Before the layered perceptron...

Fisher's Linear Discriminant Function

- Preliminaries: Gradients
  - Let $f(\vec{w})$ be a scalar function of the vector $\vec{w} = [w_1 \; w_2 \ldots w_d]^T$.
  - Then $\nabla_{\vec{w}} f(\vec{w})$ is a vector with components
    $\vec{\nabla}_w f(\vec{w}) = \left[\frac{\delta f}{\delta w_1} \quad \frac{\delta f}{\delta w_2} \ldots \frac{\delta f}{\delta w_d} \right]^T$
  - Example: $f(\vec{w}) = \vec{w}^T \vec{A} \vec{w} = \sum_{i=1}^{d} w_i \sum_{j=1}^{d} a_{ij} w_j$

Then

$$\left(\nabla_{\vec{w}} \vec{w}^T \vec{A} \vec{w}\right)_n = \frac{\delta}{\delta w_n} \vec{w}^T \vec{A} \vec{w}$$

$$= \frac{\delta}{\delta w_n} \sum_{i=1}^{d} \sum_{j=1}^{d} w_i a_{ij} w_j$$

$$= \sum_{j=1, j\neq n}^{d} a_{nj} w_j + \sum_{i=1, i\neq n}^{d} w_i a_{in} + 2 w_n a_{nn}$$

$$= \sum_{j=1}^{d} a_{nj} w_j + \sum_{i=1}^{d} w_i a_{in}$$

$$= \vec{a}_{nr}^T \vec{w} + \vec{w}^T \vec{a}_{nc}$$

($\vec{a}_{nr}^T$: n^{th} row of $\underline{A}$; $\vec{a}_{nc}$: n^{th} column of $\underline{A}$)

- Special case: $\underline{A} = \underline{A}^T$ (symmetric matrix)
  $\Rightarrow \vec{a}_{nr} = \vec{a}_{nc}$

$$\left(\nabla_{\vec{w}} \vec{w}^T \vec{A} \vec{w}\right)_n = 2 \vec{a}_{nr}^T \vec{w}$$

and:

$$\nabla_{\vec{w}} \vec{w}^T \underline{A} \vec{w} = 2 \underline{A} \vec{w} \qquad \text{(Der)}$$

- Problem Statement: Data in d-dimensions

  Q: How do we project this data into one dimension so that there is maximal separation?

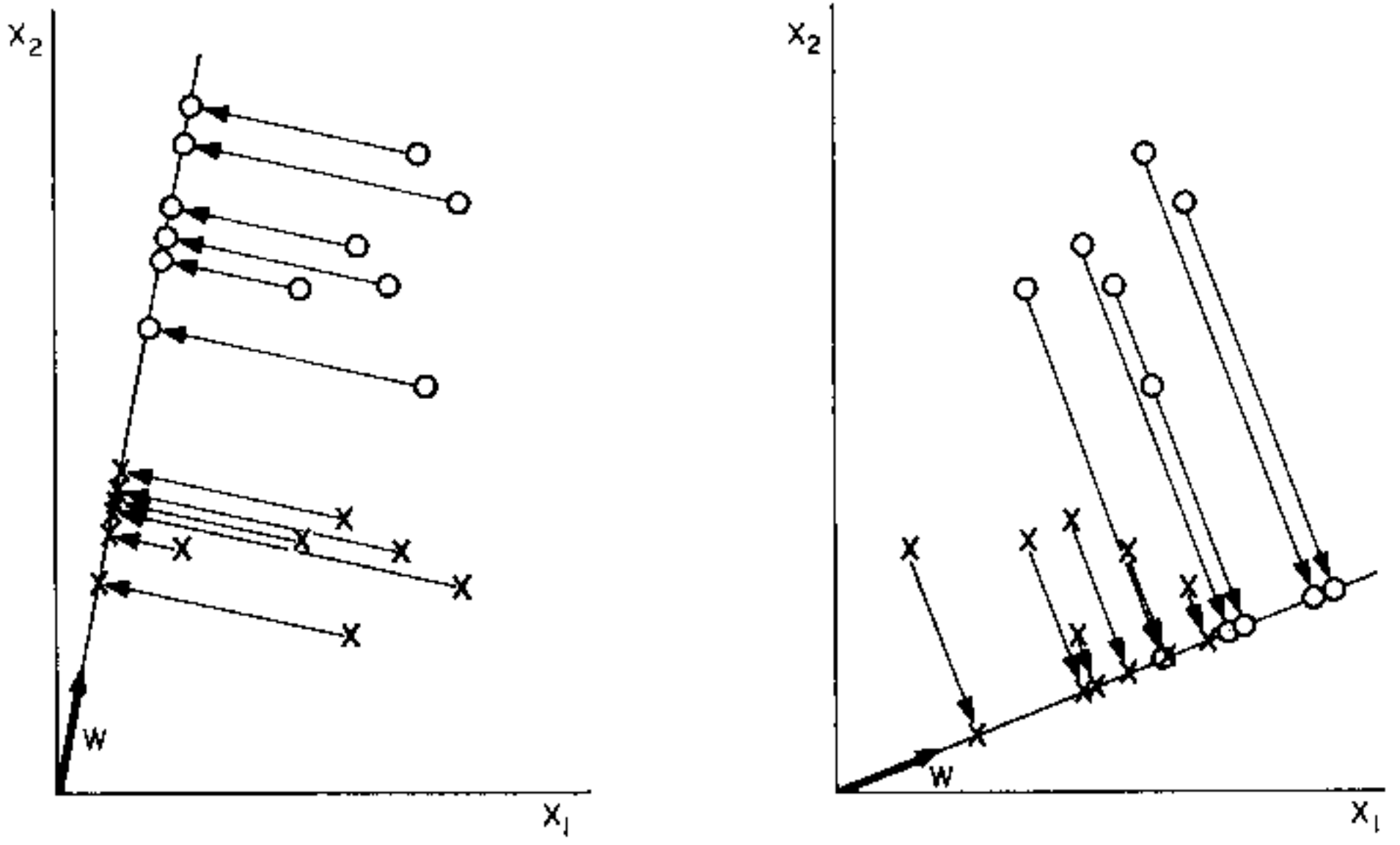
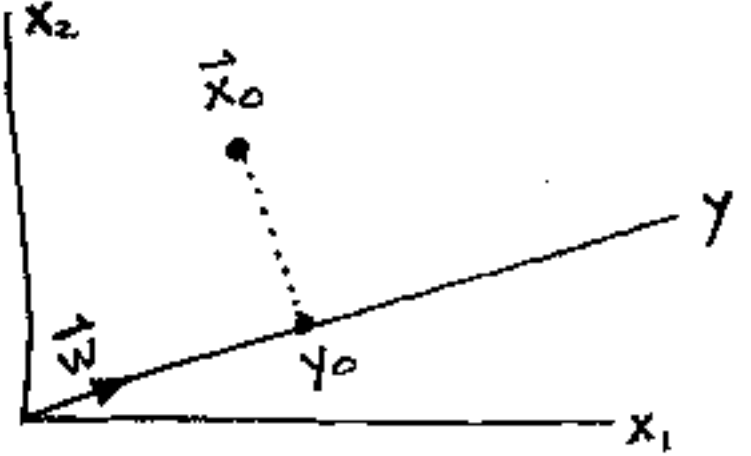


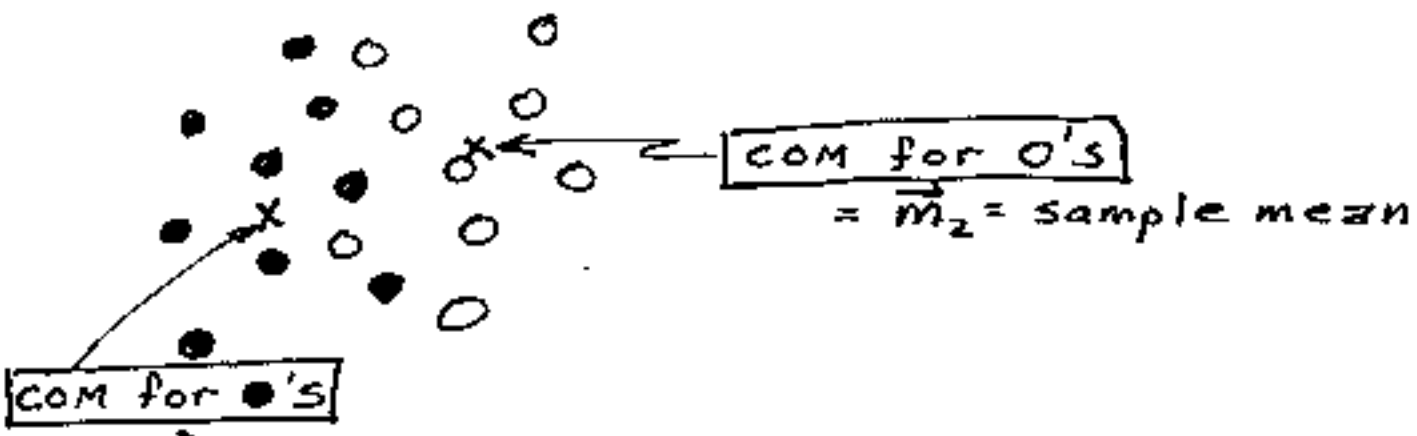
FIGURE 4.6. Projection of samples onto a line.

To project a vector onto $\vec{w}$, result is

$$y_0 = \vec{w}^T \vec{x}_0 \quad ; \quad \|\vec{w}\| = 1$$



Center of mass of data



n_1 = # data points in class 1 (χ_1)

n_2 = " " " " " 2 (χ_2)

$$\vec{m}_1 = \frac{1}{n_1} \sum_{\vec{x} \in \chi_1} \vec{x} \quad ; \quad \vec{m}_2 = \frac{1}{n_2} \sum_{\vec{x} \in \chi_2} \vec{x}$$

The projection onto some given $\vec{w}$ is

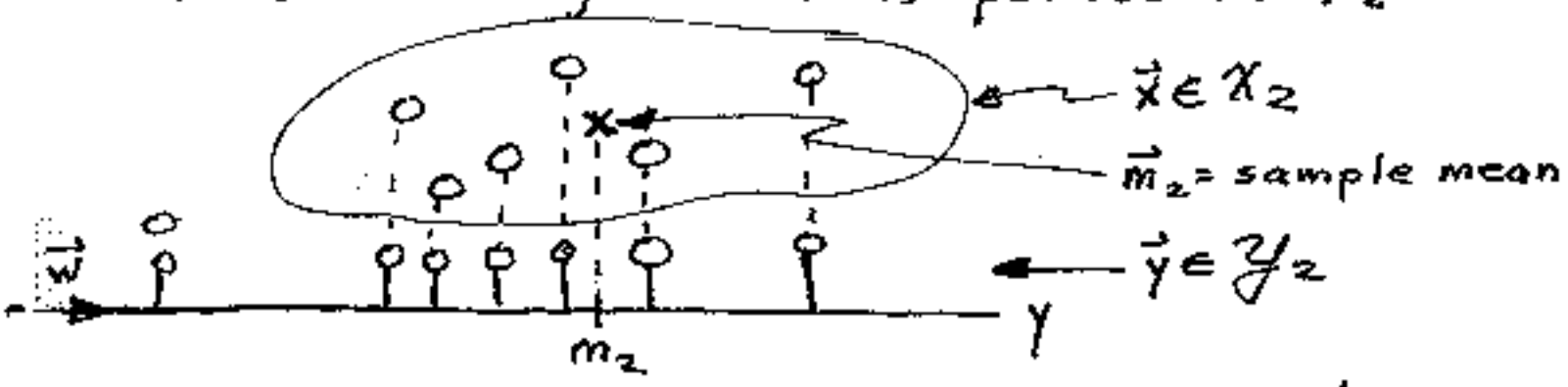
$$\tilde{m}_1 = \vec{w}^T \vec{m}_1 \quad ; \quad \tilde{m}_2 = \vec{w}^T \vec{m}_2$$

Project all the points for this $\vec{w}$

$$y = \vec{w}^T \vec{x}$$

If $\vec{x} \in \chi_1 \Rightarrow y \in \mathcal{Y}_1$. If $\vec{x} \in \chi_2 \Rightarrow y \in \mathcal{Y}_2$

Illustraton: Projection of points in $\mathcal{X}_2$



This 1-D data has a sample mean and sample variance

$$\tilde{m}_2 = \frac{1}{n_2} \sum_{y \in \mathcal{Y}_2} y = \frac{1}{n_2} \sum_{\vec{x} \in \mathcal{X}_2} \vec{w}^T \vec{x}$$

$$= \vec{w}^T \left[\frac{1}{n_2} \sum_{x \in \mathcal{X}_2} \vec{x} \right]$$

$$= \vec{w}^T \vec{m}_2$$

$$\sigma_2^2 = \frac{1}{n_2} \sum_{y \in \mathcal{Y}_2} (y - m_2^2) \; ; \quad \left(\text{some use } \frac{1}{n_2 - 1} \right)$$

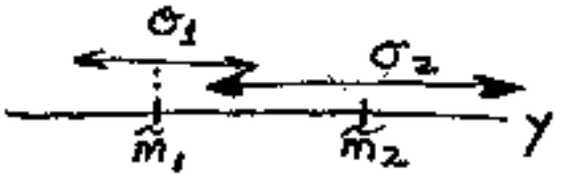
σ_2 measures the data spread

small σ_2 ⟷ ∞ ∞

larger σ_2 ⟷ 0 ∞ 0

σ_2 = standard deviation sample

For a given $\vec{w}$, projected data looks like



These four parameters are functions of $\vec{w}$.

Q: How do we define "maximally separated data"

A: Fisher's answer: Find the maximum value of

$$\frac{|m_1 - m_2|^2}{\sigma_1^2 + \sigma_2^2}$$

A small alteraton. Define the "scatter" in each class as

$$\tilde{s}_1^2 = \sum_{y \in \mathcal{Y}_1} (y - m_1)^2 = n_1 \sigma_1^2$$

$$\tilde{s}_2^2 = \sum_{y \in \mathcal{Y}_2} (y - m_2)^2 = n_2 \sigma_2^2$$

The "Fisher distance" between the data sets is

$$J(\vec{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{s_1^2 + s_2^2} \qquad \text{(FD)}$$

- Problem: Find the $\vec{w}$ that maximizes $J(\vec{w})$
- Answer: (To be derived)

$$\vec{w} = \underline{S}_w^{-1} (\vec{m}_1 - \vec{m}_2) \qquad \text{(FV)}$$

where

$$\underline{S}_w = \underline{S}_1 + \underline{S}_2$$

and

$$\underline{S}_1 = \sum_{\vec{x} \in \mathcal{X}_1} (\vec{x} - \vec{m}_1)(\vec{x} - \vec{m}_1)^T; \quad \underline{S}_2 = \sum_{x \in \mathcal{X}_2} (\vec{x} - \vec{m}_2)(\vec{x} - \vec{m}_2)^T$$

Proof:

$$\tilde{s}_1^2 = \sum_{\vec{x}\in\chi_1} (y - \tilde{m}_1)^2$$

$$= \sum_{\vec{x}\in\chi_1} \left(\vec{w}^T\vec{x} - \vec{w}^T m_1\right)^2$$

$$= \sum_{\vec{x}\in\chi_1} \left[\vec{w}^T(\vec{x} - \vec{m}_1)\right]^2$$

$$= \sum_{\vec{x}\in\chi_1} \vec{w}^T(\vec{x} - \vec{m}_1)(\vec{x} - \vec{m}_1)^T\vec{w}$$

$$= \vec{w}^T\left[\sum_{\vec{x}\in\chi_1} (\vec{x} - \vec{m}_1)(\vec{x} - \vec{m}_1)^T\right]\vec{w}$$

$$= \vec{w}^T \underline{S}_1 \vec{w} \quad ; \quad \underline{S}_1 = \sum_{\vec{x}\in\chi_1} (\vec{x} - \vec{m}_1)(\vec{x} - \vec{m}_1)^T$$

(Scatter matrix)

Similarly

$$\tilde{s}_2^2 = \vec{w}^T \underline{S}_2 \vec{w} \quad ; \quad \underline{S}_2 = \sum_{\vec{x}\in\chi_2} (\vec{x} - \vec{m}_2)(\vec{x} - \vec{m}_2)^T$$

Thus:

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \vec{w}^T \underline{S}_1 \vec{w} + \vec{w}^T \underline{S}_2 \vec{w}$$

$$= \vec{w}^T \underline{S}_W \vec{w} \quad ; \qquad \text{(Den)}$$

$\underline{S}_W = \underline{S}_1 + \underline{S}_2$ = "within class scatter matrix" (WI)

Also:

$$|\tilde{m}_1 - \tilde{m}_2|^2 = |\vec{w}^T(\vec{m}_1 - \vec{m}_2)|^2$$

$$= \vec{w}^T(\vec{m}_1 - \vec{m}_2)(\vec{m}_1 - \vec{m}_2)^T\vec{w}$$

$$= \vec{w}^T \underline{S}_B \vec{w} \qquad \text{(Num)}$$

where

$$\underline{S}_B = (\vec{m}_1 - \vec{m}_2)(\vec{m}_1 - \vec{m}_2)^T = \text{between-class scatter matrix} \quad \text{(BE)}$$

The Fisher distance (from (FD) on p.5) is

$$J(\vec{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{s_1^2 + s_2^2}$$

Substitute ((Den) and (Num) from p. 6):

$$J(\vec{w}) = \frac{\vec{w}^T \underline{S}_B \vec{w}}{\vec{w}^T \underline{S}_w \vec{w}} \qquad (FD)$$

Note: Maximizing $J(\vec{w})$ minimizes within class scatter ($\vec{w}^T \underline{S}_w \vec{w}$) and maximized between class scatter ($\vec{w}^T \underline{S}_B \vec{w}$).

Note: Both numerator & den in (FD) above are scalars

Strategy to maximize $J(\vec{w})$: Take first derivative & set to zero.

$$\nabla_{\vec{w}} J(\vec{w}) = \frac{[\nabla_{\vec{w}} \vec{w}^T \underline{S}_B \vec{w}]\vec{w}^T \underline{S}_w \vec{w} - [\nabla_w \vec{w}^T \underline{S}_w \vec{w}]\vec{w}^T \underline{S}_B \vec{w}}{(\vec{w}^T \underline{S}_w \vec{w})^2}$$

Using (Der) on p. 1 $\Rightarrow$

$$= \frac{2\ \underline{S}_B \vec{w}\ \vec{w}^T \underline{S}_w \vec{w} - 2\ \underline{S}_w \vec{w}\ \vec{w}^T \underline{S}_B \vec{w}}{(\vec{w}^T S_w \vec{w})^2}$$

Set $\nabla_{\vec{w}} J(\vec{w}) = \vec{0}$ = zero vector. This gives

$$\underline{S}_B \vec{w}\ \vec{w}^T \underline{S}_w \vec{w} = S_w\ \vec{w}\ \vec{w}^T \underline{S}_B \vec{w}$$

(Note: Since $\underline{S}_w$ is symmetric, $S_w^{-1} = (S_w^{-1})^T$).

This is an equality when (FV) on p.5 is true. Then

$$\vec{w}\ \vec{w}^T = \underline{S}_w^{-1}(\vec{m}_1 - \vec{m}_2)(\vec{m}_1 - \vec{m}_2)S_w^{-1} = \underline{S}_w^{-1} \underline{S}_B \underline{S}_w^{-1}$$

Substituting:

$$S_B(S_w^{-1} S_B \cancel{S_w^{-1}})\cancel{S_w}\ \vec{w} = S_w(\cancel{S_w^{-1}} S_B S_w^{-1})\ S_B\ \vec{w}$$

$$\Rightarrow S_B S_w^{-1} S_B \vec{w} = S_B S_w^{-1} S_B \vec{w}$$

and the proof is done!

Summary: Fisher discriminant

- Find Fisher vector, $\vec{w}$, to maximize

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

- Solution is Fisher vector

$$\Rightarrow \quad \vec{w} = \underline{S}_W^{-1} (\vec{m}_1 - \vec{m}_2)$$

where

- $\underline{S}_W = \underline{S}_1 + \underline{S}_2 \Leftarrow$ within class scatter matrix
- $\underline{S}_1 = \sum_{\vec{x} \in \chi_1} (\vec{x} - \vec{m}_1)(\vec{x} - \vec{m}_1)^T$; $\underline{S}_2 = \sum_{\vec{x} \in \chi_2} (\vec{x} - \vec{m}_2)(x - \vec{m}_2)^T$
- $\vec{m}_1 = \frac{1}{n_1} \sum_{\vec{x} \in \chi_1} \vec{x}$; $\vec{m}_2 = \frac{1}{n_2} \sum_{\vec{x} \in \chi_2} \vec{x}$

Assumptions:

1. $\underline{S}_W$ is not singular.

- Necesary condition: $n_1 + n_2 \geq d$ (elaborate)

More generally, $\underline{S}_W$ must be "well conditioned"
The condition of a matrix, $\underline{A}$, is

$$C(\underline{A}) = \frac{|\lambda_{max}|}{|\lambda_{min}|}$$

For singular matrices, $C(\underline{A}) = \infty$.

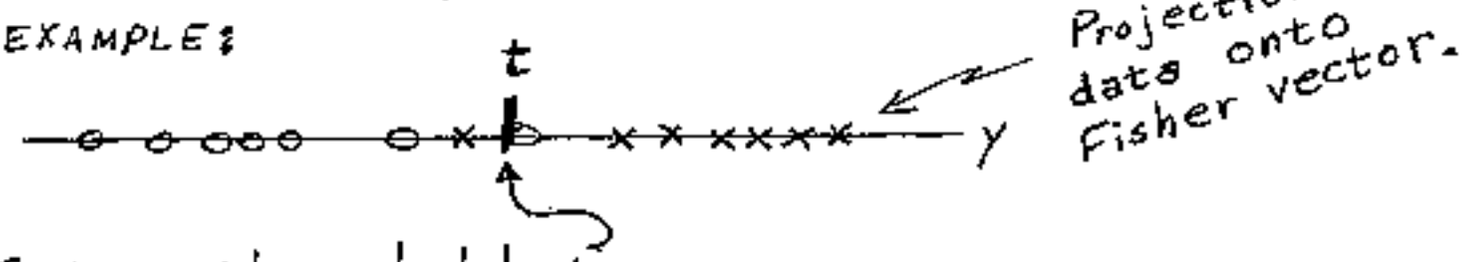
- Necesary condition: All data can not lie on a plane with dimension $< d$.
(Equivalent to non-singular requirement)

2 The data classes are linear seperable.

USING THE FISHER VECTOR FOR CLASSIFICATION (9)

Q: After projection onto the Fisher vector, how do we classify?

EXAMPLE:



A: Set a threshold, t.
Above t, we have an "x", below an "o."

Q: Where do we place t?

A: Depends on your desired performance.

Example: x = thermonulear attack
o = no attack

Couched in classical hypothesis testing:

H_0 : □ = O = no attack

H_1 : □ = X = attack

where □ is the observation.

$\Rightarrow \alpha$ = FALSE ALARM PROBABILITY

$= Pr[H_1 \mid \square = O]$

= Prob [Announcing H_1 GIVEN OBSERVATION IS "NO ATTAC

$\Rightarrow \beta$ = DETECTION PROBABILITY

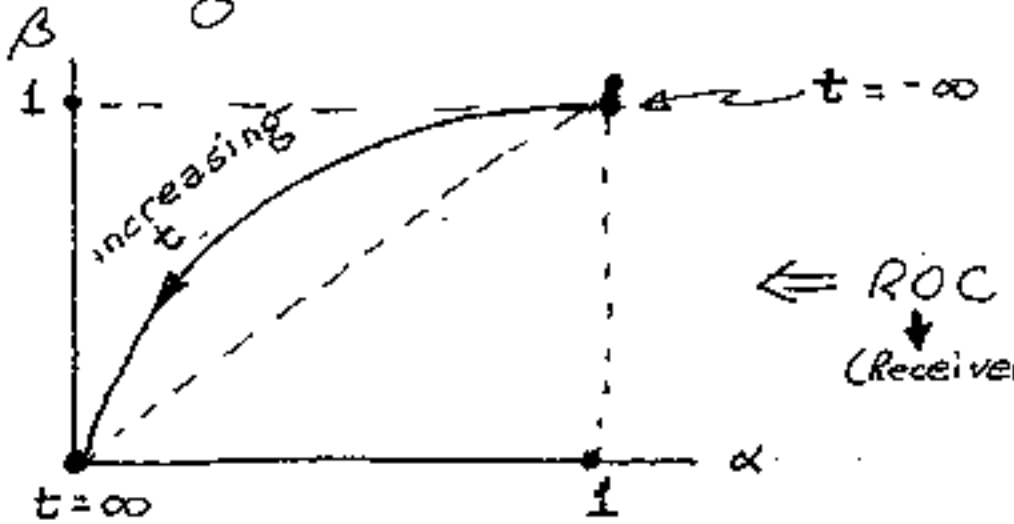
$= Pr[H_1 \mid \square = X]$

= Prob [Announcing H_1 GIVEN OBSERVATION IS "ATTACK"

To get $\beta = 1$, set $t = -\infty$. But $\alpha = 1$ (Always "attack!")

To get $\alpha = 0$, set $t = \infty$. But $\beta = 0$ (Always "no attack")

Increasing β increases α
Decreasing α decreases β

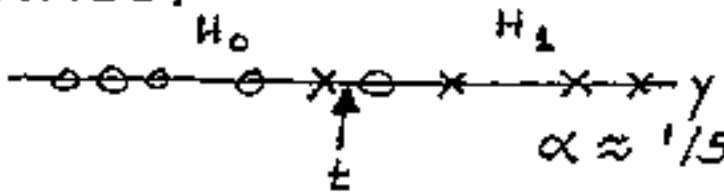


⇐ ROC Curve
↓
(Receiver Operating Characteris

Choosing t specifies your point on the ROC curve.

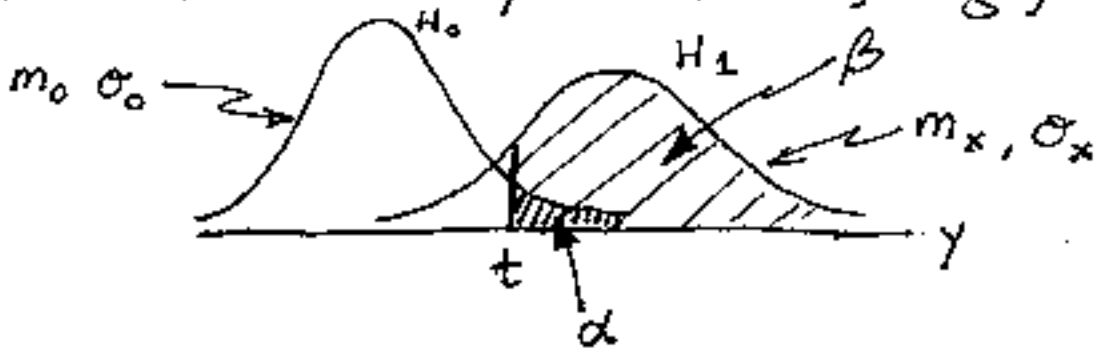
Q: How is α & β estimated?

A1: Empirical



$\alpha \approx 1/5$ $\beta \approx 3/4$

A2: Parametrically. Assume, e.g., Gaussian



Multivariate Normal Density

$$f_{\vec{X}}(\vec{x}) = \frac{1}{(2\pi)^{d/2}\,|\underline{\Sigma}|^{1/2}}\; e^{-\frac{1}{2}(\vec{x}-\vec{m})^T \underline{\Sigma}^{-1}(\vec{x}-\vec{m})}$$

$$\vec{m} = E[\vec{X}] \approx \frac{1}{n}\sum_m \vec{X}_m \Leftarrow \text{AVERAGE}$$

$$\sigma_{ij} = E[(X_i - m_i)(X_j - m_j)] = (\underline{\Sigma})_{ij}$$

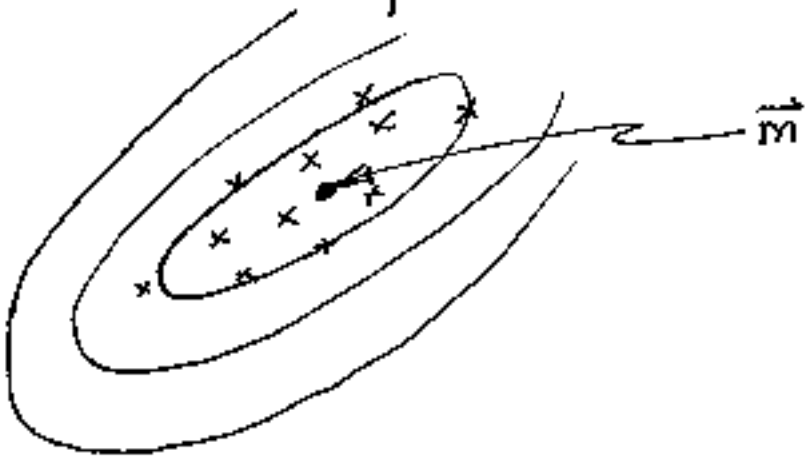
$$\underline{\Sigma} = E\left[(\vec{X}-\vec{m})(\vec{X}-\vec{m})^T\right]$$

$$\approx \frac{1}{n}\sum_m (\vec{X}_m - \vec{m})(\vec{X}_m - \vec{m})^T$$

Q: What do contours of $f_{\vec{X}}(\vec{x})$ look like?
A: Same as

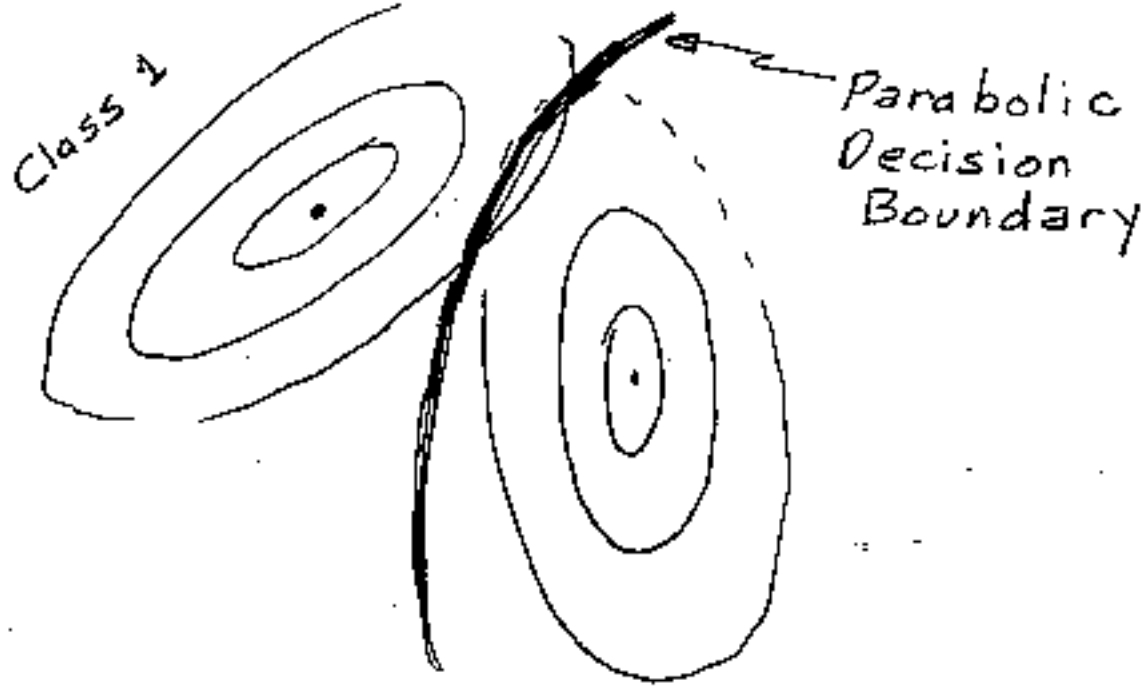
$$r^2 = (\vec{x}-\vec{m})^T \underline{\Sigma}^{-1}(\vec{x}-\vec{m})$$

In 2-D, it's ellipse centered at $\vec{m}$.



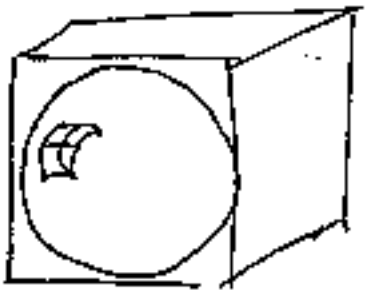
r = Mahalanobis distance

Minimum Mahalanobis distance classifier



Can extend to 3 or more classes.

← To distinguish circle, we need, say, 100 data points in 2-d.



← In 3-d, need 1000 pts

In d dimensions $\Rightarrow 10^d$ ⇐ wow!

Thus: If there are a lot of features, one might one to choose the "best" subset.

<u>But</u>, if there are n features, and we want to choose the best k, an exhaustive search would require

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!} \quad \text{evaluations}$$

Alternate idea: Choose the best k features

But: The k best features are <u>not</u> the best k features!

(Elaborate)

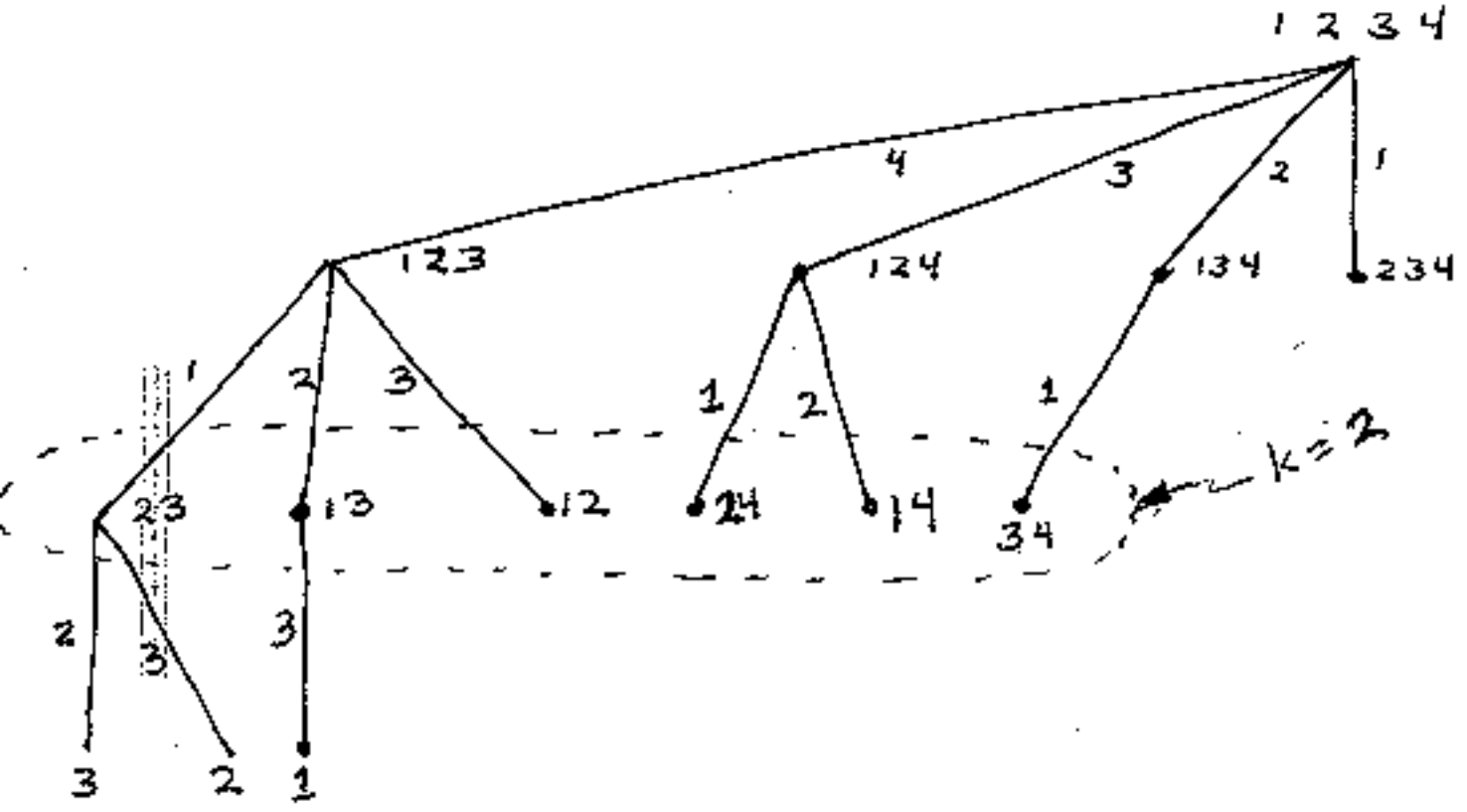
Suboptimal Choice: Sequential Selection

1. Choose best feature, a
2. " feature that goes best with a (call it b).
3. Choose feature c that goes best with a and b.

etc.

Note: Increasing by a feature increases Fisher distance.

Dynamic Programming for Optimality (15)



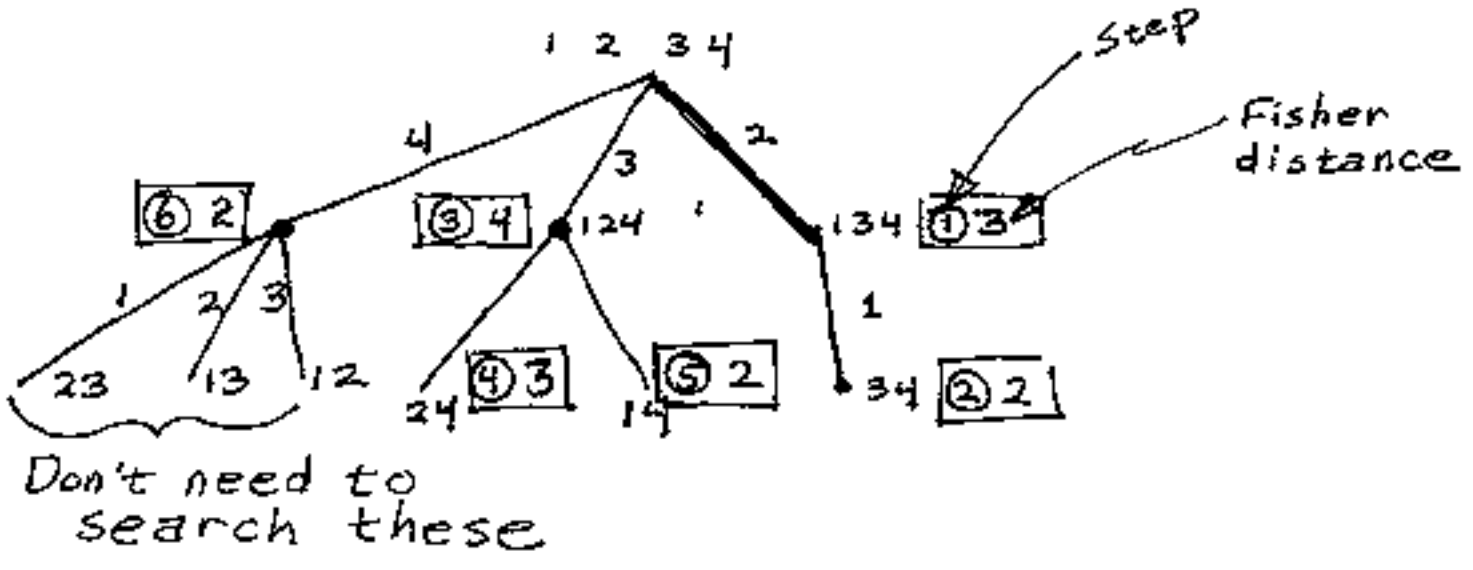
$\binom{4}{0}$

$\binom{4}{1}$

$\binom{4}{2} = 6$

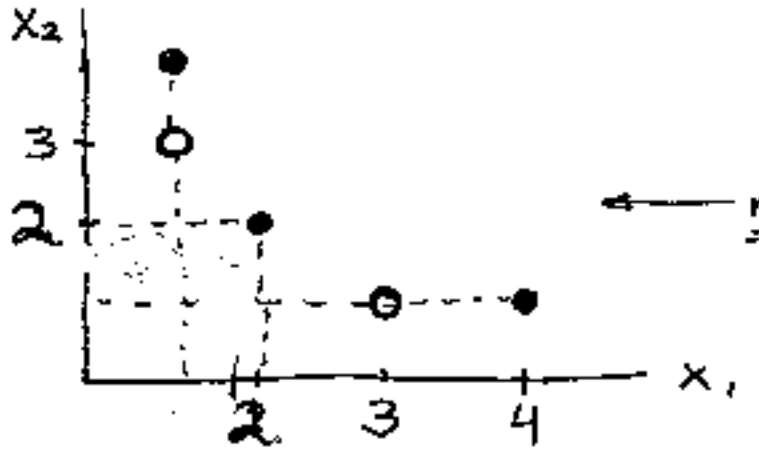
$\binom{4}{3} = 4$

Choose two best features



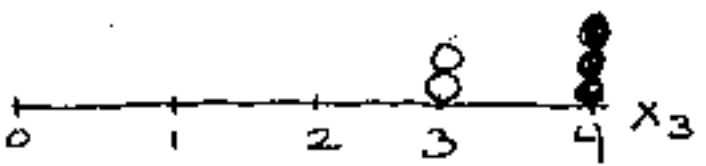
Functional Feature Alteration

Ex



← not linearly seperable

Let $X_3 = X_1 \cdot X_2$



X_1 & X_2 are sides of rectangle
X_3 is area.

PRINCIPAL COMPONENTS / EIGEN DECOMPOSITION / K-L Analysis

$\vec{y} = \underline{\underline{\Sigma}}\,\vec{x}$ ← d dimensions

The eigenvalues, λ_n, and eigenvectors, are defined by

$\lambda_n \vec{\psi}_n = \underline{\underline{\Sigma}}\,\vec{\psi}_n \quad ; \quad 1 \le n \le d$

They have the following properties:

(a) Orthogonality : $\vec{\psi}_n^T \vec{\psi}_m = \delta[n-m] = \begin{cases} 1 & ; n=m \\ 0 & ; n \neq m \end{cases}$

(b) $\underline{\underline{\Sigma}} = \sum_{n=1}^{d} \lambda_n \vec{\psi}_n \vec{\psi}_n^T$

Any vector can be expressed as

$\vec{x} = \sum_{n=1}^{d} \alpha_n \vec{\psi}_n$

where $\alpha_n = \vec{x}^T \vec{\psi}_n$

Thus: $\vec{y} = \underline{\underline{\Sigma}}\,\vec{x} = \underline{\underline{\Sigma}} \sum_{n=1}^{d} \alpha_n \vec{\psi}_n$

$= \sum_{n=1}^{d} \alpha_n \underline{\underline{\Sigma}}\,\vec{\psi}_n$

$= \sum_{n=1}^{d} \alpha_n \lambda_n \vec{\psi}_n$ ⇐

PRINCIPAL COMPONENTS

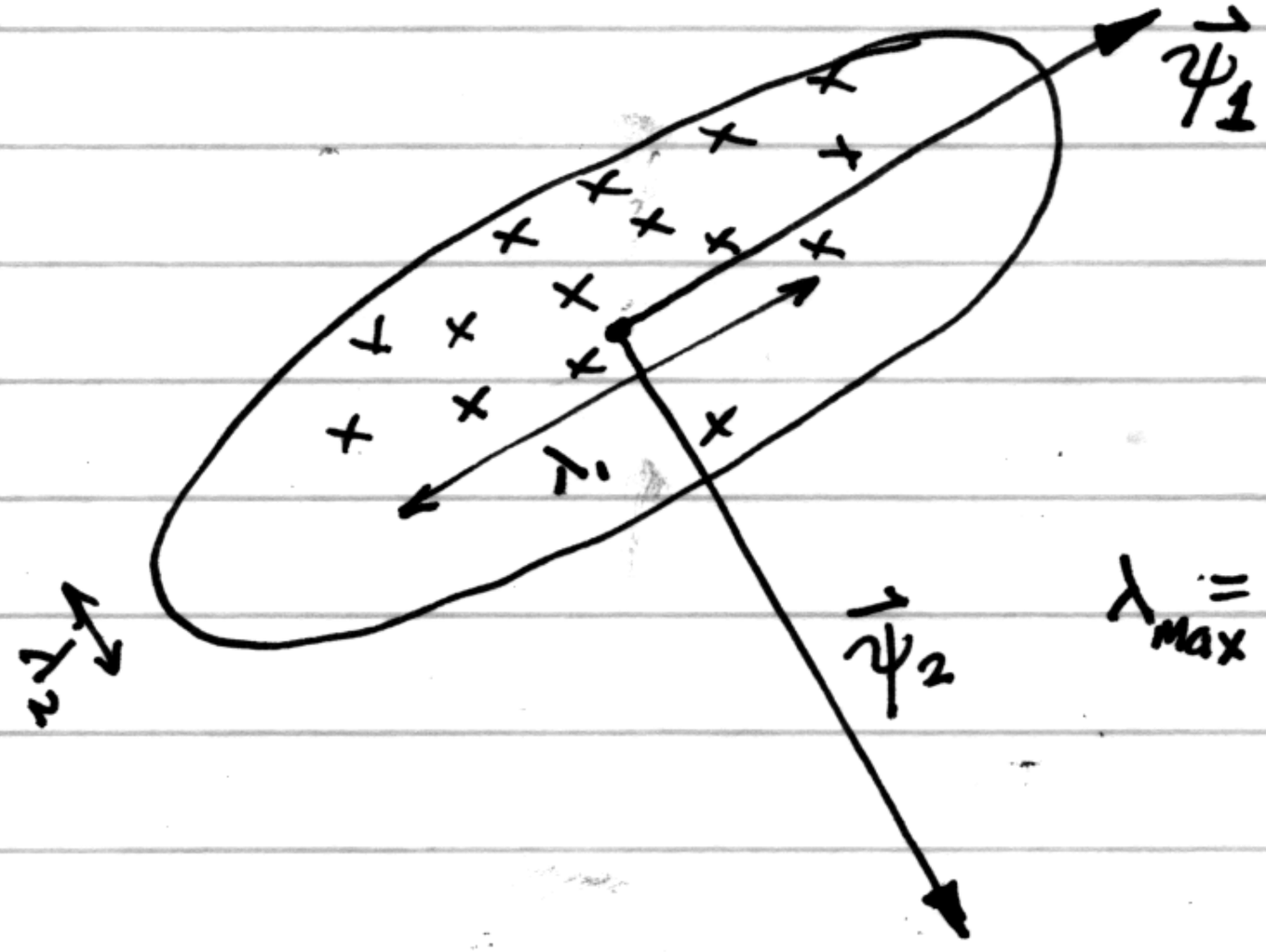
Idea: Many of λ_n's are such that $|\lambda_n| << |\lambda|_{max}$

Interpretation: If $\underline{\underline{\Sigma}}$ is estimated by data:

$\underline{\underline{\Sigma}} \approx \frac{1}{d-1} \sum_{n=1}^{d} (\vec{x}_n - \vec{m})(\vec{x}_n - \vec{m})^T$

$\vec{m} \approx \frac{1}{d} \sum_{n=1}^{d} \vec{x}_n$

Relation of eigenvalues to ellipses



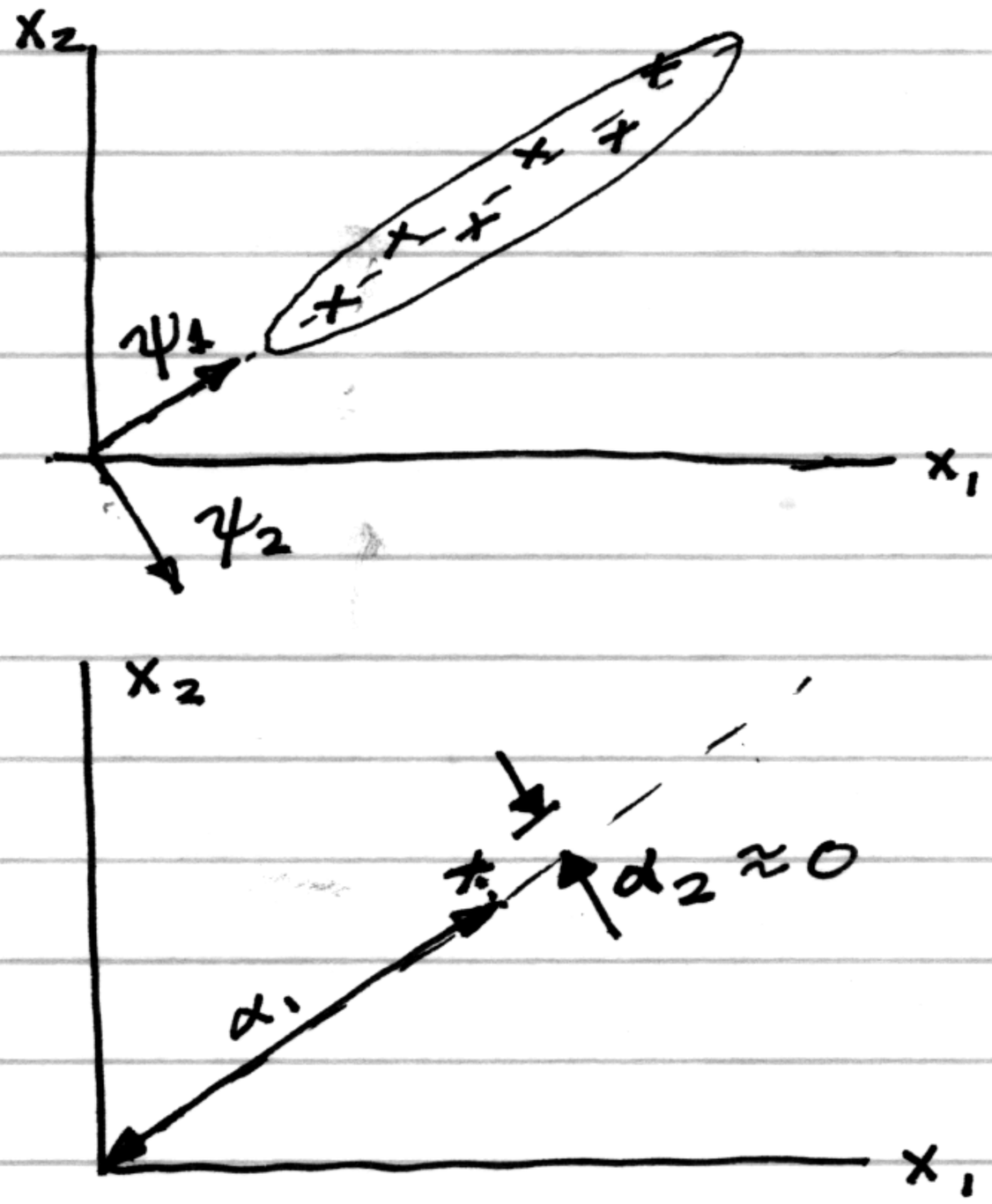
Note: $\underline{\Sigma}$ is positive semi-definite, so

$$\lambda_{max} = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_1 \geq 0$$

If ellipsoid is "flat" in certain dimensions, the corresponding λ_n's are "small". Then, for all of the $\vec{x}$'s in this flat ellipse,

$$\alpha_n \cong 0$$

Example:



Thus, if λ_n's are "small" for $n > \hat{d}$,

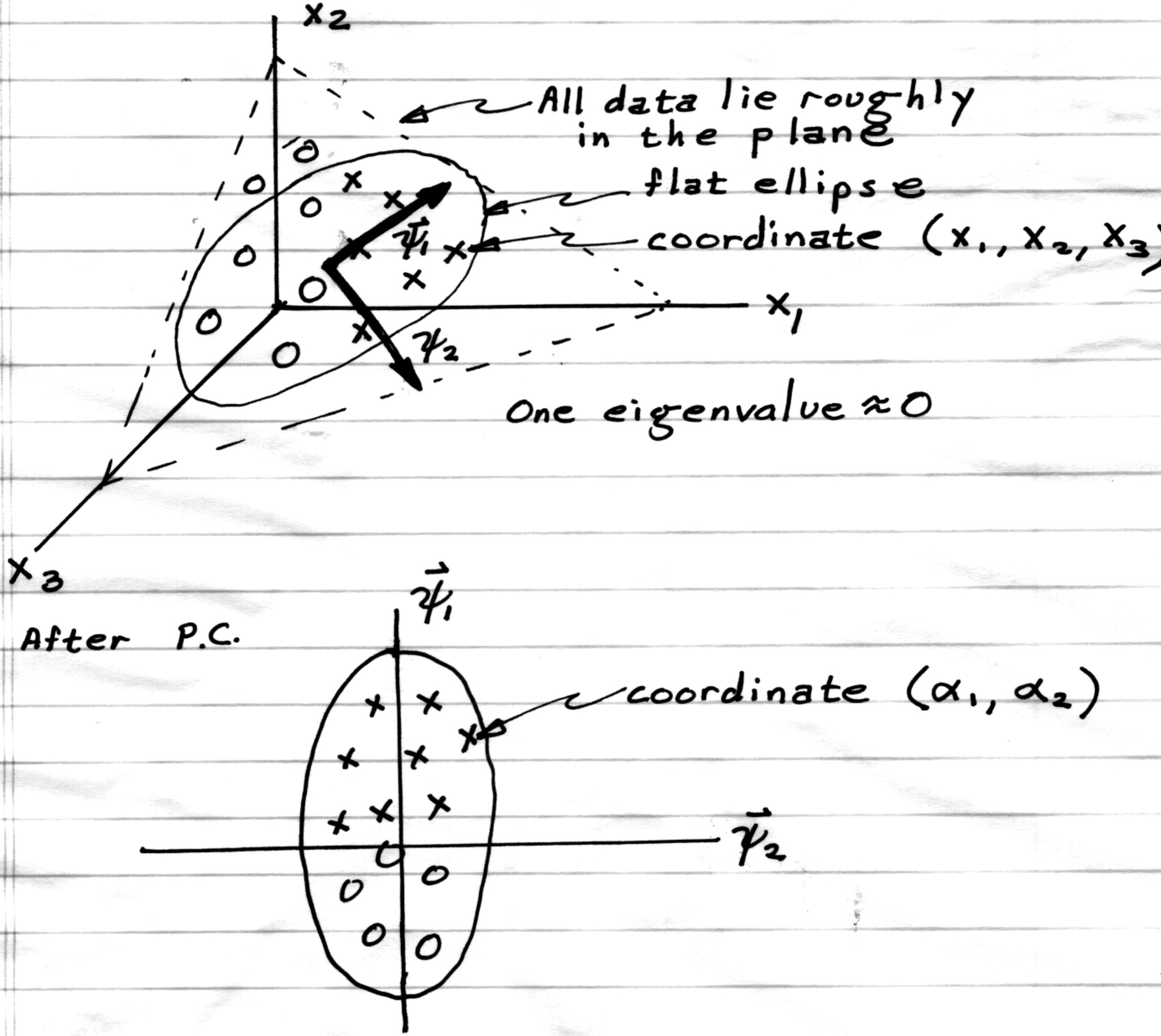
$$\vec{x} = \sum_{n=1}^{d} \alpha_n \vec{\psi}_n \approx \sum_{n=1}^{\hat{d}} \alpha_n \vec{\psi}_n$$

The $\hat{d} < d$ numbers, $\{\alpha_n \mid 1 \le n \le \hat{d}\}$ have about the same information as $\{x_m \mid 1 \le m \le d\}$.

We have reduced the dimension by $d - \hat{d} \Rightarrow$ Good news against the curse.

Use $\vec{\alpha}$ vectors in place of $\vec{x}$ vectors when training.

Geometrical Interpretation



After P.C.

Note:

- Principal Components (a.k.a. Karhunen-Loeve) Analysis can be used for FEATURE EXTRACTION
- Choosing the k best features is FEATURE SELECTION

→ Both reduce dimensions, but are different.

1. All features are used in PC
2. The classification label is not used in P.C.

Use in data visualization.

Problem: We have d-dimensional data.
We desire to view it projected onto a two-dimensional plane so that the data is "spread-out" the most.

Solution: Use the <u>two</u> principal components. Plot the data in the resulting two dimensional plane.