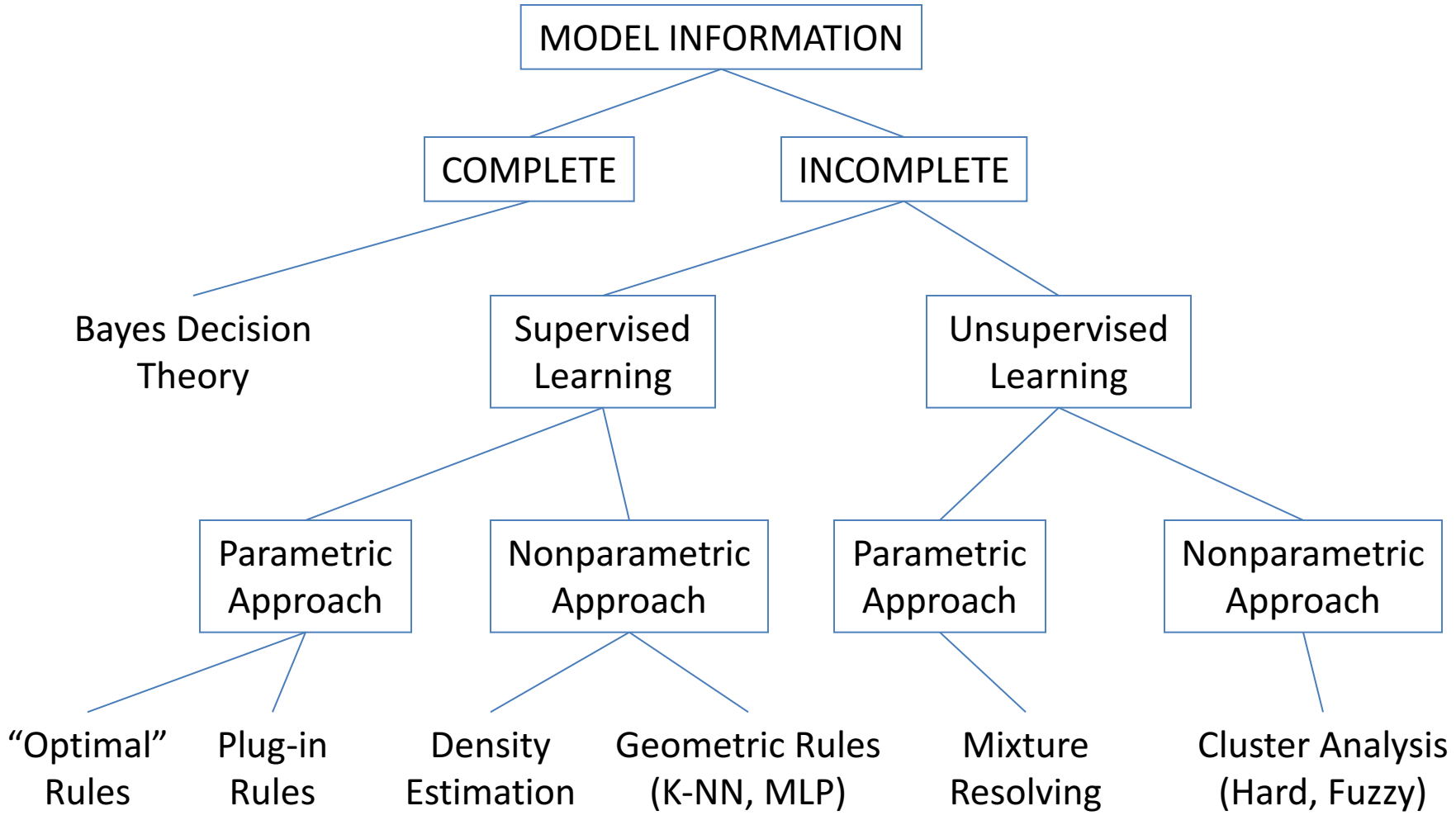
Course Outline



Supervised Learning

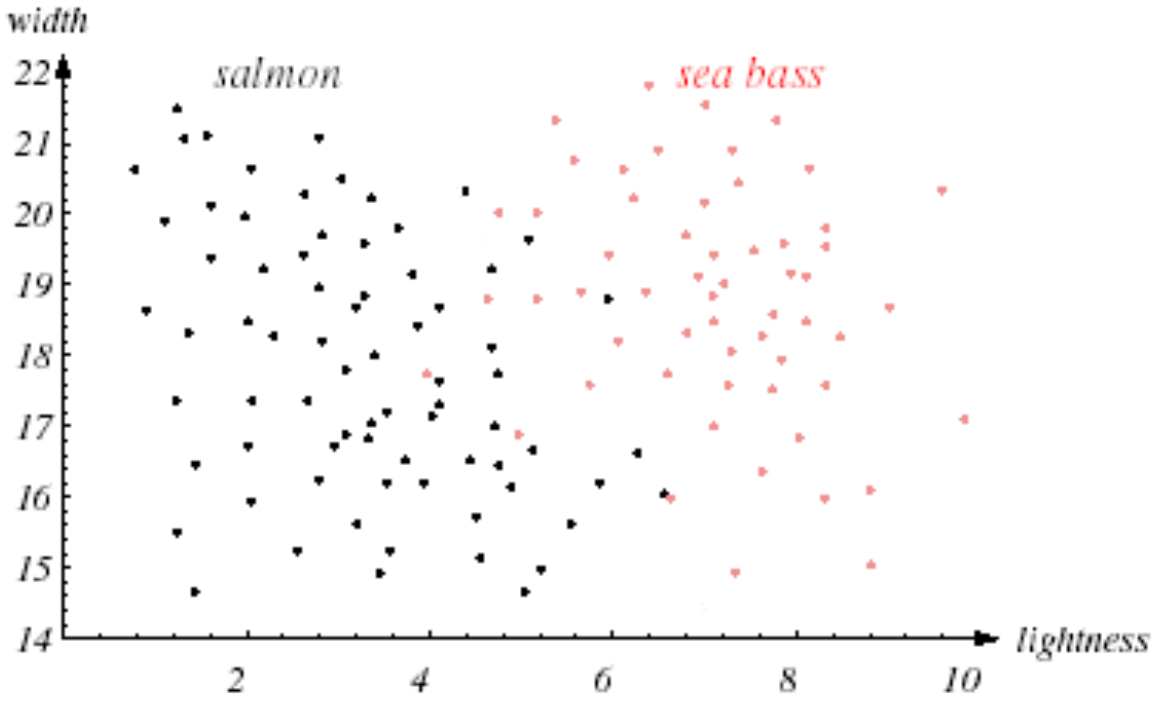


FIGURE 1.4. The two features of lightness and width for sea bass and salmon. The dark line could serve as a decision boundary of our classifier. Overall classification error on the data shown is lower than if we use only one feature as in Fig. 1.3, but there will still be some errors. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Chapter 3: Maximum-Likelihood & Bayesian Parameter Estimation

- Introduction
- Maximum-Likelihood Estimation
- Bayesian Estimation
- Curse of Dimensionality
- Component analysis & Discriminants

- Bayesian framework
  - To design an optimal classifier we need:
    - $P(\omega_i)$: priors
    - $P(x \mid \omega_i)$: class-conditional densities

    What if this information is not available?

- Supervised Learning: Design a classifier based on a set of labeled training samples
  - Assume priors are known
  - Sufficient no. of training samples available to estimate $P(x \mid \omega_i)$

- Assumption:
  - Parametric model of $P(x \mid \omega_i)$ is available
- For example, for Gaussian pdf assume

$$P(x \mid \omega_i) \sim N(\mu_i, \Sigma_i),\ i = 1,..,c$$

Parameters (μ_i, Σ_i) are not known, but labeled training samples are available to estimate them

- Parameter estimation
  - Maximum-Likelihood (ML) estimation
  - Bayesian estimation
  - For large n, estimates from the two methods are nearly identical

- ML parameter estimation (MLE):
  - Parameters are assumed to be fixed but unknown!
  - Best parametric estimates are obtained by maximizing the probability of obtaining the samples observed
- Bayesian parameter estimation:
  - Unknown parameters are random variables with some known prior distribution;
  - Use prior and samples to obtain the posteriori density
  - Parameter estimate is derived from posteriori & loss fn.
- Both methods use $P(\omega_i \mid x)$ for decision rule!

Maximum-Likelihood Parameter Estimation

- Has good convergence properties as the sample size increases; estimated parameter value approaches the true value as n increases
- Most simple method for parameter estimation

- 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

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

Use class ω_j samples to estimate class ω_j parameters: μ_j, Σ_j

- Use the training samples to estimate θ

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

θ_i (i = 1, 2, …, c) is parameter for the ω_i

- Sample set D contains n iid samples, $x_1, x_2, \ldots, x_n$

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

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

- ML estimate of θ is the value $\hat{\theta}$ that maximizes P(D | θ)

  It is the value of θ that best agrees with the observed training samples

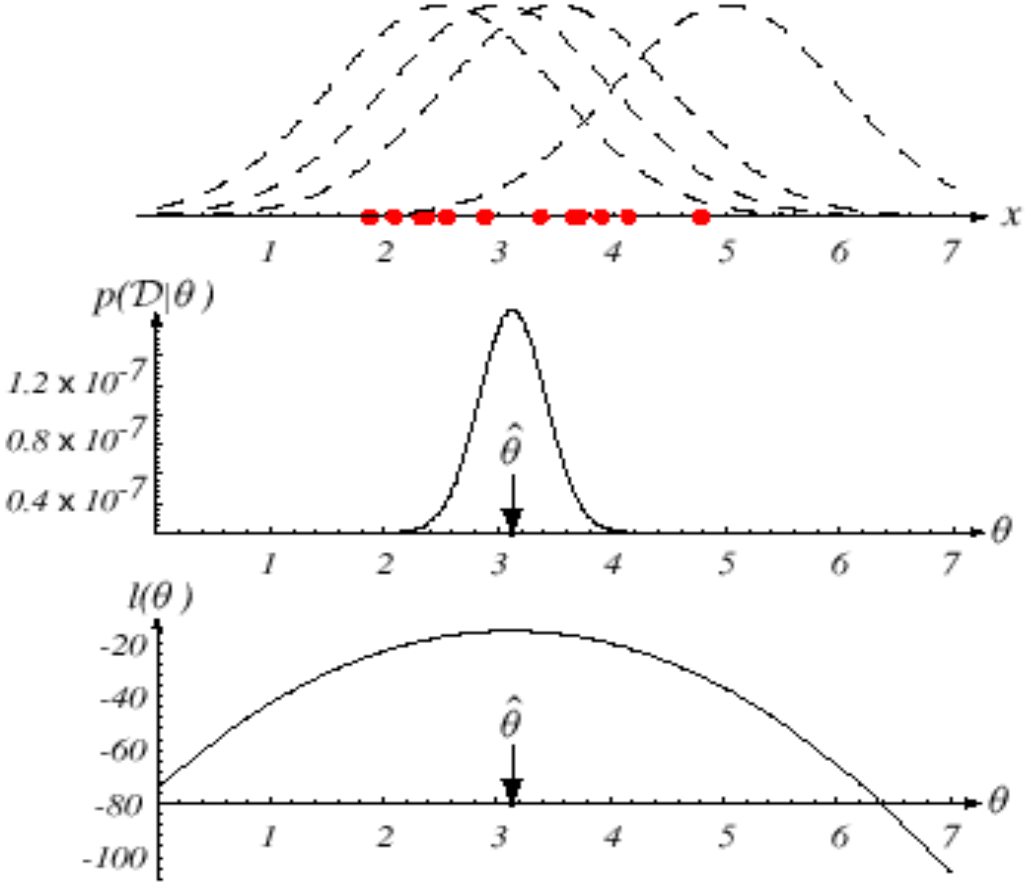


FIGURE 3.1. The top graph shows several training points in one dimension, known or assumed to be drawn from a Gaussian of a particular variance, but unknown mean. Four of the infinite number of candidate source distributions are shown in dashed lines. The middle figure shows the likelihood $p(\mathcal{D}|\theta)$ 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*.

- ML estimation
  - Let $\theta = (\theta_1, \theta_2, \ldots, \theta_p)^t$ and ∇_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

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

  - Determine θ that maximizes the log-likelihood

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

Set of necessary conditions for an optimum:

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

$$\nabla_\theta l = 0$$

- $P(x \mid \mu) \sim N(\mu, \Sigma)$; μ is not known but Σ is known
  Samples are drawn from a multivariate Gaussian

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

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

The ML estimate for μ must satisfy:

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

- Multiplying by Σ and rearranging terms:

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

MLE of the mean of the Gaussian distribution is the "sample mean"

Conclusion:

Given $P(x_k \mid \omega_j, \theta_j)$, j = 1, 2, …,c to be Gaussian in *d*-dimensions, estimate the vector $\theta = (\theta_1, \theta_2, \ldots, \theta_c)^t$ and then use the maximum a posteriori rule (Bayes decision rule)

- ML Estimation:
  - Univariate Gaussian Case: *unknown* μ & σ

    $\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$
  - For the kth sample (observation)

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

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

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

Introduce summation to account for n samples:

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

Combining (1) and (2), we get:

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

ML estimate for σ^2 is biased

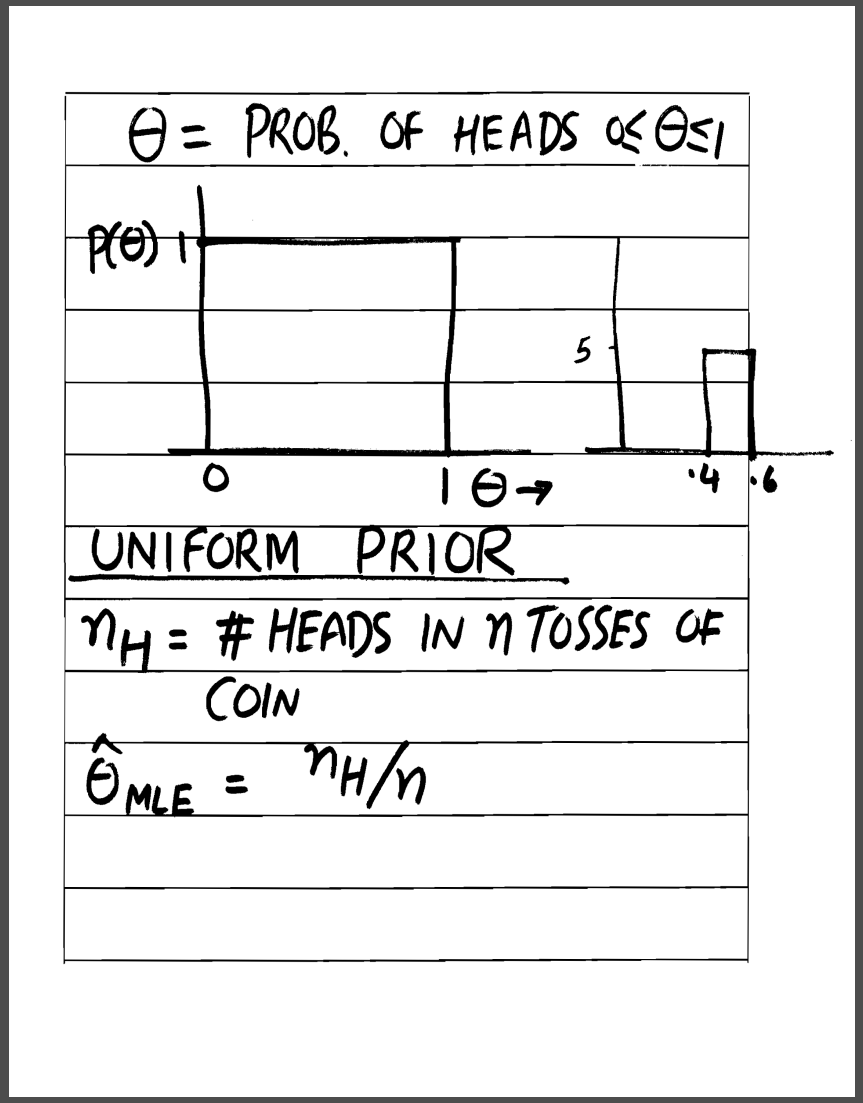
$$E\left[\frac{1}{n}\Sigma(x_i - \bar{x})^2\right] = \frac{n-1}{n}.\sigma^2 \neq \sigma^2$$

An unbiased estimator for Σ is:

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

ML vs. Bayesian Parameter Estimation

Unknown Parameter is the Prob. of Heads of a coin



$$p(n_H|\theta) = \binom{n}{n_H} \theta^{n_H} (1-\theta)^{n-n_H}$$

$$p(\theta|n_H) = \frac{p(n_H|\theta)p(\theta)}{\int_0^1 p(n_H|\theta)p(\theta)d\theta}$$

$$\int_0^1 p(n_H|\theta)p(\theta)d\theta = \binom{n}{n_H}\int_0^1 \theta^{n_H}(1-\theta)^{n-n_H} d\theta$$

$$= \binom{n}{n_H} \frac{\Gamma(n_H+1)\Gamma(n-n_H+1)}{\Gamma(n+2)}$$

$$= \frac{1}{n+1}$$

$$p(\theta|n_H) = (n+1)\binom{n}{n_H}\theta^{n_H}(1-\theta)^{n-n_H}$$

FOR SQUARED-ERROR LOSS FN.

$$\hat{\theta}_{BAYES} = \int_0^1 \theta\, p(\theta | n_H)\, d\theta$$

$$= \frac{n_H + 1}{n+2}$$

$$\hat{\theta}_{MLE} = \frac{n_H}{n}$$

(1) SUPPOSE $n = 0$,

$\hat{\theta}_{MLE}$ UNDEFINED

$\hat{\theta}_{BAYES} = \frac{1}{2}$!!

(2) $n = 1, n_H = 0$

$\hat{\theta}_{MLE} = 0$

$\hat{\theta}_{BAYES} = \frac{1}{3}$

(3) n LARGE

$\frac{n_H + 1}{n+2} \rightarrow \frac{n_H}{n}$

THEOREM

- The Bayes estimator $\hat{\theta}$ for the quadratic loss function

$$L(\theta, \hat{\theta}) = (\theta - \hat{\theta})^2$$

is the conditional expectation

$$\hat{\theta} = E_\theta[\theta | x_1, \cdots x_n] = \int \theta\, p(\theta | x_1, \cdots, x_n)\, d\theta$$

$$= E_\theta[\theta | D] = \int \theta\, p(\theta | D)\, d\theta$$

PROOF

- THE AVERAGE RISK FOR THE QUADRATIC LOSS FUNCTION IS

$$R = \int \underbrace{\left[\int L(\theta, \hat{\theta})\, p(\theta|D)\, d\theta\right]}_{\text{AVG. CONDITIONAL RISK}} p(D)\, dD$$

$$= \int \left[\int (\theta - \hat{\theta})^2\, p(\theta|D)\, d\theta\right] p(D)\, dD$$

SINCE $p(D) \equiv p(x_1, x_2, \cdots, x_n)$ IS NONNEGATIVE, R IS MINIMUM IF WE MINIMIZE

$$\int (\theta - \hat{\theta})^2\, p(\theta|D)\, d\theta$$

for every $D = (x_1, x_2, \cdots, x_n)$

TAKING A PARTIAL DERIVATIVE OF

$$\int (\theta - \hat{\theta})^2 \, p(\theta|D) \, d\theta$$

W.R.T. $\hat{\theta}$ AND SETTING IT TO ZERO

$$\int 2(\theta - \hat{\theta}) \, p(\theta|D) \, d\theta = 0$$

$$\hat{\theta} \underbrace{\int p(\theta|D) \, d\theta}_{=1} = \int \theta \, p(\theta|D) \, d\theta$$

$$\boxed{\hat{\theta}_{\text{Bayes}} = \int \theta \, p(\theta|D) \, d\theta}$$

0-1 LOSS FUNCTION

MINIMIZE $\int L(\theta, \hat{\theta}) \, p(\theta|D) \, d\theta$

$$L(\theta, \hat{\theta}) = \begin{cases} 0 & \theta = \hat{\theta} \\ 1 & \theta \neq \hat{\theta} \end{cases}$$

MINIMIZE $1 - \int p(\hat{\theta}|D) \, d\theta$

$\equiv 1 - p(\hat{\theta}|D)$

OR MAXIMIZE $p(\hat{\theta}|D)$

- Bayesian Estimation (Bayesian learning)
  - In MLE θ was supposed to have a fixed value
  - In Bayesian learning θ is a random variable
  - Direct estimation 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 training sample set D, Bayes formula can be written

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

- Derivation of the preceding equation:

$$P(x, D \mid \omega_i) = P(x \mid D\omega_i).P(D \mid \omega_i)$$

$$P(x \mid D) = \sum_j P(x, \omega_j \mid D)$$

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

Thus :

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

- Bayesian Parameter Estimation: Gaussian Case

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

- The univariate Gaussian 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!

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

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

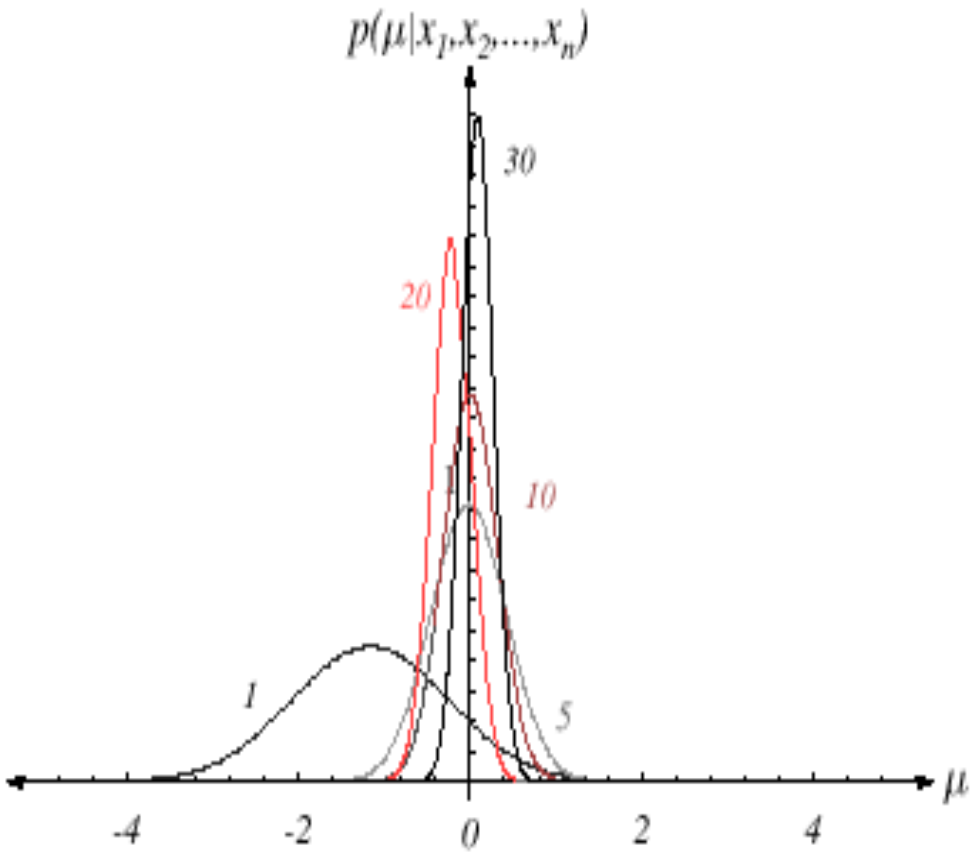
- Reproducing density

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

The updated parameters of the prior:

$$\mu_n = \left(\frac{n\sigma_0^2}{n_0\sigma_0^2 + \sigma^2} \right) \hat{\mu}_n + \frac{\sigma^2}{n\sigma_0^2 + \sigma^2} .\mu_0$$

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



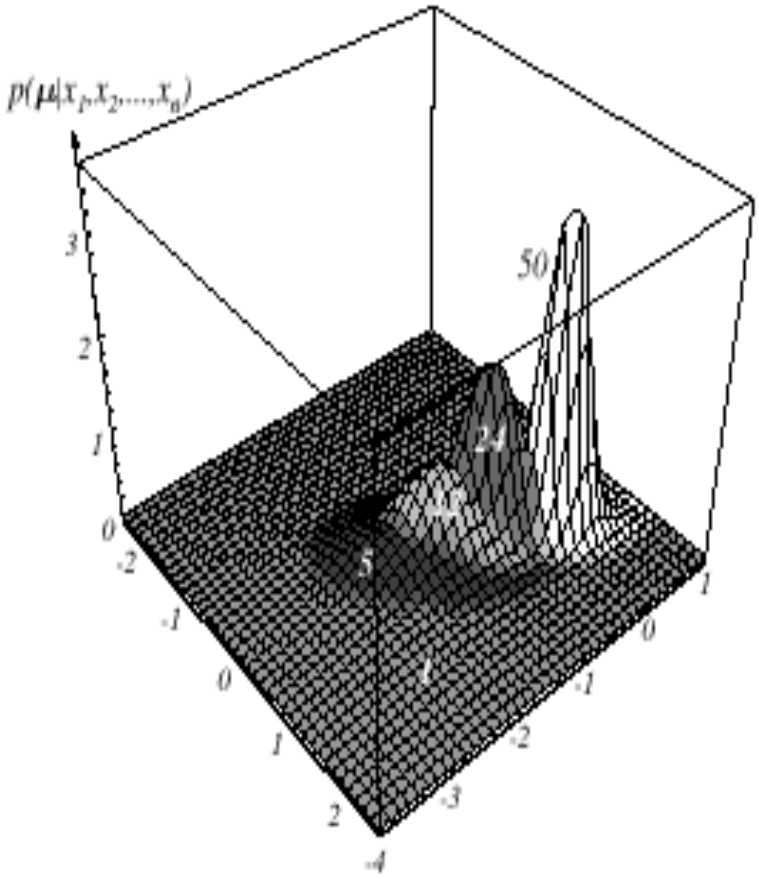


FIGURE 3.2. Bayesian learning of the mean of normal distributions in one and two dimensions. The posterior distribution estimates are labeled by the number of training samples used in the estimation. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001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

- The univariate case P(x | $\mathcal{D}$)
  - P(μ | $\mathcal{D}$) has been computed
  - P(x | $\mathcal{D}$) remains to be computed!

$$P(x \mid \mathcal{D}) = \int P(x \mid \mu).P(\mu \mid \mathcal{D})d\mu \text{ is Gaussian}$$

It provides:

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

Desired class-conditional density P(x | $\mathcal{D}_j$, ω_j)

P(x | $\mathcal{D}_j$, ω_j) together with P(ω_j) and using Bayes formula, we obtain the Bayesian classification rule:

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

- Bayesian Parameter Estimation: General Theory

  - P(x | $\mathcal{D}$) 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 knowledge about θ is assumed to be contained in a known prior density P(θ)
    - The rest of our knowledge about θ is contained in a set $\mathcal{D}$ of n random variables $x_1, x_2, \ldots, x_n$ drawn from P(x)

The basic problem is:

1. Compute the posterior density $P(\theta \mid D)$
2. Derive $P(x \mid D)$

Using Bayes formula, we have:

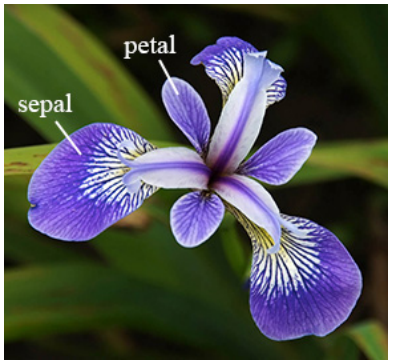
$$P(\theta \mid D) = \frac{P(D \mid \theta).P(\theta)}{\int P(D \mid \theta).P(\theta)d\theta},$$

And by independence assumption:

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

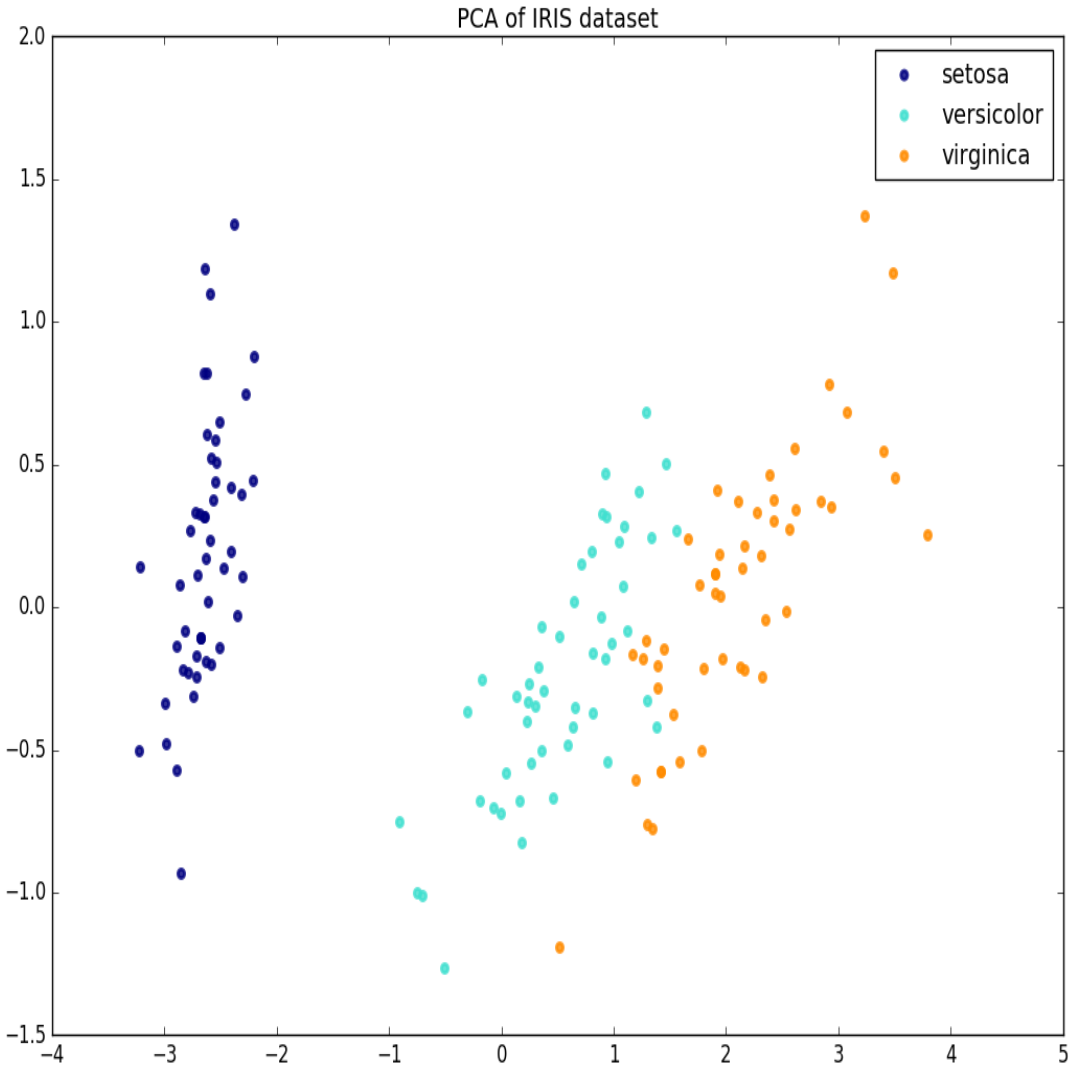
Iris Dataset

- Three types of iris flower: Setosa, Versicolor, Virginica
- Four features: Sepal length, sepal width, petal length petal width (all in cm.)
- 50 patterns/class
- Available in UCI Machine Learning Repository



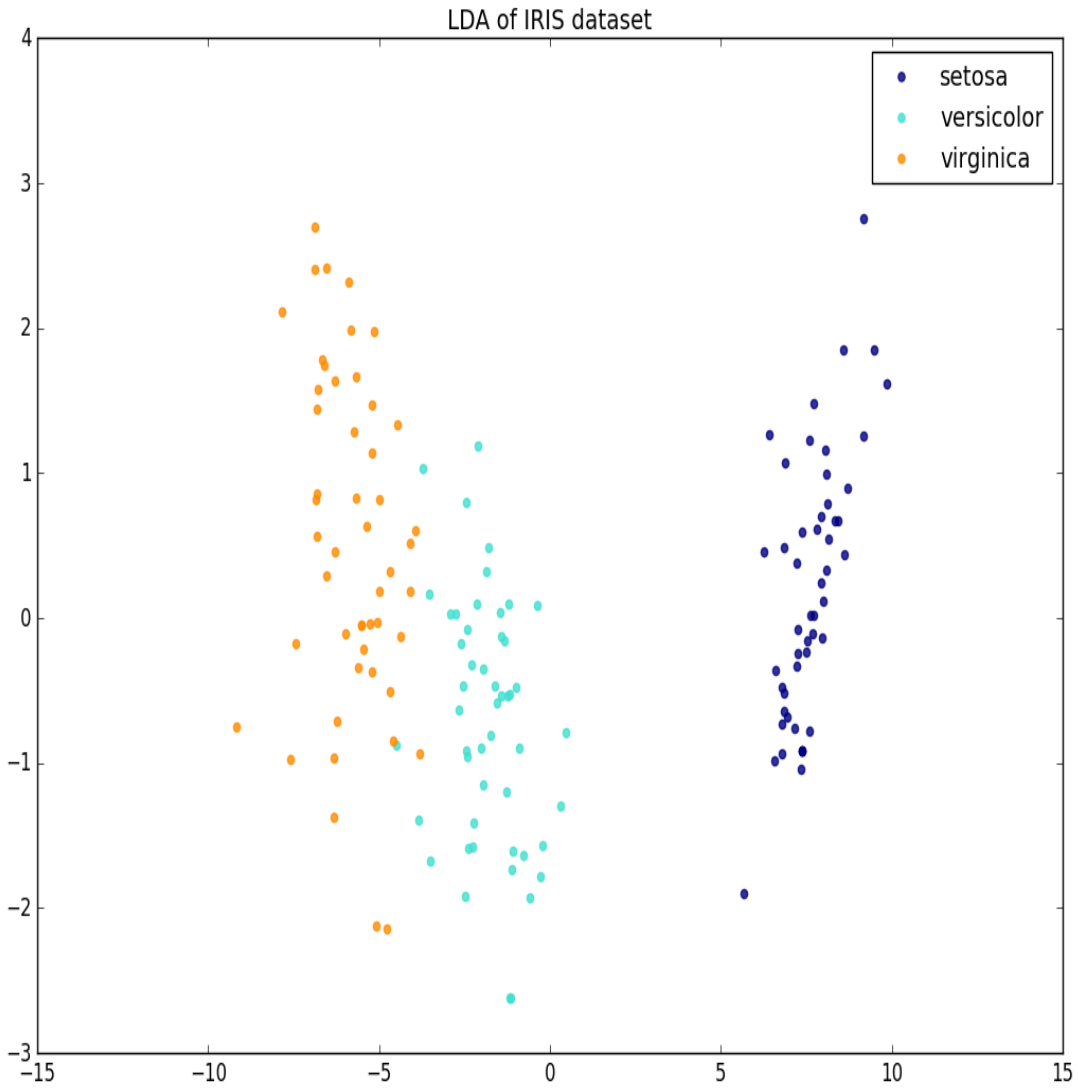
Fisher, R. A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936)

PCA



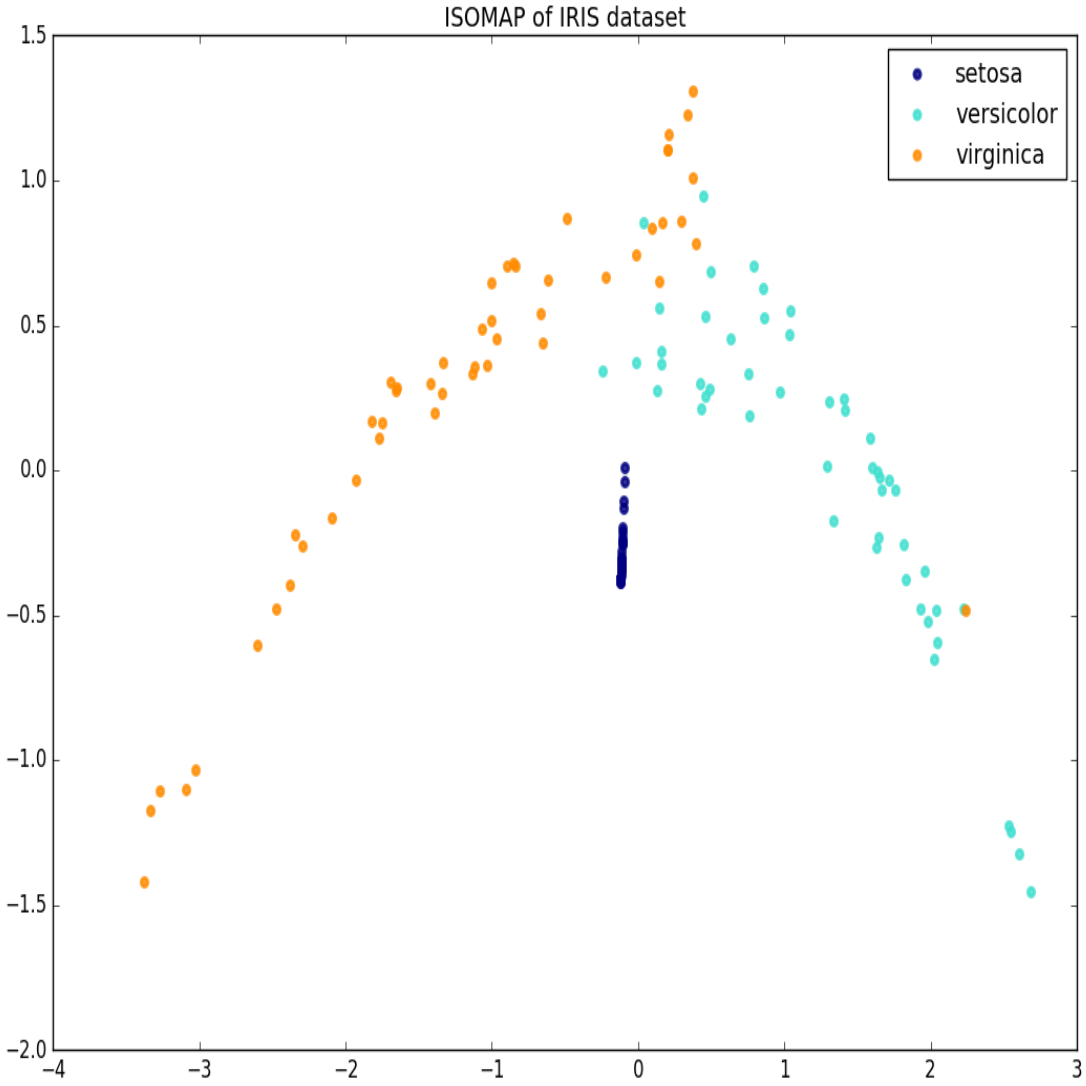
Explained variance ratio	
1st component	0.925
2nd component	0.053

LDA



Explained variance ratio	
1st component	0.992
2nd component	0.009

ISOMAP


ISOMAP of IRIS dataset
setosa
versicolor
virginica
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−4
−3
−2
−1
0
1
2
3

Low Dimensional Embedding of High Dimensional Data

- Given n patterns in a d-dim space, embed the points in m dimensions, m<<d
- Purpose: data compression; avoid overfitting by reducing dimensionality; find “meaningful” low-dim structures in their high-dimensional observations
- Feature selection v. feature extraction
- Feature extraction: linear v. non-linear
- Linear feature extraction or projection: unsupervised (PCA) v. supervised (LDA)
- Non-linear feature extraction (Isompap)

Eigen Decomposition

- Given a linear transformation **A**, a non-zero vector *w* is an eigen-vector of **A** if it satisfies the *eigenvalue* equation for some scalar λ

$$\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$$

Solution:

$$\mathbf{A}\mathbf{w} - \lambda\mathbf{I}\mathbf{w} = 0$$
$$\Rightarrow (\mathbf{A} - \lambda\mathbf{I})\mathbf{w} = 0$$
$$\Rightarrow \det(\mathbf{A} - \lambda\mathbf{I}) = 0$$

(Characteristic equation)

Ex 1. $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

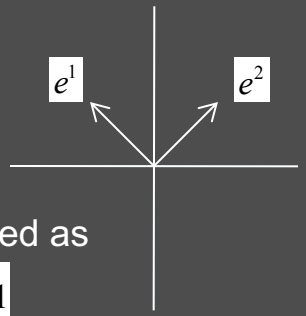
Eigenvalue:

$$\det\begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix} = 0$$
$$(2-\lambda)^2 - 1 = 0$$
$$\lambda^2 - 4\lambda + 3 = 0$$
$$\Rightarrow \lambda_1 = 1 \text{ and } \lambda_2 = 3$$

Eigenvector:

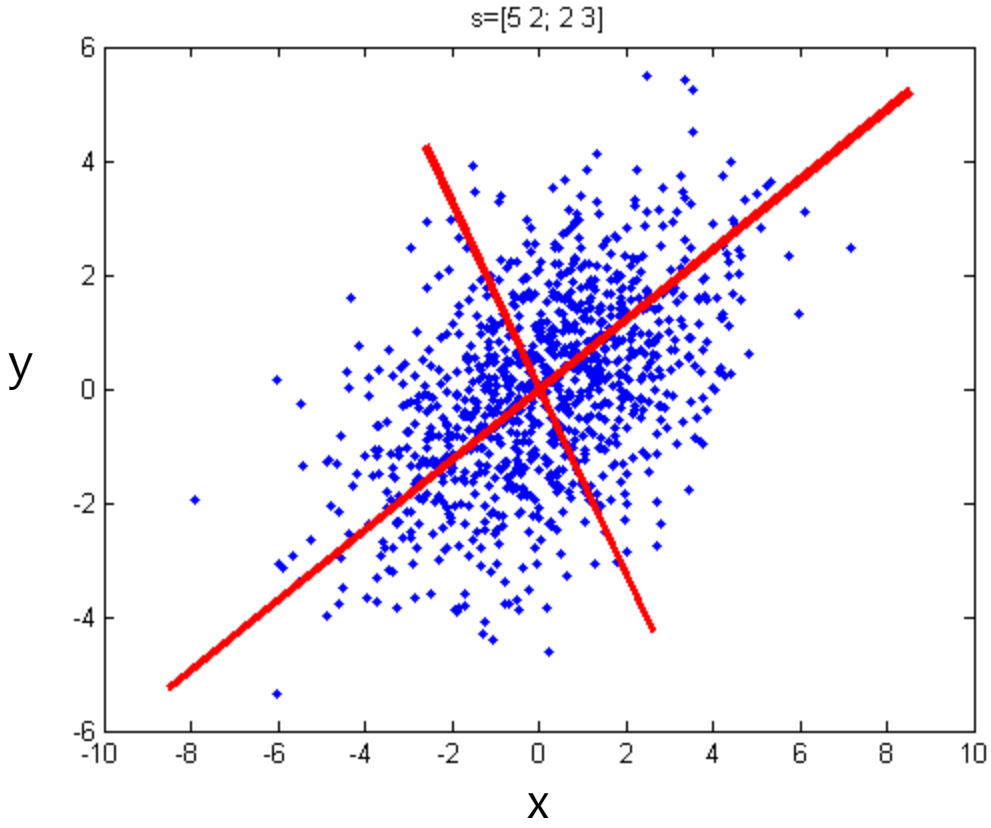
$$\begin{bmatrix} 2-\lambda_1 & 1 \\ 1 & 2-\lambda_1 \end{bmatrix}\begin{bmatrix} e_1^1 \\ e_2^1 \end{bmatrix} = 0$$
$$\begin{bmatrix} 2-\lambda_2 & 1 \\ 1 & 2-\lambda_2 \end{bmatrix}\begin{bmatrix} e_1^2 \\ e_2^2 \end{bmatrix} = 0$$

$$\begin{bmatrix} e_1^1 \\ e_2^1 \end{bmatrix} = \begin{bmatrix} -0.7071 \\ 0.7071 \end{bmatrix}$$
$$\begin{bmatrix} e_1^2 \\ e_2^2 \end{bmatrix} = \begin{bmatrix} 0.7071 \\ 0.7071 \end{bmatrix}$$



Eigenvector is normalized as

$$\sqrt{e_1^2 + e_2^2} = 1$$

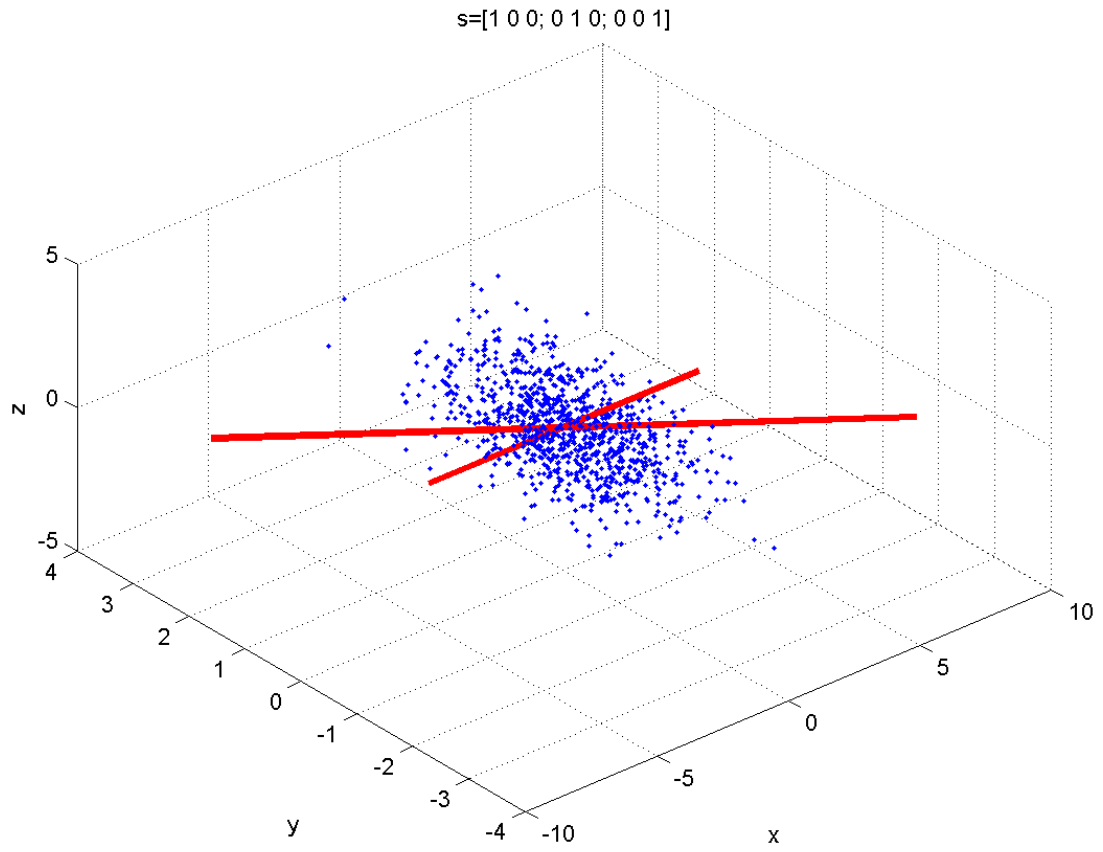


$\boldsymbol{\mu} = [2, 1]$

$\boldsymbol{\Sigma} = \begin{bmatrix} 5 & 2 \\ 2 & 3 \end{bmatrix}$

Eigenvectors: $\begin{bmatrix} 0.5238 & -0.8519 \\ -0.8519 & -0.5238 \end{bmatrix}$

Eigenvalues: $\begin{bmatrix} 1.7230 & 0 \\ 0 & 5.6644 \end{bmatrix}$



$\boldsymbol{\mu} = [4, 2, 1]$

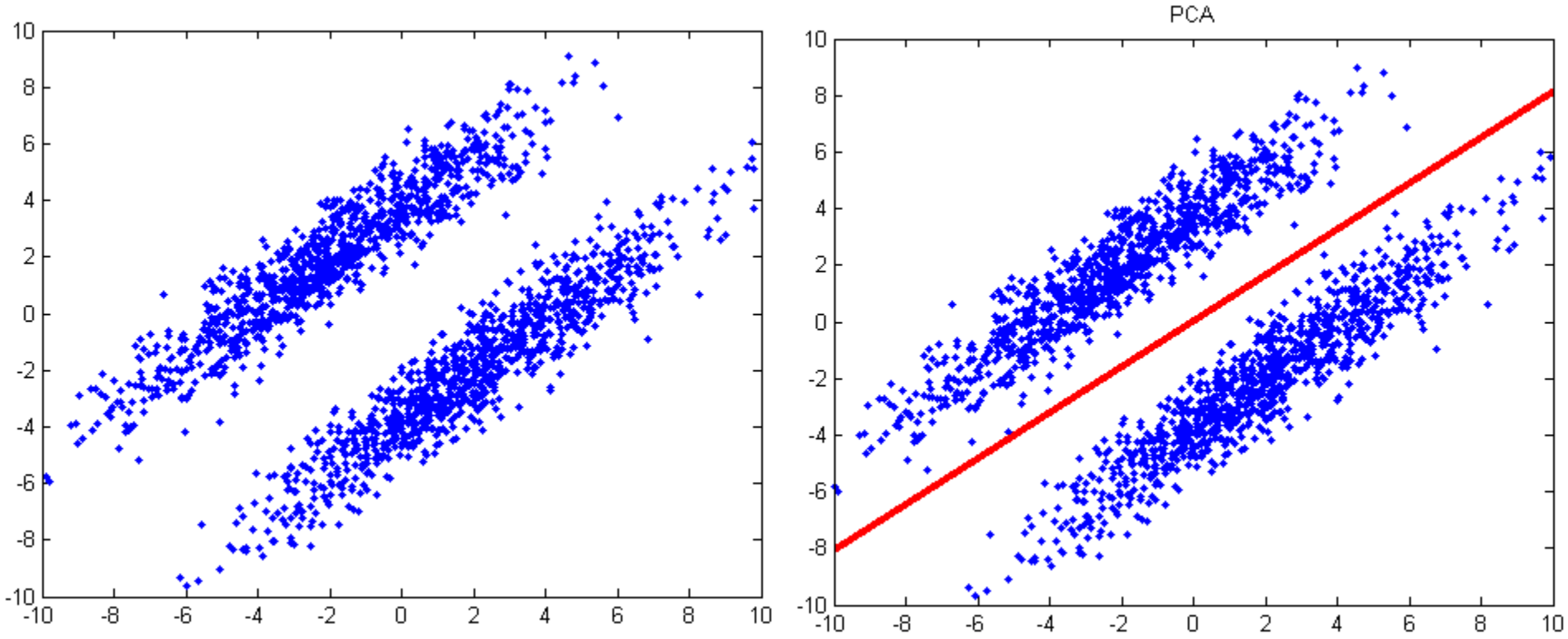
Eigenvectors: $\begin{bmatrix} 0.2190 & 0.0522 & -0.9743 \\ 0.8735 & -0.4554 & 0.1720 \\ 0.4347 & 0.8888 & 0.1453 \end{bmatrix}$

$\boldsymbol{\Sigma} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Eigenvalues: $\begin{bmatrix} 480.4256 & 0 & 0 \\ 0 & 498.6763 & 0 \\ 0 & 0 & 568.5106 \end{bmatrix}$

PCA

$$Y = w^T X$$



Find a transformation **w**, such that the $\mathbf{w}^T\mathbf{x}$ is dispersed the most (maximum distribution)

Scatter Matrices

- **m =** mean vector of all n patterns (grand mean)
- **mi =** mean vector of class i patterns
- $\mathbf{S_W}$ = *within-class scatter matrix*. It is proportional to the sample covariance matrix for the pooled *d*-dimensional data. It is symmetric and positive semidefinite, and is usually nonsingular if $n > d$
- $\mathbf{S_B}$ = *between-class scatter matrix*. It is symmetric and positive semidefinite, but because it is the outer product of two vectors, its rank is at most (C-1)
- $\mathbf{S_T}$ = total scatter of all n patterns
- For any **w**, $\mathbf{S_B w}$ is in the direction of $(\mathbf{m_1 - m_2})$

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in D_i} \mathbf{x} \qquad (92)$$

$$\mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \qquad (97)$$

$$\mathbf{S}_W = \sum_{i=1}^{c} \mathbf{S}_i \qquad (109)$$

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t \qquad (115)$$

$$\mathbf{S}_T = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t \qquad (113)$$

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B \qquad (116)$$

Principal Component Analysis (PCA)

- What is the best representation of *n* *d*-dim samples $\mathbf{x_1},\ldots,\mathbf{x_n}$ by a single point $\mathbf{x_0}$?
- Find $\mathbf{x_0}$ such that the sum of the squared distances between $\mathbf{x_0}$ and all $\mathbf{x_k}$ is minimized
- Define squared-error criterion function $J_0(\mathbf{x_0})$ by

$$J_o\left(\mathbf{x_0}\right)=\sum_{k=1}^{n}\left\|\mathbf{x_0}-\mathbf{x_k}\right\|^2,$$

  and find $\mathbf{x_0}$ that minimizes J_0.
- The solution is given by $\mathbf{x_0}=\mathbf{m}$, where $\mathbf{m}$ is the sample mean,

$$\mathbf{m}=\frac{1}{n}\sum_{k=1}^{n}\mathbf{x_k}.$$

Principal Component Analysis

- Sample mean is a zero-dim representation of data; It does not reveal any of the data variability
- What is the best one-dim representation?
- Project data to a line through the sample mean. If e is a unit vector in the direction of the line, equation of the line can be written as

$$\mathbf{x} = \mathbf{m} + a\mathbf{e},$$

Representing $\mathbf{x_k}$ by $\mathbf{m}+a_k\mathbf{e}$, find the "optimal" set of coefficients a_k by minimizing the squared error

$$J_1(a_1,\ldots,a_n,\mathbf{e}) = \sum_{k=1}^{n} \| (\mathbf{m}+a_k\mathbf{e}) - \mathbf{x_k} \|^2 = \sum_{k=1}^{n} \| a_k\mathbf{e} - (\mathbf{x_k}-\mathbf{m}) \|^2$$

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

Principal Component Analysis

- Since $\|\mathbf{e}\| = 1$ differentiate with respect to a_k, and set the derivative to zero

$$a_k = \mathbf{e}^t(\mathbf{x_k} - \mathbf{m}).$$

- To obtain a least-squares solution, project the vectors $\mathbf{x_k}$ to the line in the direction of e that passes through the sample mean
- What is the *best direction* e for the line? The solution involves the *scatter matrix* **S**

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

- The best direction is the eigenvector of the scatter matrix with the largest eigenvalue

Principal Component Analysis

- Scatter matrix $\mathbf{S_T}$ is real and symmetric; it's eigenvectors are orthogonal and form a set of basis vectors for representing any vector $\mathbf{x}$
- The coefficients a_i in Eq. (89) are the components of x in that basis, called the *principal components*
- Data points $\mathbf{x_1},\ldots\mathbf{x_n}$ can be viewed as a cloud in *d*-dimensions; eigenvectors of the scatter matrix are the principal axes of the point cloud
- PCA reduces dimensionality by restricting attention to those directions along which the scatter of the cloud is greatest (largest eigenvalues)

Face Representation using PCA and LDA

Input face

EigenFaces

56.4 38.6 -19.7 9.8 -45.9 19.6 - 14.2 ...

Fisherfaces

18.3 35.6 -17.5 -27.6 60.6 -20.8 41.9 -9.6 ...

Reconstructed face

PCA

Minimize reconstruction error

LDA

Maximize between-class to within-class scatter

Discriminant Analysis

- PCA finds components that explain data variance; the components may not be useful for discrimination between different classes
- Since no category label is used, components discarded by PCA might be exactly those that are needed for distinguishing between classes
- Whereas PCA seeks directions that are effective for representation, *discriminant analysis* seeks directions that are effective for discrimination
- Special case of multiple discriminant analysis is Fisher linear discriminant for C=2

Fisher Linear Discriminant

- Given n d-dim samples $\mathbf{x_1}$, ..., $\mathbf{x_n}$; n_1 in the subset D_1 labeled ω_1 and n_2 in the subset D_2 labeled ω_2
- Find a projection that maintains separation present in the d-dim. space

$$y = \mathbf{w}^t \mathbf{x}$$

- Geometrically, if $||\mathbf{w}|| = 1$, each y_i is the projection of the corresponding $\mathbf{x_i}$ onto a line in the direction of $\mathbf{w}$. The magnitude of $\mathbf{w}$ is of no significance, since it merely scales y
- Find $\mathbf{w}$ s.t. if d-dim samples labeled ω_1 fall more or less into one cluster while those labeled ω_2 fall in another, we want the projected points onto the line to be well separated as well

Fisher Linear Discriminant

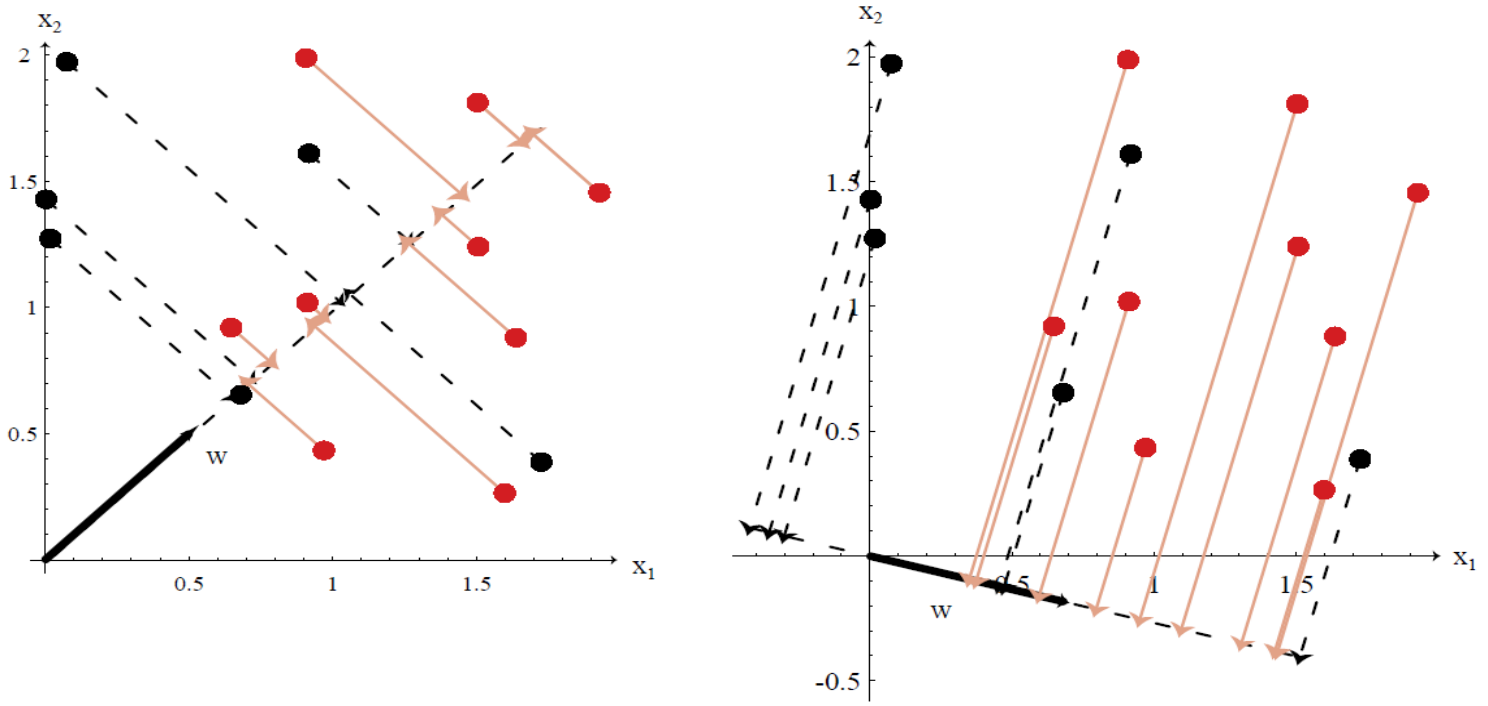


Figure 3.5 illustrates the effect of choosing two different values for **w** for a two-dimensional example. If the original distributions are multimodal and highly overlapping, even the "best" **w** is unlikely to provide adequate separation

Fisher Linear Discriminant

- Fisher linear discriminant is the linear function that maximizes ratio of between-class scatter to within-class scatter
- 2-class classification problem has been converted from the given *d*-dimensional space to one-dimensional projected space
- Find a threshold, i.e., a point along the one-dimensional subspace separating the projected points from the two classes

Fisher Linear Discriminant

- In terms of $\mathbf{S_B}$ and $\mathbf{S_W}$, the criterion function $J(\cdot)$ can be written as

$$J(\mathbf{w}) = \frac{\mathbf{w}^t \mathbf{S}_B \mathbf{w}}{\mathbf{w}^t \mathbf{S}_W \mathbf{w}}.$$

- A vector $\mathbf{w}$ that maximizes $J(\cdot)$ must satisfy

$$\mathbf{S}_B \mathbf{w} = \lambda \mathbf{S}_W \mathbf{w},$$

  for some constant λ, which is a generalized eigenvalue problem

- If $\mathbf{S_W}$ is nonsingular we can obtain a conventional eigenvalue problem by writing

$$\mathbf{S}_W^{-1} \mathbf{S}_B \mathbf{w} = \lambda \mathbf{w}.$$

Fisher Linear Discriminant

In our particular case, it is unnecessary to solve for the eigenvalues and eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ due to the fact that $\mathbf{S_B w}$ is always in the direction of $\mathbf{m_1 - m_2}$. Since the scale factor for **w** is immaterial, we can immediately write the solution for the w that optimizes $J(\cdot)$:

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2).$$

Fisher Linear Discriminant

- When the conditional densities $p(\mathbf{x}|\omega_i)$ are multivariate normal with equal covariance $\mathbf{\Sigma}$, the threshold can be computed directly from the optimal decision boundary (Chapter 2)

$$\mathbf{w}^t\mathbf{x} + w_0 = 0$$
$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2),$$

  where w_0 is a constant involving w and the prior.

- Thus, for the normal, equal-covariance case, the optimal decision rule is merely to decide ω_1 if Fisher's linear discriminant exceed some threshold, and to decide ω_2 otherwise.

Multiple Discriminant Analysis

- Generalize 2-class Fisher's linear discriminant to *c*-class problem
- Now, the projection is from a *d*-dimensional space to a (*c* - 1)-dimensional space, $d \geq c$

Multiple Discriminant Analysis

- Because $\mathbf{S_B}$ is the sum of c matrices of rank one or less, and because only $c-1$ of these are independent, $\mathbf{S_B}$ is of rank $c-1$ or less. Thus, no more than $c-1$ of the eigenvalues are nonzero, and so the new dimensionality is up to (c-1).

Multiple Discriminant Analysis

- The projection from a *d*-dimensional space to a *(c-1)*-dimensional space is accomplished by *c*-1 discriminant functions

$$y_i = \mathbf{w}_i^t \mathbf{x} \qquad i = 1, ..., c-1.$$

- If the y_i are viewed as components of a vector *y* and the weight vectors $\mathbf{w_i}$ are viewed as the columns of a *dx(c – 1)* matrix **W**, then the projection can be written as a single matrix equation

$$\mathbf{y} = \mathbf{W}^t \mathbf{x}.$$

The columns of an optimal **W** are the generalized eigenvectors that correspond to the largest eigenvalues in

Multiple Discriminant Analysis

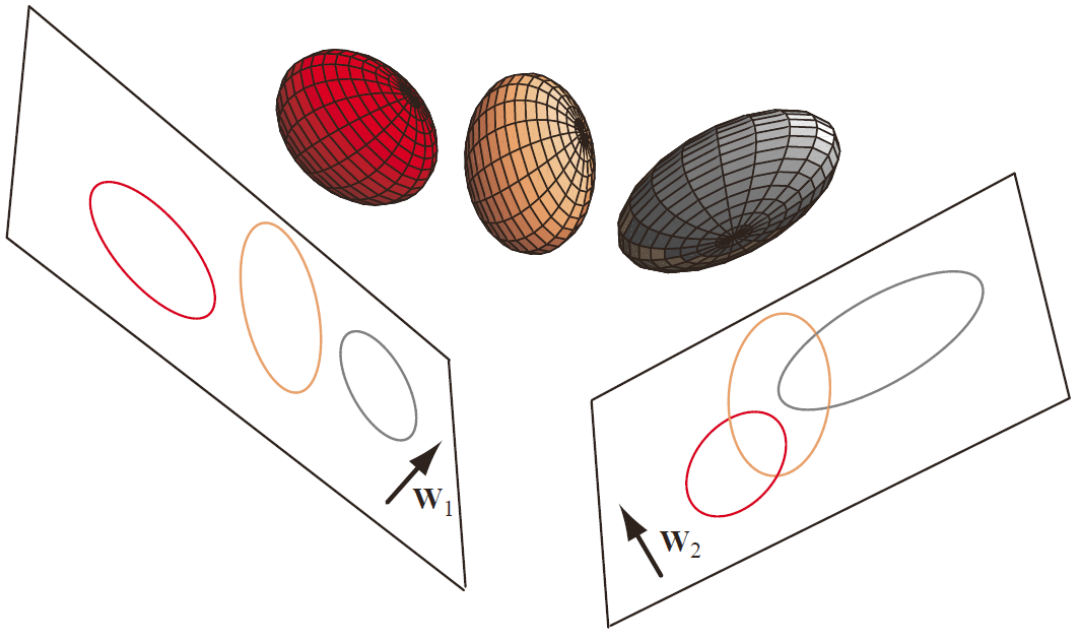
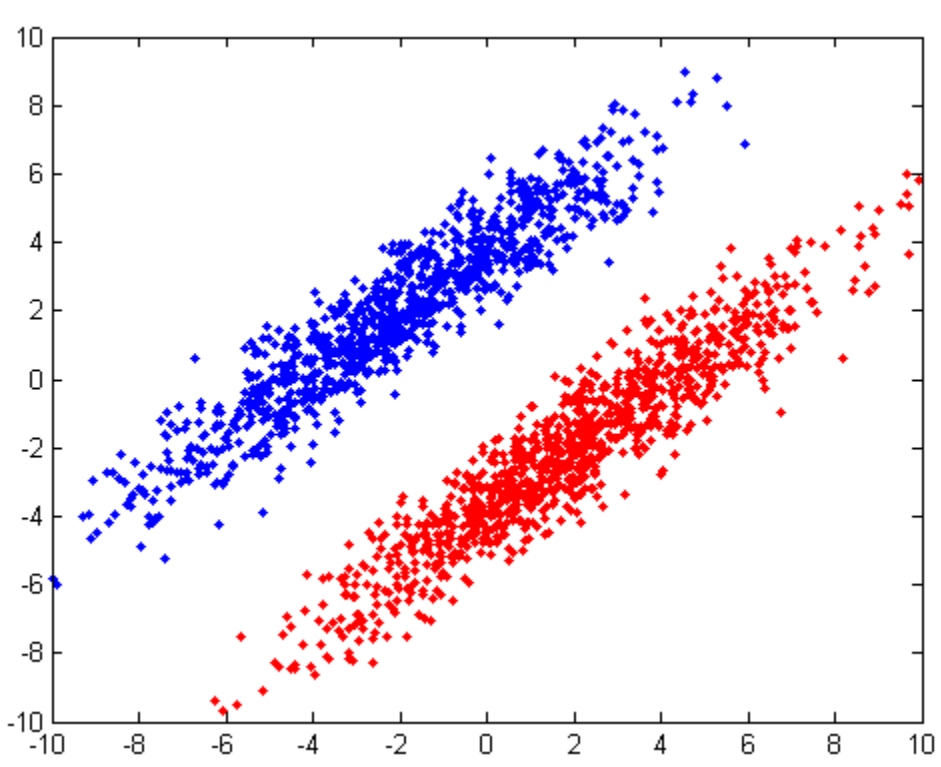
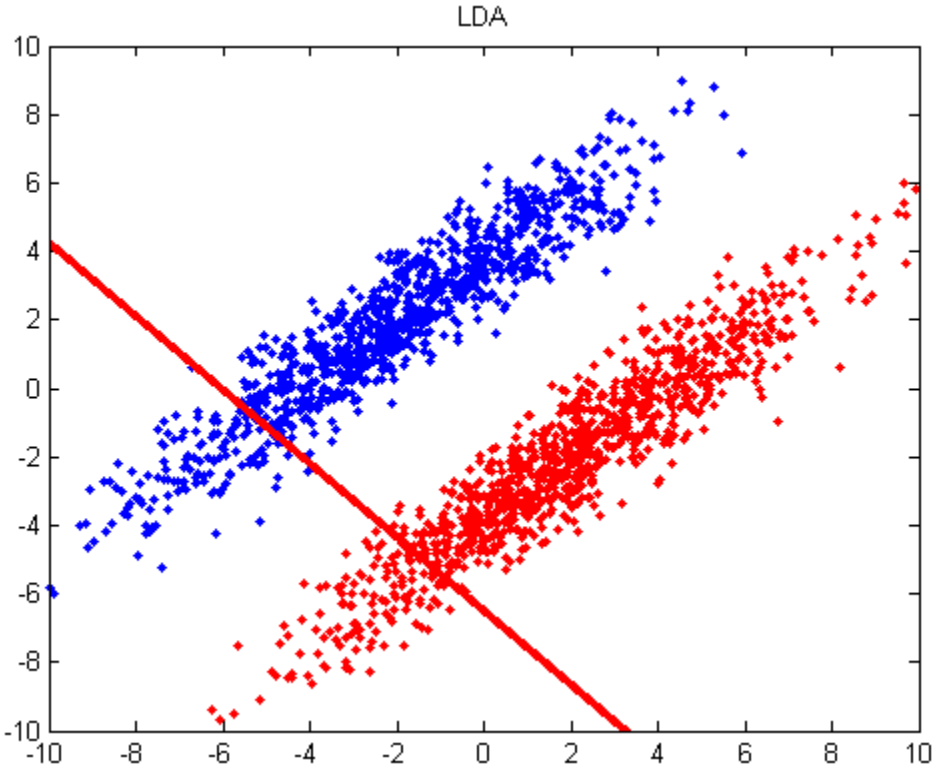


Figure 3.6: Three 3-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors $\mathbf{w_1}$ and $\mathbf{w_2}$. Informally, multiple discriminant methods seek the optimum such subspace, i.e., the one with the greatest separation of the projected distributions for a given total within-scatter matrix, here as associated with $\mathbf{w_1}$.

LDA

$$Y_1 = \mathbf{w}^T \mathbf{X_1} \qquad Y_2 = \mathbf{w}^T \mathbf{X}_2$$





Find a transformation **w**, such that the $\mathbf{w}^T\mathbf{X_1}$ and $\mathbf{w}^T\mathbf{X_2}$ are maximally separated & each class is minimally dispersed (maximum separation)

PCA vs. LDA

	PCA	LDA	
Sample mean	$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x_i}$	$\boldsymbol{\mu_i} = \frac{1}{N_i}\sum_{\mathbf{x}\in\omega_i} \mathbf{x}$	Mean for each class
Scatter matrix	$\mathbf{S} = \sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$	$\mathbf{S_w} = \sum_{i=1}^{c}\sum_{x_j \in C_i}(x_j - \mu_i)(x_j - \mu_i)^T$	Within-class scatter
		$\mathbf{S_b} = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$	Between-class scatter
Eigen decomposition	$\mathbf{Sw} = \lambda\mathbf{w}$	$\mathbf{S_W^{-1}S_B w} = \lambda\mathbf{w}$	Eigen decomposition

$$\mathbf{Y} = \mathbf{w}^T\mathbf{X}$$

X is transformed to Y using w

Principal Component Analysis (PCA)

- Example
  - X={(4,1),(2,4),(2,3),(3,6),(4,4)}
  - Statistics

$$\boldsymbol{\mu} = [3.0 \quad 3.6],$$
$$\mathbf{S} = \begin{bmatrix} 4.0 & -2.0 \\ -2.0 & 13.2 \end{bmatrix}$$

  - Solve the Eigen value problem

$$\mathbf{S}\mathbf{w} = \lambda\mathbf{w}$$

Linear Discriminant Analysis (LDA)

- Example
  - $\mathbf{X}_1$= {(4,1),(2,4),(2,3),(3,6),(4,4)}
  - $\mathbf{X}_2$= {(9,10),(6,8),(9,5),(8,7),(10,8)}
- Class statistics

$$\boldsymbol{\mu}_1 = [3.0 \quad 3.6], \quad \boldsymbol{\mu}_2 = [7.67 \quad 7.0], \quad \boldsymbol{\mu} = [5.7 \quad 5.6]$$

$$\mathbf{S}_1 = \begin{bmatrix} 4.0 & -2.0 \\ -2.0 & 13.2 \end{bmatrix}, \quad \mathbf{S}_2 = \begin{bmatrix} 11.89 & 2.0 \\ 2.0 & 15.0 \end{bmatrix}$$

- Within and between class scatter

$$\mathbf{S}_B = \begin{bmatrix} 72 & 54 \\ 54 & 40 \end{bmatrix}, \quad \mathbf{S}_W = \begin{bmatrix} 15.89 & 0.0 \\ 0.0 & 28.2 \end{bmatrix}$$

- Solve the Eigen value problem

$$\mathbf{S}_W^{-1}\mathbf{S}_B\mathbf{w} = \lambda\mathbf{w}$$

A Global Geometric Framework for Nonlinear Dimensionality Reduction

Tenenbaum, de Silva and Langford, Science, V. 290, 22 Dec 2000

- Although input dimensionality may be quite high (e.g. 4096 for 64x64 pixel images in Fig 1A), the perceptually meaningful structure has many fewer independent degrees of freedom
- The images in 1A lie on an intrinsically 3-dim manifold, or constraint surface (two pose variables & an a lighting angle)
- Given unordered high-dim inputs, discover low-dim representations
- PCA finds a linear subspace; Fig. 3A illustrates the challenge of non-linearity; points far apart on the underlying manifold, as measured by their geodesic, or shortest path, distances, may appear close in high-dim input space, as measured by their straight line Euclidean distance.

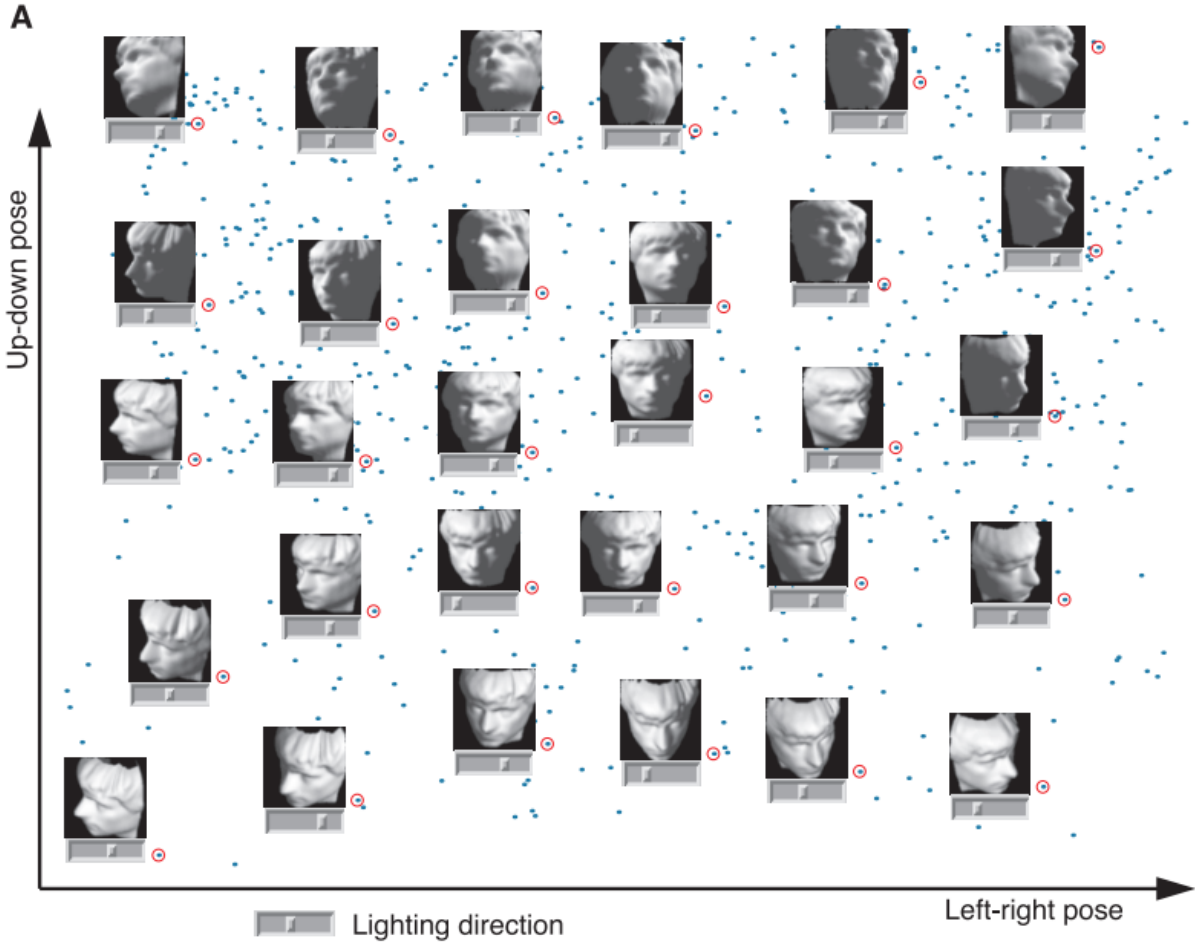
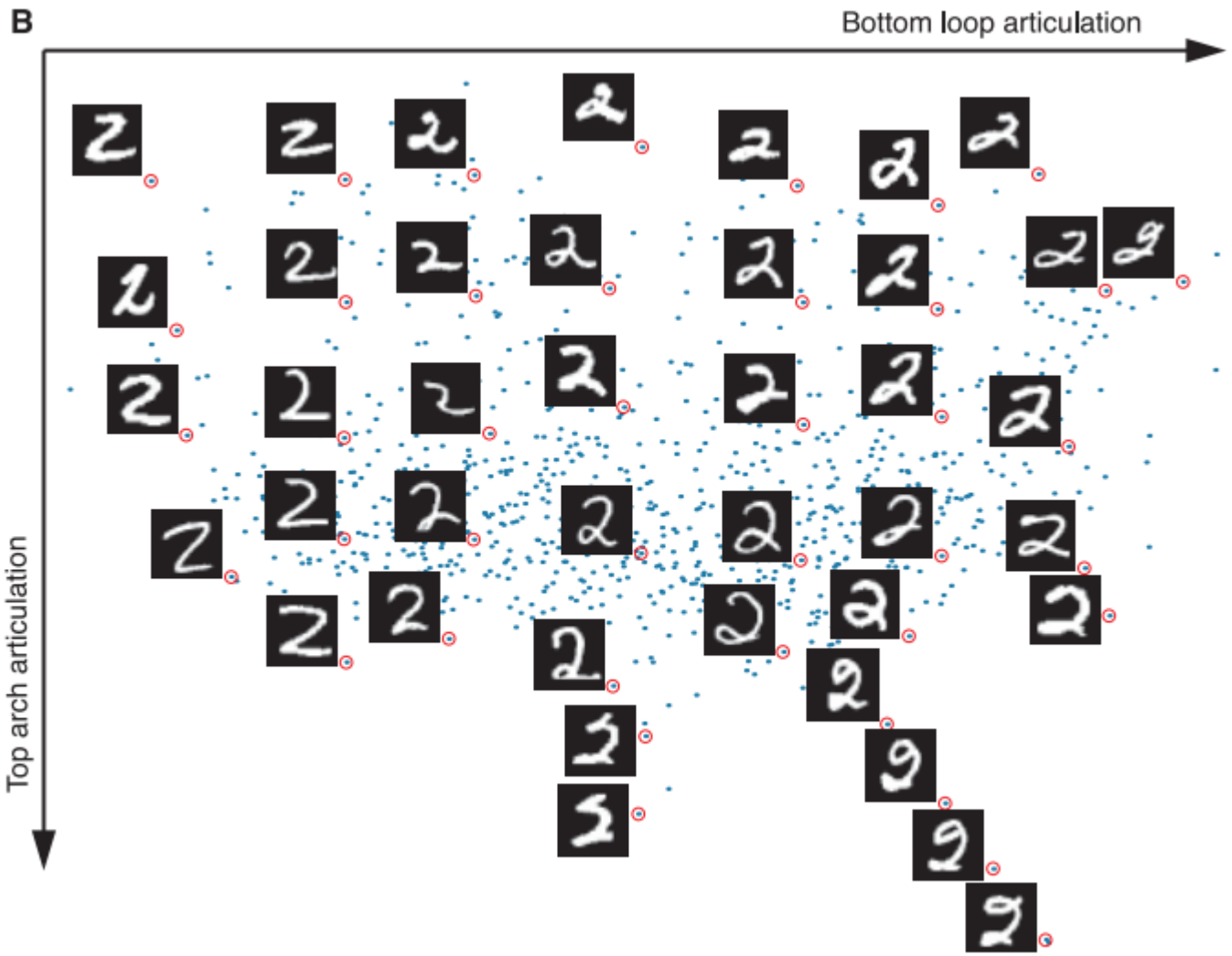


Fig. 1. (**A**) A canonical dimensionality reduction problem from visual perception. The input consists of a sequence of 4096-dimensional vectors, representing the brightness values of 64 pixel by 64 pixel images of a face rendered with different poses and lighting directions. Applied to $N = 698$ raw images, Isomap ($K = 6$) learns a three-dimensional embedding of the data's intrinsic geometric structure. A two-dimensional projection is shown, with a sample of the original input images (red circles) superimposed on all the data points (blue) and horizontal sliders (under the images) representing the third dimension. Each coordinate axis of the embedding correlates highly with one degree of freedom underlying the original data: left-right pose (x axis, $R = 0.99$), up-down pose (y axis, $R = 0.90$), and lighting direction (slider position, $R = 0.92$). The input-space distances $d_X(i,j)$ given to Isomap were Euclidean distances between the 4096-dimensional image vectors. (**B**) Isomap applied to $N = 1000$ handwritten "2"s from the MNIST database (*40*). The two most significant dimensions in the Isomap embedding, shown here, articulate the major features of the "2": bottom loop (x axis) and top arch (y axis). Input-space distances $d_X(i,j)$ were measured by tangent distance, a metric designed to capture the invariances relevant in handwriting recognition (*41*). Here we used ϵ-Isomap (with $\epsilon = 4.2$) because we did not expect a constant dimensionality to hold over the whole data set; consistent with this, Isomap finds several tendrils projecting from the higher dimensional mass of data and representing successive exaggerations of an extra stroke or ornament in the digit.


B
Bottom loop articulation
Top arch articulation

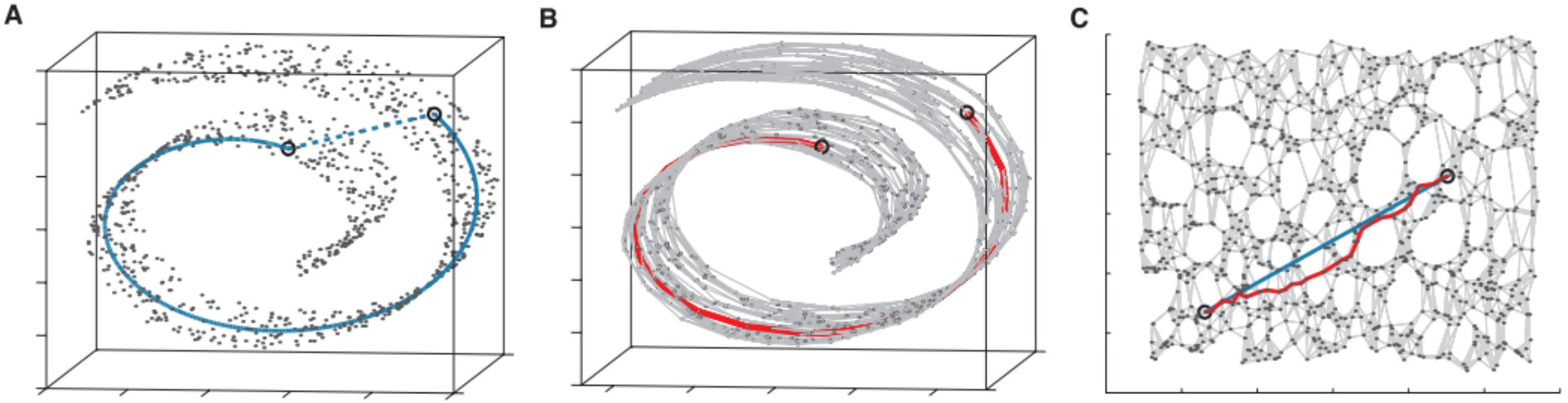


Fig. 3. The "Swiss roll" data set, illustrating how Isomap exploits geodesic paths for nonlinear dimensionality reduction. (**A**) For two arbitrary points (circled) on a nonlinear manifold, their Euclidean distance in the high-dimensional input space (length of dashed line) may not accurately reflect their intrinsic similarity, as measured by geodesic distance along the low-dimensional manifold (length of solid curve). (**B**) The neighborhood graph G constructed in step one of Isomap (with $K = 7$ and $N = 1000$ data points) allows an approximation (red segments) to the true geodesic path to be computed efficiently in step two, as the shortest path in G. (**C**) The two-dimensional embedding recovered by Isomap in step three, which best preserves the shortest path distances in the neighborhood graph (overlaid). Straight lines in the embedding (blue) now represent simpler and cleaner approximations to the true geodesic paths than do the corresponding graph paths (red).

Table 1. The Isomap algorithm takes as input the distances $d_X(i,j)$ between all pairs i,j from N data points in the high-dimensional input space X, measured either in the standard Euclidean metric (as in Fig. 1A) or in some domain-specific metric (as in Fig. 1B). The algorithm outputs coordinate vectors $\mathbf{y}_i$ in a d-dimensional Euclidean space Y that (according to Eq. 1) best represent the intrinsic geometry of the data. The only free parameter (ϵ or K) appears in Step 1.

Step		
1	Construct neighborhood graph	Define the graph G over all data points by connecting points i and j if [as measured by $d_X(i,j)$] they are closer than ϵ (ϵ-Isomap), or if i is one of the K nearest neighbors of j (K-Isomap). Set edge lengths equal to $d_X(i,j)$.
2	Compute shortest paths	Initialize $d_G(i,j) = d_X(i,j)$ if i,j are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G = \{d_G(i,j)\}$ will contain the shortest path distances between all pairs of points in G (*16*, *19*).
3	Construct d-dimensional embedding	Let λ_p be the p-th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (*17*), and v_p^i be the i-th component of the p-th eigenvector. Then set the p-th component of the d-dimensional coordinate vector $\mathbf{y}_i$ equal to $\sqrt{\lambda_p} v_p^i$.

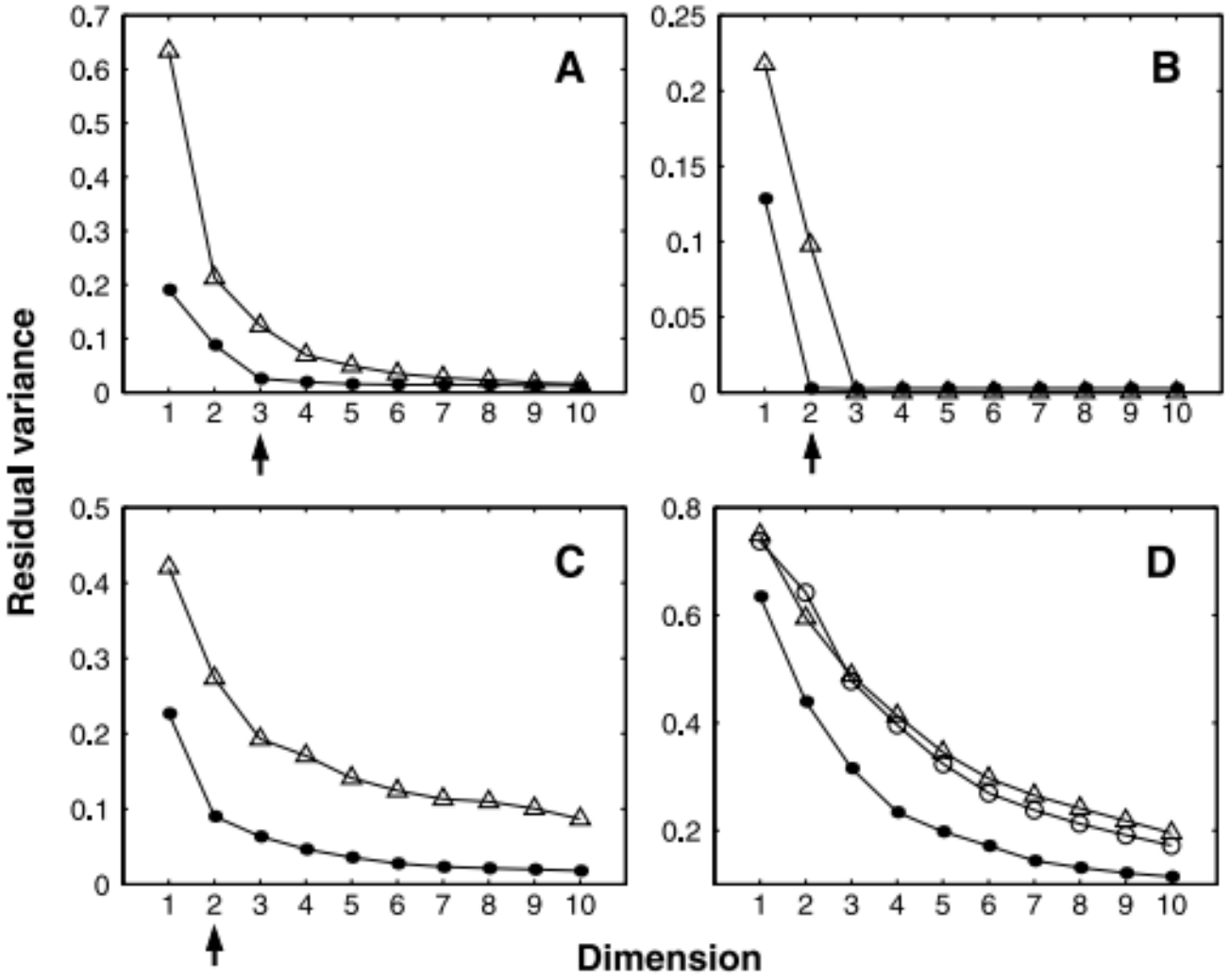
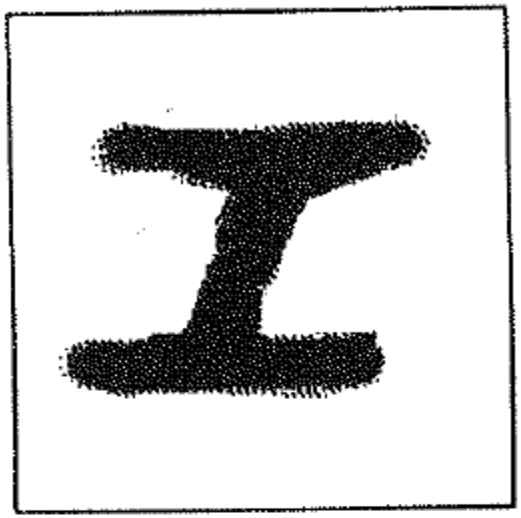
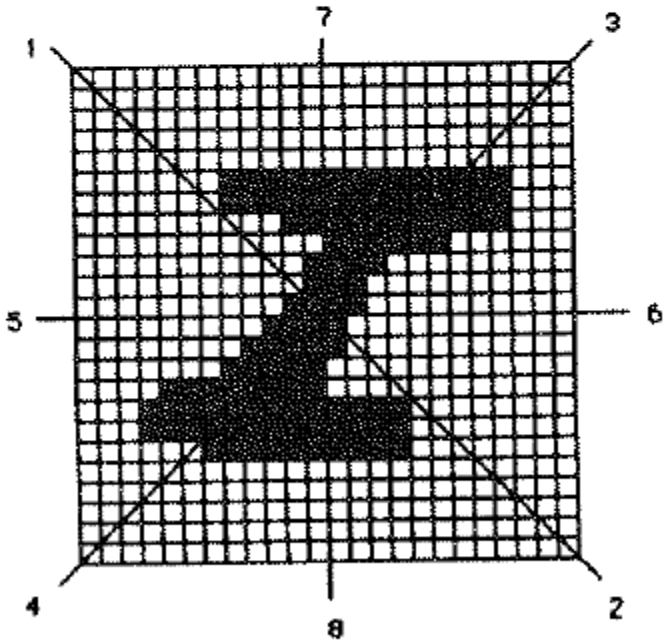


Fig. 2. The residual variance of PCA (open triangles), MDS [open triangles in (A) through (C); open circles in (D)], and Isomap (filled circles) on four data sets (*42*). (**A**) Face images varying in pose and illumination (Fig. 1A). (**B**) Swiss roll data (Fig. 3). (**C**) Hand images varying in finger extension and wrist rotation (*20*). (**D**) Handwritten "2"s (Fig. 1B). In all cases, residual variance decreases as the dimensionality *d* is increased. The intrinsic dimensionality of the data can be estimated by looking for the "elbow" at which this curve ceases to decrease significantly with added dimensions. Arrows mark the true or approximate dimensionality, when known. Note the tendency of PCA and MDS to overestimate the dimensionality, in contrast to Isomap.





Pattern: (11, 11, 5, 6, 10, 10, 5, 5)

Figure 2.1 Binary representation of a handwritten character.

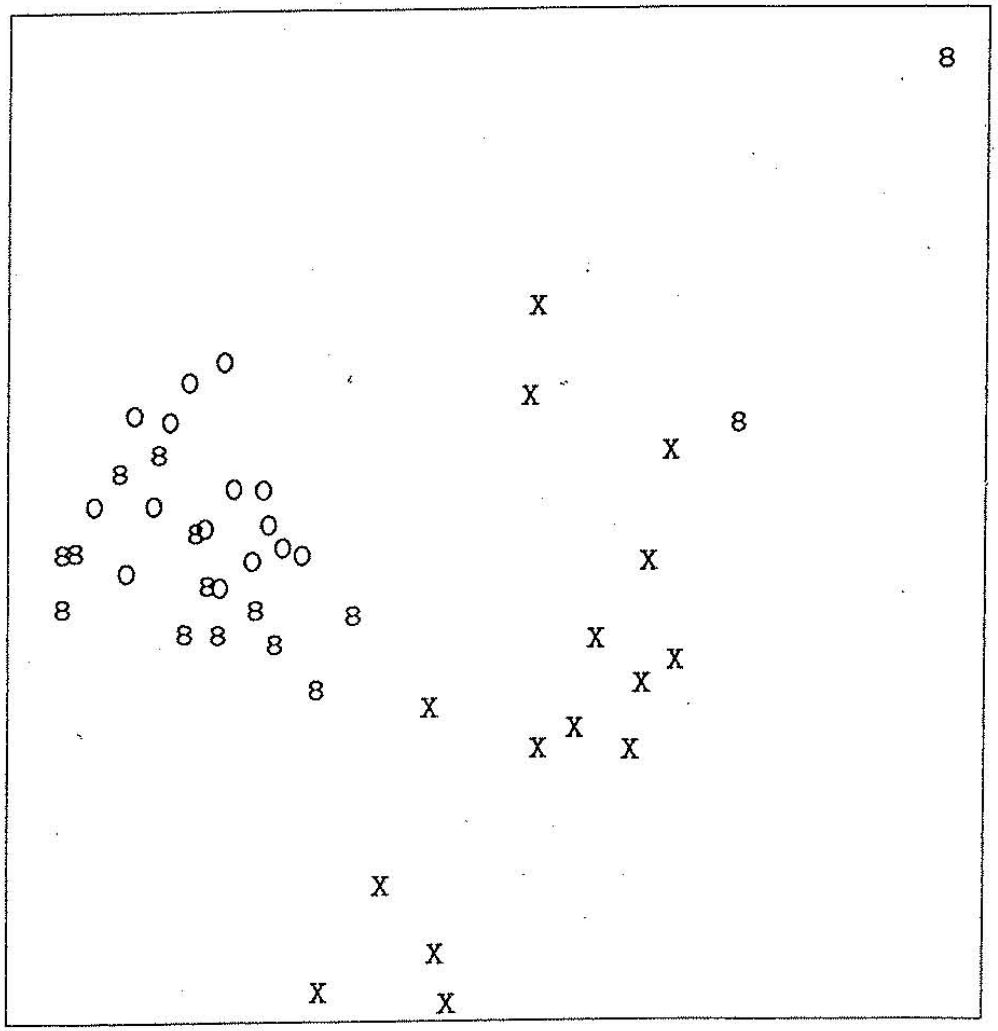


Figure 2.6 Two-dimensional representation of 80X data on the first two principal components.

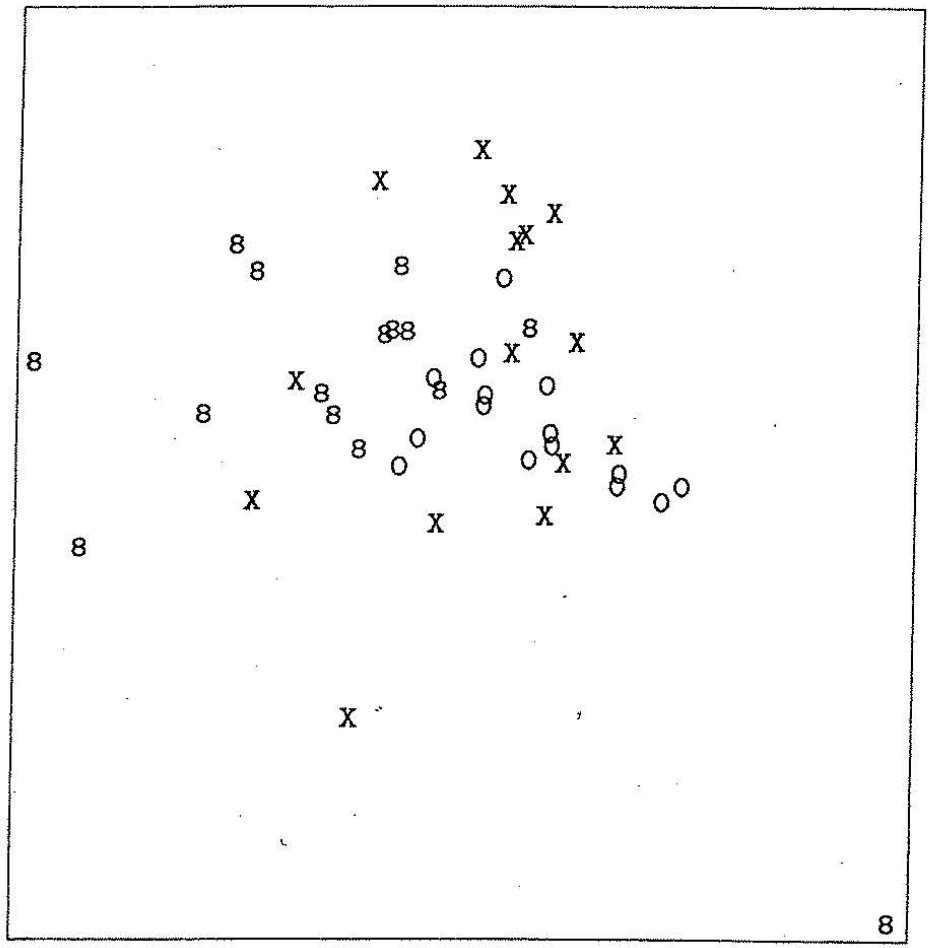


Figure 2.7 Two-dimensional projection of 80X data on the third and fourth principal components.

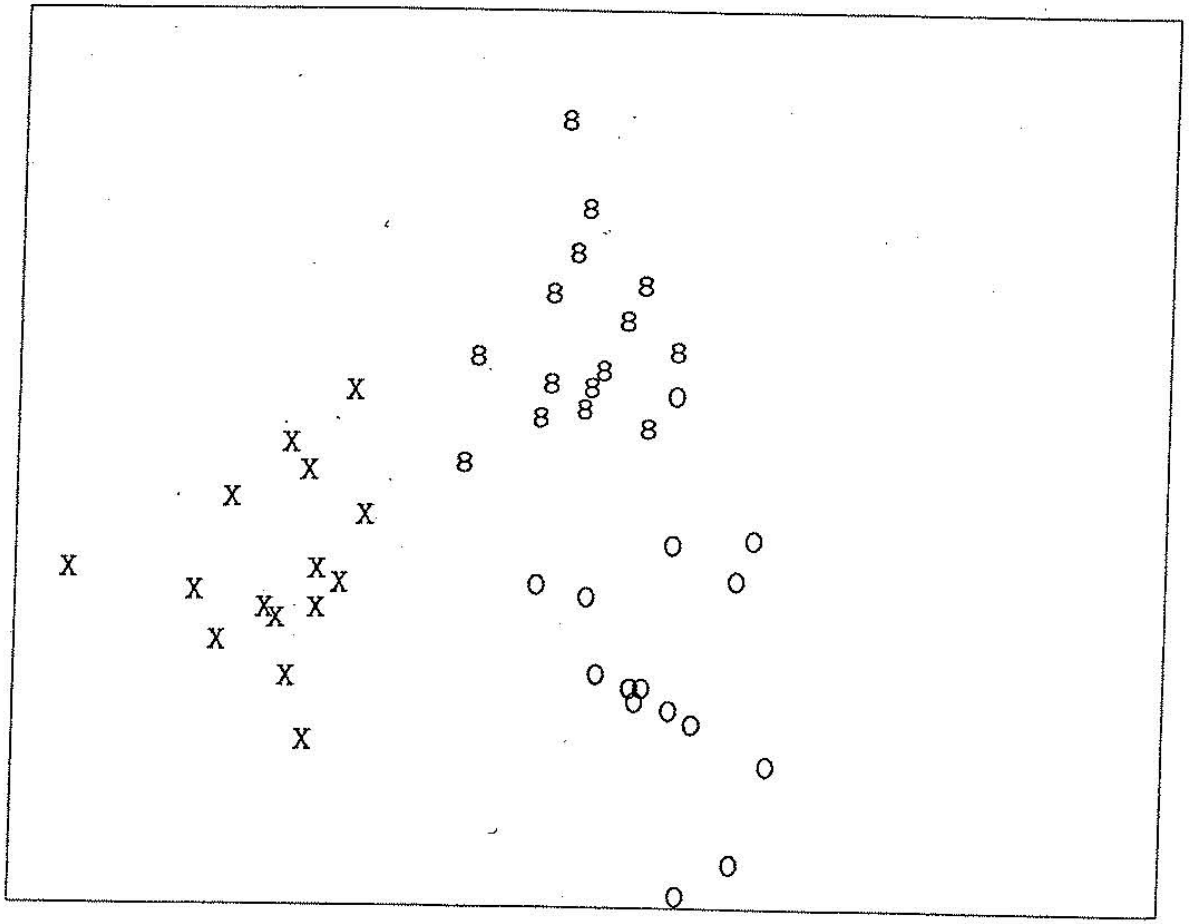


Figure 2.12 Two-dimensional projection of 80X data using discriminant analysis.

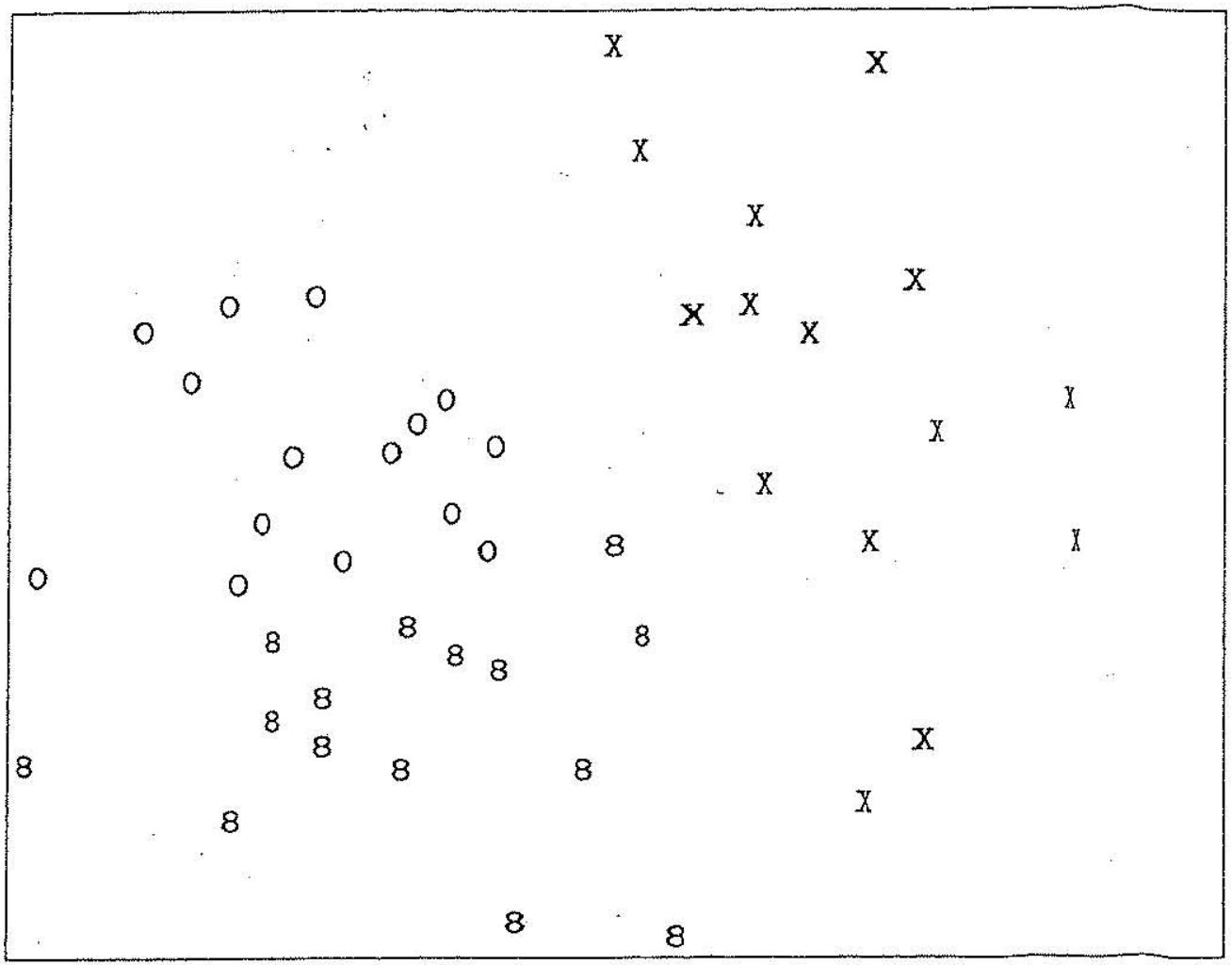


Figure 2.13 Projection of 8OX data by Sammon's method

Low Dimensional Representations And Multidimensional Scaling (MDS) (Sec 10.14)

- Given *n* points (objects) $\mathbf{x}_1, \ldots, \mathbf{x}_n$. No class labels
- Suppose only the similarities between the n objects are provided
- Goal is to represent these n objects in some low dimensional space in such a way that the distances between points in that space corresponds to the dissimilarities in the original space
- If an accurate representation can be found in 2 or 3 dimensions than we can visualize the structure of the data
- Find a configuration of points $\mathbf{y}_1, \ldots, \mathbf{y}_n$ for which the *n(n-1)* distances d_{ij} are as close as possible to the original similarities; this is called *Multidimensional scaling*
- Two cases
  - Meaningful to talk about the distances between given n

Distances Between Given Points is Meaningful

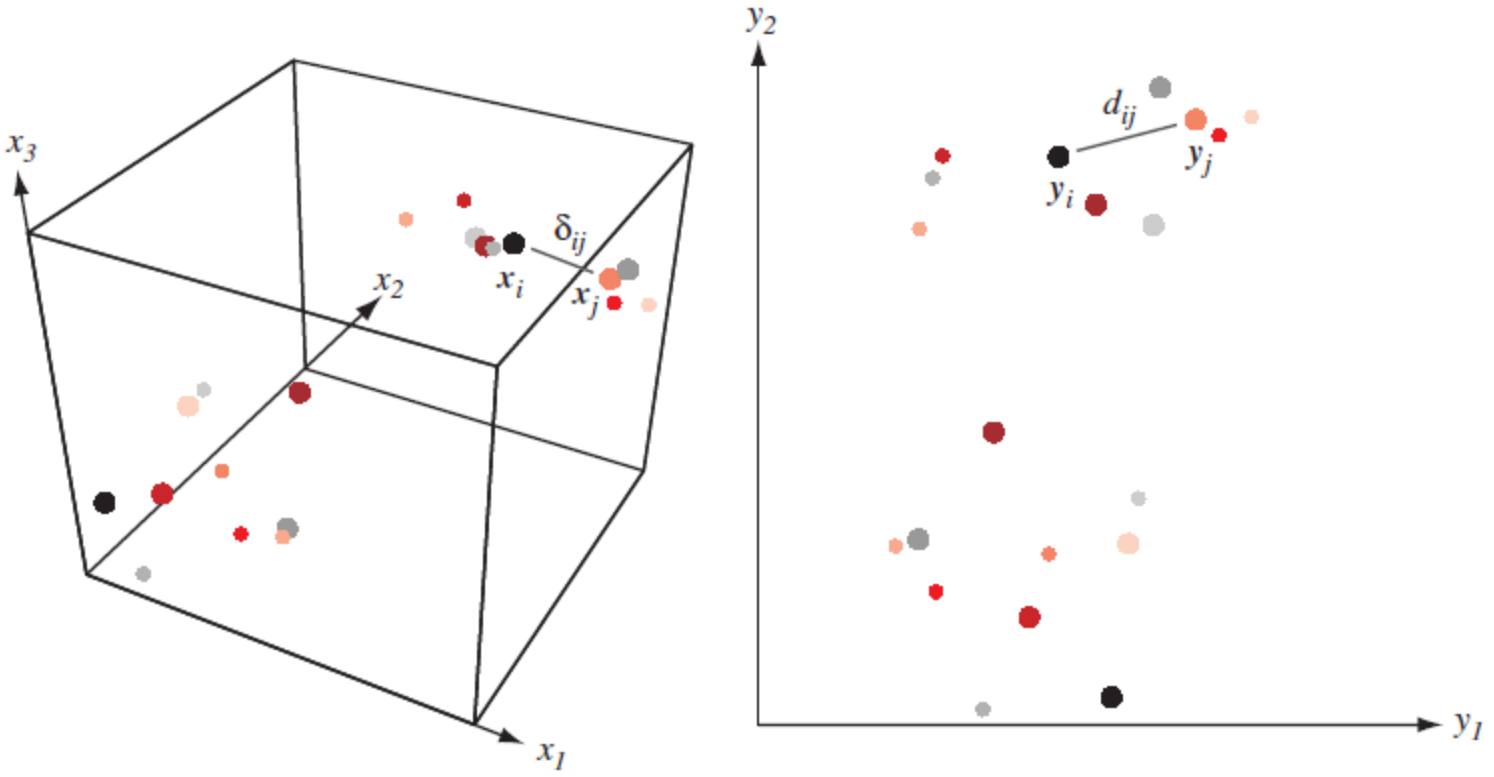


Figure 10.26: The distance between points in the original space are δ_{ij} while in the projected space d_{ij}. In practice, the source space is typically of very high dimension, and the mapped space of just two or three dimensions, to aid visualization. (In order to illustrate the correspondence between points in the two spaces, the size and color of each point $\mathbf{x}_i$ matches that of its image $\mathbf{y}_i$.

Criterion Functions

- Sum of squared error functions
- Since they only involve distances between points, they are invariant to rigid body motions of the configuration
- Criterion functions have been normalized so their minimum values are invariant to dilations of the sample points

$$J_{ee} = \frac{\sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}$$

$$J_{ff} = \sum_{i<j} \left(\frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$$

$$J_{ef} = \frac{1}{\sum_{i<j} \delta_{ij}} \sum_{i<j} \frac{(d_{ij} - \delta_{ij})^2}{\delta_{ij}}.$$

Finding the Optimum Configuration

- Use gradient-descent procedure to find an optimal configuration $\mathbf{y}_1, \ldots, \mathbf{y}_n$

$$\nabla_{\mathbf{y}_k} J_{ee} = \frac{2}{\sum_{i<j} \delta_{ij}^2} \sum_{j \neq k} (d_{kj} - \delta_{kj}) \frac{\mathbf{y}_k - \mathbf{y}_j}{d_{kj}}$$

$$\nabla_{\mathbf{y}_k} J_{ff} = 2 \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}^2} \frac{\mathbf{y}_k - \mathbf{y}_j}{d_{kj}}$$

$$\nabla_{\mathbf{y}_k} J_{ef} = \frac{2}{\sum_{i<j} \delta_{ij}} \sum_{j \neq k} \frac{d_{kj} - \delta_{kj}}{\delta_{kj}} \frac{\mathbf{y}_k - \mathbf{y}_j}{d_{kj}}.$$

Example

$$
\begin{aligned}
x_1(k) &= \cos(k/\sqrt{2}) \\
x_2(k) &= \sin(k/\sqrt{2}) \\
x_3(k) &= k/\sqrt{2}, \qquad k = 0, 1, \ldots, 29.
\end{aligned}
$$

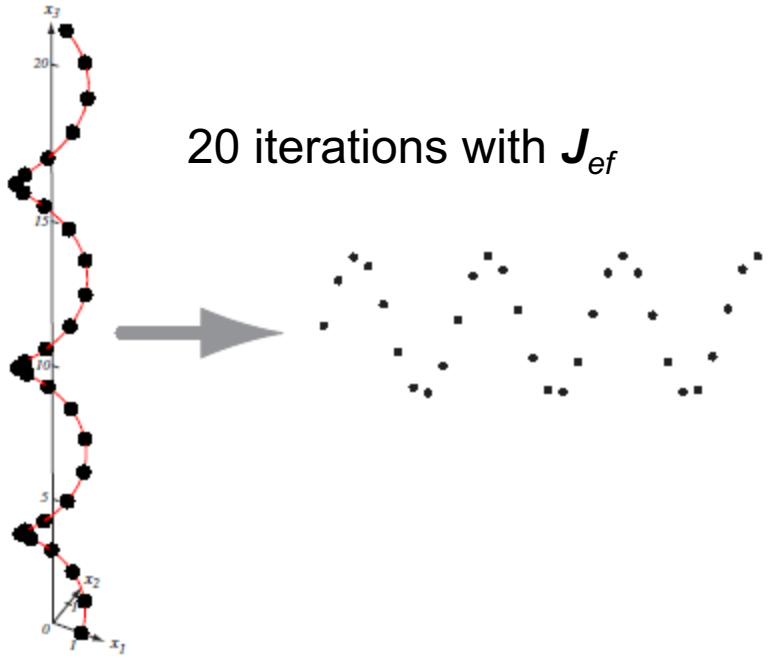


Figure 10.27: Thirty points of the form $(\cos(k/\sqrt{2}), \sin(k/\sqrt{2}), k/\sqrt{2})^t$ for $k = 0, 1, \ldots, 29$ are shown at the left. Multidimensional scaling using the J_{ef} criterion (Eq. 105) and a two-dimensional target space leads to the image points shown at the right. This lower-dimensional representation shows clearly the fundamental sequential nature of the points in the original, source space.

Nonmetric Multidimensional Scaling

- Numerical values of dissimilarities are not as important as their rank order
- Monotonicity constraint: rank order of d_{ij} = rank order of δ_{ij}

$$\hat{d}_{i_1 j_1} \leq \hat{d}_{i_2 j_2} \leq \cdots \leq \hat{d}_{i_m j_m}.$$

- The degree to which d_{ij} satisfy the monotonicy constraint is measured by

$$\hat{J}_{mon} = \min_{\hat{d}_{ij}} \sum_{i<j} (d_{ij} - \hat{d}_{ij})^2,$$

- Normalize $\hat{J}_{mon}$ to prevent it from being collapsed

$$J_{mon} = \frac{\hat{J}_{mon}}{\sum_{i<j} d_{ij}^2}.$$

Overfitting

Problem of Insufficient Data

- How to train a classifier (e.g., estimate the covariance matrix) when the training set size is small (compared to the number of features)
- Reduce the dimensionality
  - Select a subset of features
  - Combine available features to get a smaller number of more "salient" features.
- Bayesian techniques
  - Assume a reasonable prior on the parameters to compensate for small amount of training data
- Model Simplification
  - Assume statistical independence
- Heuristics
  - Threshold the estimated covariance matrix such that only correlations above a threshold are retained.

Practical Observations

- Most heuristics and model simplifications are almost surely incorrect
- In practice, however, the performance of the classifiers base don model simplification is better than with full parameter estimation
- Paradox: How can a suboptimal/simplified model perform better than the MLE of full parameter set, on test dataset?
  - The answer involves the problem of insufficient data

Insufficient Data in Curve Fitting

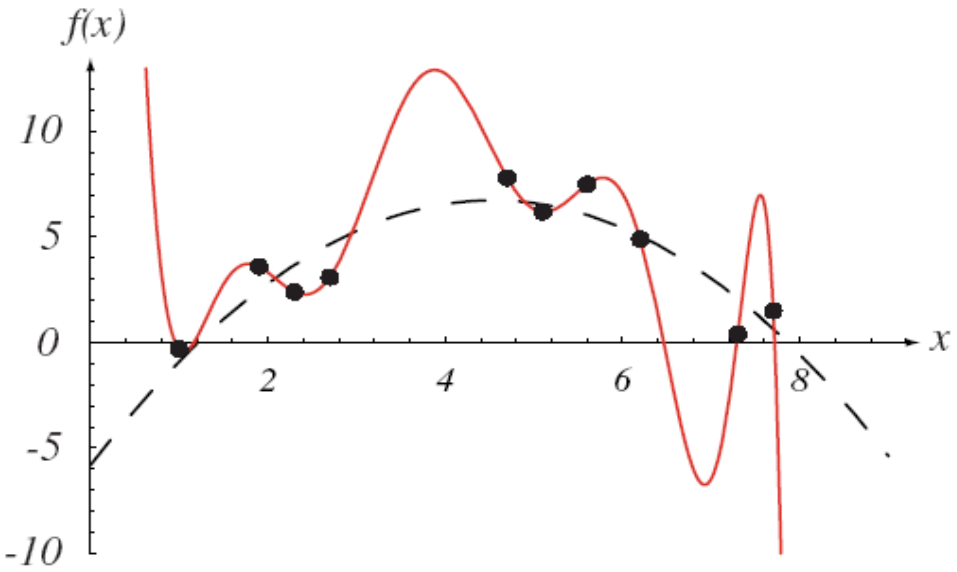


FIGURE 3.4. The "training data" (black dots) were selected from a quadratic function plus Gaussian noise, i.e., $f(x) = ax^2+bx+c+\epsilon$ where $p(\epsilon) \sim N(0, \sigma^2)$. The 10th-degree polynomial shown fits the data perfectly, but we desire instead the second-order function $f(x)$, because 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.

Curve Fitting Example (contd)

- The example shows that a 10^{th}-degree polynomial fits the training data with zero error
  - However, the test or the generalization error is much higher for this fitted curve
- When the data size is small, one cannot be sure about how complex the model should be
- A small change in the data will change the parameters of the 10^{th}-degree polynomial significantly, which is not a desirable quality; stability

Handling insufficient data

- Heuristics and model simplifications
- Shrinkage is an intermediate approach, which combines "common covariance" with individual covariance matrices
  - Individual covariance matrices shrink towards a common covariance matrix.
  - Also called regularized discriminant analysis
- Shrinkage Estimator for a covariance matrix, given shrinkage factor $0 < \alpha < 1$,

$$\Sigma_i(\alpha) = \frac{(1-\alpha)n_i\Sigma_i + \alpha n\Sigma}{(1-\alpha)n_i + \alpha n}$$

- Further, the common covariance can be shrunk towards the Identity matrix,

$$\Sigma(\beta) = (1-\beta)\Sigma + \beta\mathbf{I}$$

Principle of Parsimony

- By allowing the covariance matrices of Gaussian conditional densities to be arbitrary, the no. of parameters in the resulting quadratic discriminant analysis to be estimated for large d or C can be rather large
- In such situations, LDF is often preferred with the principle of parsimony as the main underlying thought
- Dempster (1972) suggested that parameters should be introduced sparingly and only when data indicate they are required.

A.P. Dempster (1972), Covariance selection, Biometrics 28, 157-175

Problems of Dimensionality

Introduction

- Real world applications usually come with a large number of features
  - Text in documents is represented using frequencies of tens of thousands of words
  - Images are often represented by extracting local features from a large number of regions within an image
- Naive intuition: more the number of features, the better the classification performance? – Not always!
- There are two issues that must be confronted with high dimensional feature spaces
  - How does the classification accuracy depend on the dimensionality and the number of training samples?
  - What is the computational complexity of the classifier?

Statistically Independent Features

- If features are statistically independent, it is possible to get excellent performance as dimensionality increases
- For a two class problem with multivariate normal classes $P(x|\omega_j) \sim N(\boldsymbol{\mu}_j, \Sigma)$, and equal prior probabilities, the probability of error is

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

where the Mahalanobis distance is defined as

$$r^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$

Statistically Independent Features

- When features are independent, the covariance matrix is diagonal, and we have

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

- Since r^2 increases monotonically with an increase in the number of features, *P(e)* decreases
- As long as the means of features in the differ, the error decreases

Increasing Dimensionality

- If a given set of features does not result in good classification performance, it is natural to add more features
- High dimensionality results in increased cost and complexity for both feature extraction and classification
- If the probabilistic structure of the problem is completely known, adding new features will not possibly increase the Bayes risk

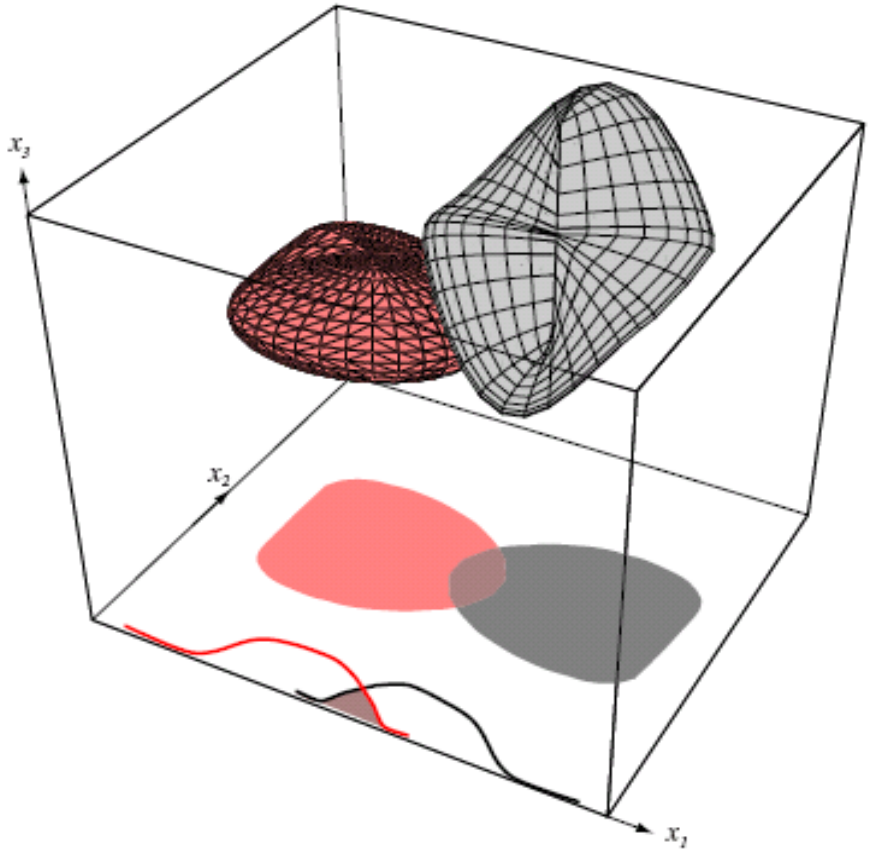


FIGURE 3.3. Two three-dimensional distributions have nonoverlapping densities, and thus in three dimensions the Bayes error vanishes. When projected to a subspace—here, the two-dimensional $x_1 - x_2$ subspace or a one-dimensional x_1 subspace—there can be greater overlap of the projected distributions,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

Curse of Dimensionality

- In practice, increasing dimensionality beyond a certain point in the presence of finite number of training samples often leads to lower performance, rather than better performance
- The main reasons for this paradox are as follows:
  - the Gaussian assumption, that is typically made, is almost surely incorrect
  - Training sample size is always finite, so the estimation of the class conditional density is not very accurate
- Analysis of this "curse of dimensionality" problem is difficult

A Simple Example

- Trunk (PAMI, 1979) provided a simple example illustrating this phenomenon.

$$p(\omega_1) = p(\omega_2) = \frac{1}{2} \qquad \boldsymbol{\mu}_1 = \boldsymbol{\mu}, \boldsymbol{\mu}_2 = -\boldsymbol{\mu}$$

$$p(X \mid \omega_1) \sim G(\boldsymbol{\mu}_1, \mathbf{I})$$

$$p(X \mid \omega_2) \sim G(\boldsymbol{\mu}_2, \mathbf{I})$$

$$p(\mathbf{x} \mid \omega_1) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(x_i - \frac{1}{\sqrt{i}}\right)^2}$$

$$p(\mathbf{x} \mid \omega_2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(x_i - \frac{1}{\sqrt{i}}\right)^2}$$

N: Number of features

$$\mu_i = \sqrt{\left(\frac{1}{i}\right)} = i^{th} \text{ component of the mean vector}$$

$$\boldsymbol{\mu}_1 = \left(1, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{4}}, \ldots\right) \qquad \boldsymbol{\mu}_2 = \left(-1, -\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{4}}, \ldots\right)$$

Case 1: Mean Values Known

- Bayes decision rule:

$$\text{Decide } \omega_1 \text{ if } \mathbf{x}^t\boldsymbol{\mu} = x_1\mu_1 + x_2\mu_2 + \ldots + x_N\mu_N > 0$$

$$\text{or} \qquad \sum_{i=1}^{N} \frac{x_i}{\sqrt{i}} > 0$$

$$P_e = \int_{\gamma/2}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}\, dz \qquad \gamma^2 = \left\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\right\|^2 = 4\sum_{i=1}^{N}\left(\frac{1}{i}\right)$$

$$P_e(N) = \int_{\sqrt{\sum_{i=1}^{N}\left(\frac{1}{i}\right)}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}\, dz$$

$\sum\left(\frac{1}{i}\right)$ is a divergent series $\quad \therefore P_e(N) \to 0 \quad$ as $\quad N \to \infty$

Case 2: Mean Values Unknown

- m labeled training samples are available

$$\hat{\boldsymbol{\mu}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}^i, \left\{ \mathbf{x}^i \text{ is replaced by } -\mathbf{x}^i \text{ if } \mathbf{x}^i \in \omega_2 \right.$$

POOLED ESTIMATE

Plug-in decision rule

$$\text{Decide } \omega_1 \text{ if } \mathbf{x}^t \hat{\boldsymbol{\mu}} = x_1\hat{\mu}_1 + x_2\hat{\mu}_2 + \ldots + x_N\hat{\mu}_N > 0$$

$$P_e(N,m) = P(\omega_2).P\left(\mathbf{x}^t\hat{\boldsymbol{\mu}} \geq 0 \mid \mathbf{x} \in \omega_2\right) + P(\omega_1).P\left(\mathbf{x}^t\hat{\boldsymbol{\mu}} \geq 0 \mid \mathbf{x} \in \omega_1\right)$$

$$= P\left(\mathbf{x}^t\hat{\boldsymbol{\mu}} \geq 0 \mid \mathbf{x} \in \omega_2\right) \left\{ \text{due to symmetry} \right.$$

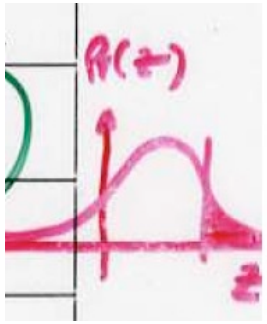
$$\text{Let } \mathbf{z} = \mathbf{x}^t\boldsymbol{\mu} = \sum_{i=1}^{N} x_i\hat{\mu}_i$$

It is difficult to computer the distribution of $\mathbf{z}$

Case 2: Mean Values Unknown

$$E(\mathbf{z}) = \sum_{i=1}^{N}\left(\frac{1}{i}\right) \qquad VAR(\mathbf{z}) = \left(1+\frac{1}{m}\right)\sum_{i=1}^{N}\left(\frac{1}{i}\right)+\frac{N}{m}$$

$$\lim_{N\to\infty}\frac{\mathbf{z}-E(\mathbf{z})}{\sqrt{VAR(\mathbf{z})}} \sim G(0,1), \{\text{Standard Normal}$$



$$P_e(N,m) = P(z \geq 0 \mid \mathbf{x} \in \omega_2) = P\left(\frac{z-E(z)}{\sqrt{VAR(z)}} \geq \frac{-E(z)}{\sqrt{VAR(z)}}\right)$$

$$P_e(m,N) = \int_{\gamma/2}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}z^2}\,dz$$

$$\gamma_N = -\frac{E(z)}{\sqrt{VAR(z)}} = \frac{\sum_{i=1}^{N}\left(\frac{1}{c}\right)}{\sqrt{\left(1+\frac{1}{m}\right)\sum_{i=1}^{N}\left(\frac{1}{i}\right)+\frac{N}{m}}}$$

$$\lim_{N\to\infty} P_N = 0$$

$$\therefore \lim_{N\to\infty} P_e(m,N) = \frac{1}{2}$$

Case 2: Mean Values Unknown

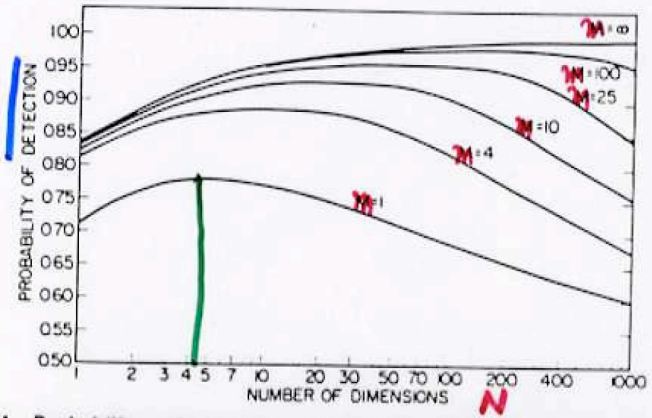
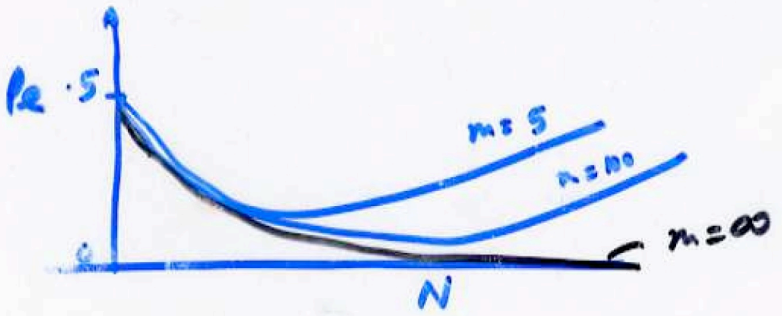


Fig. 1. Probability of detection versus dimensionality for a various number of design samples m

$m \geq 10 \cdot N$

rule of "thumb"

$1 - P_e(N, m)$



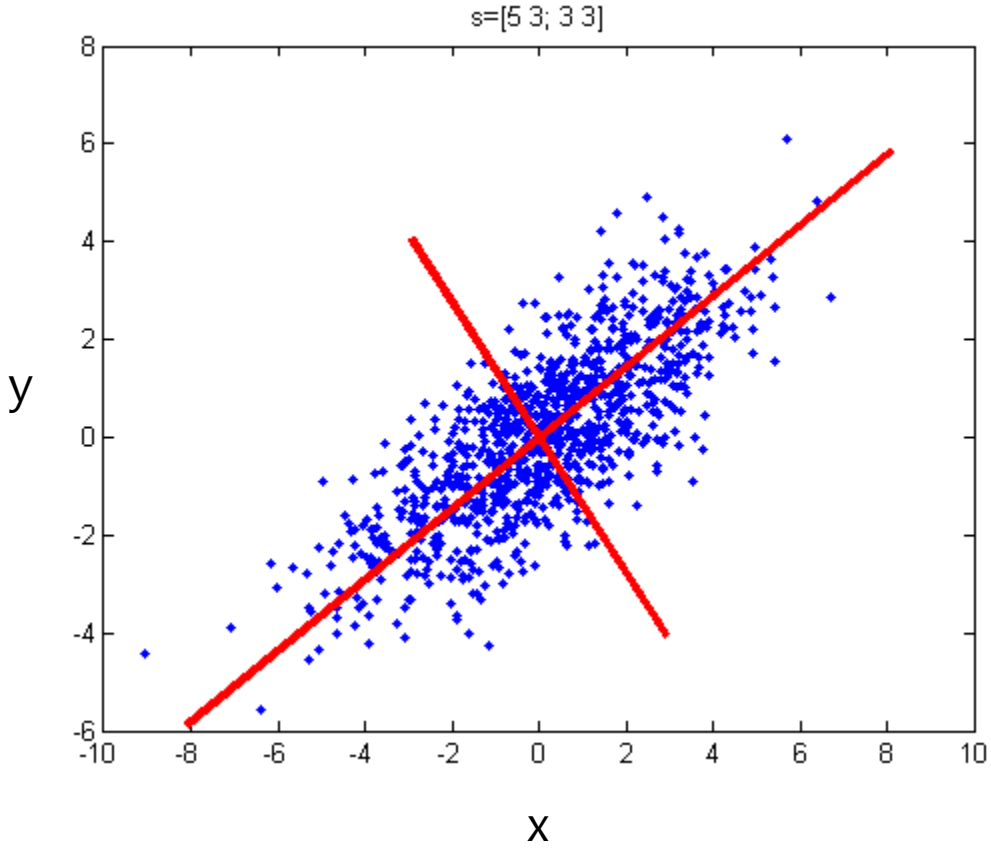
- Component Analysis and Discriminants
  - Combine features to increase discriminability & reduce dimensionality
  - Project d-dim. data to m dimensions, m<<d
  - Linear combinations are simple & tractable
  - Two approaches for linear transformation
    - PCA (Principal Component Analysis) "Projection that best represents the data in a least-square sense"; also called K-L

Diagonalization of Covariance Matrix

- Find a basis for which the components of a random vector X are uncorrelated
- It can be shown that the eignevectors of the covariance matrix for X form such a basis
- Covariance matrices (d x d) are positive semidefinite, so there exist d linearly independent eignevectors that form a basis for X
- If K is the covariance matrix, an eignevector e and an eignevalue a satisfy

$$Ke = ae$$

$$(K-aI)e = 0$$



$\boldsymbol{\mu} = [2, 1]$

$\boldsymbol{\Sigma} = \begin{bmatrix} 5 & 3 \\ 3 & 3 \end{bmatrix}$

Eigenvectors: $\begin{bmatrix} 0.5863 & -0.8101 \\ -0.8101 & -0.5863 \end{bmatrix}$

Eigenvalues: $\begin{bmatrix} 0.8344 & 0 \\ 0 & 6.9753 \end{bmatrix}$

Principal Component Analysis

This can be easily verified by writing

$$
\begin{aligned}
J_0(\mathbf{x_0}) &= \sum_{k=1}^{n} \mathsf{P}(\mathbf{x_0}-\mathbf{m})-(\mathbf{x_k}-\mathbf{m})\,\mathsf{P}^2 \\
&= \sum_{k=1}^{n} \mathsf{P}(\mathbf{x_0}-\mathbf{m})\,\mathsf{P}^2 - 2\sum_{k=1}^{n}(\mathbf{x_0}-\mathbf{m})^t(\mathbf{x_k}-\mathbf{m}) + \sum_{k=1}^{n} \mathsf{P}(\mathbf{x_k}-\mathbf{m})\,\mathsf{P}^2 \\
&= \sum_{k=1}^{n} \mathsf{P}(\mathbf{x_0}-\mathbf{m})\,\mathsf{P}^2 - 2(\mathbf{x_0}-\mathbf{m})^t\sum_{k=1}^{n}(\mathbf{x_k}-\mathbf{m}) + \sum_{k=1}^{n} \mathsf{P}(\mathbf{x_k}-\mathbf{m})\,\mathsf{P}^2 \\
&= \sum_{k=1}^{n} \mathsf{P}(\mathbf{x_0}-\mathbf{m})\,\mathsf{P}^2 + \underbrace{\sum_{k=1}^{n} \mathsf{P}(\mathbf{x_k}-\mathbf{m})\,\mathsf{P}^2}_{\text{Independent of } \mathbf{x_0}}.
\end{aligned}
$$

Since the second sum is independent of $\mathbf{x_0}$, this expression is minimized by the choice $\mathbf{x_0}=\mathbf{m}$.

Principal Component Analysis

- The scatter matrix is merely (n-1) times the sample covariance matrix. It arises here when we substitute a_k found in Eq. (83) into Eq. (82) to obtain

$$
\begin{aligned}
J_1(\mathbf{e}) &= \sum_{k=1}^{n} a_k^2 - 2\sum_{k=1}^{n} a_k^2 + \sum_{k=1}^{n} \| \mathbf{x_k} - \mathbf{m} \|^2 \\
&= -\sum_{k=1}^{n} [\mathbf{e}^t (\mathbf{x_k} - \mathbf{m})]^2 + \sum_{k=1}^{n} \| \mathbf{x_k} - \mathbf{m} \|^2 \\
&= -\sum_{k=1}^{n} \mathbf{e}^t (\mathbf{x_k} - \mathbf{m})(\mathbf{x_k} - \mathbf{m})^t \mathbf{e} + \sum_{k=1}^{n} \| \mathbf{x_k} - \mathbf{m} \|^2 \\
&= -\mathbf{e}^t \mathbf{S} \mathbf{e} + \sum_{k=1}^{n} \| \mathbf{x_k} - \mathbf{m} \|^2 .
\end{aligned}
$$

Principal Component Analysis

- The vector e that minimizes J_1 also maximizes $\mathbf{e}^t\mathbf{S}\mathbf{e}$. We use the method of Lagrange multipliers (Section A.3 of the Appendix) to maximize $\mathbf{e}^t\mathbf{S}\mathbf{e}$ subject to the constraint that $\|\mathbf{e}\| = 1$. Letting λ be the undetermined multiplier, we differentiate

$$u = \mathbf{e}^t\mathbf{S}\mathbf{e} - \lambda(\mathbf{e}^t\mathbf{e} - 1)$$

  with respect to **e** to obtain

$$\frac{\partial u}{\partial \mathbf{e}} = 2\mathbf{S}\mathbf{e} - 2\lambda\mathbf{e}.$$

- Setting this gradient vector equal to zero, **e** is the eigenvector of the scatter matrix:

$$\mathbf{S}\mathbf{e} = \lambda\mathbf{e}.$$

Principal Component Analysis

- Since $\mathbf{e}^t\mathbf{S}\mathbf{e} = \lambda\, \mathbf{e}^t\mathbf{e} = \lambda$, it follows that to maximize $\mathbf{e}^t\mathbf{S}\mathbf{e}$, we want to select the eigenvector corresponding to the largest eigenvalue of the scatter matrix.
- In other words, to find the best one-dimensional projection of the *d*-dimensional data (in the least-sum-of-squared-error sense), project the data onto a line through the sample mean in the direction of the eigenvector of the scatter matrix with the largest eigenvalue.

Principal Component Analysis

- This result can be readily extended from a one-dimensional projection to a d'-dimensional projection ($d'<d$). In place of Eq. (81), we write

$$\mathbf{x} = \mathbf{m} + \sum_{i=1}^{d'} a_i \mathbf{e_i},$$

where $d' \leq d$.

- It is not difficult to show that the criterion function

$$J_{d'} = \sum_{k=1}^{n} \left\| (\mathbf{m} + \sum_{i=1}^{d'} a_{ki} \mathbf{e_i}) - \mathbf{x_k} \right\|^2$$

is minimized when the vectors $\mathbf{e_1}, \ldots \mathbf{e_{d'}}$ are the d' eigenvectors of $\mathbf{S}$ with the largest eigenvalues.

Fisher Linear Discriminant

- How to find the best direction **w** that will enable accurate classification?
- A measure of the separation between the projected points is the difference of the sample means. If $\mathbf{m_i}$ is the *d*-dimensional sample mean

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

then the sample mean for the projected points is

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y_i} y$$

$$= \frac{1}{n_i} \sum_{x \in D_i} \mathbf{w}^t \mathbf{x} = \mathbf{w}^t \mathbf{m_i}$$

Fisher Linear Discriminant

- The distance between the projected means is

$$|\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^t(\mathbf{m}_1 - \mathbf{m}_2)|,$$

  and we can make this difference as large as we wish merely by scaling **w**.
- To obtain good separation of the projected data we really want the difference between the means to be large relative to some measure of the standard deviations for each class.
- Define the *scatter* for projected samples labeled ω_i by

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

Fisher Linear Discriminant

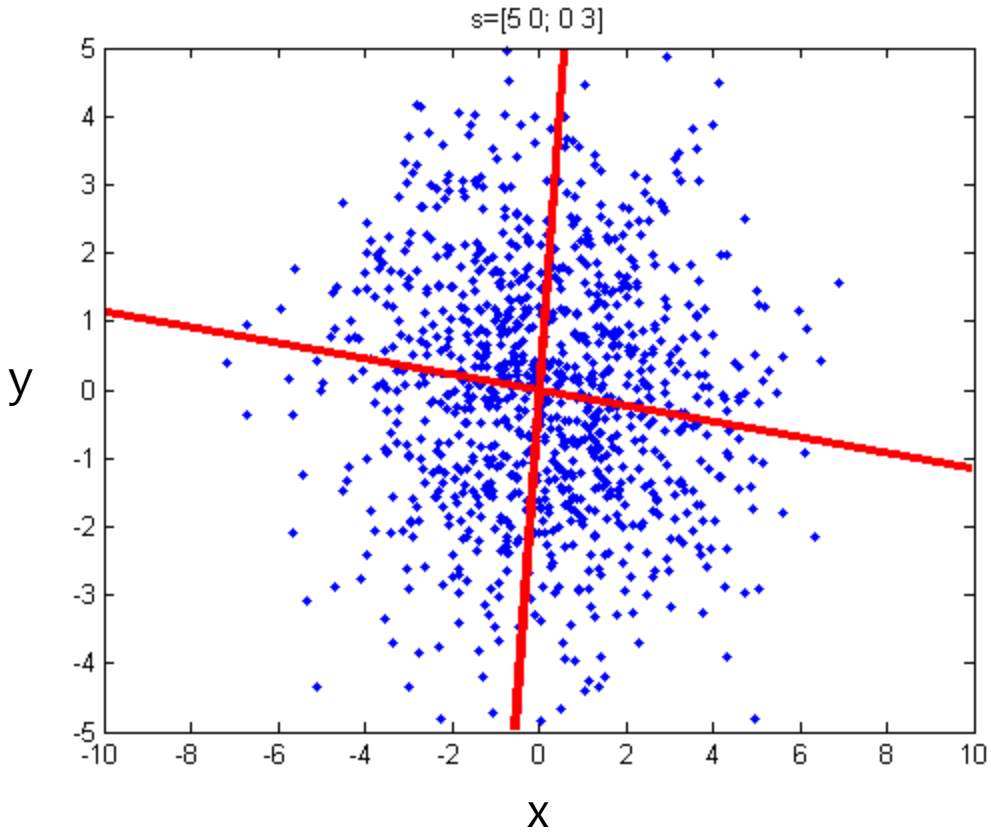
- Thus, $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$ is an estimate of the variance of the pooled data, and $\tilde{s}_1^2 + \tilde{s}_2^2$ is called the total *within-class scatter* of the projected samples. The *Fisher linear discriminant* employs that linear function $\mathbf{w}^t\mathbf{x}$ for which the criterion function

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

  is maximum (and independent of $||\mathbf{w}||$).

- The vector w maximizing $J(\cdot)$ leads to the best separation between the two projected sets.

- How to solve for the optimal **w**?



$\boldsymbol{\mu} = [2,\ 1]$

$\boldsymbol{\Sigma} = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix}$

Eigenvectors: $\begin{bmatrix} -0.1137 & -0.9935 \\ -0.9935 & 0.1137 \end{bmatrix}$

Eigenvalues: $\begin{bmatrix} 3.1757 & 0 \\ 0 & 5.3882 \end{bmatrix}$