Object Recognition

Javier González Jiménez

Reference Books:

- *Computer Vision: Algorithms and Applications*. Richard Szeliski. Springer. 2010.

http://szeliski.org/Book

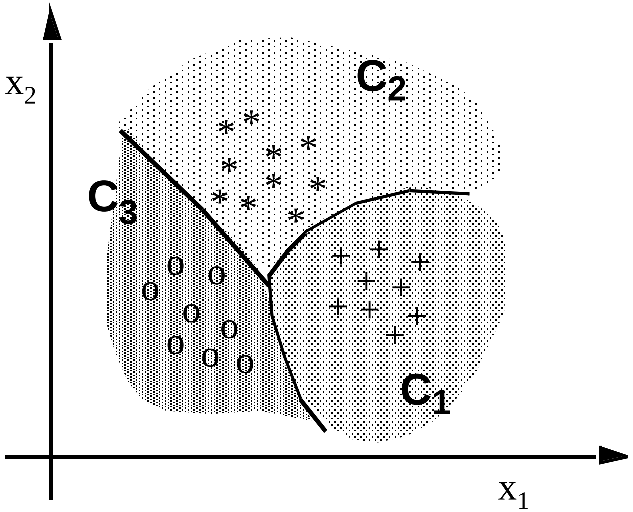
- *Pattern Recognition and Machine Learning*. Christopher Bishop. Springer-Verlag New York,.2006.

Content

- Introduction
- Discriminant functions
  - Linear discriminant functions
  - Linear Basis Function Models
- Naive Bayes Classifiers
  - Basis
  - Binomial distribution
  - Gaussian distribution
- Support Vector Machine (not included)

1. Introduction

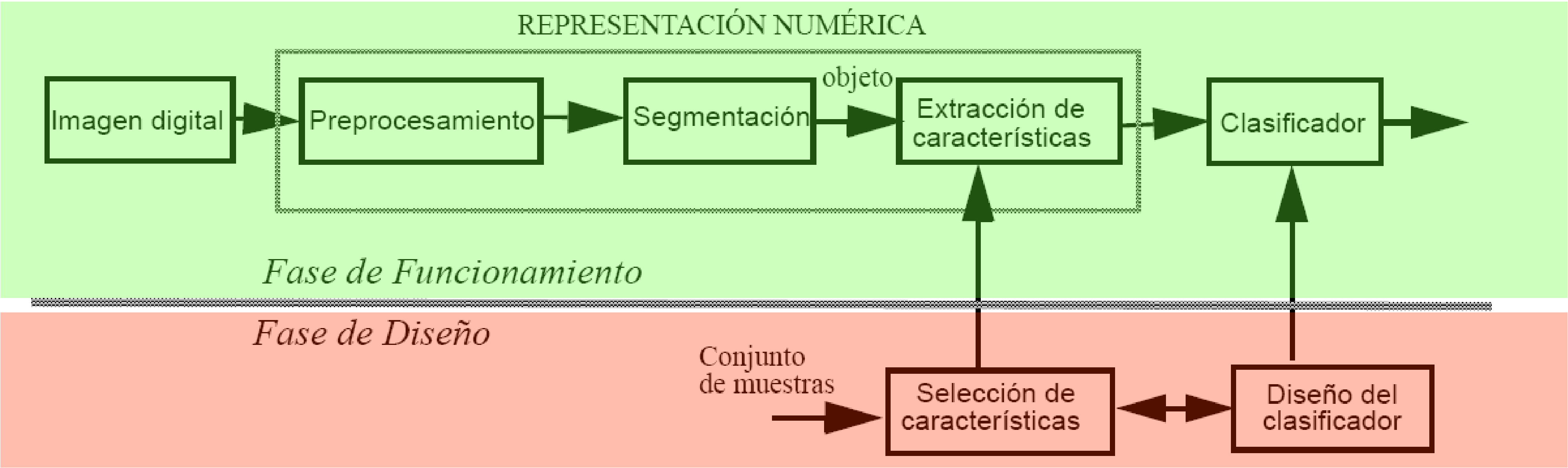
- In previous lecture we learned how to extract a feature vector $\mathbf{x}$ to describe an image region
- Now, we want to **classify the region**, based on that vector $\boldsymbol{x}$, as one of M posible object classes (or categories)
- For that, we **divide the feature space** into a number of prediction subspaces C_i: if a feature $\mathbf{x}$ lies in C_i it is assigned to the object class C_i represents



1. Introduction

The recognition process comprises <u>two steps</u>:

- a **training (design) phase**, where sample vectors of known objects are used to learn the classifier (supervised learning)
- a **prediction (online) phase**, where the image objects are classified to one of the classes based on the learned prediction model



1. Introduction

Approaches:

Statistical classifiers:

- We assume the feature vectors $\mathbf{x}$ of the classes C follow a **statistical distribution**
- The parameters of such distribution need to be **learned from known objects**
- Two statistical models can be considered:
  - **Generative models**: the parameters of the joint $p(C, \mathbf{x}) = p(\mathbf{x}|C)p(C)$ are learned

    Example: *Naïve Bayesian Classifier*
  - **Discriminative models**: the parameter of the posterior $p(C|\mathbf{x})$ are learned

    Example: *Logistic regression, conditional random fields*

Non-statistical classifiers:

- No assumption is made on the statistical distribution of the feature vector
- The coefficient of deterministic discriminant functions are learned

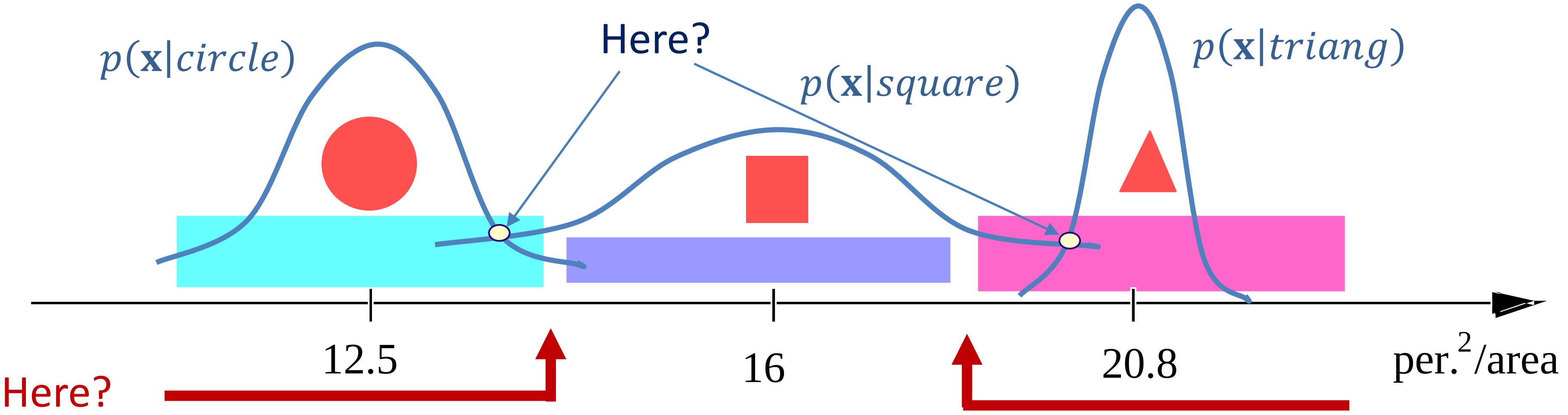
Example: *Support Vector Machine, Perceptron, AdaBoost*

Example:

Classification of objects, based on compactness ($\mathbf{x}$), into 3 classes (C)

		AREA	PERIMETER	$\mathbf{x}$ = PER.^{2}/AREA (COMPACTNESS^{-1})
circle		πR^2	$2\pi R$	12.5
square		L^2	$4L$	16
equilateral triangle		$(\sqrt{3}/4)L^2$	$3L$	20.8

Partition of the $\mathbf{x}$ space: Where to set the threshold for an optimal classification?



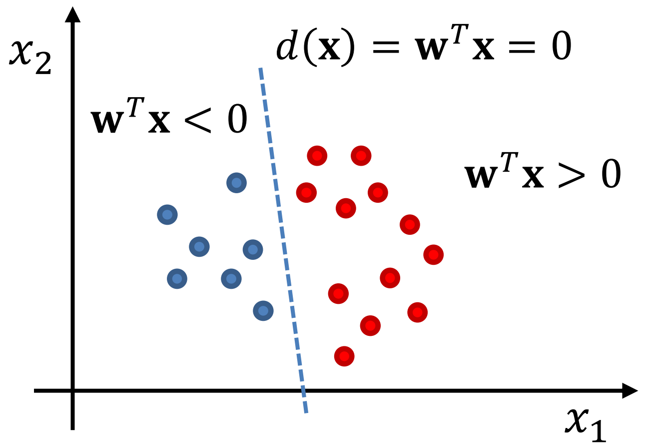
2. Discriminant functions

Linear discriminant functions

For two classes C_1, C_2 :

Hyper-planes in the n-dimensional space

$$d(\mathbf{x}) = w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n + w_{n+1} = \mathbf{w}_0^T \mathbf{x} + w_{n+1}$$



$\mathbf{w} = [w_1 \quad w_2 \quad \ldots \quad w_n]^T$ **Weight Vector** → To be learned in the training phase

$\mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n]^T$ **Feature Vector**

More convenient as an **augmented form**:

$$\left.\begin{array}{l} \mathbf{w} = [w_1 \quad w_2 \quad \ldots \quad w_n \quad w_{n+1}]^T \\ \mathbf{x} = [x_1 \quad x_2 \quad \ldots \quad x_n \quad \underbrace{1]^T}_{\text{augmentation}} \end{array}\right\} \Rightarrow d(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

Dot (scalar) product of the two vectors

2. Discriminant functions

Linear discriminant functions

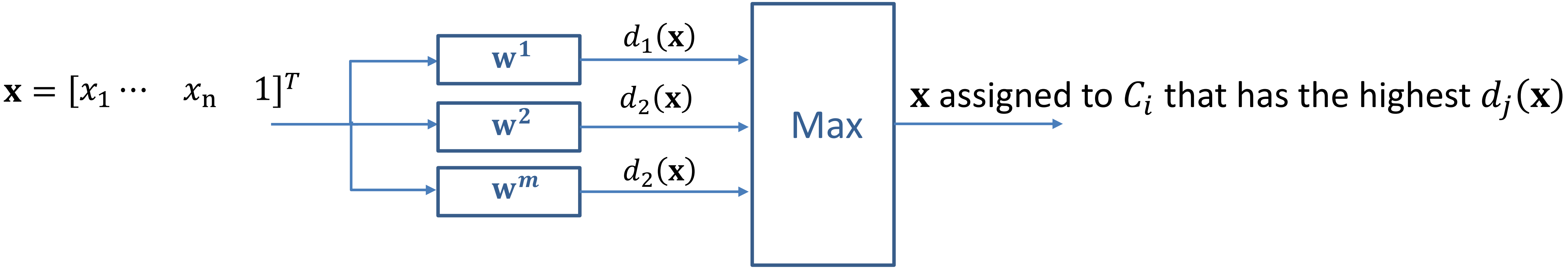
More than two classes (m classes): A linear function for each class

$$d_i(\mathbf{x}) = {w_1}^i \cdot x_1 + {w_2}^i \cdot\cdot\, x_2 + \ldots + {w_n}^i \cdot\cdot\, x_n + {w_{n+1}}^i = (\mathbf{w}^i)^T \cdot \mathbf{x}$$

Weights to be computed during the training phase

Classification criterion:

IF $d_i(\mathbf{x}) > d_j(\mathbf{x}) \;\; \forall i \neq j \;\; \text{THEN} \;\; \mathbf{x} \in C_i$

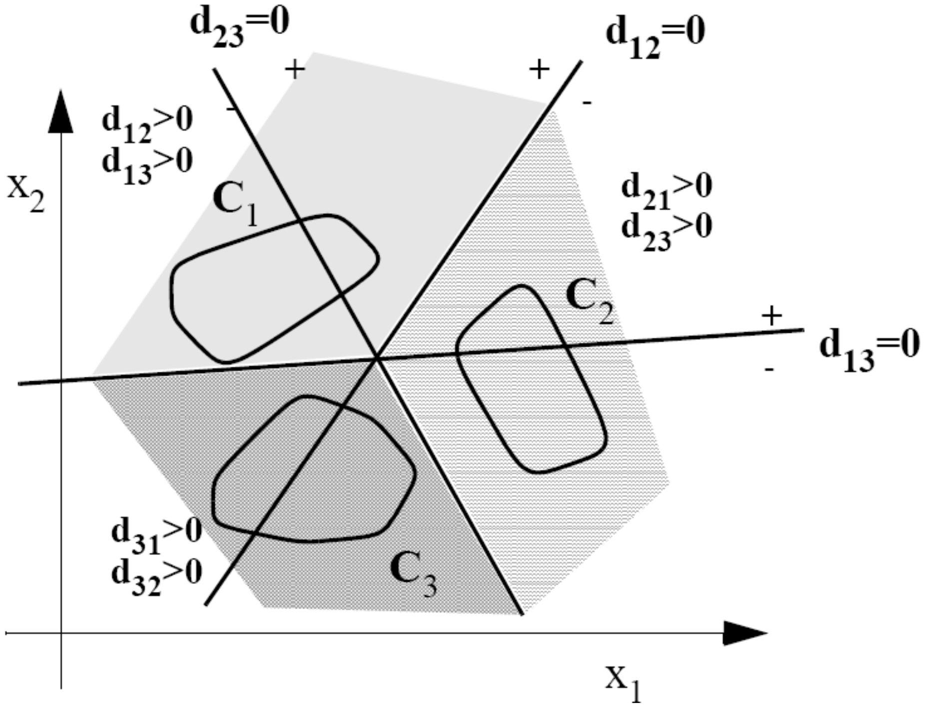


Linear discriminant functions

More than two classes (m classes):

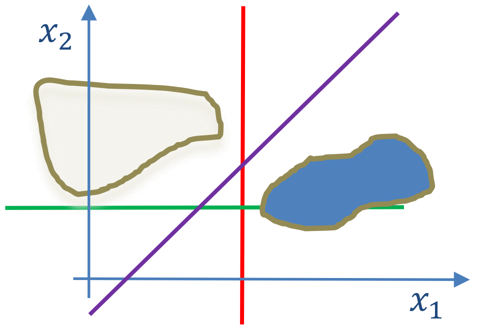
The border (linear) function separating the classes C_i y C_j is computed as:

$$d_{ij}(\mathbf{x}) = d_i(\mathbf{x}) - d_j(\mathbf{x}) = \left(\mathbf{w}_i^{\mathrm{T}} - \mathbf{w}_j^{\mathrm{T}}\right) \cdot \mathbf{x} = \mathbf{w}_{ij}^{T} \cdot \mathbf{x} \begin{cases} > \mathbf{0} & d_i(\mathbf{x}) > d_j(\mathbf{x}),\ \mathbf{x} \notin C_j \\ = \mathbf{0} & the\ frontier \\ < \mathbf{0} & d_j(\mathbf{x}) > d_j(\mathbf{x}),\ \mathbf{x} \notin C_i \end{cases}$$

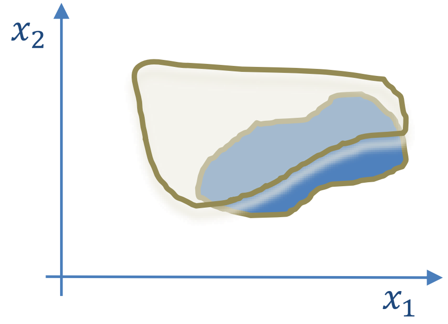


2. Discriminant functions

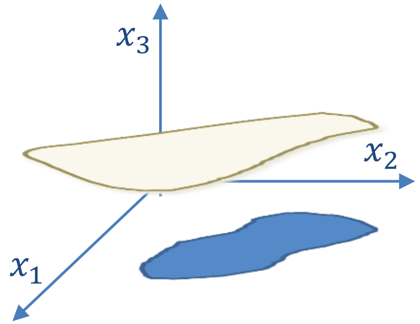
SEPARABILITY:

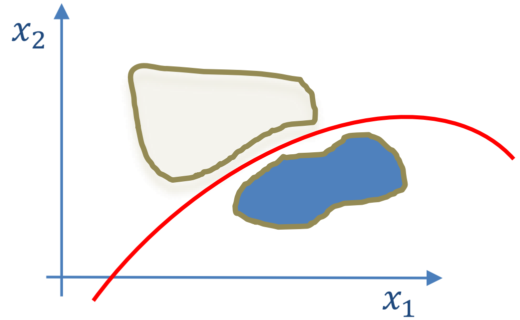


- Classes separable just with x_1
- Classes DO NOT separable with x_2
- x_1 more discriminative than x_2
- In the x_1-x_2 space, classes are more clearly separables (x_2 helps)



- Classes are not separable in x_1-x_2
- The feature x_1 provides nothing! (No discriminative at all)
- Can be separated with an additional feature x_3



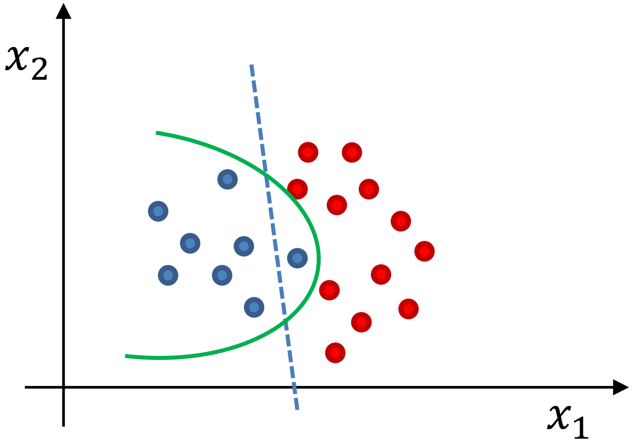


- Clases NOT-linearly separables in x_1-x_2
- Can be separated with a **Not-linear** discriminant function

2. Discriminant functions

Linear Basis Function Models (also called *generalized discriminant functions*)

Needed when the classes are better separated with not non-linear functions:



How: Through a transformation from the $\mathbf{x}$ space to a $\mathbf{x}'$=f($\mathbf{x}$) space (dim($\mathbf{x}$)<dim($\mathbf{x}'$)):

New space

$$d(\mathbf{x}) = w_1 \cdot f_1(\mathbf{x}) + w_2 \cdot f_2(\mathbf{x}) + \ldots + w_k \cdot f_k(\mathbf{x}) + w_{k+1} = \sum_{i=1}^{k+1} w_i \cdot \boxed{f_i(\mathbf{x})} = \mathbf{w}^T \cdot \boldsymbol{f}(\mathbf{x})$$

$f_{k+1}(\mathbf{x}) = 1$

$f_i(\mathbf{x}) = x_i{}'$

$$= w_1 \cdot x'_1 + w_2 \cdot x'_2 + \ldots + w_k \cdot x'_k + w_{k+1} = \sum_{i=1}^{k+1} w_i \cdot x'_i = \mathbf{w}^T \cdot \boxed{\mathbf{x}'} = d'(\mathbf{x}')$$

2. Discriminant functions

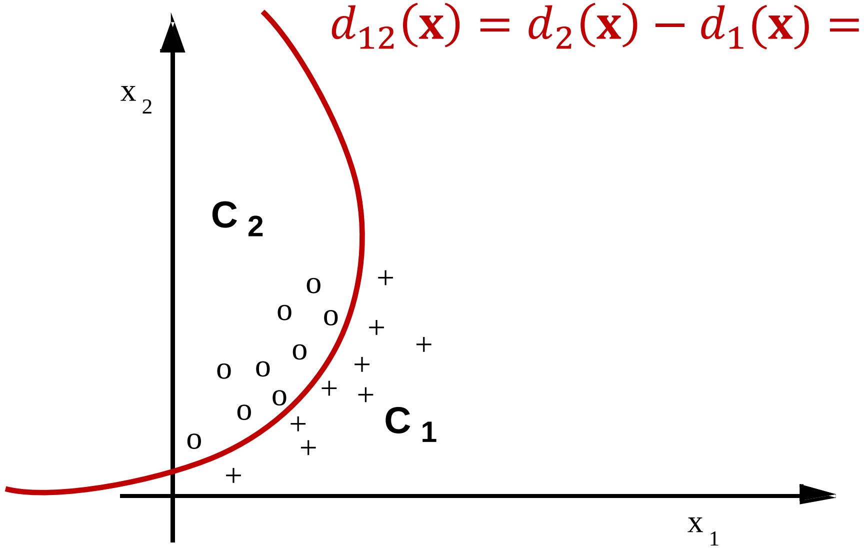
Linear Basis Function Models

$$\left.\begin{array}{r} \mathbf{x}' = \boldsymbol{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \cdots \quad f_k(\mathbf{x}) \quad 1]^T \\ \mathbf{w}^{\boldsymbol{i}} = [w_1{}^i \quad w_2{}^i \quad \ldots \quad w_k{}^i \quad w_{k+1}{}^i]^T \end{array}\right\} \; d_i(\mathbf{x}) = \mathbf{w}^{\boldsymbol{i}^T}\,\mathbf{x}' = \mathbf{w}^{\boldsymbol{i}^T}\,\boldsymbol{f}(\mathbf{x})$$

Basis functions

<u>EXAMPLE</u> : Quadratic function (n=2, k=5)

$$d_{12}(\mathbf{x}) = d_2(\mathbf{x}) - d_1(\mathbf{x}) = w_1 x_1^2 - w_5 x_2 = 0$$



$$\mathbf{x}' = [x_1^2 \quad \overbrace{x_1 x_2 \quad x_2^2 \quad x_1}^{w_2, w_3, w_4 = 0} \quad x_2 \quad 1]^T$$

$f_1(\mathbf{x}) = x'_1$

$f_2(\mathbf{x}) = x'_2$

$f_3(\mathbf{x}) = x'_3$

$f_2(\mathbf{x}) = x'_2$

$f_5(\mathbf{x}) = x'_5$

RECALL: Linear and quadratic functions

- **Linear function**:

$$\mathrm{f}(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} = \left(\mathbf{w}^{\mathrm{T}}\mathbf{x}\right)^{\mathrm{T}} = \mathbf{x}^{\mathrm{T}}\mathbf{w} \quad \text{(Dot product of vectors)}$$

Special case: *Square Euclidean distance* (o square 2-norm)

$$\mathbf{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{x} = \sum_{\mathbf{i}} \mathrm{x}_{\mathbf{i}}^{\mathbf{2}} = \|\mathbf{x}\|_{\mathbf{2}}^{\mathbf{2}}$$

- **Quadratic function**:

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} = \sum_{\mathbf{i}} \mathrm{x_i x_j q_{ij}} \quad \text{with } \mathbf{Q} = \left[\mathrm{q_{ij}}\right]_{\mathbf{nxn}}$$

Special cases:

- $\mathbf{Q}$ symmetric ($\mathbf{Q}^{\mathbf{T}} = \mathbf{Q}$) and positive-semidefinive ($\mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} \geq \mathbf{0}, \forall\mathbf{x}$)

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} = \mathbf{0} \quad \text{(equation of an ellipse)}$$

- Beside, if $\mathbf{Q}$ diagonal ($\mathrm{q_{ij}} = 0$ if $\mathrm{i} \neq \mathrm{j}$)

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x} = \sum_{\mathbf{i}} \mathrm{x_{ii}^2 q_{ii}} = 0 \quad \text{(equation of an ellipse aligned with the axes } \mathrm{x}_{\boldsymbol{i}} \text{)}$$

3. Bayesian Classifier

Principle: assign $\mathbf{x}$ to the class C_i that has the highest posterior probability

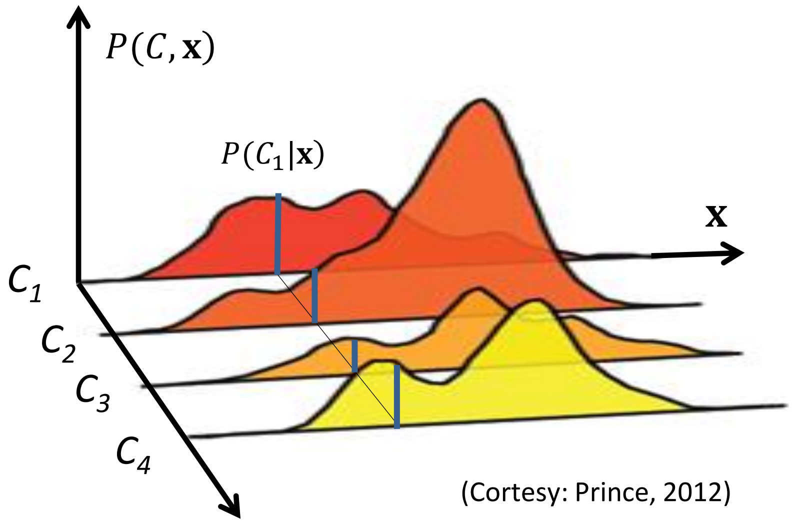
Why: the more probable the less probability of making a mistake

Example: 4 classes, $\mathbf{x}$ belongs to class C_2. Making an error means assigning it to C_1 or C_3

$$P(error|\mathbf{x}) = P(C_1|\mathbf{x}) + P(C_3|\mathbf{x}) + P(C_4|\mathbf{x}) = 1 - P(C_2|\mathbf{x})$$

Recall: $\sum_{k=1}^{M\ classes} P(C_k|\mathbf{x}) = 1$

Minimize $P(error|\mathbf{x})$ → assign $\mathbf{x}$ to the class with the highest probability



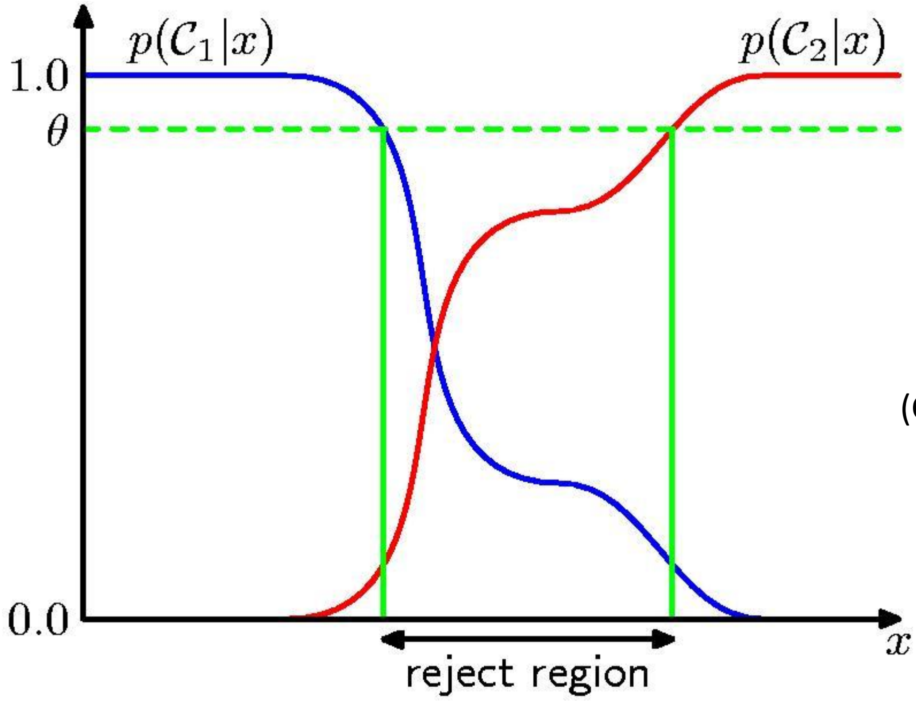
(Cortesy: Prince, 2012)

In this example, given a feature vector $\mathbf{x} = \mathbf{x_1}$, it must be assigned to C_1, since $P(C_1|\mathbf{x}))$ is the highest value.

This is called: a **MAP** prediction
(**M**aximum **A** **P**osteriori)

3. Bayesian Classifier

Sometimes, it is convenient to have a *reject region*, where no decision is made



(Cortesy of Bishop, 2006)

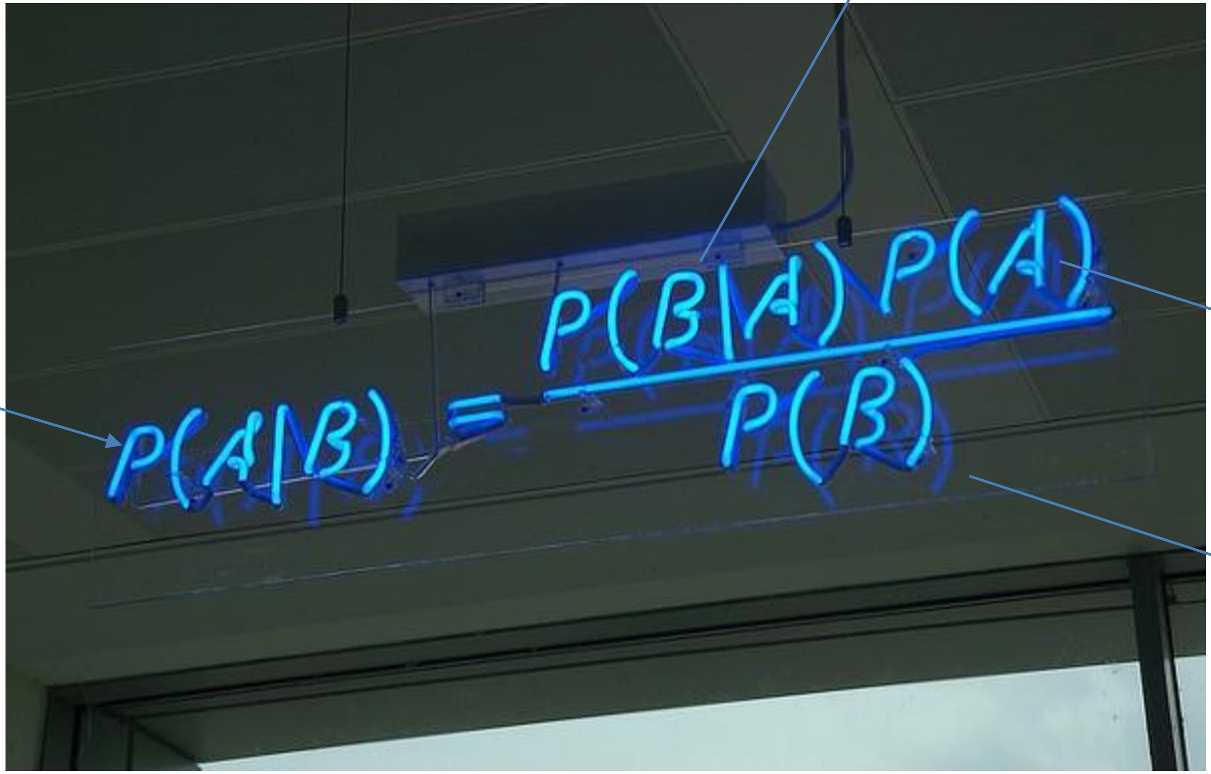
Reject region: None probability is high enough (above a certain posterior probability θ)

3. Bayesian Classifier

RECALL: Bayes theorem

Likelihood (function of A)

A posteriori
(B given)



A priori

Total
probability

3. Bayesian Classifier

We can build a classifier based on discriminant functions for a Bayesian classifier

How: Create a discriminant function d_i (**x**) for each class C_i, such that $d_i(\mathbf{x}) > d_j(\mathbf{x})$ whenever $P(C_i|\mathbf{x}) > P(C_j|\mathbf{x})$

Then:

$$d_i(\mathbf{x}) = P(C_i/\mathbf{x}) = \frac{p(\mathbf{x}/C_i)P(C_i)}{P(\mathbf{x})}$$

MAP estimation
(**M**aximum **A P**osteriori)

$P(\mathbf{x})$ is a scale factor for all the classes C_i

$$d_i(\mathbf{x}) = p(\mathbf{x}/C_i)P(C_i)$$

Max $\ln(\mathrm{f}(\mathrm{x}))$ → Max $\mathrm{f}(\mathrm{x})$

$$d_i(\mathbf{x}) = \ln p(\mathbf{x}/C_i) + \ln P(C_i)$$

If $P(C_i) = P(C_j) \quad \forall i, j$

$$d_i(\mathbf{x}) = \ln p(\mathbf{x}/C_i)$$

Maximum Log-Likelihood estimation

3. Bayesian Classifier

BINOMIAL DISTRIBUTION:

$$\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_f \quad \cdots \quad x_n]^T$$

Features are either 0 or 1 (Bernoulli trial)

$$p(x_f/C_i) = {p_f^i}^{x_f} \cdot (1 - p_f^i)^{(1-x_f)}$$

p_f^i probability that $x_f = 1$ if $\mathbf{x} \in C_i$

$1 - p_f^i$ probability that $x_f = 0$ if $\mathbf{x} \in C_i$

If features independent
(Naïve Bayes Classifier)

$$p(\mathbf{x}/C_i) = \prod_{f=1}^{n} p(x_f/C_i) = \prod_{f=1}^{n} {p_f^i}^{x_f} \cdot (1 - p_f^i)^{(1-x_f)}$$

$$\ln p(\mathbf{x}/C_i) = \ln \prod_{f=1}^{n} {p_f^i}^{x_f} \cdot (1 - p_f^i)^{(1-x_f)} = \sum_{f=1}^{n} [x_f \cdot \ln p_f^i + (1 - x_f) \cdot \ln(1 - p_f^i)] = \boxed{\sum_{f=1}^{n} x_f \cdot \ln \frac{p_f^i}{1-p_f^i} + \sum_{f=1}^{n} \ln(1 - p_f^i)}$$

$$d_i(\mathbf{x}) = \ln P(C_i) + \ln p(\mathbf{x}/C_i) = \underbrace{\ln P(C_i) + \sum_{f=1}^{n} \ln(1 - p_f^i)}_{w_{n+1}^i} + \sum_{f=1}^{n} x_f \cdot \ln \frac{p_f^i}{1 - p_f^i} = w_{n+1}^i + \sum_{f=1}^{n} {w_f}^i \cdot x_f$$

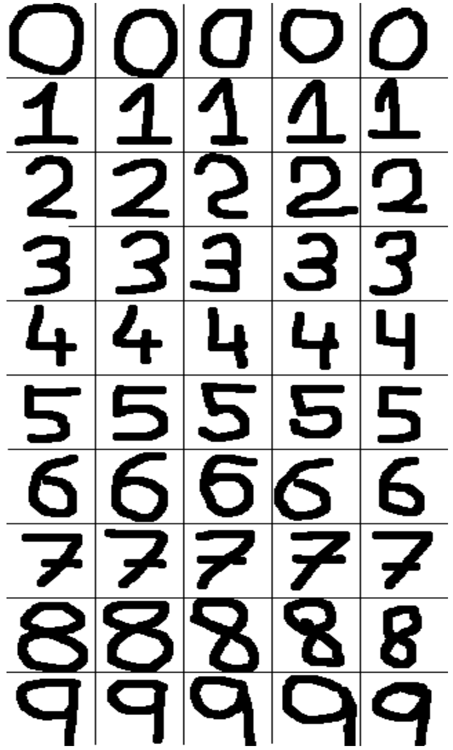
Linear function!

WATCH OUT: we can not use p_f^i exactly equals to 0 or 1 because we would have numerical problems. Instead, take values close to 0 or 1.

3. Bayesian Classifier

EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Learning the classifier: Estimate the probabilities p_f^i



Input: many binay images with the numbers handwriten

- The numbers are segmented and binarized.
- The bounding box around the segmented number is resampled to have a fixed size of 16 x 16 pixels
- The bounding box image is rearanged in a vector of 16x16= 256 elements

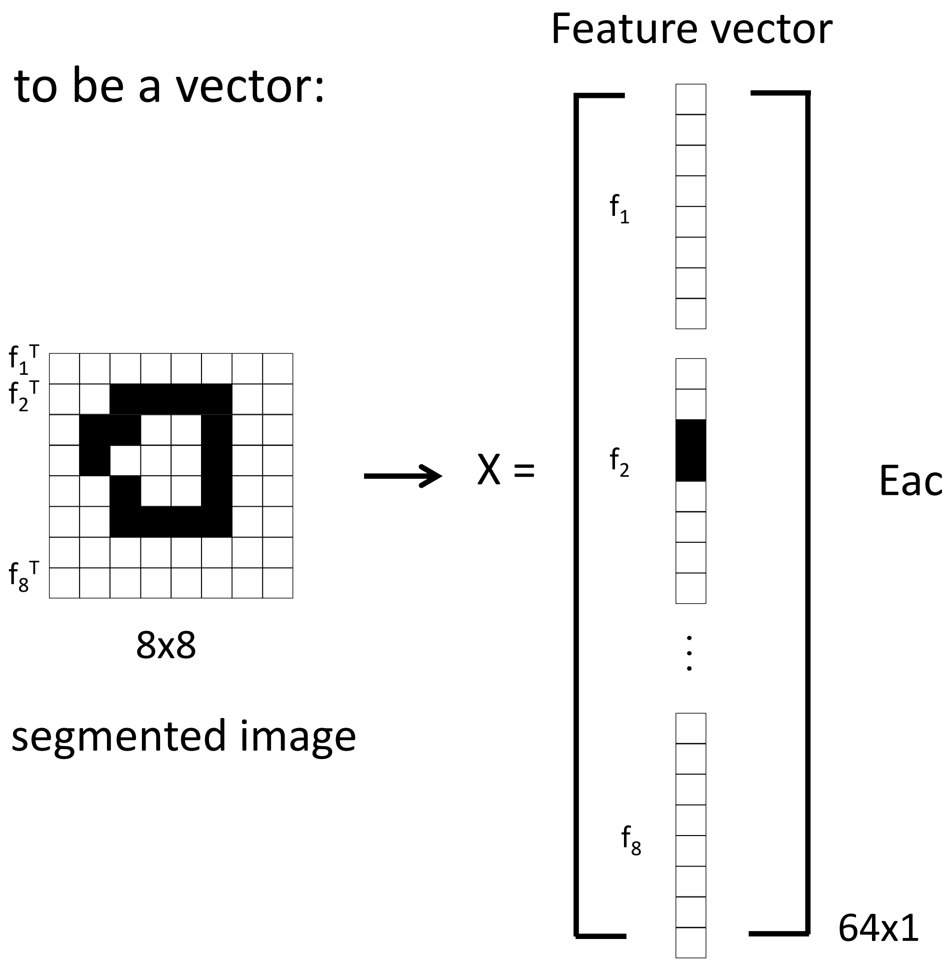
Output: 10 discriminant functions

[More detail next]

EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Learning the classifier

Reshape the segmented image to be a vector:

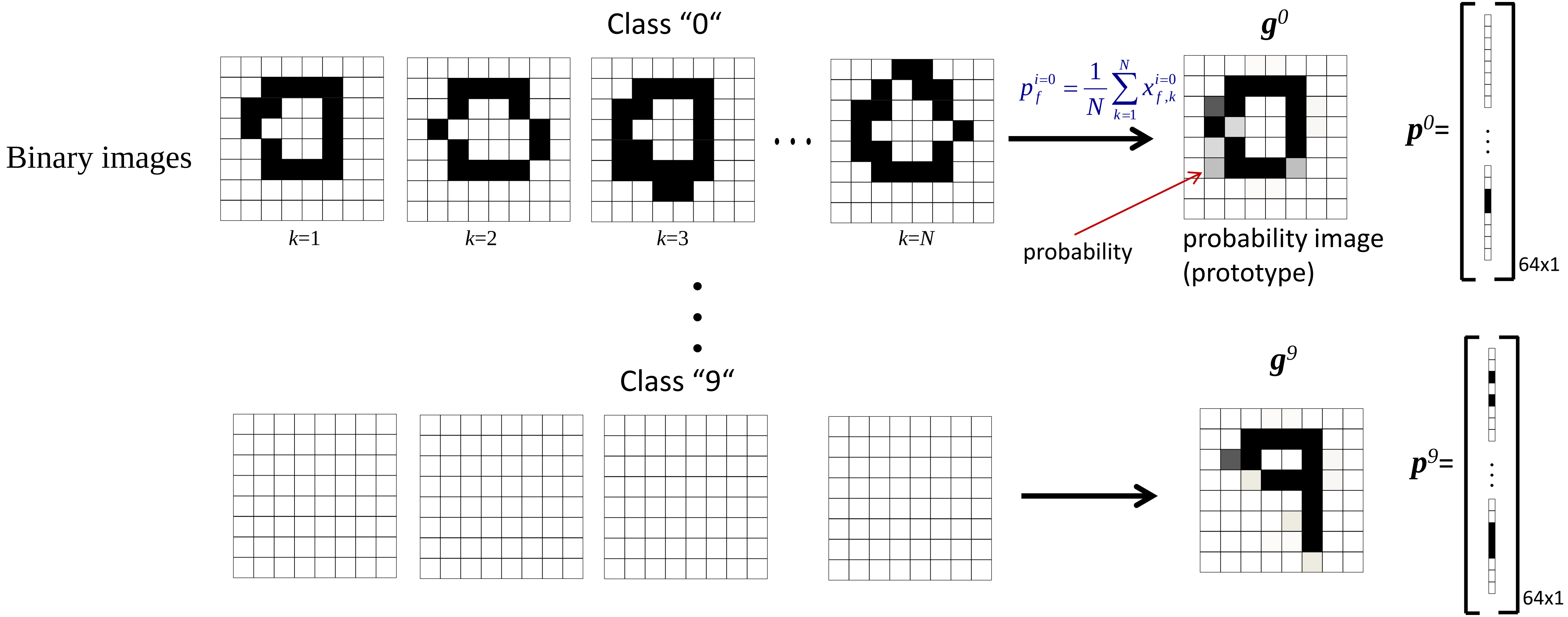


Each element equals 0 or 1

EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Learning the classifier

Computation of the p_f for each class i=0, 9



EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Learning the classifier

Prototype images $\mathbf{g^i}$: The value of a pixel $g^i(i,j)$ represents the probability of pixel (i,j) of being "1" (black on a white paper sheet)

$$g^i(i,j) = p_f = \frac{1}{N}\sum_{k=1}^{N} x_{f,k}^i$$

N is the number of training samples (given to the recognizer)

EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Learning the classifier

Computation of the discriminant functions for each class $i = 0, \ldots, 0$

Compute the 256 weights (f) for each class $i = 10 \;\ldots, 9$

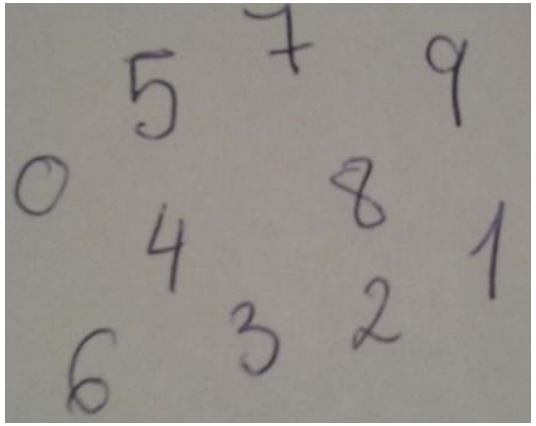
$$= {w_f}^i \ln \frac{p_f^i}{1 - p_f^i}$$

$${w_{n+1}}^i = \ln P(C_k) + \sum_{f=1}^{n} \ln\left(1 - p_f^i\right)$$

EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Prediction phase:

Input: Image of handwritten numbers (not rotated)



- The numbers are segmented and binarized.
- The bounding box around the segmented number is resampled to have a fixed size of 16 x 16 pixels
- The bounding box image is rearanged in a vector of f= 16x16 = 256 elements

EXAMPLE BINOMIAL DISTRIBUTION . Number recognition

Prediction phase:

Evaluate the 10 discriminant functions $d_i(\mathbf{x})$ $(i = 0, \ldots, 9)$ for each vector $\mathbf{x}$

$$d_i(\mathbf{x}) = {w_{n+1}}^i + \sum_{f=1}^{256} {w_f}^i \cdot x_f$$

Output:

- class C_i with the highest $d_i(\mathbf{x})$
- Probability of $\mathbf{x}$ to belong to any class C_i

$$p(C_i/\mathbf{x}) = \eta p(\mathbf{x}/C_i)P(C_i)$$

with $\eta \sum_{i=0}^{9} p(\mathbf{x}/C_i)P(C_i) = 1 \quad \Rightarrow \quad \eta = 1 \Big/ \sum_{i=0}^{9} p(\mathbf{x}/C_i)P(C_i)$

3. Bayesian Classifier

GAUSSIAN DISTRIBUTION:

Feature vector of dimension *n*: $\mathbf{x} = [x_1 \quad x_2 \quad \cdots \quad x_f \quad \cdots \quad x_n]^T$

Features can take continuous values following the probability density function (pdf):

$$p(\mathbf{x}/C_i) = \frac{1}{(2\pi)^{n/2}|\Sigma^i|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}^i)^T {\Sigma^i}^{-1}(\mathbf{x}-\boldsymbol{\mu}^i)}$$

Given by two set of parameters:

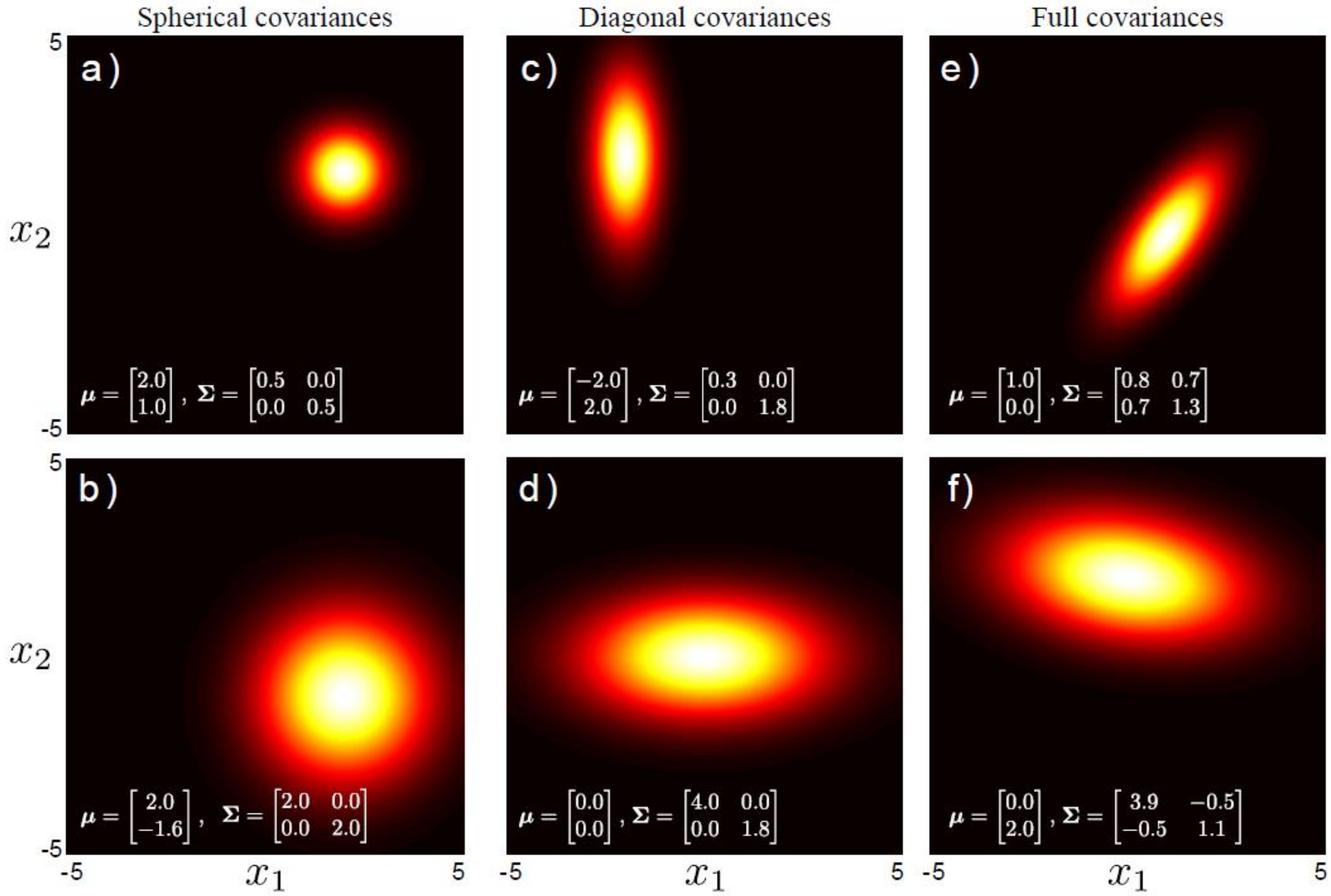
- **Mean vector:** $\boldsymbol{\mu} = [\mu_1 \quad \mu_2 \quad \cdots \quad \mu_f \quad \cdots \quad \mu_n]^T$
- **Covariance matrix:**

$$\Sigma = E[(\mathbf{x}-\boldsymbol{\mu}) \cdot (\mathbf{x}-\boldsymbol{\mu})^T] = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1f} & \cdots & \sigma_{1n} \\ \vdots & & \vdots & & \vdots \\ \sigma_{n1} & \cdots & \sigma_{nf} & \cdots & \sigma_{nn} \end{bmatrix} = \begin{bmatrix} \sigma_{11} & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & \sigma_{nn} \end{bmatrix}$$

Features are independent (no correlated)
(Naïve Bayes Classifier)

GAUSSIAN DISTRIBUTION:

RECALL: How does a Gaussian depend on its two parameters μ y Σ



(Cortesía: Prince, 2012)

3. Bayesian Classifier

GAUSSIAN DISTRIBUTION:

$$d_k(\mathbf{x}) = \ln P\,(C_k) + \ln p\,(\mathbf{x}/C_k) = \ln P\,(C_k) + \ln \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma^k|^{\frac{1}{2}}} e^{-\frac{1}{2}\overbrace{(\mathbf{x}-\boldsymbol{\mu}^k)^T(\Sigma^k)^{-1}(\mathbf{x}-\boldsymbol{\mu}^k)}^{\text{Squared Mahalanobis distance } D_k^2(\mathbf{x})}} =$$

$$= \ln P\,(C_k) + \ln \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma^k|^{\frac{1}{2}}} - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}^k)^T(\Sigma^k)^{-1}(\mathbf{x}-\boldsymbol{\mu}^k)$$

$$= \ln P\,(C_k) - \frac{1}{2}\left[n\ln(2\pi) + \ln|\Sigma^k| + D_k^2(\mathbf{x})\right]$$

Constant → can be removed

$$d_k(\mathbf{x}) = \underbrace{\ln P\,(C_k) - \frac{1}{2}\left[\ln|\Sigma^k| + {\boldsymbol{\mu}^k}^T(\Sigma^k)^{-1}\boldsymbol{\mu}^k\right]}_{\text{Independent term}} + \mathbf{x}^T\underbrace{(\Sigma^k)^{-1}\boldsymbol{\mu}^k}_{\text{Linear weights}} - \frac{1}{2}\mathbf{x}^T\underbrace{(\Sigma^k)^{-1}}_{\text{Quadratic weights}}\mathbf{x}$$

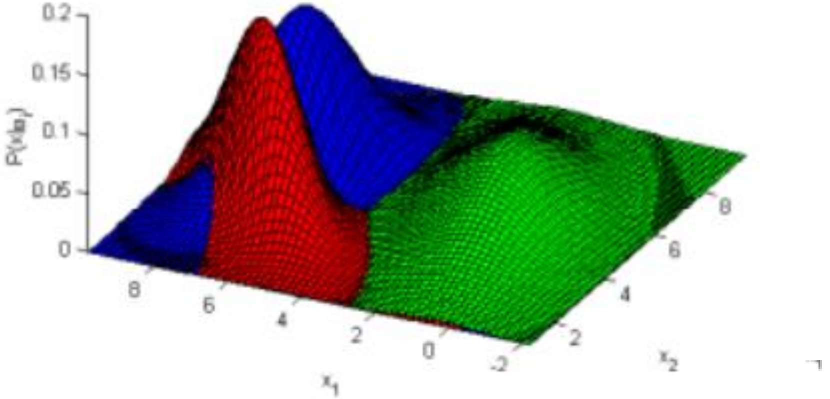
The discriminant function is a quadratic polynomial! $d_k(\mathbf{x}) = w_{\mathbf{n+1}} + \mathbf{x}^T\mathbf{w} + \mathbf{x}^T\boldsymbol{Q}\mathbf{x}$

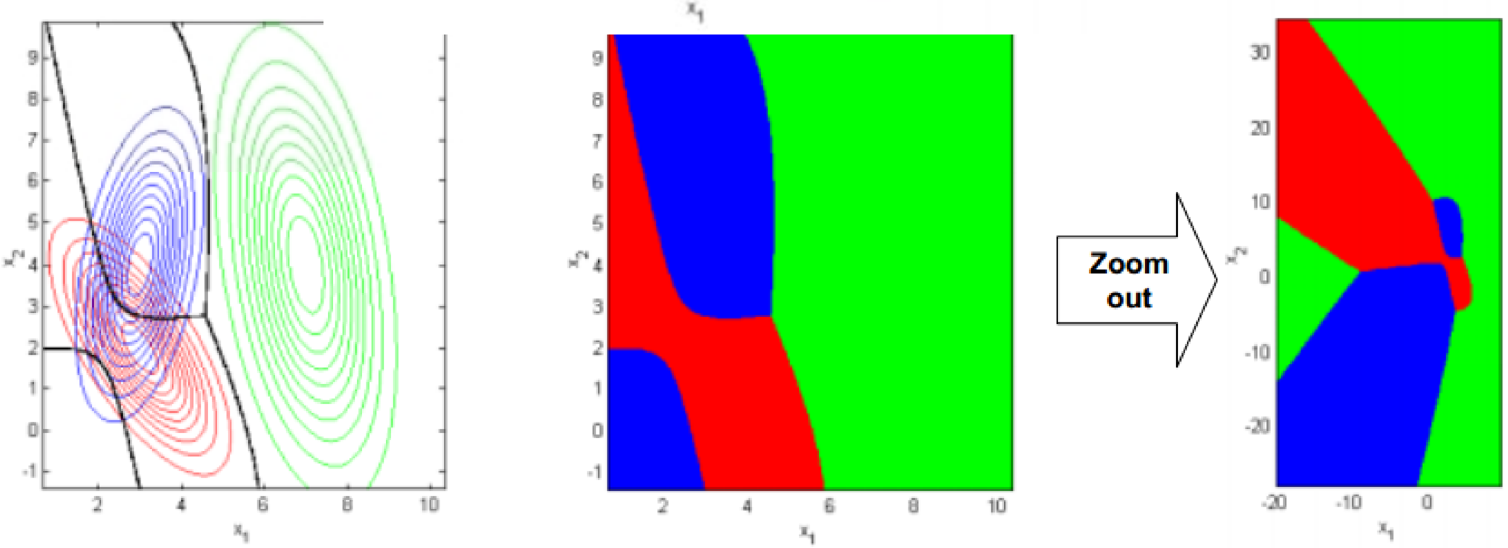
GAUSSIAN DISTRIBUTION :

Visually:

$$d_k(\mathbf{x}) = \ln P(C_k) - \frac{1}{2}\ln\left|\Sigma^k\right| - \left[\left(\mathbf{x}-\boldsymbol{\mu}^k\right)^T\left(\Sigma^k\right)^{-1}\left(\mathbf{x}-\boldsymbol{\mu}^k\right)\right]$$

$D_k^2(x)$ Squared Mahalanobis distance





NORMAL DISTRIBUTION:

SIMPLIFICATIONS

Equal priors: $P(C_1) = P(C_2) = \cdots = P(C_m) = P(C)$

Same covariance matrices: $\Sigma^1 = \Sigma^2 = \cdots = \Sigma^m = \Sigma$

$$d_k(\mathbf{x}) = \ln P(C_k) - \frac{1}{2}\left[n\ln 2\pi + \ln|\Sigma^k| + \mathbf{x}^T {\Sigma^k}^{-1}\mathbf{x} - 2\mathbf{x}^T {\Sigma^k}^{-1}\boldsymbol{\mu}^k + {\boldsymbol{\mu}^k}^T {\Sigma^k}^{-1}\boldsymbol{\mu}^k\right]$$

$$= \ln P(C) - \frac{1}{2}\left[n\ln 2\pi + \ln|\Sigma| + \mathbf{x}^T \Sigma^{-1}\mathbf{x} - 2\mathbf{x}^T \Sigma^{-1}\boldsymbol{\mu}^k + {\boldsymbol{\mu}^k}^T \Sigma^{-1}\boldsymbol{\mu}^k\right]$$

Constant → can be removed

$$\boxed{d_k(\mathbf{x}) = -\left(\mathbf{x} - \boldsymbol{\mu}^k\right)^T \Sigma^{-1}\left(\mathbf{x} - \boldsymbol{\mu}^k\right) = -{D_k}^2(\mathbf{x})}$$

Classifier based on Square Mahalanobis distance

Notice: $\mathbf{x}^T \Sigma^{-1}\mathbf{x}$ is a constant term for all the classes → can be removed:

$$d_k(\mathbf{x}) = -2\mathbf{x}^T \Sigma^{-1}\boldsymbol{\mu}^k + {\boldsymbol{\mu}^k}^T \Sigma^{-1}\boldsymbol{\mu}^k = w_{\mathbf{n+1}} + \mathbf{x}^T\mathbf{w}$$

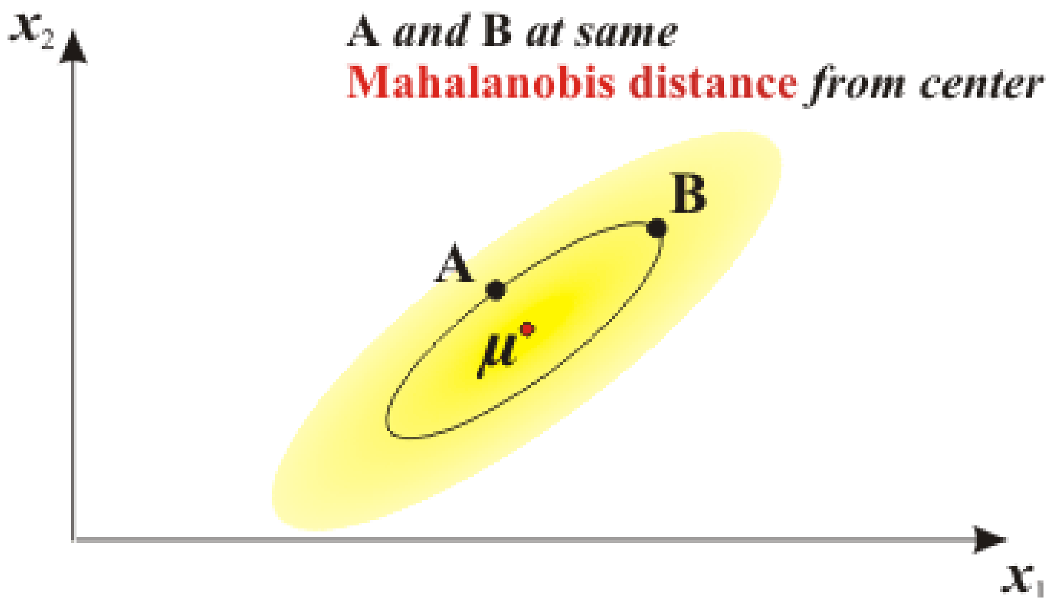
The discriminat functions are Linear!

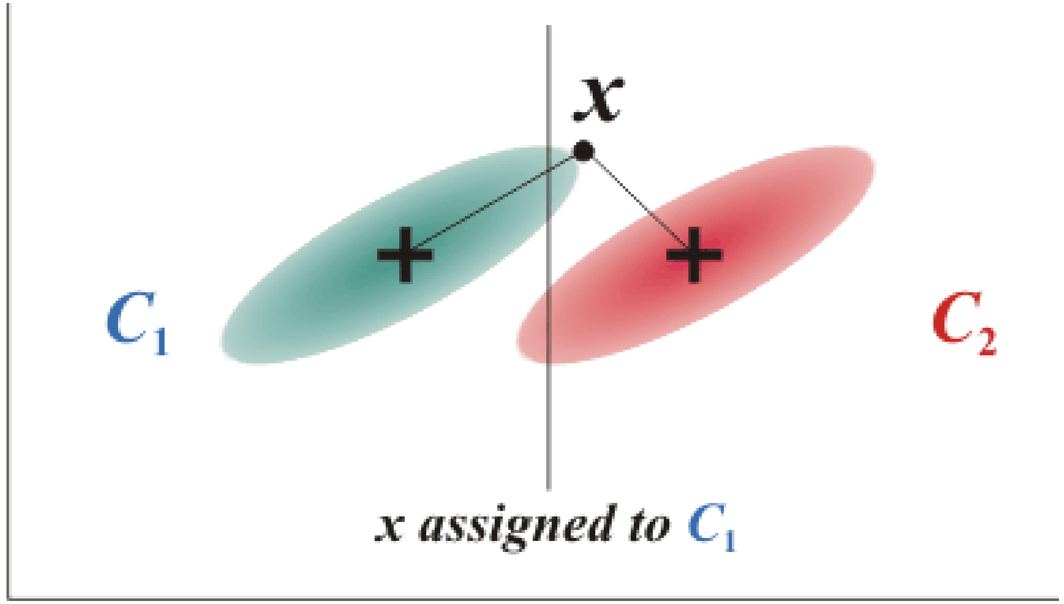
Classifier based on Square Mahalanobis distance

$$d_k(\mathbf{x}) = -\left(\mathbf{x} - \boldsymbol{\mu}^k\right)^T \Sigma^{-1}\left(\mathbf{x} - \boldsymbol{\mu}^k\right) = -{D_k}^2(\mathbf{x})$$

Recall: Quadratic function:

$$\boldsymbol{f}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}.\mathbf{Q}.\mathbf{x} = \sum_{\mathbf{i}} \; \mathrm{x_i x_j q_{ij}} \quad \text{with } \mathbf{Q} = \left[\mathrm{q_{ij}}\right]_{\mathbf{nxn}}$$





$\mathbf{x}$ is assigned to C_1 though the closest centroid is C_2

NORMAL DISTRIBUTION:

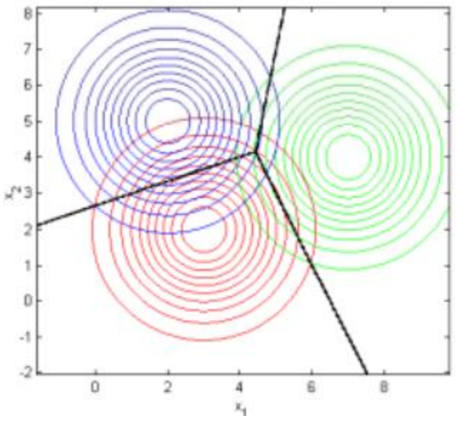
MORE SIMPLIFICATIONS

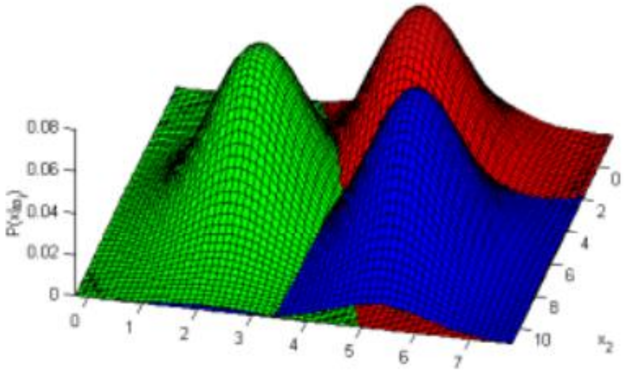
Isotropic covariance matrix: $\Sigma^k = \Sigma = \sigma^2 \cdot \mathbf{I} = \sigma^2 \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$

$$d_k(\mathbf{x}) = -\left(\mathbf{x}-\boldsymbol{\mu}^k\right)^T \Sigma^{-1}\left(\mathbf{x}-\boldsymbol{\mu}^k\right) = -\frac{1}{\sigma^2}\left(\mathbf{x}-\boldsymbol{\mu}^k\right)^T \mathbf{I}\left(\mathbf{x}-\boldsymbol{\mu}^k\right) = \underbrace{-\left(\mathbf{x}-\boldsymbol{\mu}^k\right)^T\left(\mathbf{x}-\boldsymbol{\mu}^k\right)}_{-\left\|\mathbf{x}-\boldsymbol{\mu}^k\right\|^2} = -D_k^2(\mathbf{x})$$

Linear classifier!

EUCLIDEAN distance





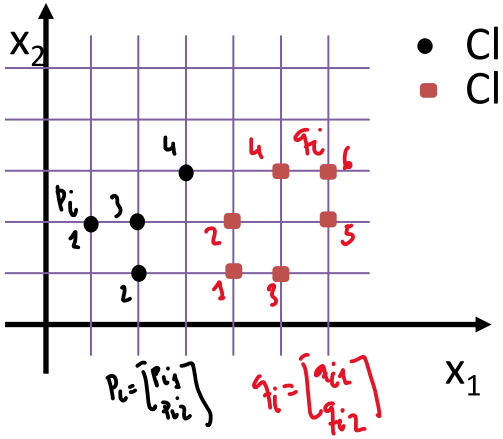
DECISION RULE:

assign $\mathbf{x}$ *to the class* C_k *such that:*

$$d_k(\mathbf{x}) > d_j(\mathbf{x}) \quad \forall k \neq j$$

It's called the NATURAL classifier

EXAMPLE: Design a classifier based on mínimum Mahalanobis distance for the following example.



- Class 1
- Class 2

- Draw approximately the two elipses representing the Covariance matrices
- What class do $p_1 = [3,2]^T$ and $p_2 = [4,3]^T$ belong to?

Classifier based on Square Mahalanobis distance $\quad d_k(\mathbf{x}) = -(\mathbf{x}-\boldsymbol{\mu}^k)^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}^k) = -D_k{}^2(\mathbf{x})$

we need the two means μ^k and a single Σ for the two set of data points.

$$\mu^1 = \frac{1}{4}\sum_i p_i = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \qquad \mu^2 = \sum_i q_i = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$$

$$\Sigma = \frac{1}{4}\sum (p_i - \mu^1)(q_i - \mu^1)^T + \frac{1}{6}\sum (q_i - \mu^2)(q_i - \mu^2)^T$$

$$\Sigma = \frac{1}{4}\left\{\begin{pmatrix} -1 \\ 0 \end{pmatrix}(-1\ 0) + \begin{pmatrix} 0 \\ -1 \end{pmatrix}(0\ -1) + \begin{pmatrix} 0 \\ 0 \end{pmatrix}(0\ 0) + \begin{pmatrix} 1 \\ 1 \end{pmatrix}(1\ 1)\right\} + \frac{1}{6}\left\{\begin{pmatrix} -1 \\ -1 \end{pmatrix}(-1\ -1) + \begin{pmatrix} -1 \\ 0 \end{pmatrix}(-1\ 0) + \begin{pmatrix} 0 \\ -1 \end{pmatrix}(0\ -1) + \begin{pmatrix} 0 \\ 1 \end{pmatrix}(0\ 1) + \begin{pmatrix} 1 \\ 0 \end{pmatrix}(1\ 0) + \begin{pmatrix} 1 \\ 1 \end{pmatrix}(1\ 1)\right\} = \frac{1}{4}\begin{bmatrix} 1+0+0+1 & 0+0+0+1 \\ 1 & 0+0+0+1 \end{bmatrix} + \frac{1}{6}\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$$

$$\Sigma = \underbrace{\frac{1}{4}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}_{\Sigma^1} + \underbrace{\frac{1}{3}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}}_{\Sigma^2} = \frac{1}{12}\left(\begin{bmatrix} 6 & 3 \\ 3 & 6 \end{bmatrix} + \begin{bmatrix} 8 & 4 \\ 4 & 8 \end{bmatrix}\right) = \frac{1}{12}\begin{bmatrix} 14 & 7 \\ 7 & 14 \end{bmatrix} = \frac{7}{12}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \rightarrow \Sigma^{-1} = \frac{4}{7}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \xrightarrow{\text{check}} \Sigma\Sigma^{-1} = \frac{7 \cdot 4}{12 \cdot 7}\begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A^{-1} = \frac{1}{|A|}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Both gaussians have the same orientation (+45°) why? ∴ lets compute the eigenvectors

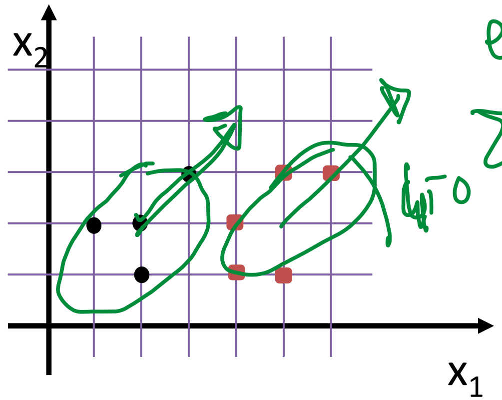
$$|\Sigma - \lambda I| = 0 \rightarrow \begin{vmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{vmatrix} = 0 \rightarrow (2-\lambda)^2 - 1 = 0$$

$$\lambda^2 - 4\lambda + 4 - 1 = 0 \qquad \lambda = \frac{4 \pm \sqrt{16-12}}{2} = 2 \pm \sqrt{1} \quad \lambda_1 = 3,\ \lambda_2 = 1$$

$$\lambda_1 = 3 \rightarrow (\Sigma - \lambda I)u = 0 \qquad \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} v_{1x} \\ v_{1y} \end{bmatrix} = 0 \qquad v = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\theta = \arctan \frac{1}{1} = 1 \Rightarrow 4$$

Recall: $(x_1, x_2)\begin{pmatrix} a & b \\ b & c \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = ax_1^2 + b x_2^2 + 2b x_1 x_2$



Assuming the same Σ for both classes $\quad \Sigma^{-1} = \frac{4}{7}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$

$$d^1(x) = -\left[(x_1-2) \;\; (x_2-2)\right]\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} x_1-2 \\ x_2-2 \end{bmatrix} = -\left[2(x_1-2)^2 + 2(x_2-2) - 2(x_1-2)(x_2-2)\right]$$

$$d^2(x) = -\left[(x_1-5) \;\; (x_2-2)\right]\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1-5 \\ x_2-2 \end{pmatrix} = -\left[2(x_1-5)^2 + 2(x_2-2)^2 - 2(x_1-5)(x_2-2)\right]$$

$$d^1\begin{pmatrix} 3 \\ 2 \end{pmatrix} = -(2+0-2(1)(0)) = -2$$

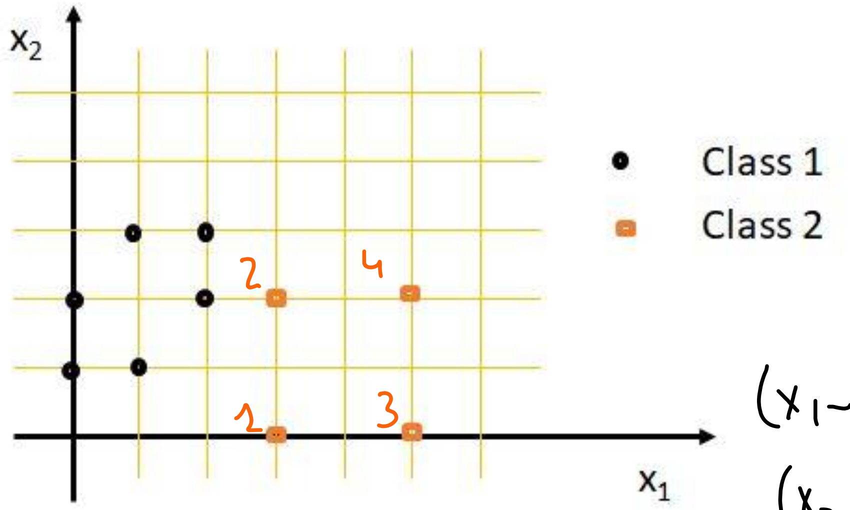
$$d^2\begin{pmatrix} 3 \\ 2 \end{pmatrix} = -(2\cdot 4 + 0 + 0) = -8$$

$d^1 > d^2 \Rightarrow \begin{bmatrix} 3 \\ 2 \end{bmatrix} \in C_1$

Frontier between classes: $d^{12}(x) = d^1 - d^2 = 0$ The quadratic terms (x_1^2, x_2^2, x_1x_2) cancel out $\Rightarrow d^{12}(x) = 0$ is a line at $45°$

Pregunta examen:

1. Compute the value of the feature sector x=[2,0]^T for the discriminant function for Class 2 (d_2). Assume $\ln P\,(C_k) = -0{,}5$
2. Which is the orientation of the Gaussian of the Class 1 .



$$\mu_2 = \frac{1}{4}\left[\begin{pmatrix}3\\0\end{pmatrix}+\begin{pmatrix}3\\2\end{pmatrix}+\begin{pmatrix}5\\0\end{pmatrix}+\begin{pmatrix}5\\2\end{pmatrix}\right] = \begin{bmatrix}4\\1\end{bmatrix}$$

$$\Sigma_2 = \frac{1}{N}\sum_i (x_i-\mu)(x_i-\mu)^T = \frac{1}{4}\begin{bmatrix}4&0\\0&4\end{bmatrix} = \begin{bmatrix}1&0\\0&1\end{bmatrix} \to \Sigma^{-1} = \frac{1}{1}\begin{bmatrix}1&0\\0&1\end{bmatrix}$$

$$(x_1-\mu)(x_1-\mu)^T = \begin{pmatrix}-1\\-1\end{pmatrix}(-1\ \ -1) = \begin{pmatrix}1&1\\1&1\end{pmatrix} \qquad (x_3-\mu)(x_3-\mu)^T = \begin{pmatrix}1\\-1\end{pmatrix}(1\ \ -1) = \begin{pmatrix}1&-1\\-1&1\end{pmatrix}$$

$$(x_2-\mu)(x_2-\mu)^T = \begin{pmatrix}-1\\1\end{pmatrix}(-1\ \ 1) = \begin{pmatrix}1&-1\\-1&1\end{pmatrix} \qquad (x_4-\mu)(x_4-\mu)^T = \begin{pmatrix}1\\1\end{pmatrix}(1\ \ 1) = \begin{pmatrix}1&1\\1&1\end{pmatrix}$$

$$d_k(\mathbf{x}) = \ln P\,(C_k) - \frac{1}{2}\left[\ln|\Sigma^k| + \boldsymbol{\mu}^{k^T}(\Sigma^k)^{-1}\boldsymbol{\mu}^k\right] + \mathbf{x}^T(\Sigma^k)^{-1}\boldsymbol{\mu}^k - \frac{1}{2}\mathbf{x}^T(\Sigma^k)^{-1}\mathbf{x}$$

$$\mu^T\Sigma^{-1}\mu = (4\ \ 1)\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}4\\1\end{pmatrix} = 16+1 = 17$$

$$x^T\Sigma^{-1}\mu = (x_1\ \ x_2)\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}4\\1\end{pmatrix} = 4x_1+x_2$$

$$x^T\Sigma x = x^Tx = x_1^2+x_2^2$$

$$d_2(x) = \underbrace{-0.5-\frac{1}{2}\left[0+17\right]}_{-9} + 4x_1+x_2+x_1^2+x_2^2 = x_1^2+x_2^2+4x_1+x_2-9$$

$$d\begin{bmatrix}2\\0\end{bmatrix} = 4+0+8+0-9 = 3$$