



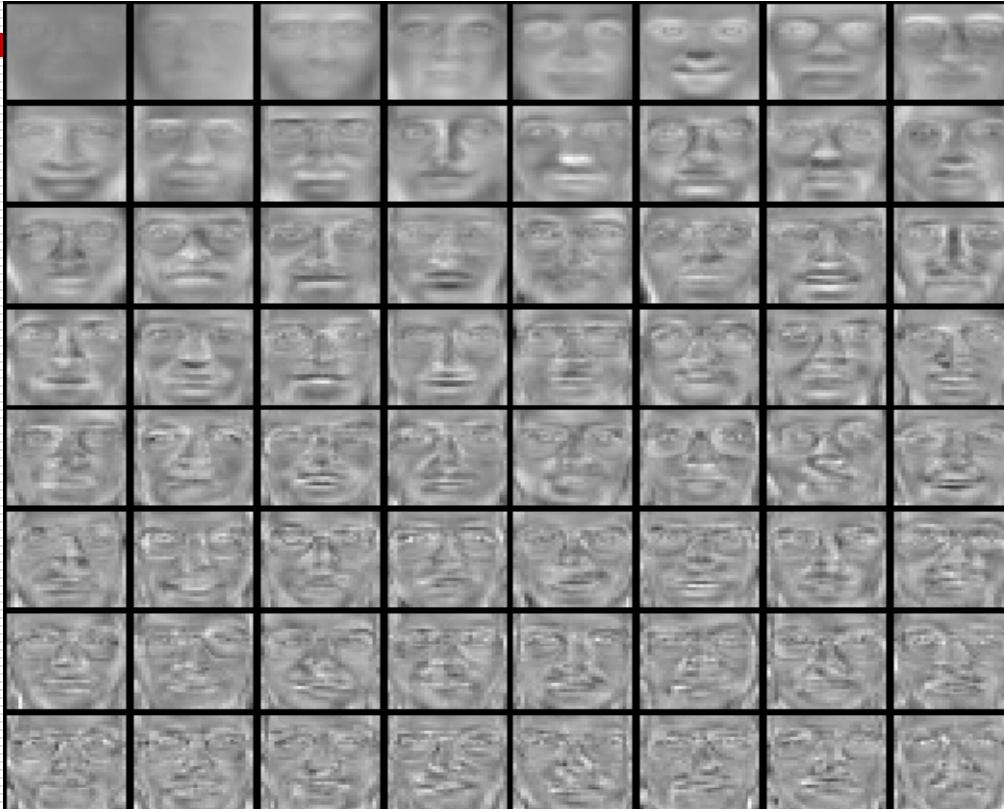
Patter Classification chapter 3 part 3



- Principal Component Analysis (PCA)
- 主成分分析(Principal Component Analysis, 简称PCA)是一种常用的基于变量协方差矩阵对信息进行处理、压缩和抽提的有效方法。



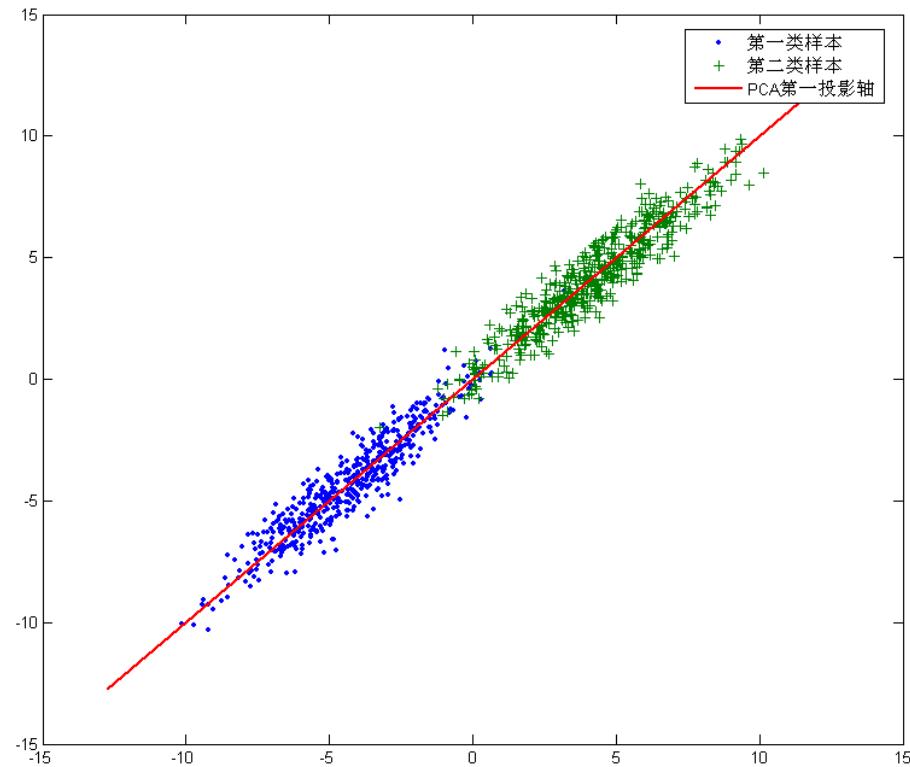
Face representation and recognition methods



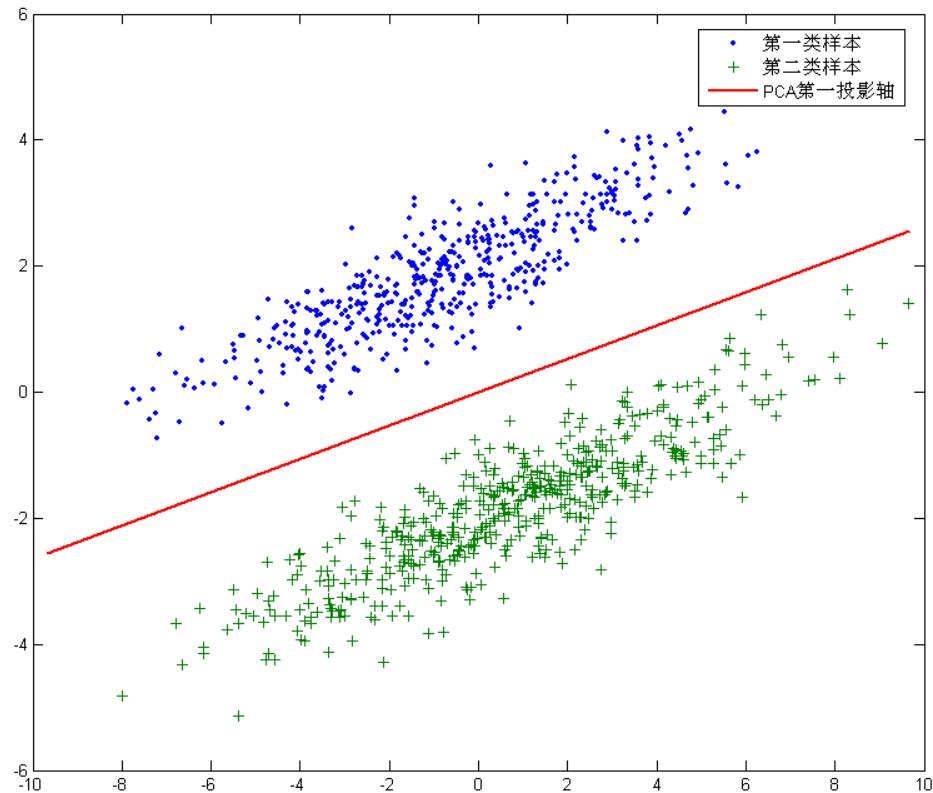
eigenfaces

As in face recognition applications the transform axes used in PCA also look like “faces”, they are referred to as “eigenfaces”!

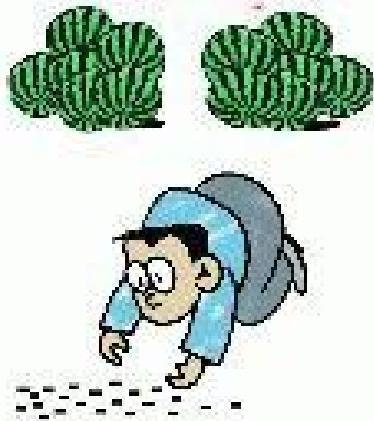
PCA is useful for classification



PCA might perform badly in classification



动机



许多系统是多要素的复杂系统，多变量问题是经常会遇到的。变量太多，无疑会增加分析问题的难度与复杂性，而且在许多实际问题中，多个变量之间是具有一定的相关关系的。

因此，人们会很自然地想到，能否在相关分析的基础上，用较少的新变量代替原来较多的旧变量，而且使这些较少的新变量尽可能多地保留原来变量所反映的信息？



事实上，这种想法是可以实现的，主分量分析方法就是综合处理这种问题的一种强有力的工具。

主分量分析是把原来多个变量划为少数几个综合指标的一种统计分析方法。

从数学角度来看，这是一种降维处理技术。

藏到书架

↓ 下载文档

<

3

/89

>



-

+

全屏

成分分析 优5 心 4 评论 0 加入豆单 △ 举报 腿 手机看分享: ☆ 微博 人人网 邮箱 +

raojun0035

分享于2014-12-25 10:35

详

1 主成分分析的基本思想

一项十分著名的工作是美国的统计学家斯通(stone)在1947年关于国民经济的研究。他曾利用美国1929—1938年各年的数据，得到了17个反映国民收入与支出的变量要素，例如雇主补贴、消费资料和生产资料、纯公共支出、净增库存、股息、利息外贸平衡等等。

在进行主成分分析后，竟以97.4%的精度，用三新变量就取代了17个变量。根据经济学知识，斯通给这三个新变量分别命名为总收入F1、总收入变化率F2和经济发展或衰退的趋势F3。

相关文档

Office 办公软件的应用

专题

PPT模板素材精选专题

专题

P 优 主成分分析new

热度:

P 优 [统计学]多元统计分析(何晓群 中科大)

热度:

P 主成分分析

热度:

P 主成分分析ppt

热度:

A 主成分分析PCA

热度:

W 优 汽车企业竞争力评价方法 (主成分分析)

热度:

P 优 数学建模之主成分分析法

热度:

P 11.主成分分析

热度:

) 主成分分析 优

406 4 0 加入豆单 举报 手机看

分享: +

主成分分析: 将原来具有相关关系的多个指标简化为少数几个新的综合指标的多元统计方法。

主成分: 由原始指标综合形成的几个新指标。依据主成分所含信息量的大小成为第一主成分, 第二主成分等等。

主成分与原始变量之间的关系:

- (1) 主成分保留了原始变量绝大多数信息。
- (2) 主成分的个数大大少于原始变量的数目。
- (3) 各个主成分之间互不相关。
- (4) 每个主成分都是原始变量的线性组合。



raojun0035

分享于2014-12-25 10:35

相关文档

Office 办公软件的应用
专题

PPT模板素材精选专题
专题

P **优** 主成分分析new
热度:

P **优** [统计学]多元统计分析(何晓群)
热度:

P 主成分分析
热度:

P 主成分分析ppt
热度:

P 主成分分析PCA
热度:

W **优** 汽车企业竞争力评价方法 (三)
热度:

P **优** 数学建模之主成分分析法
热度:

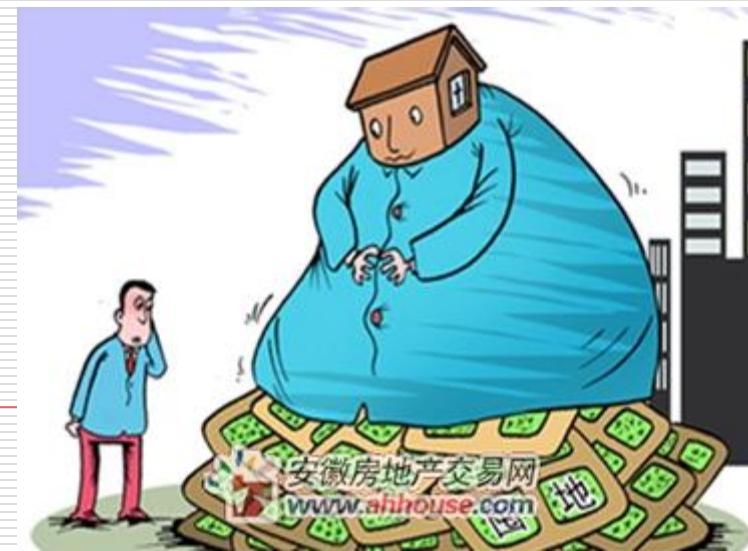
P 11.主成分分析
热度:

申请国外研究生条件

主成分概念首先由 Karl Parson 在 1901 年首先提出，当时只是对非随机变量来讨论的。1933 年 Hotelling 将这个概念推广到随机变量，作了进一步发展。把从混合信号中求出主分量（能量最大的成份）的方法称为主分量分析（PCA），而次分量（Minor Components, MCs）与主分量（Principal Components, PCs）相对，它是混合信号中能量最小的成分，被认为是不重要的或是噪声有关的信号，把确定次分量的方法称为次分量分析（MCA）。

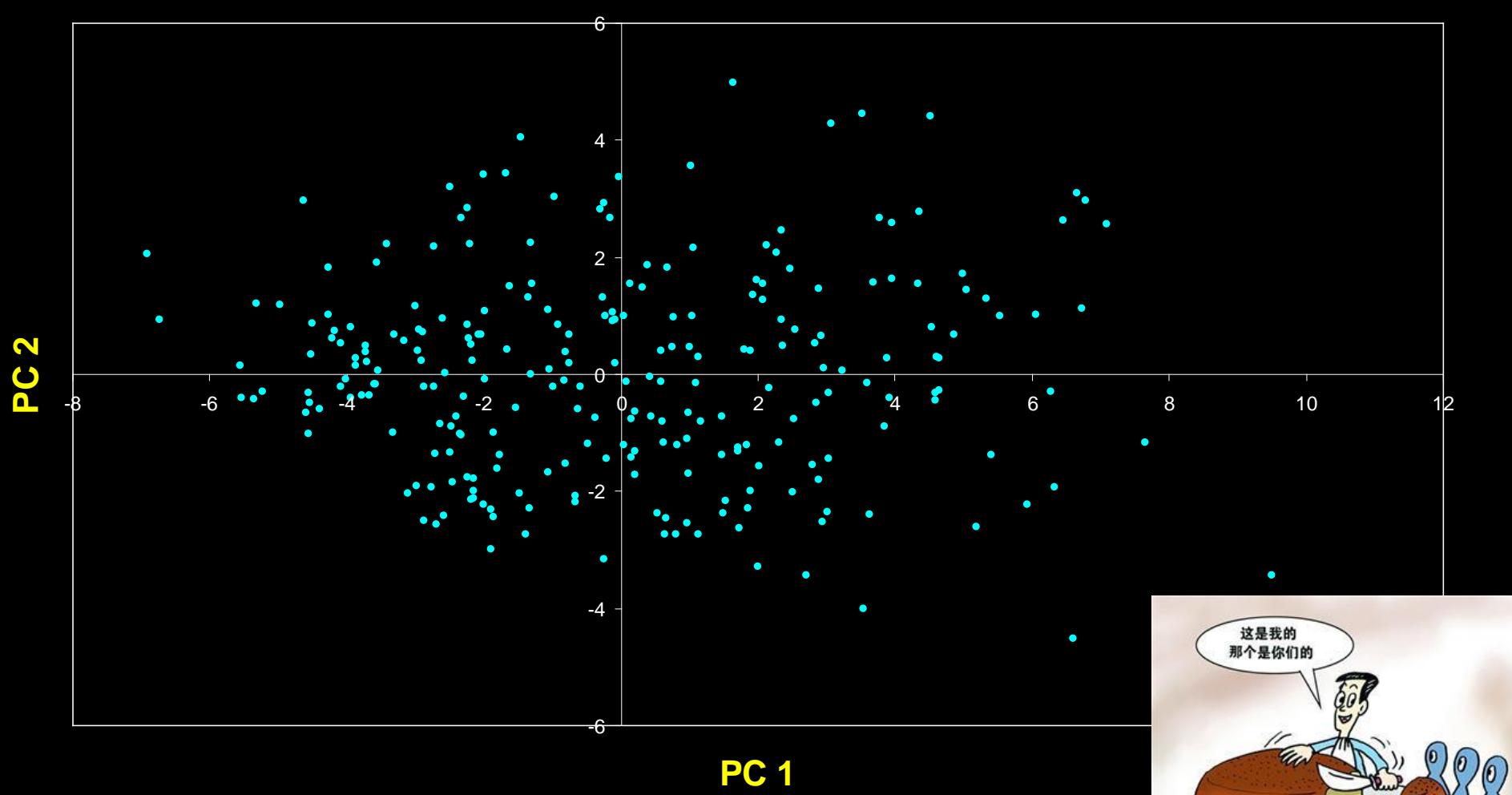


- 主分量分析又称主成分分析，也称为经验正交函数分解或特征向量分析。
- 分析对象：以网格点为**空间点**（多个变量）
随时间变化的样本。
- 主分量分析与回归分析、差别分析不同，它是一种分析方法而不是一种预报方法。
- 我们希望通过某种线性组合的方法使某个变量或者某些变量的解释方差变得比较大，这些具有较大解释方差的变量就称为主分量。



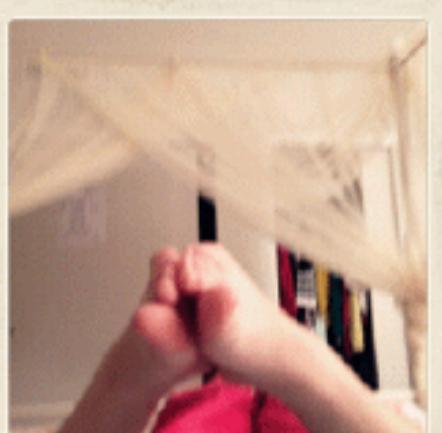
Principal Components are Computed

- PC 1 has the highest possible variance (9.88)
- PC 2 has a variance of 3.03



基于PCA算法的人脸识别

- PCA方法由于其在降维和特征提取方面的有效性，在人脸识别领域得到了广泛的应用。
- PCA方法的基本原理是：利用K-L变换抽取人脸的主要成分，构成特征脸空间，识别时将测试图像投影到此空间，得到一组投影系数，通过与各个人脸图像比较进行识别。



- 利用特征脸法进行人脸识别的过程由训练阶段和识别阶段两个阶段组成
- 其具体步骤如下：

训练阶段

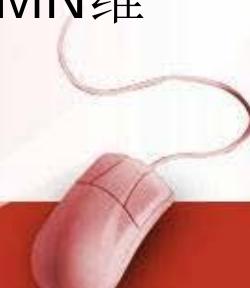
- 第一步：假设训练集有200个样本，由灰度图组成，每个样本大小为M*N



- 写出训练样本矩阵：

$$x = (x_1, x_2, \dots, x_{200})^T$$

- 其中向量 x_i 为由第*i*个图像的每一列向量堆叠成一列的MN维列向量,即把矩阵向量化,如下图所示:



训练阶段

□ 如：第*i*个图像矩阵为

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

□ 则 x_i 为

$$\begin{bmatrix} 1 \\ 4 \\ 7 \\ 2 \\ 5 \\ 8 \\ 3 \\ 6 \\ 9 \end{bmatrix}$$



训练阶段

□ 第二步：计算平均脸

计算训练图片的平均脸：

$$\Psi = \frac{1}{200} \sum_{i=1}^{i=200} x_i$$

训练阶段

□ 第三步：计算差值脸

计算每一张人脸与平均脸的差值

$$d_i = x_i - \Psi, i = 1, 2, \dots, 200$$



训练阶段

□ 第四步：构建协方差矩阵

$$C = \frac{1}{200} \sum_{i=1}^{200} d_i d_i^T = \frac{1}{200} A A^T$$

$$A = (d_1, d_2, \dots, d_{200})$$



训练阶段

□ 第五步：求协方差矩阵的特征值和特征向量，
构造特征脸空间

训练阶段

- 求出 $A^T A$ 的特征值 λ_i 及其正交归一化特征向量 V_i
- 根据特征值的贡献率选取前 p 个最大特征向量及其对应的特征向量
- 贡献率是指选取的特征值的和与占所有特征值的和比，即：

$$\varphi = \frac{\sum_{i=1}^{i=p} \lambda_i}{\sum_{i=1}^{i=200} \lambda_i} \geq a$$



训练阶段

- 一般取 $a = 99\%$ 即使训练样本在前 p 个特征向量集上的投影有99%的能量

求出原协方差矩阵的特征向量

$$u_i (i = 1, 2, \dots, p)$$

则“特征脸”空间为：

$$w = (u_1, u_2, \dots, u_p)$$



训练阶段：一个关于各特征向量贡献率的例子

主成分	特征值	贡献率/%	累计贡献率/%
z_1	4.661	51.791	51.791
z_2	2.089	23.216	75.007
z_3	1.043	11.589	86.596
z_4	0.507	5.638	92.234
z_5	0.315	3.502	95.736
z_6	0.193	2.14	97.876
z_7	0.114	1.271	99.147
z_8	0.045 3	0.504	99.65
z_9	0.0315	0.35	100

训练阶段

- 第六步
- 将每一幅人脸与平均脸的差值脸矢量投影到“特征脸”空间，即

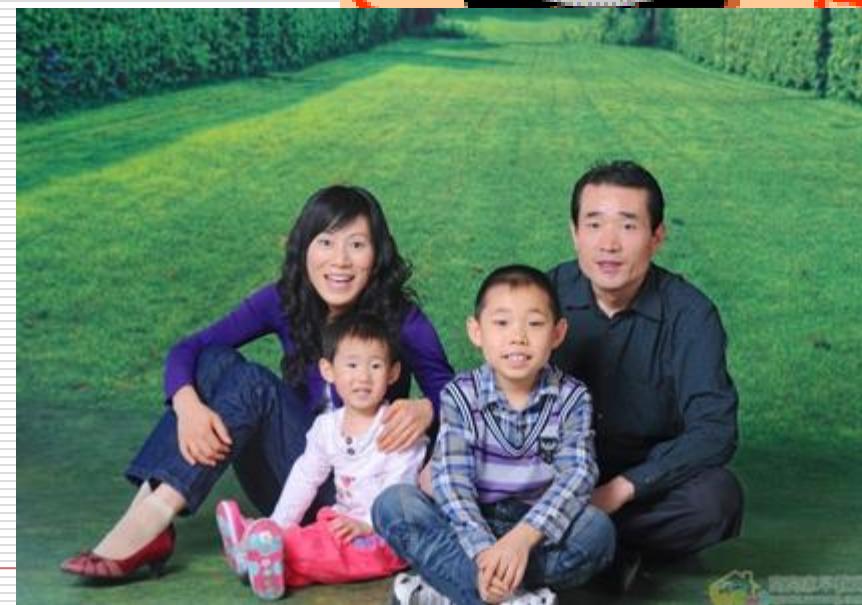
$$\Omega_i = w^T d_i \quad (i = 1, 2, \dots, 200)$$



□ 问题思考

- 大型协方差矩阵的特征向量计算？
 - 奇异值分解定理的应用：
-

识别：分类



一般的识别阶段（确定测试样本与哪一类的训练样本最近，不考虑拒识）

- 第一步：将待识别的人脸图像 Γ 与平均脸的差值脸投影到特征空间，得到其特征向量表示：

$$\Omega^\Gamma = w^T (\Gamma - \Psi)$$

- 第二步：将每一训练样本按照第一步的方式进行投影
- 第三步：找出距离待识别的人脸图像最近的训练样本，将训练样本的类别作为待识别的人脸图像的类别

□ 问题思考

■ 拒识如何实现？

2D-PCA

- 2D-PCA是在基本PCA算法上的改进，主要不同是协方差矩阵构造方法不同。
- 训练阶段复杂度更低
- 分类正确率很多情况下更高（例如人脸识别问题）

2D-PCA



- 二维主分量分析：直接对二维矩阵基础上的主分量分析方法



训练阶段

□ 1设训练样本集合为:

$$\left\{ s_j^i \in R^{m \cdot n}, i = 1, 2, \dots, N, j = 1, 2, \dots, K \right\}$$

□ 其中:

i表示第i个人，即类别数，

j表示第i个人的第j幅图像

N表示识别的人数，

K表示每个人包含K幅图像，

M表示样本总数且M=NK

训练阶段

- 2 计算所有训练样本的平均图像

$$S = \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^K S_j^{(i)}$$



训练阶段

□ 3计算样本的协方差矩阵：

$$G = \frac{1}{M} \sum_{i=1}^N \sum_{j=1}^K \left(S_j^i - S \right)^T \left(S_j^i - S \right)$$



训练阶段

- 4求出协方差矩阵的特征值，选取其中最大特征值 $u_1 \dots u_p$ 对应的正交特征向量 $X_1 \dots X_p$ 作为投影向量。

用投影矩阵Y的总离散度作为准则函数J(U)来衡量投影空间U的优劣：

$$J(U) = \text{tr}(S_u)$$



训练阶段

- S_u 是投影矩阵Y的协方差矩阵, $tr(S_u)$ 是 S_u 的迹, 且:

$$S_U = U^T E \left\{ [x - E(x)]^T [x - E(x)] \right\} U$$

- 选取的特征向量为

$$U = (X_1, X_2, \dots, X_p) = \arg \max [J(U)],$$

$$X_i^T X_j = 0; i \neq j; i, j = 1, 2, \dots, p$$



训练阶段

- 5 训练样本 $\{s_j^i, i = 1, 2, \dots, N, j = 1, 2, \dots, K\}$ 向 $X_1 \dots X_p$ 空间投影得到：

$$Y_j^i = [S_j^i X_i, \dots, S_j^i X_p] = [Y_j^i(1), \dots, Y_j^i(p)] \in R^{m*p}$$

识别阶段

- 1 测试样本 $W \in R^{m*n}$ 向 $X_1 \dots X_p$ 空间投影后得到样本W的特征矩阵 Y_t 和主成分分量 $Y_j^i(1), \dots, Y_j^i(p)$:

$$Y_t = [Y_j^i(1), \dots, Y_j^i(p)] = [WX_1, \dots, WX_p]$$



识别阶段

- 2根据测试样本投影特征矩阵与所有训练样本投影特征矩阵之间的最小距离来判断测试样本所属的类别。定义如下的距离度量准则：

$$P(Y_j^i, Y_t) = \sum_{n=1}^p \|Y_j^i(n) - Y_t(n)\|^2$$

- 其中 $\|Y_j^i(n) - Y_t(n)\|^2$ 表示两个特征向量之间的欧氏距离。



识别阶段

□ 3 若

$$p(Y_d^q, Y_t) = \min_i \min_j p(Y_j^i, Y_t)$$

则 Y_t 属于第q个人



-
- 部分相关参考文献:
 - 1. Yong Xu, David Zhang, Jing-Yu Yang, A feature extraction method for use with bimodal biometrics, Pattern recognition, 43(3) 1106-1115, 2010.
 - 2. Yong Xu, David Zhang, Jian Yang, Jing-Yu Yang, An approach for directly extracting features from matrix data and its application in face recognition, Neurocomputing, 71, 1857-1865, 2008.
 - 3. Yong Xu, David Zhang, Represent and fuse bimodal biometric images at the feature level: complex-matrix-based fusion scheme, Opt. Eng. 49, 037002 (2010) doi:10.1117/1.3359514.
 - 4. Yong Xu, Jing-Yu Yang, Zhong Jin, A novel method for Fisher discriminant Analysis. Pattern Recognition, 37 (2), 381-384, 2004
-

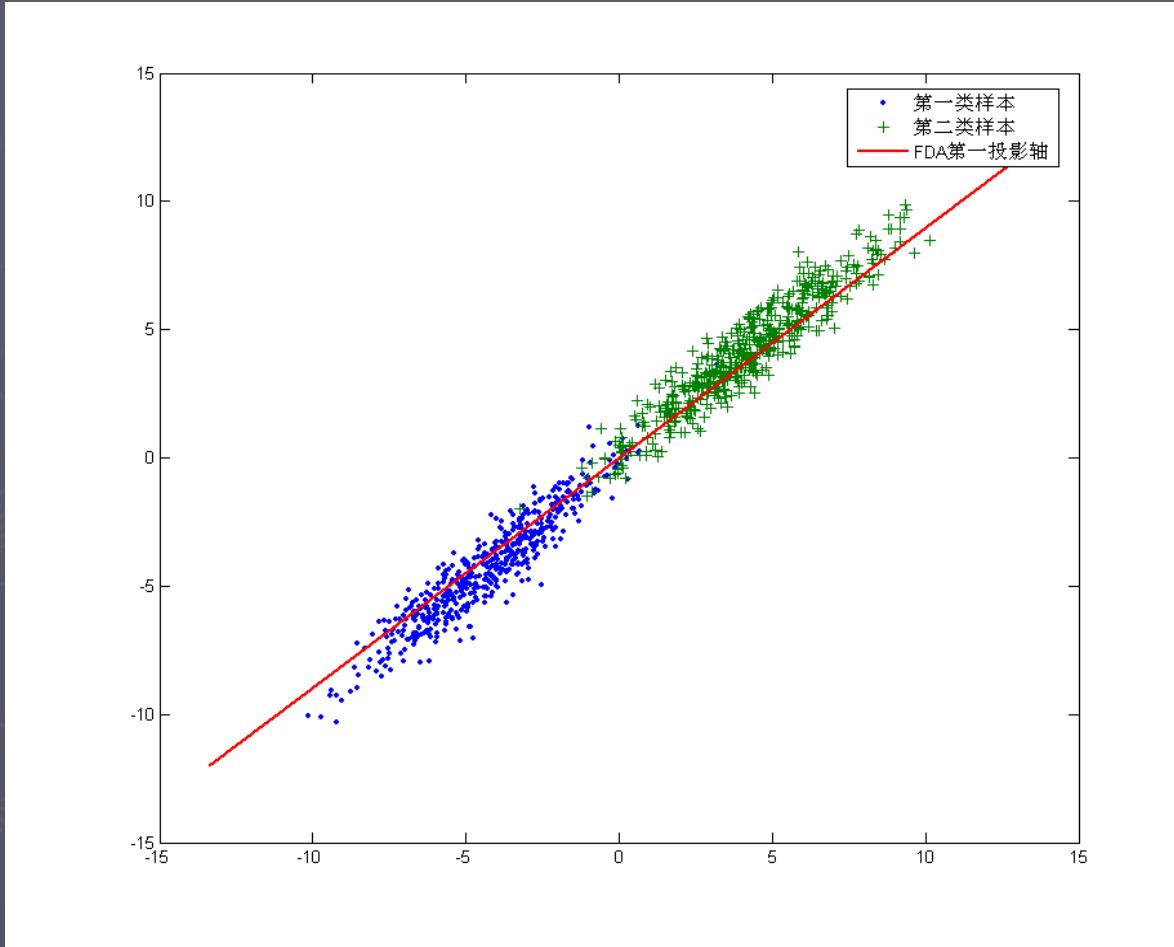
Impact on Ordination History

- ▶ by 1970 PCA was the ordination method of choice for community data
- ▶ simulation studies by Swan (1970) & Austin & Noy-Meir (1971) demonstrated the horseshoe effect and showed that the linear assumption of PCA was not compatible with the nonlinear structure of community data
- ▶ stimulated the quest for more appropriate ordination methods.

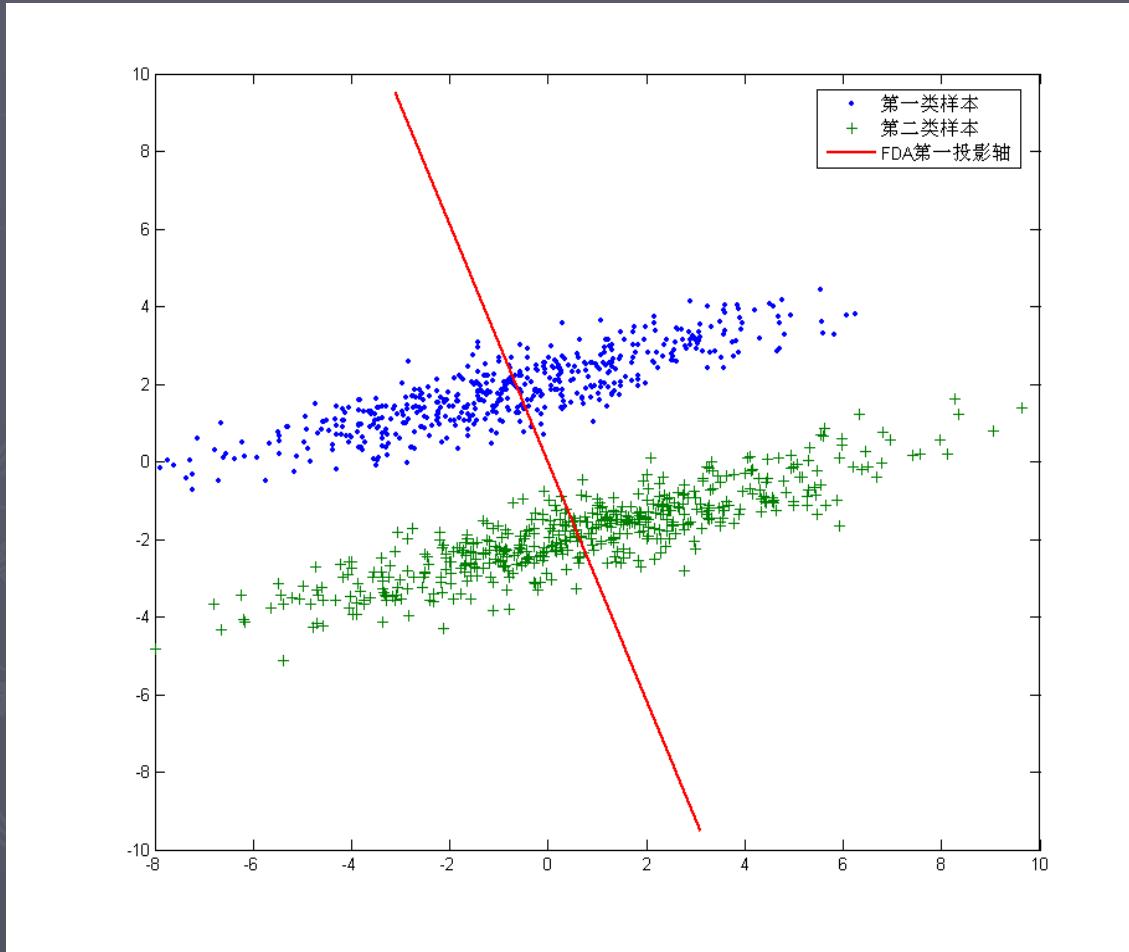
Linear discriminant analysis(LDA,FDA)

- ▶ Difference between PCA AND LDA
- ▶
- ▶ PCA is a unsupervised method and LDA is a supervised method.
- ▶ LDA seeks the axis the projections on which of samples have the maximum between-class distance and within-class distance.

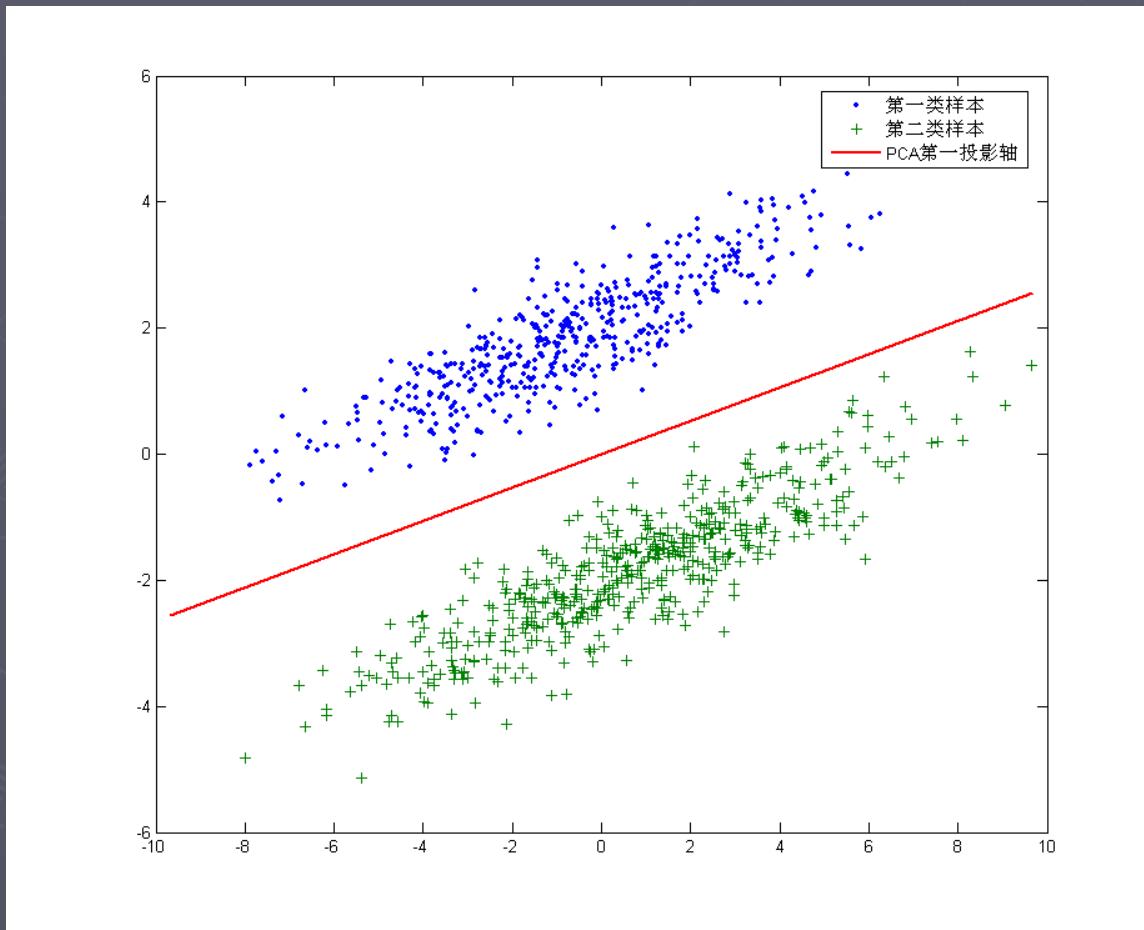
Linear discriminant analysis(LDA,FDA)



Projection axis of LDA



However, PCA performs badly in this case



Linear Discriminant Analysis (LDA)

► What is the goal of LDA?

- Perform dimensionality reduction “**while preserving as much of the class discriminatory information as possible**”.
- Seeks to find directions along which the classes are best separated.
- Takes into consideration the scatter *within-classes* but also the scatter *between-classes*.
- More capable of distinguishing image variation due to identity from variation due to other sources such as illumination and expression.

Major difference

- ▶ PCA: Class label is not considered
- ▶ LDA: Class label is fully exploited



- ▶ Training sample → Label

Linear Discriminant Analysis (LDA)

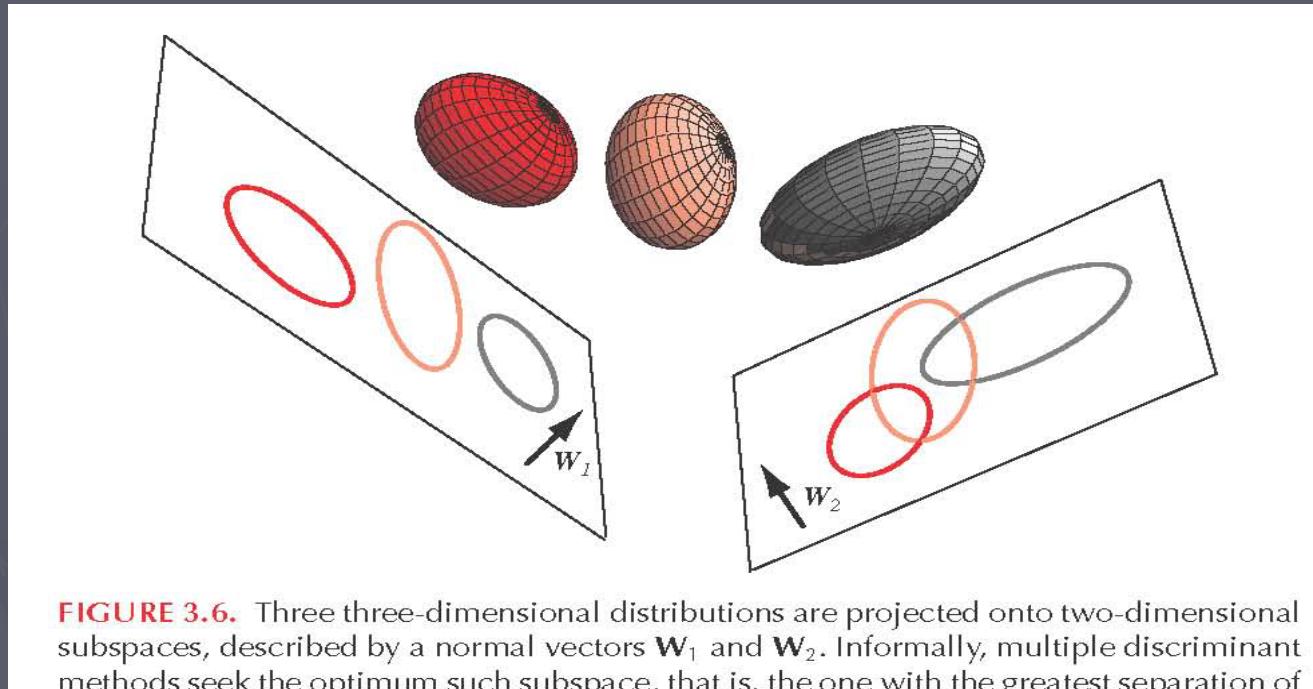


FIGURE 3.6. Three three-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors \mathbf{W}_1 and \mathbf{W}_2 . Informally, multiple discriminant methods seek the optimum such subspace, that is, the one with the greatest separation of the projected distributions for a given total within-scatter matrix, here as associated with \mathbf{W}_1 . From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*, 2nd edition, Wiley, 2001. Copyright © 2001 by John Wiley & Sons, Inc.



How does LDA differ

- ▶ Remember the label of each class and try to make every class be different from the others



Linear Discriminant Analysis (LDA)

► Notation

- Suppose there are C classes
- Let $\boldsymbol{\mu}_i$ be the mean vector of class i , $i = 1, 2, \dots, C$
- Let M_i be the number of samples within class i , $i = 1, 2, \dots, C$,
- Let $M = \sum_{i=0}^C M_i$ be the total number of samples. and

Within-class scatter matrix:

$$S_w = \sum_{i=1}^C \sum_{j=1}^{M_i} (x_{ij} - \boldsymbol{\mu}_i)(x_{ij} - \boldsymbol{\mu}_i)^T$$

Between-class scatter matrix:

(S_b has at most rank $C-1$) $S_b = \sum_{i=1}^C (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$

(each sub-matrix has
rank 1 or less, i.e., outer
product of two vectors)

$$\boldsymbol{\mu} = 1/C \sum_{i=1}^C \boldsymbol{\mu}_i \text{ (mean of entire data set)}$$

Linear Discriminant Analysis (LDA)

► Methodology

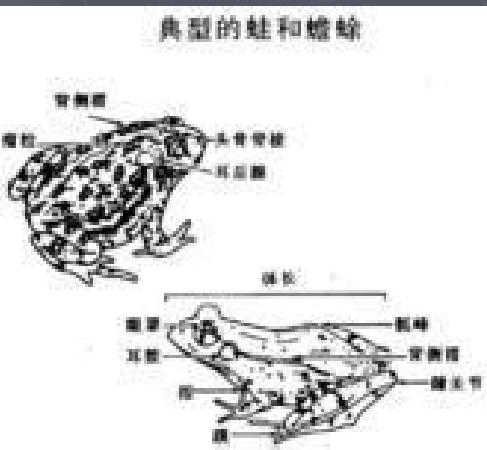
projection matrix

$$\mathbf{y} = U^T (\mathbf{x} - \boldsymbol{\mu})$$

- LDA computes a transformation that maximizes the between-class scatter while minimizing the within-class scatter:

$$\max \frac{|U^T S_b U|}{|U^T S_w U|} = \max \frac{|\tilde{S}_b|}{|\tilde{S}_w|}$$

products of eigenvalues !



\tilde{S}_b, \tilde{S}_w : scatter matrices of the projected data \mathbf{y}

Linear Discriminant Analysis (LDA)

► Linear transformation implied by LDA

- The LDA solution is given by the eigenvectors of the *generalized eigenvector problem*:

$$S_B u_k = \lambda_k S_w u_k$$

- The linear transformation is given by a matrix U whose columns are the eigenvectors of the above problem (i.e., called *Fisherfaces*).

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_K \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_K^T \end{bmatrix} (x - \mu) = U^T (x - \mu)$$

当里个当,当里个当



- Important:** Since S_b has at most rank C-1, the max number of eigenvectors with non-zero eigenvalues is C-1 (i.e., **max dimensionality of sub-space is C-1**)

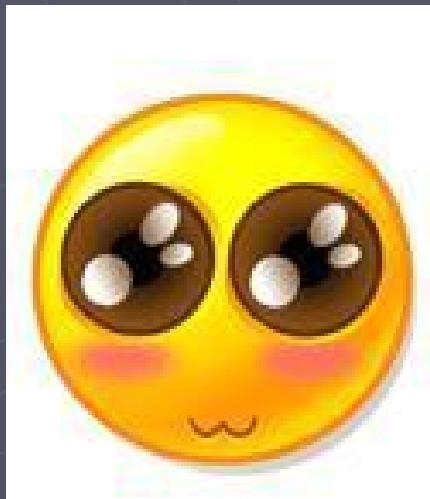
Linear Discriminant Analysis (LDA)

► Does S_w^{-1} always exist?

- If S_w is non-singular, we can obtain a conventional eigenvalue problem by writing:

$$S_w^{-1} S_B u_k = \lambda_k u_k$$

- In practice, S_w is often singular since the data are image vectors with large dimensionality while the size of the data set is much smaller ($M \ll N$)



Linear Discriminant Analysis (LDA)

► Does S_w^{-1} always exist? – cont.

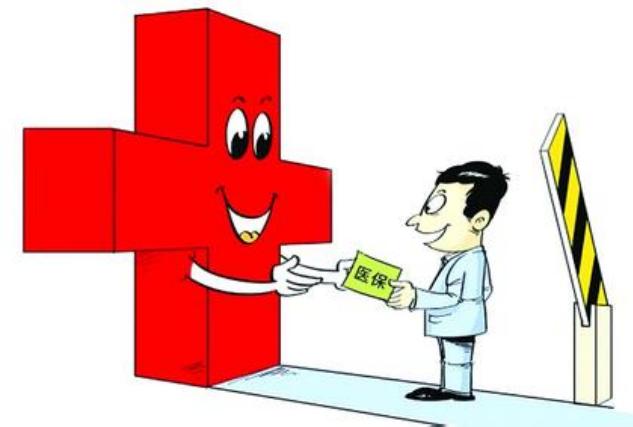
- To alleviate this problem, we can use PCA first:

1) PCA is first applied to the data set to reduce its dimensionality.

$$\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix} \dashrightarrow PCA \dashrightarrow \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix}$$

2) LDA is then applied to find the most discriminative directions:

$$\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_K \end{bmatrix} \dashrightarrow LDA \dashrightarrow \begin{bmatrix} z_1 \\ z_2 \\ \dots \\ z_{C-1} \end{bmatrix}$$



Linear Discriminant Analysis (LDA)

- ▶ **Case Study:** Using Discriminant Eigenfeatures for Image Retrieval
 - D. Swets, J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.
- ▶ Content-based image retrieval
 - The application being studied here is *query-by-example* image retrieval.
 - The paper deals with the problem of *Selecting a good set of image features* for content-based image retrieval.

Linear Discriminant Analysis (LDA)

► Assumptions

- "Well-framed" images are required as input for training and query-by-example test probes.
- Only a small variation in the size, position, and orientation of the objects in the images is allowed.

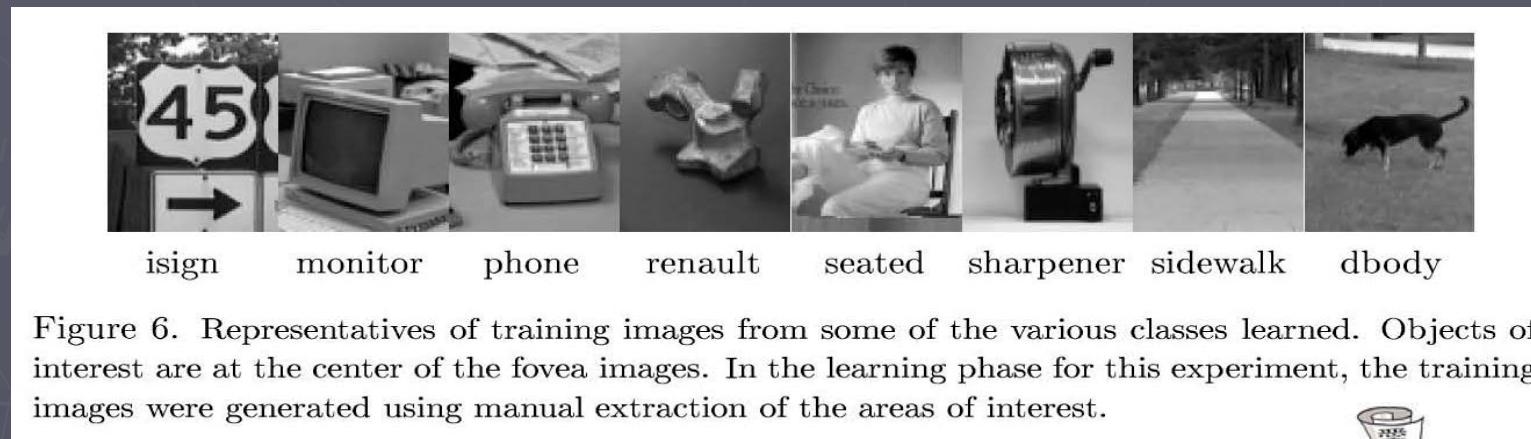


Figure 6. Representatives of training images from some of the various classes learned. Objects of interest are at the center of the fovea images. In the learning phase for this experiment, the training images were generated using manual extraction of the areas of interest.



Linear Discriminant Analysis (LDA)

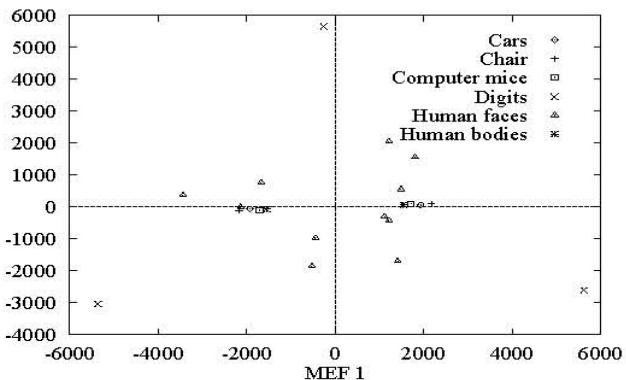
- ▶ Some terminology
- Most Expressive Features (MEF): the features (projections) obtained using PCA
- Most Discriminating Features (MDF): the features (projections) obtained using LDA.



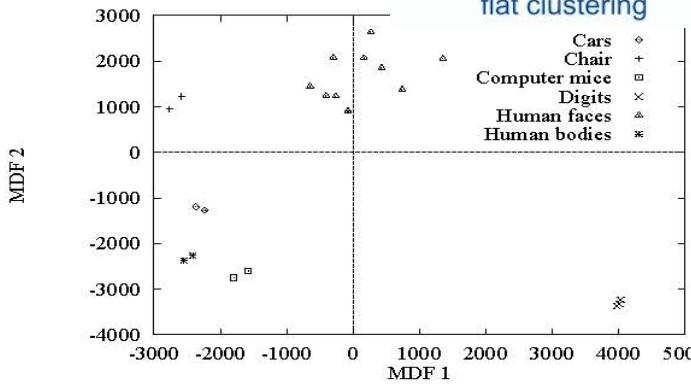
Linear Discriminant Analysis (LDA)

Clustering: unsupervised learning

► Clustering effect



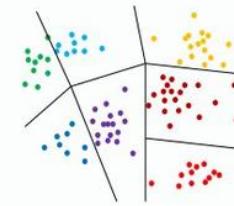
(a) MEF space



(b) MDF space

► Methodology

- 1) Generate the features for each image in the training set.
- 2) Given an query image, compute its features using the same procedure.
- 3) Find the ***k* closest neighbors** for retrieval (e.g., using Euclidean distance).



flat clustering



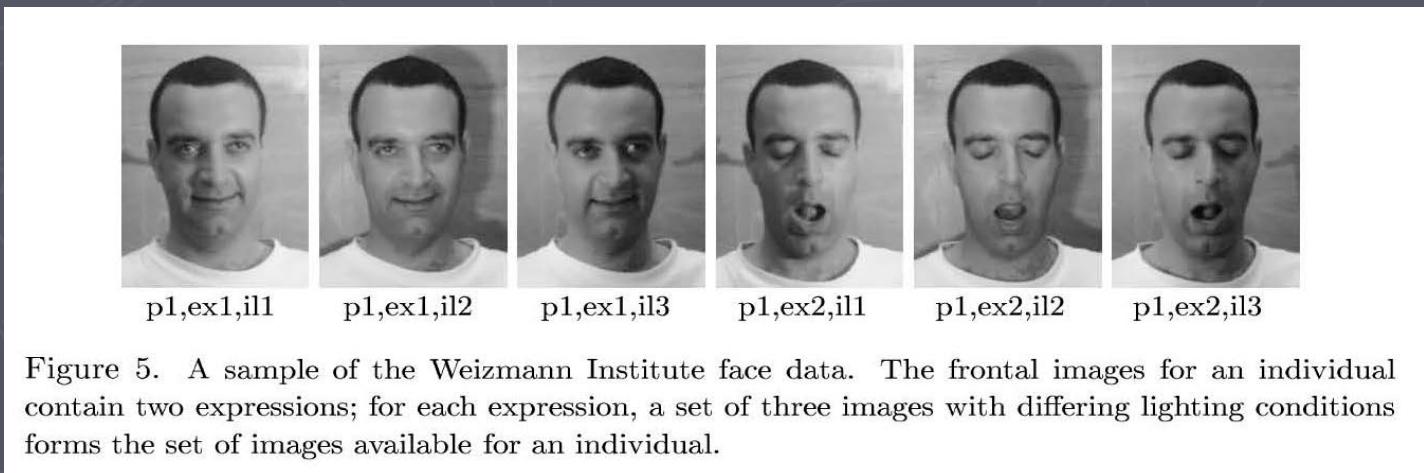
hierarchical clustering

Linear Discriminant Analysis (LDA)

► Experiments and results

■ Face images

- ▶ A set of face images was used with 2 expressions, 3 lighting conditions.
- ▶ Testing was performed using a disjoint set of images:
 - One image, randomly chosen, from each individual.



Linear Discriminant Analysis (LDA)

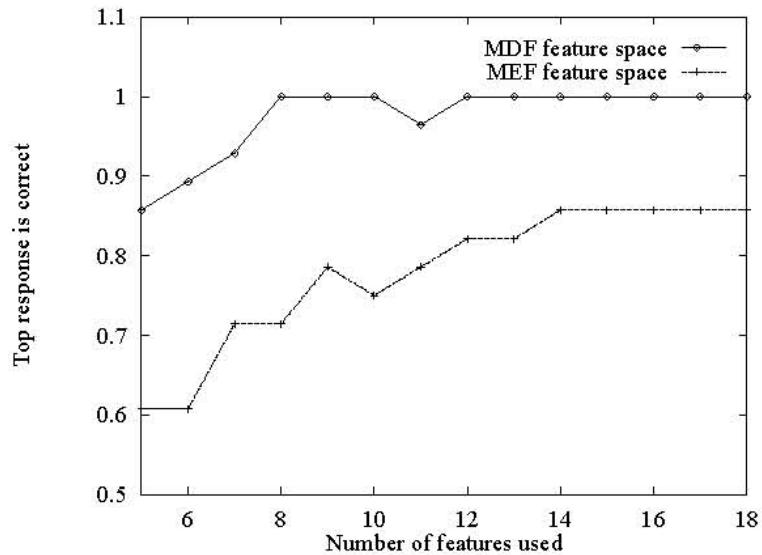


Figure 4. The performance of the system for different numbers of MEF and MDF features, respectively. The number of features from the subspace used was varied to show how the MDF subspace outperforms the MEF subspace. 95% of the variance for the MDF subspace was attained when 15 features were used; 95% of variance for the MEF subspace did not occur until 37 features were used. Using 95% of the MEF variance resulted in an 89% recognition rate, and that rate was not improved using more features.

Linear Discriminant Analysis (LDA)

- Examples of correct search probes



(a) List of training images



(b) List of search probes

Figure 8. Example of how well within-class variation is handled. The system correctly retrieved images from the class defined by the training samples for each of the search probes.

Linear Discriminant Analysis (LDA)

- Example of a failed search probe



(a) Search probe



(b) Training images

Figure 7. Example of a failed search probe. The retrieval failed to select the appropriate class due to a lack of 3D rotation in the set of training images.



Linear Discriminant Analysis (LDA)

► Case Study: PCA versus LDA

- A. Martinez, A. Kak, "PCA versus LDA", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, 2001.
 - Is LDA always better than PCA?
- There has been a tendency in the computer vision community to prefer LDA over PCA.
- This is mainly because LDA deals directly with discrimination between classes while PCA does not pay attention to the underlying class structure.
- Main results of this study:
 - (1) When the training set is small, PCA can outperform LDA.
 - (2) When the number of samples is large and representative for each class, LDA often outperforms PCA.



Linear Discriminant Analysis (LDA)

► Is LDA always better than PCA? – cont.

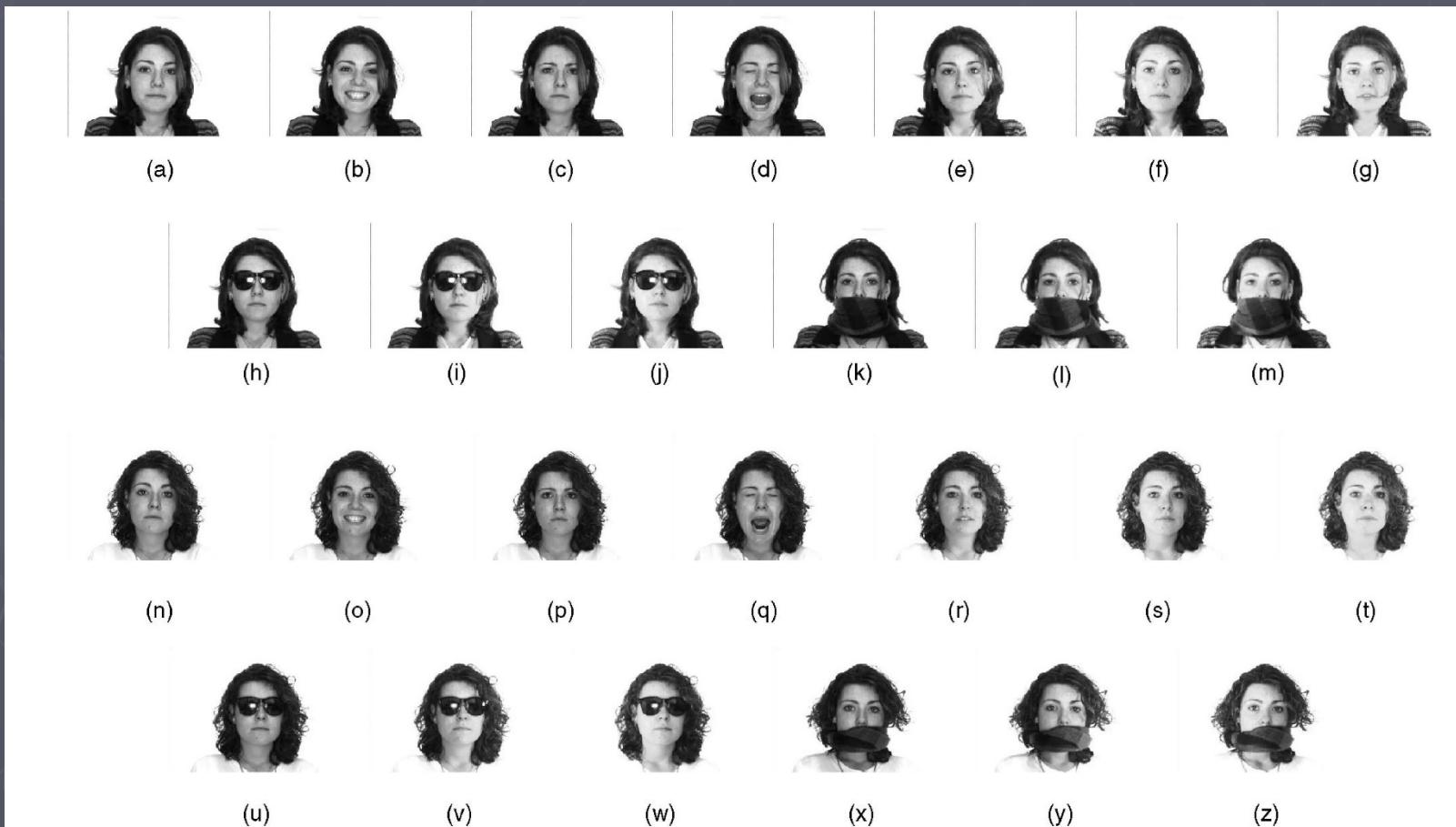


Fig. 3. Images of one subject in the AR face database. The images (a)-(m) were taken during one session and the images (n)-(z) at a different session.

Linear Discriminant Analysis (LDA)

- ▶ Is LDA always better than PCA? – cont.

LDA is not always better when training set is small

Linear Discriminant Analysis (LDA)

- ▶ Is LDA always better than PCA? – cont.

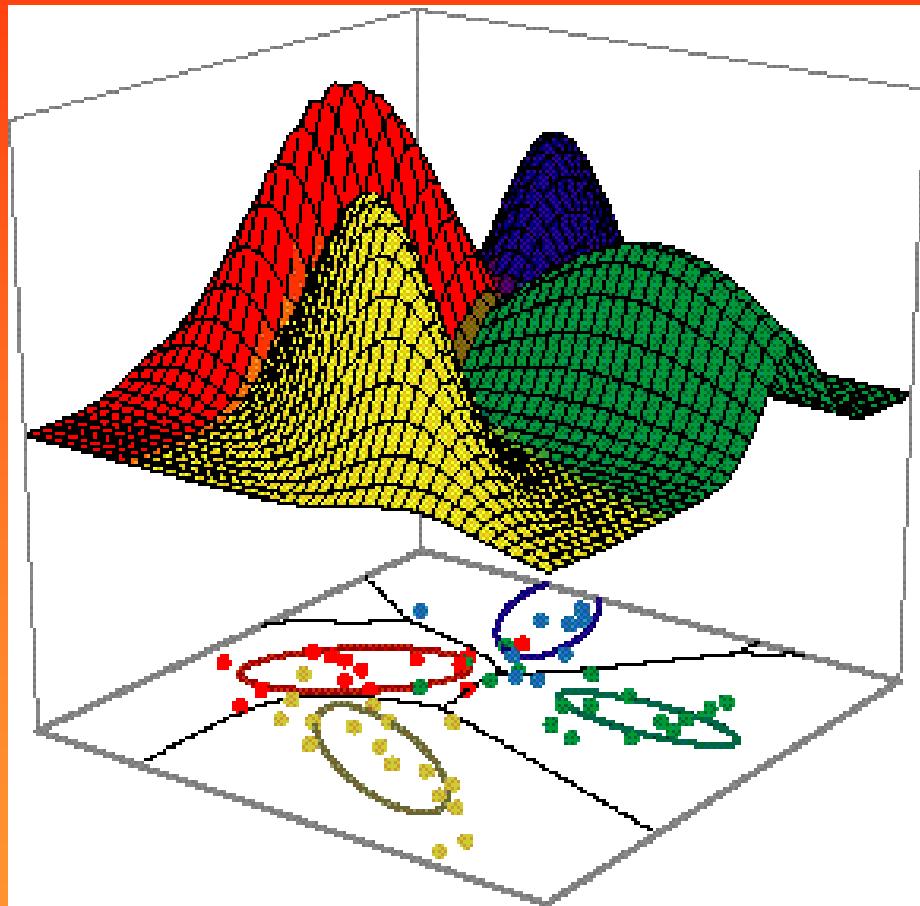
LDA often outperforms PCA when training set is large



Advances

- ▶ Jian Yang, David Zhang, Xu Yong, and Jing-yu Yang, Two-dimensional Discriminant Transform for Face Recognition, *Pattern Recognition*, 2005, 38(7), 1125-1129.
- ▶ Y. Xu, D. Zhang, Represent and fuse bimodal biometric images at the feature level: complex-matrix-based fusion scheme, *Optical Engineering*, 49(3), 037002, 2010
- ▶ Y. Xu, Quaternion-Based Discriminant Analysis Method for Color Face Recognition, *PLoS ONE*, 7(8): e43493, 2012
- ▶ <http://www.yongxu.org/lunwen.html>

Pattern Classification



All materials in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda,
P. E. Hart and D. G. Stork, John Wiley & Sons,
2000
with the permission of the authors and the
publisher

Chapter 4 (Part 1): Non-Parametric Classification (Sections 4.1-4.3)

- Introduction
- Density Estimation
- Parzen Windows

Introduction

- All Parametric densities are unimodal (have a single local maximum), whereas many practical problems involve multimodal densities
- Nonparametric procedures can be used with arbitrary distributions and without the assumption that the forms of the underlying densities are known
- There are two types of nonparametric methods:
 - Estimating $P(x | w_j)$
 - Bypass probability and go directly to a-posteriori probability estimation

Density Estimation

- Basic idea:
- Probability that a vector x will fall in region R is:

$$P = \int_{\mathcal{R}} p(x') dx' \quad (1)$$

- P is a smoothed (or averaged) version of the density function $p(x)$ if we have a sample of size n ; therefore, the probability that k points fall in R is then:

$$P_k = \binom{n}{k} P^k (1 - P)^{n-k} \quad (2)$$

and the expected value for k is:

$$E(k) = nP \quad (3)$$

ML estimation of $P = \theta$

$$\underset{\theta}{\operatorname{Max}}(P_k / \theta) \text{ is reached for } \hat{\theta} = \frac{k}{n} \cong P$$

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

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

$$\int_{\mathcal{R}} p(x') dx' \cong p(x)V \quad (4)$$

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

Combining equation (1) , (3) and (4) yields:

$$p(x) \cong \frac{k/n}{V}$$

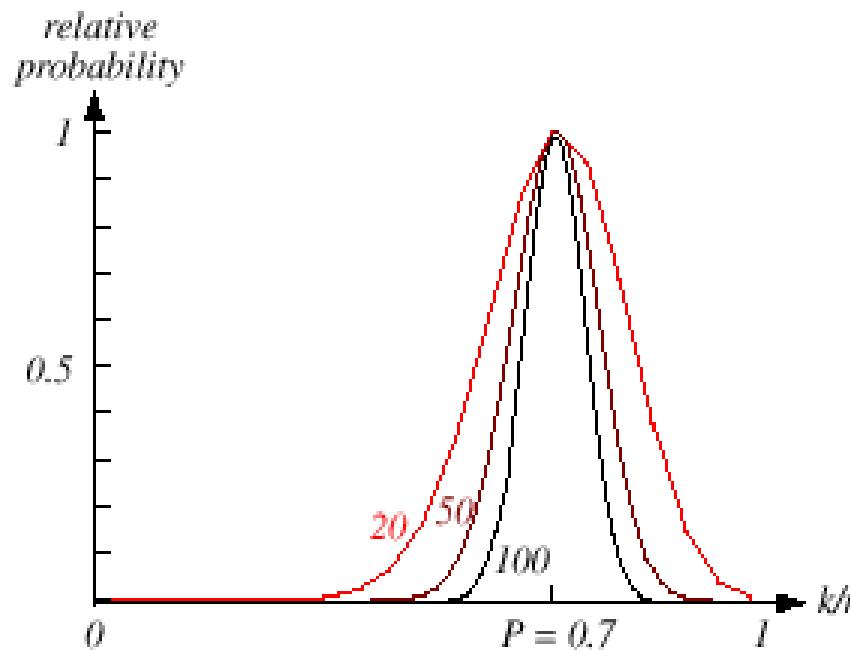


FIGURE 4.1. The relative probability an estimate given by Eq. 4 will yield a particular value for the probability density, here where the true probability was chosen to be 0.7. Each curve is labeled by the total number of patterns n sampled, and is scaled to give the same maximum (at the true probability). The form of each curve is binomial, as given by Eq. 2. For large n , such binomials peak strongly at the true probability. In the limit $n \rightarrow \infty$, the curve approaches a delta function, and we are guaranteed that our estimate will give the true probability. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Density Estimation (cont.)

- Justification of equation (4)

$$\int_{\mathcal{R}} p(x') dx' \cong p(x)V \quad (4)$$

We assume that $p(x)$ is continuous and that region \mathcal{R} is so small that p does not vary significantly within \mathcal{R} . Since $p(x) = \text{constant}$, it is not a part of the sum.

$$\int_{\mathcal{R}} p(x') dx' = p(x') \int_{\mathcal{R}} dx' = p(x') \int_{\mathcal{R}} I_{\mathcal{R}}(x) dx' = p(x') \mu(\mathcal{R})$$

Where: $\mu(\mathcal{R})$ is: a surface in the Euclidean space R^2

a volume in the Euclidean space R^3

a hypervolume in the Euclidean space R^n

Since $p(x) \cong p(x') = \text{constant}$, therefore in the Euclidean space R^3 :

$$\int_{\mathcal{R}} p(x') dx' \cong p(x) V$$

and $p(x) \cong \frac{k}{nV}$

- Condition for convergence

The fraction $k/(nV)$ is a space averaged value of $p(x)$.
 $p(x)$ is obtained only if V approaches zero.

$$\lim_{V \rightarrow 0, k=0} p(x) = 0 \text{ (if } n = \text{fixed)}$$

This is the case where no samples are included in \mathcal{R} :
it is an uninteresting case!

$$\lim_{V \rightarrow 0, k \neq 0} p(x) = \infty$$

In this case, the estimate diverges: it is an
uninteresting case!

- The volume V needs to approach 0 anyway if we want to use this estimation
 - Practically, V cannot be allowed to become small since the number of samples is always limited
 - One will have to accept a certain amount of variance in the ratio k/n
 - Theoretically, if an unlimited number of samples is available, we can circumvent this difficulty
To estimate the density of x , we form a sequence of regions $\mathcal{R}_1, \mathcal{R}_2, \dots$ containing x : the first region contains one sample, the second two samples and so on.
Let V_n be the volume of \mathcal{R}_n , k_n the number of samples falling in \mathcal{R}_n and $p_n(x)$ be the n^{th} estimate for $p(x)$:

$$p_n(x) = (k_n/n)/V_n \quad (7)$$

Three necessary conditions should apply if we want $p_n(x)$ to converge to $p(x)$

$$1) \lim_{n \rightarrow \infty} V_n = 0$$

$$2) \lim_{n \rightarrow \infty} k_n = \infty$$

$$3) \lim_{n \rightarrow \infty} k_n / n = 0$$

There are two different ways of obtaining sequences of regions that satisfy these conditions:

(a) Shrink an initial region where $V_n = 1/\sqrt{n}$ and show that

$$p_n(x) \xrightarrow{n \rightarrow \infty} p(x)$$

This is called “the Parzen-window estimation method”

(b) Specify k_n as some function of n , such as $k_n = \sqrt{n}$; the volume V_n is grown until it encloses k_n neighbors of x . This is called “the k_n -nearest neighbor estimation method”

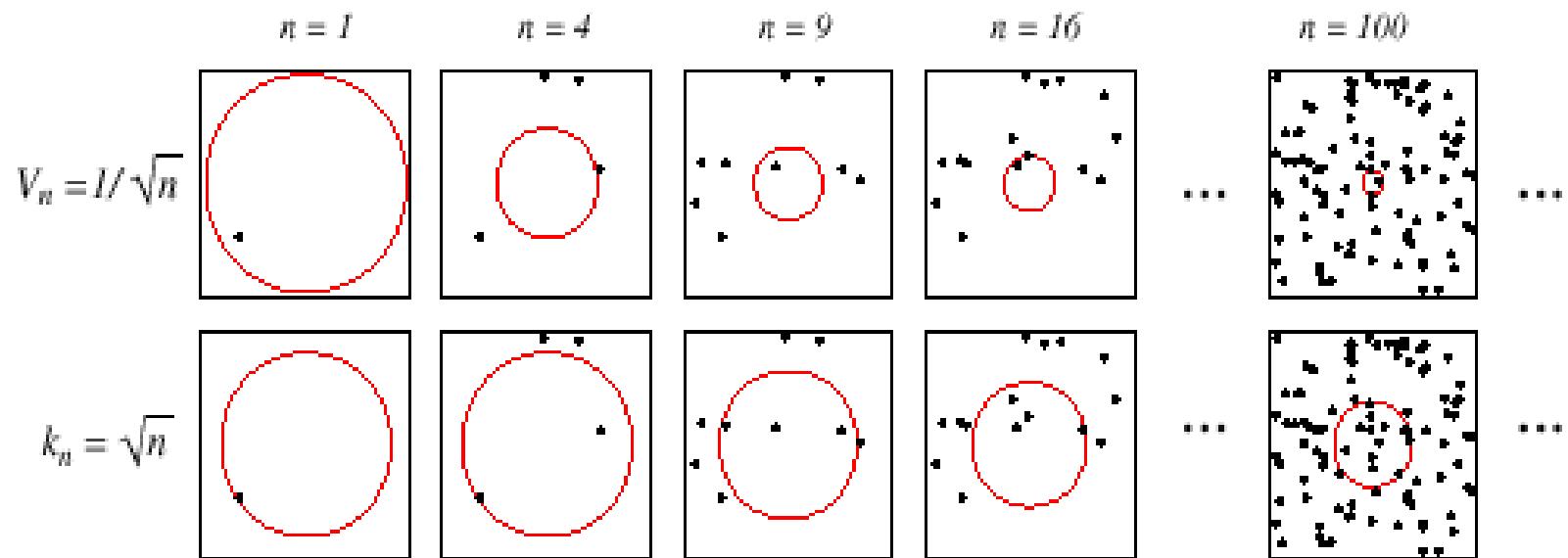
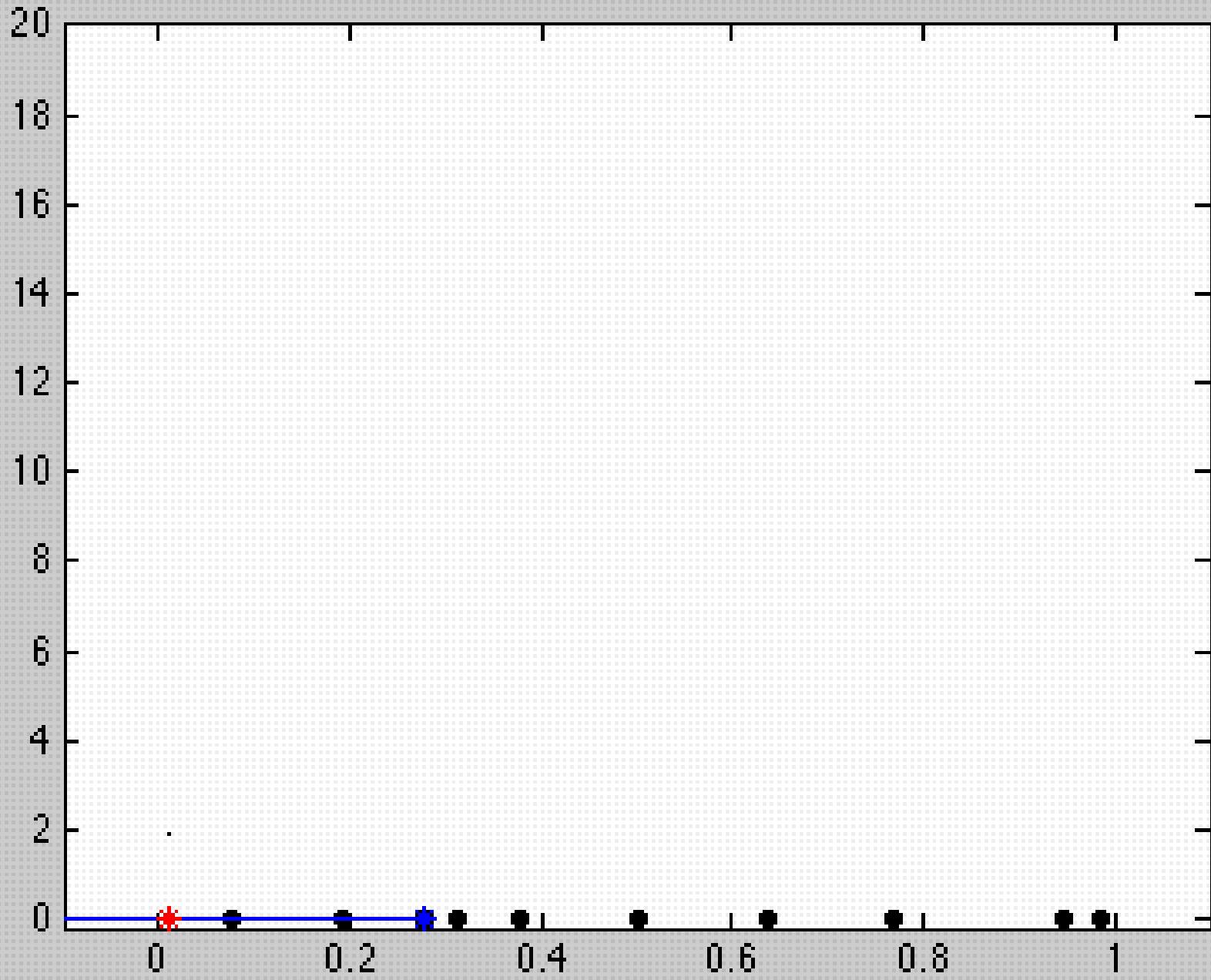


FIGURE 4.2. There are two leading methods for estimating the density at a point, here at the center of each square. The one shown in the top row is to start with a large volume centered on the test point and shrink it according to a function such as $V_n = 1/\sqrt{n}$. The other method, shown in the bottom row, is to decrease the volume in a data-dependent way, for instance letting the volume enclose some number $k_n = \sqrt{n}$ of sample points. The sequences in both cases represent random variables that generally converge and allow the true density at the test point to be calculated. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



Parzen Windows

- Parzen-window approach to estimate densities assume that the region \mathcal{R}_n is a d-dimensional hypercube

$$V_n = h_n^d \quad (h_n : \text{length of the edge of } \mathcal{R}_n)$$

Let $\varphi(u)$ be the following window function :

$$\varphi(u) = \begin{cases} 1 & |u_j| \leq \frac{1}{2} \quad j = 1, \dots, d \\ 0 & \text{otherwise} \end{cases}$$

- $\varphi((x-x_i)/h_n)$ is equal to unity if x_i falls within the hypercube of volume V_n centered at x and equal to zero otherwise.

- The number of samples in this hypercube is:

$$k_n = \sum_{i=1}^{i=n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

By substituting k_n in equation (7), we obtain the following estimate:

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{i=n} \frac{1}{V_n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

$P_n(x)$ estimates $p(x)$ as an average of functions of x and the samples (x_i) ($i = 1, \dots, n$). These functions φ can be general!

- Illustration
- The behavior of the Parzen-window method

- Case where $p(x) \rightarrow N(0, 1)$

Let $\varphi(u) = (1/\sqrt{2\pi}) \exp(-u^2/2)$ and $h_n = h_1/\sqrt{n}$ ($n > 1$)

(h_1 : known parameter)

Thus:

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_n} \varphi\left(\frac{x - x_i}{h_n}\right)$$

is an average of normal densities centered at the samples x_i .

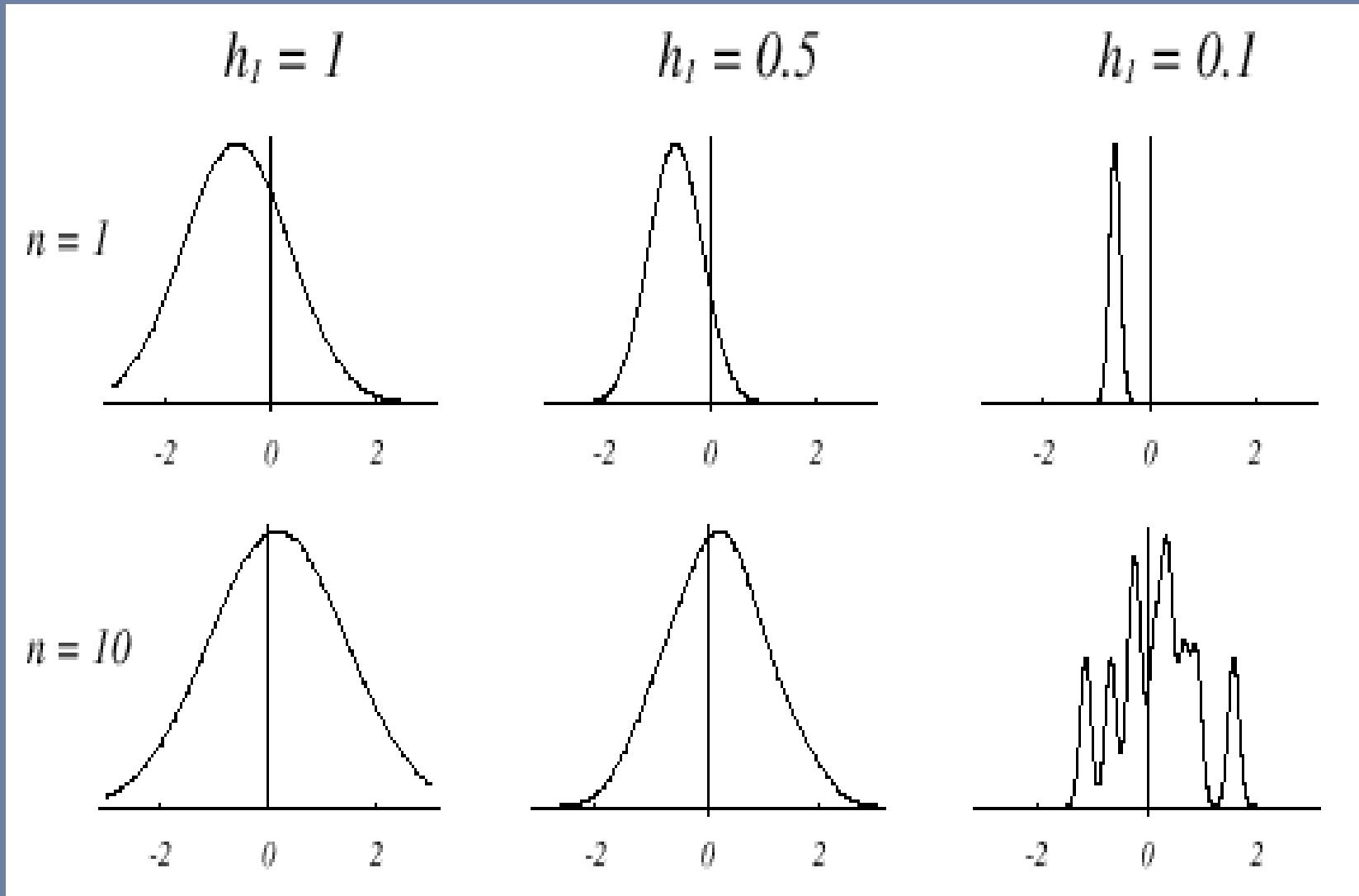
- Numerical results:

For $n = 1$ and $h_1=1$

$$p_1(x) = \varphi(x - x_1) = \frac{1}{\sqrt{2\pi}} e^{-1/2} (x - x_1)^2 \rightarrow N(x_1, 1)$$

$$p_n(x) = \frac{1}{n} \sum_{i=1, \dots, n} \frac{1}{h_n} \phi\left(\frac{x - x_i}{h_n}\right)$$

For $n = 10$ and $h = 0.1$, the contributions of the individual samples are clearly observable !



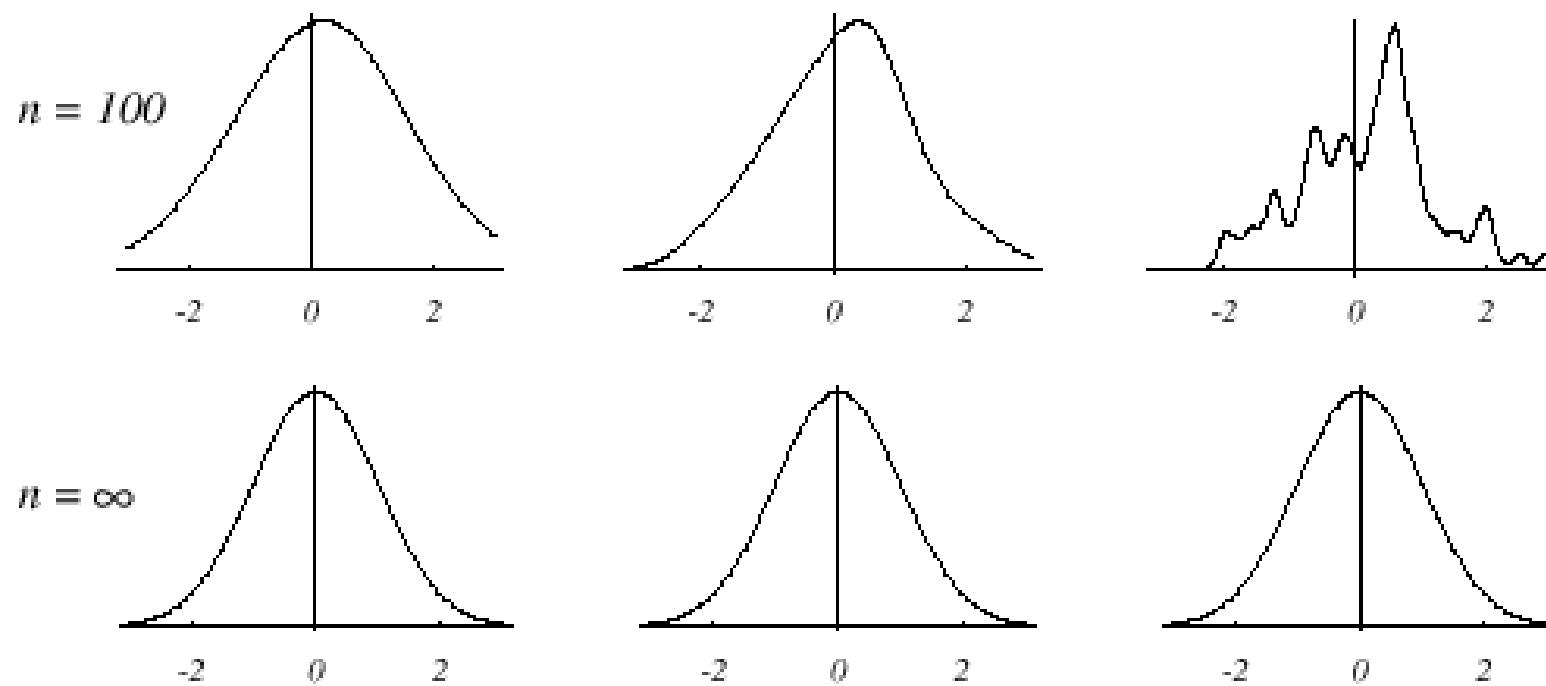
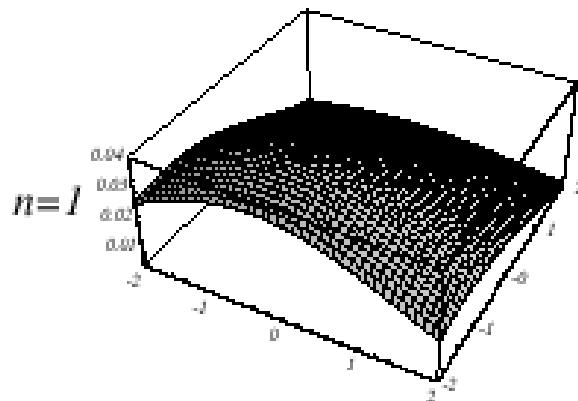


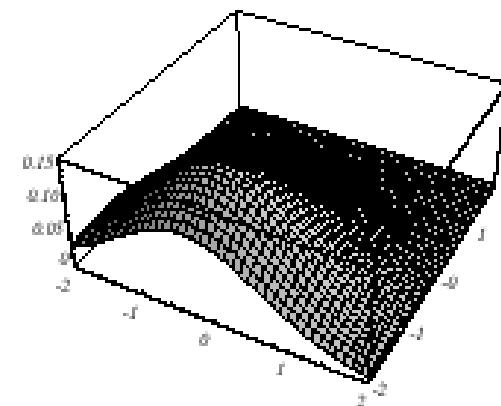
FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Analogous results are also obtained in two dimensions as illustrated:

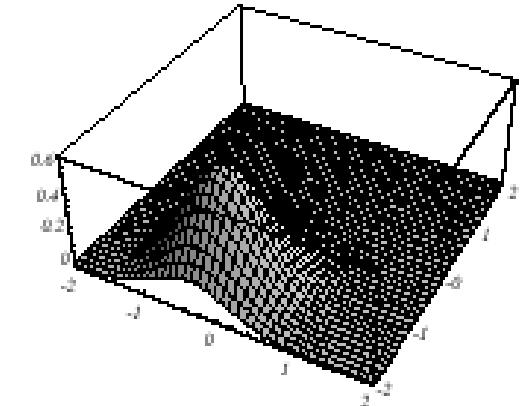
$$h_i=2$$



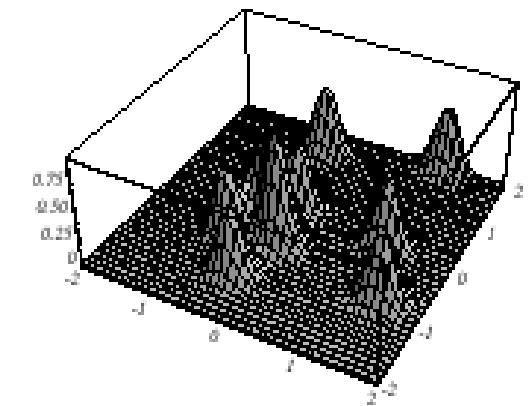
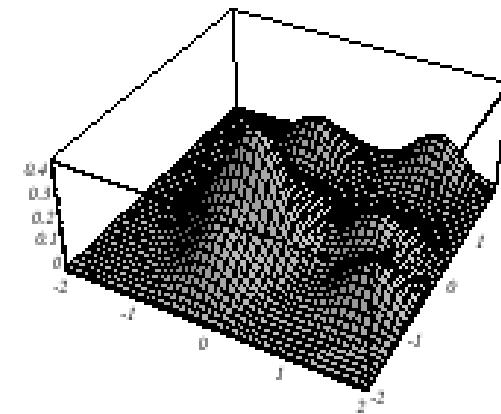
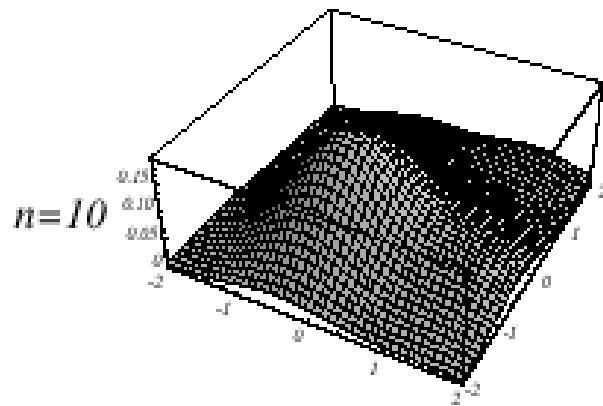
$$h_i=1$$



$$h_i=0.5$$



$$n=10$$



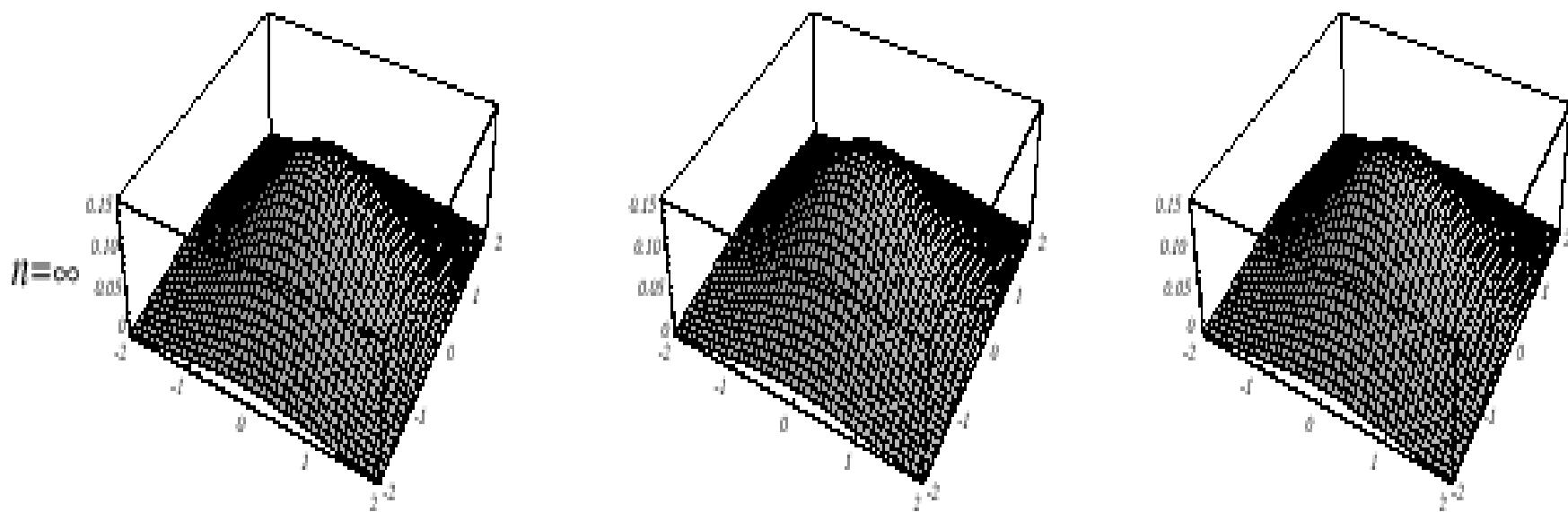
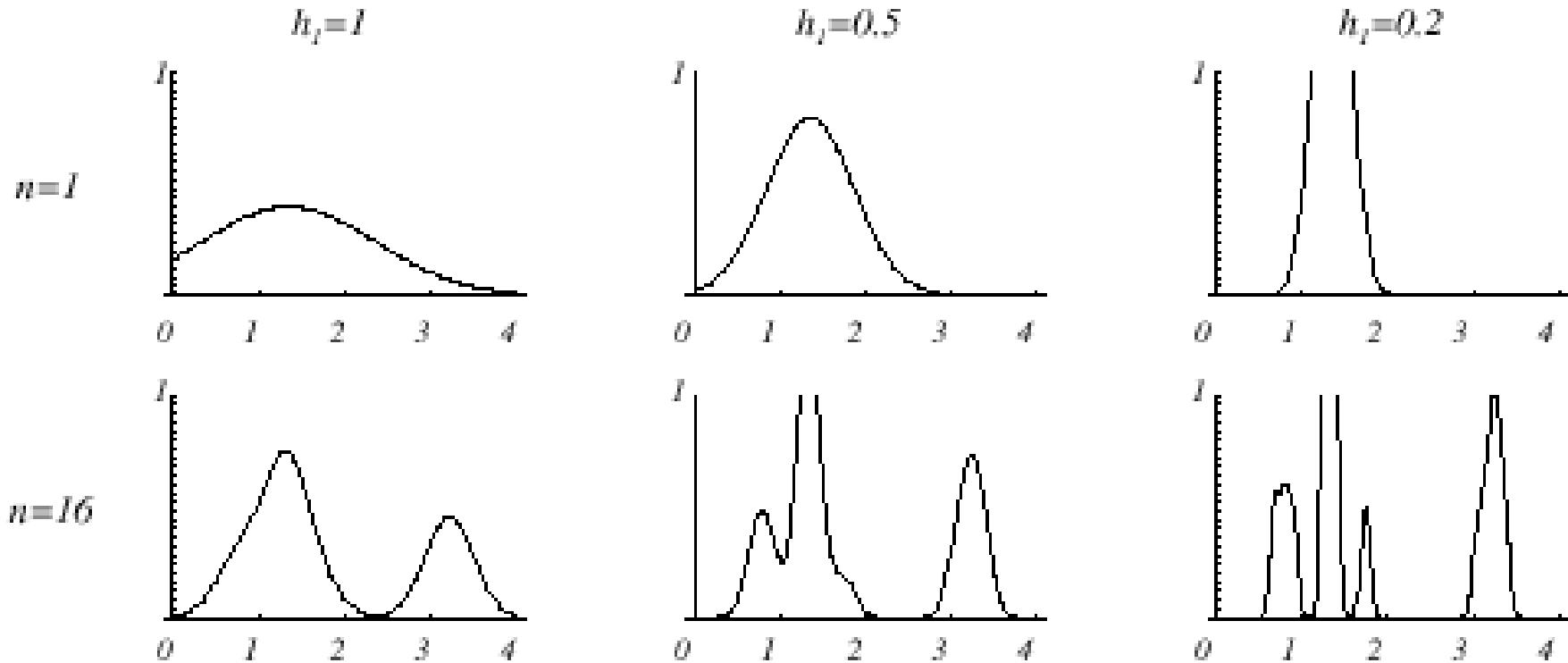


FIGURE 4.6. Parzen-window estimates of a bivariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- Case where $p(x) = \lambda_1 \cdot U(a,b) + \lambda_2 \cdot T(c,d)$
 (unknown density) (mixture of a uniform and a triangle density)



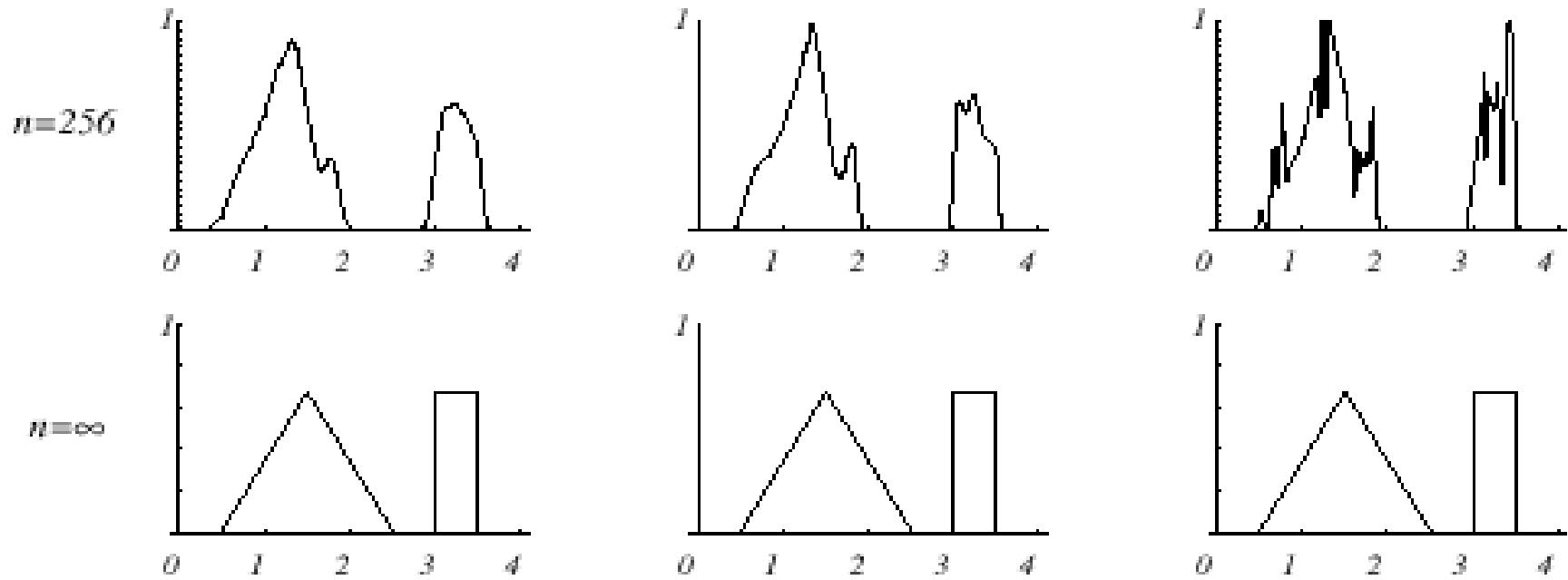


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- Classification example

In classifiers based on Parzen-window estimation:

- We estimate the densities for each category and classify a test point by the label corresponding to the maximum posterior
- The decision region for a Parzen-window classifier depends upon the choice of window function as illustrated in the following figure.

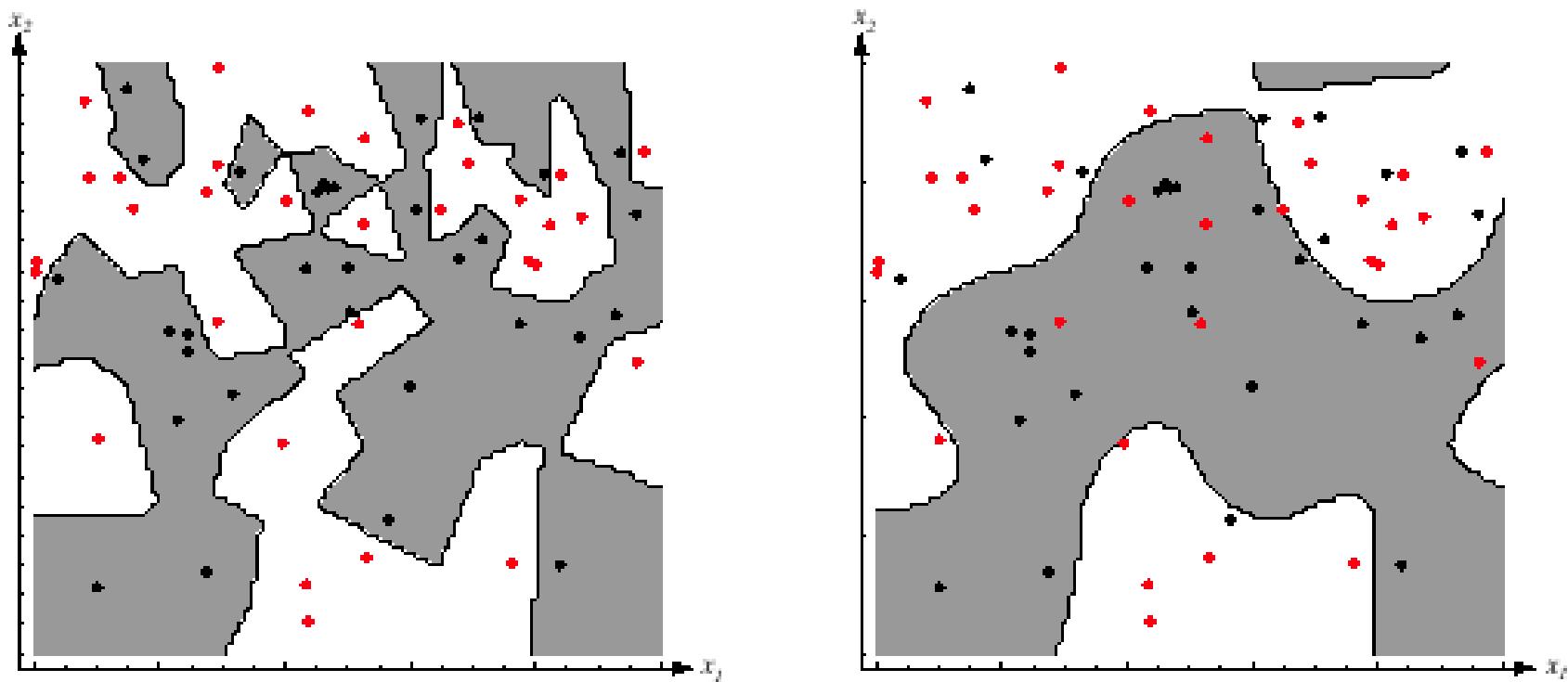
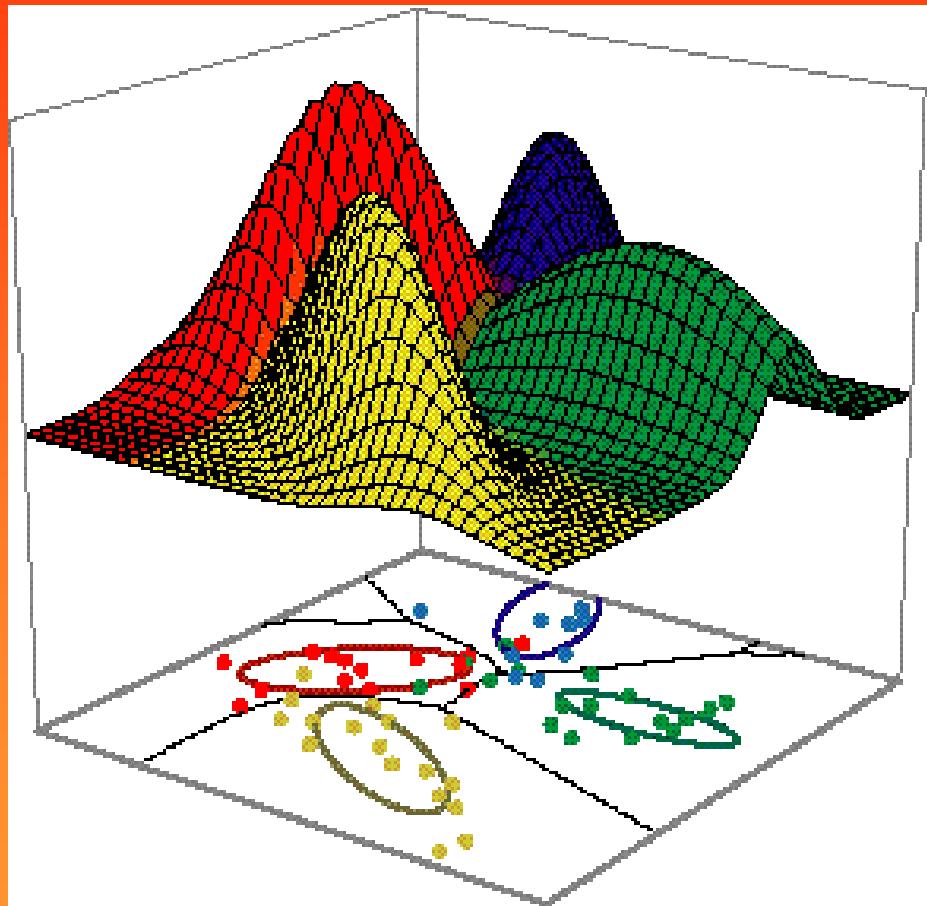


FIGURE 4.8. The decision boundaries in a two-dimensional Parzen-window dichotomizer depend on the window width h . At the left a small h leads to boundaries that are more complicated than for large h on same data set, shown at the right. Apparently, for these data a small h would be appropriate for the upper region, while a large h would be appropriate for the lower region; no single window width is ideal overall. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Pattern Classification



All materials in these slides
were taken from
*Pattern Classification (2nd
ed) by R. O. Duda, P. E. Hart
and D. G. Stork, John Wiley &
Sons, 2000*
with the permission of the
authors and the publisher

Non-Parametric Classification (Sections 4.4-4.5)

- K_n –Nearest Neighbor Estimation
- The Nearest-Neighbor Rule



4.4 K_n - Nearest neighbor estimation

- Goal: a solution for the problem of the unknown "best" window function
 - Let the cell volume be a function of the training data (k_n)
 - Center a cell about x and let it grow until it captures k_n samples ($k_n = f(n)$)
 - k_n are called the k_n nearest-neighbors of x



2 possibilities can occur:

- Density is high near x ; therefore the cell will be small which provides a good resolution
- Density is low; therefore the cell will grow large and stop until higher density regions are reached

We can obtain a family of estimates by setting $k_n = k_1 \sqrt{n}$ and choosing different values for k_1



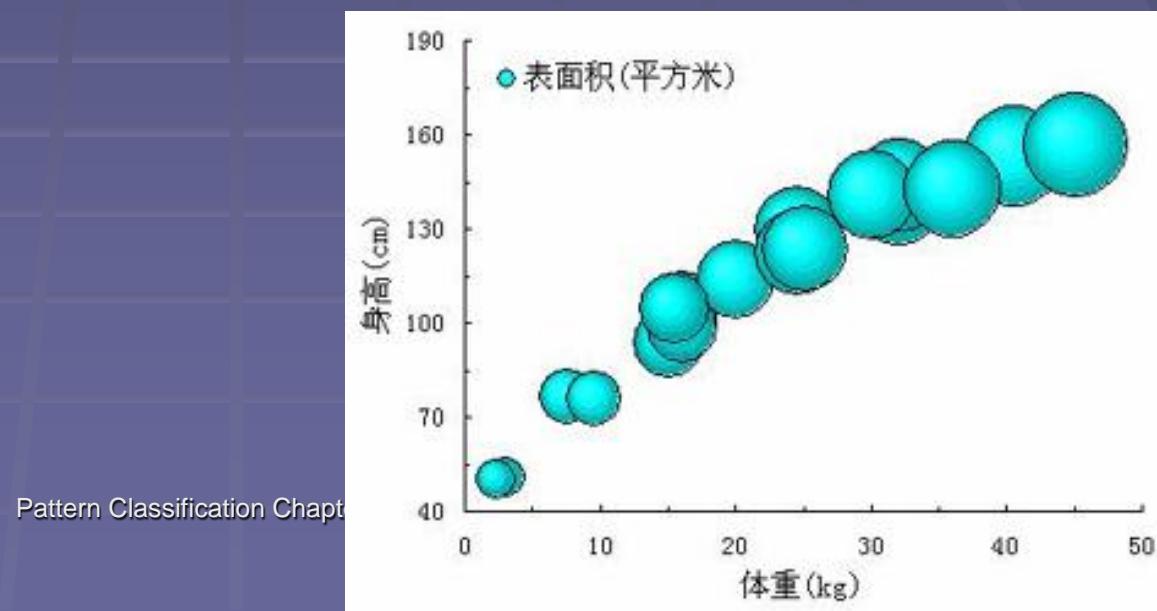
tern Classification Chapter 4 part 2

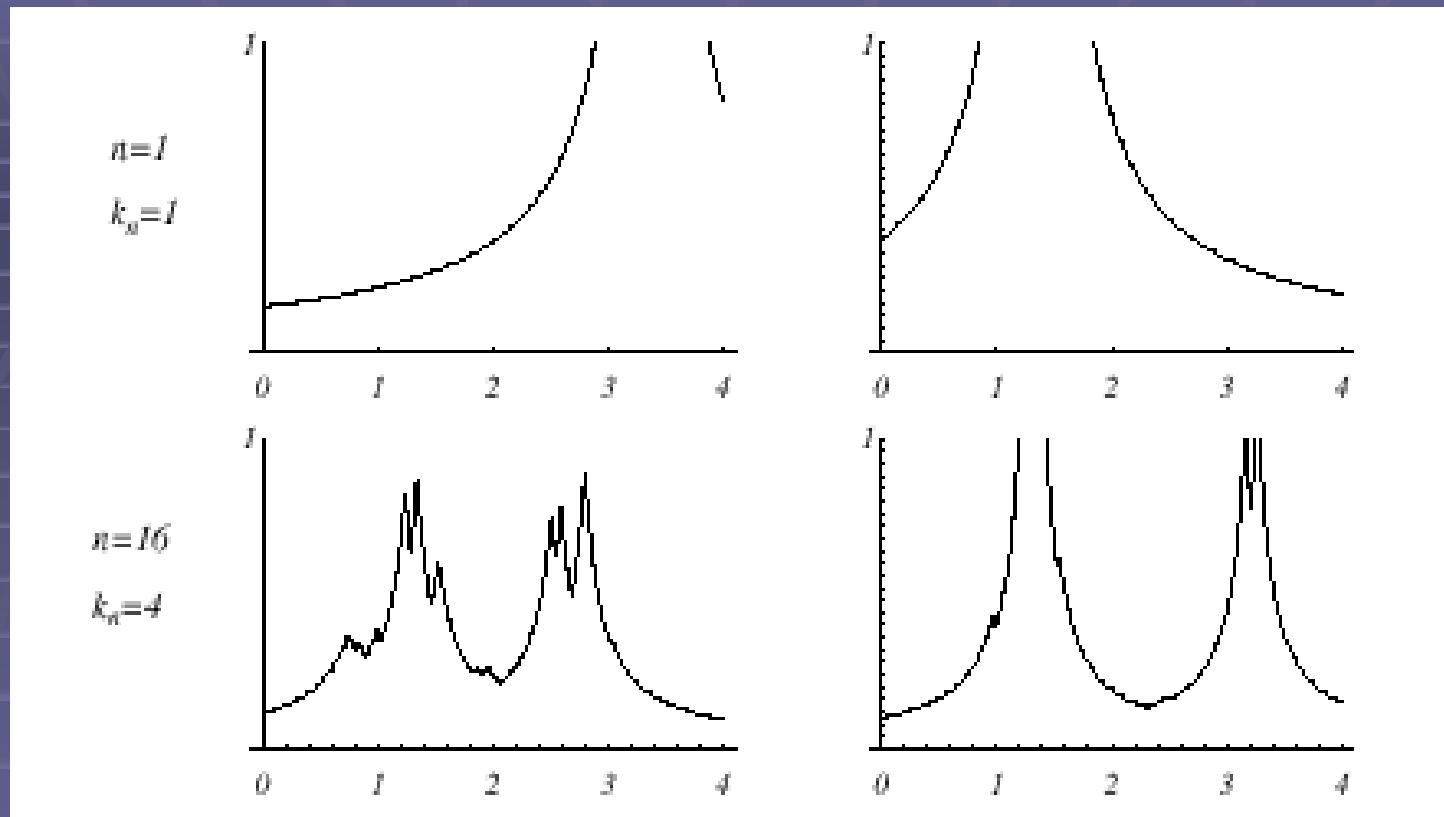


Illustration

For $k_n = \sqrt{n} = 1$; the estimate becomes:

$$P_n(x) = k_n / n/V_n = 1 / V_1 = 1 / 2|x-x_1|$$





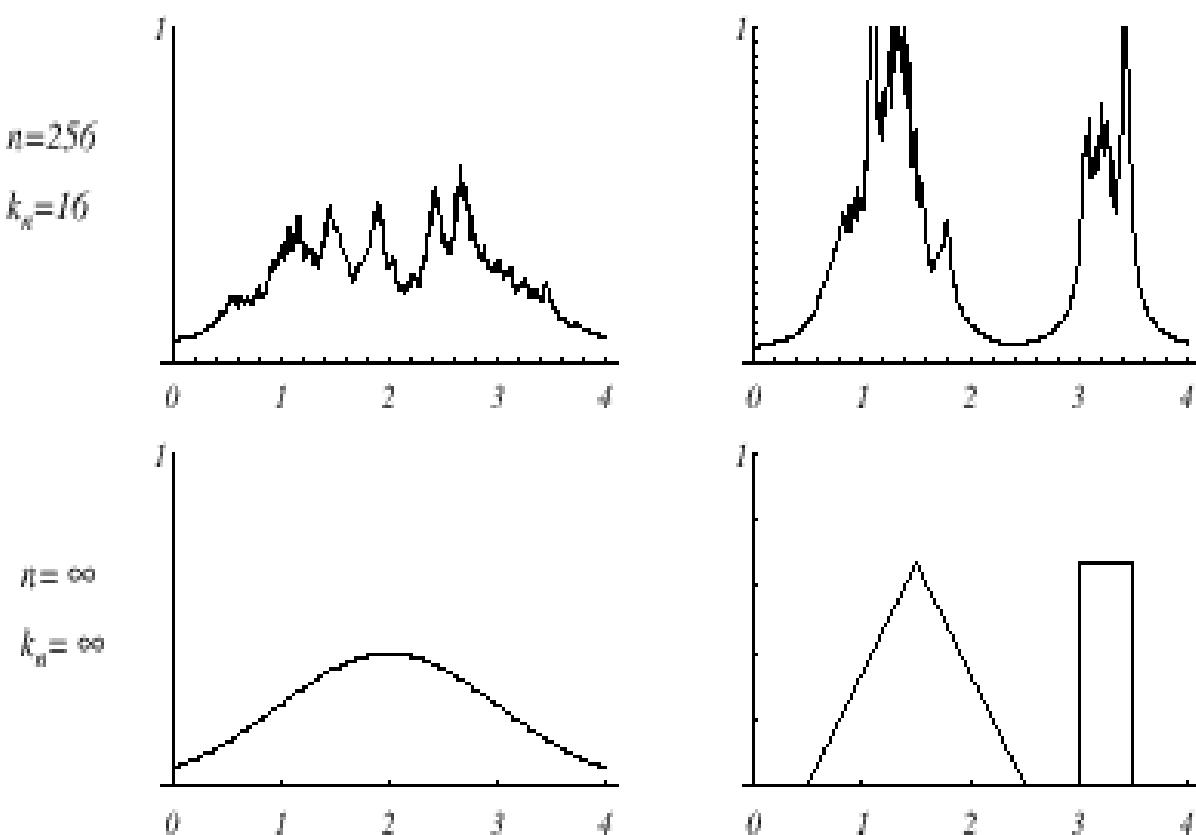
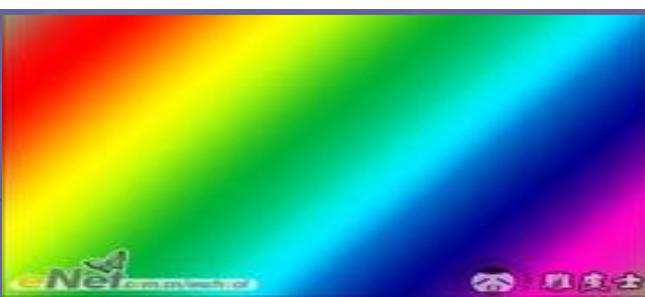


FIGURE 4.12. Several k -nearest-neighbor estimates of two unidimensional densities: a Gaussian and a bimodal distribution. Notice how the finite n estimates can be quite "spiky." From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.



Pattern Classification Chapter 4



- Estimation of a-posteriori probabilities
 - Goal: estimate $P(\omega_i | x)$ from a set of n labeled samples
 - Let's place a cell of volume V around x and capture k samples
 - k_i samples amongst k turned out to be labeled by ω_i then:

$$p_n(x, \omega_i) = k_i / (nV)$$

An estimate for $p_n(\omega_i | x)$ is:

$$p_n(\omega_i | x) = \frac{p_n(x, \omega_i)}{\sum_{j=1}^k p_n(x, \omega_j)} = \frac{k_i}{k}$$



- k_i/k is the fraction of the samples within the cell that are labeled as ω_i
- For minimum error rate, the most frequently represented category within the cell is selected
- If k is large and the cell sufficiently small, the performance will **approach the best value**.



4.5 The Nearest-Neighbor Rule

■ The nearest –neighbor rule

- Let $D_n = \{x_1, x_2, \dots, x_n\}$ be a set of n labeled prototypes
- Let $x' \in D_n$ be the closest prototype to a test point x then the nearest-neighbor rule for classifying x is to assign it the label associated with x'
- The nearest-neighbor rule leads to an error rate greater than the minimum possible value of the Bayes rate
- If the number of prototypes is large (unlimited), the error rate of the nearest-neighbor classifier is never worse than twice the Bayes rate (it can be demonstrated!)
- If $n \rightarrow \infty$, it is always possible to find x' sufficiently close so that:

$$P(\omega_i | x') \cong P(\omega_i | x)$$

$$P(\omega_m | x) = \max_i P(\omega_i | x)$$

$$P^*(e | x) = 1 - P(\omega_m | x)$$

- If $P(\omega_m | x) \approx 1$, then the nearest neighbor selection is almost always the same as the Bayes selection
- Convergence of the Nearest Neighbor

$$P^*(e | x) = 1 - P(\omega_m | x)$$

$$P^*(e) = \int P^*(e | x) p(x) dx$$

$$P(e) = \lim_{n \rightarrow \infty} P_n(e)$$

$$P^*(e) \leq P(e) \leq P^*(e) \left(2 - \frac{c}{c-1} P^*(e) \right)$$



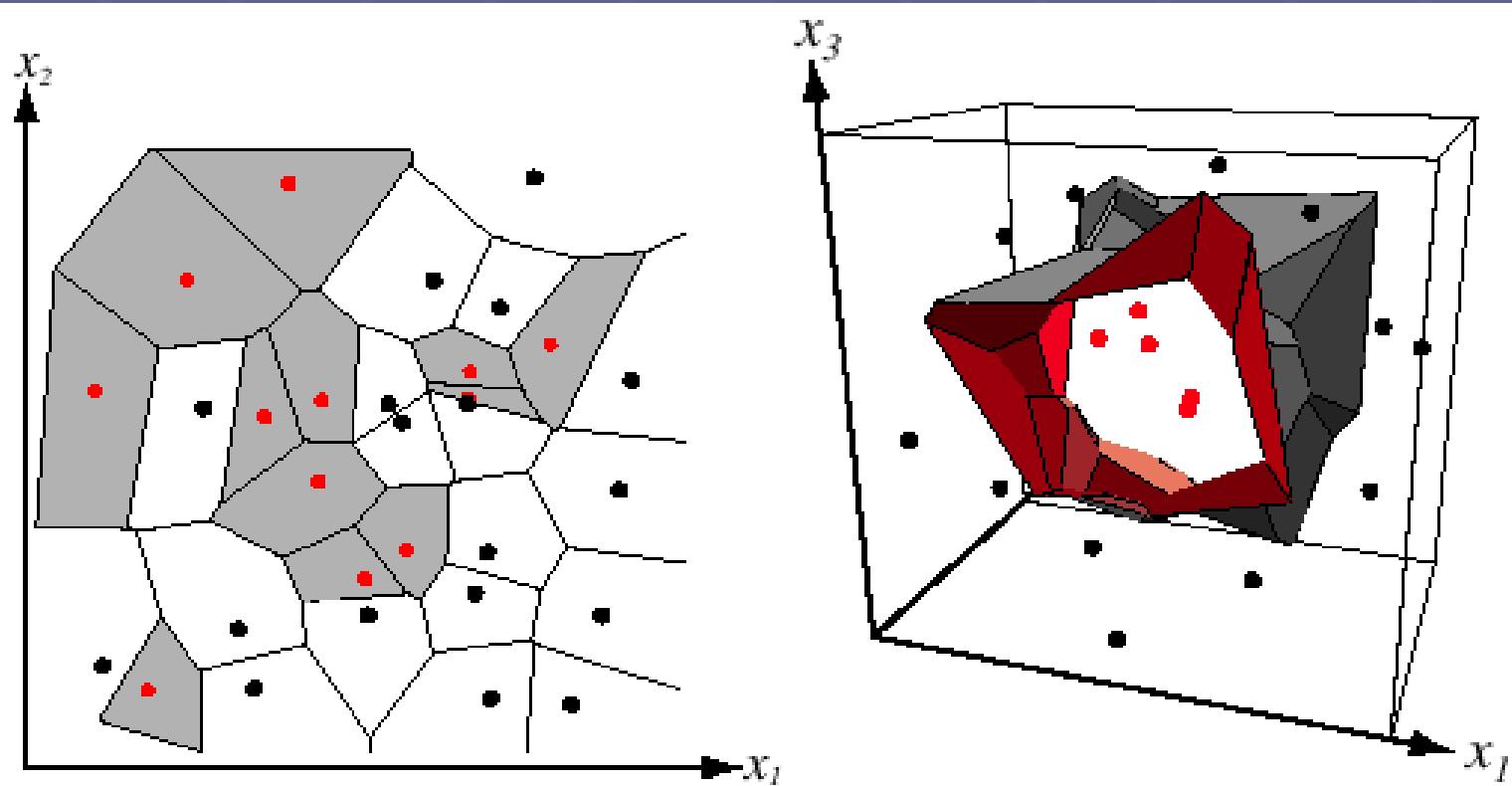


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

- The k – nearest-neighbor rule
- **Goal:** Classify x by assigning it the label most frequently represented among the k nearest samples and use a voting scheme



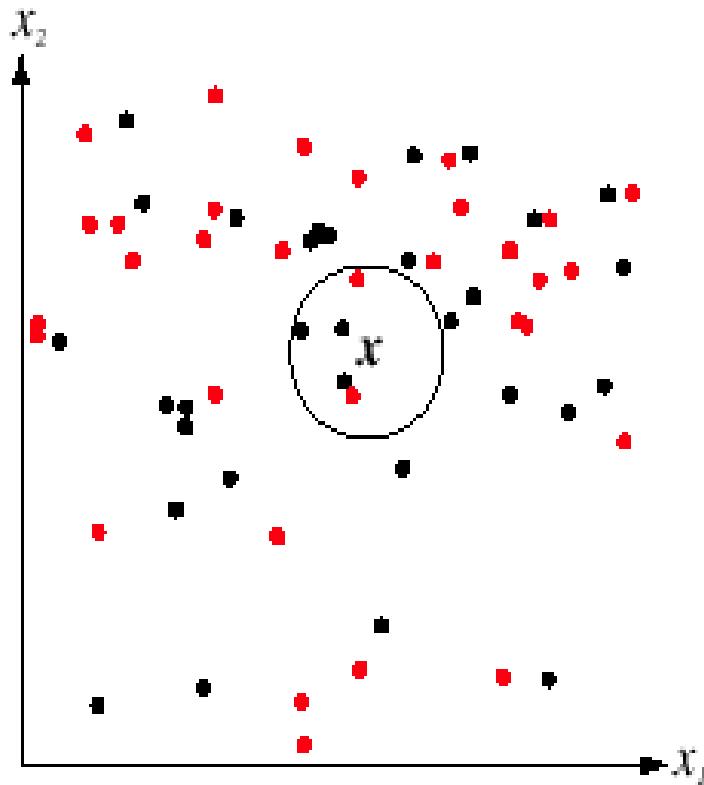


FIGURE 4.15. The k -nearest-neighbor query starts at the test point x and grows a spherical region until it encloses k training samples, and it labels the test point by a majority vote of these samples. In this $k = 5$ case, the test point x would be labeled the category of the black points. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Example:

$k = 3$ (odd value) and $x = (0.10, 0.25)^t$

Prototypes	Labels
(0.15, 0.35)	ω_1
(0.10, 0.28)	ω_2
(0.09, 0.30)	ω_5
(0.12, 0.20)	ω_2



Closest vectors to x with their labels are:

$$\{(0.10, 0.28, \omega_2); (0.12, 0.20, \omega_2); (0.15, 0.35, \omega_1)\}$$

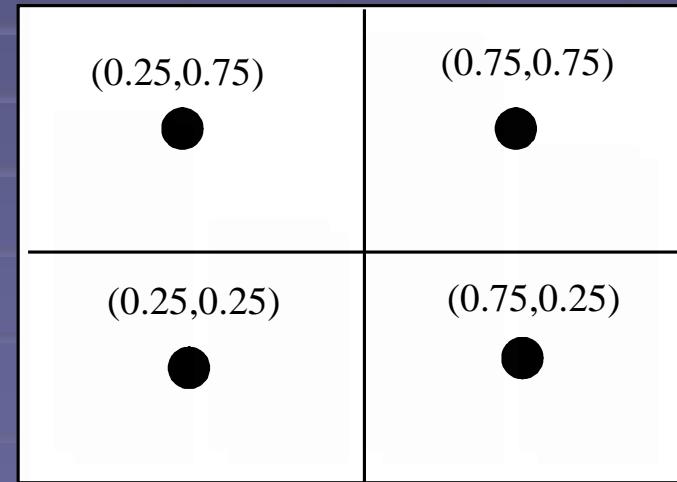
One voting scheme assigns the label ω_2 to x since ω_2 is the most frequently represented

- Reducing the computational complex in nearest-neighbor search
 - Computing partial distance

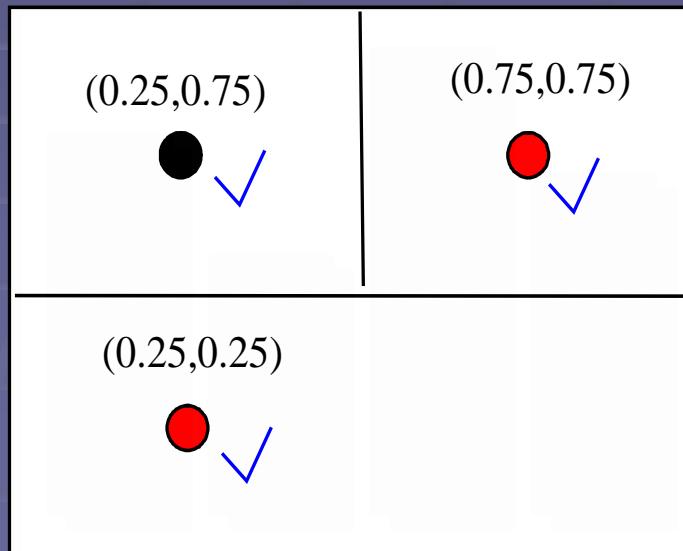
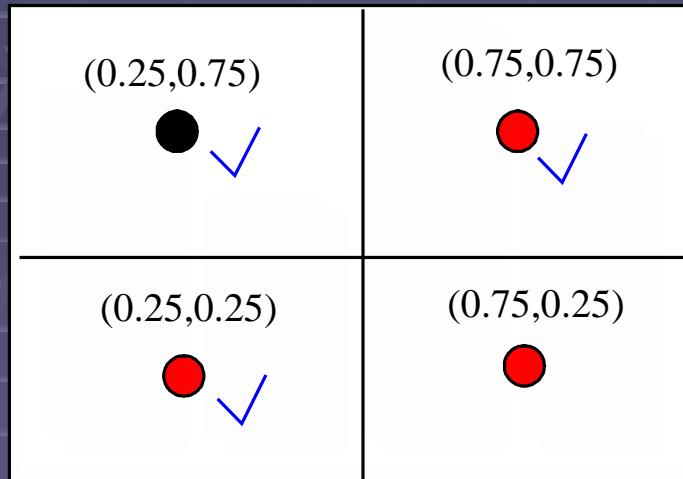
$$D_r(a, b) = \left(\sum_{k=1}^r (a_k - b_k)^2 \right)^{1/2}$$

- Create some form of search tree, for example

$$p(x) \sim U\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}\right)$$



▪ Nearest-Neighbor Editing



4.6 Metrics

Properties of Metrics

- Nonnegativity: $D(a,b) \geq 0$
- Reflexivity: $D(a,b) = 0$ if and only if $a=b$
- Symmetry: $D(a,b) = D(b,a)$
- *Euclidean* formula for distance in d dimensions

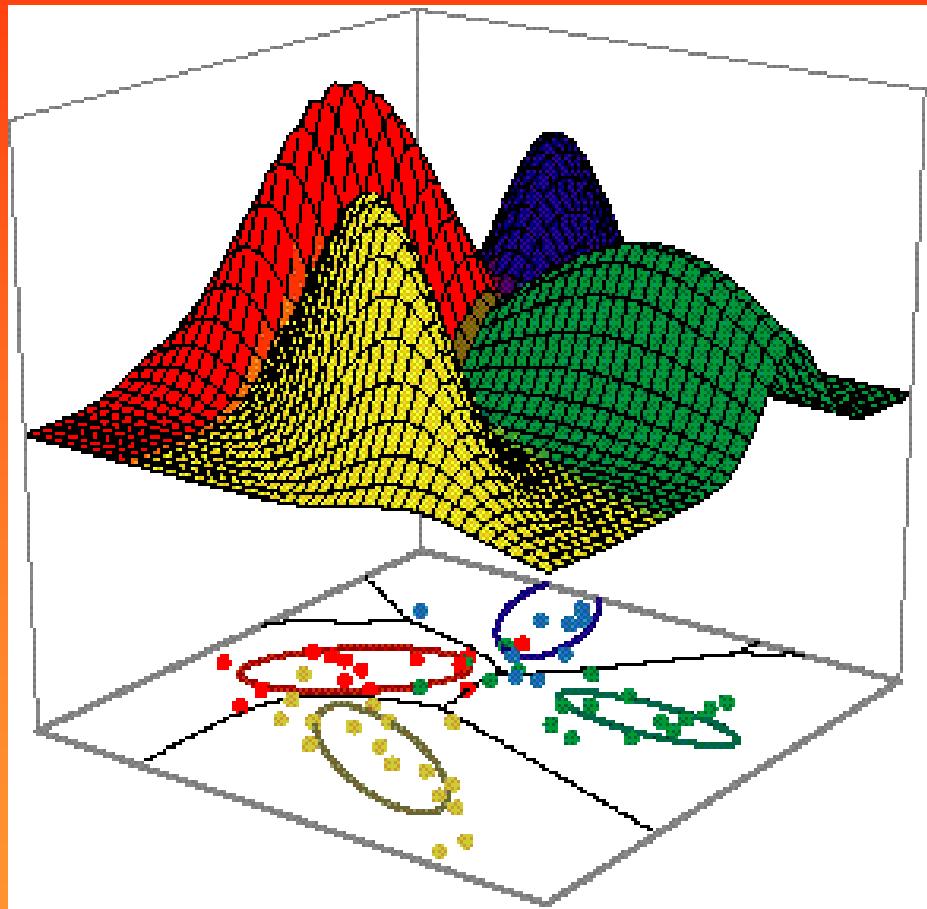
$$D(a,b) = \left(\sum_{k=1}^d (a_k - b_k)^2 \right)^{1/2}$$



Minkowski metric for distance in d dimensions

$$L_k(a,b) = \left(\sum_{k=1}^d (a_k - b_k)^k \right)^{1/k}$$

Pattern Classification

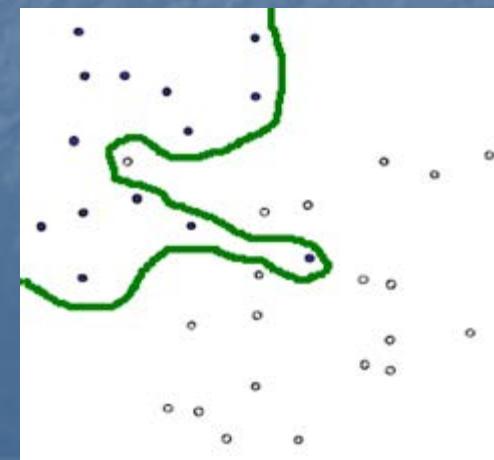
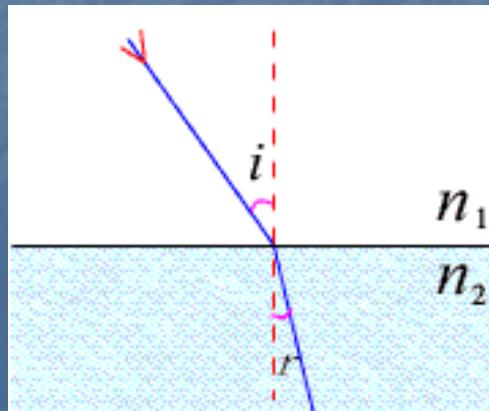
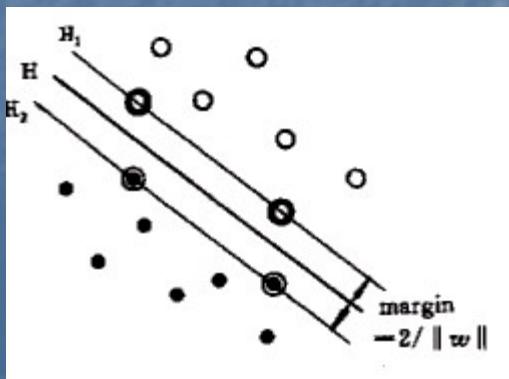


All materials in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors and the publisher

5.4 The Two-Category Linearly Separable Case

■ Linearly separable

n samples y_1, y_2, \dots, y_n belong to ω_1, ω_2 , if there exists a linear discriminant function $g(x) = a^t y$ that classifies all of them correctly, the samples are said to be linearly separable. Weight vector a is called a separating vector or solution vector



■ Normalization

for $y_i \in \omega_1$, $a^t y_i > 0$.

for $y_i \in \omega_2$, $a^t y_i < 0$.

Multiplying all samples labeled with
normalization



ω_2 by -1 is called

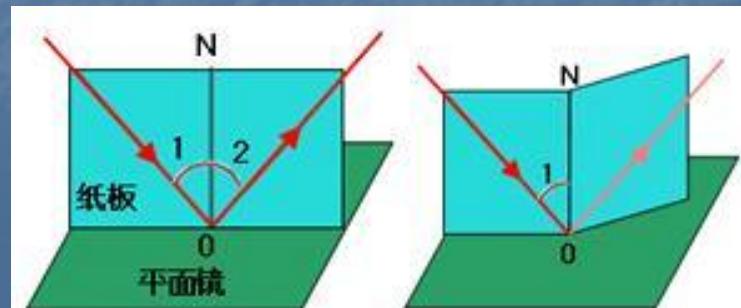
with this normalization we only need to look for a
weight vector a such that $a^t y_i > 0$.

■ Solution Region

■ Margin $a^t y_i \geq b > 0$

the distance between old boundaries and new
boundaries is

$$\frac{b}{\|y_i\|}$$



- The problem of finding a linear discriminant function will be formulated as a problem of minimizing a criterion function



■ Gradient Descent Procedures

define a criterion function $J(a)$ that is minimized if a is a solution vector. This can often be solved by a gradient descent procedure.

$$a(k+1) = a(k) - \eta(k) \nabla J(a(k))$$

η is a positive scale factor or learning rate that sets the step size

Using the Taylor extension, we have

$$\eta(k) = \frac{\|\nabla J\|^2}{\nabla J^t H \nabla J}$$

H is Hessian matrix, $\partial^2 J / \partial a_i \partial a_j$



- Another Algorithm
- :Newton Descent Algorithm

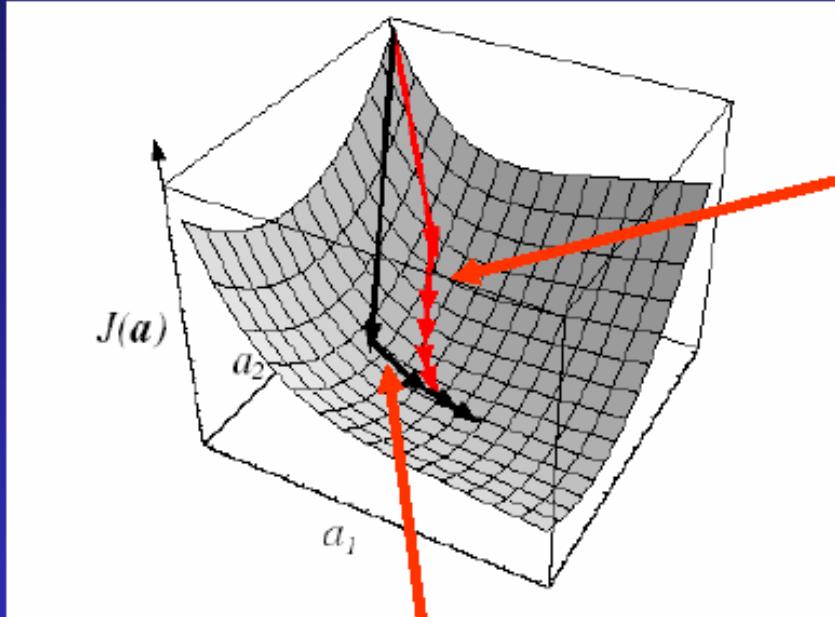
Using the Taylor extension and

let $\frac{\partial J(a(k+1))}{\partial a(k+1)} = 0$, we can get

$$a(k+1) = a(k) - H^{-1} \nabla J(a(k))$$

This the so-called Newton Descent Algorithm

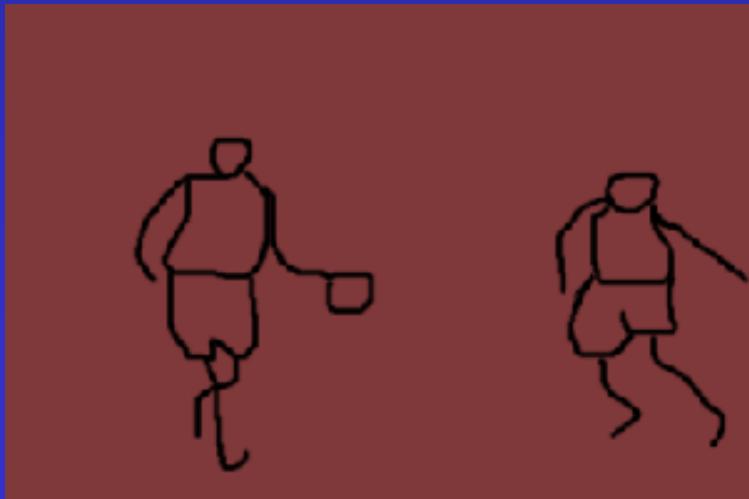
■ Newton's algorithm vs the simple gradient decent algorithm



Simple gradient descent method

Newton's second order method

Has greater improvement per step even
When using optimal learning rates for both
Method.
However added computational
Burden of inverting the Hessian matrix.



5.5 Minimizing The Perception Criterion Function

$J_p(a) = \sum_{y \in Y} (-a^T y)$, where $Y(a)$ is the set of samples misclassified by the Batch Perception Algorithm

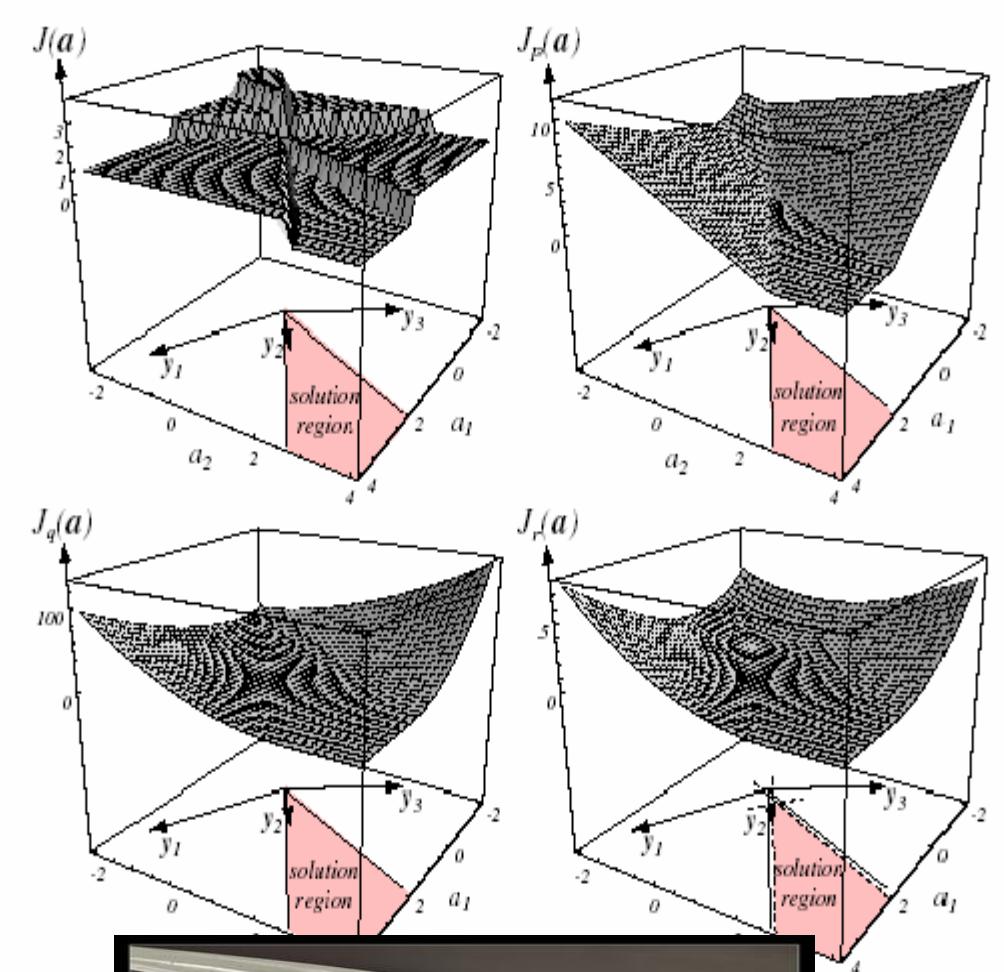
- The Perception Criterion Function of the Batch Perception Algorithm

$$\nabla J_p = \sum_{y \in Y} -y$$

$$a(k+1) = a(k) + \eta(k) \sum_{y \in Y} y$$

■ Comparison of Four Criterion functions

No of misclassified samples:
Piecewise constant,
unacceptable



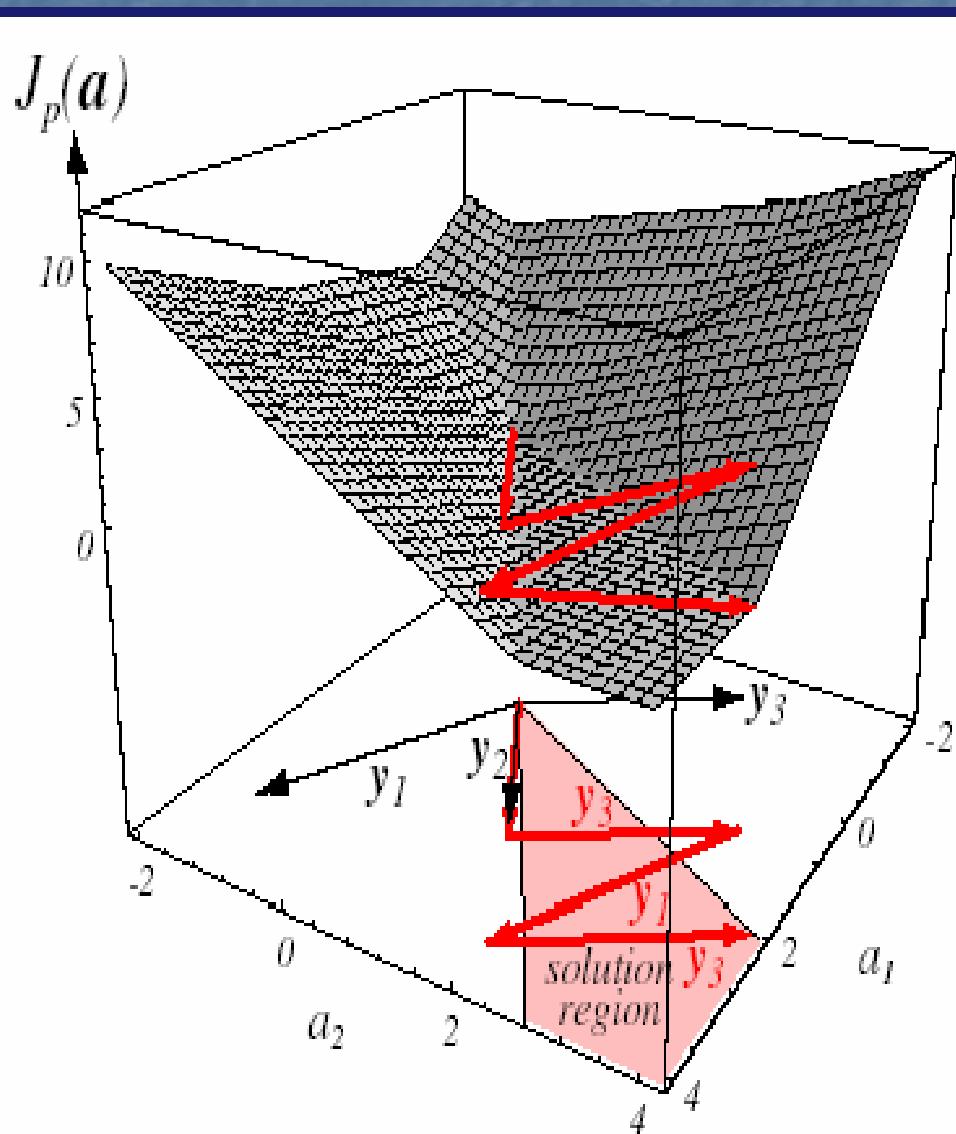
Perceptron criterion:
Piecewise linear,
acceptable for
gradient descent

Squared error:
Useful when patterns
are not linearly separable



Squared Error
with margin

■ Perceptron Criterion as function of weights: Demo



Criterion function plotted as a function of weights a_1 and a_2 ,
Starts at origin,
Sequence is y_2, y_3, y_1, y_3
Second update by y_3 takes
solution farther than
first update

■ Single-Sample Correction

$a(1)$ arbitrary

$$a(k+1) = a(k) + \eta(k)y^k$$

■ Interpretation of Single-Sample Correction

Single-Sample correction and batch perception algorithm are two algorithms of perception machine.

Either of them can be used !

- Some Direct Generalizations
 - Variable-Increment Perceptron with Margin

$a(1)$ arbitrary

$$a(k+1) = a(k) + \eta(k)y^k \quad k \geq 1 \text{ where } a^t(k)y^k \leq b \text{ for all } k$$

- Algorithm Convergence
- Batch Variable Increment Perception

$a(1)$ arbitrary

$$a(k+1) = a(k) + \eta(k) \sum_{y \in Y_k} y$$

5.6 Relaxation Procedures

- Decent Algorithm

- criterion function:

$$J_q(a) = \sum_{y \in Y} (a^t y)^2$$

- two problems of this criterion function
 - criterion function:

$$J_r(a) = \frac{1}{2} \sum_{y \in Y} \frac{(a^t y - b)^2}{\|y\|^2}$$

where $Y(a)$ is a set of samples for which $a^t y \leq b$

- Batch Relaxation with Margin
- Single-Sample Relaxation with margin
- Geometrical interpretation of Single-Sample Relaxation with margin algorithm

$$\frac{b - a^t y^k}{\|y^k\|^2} y^k = \frac{b - a^t y^k}{\|y^k\|} \times \frac{y^k}{\|y^k\|} = r(k) \times \frac{y^k}{\|y^k\|}$$

$r(k)$ is the distance from $a(k)$ to the hyperplane $a^t y^k = b$

- From Eq.35 we obtain

$$a^t (k+1) y^k - b = (1 - \eta)(a^t (k) y^k - b)$$

5.7 Nonseparable Behavior

- Error-correction procedure
- For nonseparable data the corrections in an error-correction procedure can never cease.
- By averaging the weight vector produced by the correction rule, we can reduce the risk of obtaining a bad solution.
- Some heuristic methods are used in the error-correction rules. The goal is to obtain acceptable performance on nonseparable problems while preserving the ability to find a separating vector on separable problems.
- Usually we let $\eta(k)$ approach zero as k approaches infinity

5.8 Minimum Squared Error Procedures

- Criterion function involves *all* of the samples, not just misclassified ones
- Previously we were interested in making all of the inner products $a^t y_i$ positive
- Now try to make $a^t y_i = b_i$ where b_i are some arbitrarily specified positive constants



5.8 Minimum Squared Error Procedures

- Thus replace the problem of solving a set of linear inequalities with more stringent but better understood problem of finding a solution to a set of linear equations



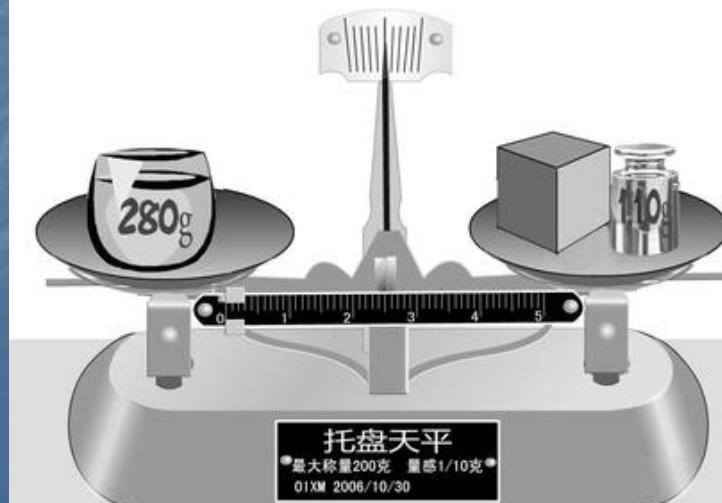
■ Minimum Squared Error and Pseudoinverse

For all the samples y_1, y_2, \dots, y_n we want a weight vector a so that $a^T y_i = b_i$ for some arbitrarily specified positive numbers. The matrix notation :

$$\begin{pmatrix} y_{10} & y_{11} & \cdots & y_{1d} \\ y_{20} & y_{21} & \cdots & y_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ y_{n0} & y_{n1} & \cdots & y_{nd} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \Leftrightarrow Ya = b$$

Error vector:

$$e = Ya - b$$



■ Sum-of-squared-error criterion function:

$$J_s(a) = \|Ya - b\|^2 = \sum_{i=1}^n (a^t y_i - b_i)^2$$

■ The gradient

$$\nabla J_s = \sum_{i=1}^n 2(a^t y_i - b_i) y_i = 2Y^t(Ya - b)$$

Set it to zero, we get

$$Y^t Ya = Y^t b$$

If $Y^t Y$ is nonsingular, $a = (Y^t Y)^{-1} Y^t b = Y^+ b$

The d by n matrix Y^+ is called the pseudoinverse of Y.

■ Remarks: For an arbitrarily fixed b, MSE solution may not be a separating vector.



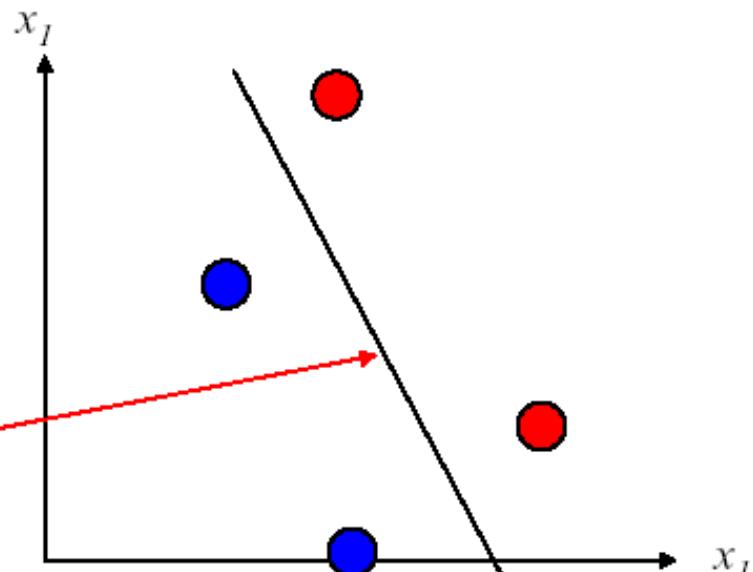
Example of Linear Classifier by Pseudoinverse

- $\omega_1: (1,2)^t$ and $(2,0)^t$
- $\omega_2: (3,1)^t$ and $(2,3)^t$

Sample Matrix ($d = 1+2$, $n = 4$)

$$Y = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 0 \\ -1 & -3 & -1 \\ -1 & -2 & -3 \end{bmatrix}$$

$$a^t \begin{pmatrix} 1 \\ x_1 \\ x_2 \end{pmatrix} = 0$$



Pseudo-inverse

$$Y^* = (Y^t Y)^{-1} Y^t = \begin{bmatrix} 5/4 & 13/12 & 3/4 & 7/12 \\ -1/2 & -1/6 & -1/2 & -1/6 \\ 0 & -1/3 & 0 & -1/3 \end{bmatrix}$$

Assuming $b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

our solution is $a = Y^t b = \begin{bmatrix} 11/3 \\ -4/3 \\ -2/3 \end{bmatrix}$

How to classify new samples (test samples)?

$$a.y > 0$$

→ First class

$$a.y < 0$$

→ Second class

y

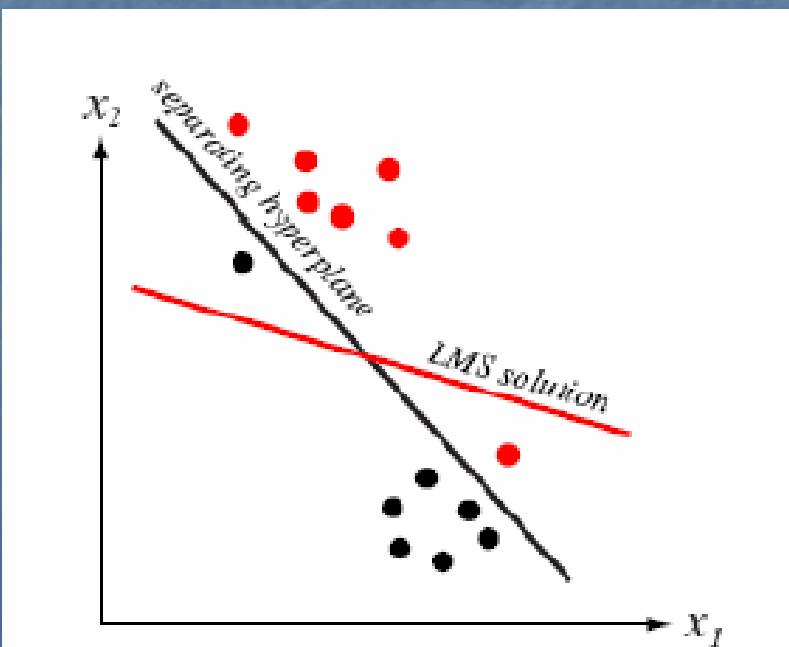
: new sample



■ The Widrow-Hoff or LMS Procedure

(1) Iterative procedure: no matrix inverse

(2) Need not converge to a separating hyperplane even if there exist one



5.9 The Ho-Kashyap Procedure

- Take the criterion function as a function of two variables a and b :

$$J_s(a, b) = \|Ya - b\|^2, \text{ where } b > 0$$

- If the training samples are linearly separable, then there should exist an \hat{a} and \hat{b} such that: $Y\hat{a} = \hat{b} > 0$

If we knew such \hat{b} beforehand. We would get the separating vector \hat{a} using the MSE procedure

$$\nabla_a J_s = 2Y^t(Ya - b)$$

$$\nabla_b J_s = -2(Ya - b)$$

$$a = Y^+ b$$

$$b(k+1) = b(k) - \eta \frac{1}{2} [\nabla_b J_s - |\nabla_b J_s|]$$

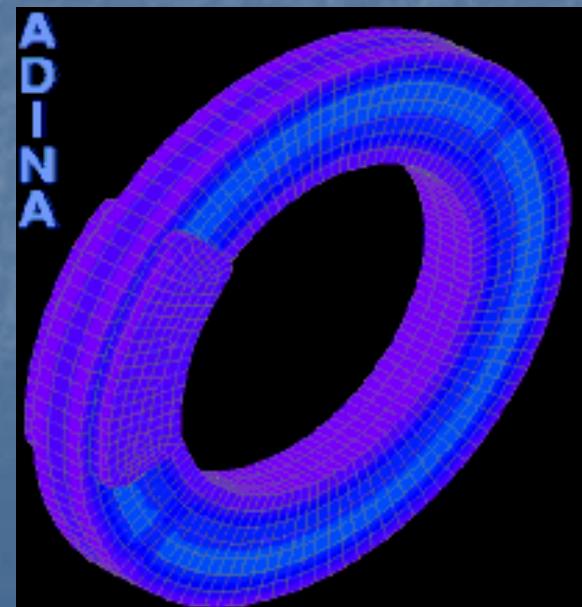


Ho-Kashyap Procedure

$$b(1) > 0$$

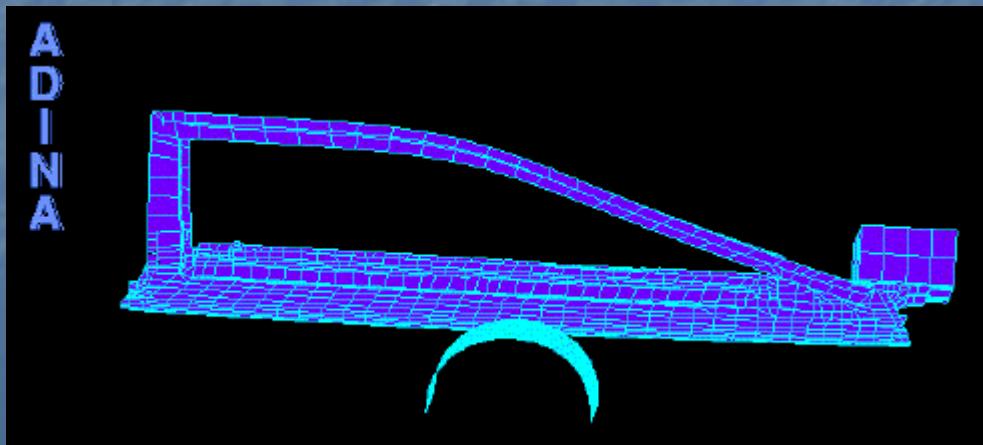
$$b(k+1) = a(k) + 2\eta(k)e+(k)$$

$$e+(k) = (e(k) + |e(k)|)/2$$



$$e(k) = Ya(k) - b(k)$$

$$a(k) = \text{inv}(Y'Y)Y'b(k)$$



■ 5.12 Multicategory Generalizations



■ Generalization for MSE Procedure

consider multiclass case as a set of c two-class problems

$$a_i^t y = 1 \text{ for all } y \in Y_i$$

$$a_i^t y = 0 \text{ for all } y \notin Y_i$$

$$A = [a_1 \quad a_2 \quad \dots \quad a_c] = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{c1} \\ a_{12} & a_{22} & \dots & a_{c2} \\ \dots & \dots & \dots & \dots \\ a_{1d} & a_{2d} & \dots & a_{cd} \end{bmatrix}$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_c \end{bmatrix} = \begin{bmatrix} y_{111} & y_{112} & \dots & y_{11d} \\ y_{121} & y_{122} & \dots & y_{12d} \\ \dots & \dots & \dots & \dots \\ y_{211} & y_{212} & \dots & y_{21d} \\ y_{221} & y_{222} & \dots & y_{22d} \\ \dots & \dots & \dots & \dots \\ y_{c11} & y_{c12} & \dots & y_{c1d} \\ y_{c21} & y_{c22} & \dots & y_{c2d} \end{bmatrix}$$



■ Generalization for MSE Procedure

consider multiclass case as a set of c two-class problem

$$B = \begin{bmatrix} B_1 \\ B_2 \\ \dots \\ B_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 1 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 1 \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

$$YA = B$$

$$A = Y^+ B$$

$$= \text{inv}(Y'Y)Y' B$$

Welcome to my Home



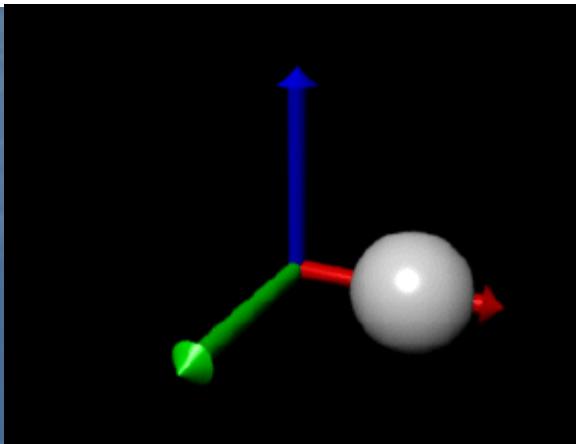
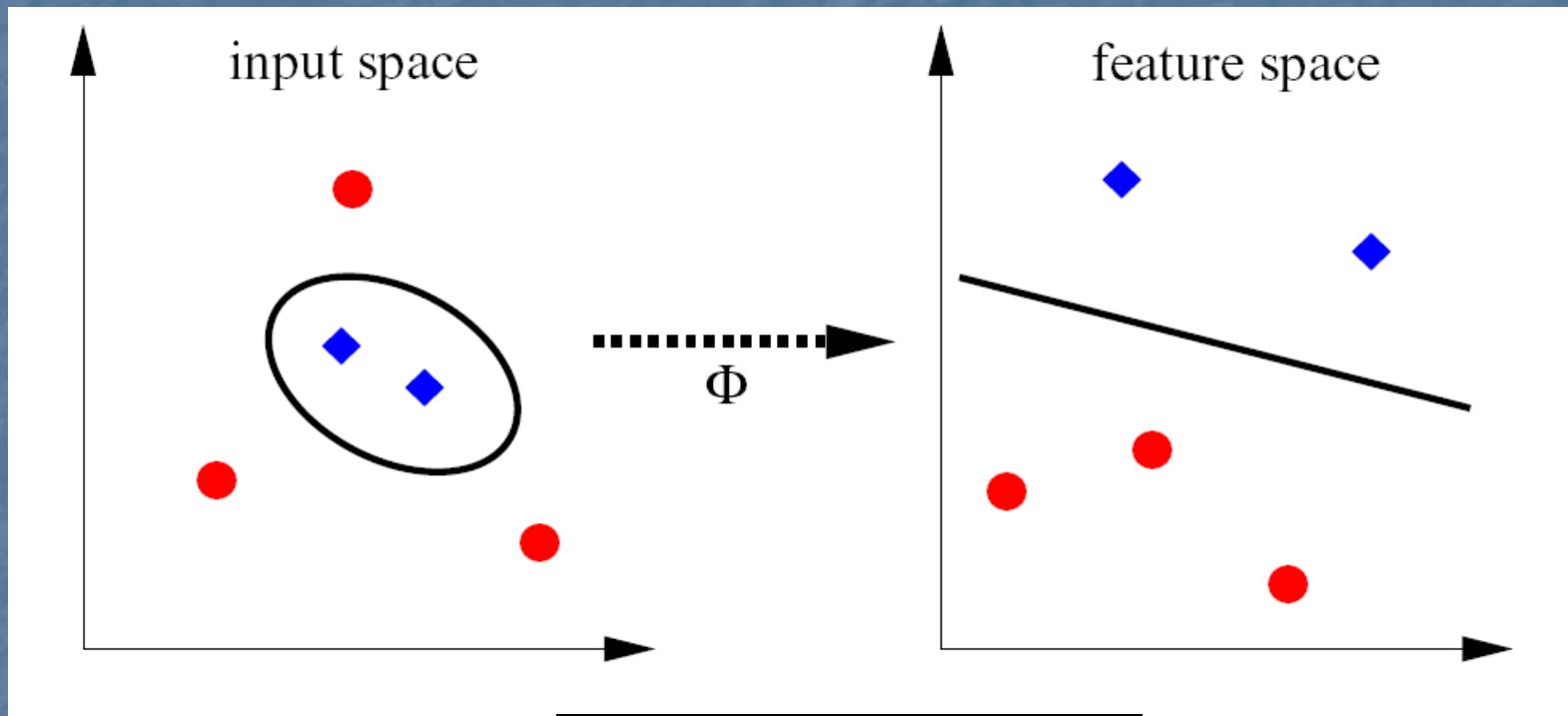
Nonlinear Minimum Squared Error Procedures:

Just for your reference

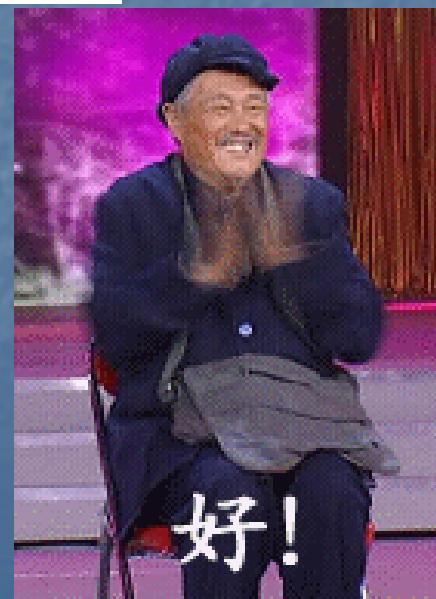
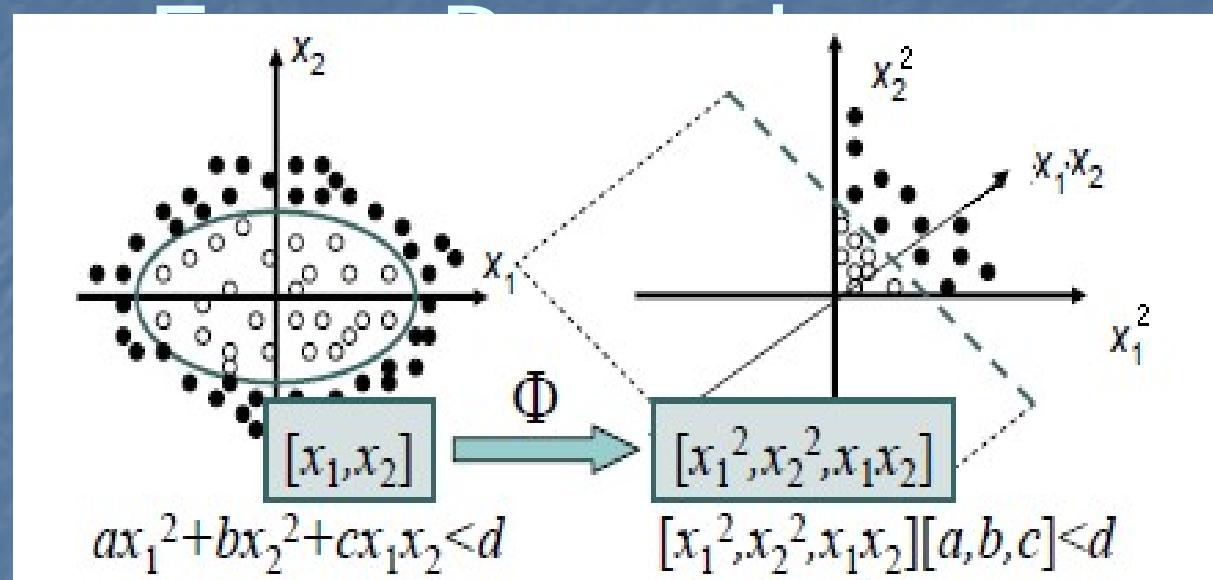
- Recently proposed new method
- Extension of Minimum Squared Error Procedures
- Equivalent to the Minimum Squared Error Procedures in feature space



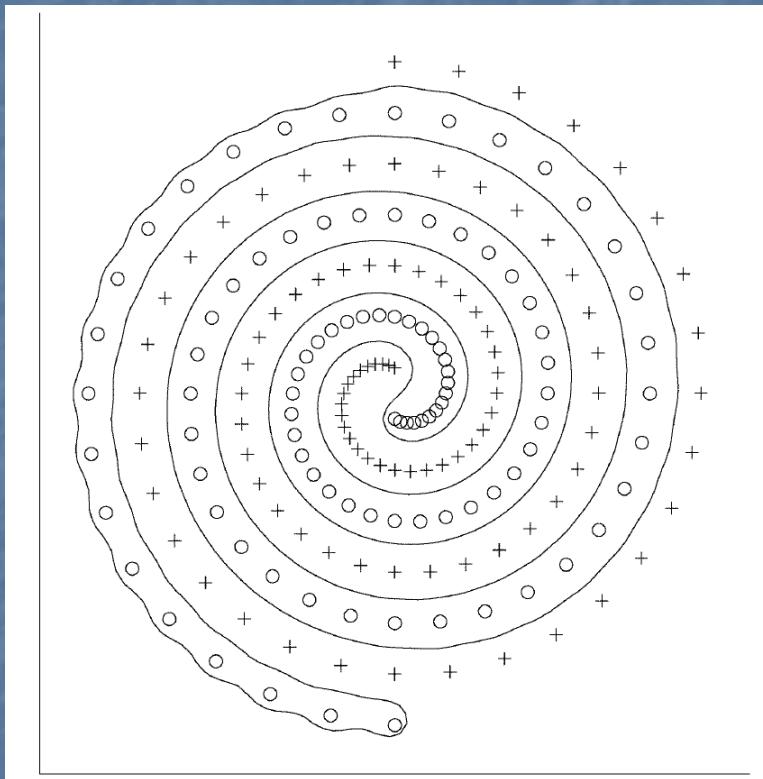
Nonlinear transform



Nonlinear Minimum Squared



Nonlinear Minimum Squared Error Procedures



Nonlinear Minimum Squared Error Procedures

Original Minimum Squared Error procedure in the original space:

$$\begin{pmatrix} y_{10} & y_{11} & \cdots & y_{1d} \\ y_{20} & y_{21} & \cdots & y_{2d} \\ \cdots & \cdots & \cdots & \cdots \\ y_{n0} & y_{n1} & \cdots & y_{nd} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \cdots \\ -1 \end{pmatrix} \Leftrightarrow Ya = B$$

Minimum Squared Error procedure in the new space:

$$Z\beta = B \quad Z = [Z_1 \dots Z_n] \quad Z_i = \varphi(Y_i)$$

Nonlinear Minimum Squared Error Procedures: KMSE

Because of

$$\beta = \sum_{j=1,\dots,n} \gamma_j \varphi(Y_j) \quad \varphi(Y_i)^T \varphi(Y_j) = k(Y_i, Y_j)$$

we have

$$K\gamma = B$$

$$K = \begin{pmatrix} k(Y_1, Y_1) & k(Y_1, Y_2) & \dots & k(Y_1, Y_n) \\ k(Y_2, Y_1) & k(Y_2, Y_2) & \dots & k(Y_2, Y_n) \\ \dots & \dots & \dots & \dots \\ k(Y_n, Y_1) & k(Y_n, Y_2) & \dots & k(Y_n, Y_n) \end{pmatrix}$$

Nonlinear Minimum Squared Error Procedures: KMSE

- Kernel functions:

- (1)
$$k(Y_i, Y_j) = \exp\left(-\frac{\|Y_i - Y_j\|^2}{\sigma^2}\right)$$

- (2)
$$k(Y_i, Y_j) = (Y_i^T Y_j + c)^d$$

Nonlinear Minimum Squared Error Procedures: KMSE

- Training phase:

- Obtain $\gamma = K^{-1}B$

Testing phase (b is the output of testing sample Y):

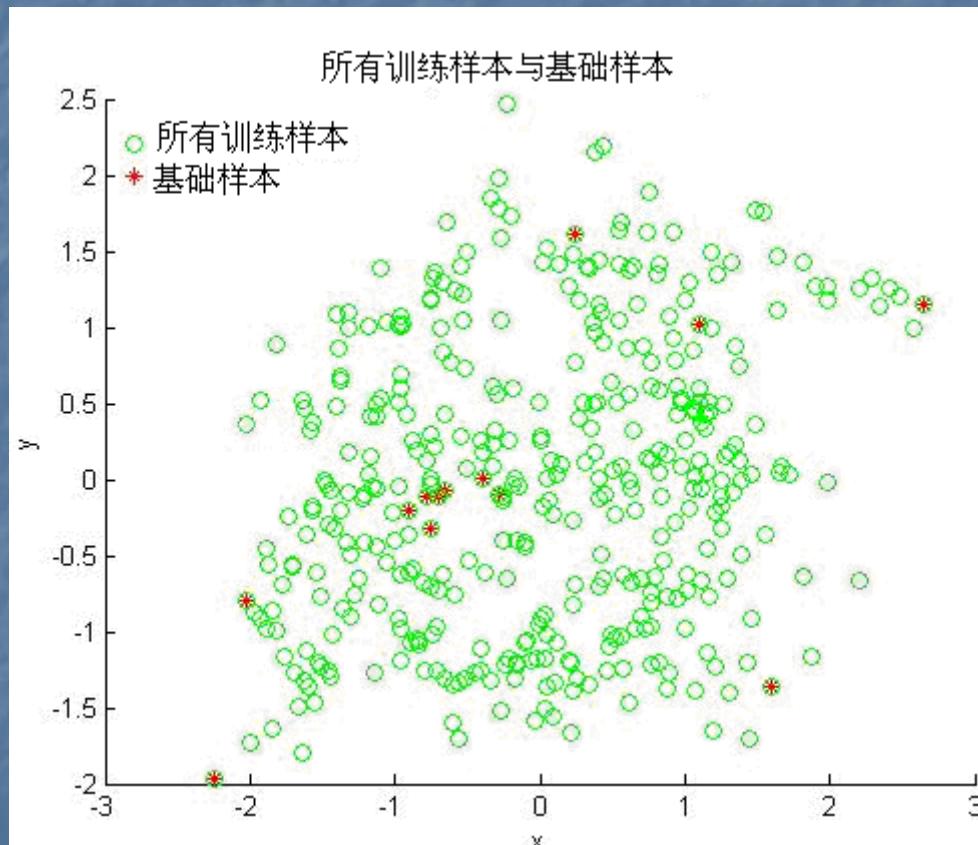
$$b = \sum_{i=1}^n \gamma_i k(Y_i, Y)$$

If b is closer to 1 than -1, then the testing sample is classified into the first class.
otherwise, it is classified into the second class.

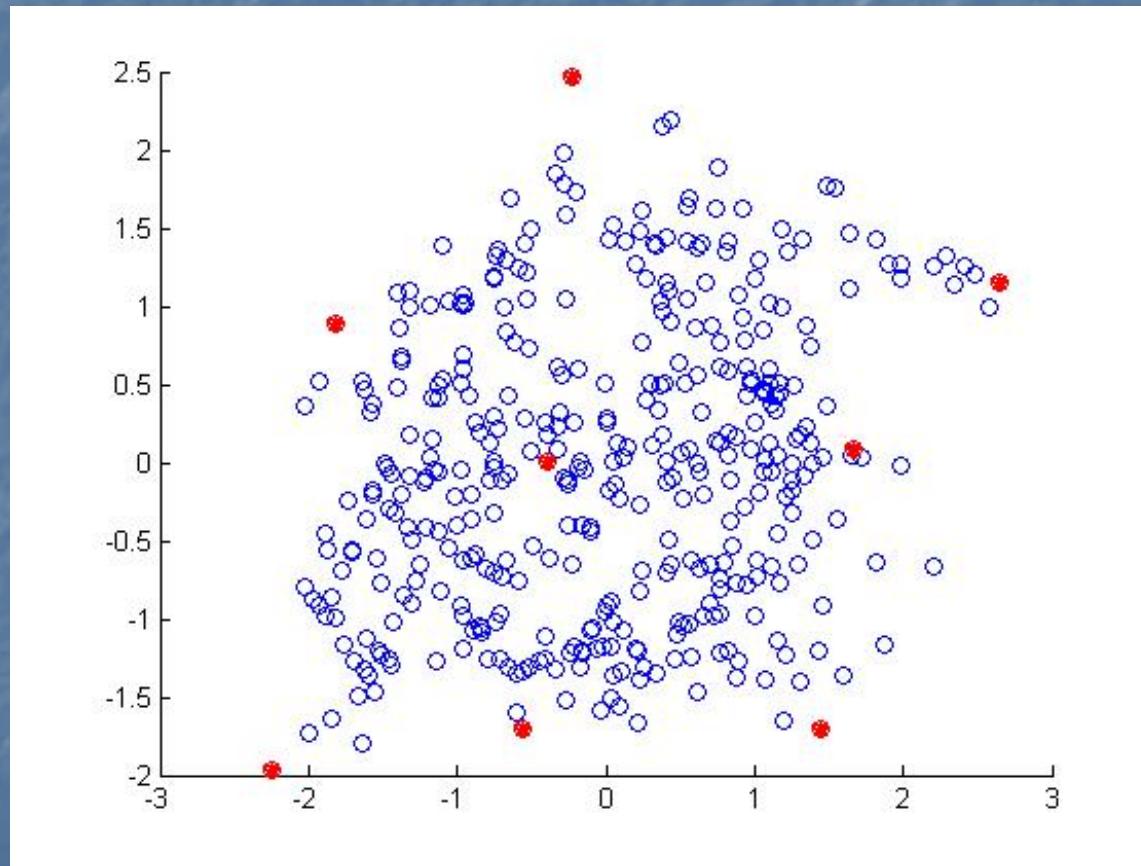
Disadvantage of and improvement to KMSE

- The more the training samples, the higher the computational complexity !
- If $\beta = \sum_{j=1,\dots,s} \gamma_j \varphi(Y_j), s \ll n$
- Then $b = \sum_{i=1}^s \gamma_i k(Y_i, Y)$ and the computational complexity will be greatly reduced.

Disadvantage of and improvement to KMSE: one improvement



Disadvantage of and improvement to KMSE: another improvement

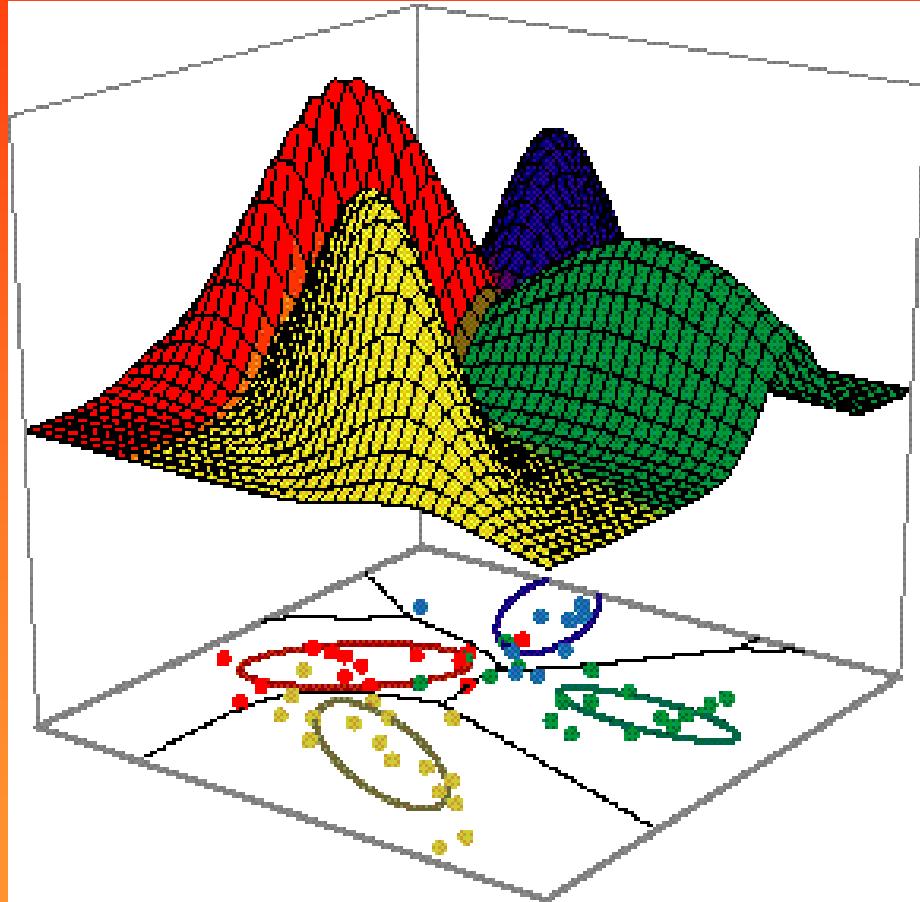


Nonlinear Minimum Squared Error Procedures : KMSE

■ References

- Yong Xu, David Zhang, Zhong Jin, Miao Li, Jing-Yu Yang, A fast kernel-based nonlinear discriminant analysis for multi-class problems, Pattern Recognition, 2006, 39(6) : 1026-1033.
- Yong Xu, J.-Y. Yang, J.-F. Lu, An efficient kernel-based nonlinear regression method for two-class classification, Proceedings of 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, August, 2005, pp.4442-4445.
- Yong Xu, David Zhang , Fengxi Song, Jing-Yu Yang, Zhong Jing , Miao Li, A method for speeding up feature extraction based on KPCA, Neurocomputing, 70, 1056-1061, 2007
- 徐勇,张大鹏,杨健,模式识别中的核方法及其应用,北京:国防工业出版社(优秀图书二等奖),2010

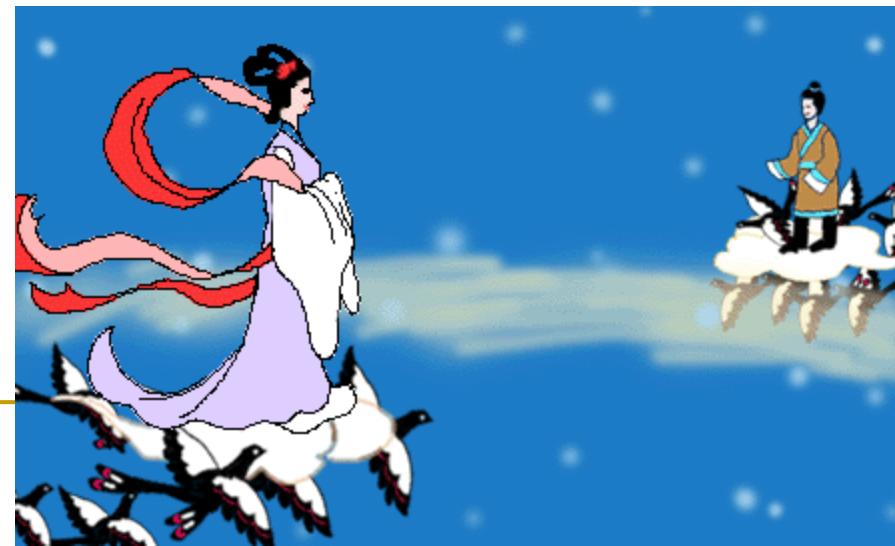
Pattern Classification



All materials in these slides were taken from
Pattern Classification (2nd ed) by R. O. Duda, P. E. Hart and D. G. Stork, John Wiley & Sons, 2000
with the permission of the authors and the publisher

5.8.2 Relation to Fisher's Linear Discriminant

- With the proper choice of the vector b , the MSE discriminant function $a^t y$ is directly related to Fisher's linear discriminant (Chapter 3 Section 3.8.2).
- To do so, we use the linear rather than generalized linear discriminant function.



Assumption

- A set of n d -dimensional samples x_1, \dots, x_n , n_1 of which are in the subset D_1 labeled w_1 , and n_2 of which are in the subset D_2 labeled w_2 .
- A sample y_i is formed from x_i by adding a threshold component $x_0=1$ to make an augmented pattern vector.
- If the sample is labeled w_2 , then the entire pattern vector is multiplied by -1
- With no loss in generality, assume the first n_1 samples are labeled w_1 and the second n_2 are labeled w_2 .

- MSE method based on $Y_a=b$ can be equivalent to Fisher's Linear Discriminant
- Condition: the number of the samples approaches to infinity.



Then the matrix Y can be partitioned as follows:

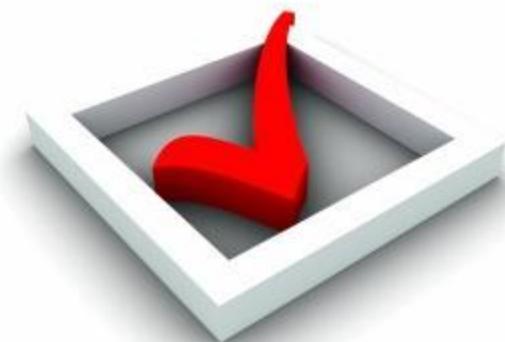
$$Y = \begin{bmatrix} 1_1 & X_1 \\ -1_2 & -X_2 \end{bmatrix},$$

Where 1_i is a column vector of n_i ones, and X_i is an n_i -by-d matrix whose rows are the samples labeled w_i .

Correspondingly,

$$a = \begin{bmatrix} \omega_0 \\ w \end{bmatrix}$$

$$b = \begin{bmatrix} \frac{n}{n_1} 1_1 \\ \frac{n}{n_1} 1_2 \end{bmatrix}$$



By Writing Eq.45 for a in terms of the partitioned matrices:

$$\begin{bmatrix} \mathbf{1}_1^t & -\mathbf{1}_2^t \\ \mathbf{X}_1^t & -\mathbf{X}_2^t \end{bmatrix} \begin{bmatrix} \mathbf{1}_1 & \mathbf{X}_1 \\ -\mathbf{1}_2 & -\mathbf{X}_2 \end{bmatrix} \begin{bmatrix} \omega_0 \\ \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_1^t & -\mathbf{1}_2^t \\ \mathbf{X}_1^t & -\mathbf{X}_2^t \end{bmatrix} \begin{bmatrix} \frac{n}{n_1} \mathbf{1}_1 \\ \frac{n}{n_2} \mathbf{1}_2 \end{bmatrix}. \quad (49)$$

Defining the sample means

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x \quad i = 1, 2 \quad (50)$$

And the pooled sample scatter matrix

$$S_w = \sum_{i=1}^2 \sum_{x \in D_i} (x - m_i)(x - m_i)^t \quad (51)$$

Multiply the matrices of Eq.49 and obtain

$$\begin{bmatrix} n & (n_1 m_1 + n_2 m_2)^t \\ (n_1 m_1 + n_2 m_2) & S_w + n_1 m_1 m_1^t + n_2 m_2 m_2^t \end{bmatrix} \begin{bmatrix} \omega_0 \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ n(m_1 - m_2) \end{bmatrix}$$

This can be viewed as a pair of equations, the first of which can be solved for ω_0 in terms of w :

$$\omega_0 = -m^t w, \quad (52)$$

Where m is the mean of all of the samples.

Substituting this in the second equation, we obtain

$$\left[\frac{1}{n} S_w + \frac{n_1 n_2}{n^2} (m_1 - m_2)(m_1 - m_2)^t \right] w = m_1 - m_2. \quad (53)$$

Because the vector $(m_1 - m_2)(m_1 - m_2)^t w$ is in the direction of $m_1 - m_2$ for any value of w , we can write

$$\frac{n_1 n_2}{n^2} (m_1 - m_2)(m_1 - m_2)^t w = (1 - a)(m_1 - m_2),$$

where a is some scalar.

Then Eq.53 yields

$$w = S_w^{-1}(m_1 - m_2), \quad (54)$$

Which, except for an unimportant scale factor, is identical to the solution for Fisher's linear discriminant.

In addition, we obtain the threshold weight w_0 and the following decision rule:

Decide ω_1 if $w^t(x - m) > 0$; otherwise decide ω_2 .

References of LDA

- David Zhang, FengXi Song, Yong Xu, ZhiZhen Liang
- "Advanced Pattern Recognition Technologies with Applications to Biometrics", Medical Information Science Reference, 2009
- Yong Xu, David Zhang, Represent and fuse bimodal biometric images at the feature level: complex-matrix-based fusion scheme, Opt. Eng. 49(3), 037002, 2010
- Yong Xu, Jing-Yu Yang, Zhong Jin, A novel method for Fisher discriminant Analysis. Pattern Recognition, 37 (2), 381-384, 2004