CpE 646 Pattern Recognition and Classification

Prof. Hong Man

Department of Electrical and Computer Engineering
Stevens Institute of Technology

STEVENS Institute of Technology

Visual Information Environment Laboratory

Bayesian Decision Theory

Chapter 3 (Section 3.7, 3.8) Outline:

- Problems of dimensionality
- Component analysis and discriminant

Problems of Dimensionality

- In practical multi-category cases, it is common to see problems involving hundreds of features, especially when features are binary valued.
- We assume every single feature is useful for some discrimination, but we do not know (usually doubt) if they all provide some independent information.
- Two concerns:
  - Classification accuracy depends upon the dimensionality and the amount of training data
  - Computational complexity of designing a classifier.

Accuracy, Dimension and Training Sample Size

- If features are statistically independent, classification performance can be excellent in theory.
- Example: two-classes multivariate normal with the same covariance, $p(x|\omega_j) \sim N(\mu_j,\Sigma)$, $j=1, 2$.

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

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

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

r^2 is called the squared Mahalanobis distance from μ_1 to μ_2

The Mahalanobis Distance

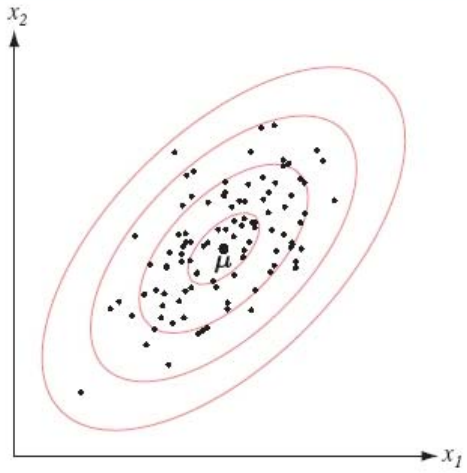


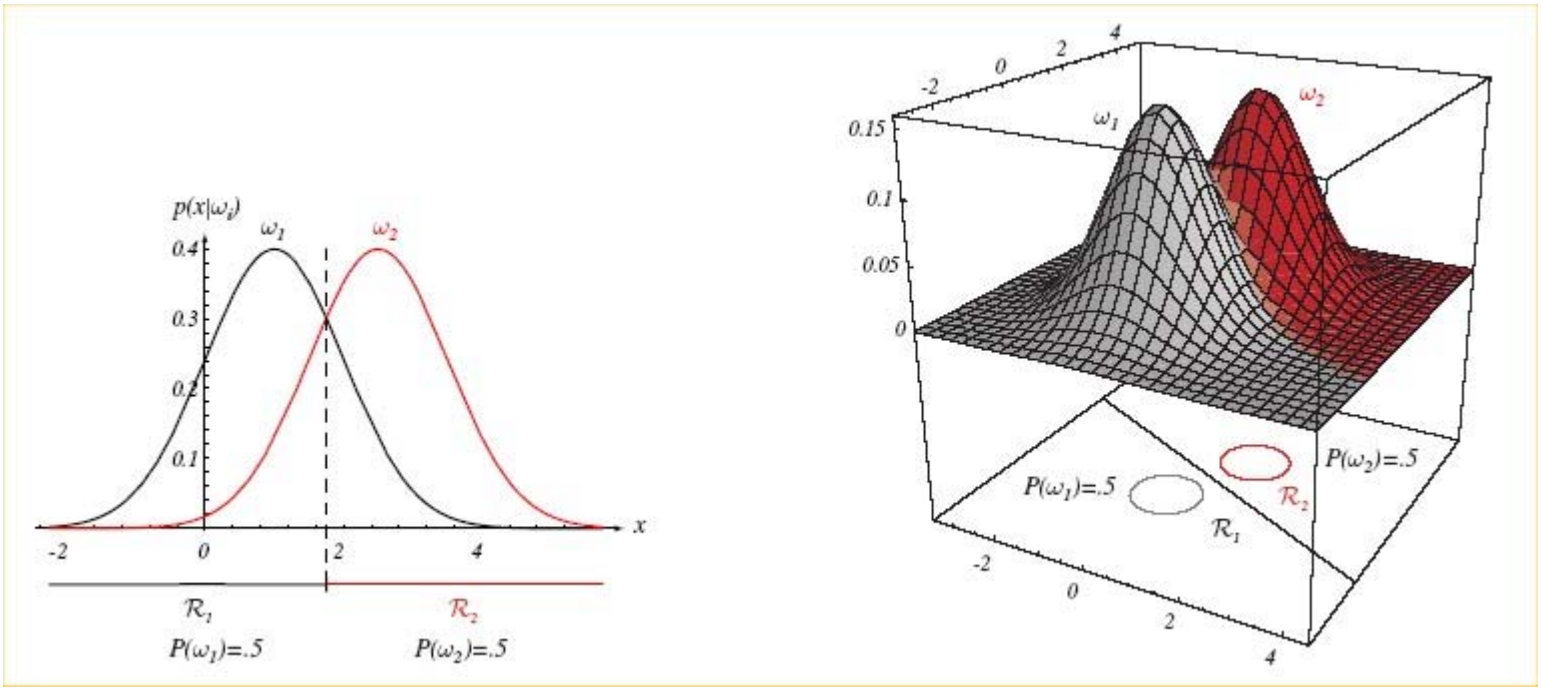
FIGURE 2.9. Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean μ. The ellipses show lines of equal probability density of the Gaussian. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

the squared Mahalanobis distance from x to μ is

$$r^2 = (x - \mu_i)^t \Sigma_i^{-1} (x - \mu_i)$$

the loci of points of constant density are hyperellipsoids with constant squred Mahalanobis distance.

Accuracy, Dimension and Training Sample Size



Accuracy, Dimension and Training Sample Size

- If features are independent then:

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

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

- This shows how different feature contributes to reducing the probability of error (~increasing r).
  - Most useful features are the ones for which the difference between the means is large relative to the standard deviation
  - Also no feature is useless
  - To further reduce the error rate, we can introduce new independent features.

Accuracy, Dimension and Training Sample Size

- In general if the probabilistic structure of the problem is known, introducing new features will improve classification performance
  - Existing feature set maybe inadequate.
  - Bayes risk can not be increased by adding new feature. At worst, the Bayes classifier may ignore the new feature.
- However adding new feature will increase the computational complexity of both the feature extractor and the classifier.

Accuracy, Dimension and Training Sample Size

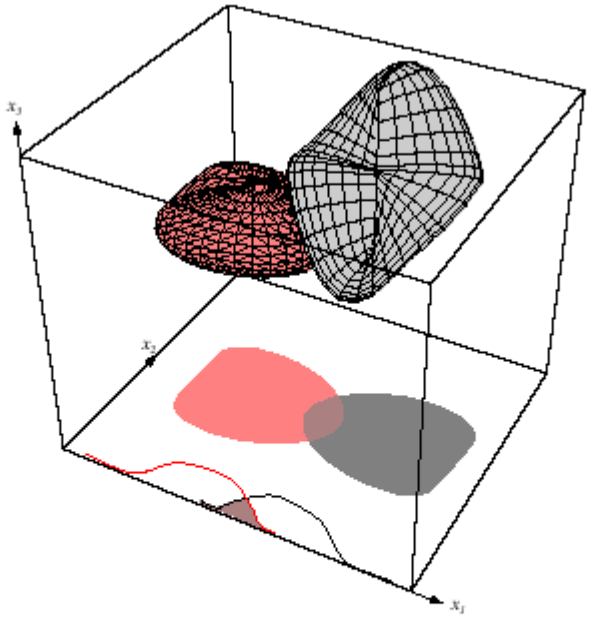


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions,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

Accuracy, Dimension and Training Sample Size

- It has frequently been observed in practice that, beyond a certain point, the inclusion of additional features leads to worse rather than better performance.
  - Frequently it is because that we have the wrong model (Gaussian) or wrong assumption (independent features),
  - Also the number of training samples may be inadequate so the distributions are not estimated accurately.

Computational Complexity

- Our design methodology is affected by the computational difficulty
- Notations of computational complexity
  - Order of a function: if $\boldsymbol{f(x)}$ is "of the order of $h(x)$", written as $\boldsymbol{f(x) = O(h(x))}$ and read as "big oh of $h(x)$", if there exists constant c and x_0 such that $\boldsymbol{|f(x)| \leq c|h(x)|}$ for all $\boldsymbol{x > x_0}$
    - An upper bound on $\boldsymbol{f(x)}$ grows no worse than $\boldsymbol{h(x)}$ for sufficiently large $\boldsymbol{x}$
    - Example: $\boldsymbol{f(x) = a_0 + a_1 x + a_2 x^2}$, we have $\boldsymbol{h(x) = x^2}$, and $\boldsymbol{f(x) = O(x^2)}$
    - Big oh order of a function is not unique. In our example: $\boldsymbol{f(x) = O(x^2)}$, or $\boldsymbol{O(x^3)}$, or $\boldsymbol{O(x^4)}$, or $\boldsymbol{O(x^2 \ln x)}$

Computational Complexity

- Big theta notation. $f(x) = \Theta(h(x))$ and read as "big theta of $h(x)$", if there exists constants x_0, c_1 and c_2 such that for all $x > x_0$, $c_1 h(x) \leq f(x) \leq c_2 h(x)$
  - In previous example, $f(x) = \Theta(x^2)$, but will not obey $f(x) = \Theta(x^3)$

- Computation complexity is measured in terms of number of basic mathematical operators, such as additions, multiplications, and divisions, or in terms of computing time, and memory requirements.

Complexity of 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

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

$\hat{\mu}$: $d \times n$ additions and 1 division

$\hat{\Sigma}$: $d(d+1)/2$ components each with n multiplications and additions

$|\bullet| = O(d^2)$ and $(\bullet)^{-1} = O(d^3)$,

estimation of $P(\omega) : O(n)$

Complexity of ML Estimation

- Total complexity for one discriminant function is dominated by the $O(d^2n)$.
- For total of c classes, the overall computational complexity for learning in this Bayes classifier is $O(cd^2n) \approx O(d^2n)$ (c is a constant much smaller than d and n)
- Complexity increases as d and n increase.

Computational Complexity

- The computational complexity for classification is usually less than parameter estimation (i.e. learning).
- Given a test point $\boldsymbol{x}$, for each of $\boldsymbol{c}$ categories,
  - Compute $(\boldsymbol{x} - \hat{\boldsymbol{\mu}})$, $O(d)$.
  - Compute the multiplication $(\boldsymbol{x} - \hat{\boldsymbol{\mu}})^t \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \hat{\boldsymbol{\mu}})$, $O(d^2)$
  - Compute the decision $\max_i \boldsymbol{g}_i(\boldsymbol{x})$, $O(c)$
  - So overall for small $\boldsymbol{c}$, classification is an $O(d^2)$ operation.

Overfitting

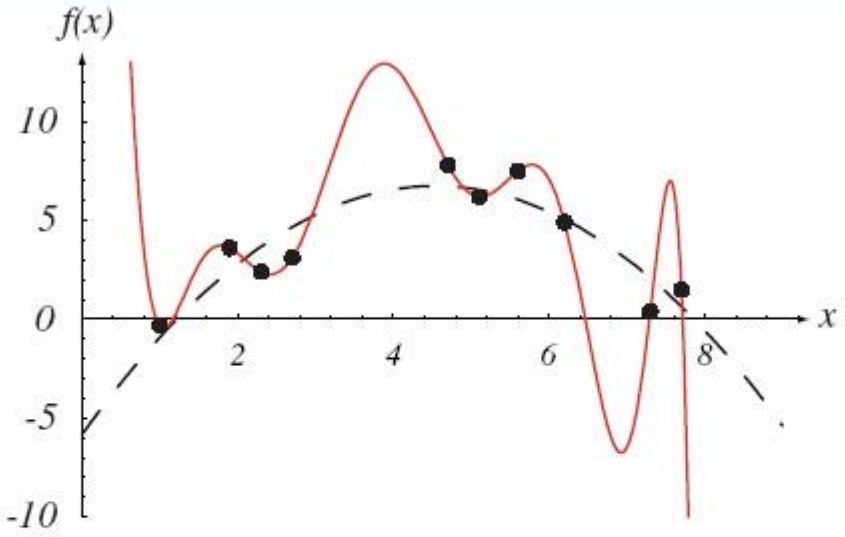


FIGURE 3.4. The "training data" (black dots) were selected from a quadratic function plus Gaussian noise, i.e., $f(x) = ax^2+bx+c+\epsilon$ where $p(\epsilon) \sim N(0, \sigma^2)$. The 10th-degree polynomial shown fits the data perfectly, but we desire instead the second-order function $f(x)$, because it would lead to better predictions for new samples. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Overfitting

- Frequently the number of training sample is inadequate. To address this problem:
  - Reduce the dimensionality
    - Redesign the feature extractor
    - Select a subset of existing features
  - Assume all classes share the same covariance matrix
  - Find a better way to estimate the covariance matrix.
    - With some prior estimate, Bayesian learning can refine this estimate
  - Assume the covariance matrix is diagonal, and off-diagonal elements are all zero, i.e. features are independent.

Overfitting

- Assuming feature independence will generally produce suboptimal classification performance. But in some cases it may outperform ML estimate with full parameter space.
  - This is because of insufficient data.
- To have accurate estimate of model parameters, we need much more data samples than parameters. If we don't, we may have to reduce the parameter space.
  - We can start with a complex model, and then smooth and simplify the model. Such solution may have poor estimation error on training data, but may generalize well on test data.

Component Analysis and Discriminants

- One of the major problems in statistical machine learning and pattern recognition is the curse of dimensionality, i.e. the large value of feature dimension ***d***.
- Combine features in order to reduce the dimension of the feature space – dimension reduction
- Linear combinations are simple to compute and tractable
- Project high dimensional data onto a lower dimensional space

Component Analysis and Discriminants

- Two classical approaches for finding "optimal" linear transformation
  - PCA (Principal Component Analysis) "Projection that best represents the data in a least- square sense"
  - MDA (Multiple Discriminant Analysis) "Projection that best separates the data in a least-squares sense"

Principal Component Analysis

- Given $\boldsymbol{n}$ data samples $\{x_1, x_2, \ldots, x_n\}$ in $\boldsymbol{d}$ dimension.
- Define a linear transform

$$\tilde{\boldsymbol{x}}_{\boldsymbol{k}} = \boldsymbol{m} + a_k \boldsymbol{e}$$

where $\tilde{\boldsymbol{x}}_{\boldsymbol{k}}$ is an approximation of a feature vector $\boldsymbol{x}_{\boldsymbol{k}}$, $\boldsymbol{e}$ is a unit vector in a particular direction and $\|\boldsymbol{e}\|=1$, a_k is the 1-D projection of $\boldsymbol{x}_{\boldsymbol{k}}$ onto basis vector $\boldsymbol{e}$, and $\boldsymbol{m}$ is the sample mean

$$\boldsymbol{m} = \frac{1}{\boldsymbol{n}} \sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} \boldsymbol{x}_{\boldsymbol{k}}$$

- We try to minimize the square error criterion function

$$\boldsymbol{J}(a_1, a_2, \ldots, a_n, \boldsymbol{e}) = \sum_{\boldsymbol{k}=1}^{\boldsymbol{n}} \left\| (\boldsymbol{m} + a_k \boldsymbol{e}) - \boldsymbol{x}_{\boldsymbol{k}} \right\|^2$$

Principal Component Analysis

– To solve a_k

$$J(a_1, a_2, ..., a_n, e) = \sum_{k=1}^{n} \left\| (m + a_k e) - x_k \right\|^2 = \sum_{k=1}^{n} \left\| a_k e - (x_k - m) \right\|^2$$

$$= \sum_{k=1}^{n} a_k^2 \left\| e \right\|^2 - 2\sum_{k=1}^{n} a_k e^t (x_k - m) + \sum_{k=1}^{n} \left\| x_k - m \right\|^2$$

$$\text{let } \frac{\partial J}{\partial a_k} = 0, \quad \text{we have } a_k = e^t (x_k - m)$$

– We also define the scatter matrix

$$S = \sum_{k=1}^{n} (x_k - m)(x_k - m)^t$$

Compare this with the covariance matrix from ML estimation (CPE646-4, p.19)

Principal Component Analysis

– To solve $\boldsymbol{e}$, given $a_k=\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m})$ and $\|\boldsymbol{e}\|=1$, we have

$$J(\boldsymbol{e}) = \sum_{k=1}^{n} a_k^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2$$

$$= -\sum_{k=1}^{n} [\boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m})]^2 + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2$$

$$= -\sum_{k=1}^{n} \boldsymbol{e}^t(\boldsymbol{x}_k - \boldsymbol{m})(\boldsymbol{x}_k - \boldsymbol{m})^t\boldsymbol{e} + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2$$

$$= -\boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e} + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2$$

Principal Component Analysis

- To minimize $\boldsymbol{J(e)}$ is the same as to maximize $\boldsymbol{e^tSe}$. We can use Lagrange Multipliers to maximize $\boldsymbol{e^tSe}$ subject to the constraint $\|\boldsymbol{e}\|=1$

$$\boldsymbol{u = e^tSe - \lambda(e^te - 1)}$$

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{e}} = 2\boldsymbol{Se} - 2\boldsymbol{\lambda e} = 0$$

$$\boldsymbol{Se = \lambda e} \quad \textbf{or} \quad \boldsymbol{e^tSe = \lambda e^te = \lambda}$$

- To maximize $\boldsymbol{e^tSe}$, we should select $\boldsymbol{e}$ as the eigenvector that corresponding to the largest eigenvalue of the scatter matrix .

Principal Component Analysis

- In general, if we extend this 1-dimension projection to ***d'***-dimension projection (***d'*** $\leq$ ***d***)

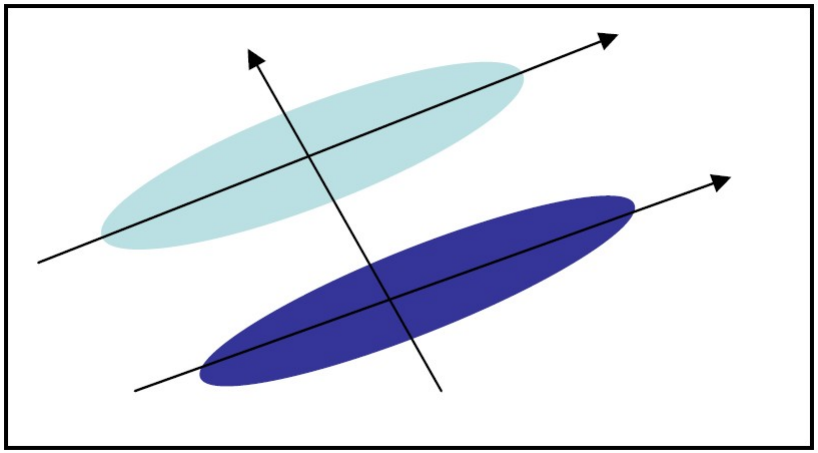
$$\tilde{x}_k = m + \sum_{i=1}^{d'} a_{ik} e_i$$

  the solution is to let e_i be the eigenvectors of the largest ***d'*** eigenvalues of the scatter matrix
- Because the scatter matrix is real and symmetric, the eigenvectors are orthogonal.
- The projections of x_k to each of these eigenvectors (i.e. bases) are called principal components.

Problem of PCA

- The PCA reduces dimension for each class individually. The resulting components are good representation of the data in each class, but they may not be good for discrimination purpose.
- For example, suppose the two classes have 2D Gaussian-like densities, represented by the two ellipsis. They are well separable. But if we project the data to the first principal component (i.e. from 2D to 1D), then they become inseparable (with a very low Chernoff information).
- The best projection is the short axis

Problem of PCA



- In fact, this is typical problem in studying generative models vs discriminative models. The generative models aim at representing the data faithfully, while discriminative models target telling objects apart.

Fisher Linear Discriminant

- Consider a two-class problem.
- Given $\boldsymbol{n}$ data samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ in $\boldsymbol{d}$ dimension. $\boldsymbol{n_1}$ samples in subset (class) $\boldsymbol{D_1}$ labeled with $\boldsymbol{\omega}_1$, and $\boldsymbol{n_2}$ samples in subset (class) $\boldsymbol{D_2}$ labeled with $\boldsymbol{\omega}_2$.
- The linear combination will project a $\boldsymbol{d}$-D feature vector to a 1-D scale $y=\boldsymbol{w}^t\boldsymbol{x}$ and $\|\boldsymbol{w}\|=1$
- The resulting n samples $y_1, \ldots, y_n$ divided into the subsets Y_1 and Y_2.
- We adjust the direction of $\boldsymbol{w}$ to maximize class separability.

Fisher Linear Discriminant

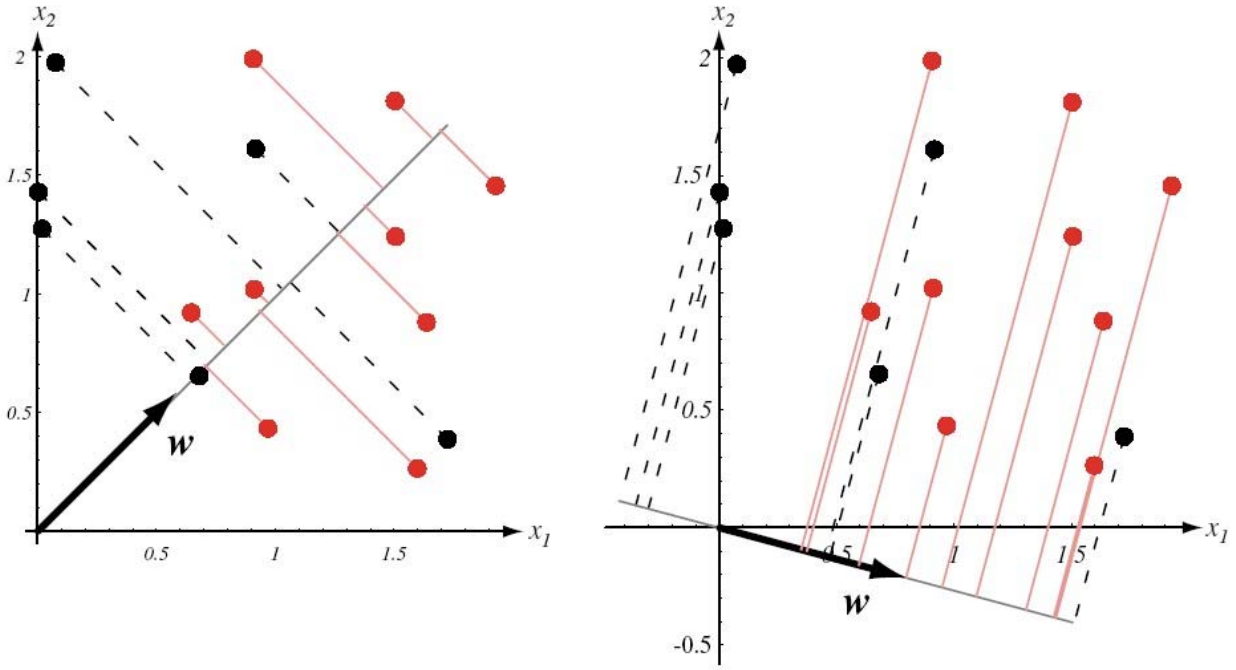


FIGURE 3.5. Projection of the same set of samples onto two different lines in the directions marked **w**. The figure on the right shows greater separation between the red and black projected points. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Fisher Linear Discriminant

$$\boldsymbol{m}_i = \frac{1}{\boldsymbol{n}_i} \sum_{x \in \boldsymbol{D}_i} \boldsymbol{x}$$

$$\tilde{m}_i = \frac{1}{\boldsymbol{n}_i} \sum_{y \in \boldsymbol{Y}_i} y = \frac{1}{\boldsymbol{n}_i} \sum_{y \in \boldsymbol{Y}_i} \boldsymbol{w}^t \boldsymbol{x} = \boldsymbol{w}^t \boldsymbol{m}_i$$

$$\left|\tilde{m}_1 - \tilde{m}_2\right| = \left|\boldsymbol{w}^t (\boldsymbol{m}_1 - \boldsymbol{m}_2)\right|$$

- We define the scatter for projected samples labeled $\boldsymbol{\omega}_1$

$$\tilde{s}_i^2 = \sum_{y \in \boldsymbol{Y}_i} (y - \tilde{m}_i)^2$$

Fisher Linear Discriminant

- The Fisher linear discriminant employs that linear function $\boldsymbol{w^t x}$ for which the criterion function is maximum.

$$J(w) = \frac{\left|\tilde{m}_1 - \tilde{m}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$$

$\left|\tilde{m}_1 - \tilde{m}_2\right|$ **is the distance of projected means --**

between-class scatter

$\tilde{s}_1^2 + \tilde{s}_2^2$ **is called the total within-class scatter**

Fisher Linear Discriminant

- We define the scatter matrices

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^t \quad \text{and}$$

$$\tilde{s}_i^2 = \sum_{x \in D_i} (w^t x - w^t m_i)^2$$

$$= \sum_{x \in D_i} w^t (x - m_i)(x - m_i)^t w$$

$$= w^t S_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^t S_W w$$

$$S_W = S_1 + S_2 \quad \text{within-class scatter matrix}$$

Fisher Linear Discriminant

- The between-class scatter

$$(\tilde{m}_1 - \tilde{m}_2)^2 = (w^t m_1 - w^t m_2)^2$$

$$= w^t (m_1 - m_2)(m_1 - m_2)^t w$$

$$= w^t S_B w$$

where

$S_B = (m_1 - m_2)(m_1 - m_2)^t$ **between-class scatter matrix**

- The criterion function becomes

$$J(w) = \frac{w^t S_B w}{w^t S_W w}$$

(generalized Rayleigh quotient)

Fisher Linear Discriminant

- The solution to maximize $J(w)$ is not unique, so the optimization has to be adjusted to

$$\textbf{optimize } J_1(w) = w^t S_B w$$

$$\textbf{subject to the constraint } J_2(w) = w^t S_W w = 1$$

- The new criterion function becomes

$$g(w, \lambda) = w^t S_B w + \lambda(1 - w^t S_W w)$$

- The vector w that maximize $g(\cdot)$ must satisfy

$$S_B w = \lambda S_W w \textbf{ for some constant } \lambda$$

$$\textbf{if } S_W \textbf{ is not singular, this becomes}$$

$$S_W^{-1} S_B w = \lambda w$$

Fisher Linear Discriminant

- This is an eigenvalue problem, but we do not need to solve for λ and w, since $S_B w$ is always in the direction of $(m_1 - m_2)$, i.e.

$$S_B w = (m_1 - m_2)(m_1 - m_2)^t w = (m_1 - m_2)\mathbf{a}$$

where **a** is a scale.

- Therefore the solution can be

$$w = S_W^{-1}(m_1 - m_2)$$

Fisher Linear Discriminant

- The remaining step is to separate two classes in 1-dimension.
- The complexity of finding $\boldsymbol{w}$ is dominated by calculating $\boldsymbol{S_W^{-1}}$, which is an $\boldsymbol{O(d^2n)}$ calculation.

Fisher Linear Discriminant

- If the class-conditional densities $p(x|\omega_i)$ are multivariate normal with equal covariance matrices Σ, the optimal decision boundary satisfies (CPE646-3, p.22)

$$w^t x + \omega_0 = 0$$

where ω_0 is a constant involving w and the prior probabilities, and

$$w = \Sigma^{-1}(\mu_1 - \mu_2)$$

This w is equivalent to

$$w = S_W^{-1}(m_1 - m_2)$$

Multiple Discriminant Analysis

- For the ***c***-class problem, the generalization of Fisher linear discriminant will project the ***d***-dimensional feature vectors to a **(*c*-1)**-dimensional space, assuming ***d*≥*c*.**
- The generalization of within-class scatter matrix

$$S_W = \sum_{i=1}^{c} S_i \quad \textbf{where}$$

$$S_i = \sum_{x \in D_i} (x - m_i)(x - m_i)^t \quad \textbf{and}$$

$$m_i = \frac{1}{n_i} \sum_{x \in D_i} x$$

Multiple Discriminant Analysis

- The generalization of between-class scatter matrix

define total mean vector $m = \frac{1}{n}\sum_{x} x = \frac{1}{n}\sum_{i=1}^{c} n_i m_i$

define total scatter matrix

$$S_T = \sum_{x} (x - m)(x - m)^t$$

$$= \sum_{i=1}^{c} \sum_{x \in D_i} (x - m_i + m_i - m)(x - m_i + m_i - m)^t$$

$$= S_W + \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^t$$

define a general between-class scatter matrix

$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^t$$

and $S_T = S_W + S_B$

Multiple Discriminant Analysis

- The projection from a ***d***-D to **(*c*-1)**-D can be expressed as

$$y_i = \mathbf{w}_i^t \mathbf{x} \quad \boldsymbol{i} = 1, ..., \boldsymbol{c} - 1 \quad \textbf{or}$$

$$\boldsymbol{y} = \boldsymbol{W}^t \boldsymbol{x}, \textbf{ where } \boldsymbol{w}_i \textbf{ are columns of } \boldsymbol{d} \times (\boldsymbol{c} - 1) \textbf{ matrix } \boldsymbol{W}$$

- The criterion function becomes

$$\boldsymbol{J}(\boldsymbol{W}) = \frac{\left| \boldsymbol{W}^t \boldsymbol{S}_B \boldsymbol{W} \right|}{\left| \boldsymbol{W}^t \boldsymbol{S}_W \boldsymbol{W} \right|}$$

- The solution to maximizing this function satisfies

$$\boldsymbol{S}_B \boldsymbol{w}_i = \lambda_i \boldsymbol{S}_W \boldsymbol{w}_i$$

Multiple Discriminant Analysis

- To solve $\boldsymbol{w_i}$, we can
  - Compute $\boldsymbol{\lambda_i}$ from $\boldsymbol{|S_B - \lambda_i S_W| = 0}$
  - Then compute $\boldsymbol{w_i}$ from $\boldsymbol{(S_B - \lambda_i S_W) w_i = 0}$
- The solution for $\boldsymbol{W}$ is not unique, they all have the same $\boldsymbol{J(W)}$.

Multiple Discriminant Analysis

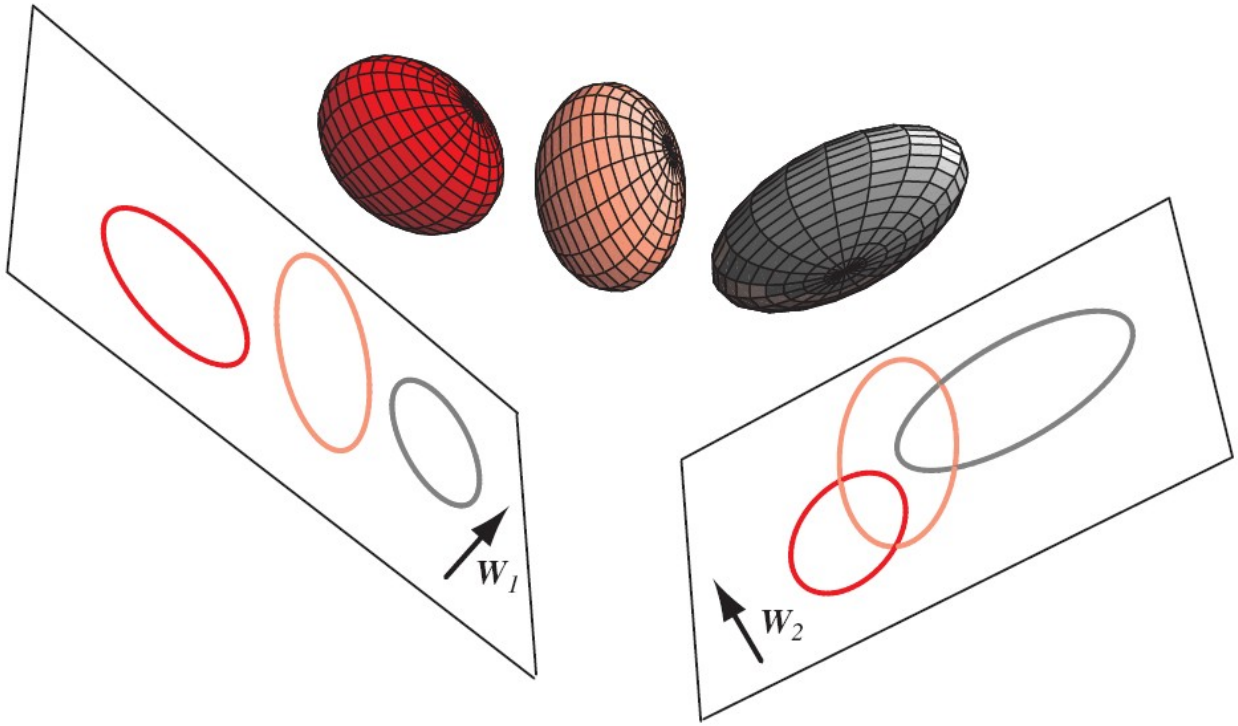


FIGURE 3.6. Three three-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors $\mathbf{W}_1$ and $\mathbf{W}_2$. Informally, multiple discriminant methods seek the optimum such subspace, that is, the one with the greatest separation of the projected distributions 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*.