MAPi – Computer Vision 2011/12

Lecture 1b - Classification Concepts

Jaime S. Cardoso
jaime.cardoso@fe.up.pt

INESC Porto, Faculdade Engenharia, Universidade do Porto

Dec. 12, 2011

Features

- P features describing an observation $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_p \end{bmatrix}$ are called a feature vector or input vector
- The set of all possible feature vectors $\mathbb{R}^p$ is called the feature space.

Classifier

- Maps a feature vector into one of K classes

$$\mathbf{x} \longrightarrow \mathcal{C}_i \in \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$$

- The classifier performs a partitioning of the feature space into K disjoint regions such that

$$f(\mathbf{x}) = \begin{cases} \mathcal{C}_1 \text{ if } \mathbf{x} \in R_1 \\ \vdots \\ \mathcal{C}_K \text{ if } \mathbf{x} \in R_K \end{cases}$$

where $\cup_{i=1}^{K} R_i = \mathbb{R}^p$

Bayesian Decision Theory

- Bayesian Decision Theory is a statistical approach that quantifies the tradeoffs between various decisions using probabilities and costs that accompany such decisions.
- Fish sorting example: define $\mathcal{C}$, the type of fish we observe (state of nature), as a random variable where
  - $\mathcal{C} = \mathcal{C}_1$ for sea bass
  - $\mathcal{C} = \mathcal{C}_2$ for salmon
  - $P(\mathcal{C}_1)$ is the a priori probability that the next fish is a sea bass
  - $P(\mathcal{C}_2)$ is the a priori probability that the next fish is a salmon

Prior Probabilities

- Prior probabilities reflect our knowledge of how likely each type of fish will appear before we actually see it.
- How can we choose $P(\mathcal{C}_1)$ and $P(\mathcal{C}_2)$?
  - Set $P(\mathcal{C}_1) = P(\mathcal{C}_2)$ if they are equiprobable (uniform priors).
  - May use different values depending on the fishing area, time of the year, etc.
- Assume there are no other types of fish

$$P(\mathcal{C}_1) + P(\mathcal{C}_2) = 1$$

(exclusivity and exhaustivity)

- In a general classification problem with K classes, prior probabilities reflect prior expectations of observing each class and $\sum_{i=1}^{K} P(\mathcal{C}_i) = 1$

Making a Decision

- How can we make a decision with only the prior information?

  Decide $\begin{cases} \mathcal{C}_1 & \text{if} \quad P(\mathcal{C}_1) > P(\mathcal{C}_2) \\ \mathcal{C}_2 & otherwise \end{cases}$

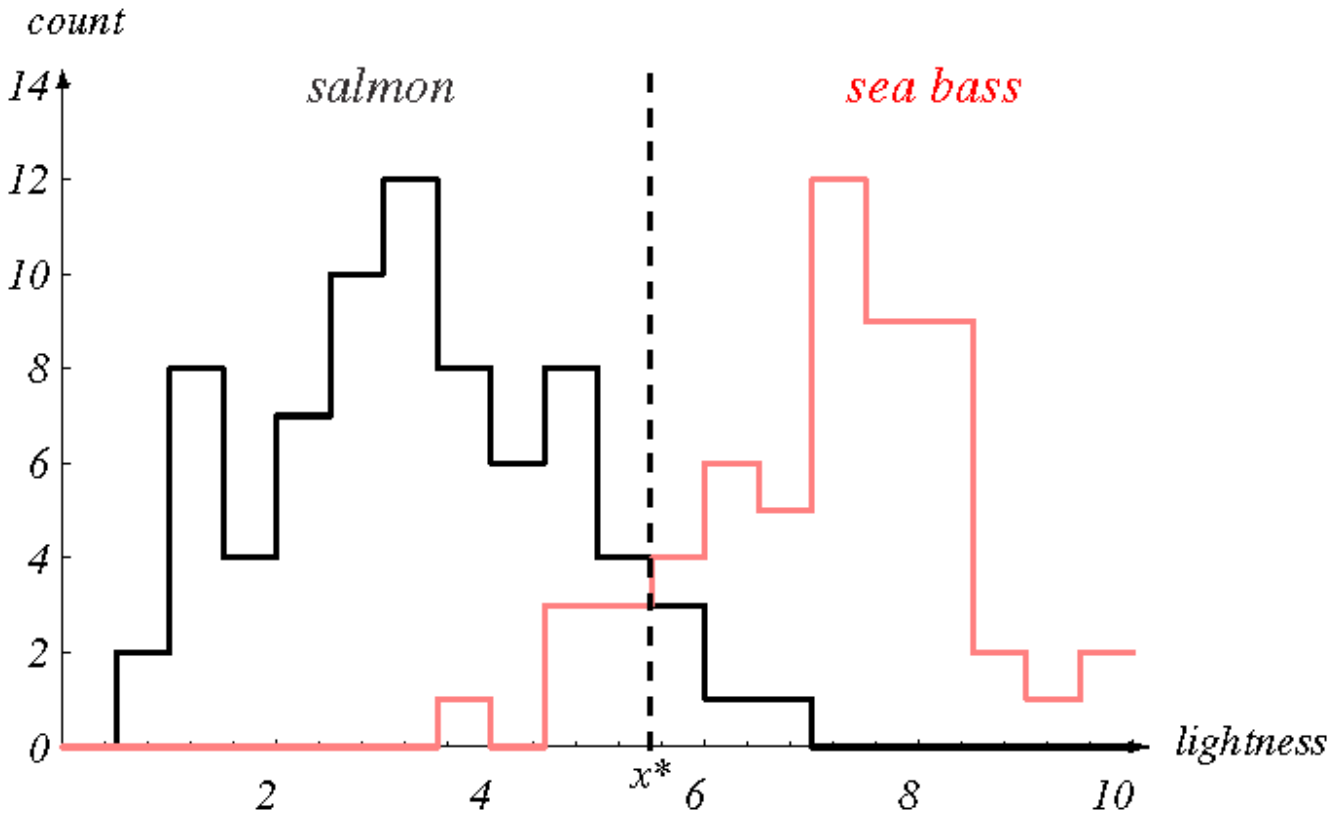
- What is the probability of error for this decision?

$$P(error) = \min\{P(\mathcal{C}_1), P(\mathcal{C}_2)\}$$

Class-conditional Probabilities

- Let's try to improve the decision using the lightness measurement x ($\in \mathbf{R}$).
- Let x be a continuous random variable.
- Define $p(x|\mathcal{C}_j)$ as the class-conditional probability density (probability of x given that the state of nature is $\mathcal{C}_j$ for $j = 1, 2$).
- $p(x|\mathcal{C}_1)$ and $p(x|\mathcal{C}_2)$ describe the difference in lightness between populations of sea bass and salmon.

Class-conditional Probabilities



Posterior Probabilities

- Suppose we know $P(\mathcal{C}_j)$ and $P(x|\mathcal{C}_j)$ for $j = 1, 2$, and measure the lightness of a fish as the value x.
- Define $P(\mathcal{C}_j|x)$ as the a posteriori probability (probability of the state of nature being $\mathcal{C}_j$ given the measurement of feature value x).
- We can use the Bayes formula to convert the prior probability to the posterior probability

$$P(\mathcal{C}_j|x) = \frac{P(x|\mathcal{C}_j)P(\mathcal{C}_j)}{P(x)}$$

where $P(x) = \sum_{j=1}^{2} P(x|\mathcal{C}_j)P(\mathcal{C}_j)$

Making a Decision

- $P(x|\mathcal{C}_j)$ is called the likelihood and $P(x)$ is called the evidence.
- How can we make a decision after observing the value of x?

$$\text{Decide} \begin{cases} \mathcal{C}_1 & \text{if} \quad P(\mathcal{C}_1|x) > P(\mathcal{C}_2|x) \\ \mathcal{C}_2 & otherwise \end{cases}$$

- Rewriting the rule gives

$$\text{Decide} \begin{cases} \mathcal{C}_1 & \text{if} \quad \frac{P(x|\mathcal{C}_1)}{P(x|\mathcal{C}_2)} > \frac{P(\mathcal{C}_2)}{P(\mathcal{C}_1)} \\ \mathcal{C}_2 & otherwise \end{cases}$$

Making a Decision

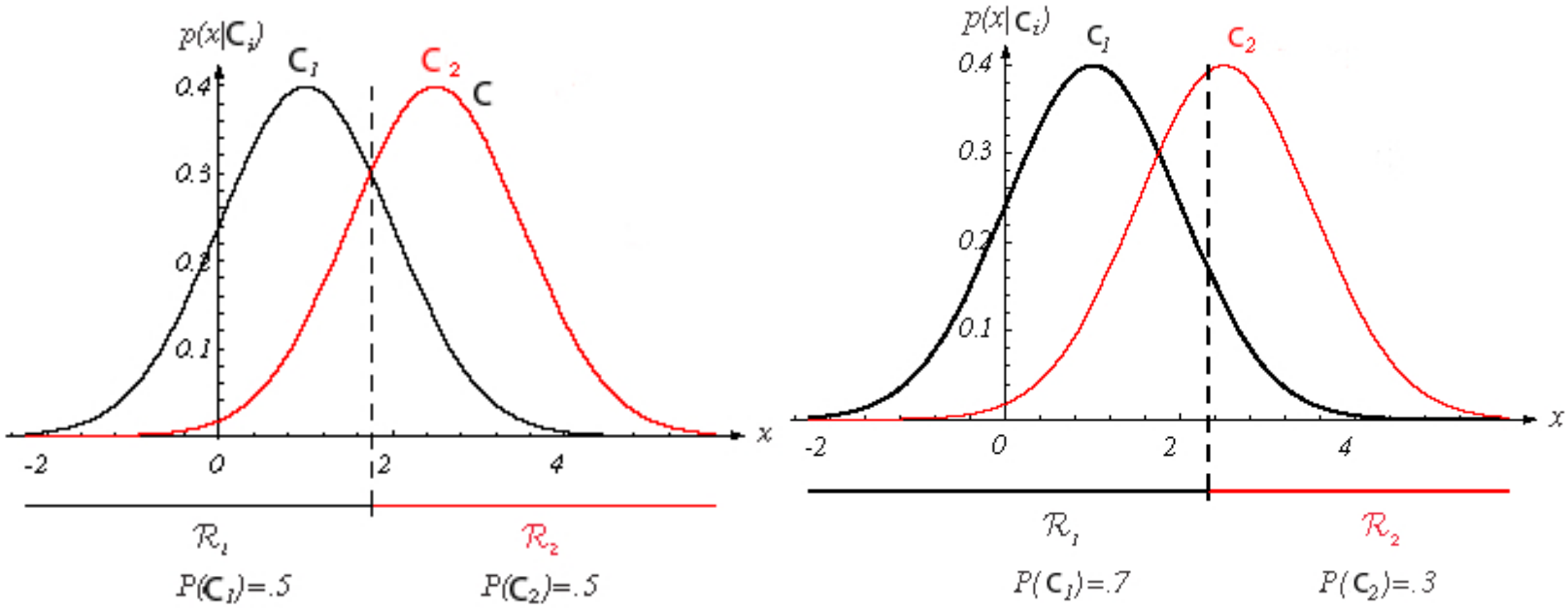


Figure: Optimum thresholds for different priors.

Probability of Error

- What is the probability of error for this decision?

$$P(error|x) = \begin{cases} P(\mathcal{C}_1|x) & \text{if we decide } \mathcal{C}_2 \\ P(\mathcal{C}_2|x) & \text{if we decide } \mathcal{C}_1 \end{cases}$$

- What is the average probability of error?

$$p(error) = \int_{-\infty}^{+\infty} P(error, x)dx = \int_{-\infty}^{+\infty} P(error|x)P(x)dx$$

- Bayes decision rule minimizes this error because

$$P(error|x) = \min\{P(\mathcal{C}_1|x), P(\mathcal{C}_2|x)\}$$

Probability of Error

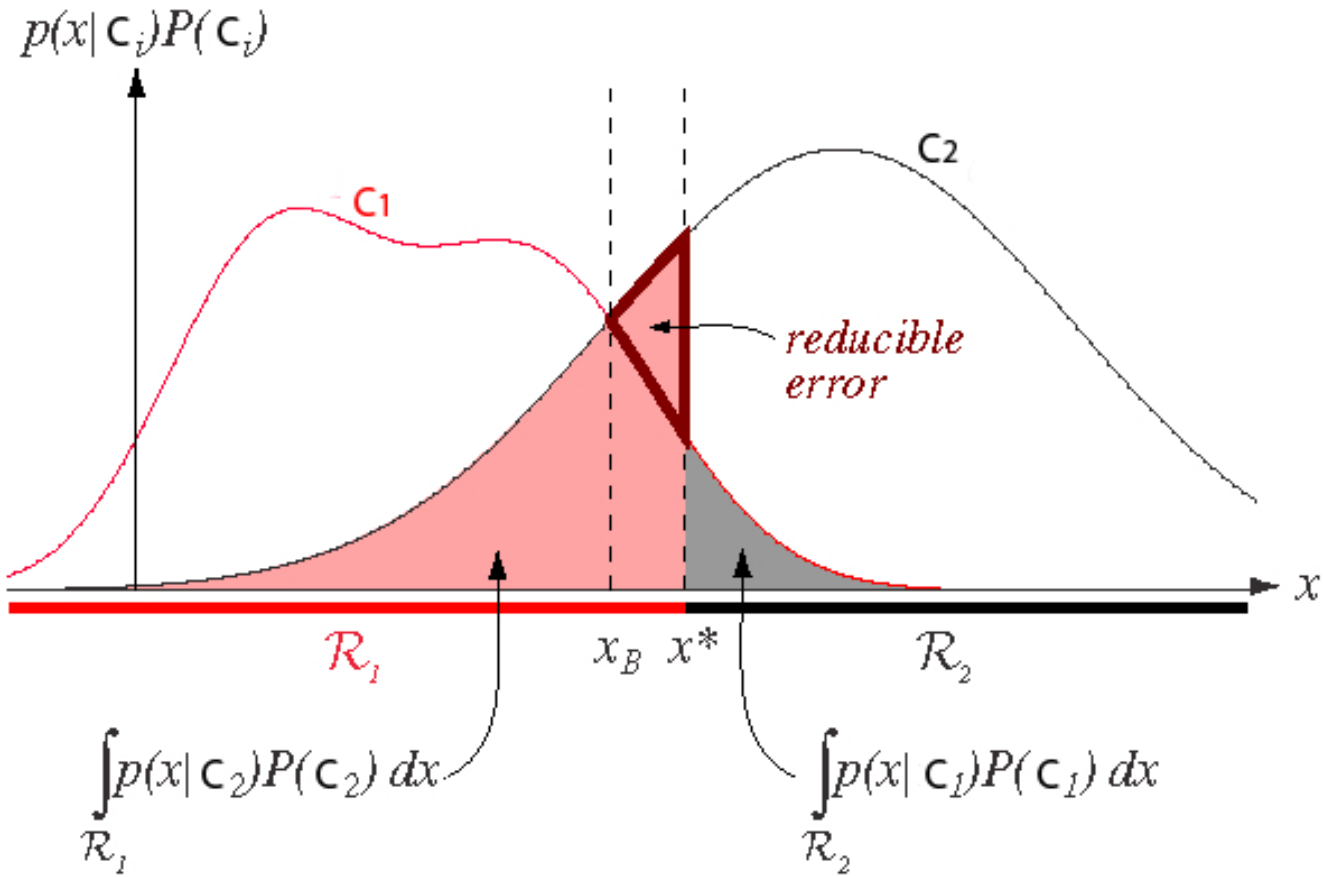


Figure: Components of the probability of error for equal priors and the non-optimal decision point x^*. The optimal point x_B minimizes the total shaded area and gives the Bayes error rate.

Confusion matrix

- Consider the two-category case and define
  - $\mathcal{C}_1$: target is present
  - $\mathcal{C}_2$: target is present

Table: Confusion matrix.

		Assigned	
		$\mathcal{C}_1$	$\mathcal{C}_2$
True	$\mathcal{C}_1$	correct detection	mis-detection
	$\mathcal{C}_2$	false alarm	correct rejection

Bayesian Decision Theory

How can we generalize to

- more than one feature?
  - replace the scalar x by the feature vector $\mathbf{x}$
- more than two states of nature?
  - just a difference in notation
- allowing actions other than just decisions?
  - allow the possibility of rejection
- different risks in the decision?
  - define how costly each action is

Minimum-error-rate Classification

- Let $\{\mathcal{C}_1, ..., \mathcal{C}_K\}$ be the finite set of K states of nature (classes, categories).
- Let $\mathbf{x}$ be the D-component vector-valued random variable called the feature vector.
- If all errors are equally costly, the minimum-error decision rule is defined as
  Decide $\mathcal{C}_i \quad$ if $P(\mathcal{C}_i|x) > P(\mathcal{C}_j|x) \quad \forall j \neq i$
- The resulting error is called the Bayes error and is the best performance that can be achieved.

Bayesian Decision Theory

- Bayesian decision theory gives the optimal decision rule under the assumption that the "true" values of the probabilities are known.
- How can we estimate (learn) the unknown $p(\mathbf{x}|\mathcal{C}_j), j = 1, ..., K$?
- Parametric models: assume that the form of the density functions are known
  - Density models (e.g., Gaussian)
  - Mixture models (e.g., mixture of Gaussians)
  - Hidden Markov Models
  - Bayesian Belief Networks
- Non-parametric models: no assumption about the form
  - Histogram-based estimation
  - Parzen window estimation
  - Nearest neighbour estimation

The Gaussian Density

- Gaussian can be considered as a model where the feature vectors for a given class are continuous-valued, randomly corrupted versions of a single typical or prototype vector.
- Some properties of the Gaussian:
  - Analytically tractable
  - Completely specified by the 1st and 2nd moments
  - Has the maximum entropy of all distributions with a given mean and variance
  - Many processes are asymptotically Gaussian (Central Limit Theorem)
  - Uncorrelatedness implies independence

Univariate Gaussian

- For $x \in \mathbf{R}$:

$$P(x) = N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}(\frac{x-\mu}{\sigma})^2\right]$$

where

$$\mu = E[x] = \int_{-\infty}^{+\infty} xP(x)dx$$

$$\sigma^2 = E[(x-\mu)^2] = \int_{-\infty}^{+\infty} (x-\mu)^2 P(x)dx$$

Univariate Gaussian

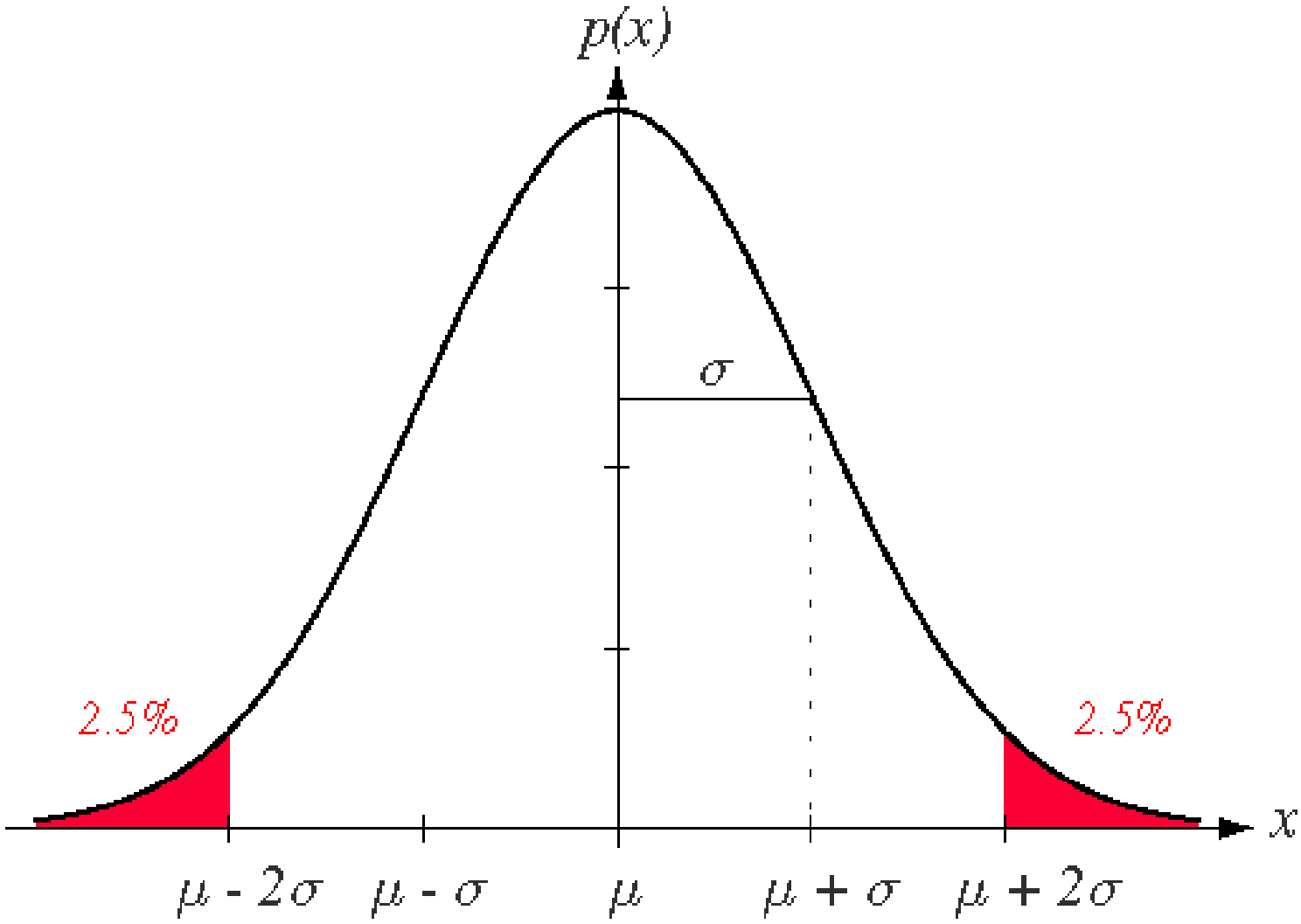


Figure: A univariate Gaussian distribution has roughly 95% of its area in the range $|x - \mu| \leq 2\sigma$

Multivariate Gaussian

- For $\mathbf{x} \in \mathbf{R}^D$:

$$p(\mathbf{x}) = N(\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right]$$

where

$$\boldsymbol{\mu} = E(\mathbf{x}) = \int \mathbf{x}P(\mathbf{x})D\mathbf{x}$$

$$\Sigma = E[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T]$$

Multivariate Gaussian

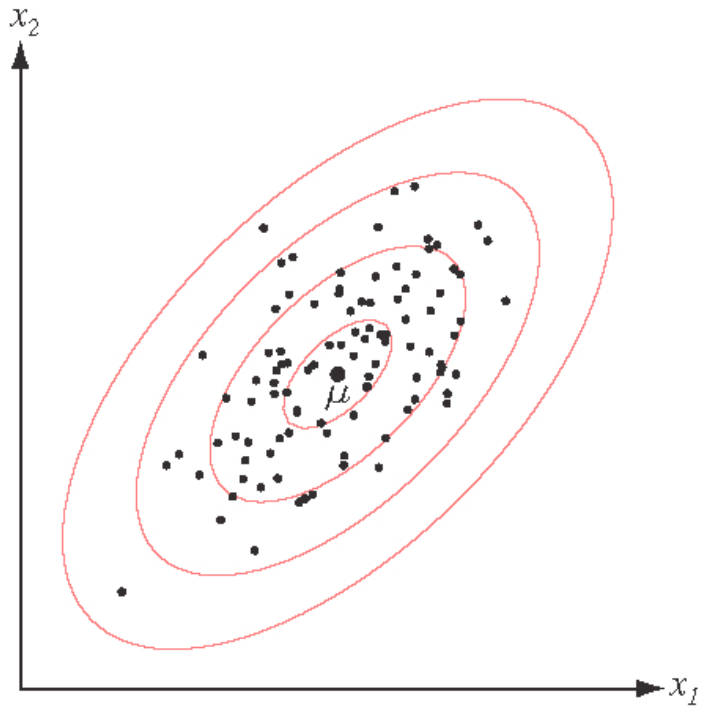


Figure: Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean $\boldsymbol{\mu}$. The loci of points of constant density are the ellipses for which $(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})$ is constant, where the eigenvectors of Σ determine the direction and the corresponding eigenvalues determine the length of the principal axes. The quantity $r^2=(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})$ is called the squared Mahalanobis distance from $\mathbf{x}$ to $\boldsymbol{\mu}$.

Bayes Linear Classifier

- Let us assume that the class-conditional densities are Gaussian and then explore the resulting form for the posterior probabilities.
- assume that all classes share the same covariance matrix. Thus the density for class $\mathcal{C}_k$is given by

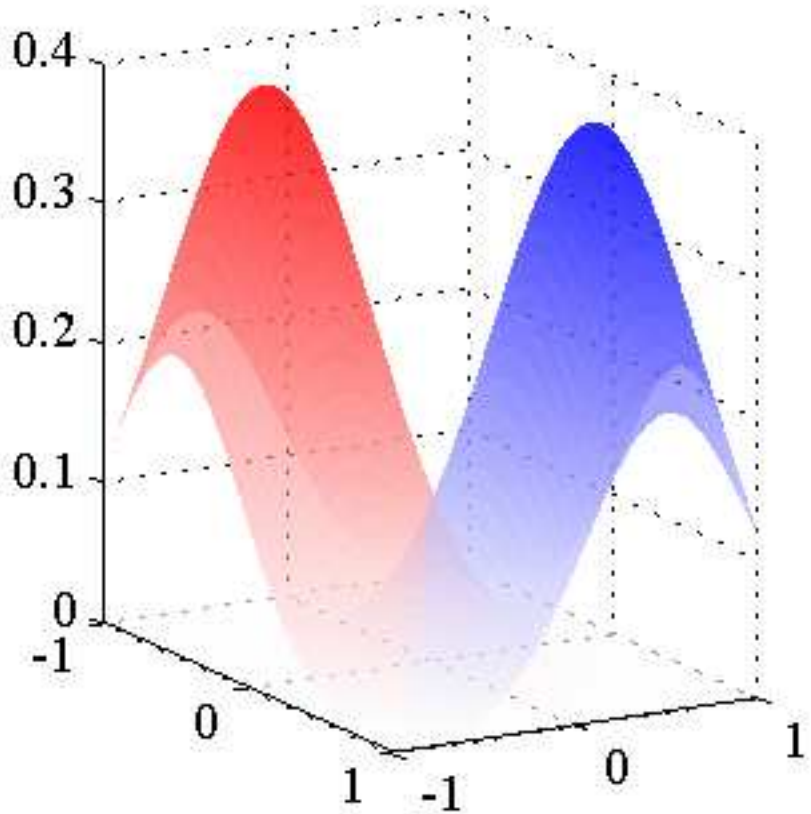
$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- We thus model the class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$ and class priors $p(\mathcal{C}_k)$
- Then use these to compute posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$ through Bayes' theorem:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{j=1}^{K} p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)}$$

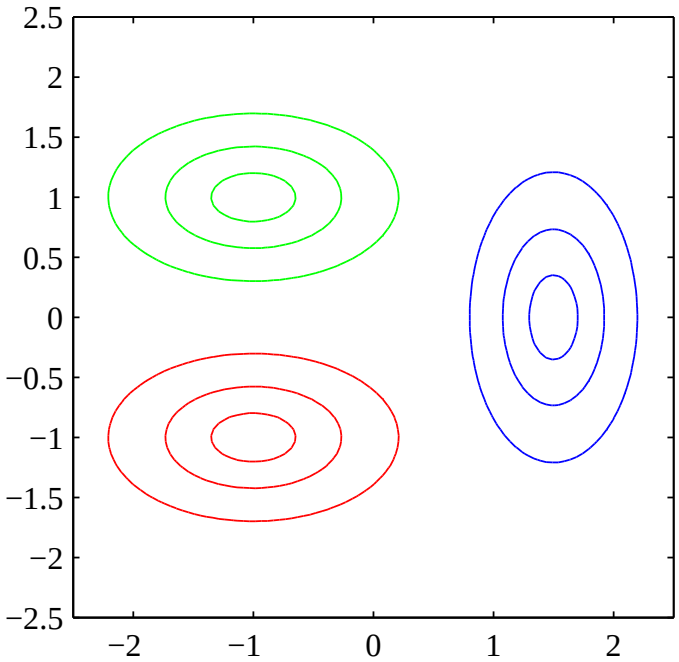
- assuming only 2 classes the decision boundary is linear: check this!

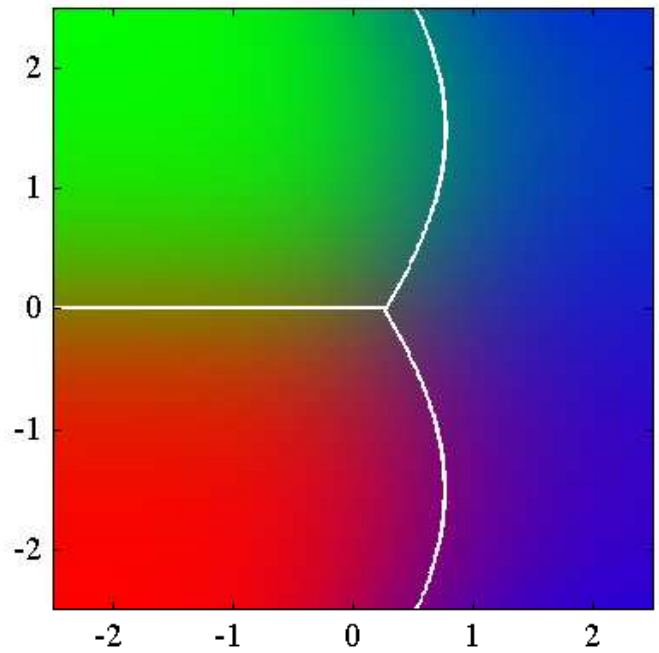
Bayes Linear Classifier



Quadratic discriminant Model

The decision surface is planar when the covariance matrices are the same and quadratic when they are not.





Bayesian Decision Theory

- Bayesian Decision Theory shows us how to design an optimal classifier if we know the prior probabilities $P(\mathcal{C}_i)$ and the class-conditional densities $P(\mathbf{x}|\mathcal{C}_i)$.
- Unfortunately, we rarely have complete knowledge of the probabilistic structure.
- However, we can often find design samples or training data that include particular representatives of the patterns we want to classify.

Gaussian Density Estimation

- The maximum likelihood estimates of a Gaussian are $\hat{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ and $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$

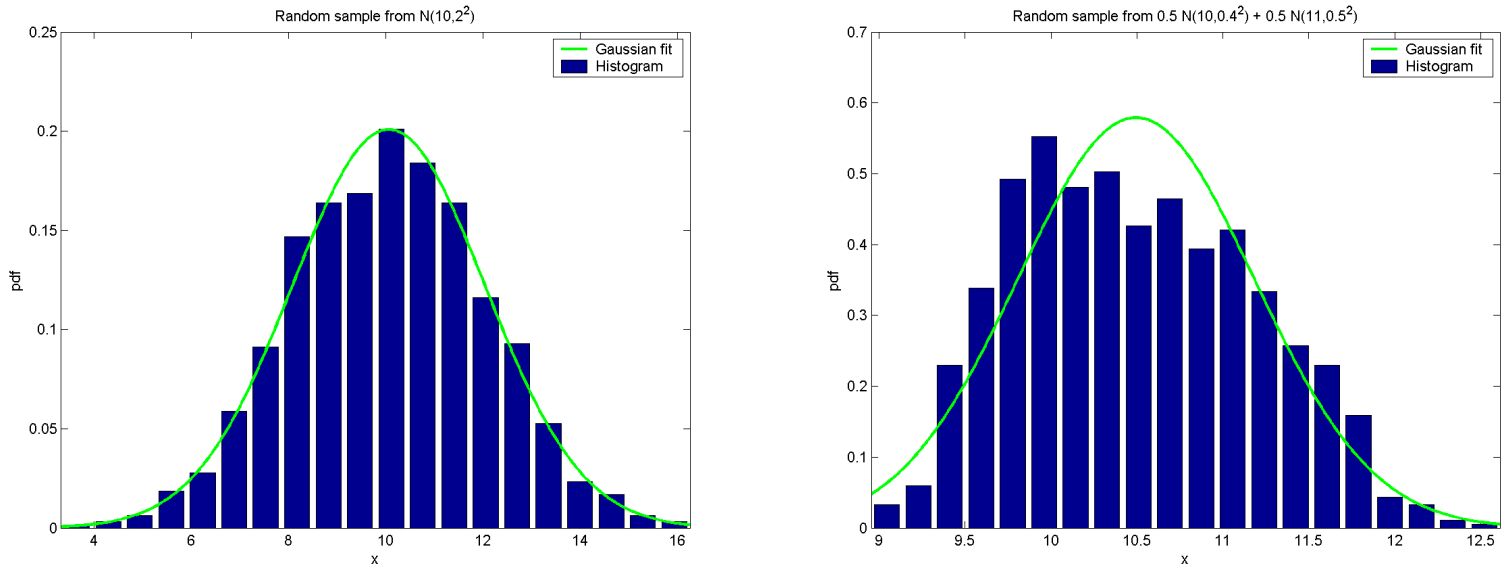
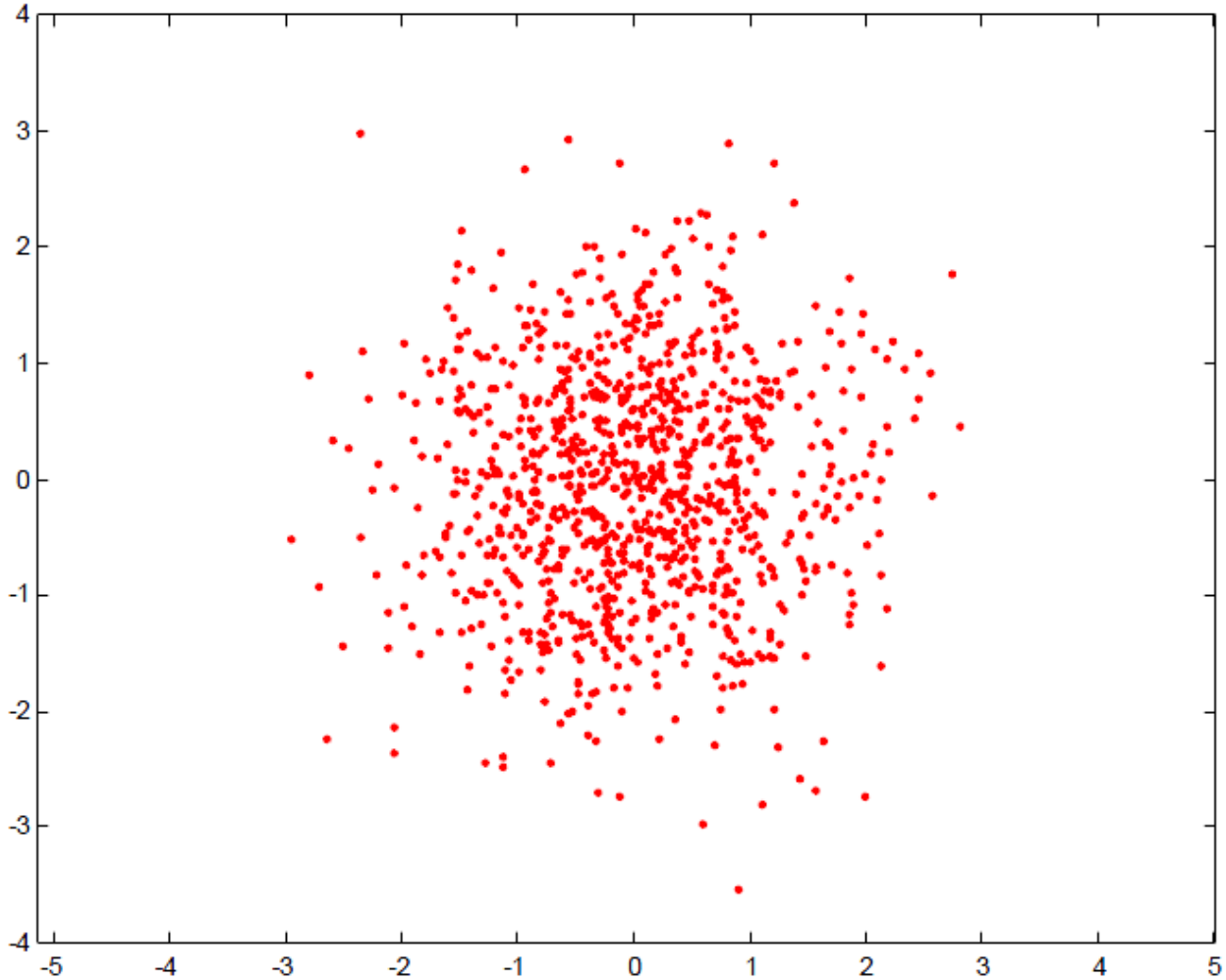


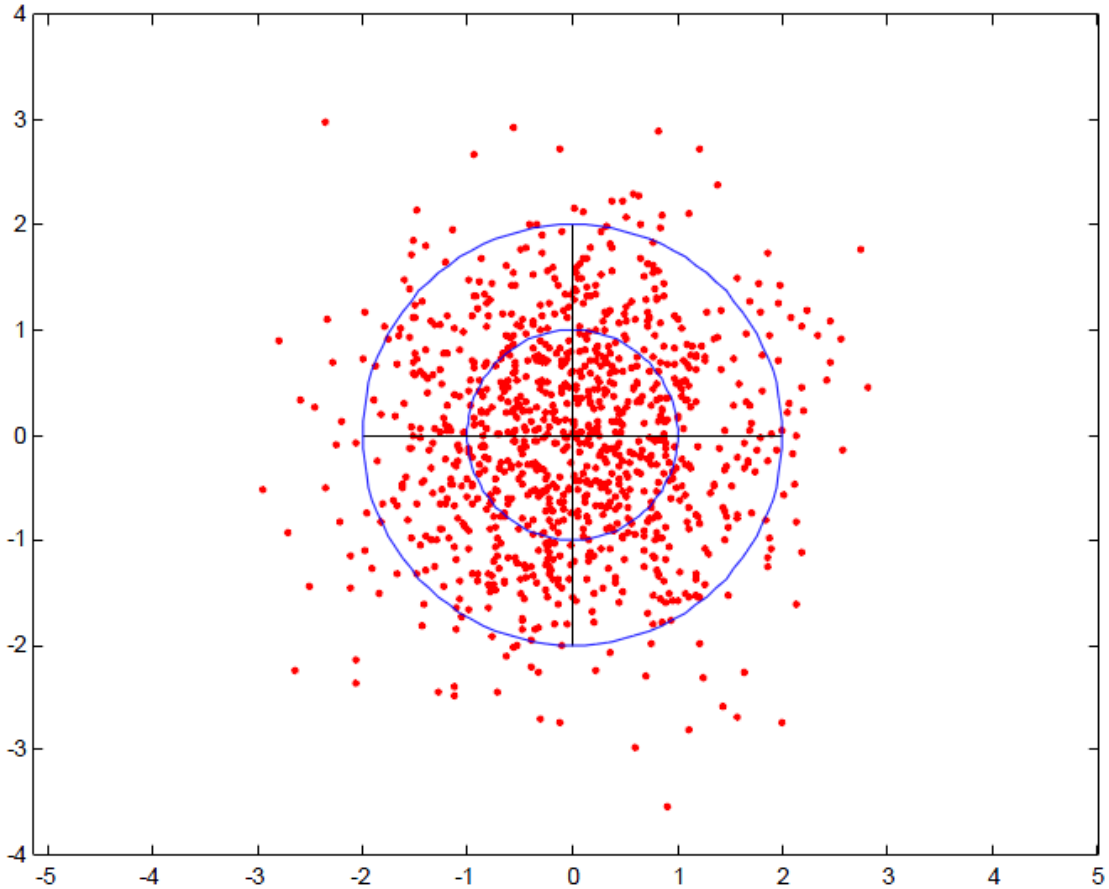
Figure: Gaussian density estimation examples.

2D Gaussian



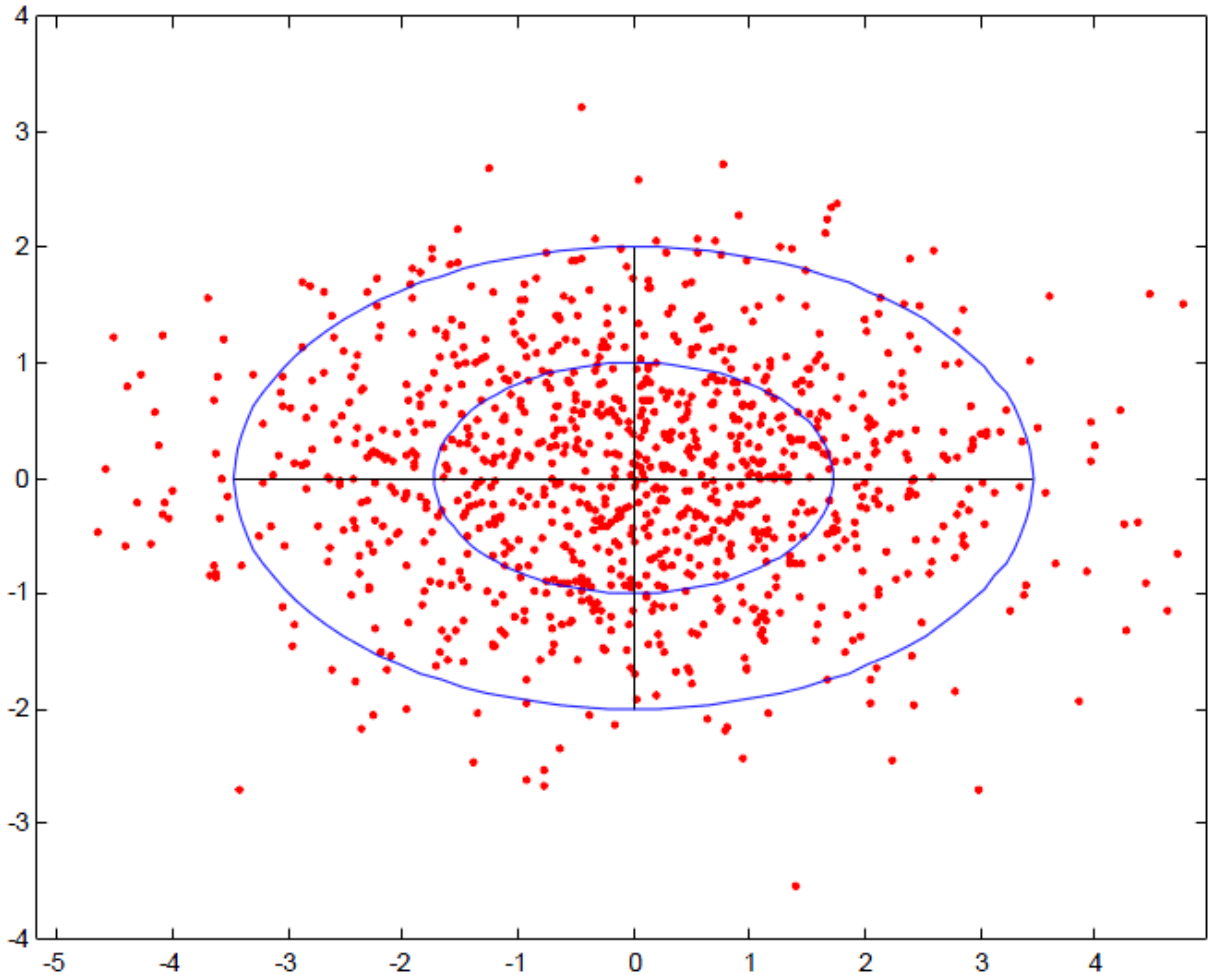
2D Gaussian

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



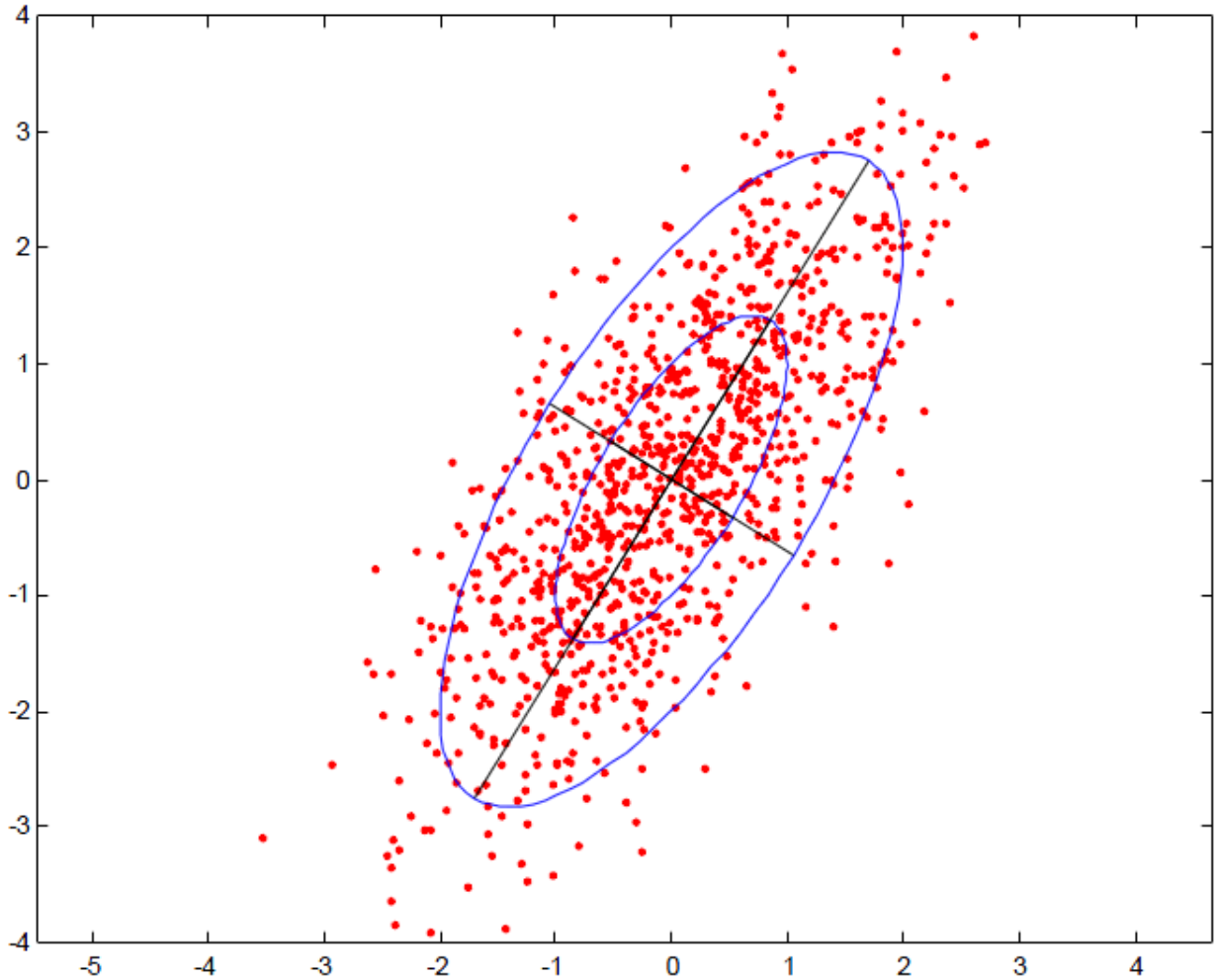
2D Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix}$$



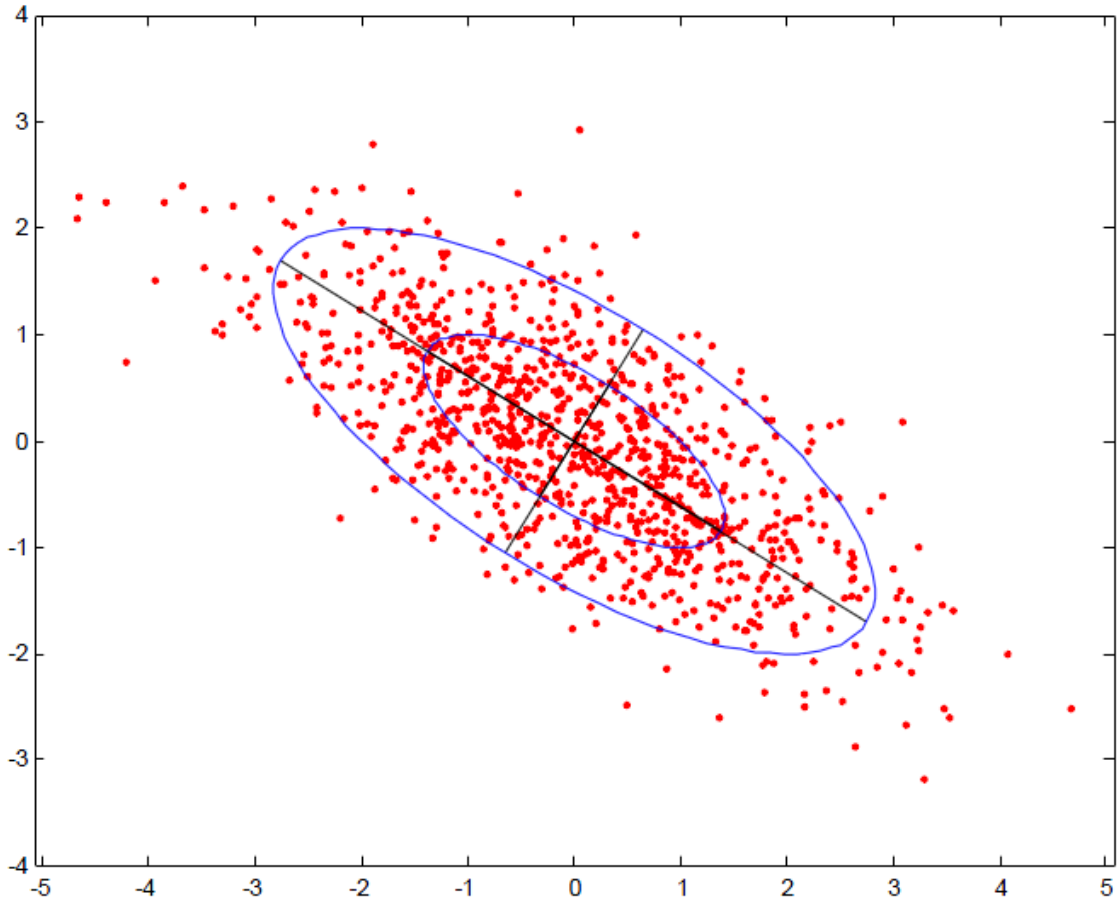
2D Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$$



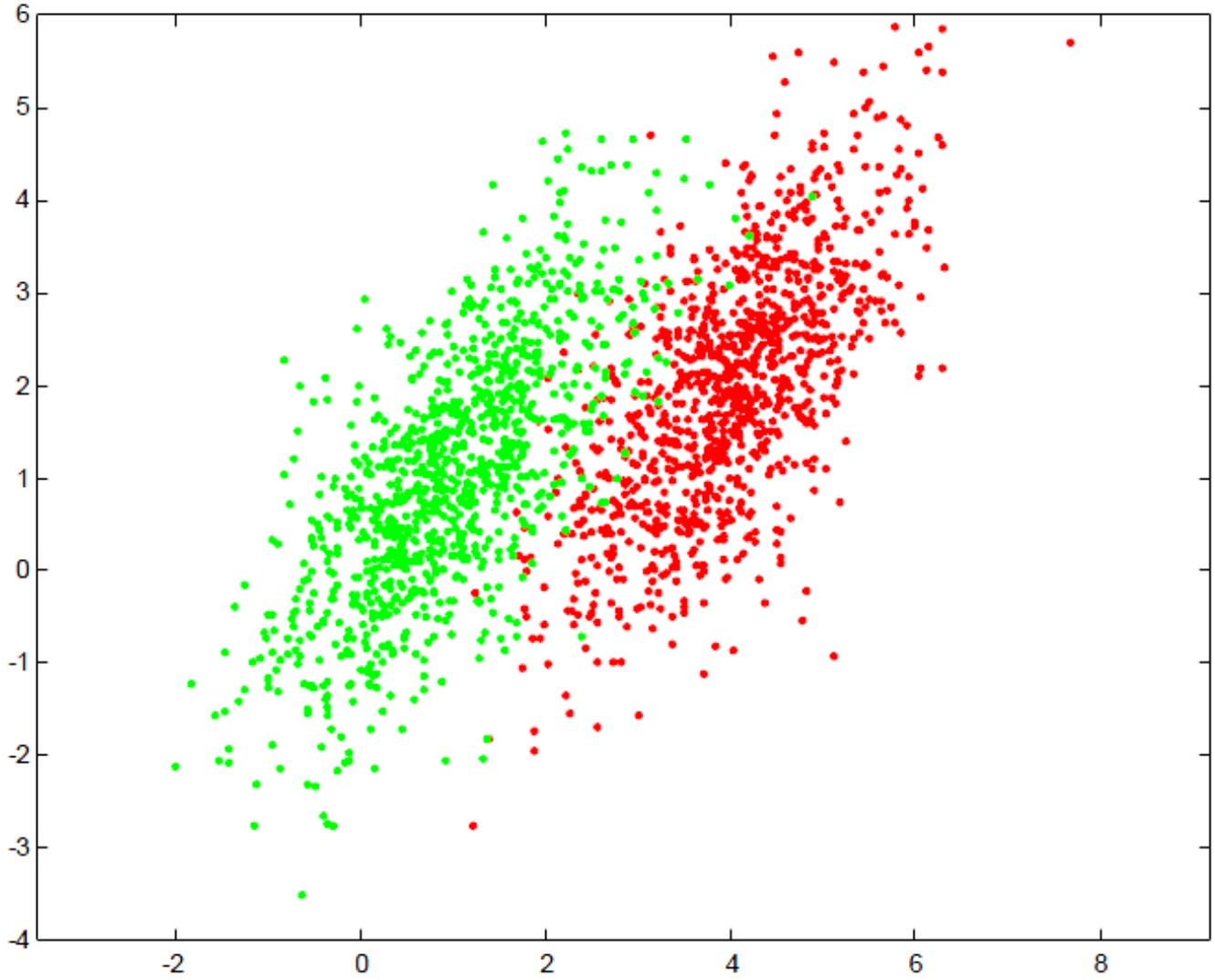
2D Gaussian

$$\boldsymbol{\Sigma} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$$



2D Example

Consider a classification problem with 2 features and 2 classes.



2D Example

- The estimated class-conditional distributions from the data are:

$$p(\mathbf{x}|\mathcal{C}_1) : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in N\left(\begin{bmatrix} 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\right)$$

$$p(\mathbf{x}|\mathcal{C}_2) : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in N\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\right)$$

- We assume equal losses and equal priors.

We wish to compute the classification rule.

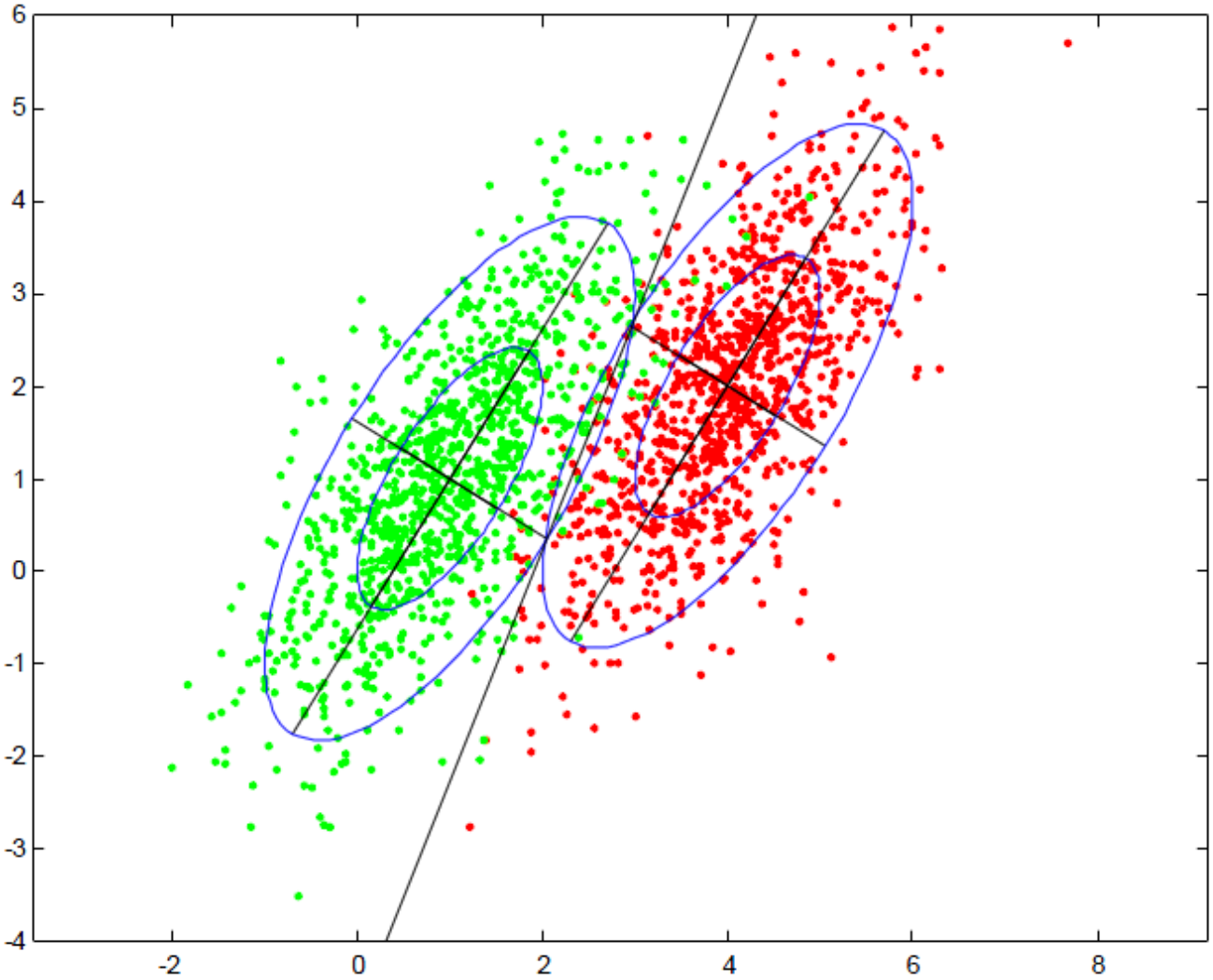
2D Example

Solution

- auxiliar computations: $\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \leftrightarrow \Sigma^{-1} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$
- $d(\mathbf{x}) = 5x_1 - 2x_2 - 9.5$

2D Example

Solution



2D Example b

Consider a classification problem with 2 features and 3 classes.

- The estimated class-conditional distributions from the data are:

$$p(\mathbf{x}|\mathcal{C}_1) : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in N\left(\begin{bmatrix} 4 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\right)$$

$$p(\mathbf{x}|\mathcal{C}_2) : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in N\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\right)$$

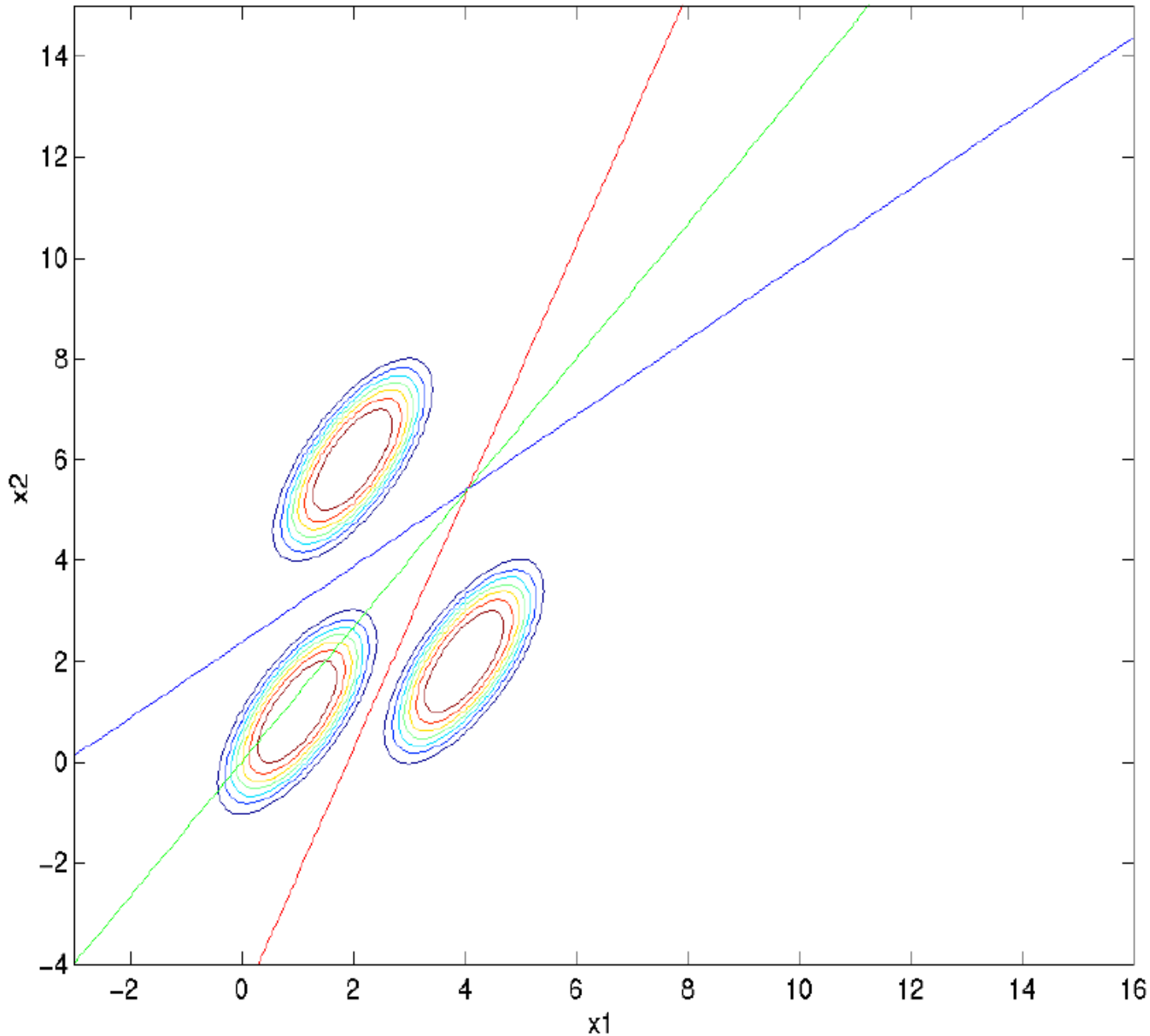
$$p(\mathbf{x}|\mathcal{C}_3) : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in N\left(\begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\right)$$

- We assume equal losses and equal priors.

We wish to compute the classification rule.

2D Example b

Solution



Classifiers based on Bayes Decision Theory

Computation of a-posteriori probabilities

- Assume known
  - a-priori probabilities $p(\mathcal{C}_1), \ldots, p(\mathcal{C}_K)$
  - $p(\mathbf{x}|\mathcal{C}_1), \ldots, p(\mathbf{x}|\mathcal{C}_K)$
    This is also known as the likelihood of $\mathbf{x}$ with respect to $\mathcal{C}_i$

Classifiers based on Bayes Decision Theory

- The Bayes rule (for $K = 2$)
  - $p(\mathbf{x})p(\mathcal{C}_i|\mathbf{x}) = p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i) \Rightarrow p(\mathcal{C}_i|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i)}{p(\mathbf{x})}$
  - $p(\mathbf{x}) = \sum_{i=1}^{2} p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i)$

Classifiers based on Bayes Decision Theory

The Bayes classification rule (for two classes K=2)

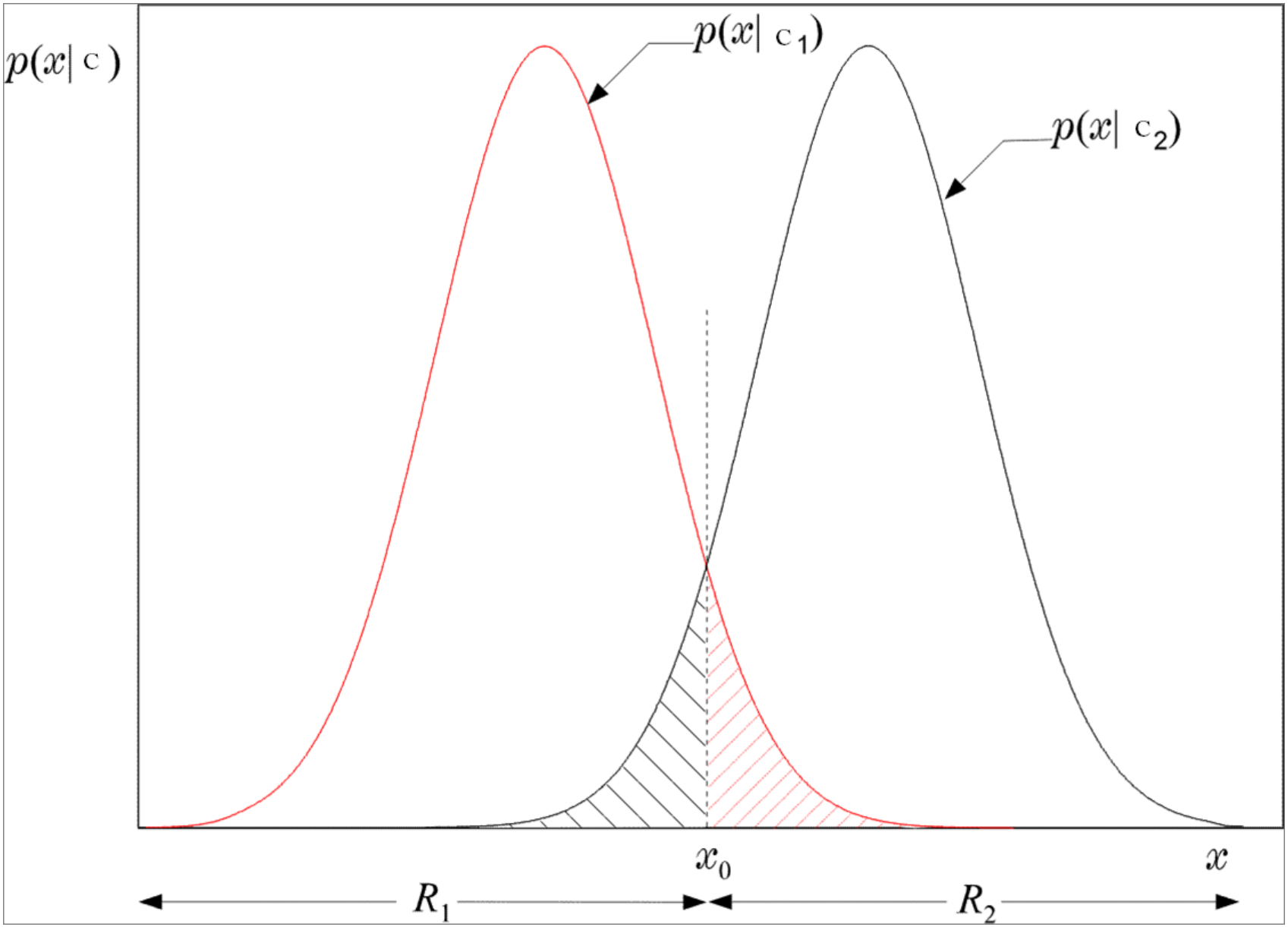
- Given $\mathbf{x}$ classify it according to the rule
  - if $p(\mathcal{C}_1|\mathbf{x}) > p(\mathcal{C}_2|\mathbf{x}) \quad \mathbf{x} \rightarrow \mathcal{C}_1$
  - if $p(\mathcal{C}_2|\mathbf{x}) > p(\mathcal{C}_1|\mathbf{x}) \quad \mathbf{x} \rightarrow \mathcal{C}_2$
- Equivalently: classify $\mathbf{x}$ according to the rule

$$p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) \gtrless p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)$$

- For equiprobable classes the test becomes

$$p(\mathbf{x}|\mathcal{C}_1) \gtrless p(\mathbf{x}|\mathcal{C}_2)$$

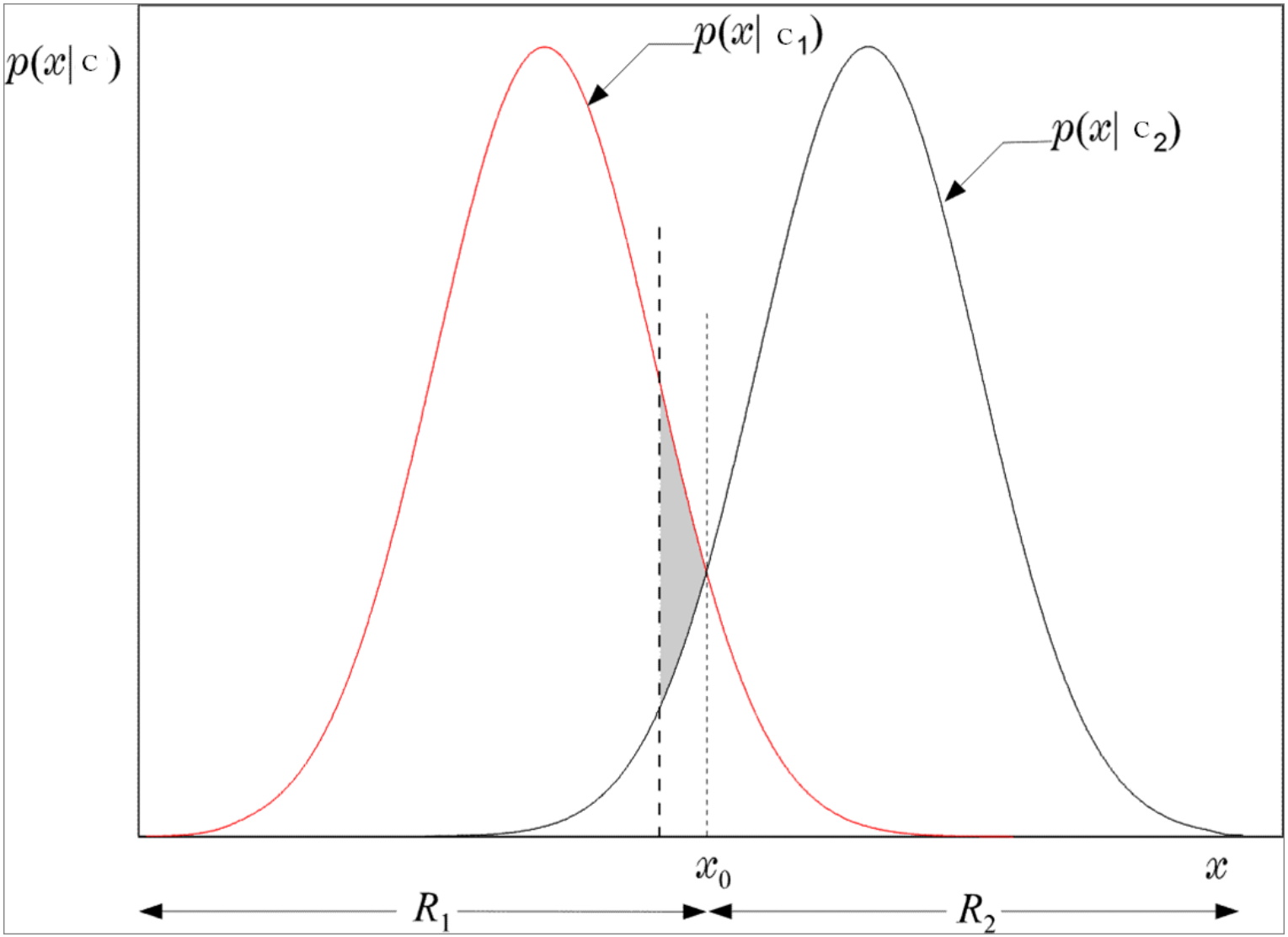
Classifiers based on Bayes Decision Theory



Classifiers based on Bayes Decision Theory

- Equivalently in words: Divide space in two regions
  - if $\mathbf{x} \in R_1$: decide for $\mathcal{C}_1$
  - if $\mathbf{x} \in R_2$: decide for $\mathcal{C}_2$
- Probability of error
  - Total shaded area
  - $P_e = 0.5 \int_{-\infty}^{x_0} p(x|\mathcal{C}_2) + 0.5 \int_{x_0}^{+\infty} p(x|\mathcal{C}_1)$
- Bayesian classifier is OPTIMAL with respect to minimising the classification error probability

Classifiers based on Bayes Decision Theory



Indeed: Moving the threshold the total shaded area INCREASES by the extra "grey" area.

Classifiers based on Bayes Decision Theory

- The Bayes classification rule for many ($K > 2$) classes:
  - Given $\mathbf{x}$ classify it to $\mathcal{C}_i$ if:

$$p(\mathcal{C}_i|\mathbf{x}) > p(\mathcal{C}_j|\mathbf{x}), \quad \forall j \neq i$$

  - Such a choice also minimizes the classification error probability
- Minimizing the average risk
  - For each wrong decision, a penalty term is assigned since some decisions are more sensitive than others

Classifiers based on Bayes Decision Theory

- For $(K = 2)$:
  - Define the loss matrix

$$L = \begin{bmatrix} \ell_{11} & \ell_{12} \\ \ell_{21} & \ell_{22} \end{bmatrix}$$

  - ℓ_{12} is the penalty term for deciding class $\mathcal{C}_2$ although the pattern belongs to $\mathcal{C}_1$
  - cost of deciding for $\mathcal{C}_1$:

$$\ell_{11} p(\mathcal{C}_1|\mathbf{x}) + \ell_{21} p(\mathcal{C}_2|\mathbf{x})$$

  - cost of deciding for $\mathcal{C}_2$:

$$\ell_{12} p(\mathcal{C}_1|\mathbf{x}) + \ell_{22} p(\mathcal{C}_2|\mathbf{x})$$

Classifiers based on Bayes Decision Theory

- For $(K = 2)$:
  - Decide for $\mathcal{C}_1$ if

$$\ell_{11}p(\mathcal{C}_1|\mathbf{x}) + \ell_{21}p(\mathcal{C}_2|\mathbf{x}) < \ell_{12}p(\mathcal{C}_1|\mathbf{x}) + \ell_{22}p(\mathcal{C}_2|\mathbf{x})$$

$$(\ell_{11} - \ell_{12})p(\mathcal{C}_1|\mathbf{x}) < (\ell_{22} - \ell_{21})p(\mathcal{C}_2|\mathbf{x})$$

$$(\ell_{12} - \ell_{11})p(\mathcal{C}_1|\mathbf{x}) > (\ell_{21} - \ell_{22})p(\mathcal{C}_2|\mathbf{x})$$

$$(\ell_{12} - \ell_{11})p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) > (\ell_{21} - \ell_{22})p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)$$

$$\frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} > \frac{p(\mathcal{C}_2)}{p(\mathcal{C}_1)} \frac{(\ell_{21} - \ell_{22})}{(\ell_{12} - \ell_{11})}$$

Classification Error

- To apply these results to multiple classes, separate the training samples to K subsets $\mathcal{D}_1, \ldots, \mathcal{D}_K$, with the samples in $\mathcal{D}_i$ belonging to class $\mathcal{C}_i$, and then estimate each density $p(x|\mathcal{C}_i, \mathcal{D}_i)$ separately.
- Different sources of error:
  - Bayes error: due to overlapping class-conditional densities (related to features used)
  - Model error: due to incorrect model
  - Estimation error: due to estimation from a finite sample (can be reduced by increasing the amount of training data)

Discriminant functions
Decision Surfaces

- $g(\mathbf{x}) \equiv p(\mathcal{C}_i|\mathbf{x}) - p(\mathcal{C}_j|\mathbf{x}) = 0$
  is the surface separating the regions. On the one side is positive $(+)$, on the other is negative (-). It is known as Decision Surface.
- If $f(.)$ monotonically increasing, the rule remains the same if we use: $\mathbf{x} \to \mathcal{C}_i$ if $f(p(\mathcal{C}_i|\mathbf{x})) > f(p(\mathcal{C}_j|\mathbf{x})) \quad \forall j \neq i$

Discriminant functions
Decision Surfaces

- In general, discriminant functions can be defined independent of the Bayesian rule. They lead to suboptimal solutions, yet, if chosen appropriately, they can be computationally more tractable. Moreover, in practice, they may also lead to better solutions. This, for example, may be case if the nature of the underlying pdf's are unknown.

Non-Bayesian Classifiers

- Distance-based classifiers:
  - Minimum (mean) distance classifier
  - Nearest neighbour classifier
- Decision boundary-based classifiers:
  - Linear discriminant functions
  - Support vector machines
  - Neural networks
  - Decision trees

The k-Nearest neighbour Classifier

- Given the training data $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ as a set of n labeled examples, the nearest neighbour classifier assigns a test point $\mathbf{x}$ the label associated with its closest neighbour in $\mathcal{D}$.
- .

The k-nearest neighbour classifier classifies $\mathbf{x}$ by assigning it the label most frequently represented among the k nearest samples.

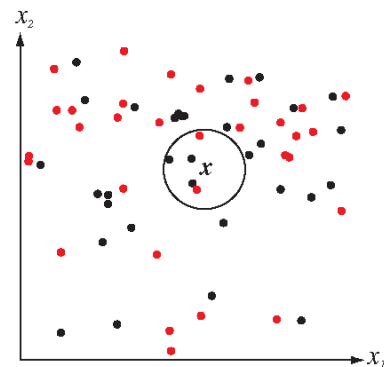


Figure: Classifier for $k = 5$.

- Closeness is defined using a distance function.

Distance Functions

- A general class of metrics for d-dimensional patterns is the Minkowski metric

$$L_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

  also referred to as the L_p norm.

- The Euclidean distance is the L_2 norm

$$L_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^2 \right)^{1/2}$$

- The Manhattan or city block distance is the L_1 norm

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i|$$

Linear Discriminant Functions

The L_∞ norm is the maximum of the distances along individual coordinate axes

$$L_\infty(\mathbf{x}, \mathbf{y}) = \max_{i=1}^{d} |x_i - y_i|$$

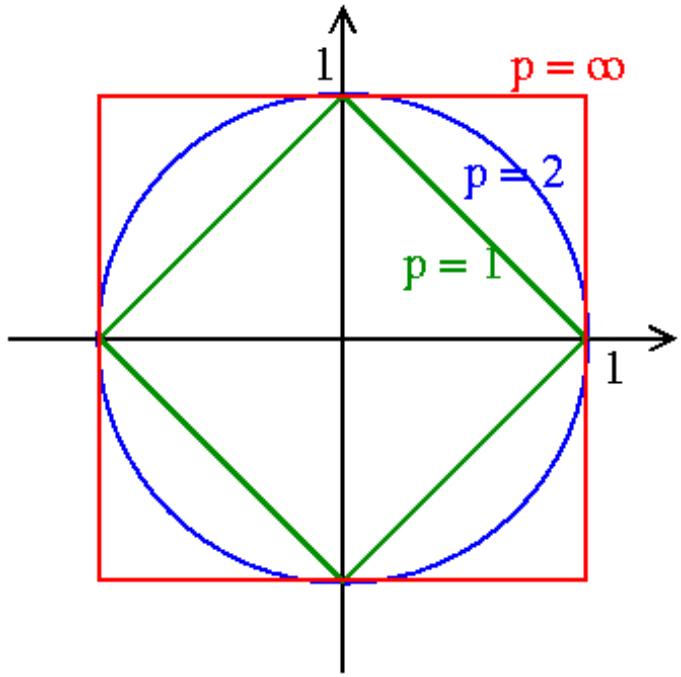


Figure: Each colored shape consists of points at a distance 1.0 from the origin, measured using different values of p in the Minkowski L_p metric.

Linear Discriminant Functions

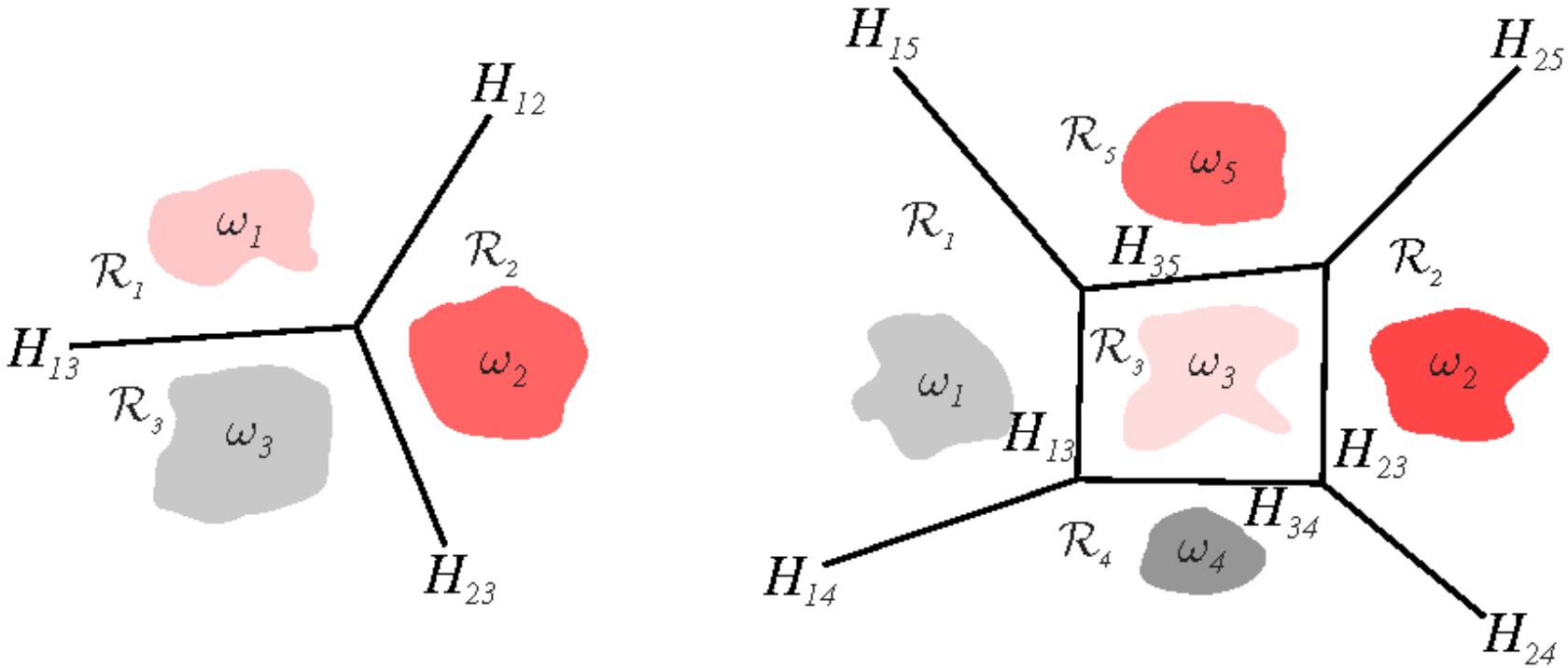


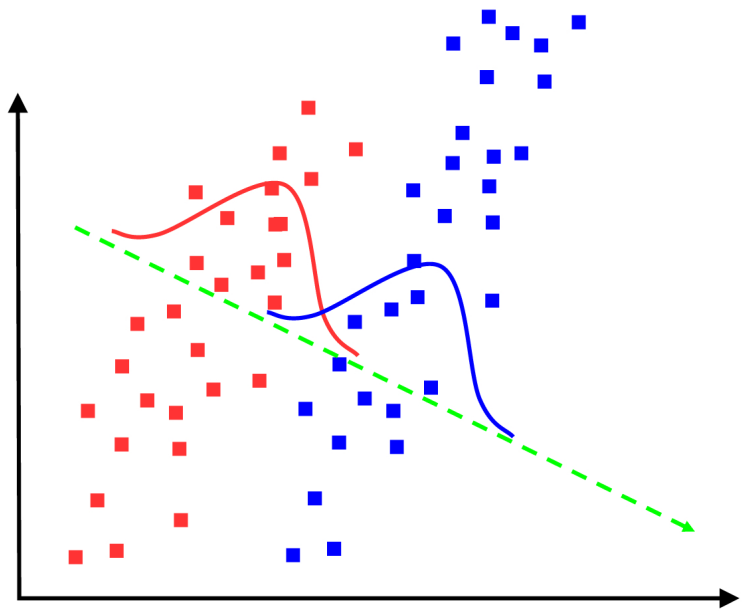
Figure: Linear decision boundaries produced by using one linear discriminant for each class.

Linear Models for Classification

Fisher's linear discriminant

Also known as Linear Discriminant Analysis

- One way to view a linear classification model is in terms of dimensionality reduction.
- Projection that best separates the data in a least-squares sense.
- Projection of D-dimensional data onto a line.



Linear Models for Classification

Fisher's linear discriminant

Also known as Linear Discriminant Analysis

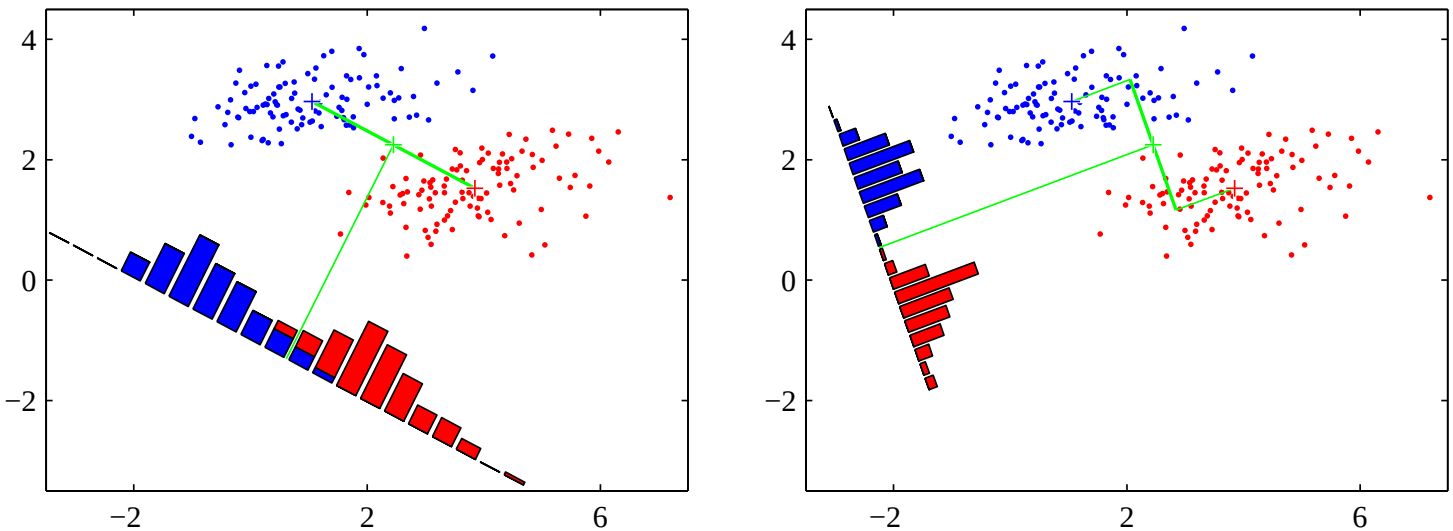
- A simple linear discriminant function is a projection of the data down to 1-D.
  - So choose the projection that gives the best separation of the classes. What do we mean by "best separation"?
- An obvious direction to choose is the direction of the line joining the class means.
  - But if the main direction of variance in each class is not orthogonal to this line, this will not give good separation (see the next figure).
- LDA chooses the direction that maximizes the ratio of between class variance to within class variance.
  - This is the direction in which the projected points contain the most information about class membership (under Gaussian assumptions)

Linear Models for Classification

Fisher's linear discriminant

Also known as Linear Discriminant Analysis

- When projected onto the line joining the class means, the classes are not well separated.
- Fisher chooses a direction that makes the projected classes much tighter, even though their projected means are less far apart.



Linear Models for Classification

Fisher's linear discriminant

Math of Fisher's linear discriminant

- What linear transformation is best for discrimination?
  $y = \mathbf{w}^T \mathbf{x}$
- The projection onto the vector separating the class means seems sensible: $\mathbf{w} \propto \mathbf{m}_2 - \mathbf{m}_1$
  with $\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n$ $\mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n$
  This $\mathbf{w}$ maximizes $m_2 - m_1 = \mathbf{w}^t(\mathbf{m}_2 - \mathbf{m}_1)$, subject to $||\mathbf{w}|| = 1$
- But we also want small variance within each class:
  $s_k^2 = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} (\mathbf{w}^t \mathbf{x}_n - m_k)^2$
- Fisher's objective function is

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

Linear Models for Classification

Fisher's linear discriminant

More Math of Fisher's linear discriminant

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}$$

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$$

$$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$$

$$\textit{Optimal solution}: \quad \mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$$

Support Vector Machines

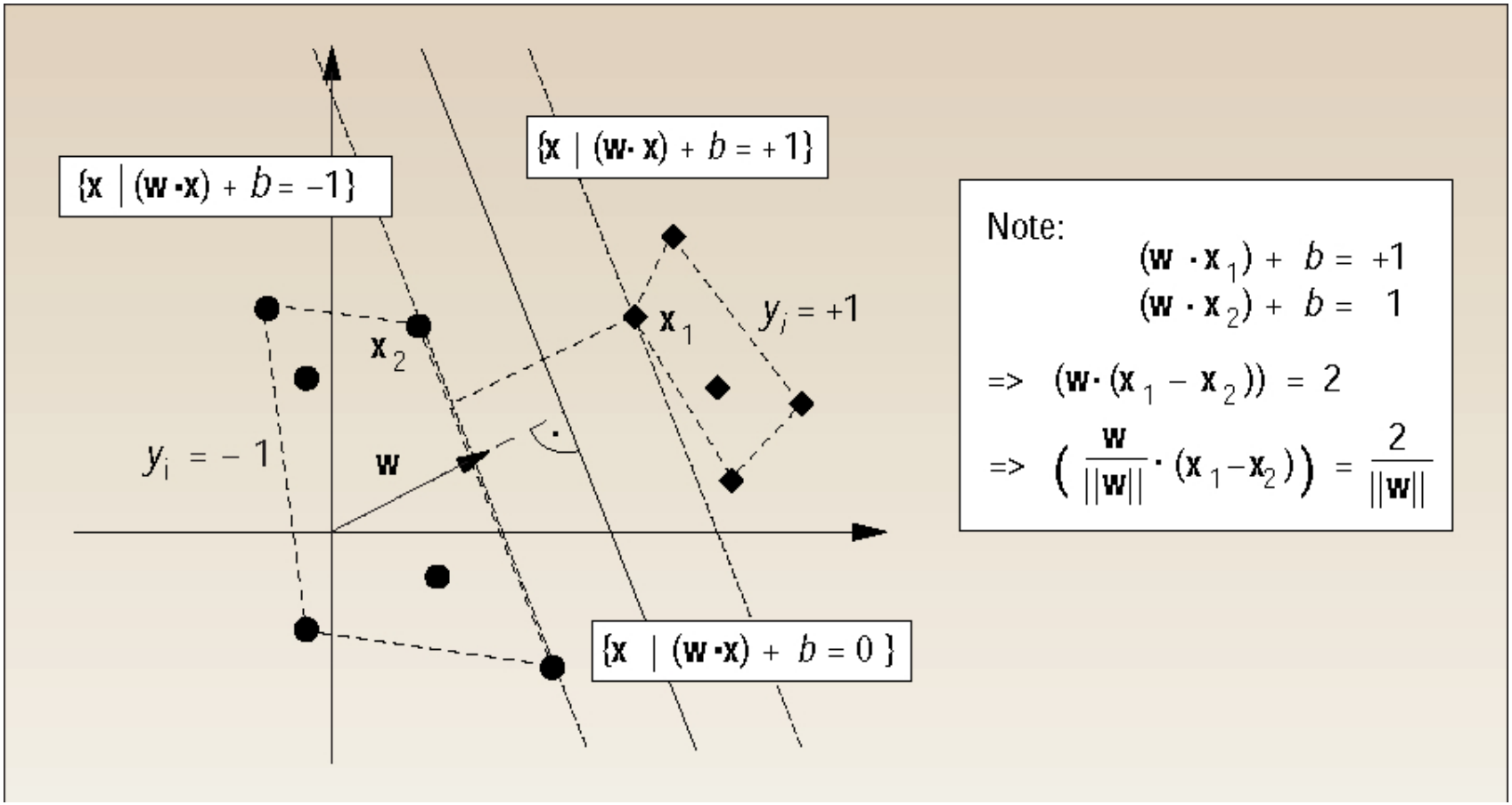


Figure: A binary classification problem of separating balls from diamonds. Support vector machines find hyperplane decision boundaries that yield the maximum margin of separation between the classes. The optimal hyperplane is orthogonal to the shortest line connecting the convex hulls of the two classes (dotted), and intersects it half way between the two classes.

Neural Networks

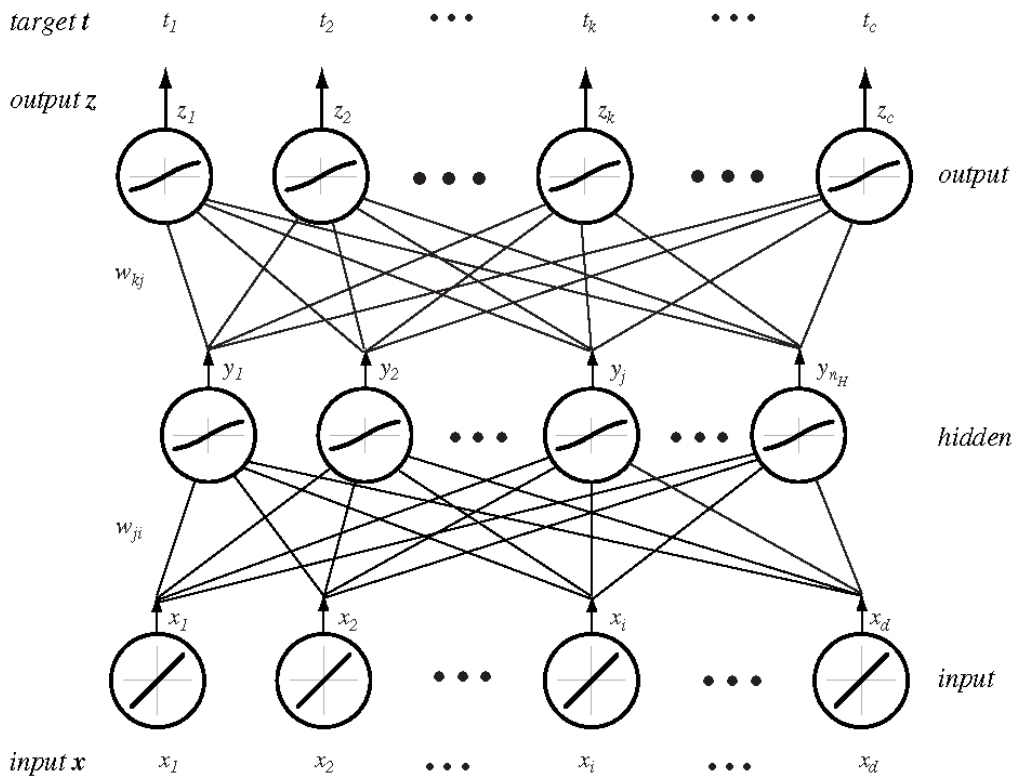


Figure: A neural network consists of an input layer, an output layer and usually one or more hidden layers that are interconnected by modifiable weights represented by links between layers. They learn the values of these weights as a mapping from the input to the output.

Decision Trees

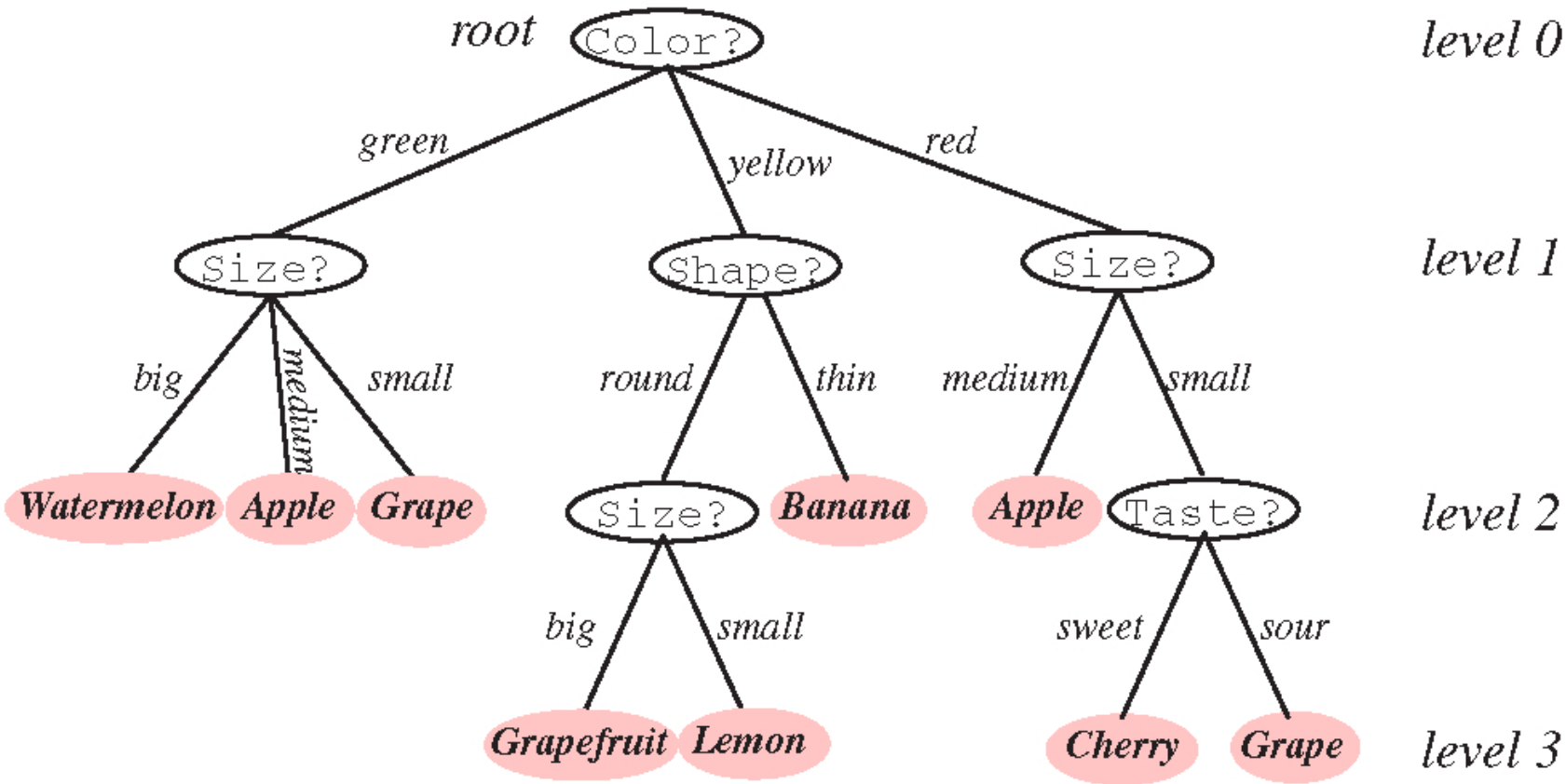


Figure: Decision trees classify a pattern through a sequence of questions, in which the next question asked depends on the answer to the current question.

References


Selim Aksoy
Introduction to Pattern Recognition, Part II,
http://retina.cs.bilkent.edu.tr/papers/patrec_tutorial2.pdf


Richard O. Duda, Peter E. Hart, David G. Stork
Pattern classification
John Wiley & Sons, 2001.


Miguel Coimbra
Pattern Recognition for Computer Vision
http://www.dcc.fc.up.pt/~mcoimbra/


Christopher M. Bishop
Pattern recognition and machine learning,
Springer.