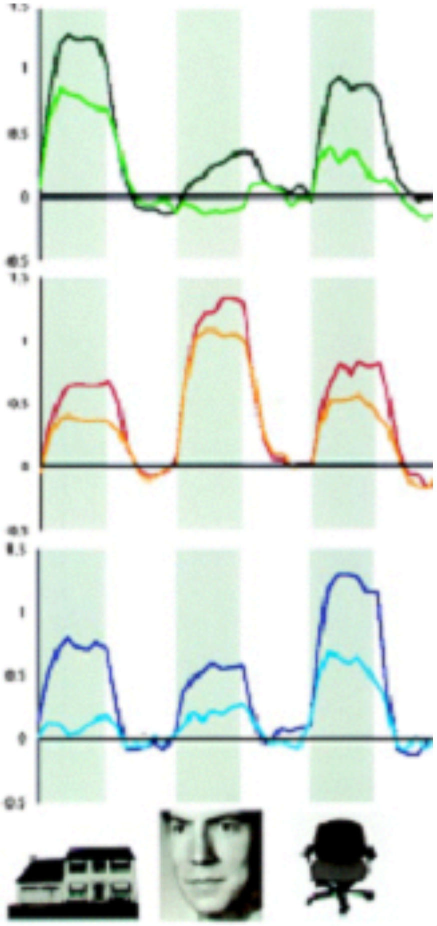
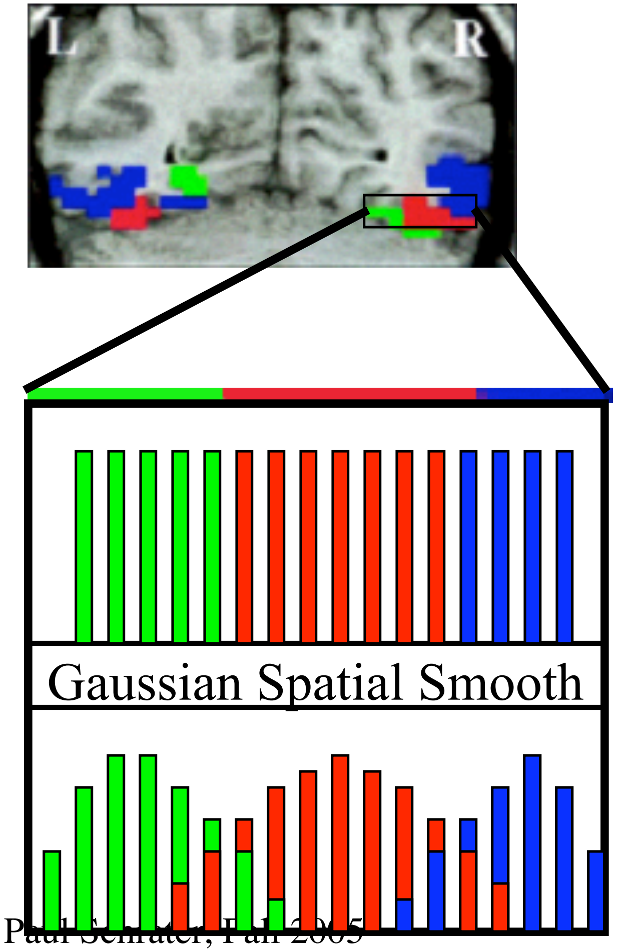
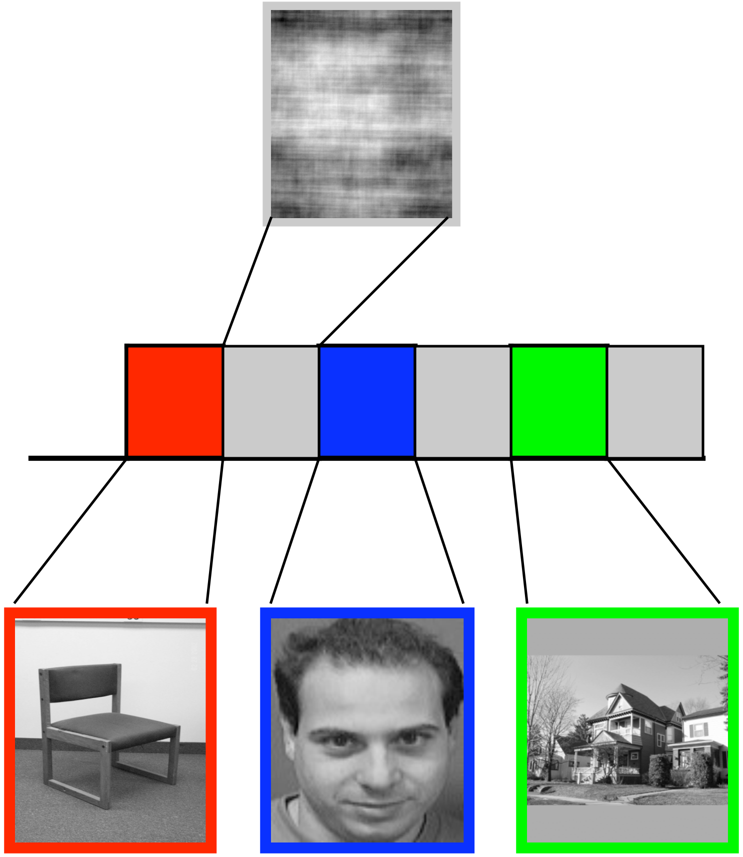
Application: Can we tell what people are looking at from their brain activity (in 'real' time)?





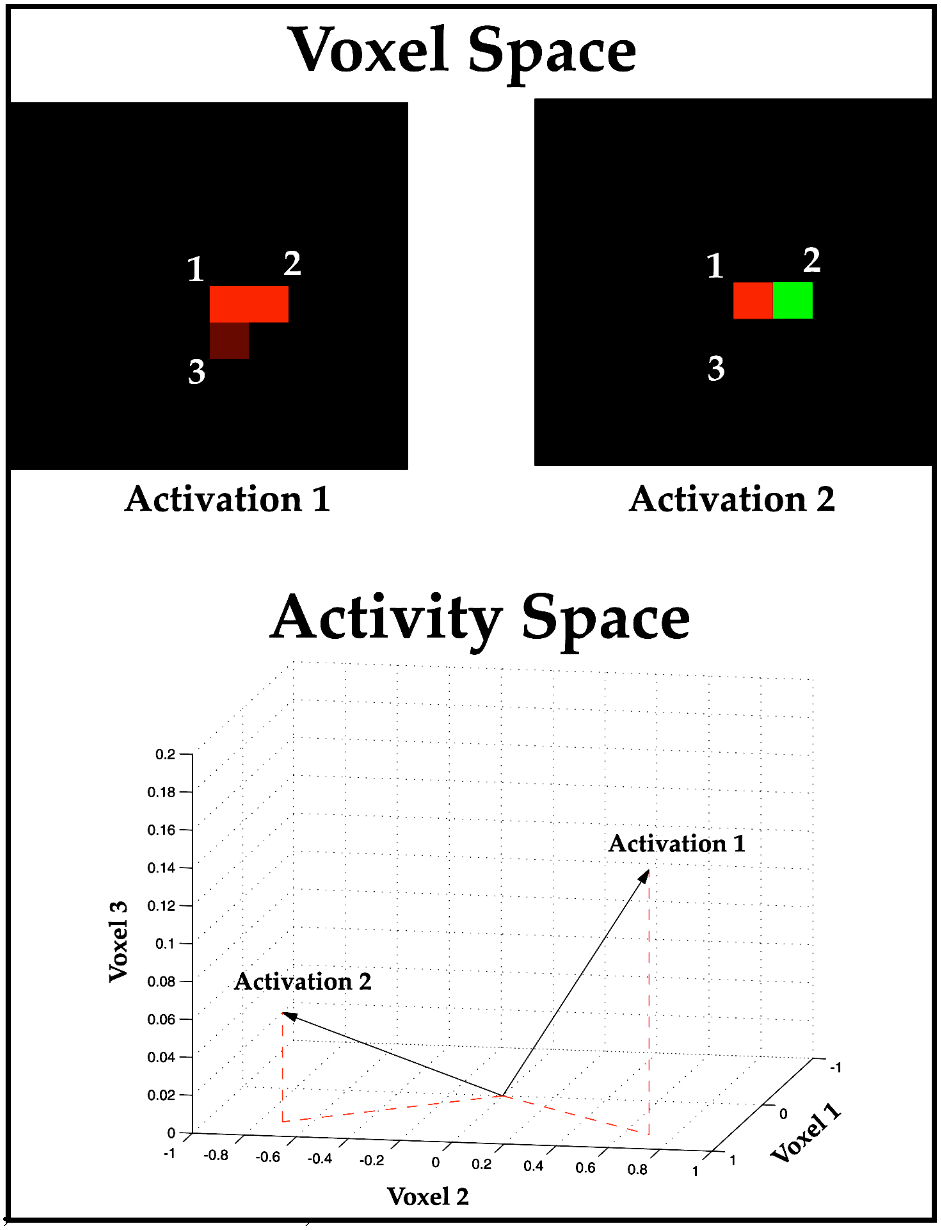
The Data

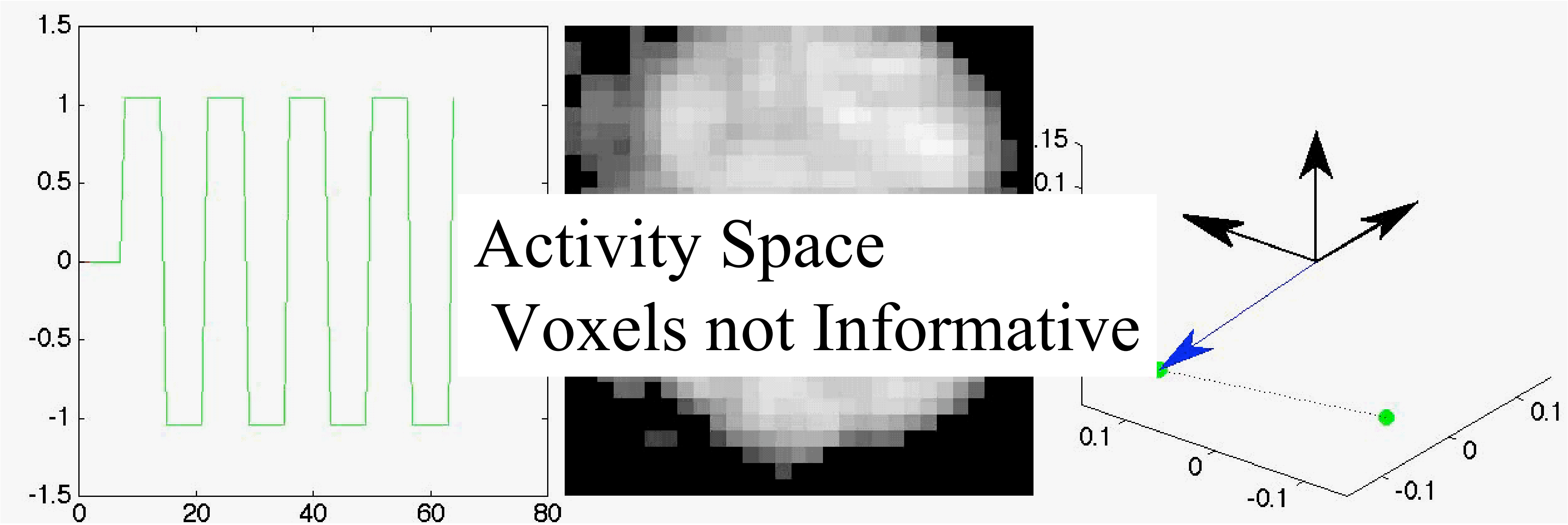
- Block Paradigm (six runs per subject)
- Three Categories of Objects (counterbalanced across runs)
  - Chairs, Faces and Houses
  - Phase scrambled control stimulus
- Two Tasks
  - Delayed matching
  - Passive Viewing



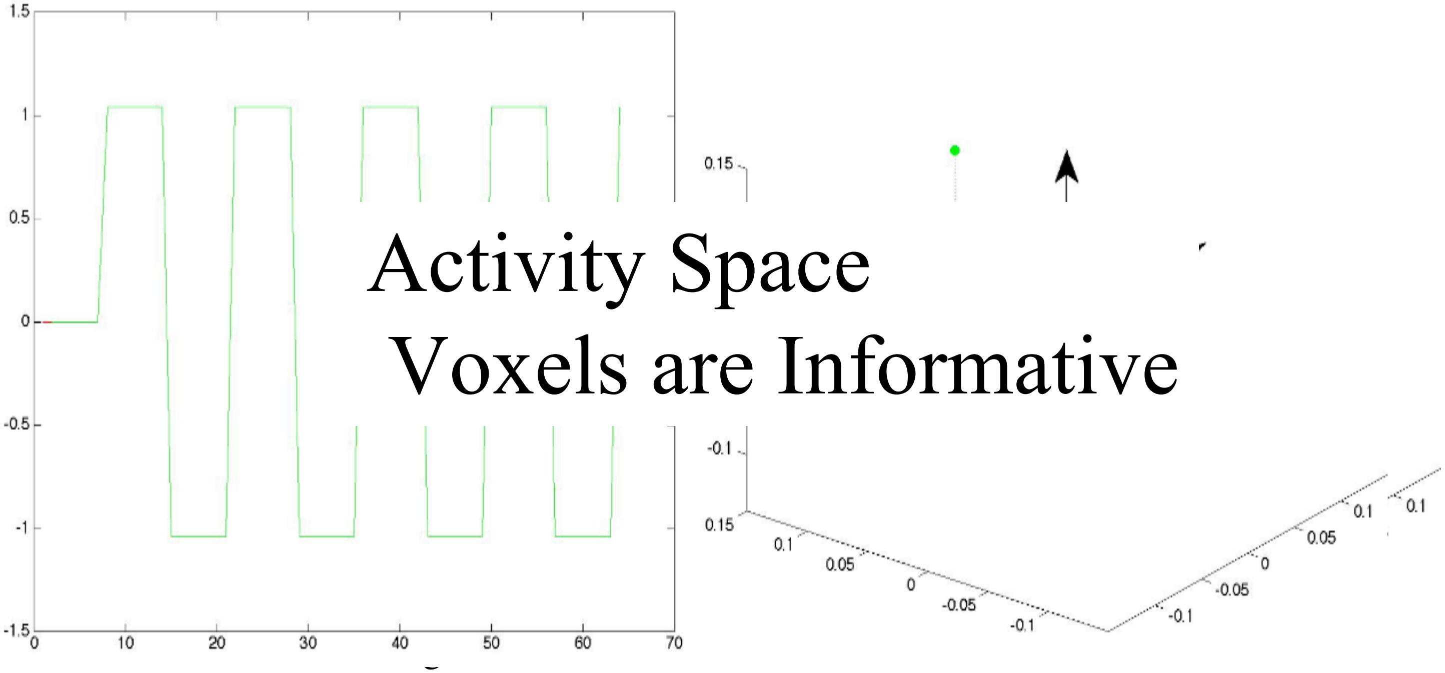
Patterns of Activation

- Activity Space
  - voxels are considered as axes in a high dimensional space
  - Every brain response can be represented as a point in the multidimensional activity space





Activity Space
Voxels not Informative



Activity Space
Voxels are Informative

Discriminant Functions for the Normal Density

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

$$g_i(x) = \ln P(x \mid \omega_i) + \ln P(\omega_i)$$

- Case of multivariate normal

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

Special case $\Sigma_i = \Sigma$

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

$$g_i(x) - g_j(x) > 0$$

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

$$\left(-\frac{1}{2}(x-\mu_j)^t \Sigma^{-1}(x-\mu_j) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\Sigma\ | + \ln P(\omega_j)\right)$$

$$\text{Now} \quad (x-\mu_i)^t \Sigma^{-1}(x-\mu_i) = x^t\Sigma^{-1}x - 2\mu_i^t\Sigma^{-1}x + \mu_i^t\Sigma^{-1}\mu_i$$

$$= \mu_i^t\Sigma^{-1}x - \mu_j^t\Sigma^{-1}x - \frac{1}{2}\mu_i^t\Sigma^{-1}\mu_i + \frac{1}{2}\mu_j^t\Sigma^{-1}\mu_j + \ln\frac{P(\omega_i)}{P(\omega_j)}$$

$$= w_i^t x - w_j^t x + w_{i0} - w_{j0}$$

- 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

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

$$\begin{aligned}
&g_i(x) - g_j(x) \\
&= w_i^t x + w_{i0} - w_j^t x + w_{j0} \\
&= \left(w_i^t - w_j^t\right)x + \left(w_{i0} - w_{j0}\right) \\
&= \left(w_i^t - w_j^t\right)(x - x_0) \\
&\text{For Identity covariance case:} \\
&= \frac{1}{\sigma^2}\left(\mu_i^t - \mu_j^t\right)(x - x_0)
\end{aligned}$$

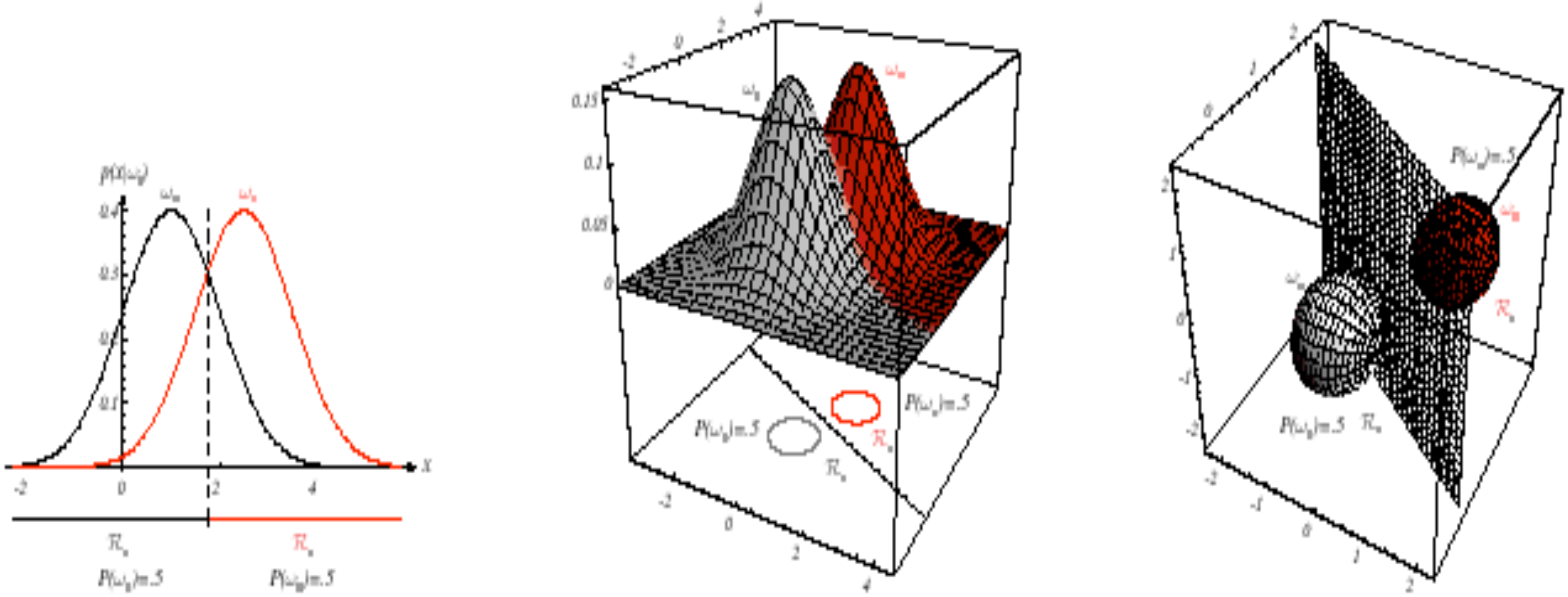


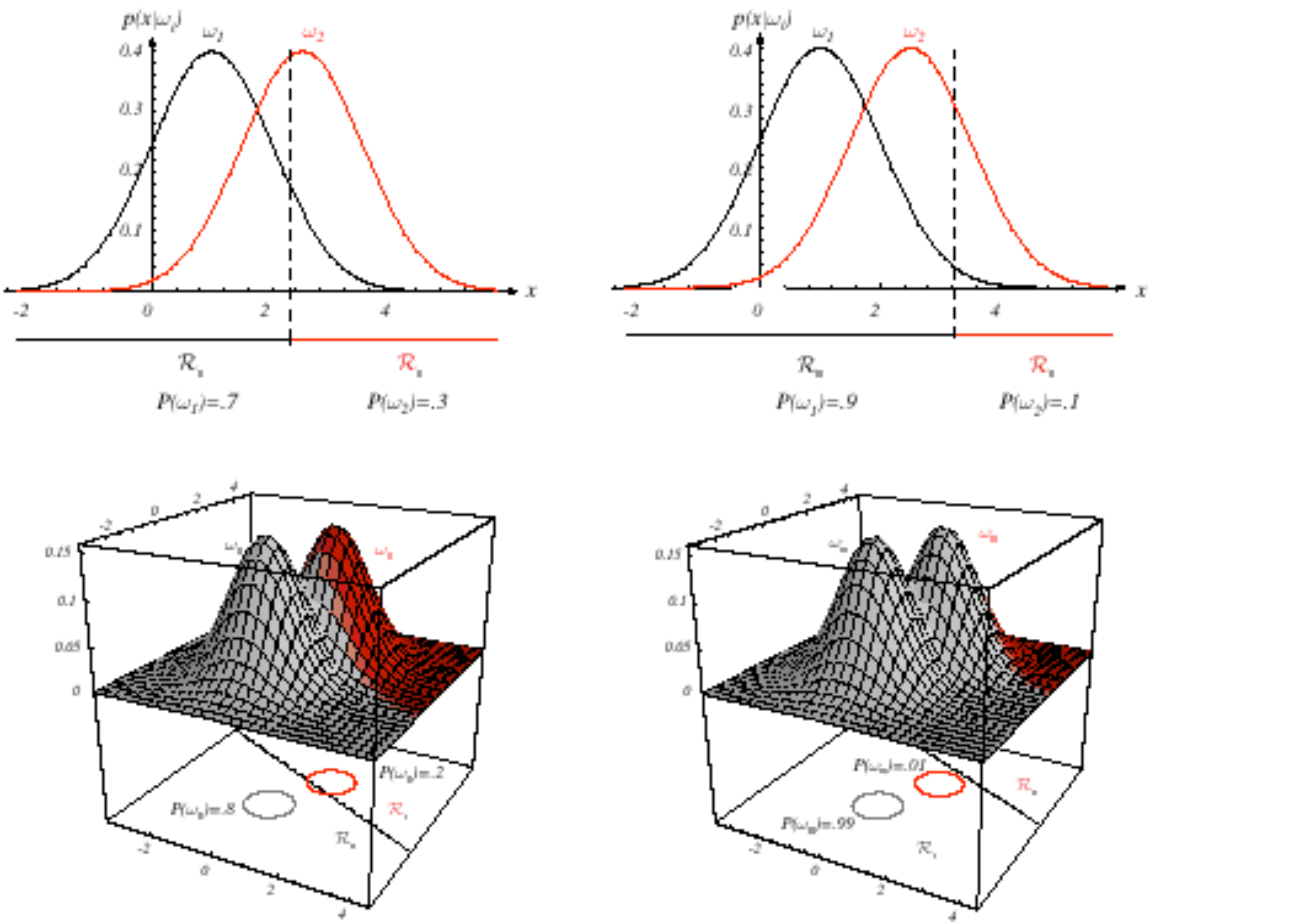
FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, 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 $\mathcal{R}_j$

$$x_0 = \frac{1}{2}(\mu_i + \mu_j) - \frac{\sigma^2}{\left\|\mu_i - \mu_j\right\|^2} \ln \frac{P(\omega_i)}{P(\omega_j)} (\mu_i - \mu_j)$$

always orthogonal to the line linking the means!

$$if \;\; P(\omega_i) = P(\omega_j) \;\; then \;\; x_0 = \frac{1}{2}(\mu_i + \mu_j)$$



$p(x|\omega_i)$
ω_1
ω_2
0.4
0.3
0.2
0.1
x
-2
0
2
4
$\mathcal{R}_1$
$\mathcal{R}_2$
$P(\omega_1)=.7$
$P(\omega_2)=.3$
$p(x|\omega_i)$
ω_1
ω_2
0.4
0.3
0.2
0.1
x
-2
0
2
4
$\mathcal{R}_1$
$\mathcal{R}_2$
$P(\omega_1)=.9$
$P(\omega_2)=.1$
-2
0
2
4
0.15
0.1
0.05
0
ω_1
ω_2
$P(\omega_2)=.2$
$P(\omega_1)=.8$
$\mathcal{R}_2$
$\mathcal{R}_1$
-2
0
2
4
-2
0
2
4
0.15
0.1
0.05
0
ω_1
ω_2
$P(\omega_2)=.01$
$P(\omega_1)=.99$
$\mathcal{R}_2$
$\mathcal{R}_1$
-2
0
2
4

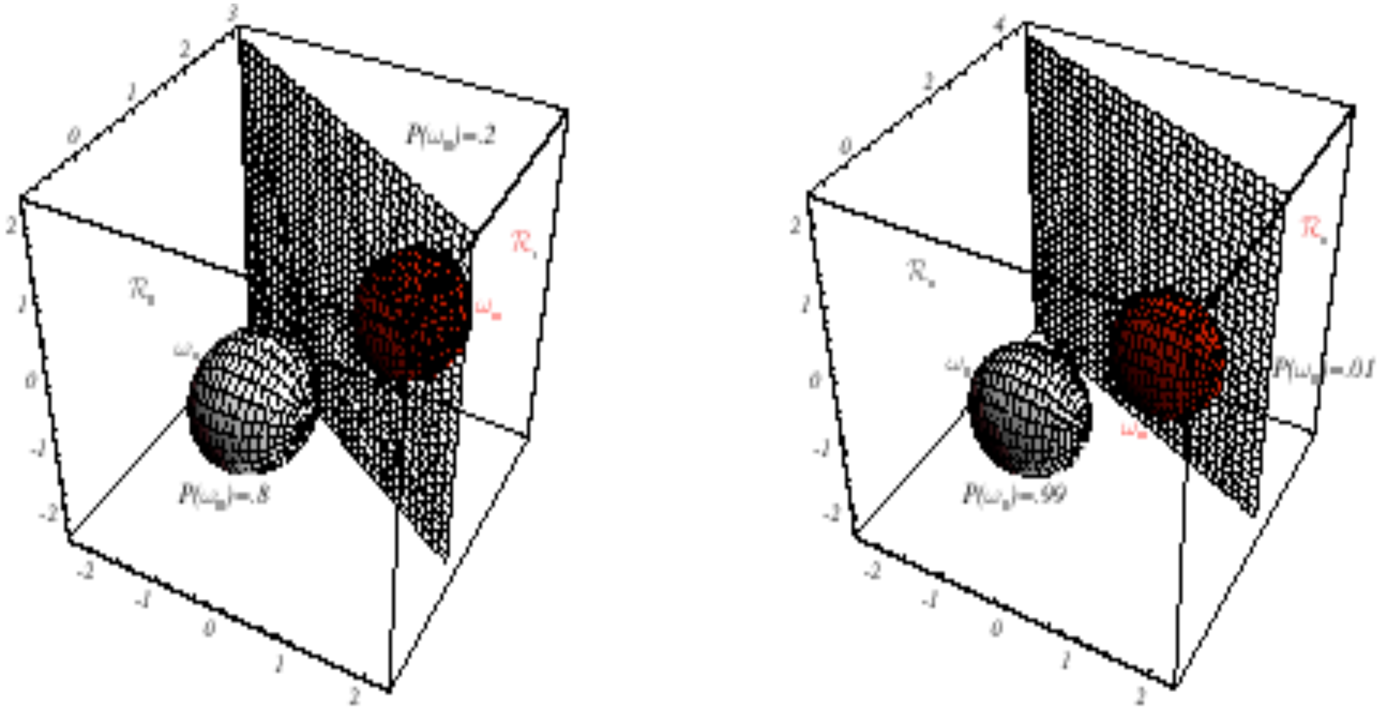


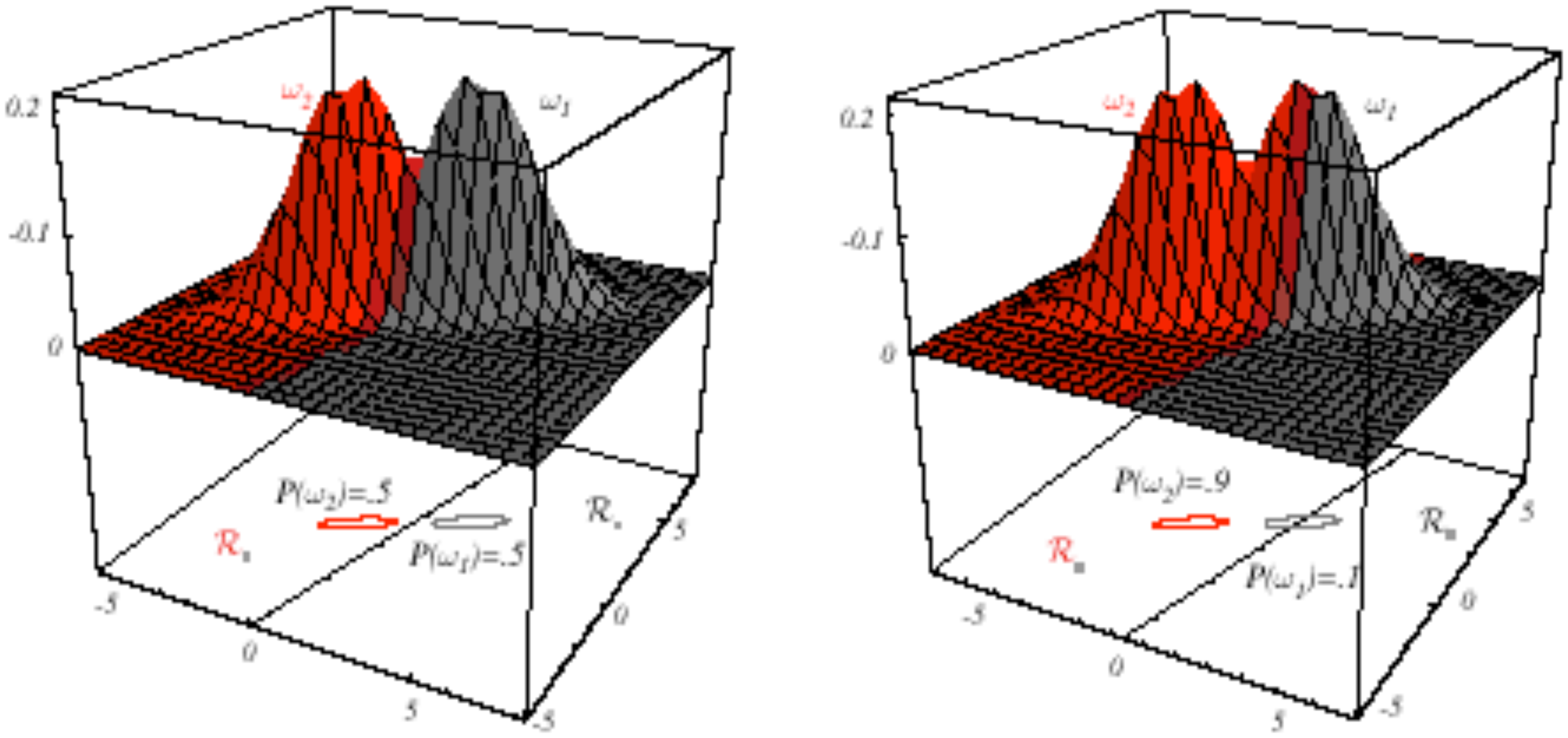
FIGURE 2.11. As the priors are changed, the decision boundary shifts; for sufficiently disparate priors the boundary will not lie between the means of these one-, two- and three-dimensional spherical Gaussian distributions. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*.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

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

  - Hyperplane separating $\mathcal{R}_i$ and $\mathcal{R}_j$

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

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


0.2
-0.1
0
ω_2
ω_1
$P(\omega_2)=.5$
$P(\omega_1)=.5$
$\mathcal{R}_2$
$\mathcal{R}_1$
-5
0
5
5
0
-5
0.2
-0.1
0
ω_2
ω_1
$P(\omega_2)=.9$
$P(\omega_1)=.1$
$\mathcal{R}_2$
$\mathcal{R}_1$
-5
0
5
5
0
-5

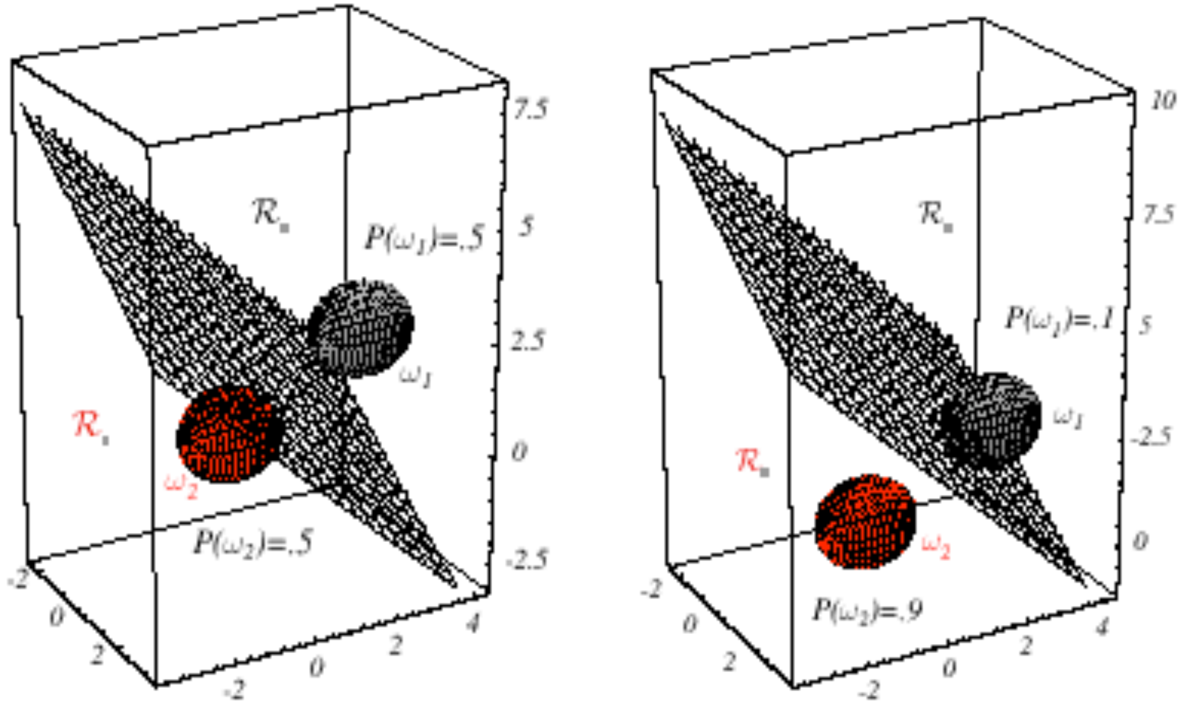


FIGURE 2.12. Probability densities (indicated by the surfaces in two dimensions and ellipsoidal surfaces in three dimensions) and decision regions for equal but asymmetric Gaussian distributions. The decision hyperplanes need not be perpendicular to the line connecting the means. From: Richard O. Duda,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

- Case Σ_i = arbitrary
  - The covariance matrices are different for each category

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

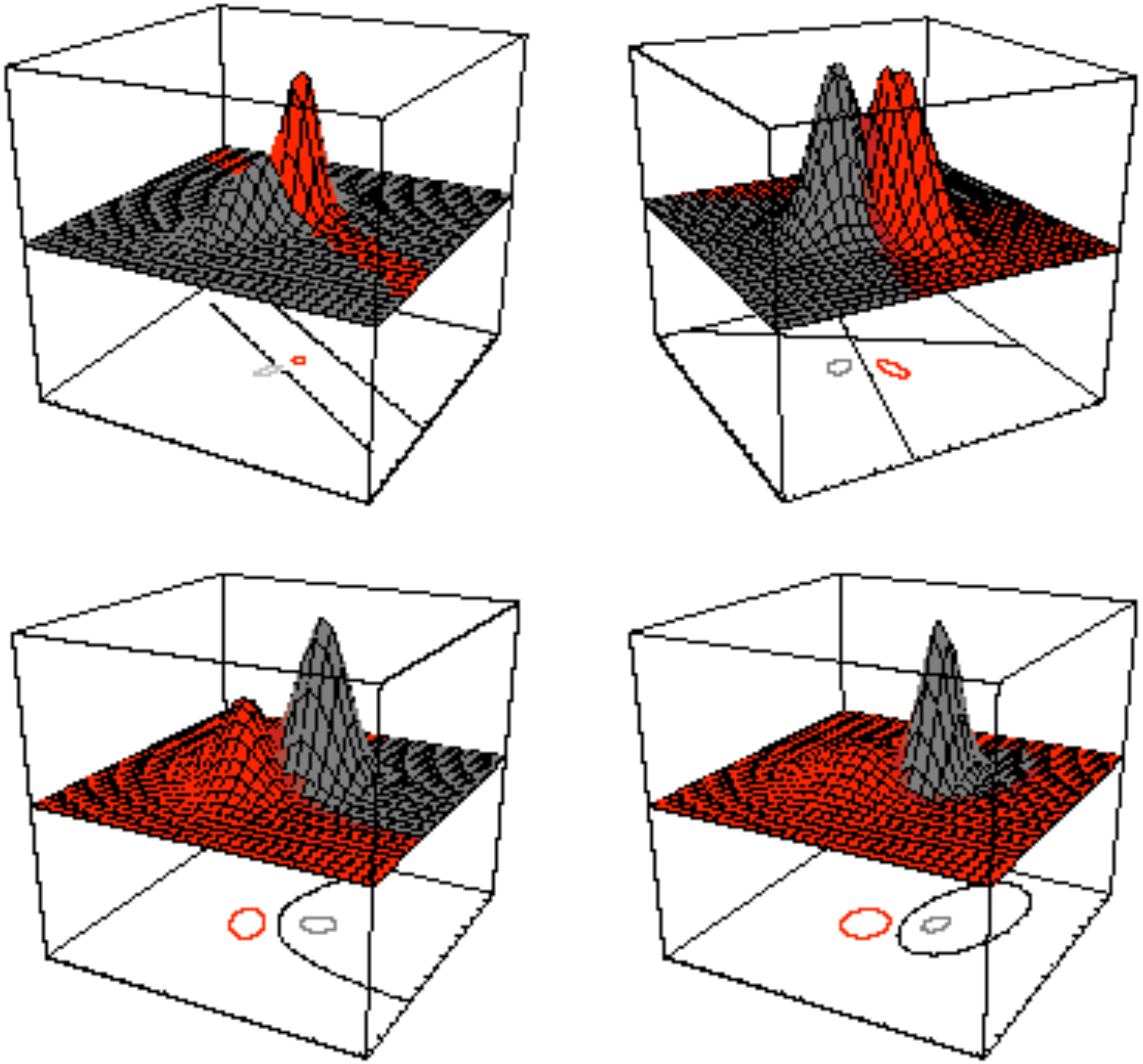
where :

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

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

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

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



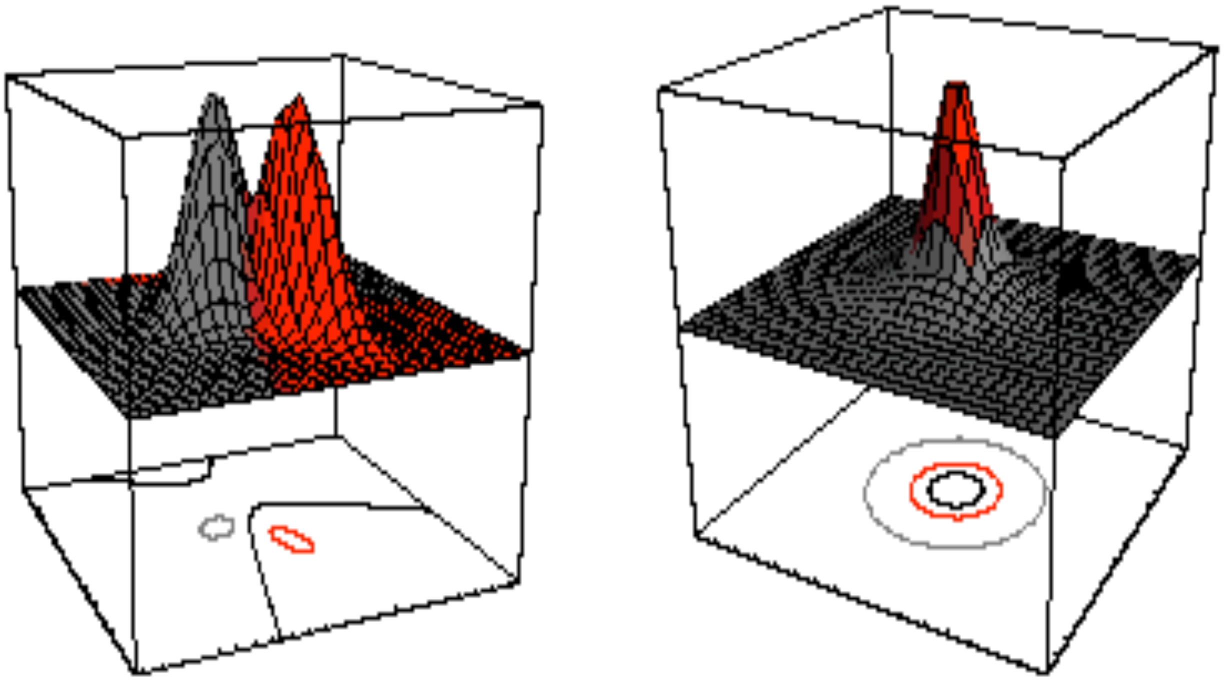


FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaussian distributions whose Bayes decision boundary is that hyperquadric. These variances are indicated by the contours of constant probability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

Estimating model parameters

Given Class-conditional, parametric density $P(x\,|\,\omega_i,\theta)$

$$Example: \quad P(x\,|\,\omega_i,\theta_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_i)^t\Sigma_i^{-1}(x-\mu_i)\right)$$

$$where \qquad \theta_i = \{\mu_i,\Sigma_i\}$$

- Plug in estimates: Use procedure to get Mean and Covariance. Plug these values in.
  - Maximum likelihood
  - Maximum A posteriori
  - Overfits training data
- Bayesian estimates:
  - Take into account the reliability of your mean and covariance estimates to get better generalization.

- Bayesian Estimation (Bayesian learning in pattern classification problems)
  - MLE: θ presumed fixed
  - BE: θ random variable (ignorant of value)
  - The computation of posterior probabilities $P(\omega_i \mid x)$ lies at the heart of Bayesian classification
  - Goal: compute $P(\omega_i \mid x, D)$

    Given the sample D, Bayes formula can be written

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

- Maximum-Likelihood Estimation

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

– General principle

  - Assume we have c classes and

    $P(x \mid \omega_j) \sim N(\mu_j, \Sigma_j)$

    $P(x \mid \omega_j) \equiv P(x \mid \omega_j, \theta_j)$ where:

$$\theta = (\mu_j, \Sigma_j) = (\mu_j^1, \mu_j^2, \ldots, \sigma_j^{11}, \sigma_j^{22}, \mathrm{cov}(x_j^m, x_j^n) \ldots)$$

- ML Problem Statement

  – Let D = {x_1, x_2, …, x_n}

  For independent feature values,

  $P(x_1,\ldots, x_n \mid \theta) = \prod_{k=1:n} P(x_k \mid \theta)$;

  Our goal is to determine $\hat{\theta}$

$$\hat{\theta} = \arg\max_{\theta} P(x_1,\cdots,x_n \mid \theta)$$

  (value of θ that makes this sample the most representative!)

- Use the training samples to estimate:

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

each θ_i $(i = 1, 2, \ldots, c)$ is associated with each category

  Suppose that D contains n samples, $x_1, x_2, \ldots, x_n$

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

P(D | θ) is called the likelihood of θ w.r.t. the set of samples)

$\hat{\theta}$

- ML estimate of θ is, by definition the value that maximizes P(D | θ)

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

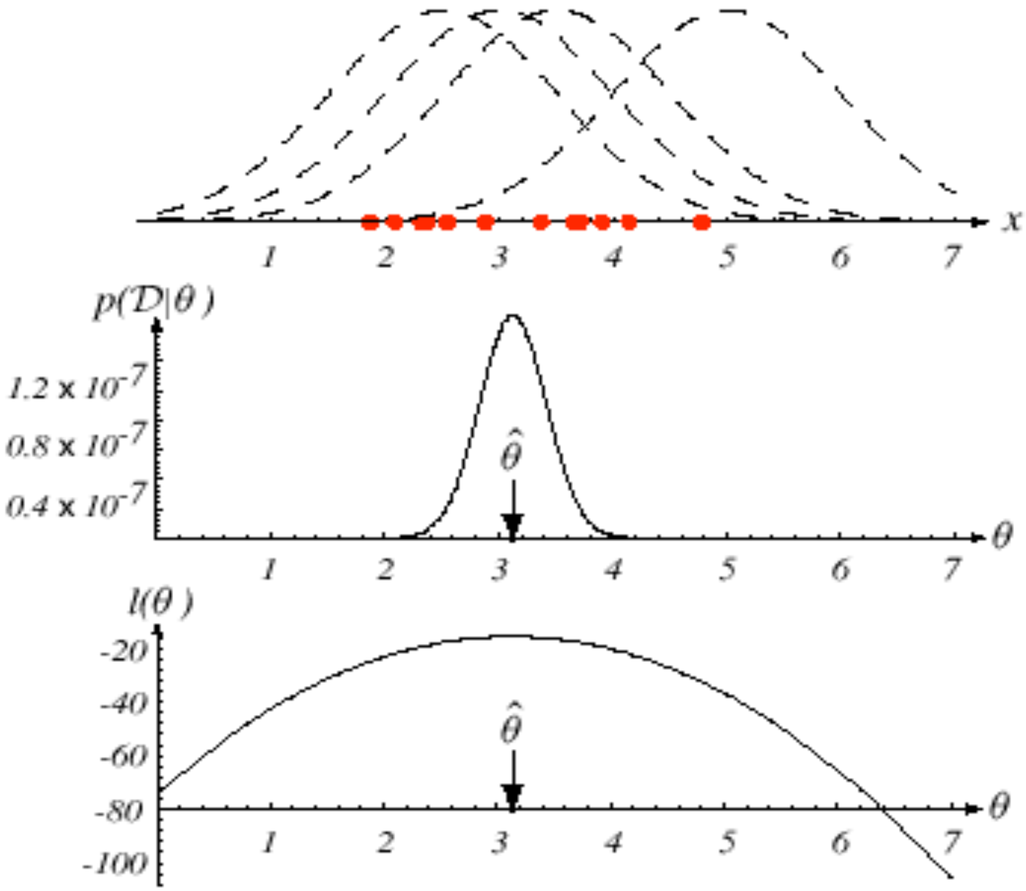


FIGURE 3.1. The top graph shows several training points in one dimension, known or assumed to be drawn from a Gaussian of a particular variance, but unknown mean. Four of the infinite number of candidate source distributions are shown in dashed lines. The middle figure shows the likelihood $p(\mathcal{D}|\theta)$ 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 $l(\theta)$, 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 $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^t$ and let ∇_θ be the gradient operator

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

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

$$l(\theta) = \ln P(D \mid \theta)$$

  - Find θ that maximizes the log-likelihood

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

Set of necessary conditions for an optimum is:

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

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

- Example of a specific case: unknown μ
  - P(x_i | μ) ~ N(μ, Σ) (Samples from a multivariate normal dist.)

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

$$and \quad \nabla_\mu \ln P(x_k \mid \mu) = \Sigma^{-1}(x_k - \mu)$$

  - θ = μ therefore:

    The ML estimate for μ must satisfy:

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

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

$$= \sum_{k=1}^{n} x_k - n\hat{\mu} = 0 \quad \Rightarrow \quad \hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

- Multiplying by Σ and rearranging, we obtain:

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

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

Conclusion:

If $P(x_k \mid \omega_j)$ ($j = 1, 2, \ldots, c$) is a Gaussian in a *d*-dimensional feature space

Then we can estimate the vector

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

- ML Estimation:
  - Gaussian Case: *unknown μ and σ*

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

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

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

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

Summation:

$$\begin{cases} \displaystyle\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2}(x_k - \theta_1) = 0 & (1) \\ \displaystyle -\sum_{k=1}^{n} \frac{1}{\hat{\theta}_2} + \sum_{k=1}^{k=n} \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0 & (2) \end{cases}$$

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

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

- Bias
  - ML estimate for σ^2 is biased

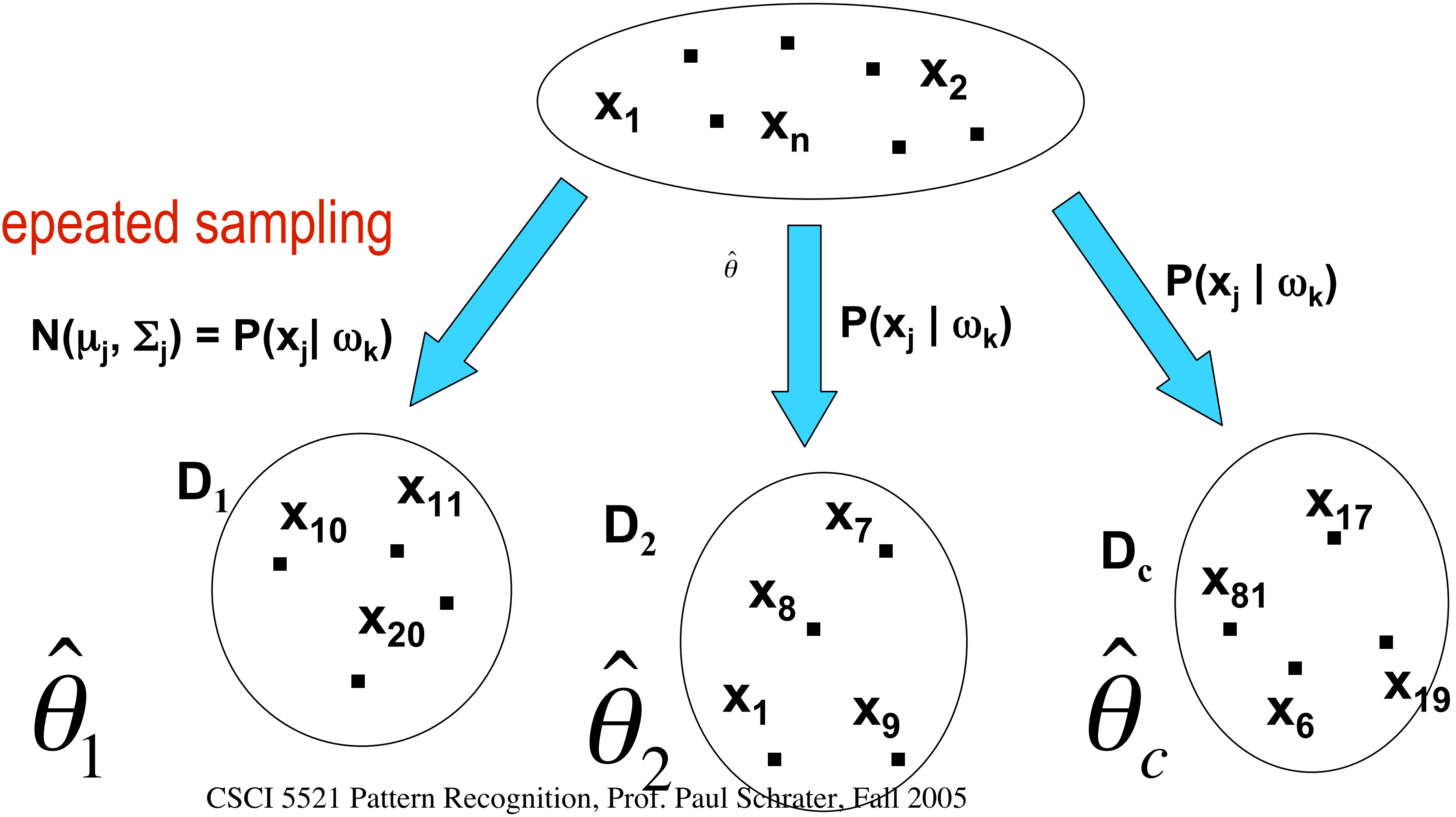
$$E\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right] = \frac{n-1}{n}.\sigma^2 \neq \sigma^2$$

  - An elementary unbiased estimator for Σ is:

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

Problems with using the ML estimate as a Plug-in estimate

Population Distribution



Repeated sampling

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

$P(x_j \mid \omega_k)$

$P(x_j \mid \omega_k)$

- Bayesian Estimation (Bayesian learning to pattern classification problems)
  - In MLE θ is presumed fixed
  - In BE θ is a random variable
  - The computation of posterior probabilities $P(\omega_i \mid x)$ lies at the heart of Bayesian classification
  - Goal: compute $P(\omega_i \mid x, D)$

  Given the sample D, Bayes formula can be written

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

Problem: Find

$$P(x \mid \omega_i) \quad \textit{Given} \quad P(x \mid \vec{\theta}_i), \text{ (but } \vec{\theta}_i \text{ unknown)}$$

$\text{Solution: Learn } P(\vec{\theta}_i \mid \mathsf{D}) \text{ from data}$

then

$$P(x \mid \omega_i) = \int \ldots \int P(x \mid \vec{\theta}_i) P(\vec{\theta}_i \mid \mathsf{D}) d\vec{\theta}_i$$

and

$$P(\omega_i) = P(\omega_i \mid \mathsf{D}) \text{ (Training sample provides this!)}$$

Thus:

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

- Bayesian Parameter Estimation: Gaussian Case

Goal: Estimate θ using the a-posteriori density $P(\theta \mid D)$

– The univariate case: $P(\mu \mid D)$

μ is the only unknown parameter

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

(μ_0 and σ_0 are known!)

$$P(\mu \mid \mathrm{D}) = \frac{P(\mathrm{D} \mid \mu).P(\mu)}{\int P(\mathrm{D} \mid \mu).P(\mu)d\mu} \qquad (1)$$

$$= \alpha \prod_{k=1}^{n} P(x_k \mid \mu).P(\mu)$$

– Gaussian is a Reproducing density

$$P(\mu \mid \mathrm{D}) \sim N(\mu_n, \sigma_n^2) \qquad (2)$$

Identifying (1) and (2) yields:

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

$$and\ \ \sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{n\sigma_0^2 + \sigma^2}$$

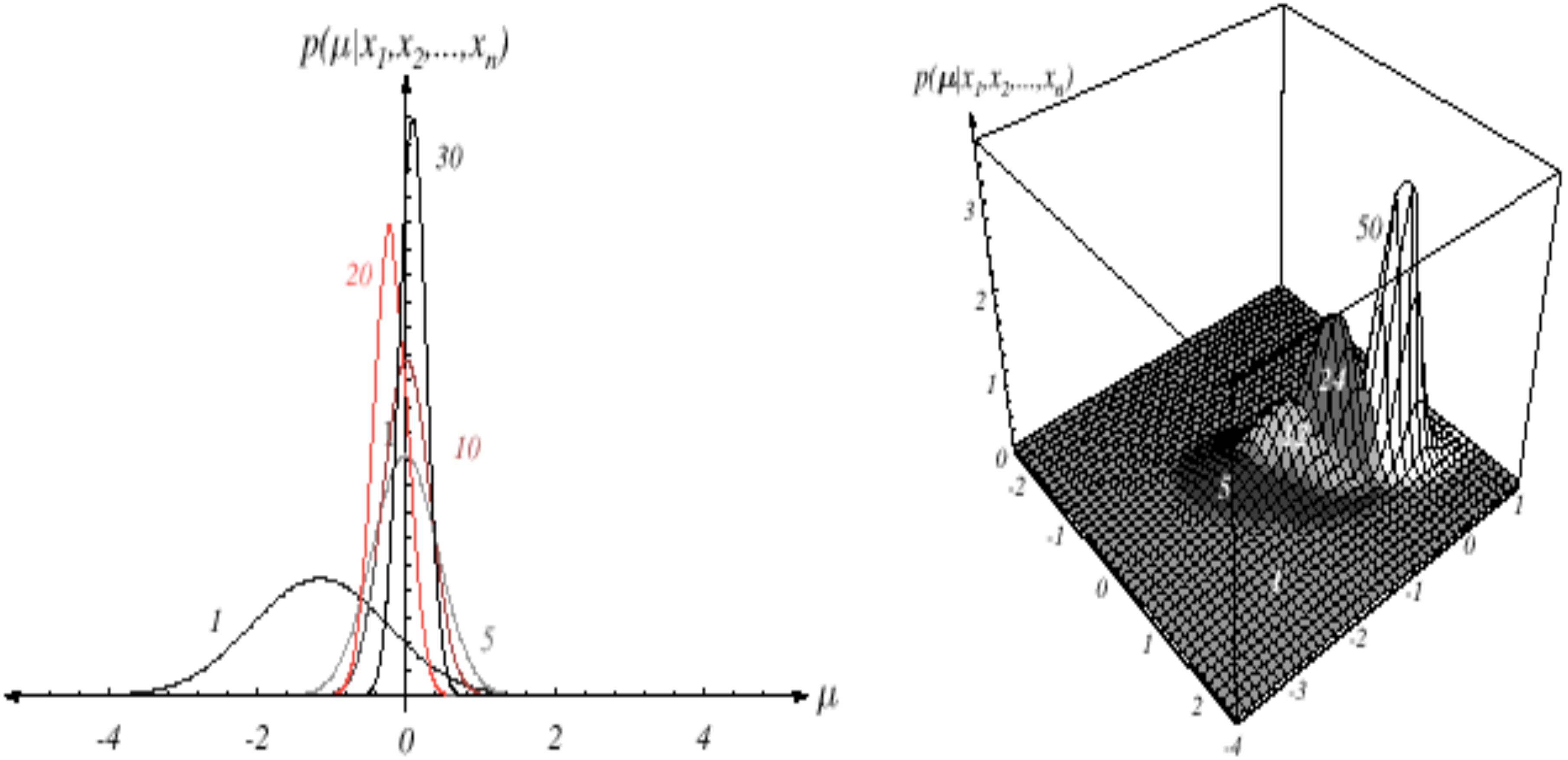


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001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

– The univariate case P(x | D)

- P(μ | D) computed
- P(x | D) remains to be computed!

$$\mathbf{P(x \mid D) = \int P(x \mid \mu).P(\mu \mid D)d\mu} \quad \textbf{is Gaussian}$$

It provides:

$$\mathbf{P(x \mid D) \sim N(\mu_n, \sigma^2 + \sigma_n^2)}$$

(Desired class-conditional density P(x | D_j, ω_j))

Therefore: P(x | D_j, ω_j) together with P(ω_j)

And using Bayes formula, we obtain the

Bayesian classification rule:

$$\max_{\omega_j} P\left(\omega_j \mid x, D\right) \equiv \max_{\omega_j}\left[P(x \mid \omega_j).P(\omega_j \mid D_j)\right]$$

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

The basic problem is:

"Compute the posterior density P(θ | D)"

then "Derive P(x | D)"

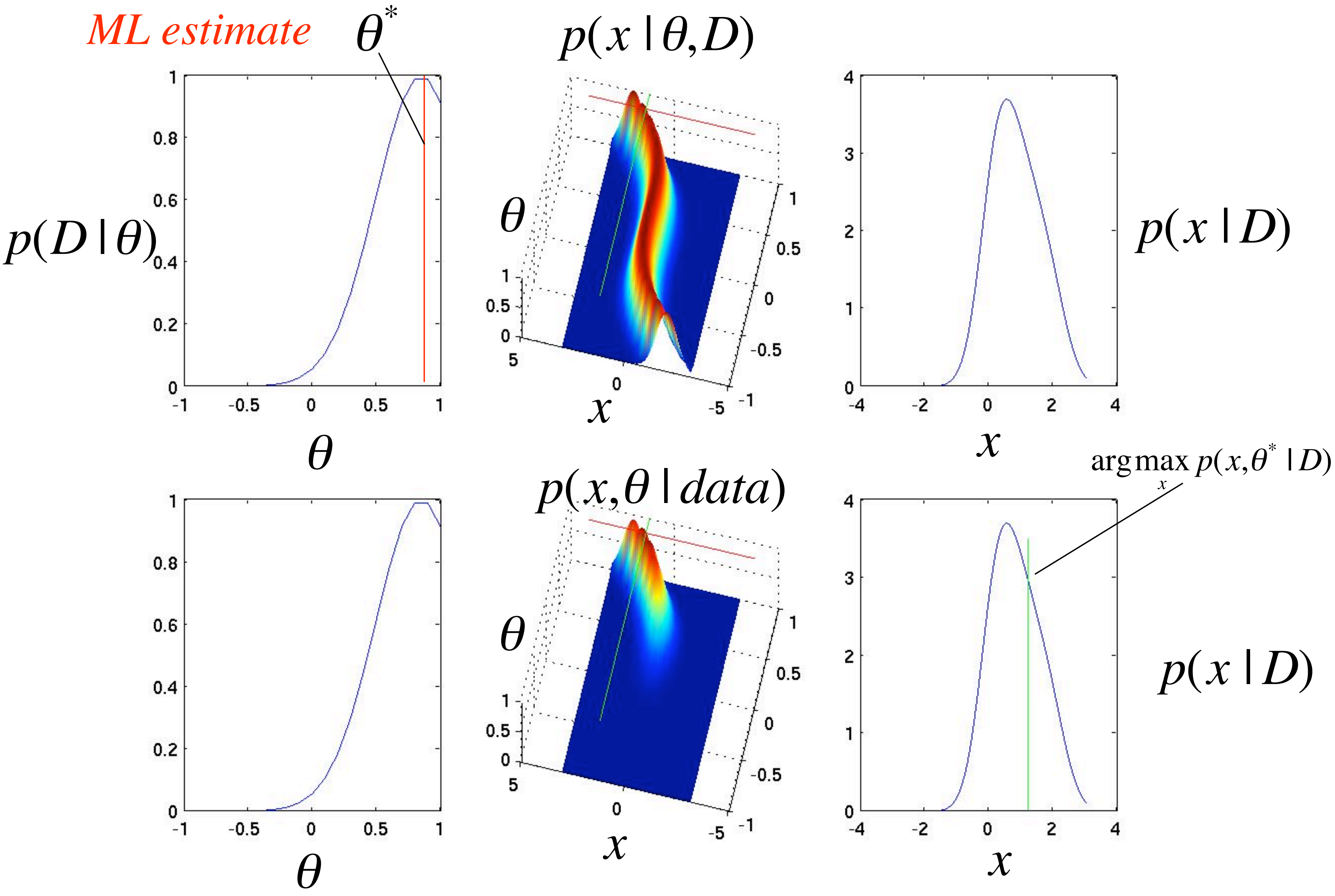
Using Bayes formula, we have:

$$P(\theta \mid D) = \frac{P(D \mid \theta).P(\theta)}{\int P(D \mid \theta).P(\theta)d\theta},$$

And by independence assumption:

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

Plug-in vs. BE example



ML estimate θ^*

$p(x \mid \theta, D)$

$p(x, \theta \mid data)$

$\arg\max_x p(x, \theta^* \mid D)$

- Problems of Dimensionality
  - Problems involving 50 or 100 features (binary valued)
    - Classification accuracy depends upon the dimensionality and the amount of training data
    - Case of two classes multivariate normal with the same covariance

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

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

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

- If features are independent then:

$$\Sigma = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_d^2)$$

Simple Feature Selection possible for ind. features

- Most useful features are the ones for which the difference between the means is large relative to the standard deviation

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

- It has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better performance: ***i.e. we have the wrong model***

Case where original features axes cannot be selected (pruned) without performance loss

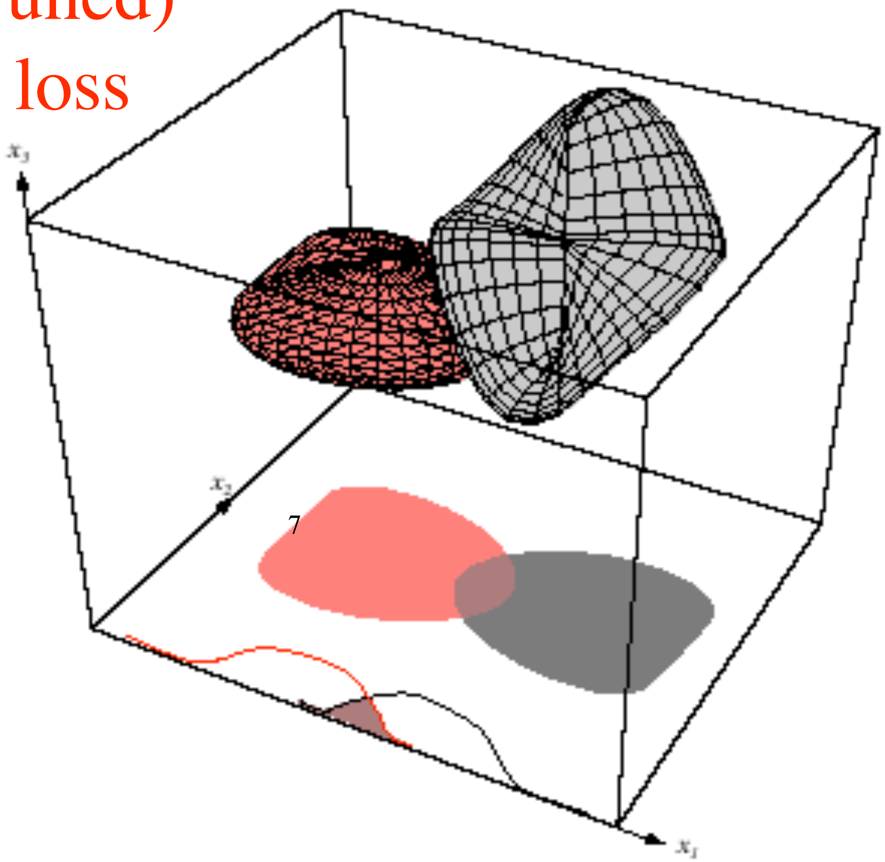
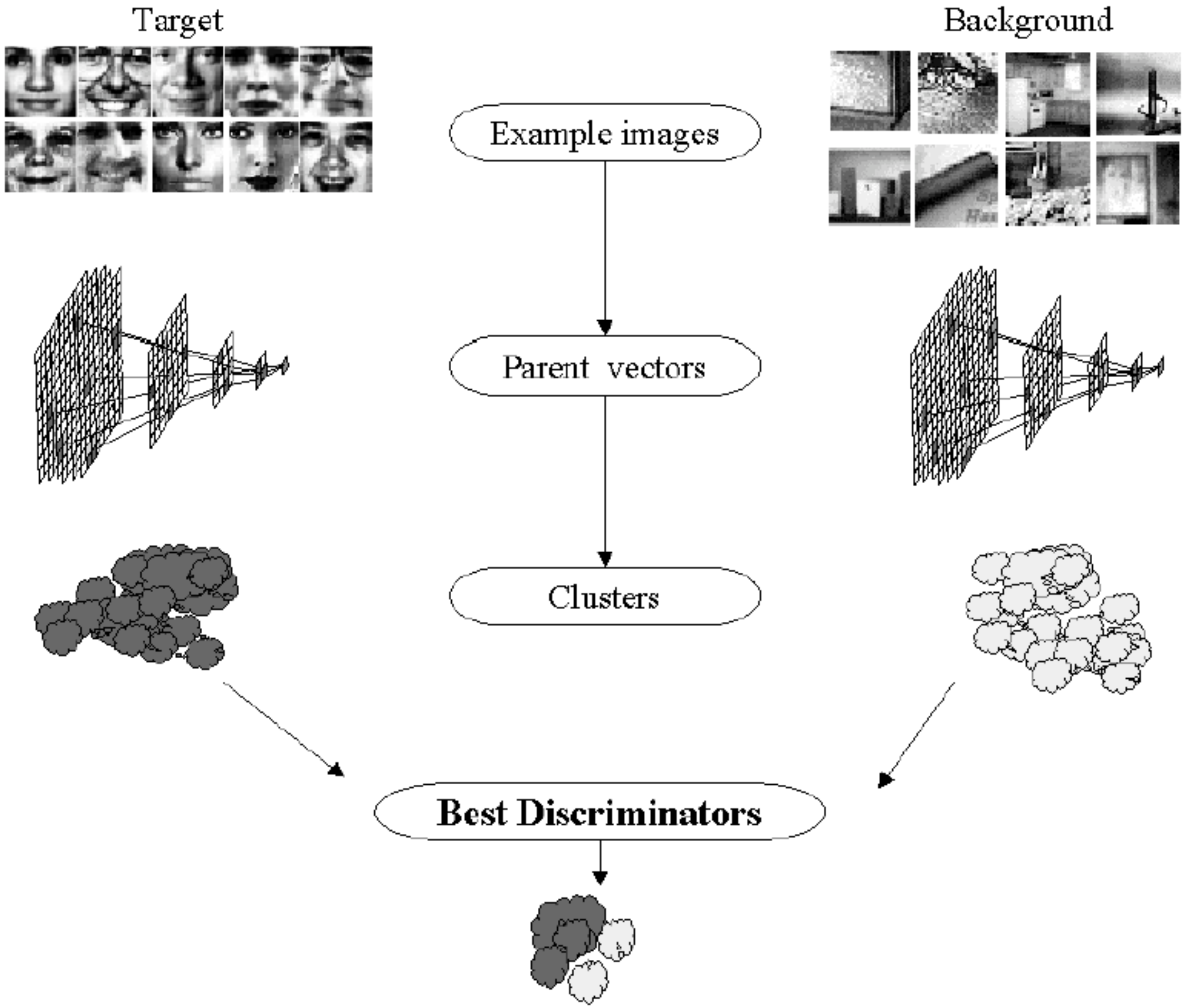


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions, 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.



Target

Background

Example images

Parent vectors

Clusters

Best Discriminators