PDEEC – Machine Learning 2018/19

Bayesian Classifiers, Conditional Independence and Naive Bayes; Non Parametric Density Estimation

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

INESC TEC and Faculdade Engenharia, Universidade do Porto

Oct. 11, 2018

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

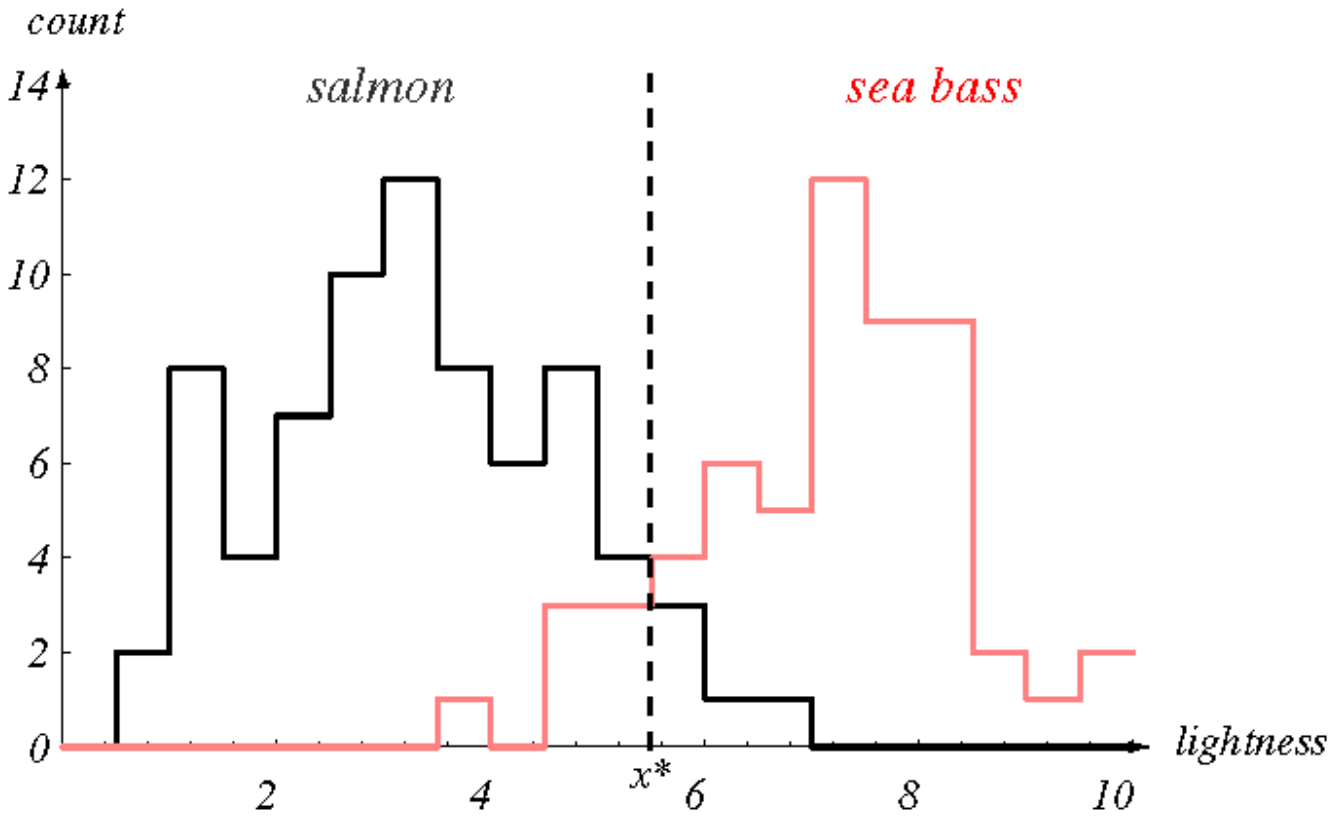
- What is the probability of error for this decision?

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

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

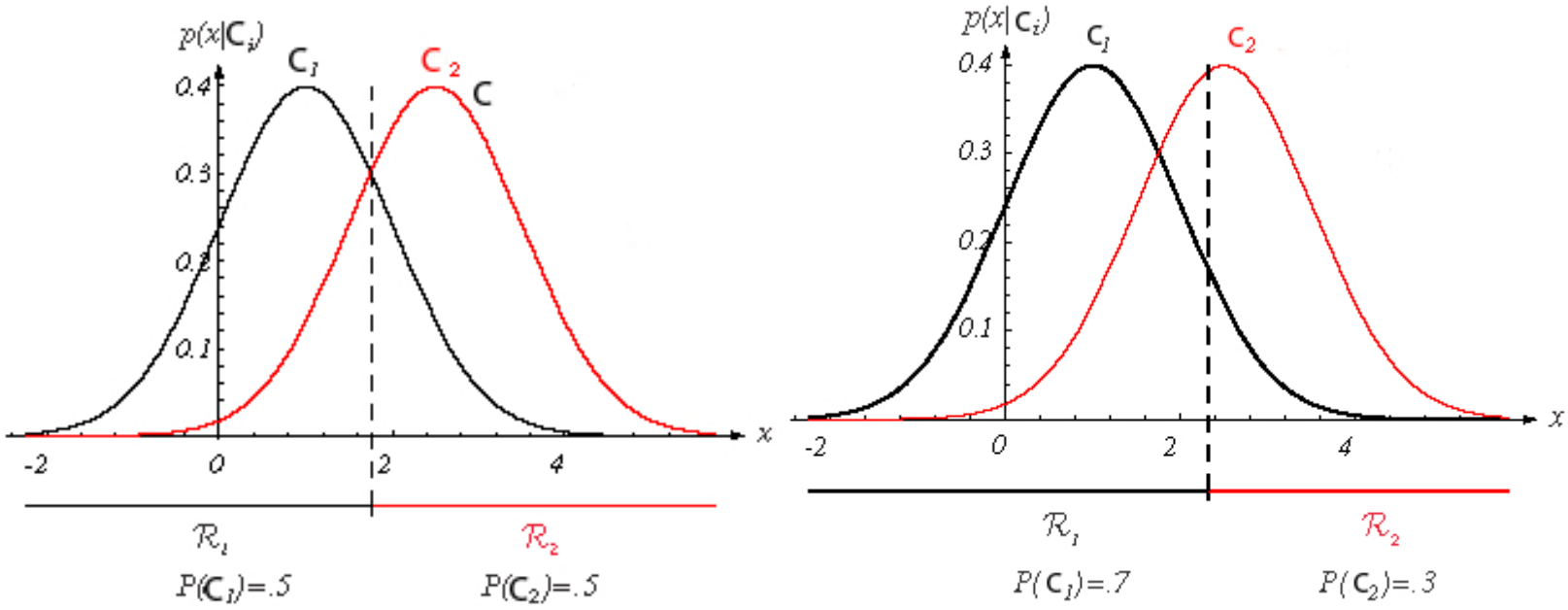


Figure: Optimum thresholds for different priors.

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

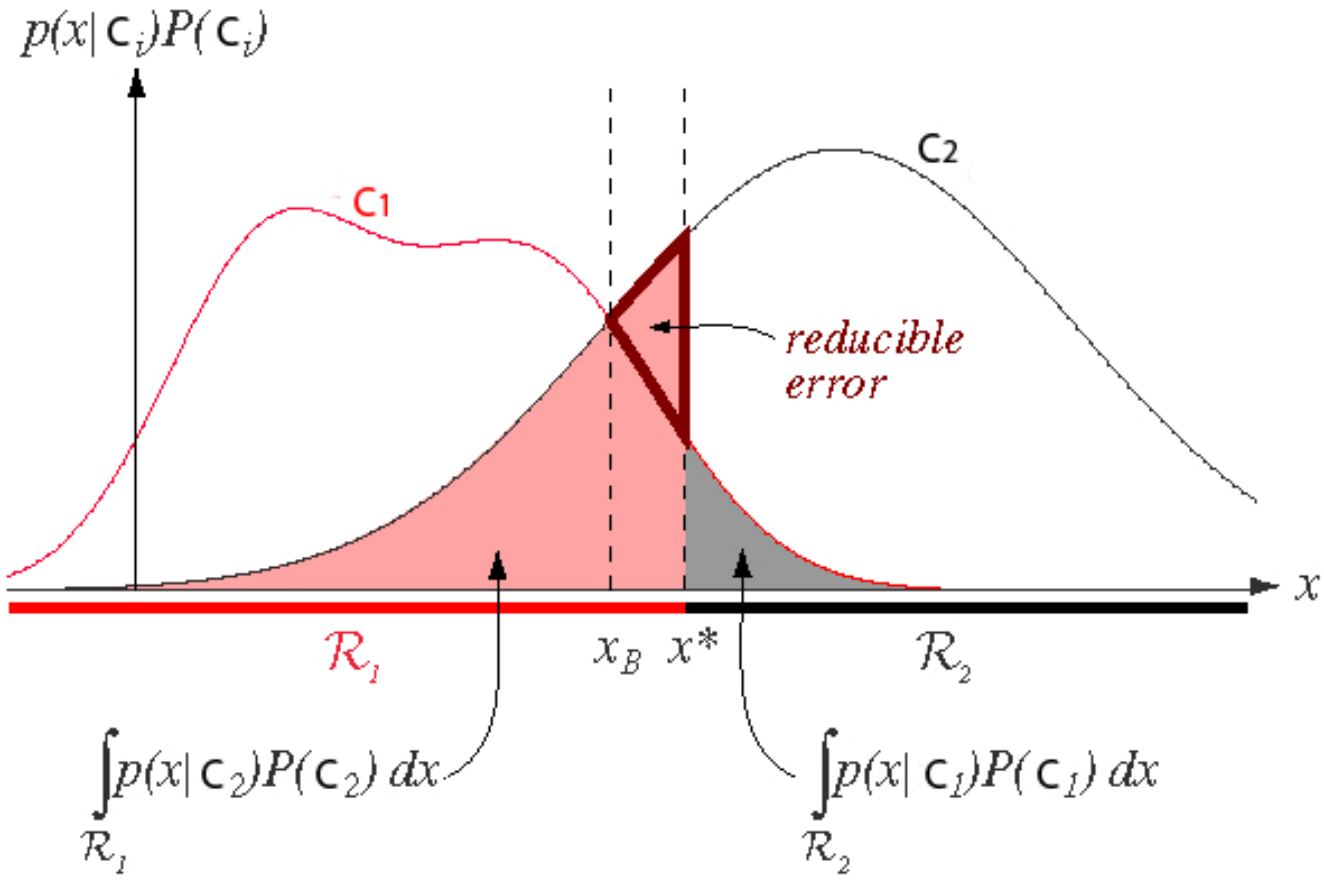


Figure: Components of the probability of error for equal priors and the non-optimal decision point x^*. The optimal point x_B minimizes the total shaded area and gives the Bayes error rate.

Confusion matrix

- Consider the two-category case and define
  - $\mathcal{C}_1$: target is present
  - $\mathcal{C}_2$: target is not 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

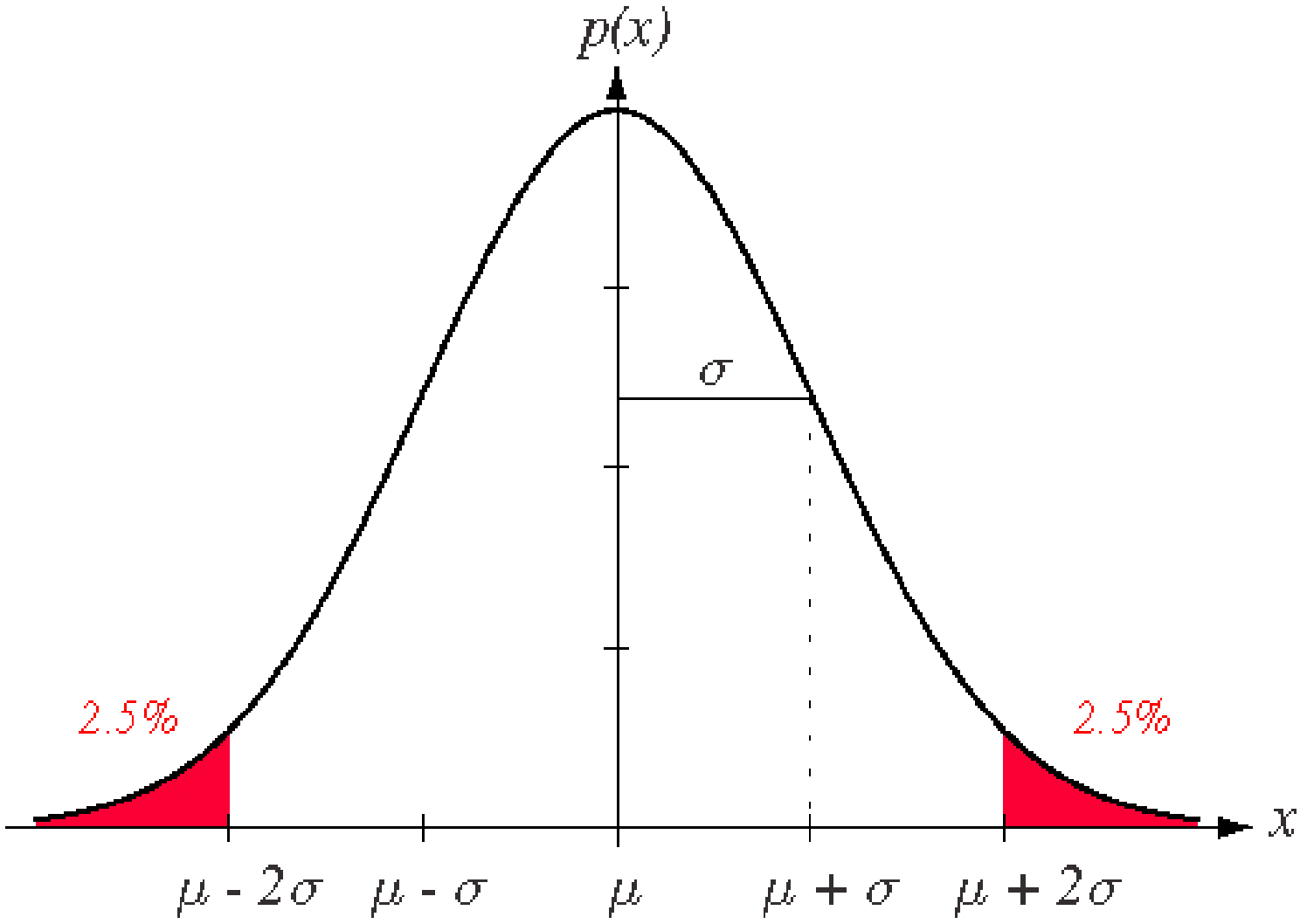


Figure: A univariate Gaussian distribution has roughly 95% of its area in the range $|x - \mu| \leq 2\sigma$

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

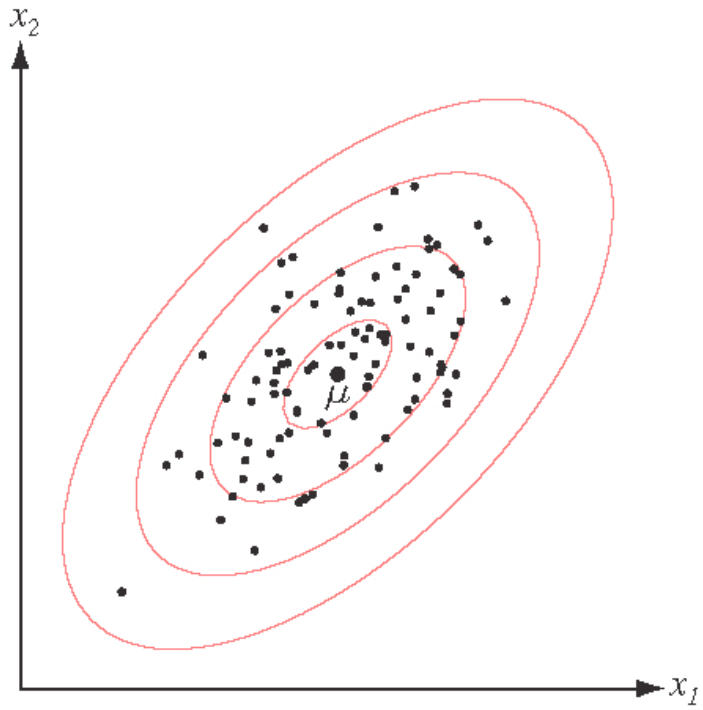


Figure: Samples drawn from a two-dimensional Gaussian lie in a cloud centered on the mean $\boldsymbol{\mu}$. The loci of points of constant density are the ellipses for which $(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})$ is constant, where the eigenvectors of Σ determine 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

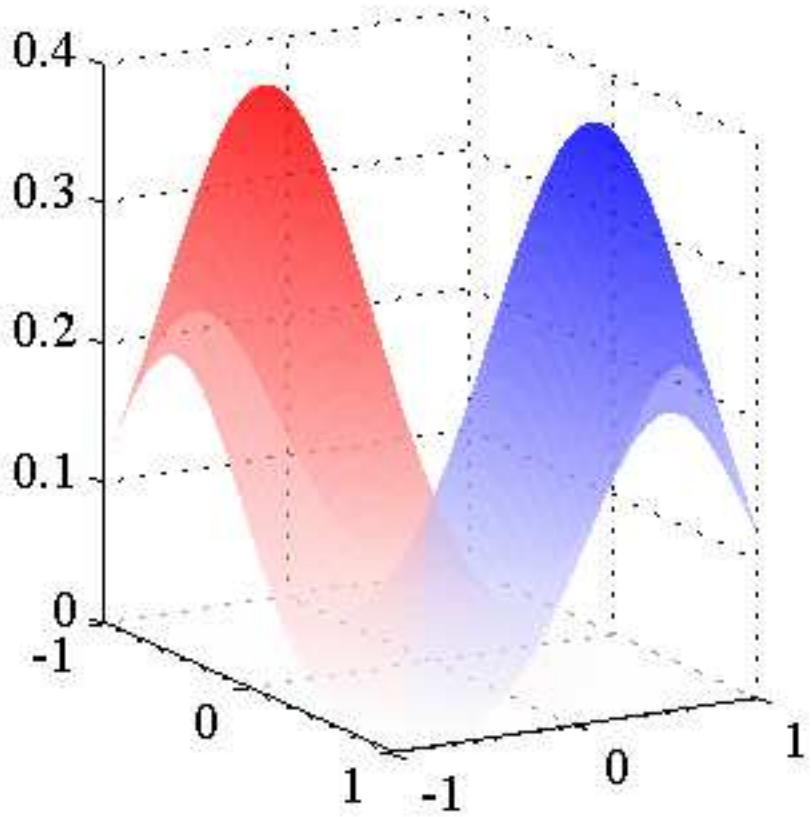
$$p(\mathbf{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right\}$$

- 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:

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

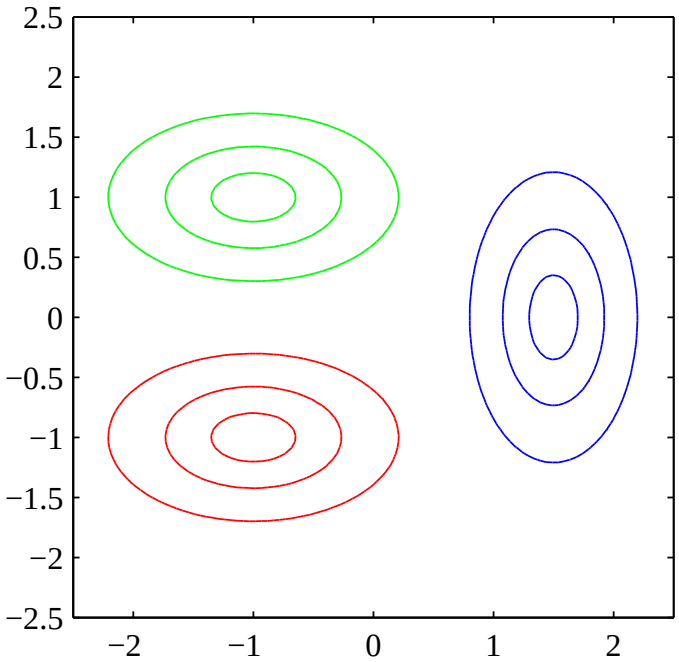
- assuming only 2 classes the decision boundary is linear: check this!

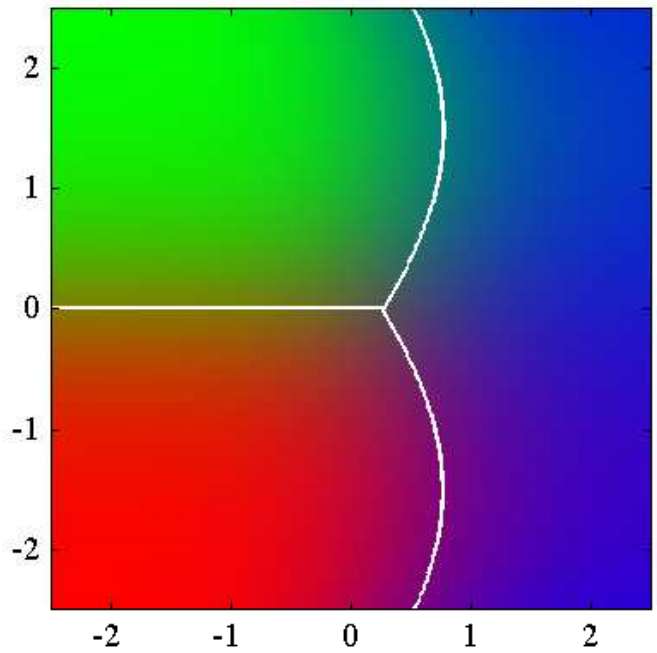
Bayes Linear Classifier



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 $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\boldsymbol{\mu}})(\mathbf{x}_i - \hat{\boldsymbol{\mu}})^T$

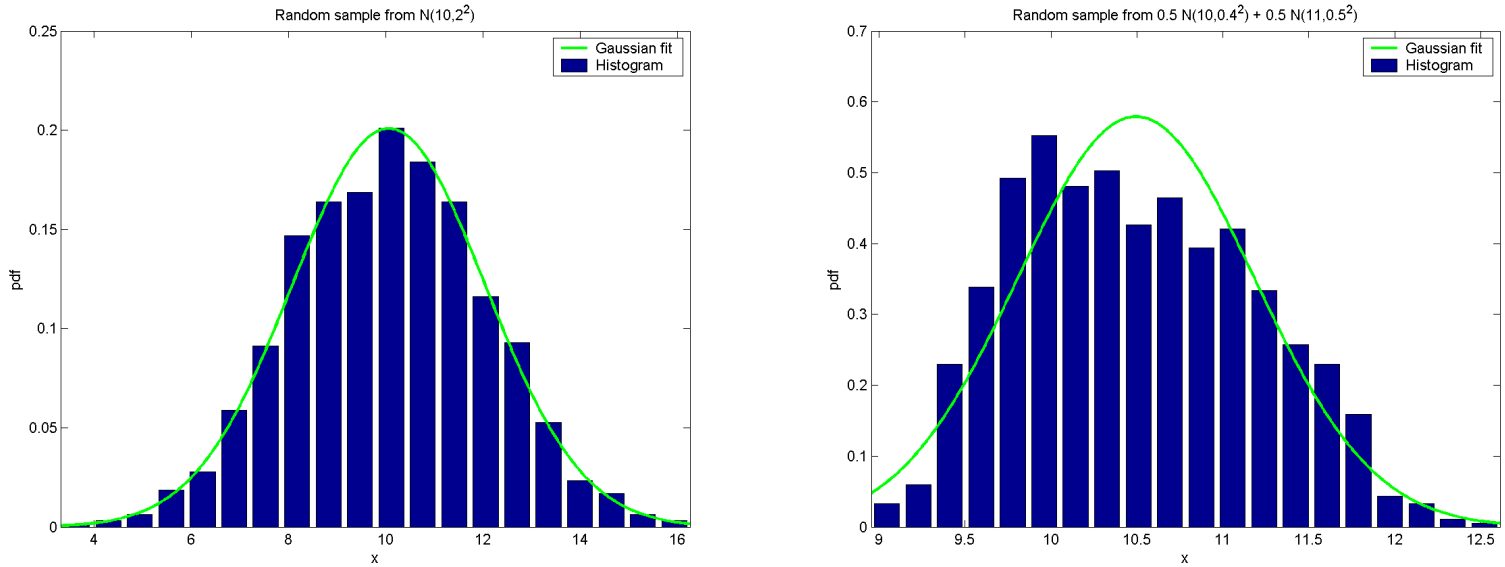
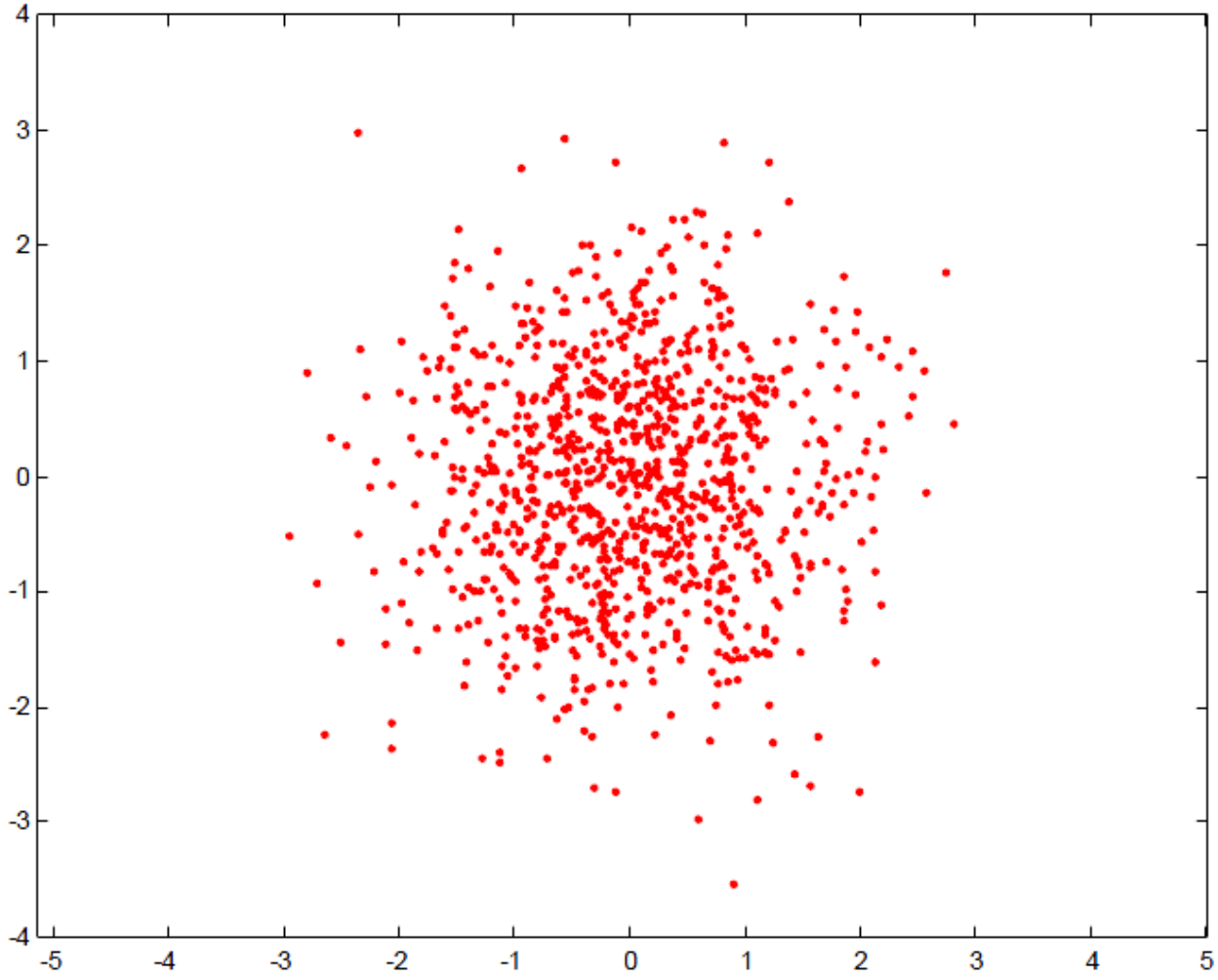


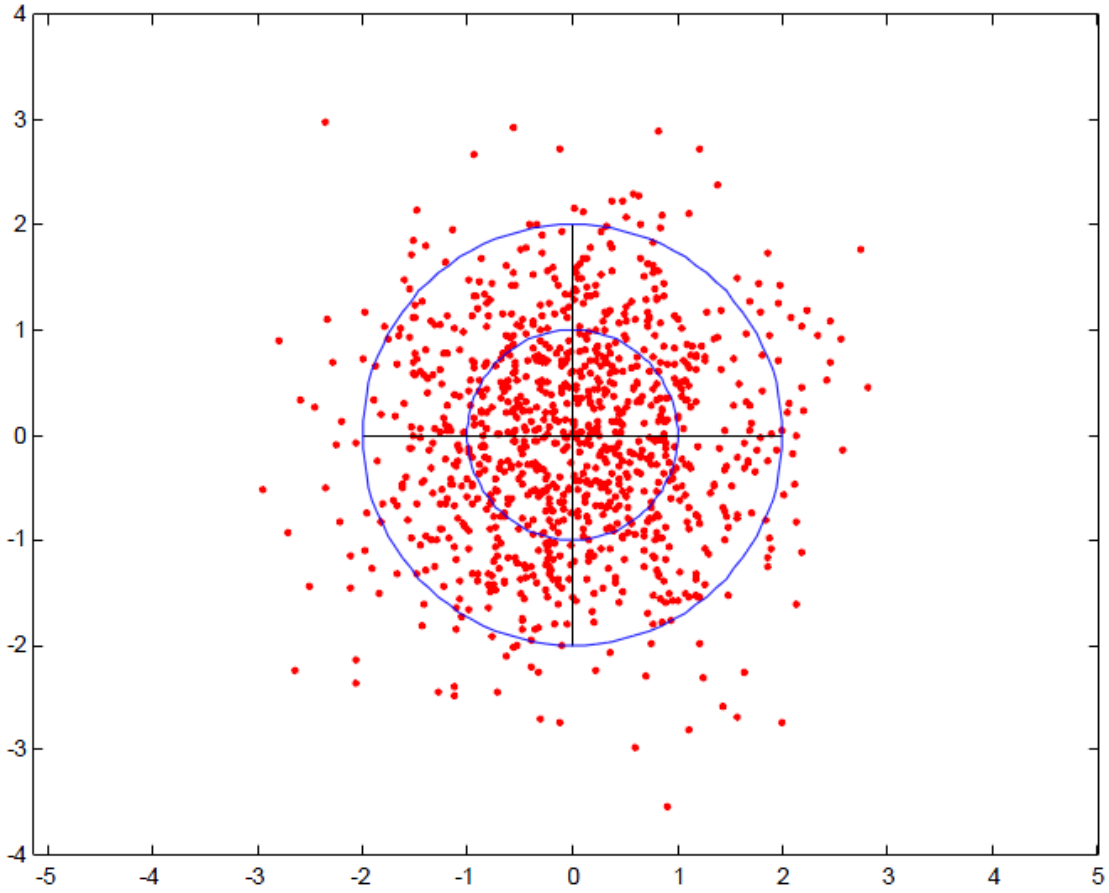
Figure: Gaussian density estimation examples.

2D Gaussian



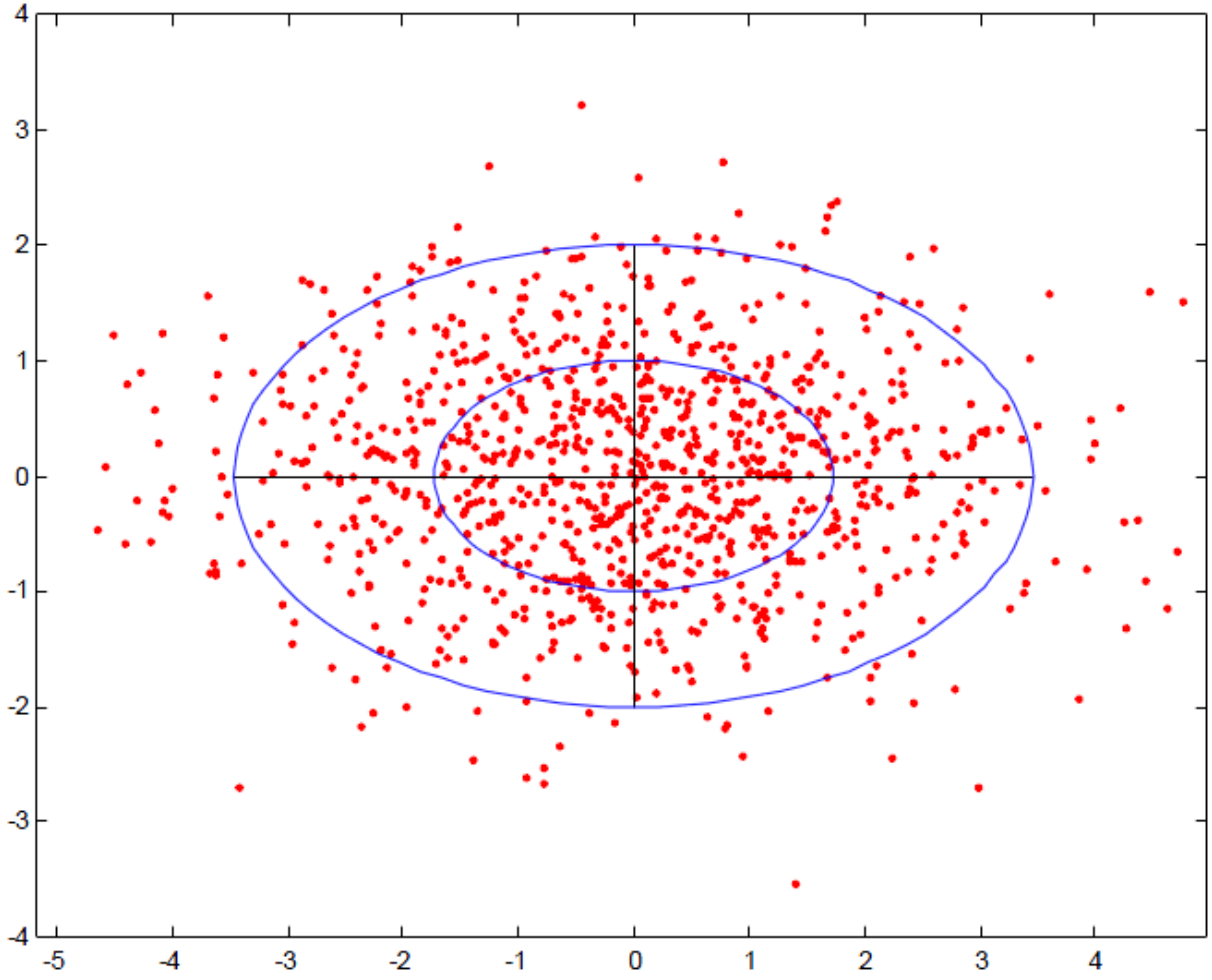
2D Gaussian

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



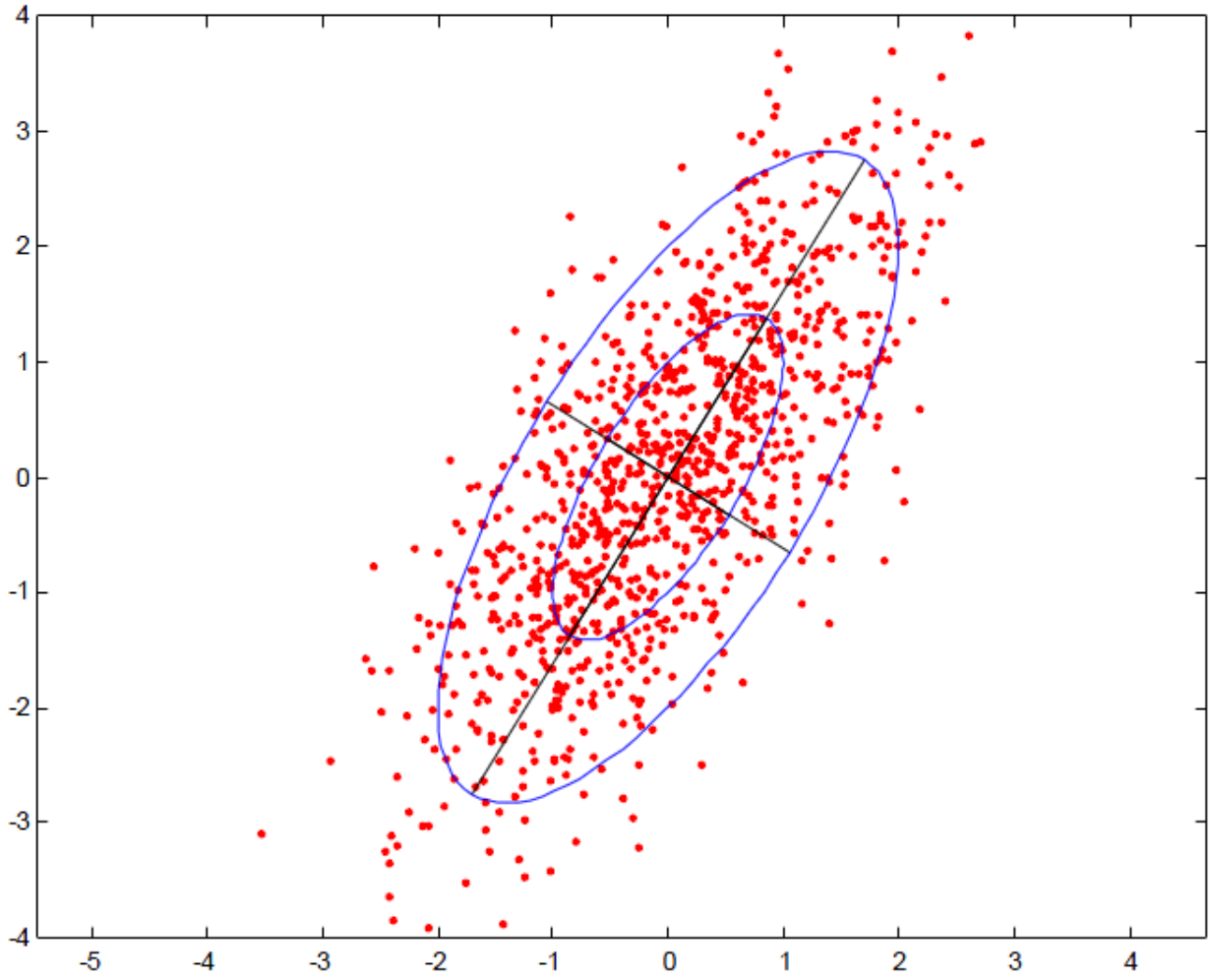
2D Gaussian

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



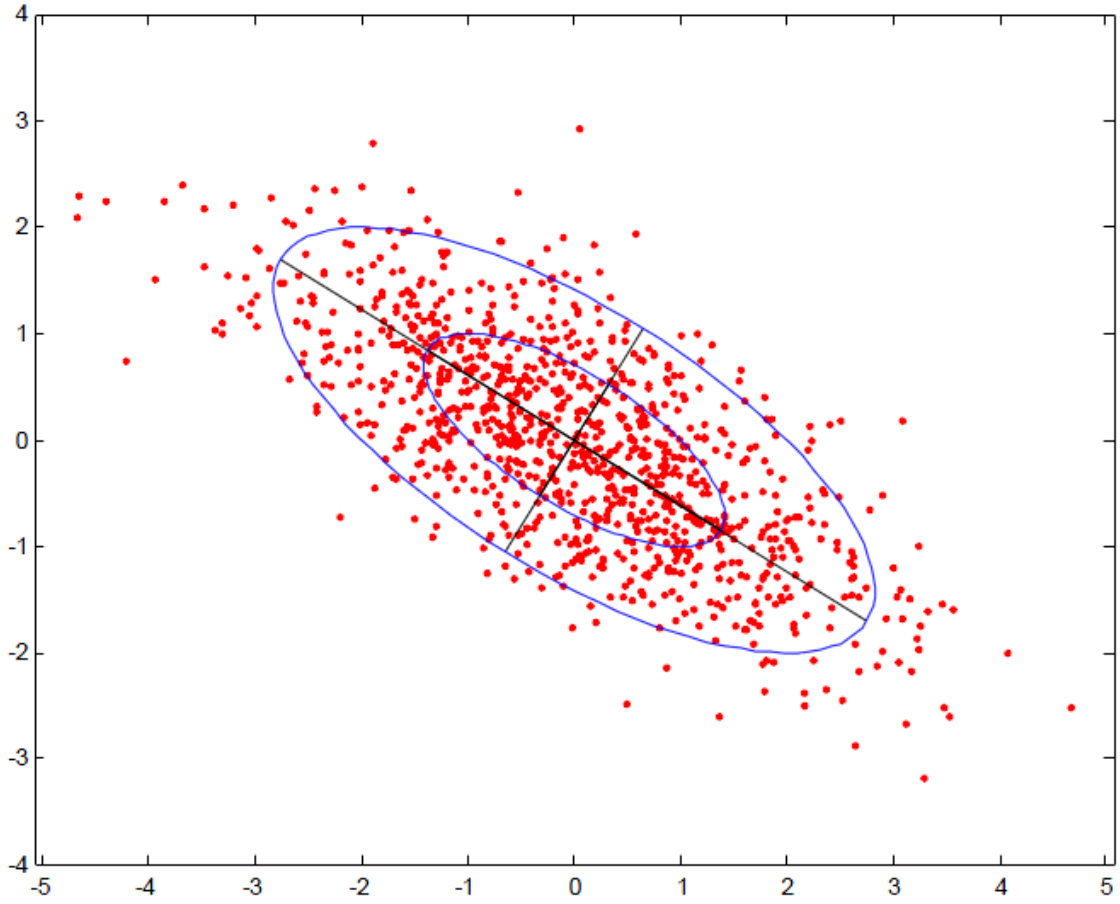
2D Gaussian

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



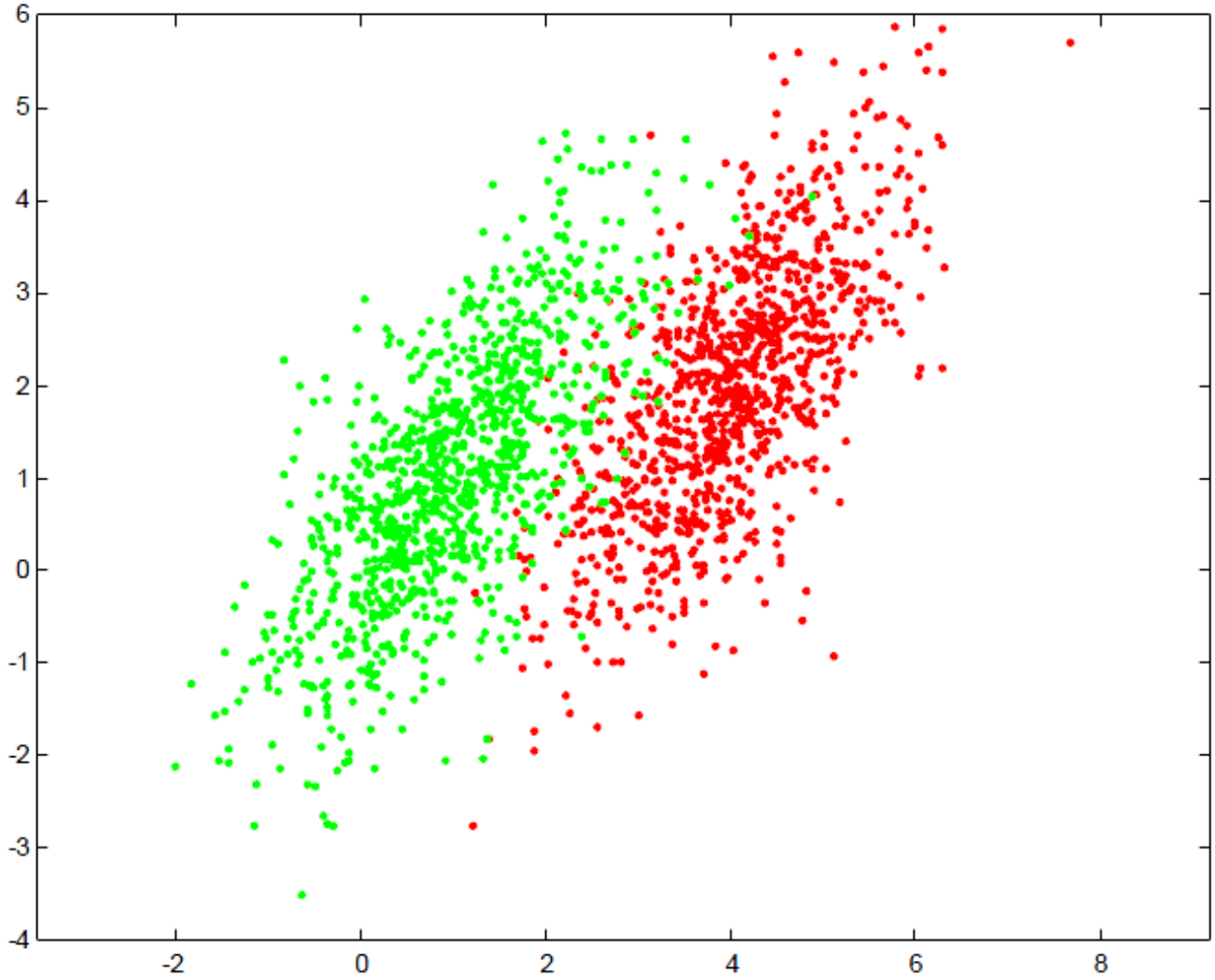
2D Gaussian

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



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.

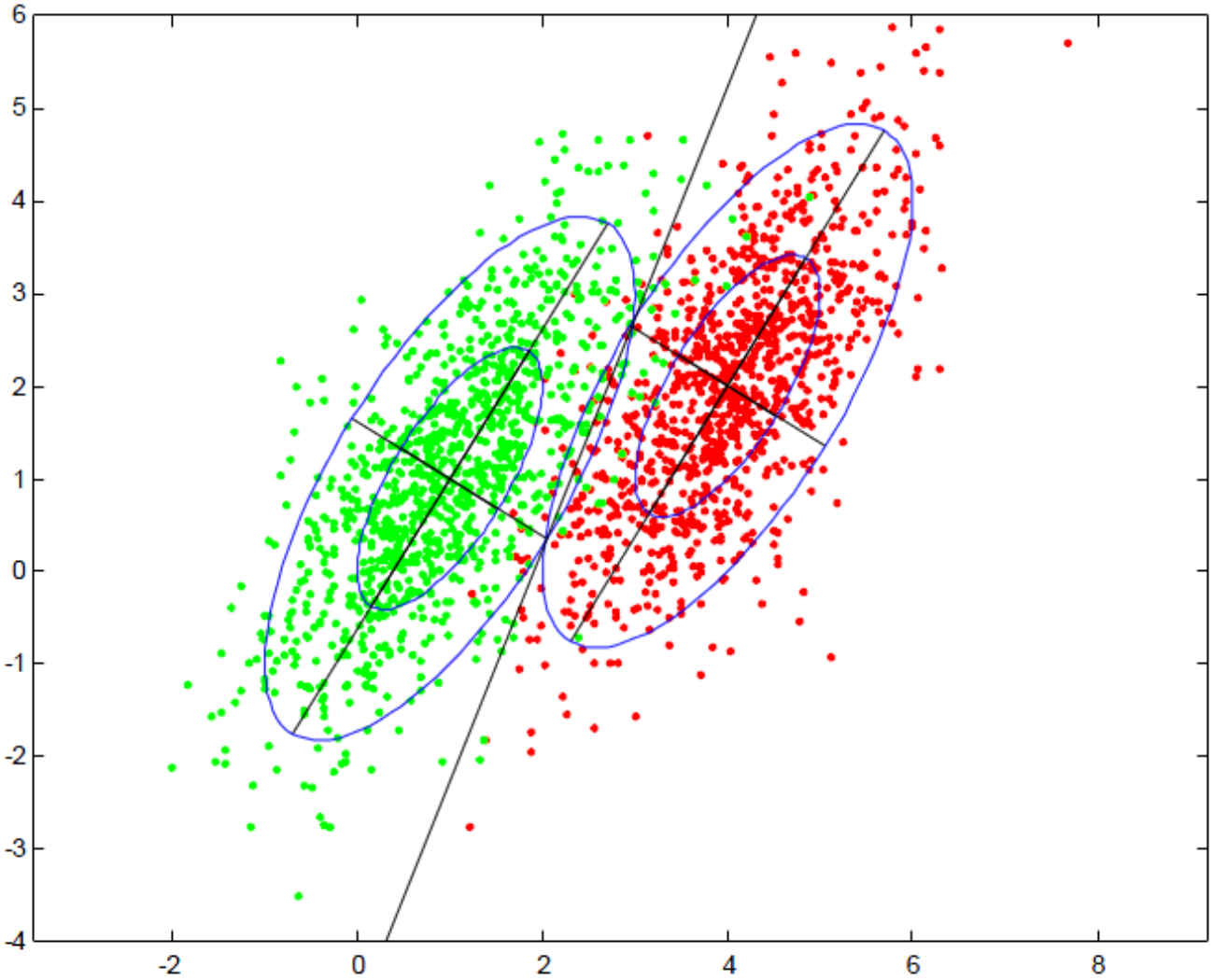
2D Example

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:

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

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

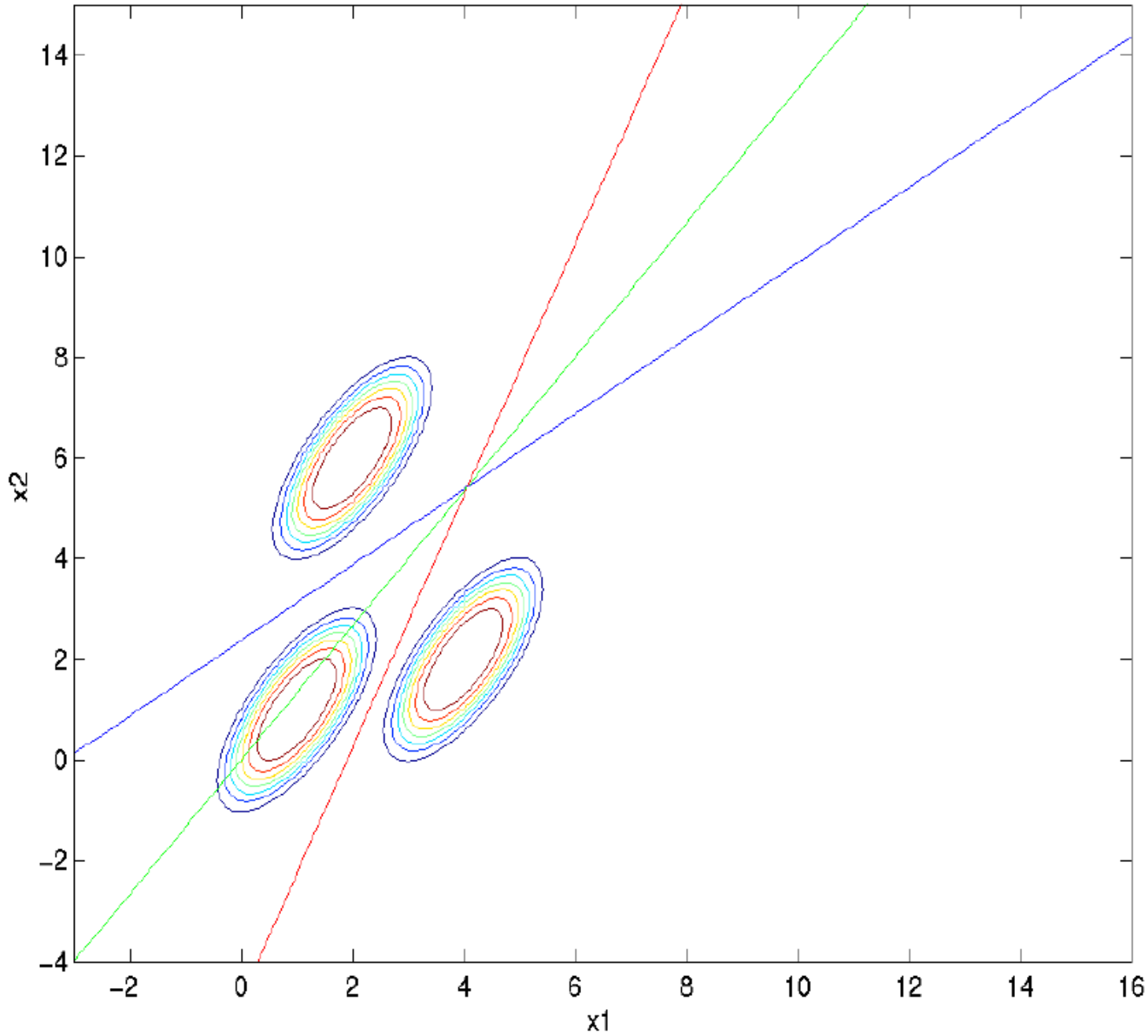
$$p(\mathbf{x}|\mathcal{C}_3) : \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \in N\left(\begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}\right)$$

- 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)

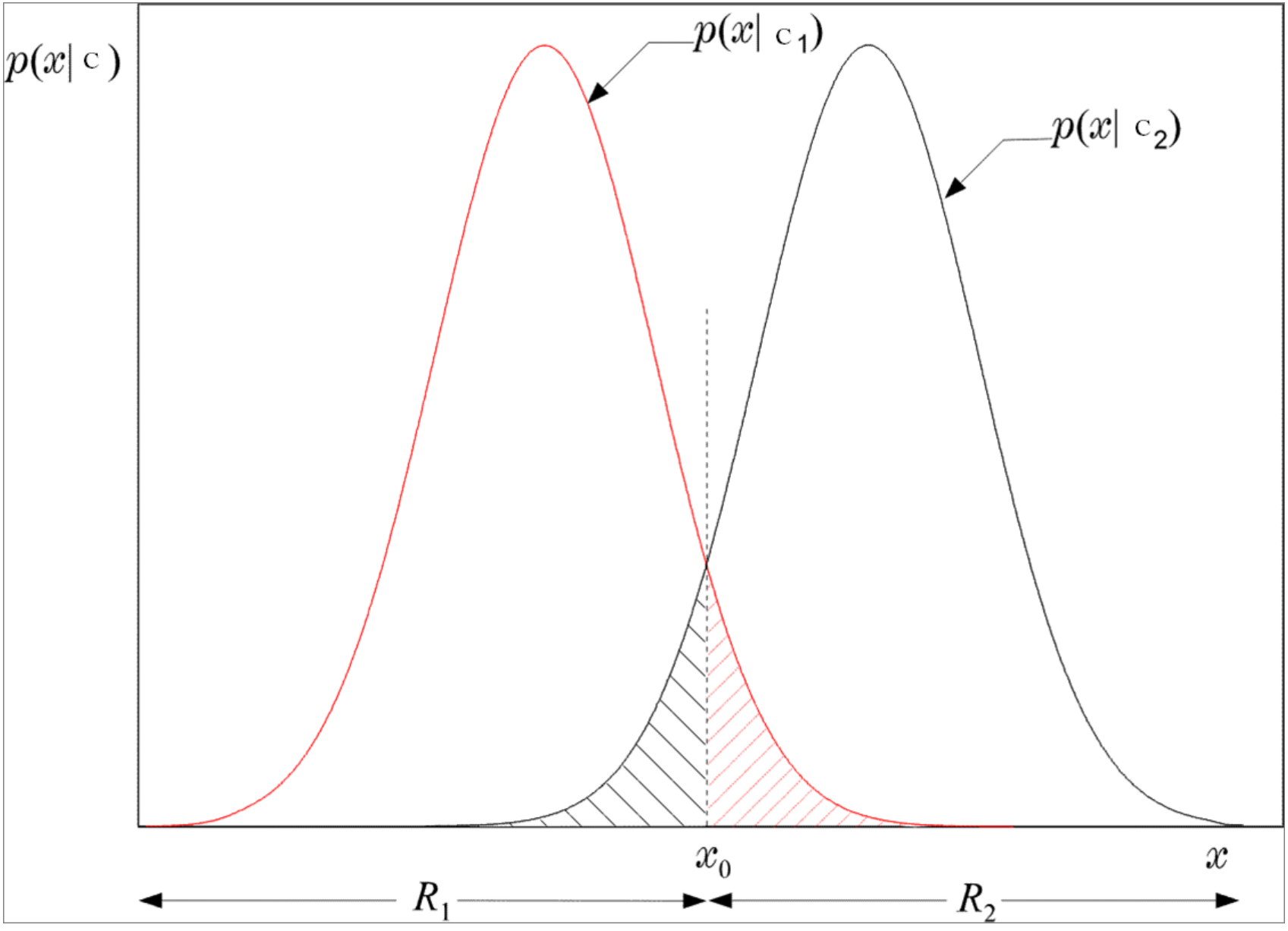
- Given $\mathbf{x}$ classify it according to the rule
  - if $p(\mathcal{C}_1|\mathbf{x}) > p(\mathcal{C}_2|\mathbf{x}) \quad \mathbf{x} \rightarrow \mathcal{C}_1$
  - if $p(\mathcal{C}_2|\mathbf{x}) > p(\mathcal{C}_1|\mathbf{x}) \quad \mathbf{x} \rightarrow \mathcal{C}_2$
- Equivalently: classify $\mathbf{x}$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

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

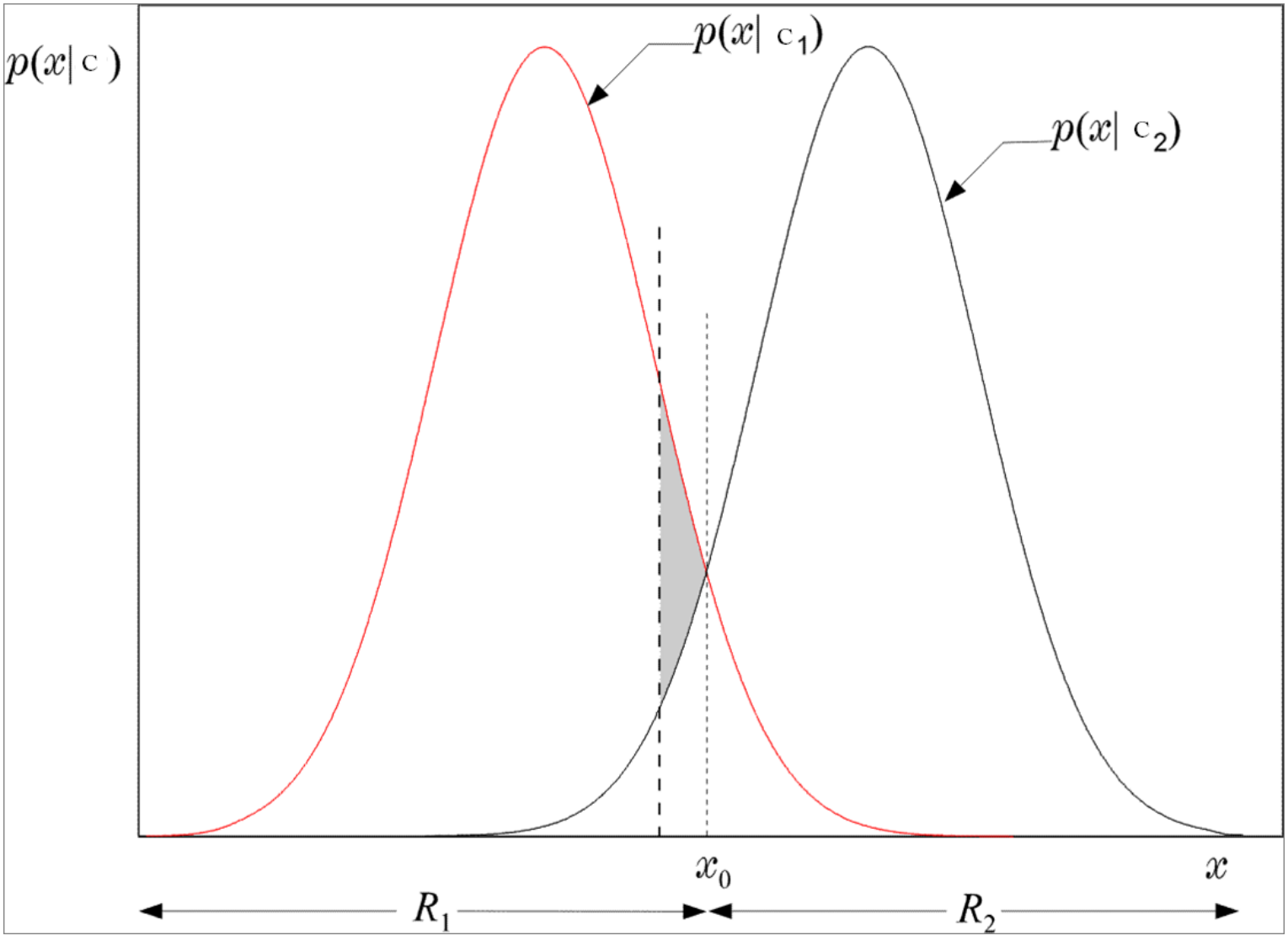
Classifiers based on Bayes Decision Theory



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)

Taxonomy of classification methods

- generative models:
  - first determine the class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$
  - infer the prior class probabilities $p(\mathcal{C}_k)$
  - Use Bayes' theorem in the form

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

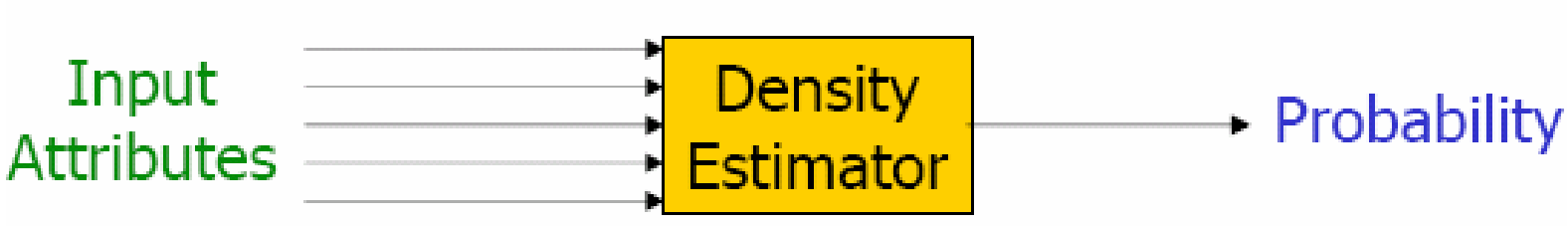
    to find the posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$
  - use these posterior probabilities to make class assignments, assigning each new $\mathbf{x}$ to one of the classes
- discriminative models:
  - infer directly $p(\mathcal{C}_k|\mathbf{x})$
  - use these posterior probabilities to make class assignments, assigning each new $\mathbf{x}$ to one of the classes
- models that find directly the discriminant function, mapping each input $\mathbf{x}$ directly onto a class label.

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

Density Estimation

- A Density Estimator learns a mapping from a set of attributes to a Probability



- Often known as parameter estimation if the distribution form is specified
  - Binomial, Gaussian ...
- Four important issues:
  - Nature of the data (iid, correlated, ...)
  - Objective function (MLE, MAP, ...)
  - Algorithm (simple algebra, gradient methods, EM, ...)
  - Evaluation scheme (likelihood on test data, predictability, consistency, ...)

Estimation Schemes

$(x_1^1, \ldots, x_1^K)$
$(x_2^1, \ldots, x_2^K)$
$\ldots$
$(x_n^1, \ldots, x_n^K)$

Objective Function

- **Maximum likelihood**
- **Bayesian**
- **Conditional likelihood**
- **Margin**
- **...**

Algorithm

- **Analytical**
- **Gradient**
- **EM**
- **Sampling**
- **...**

Score param

- 10^{-5}
- 10^{-3}
- 10^{-15}
- ...

Parameter Learning from iid Data

- Goal: estimate distribution parameters $\boldsymbol{\theta}$ from a dataset of N independent, identically distributed (iid), fully observed, training cases

$$D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$$

- Maximum likelihood estimation (MLE)
  1. One of the most common estimators
  2. With iid and full-observability assumption, write $L(\boldsymbol{\theta})$ as the likelihood of the data:

$$L(\theta) \quad = P(\mathbf{x}_1, \ldots, \mathbf{x}_N; \theta)$$

  3. pick the setting of parameters most likely to have generated the data we saw:

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta})$$

Linear Models for Classification

Bayes Linear Classifier

We will continue with a probabilistic view of classification and show how models with linear decision boundaries arise from simple assumptions about the distribution of the data.

- we 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:

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

Linear Models for Classification

Bayes Linear Classifier

We will continue with a probabilistic view of classification and show how models with linear decision boundaries arise from simple assumptions about the distribution of the data.

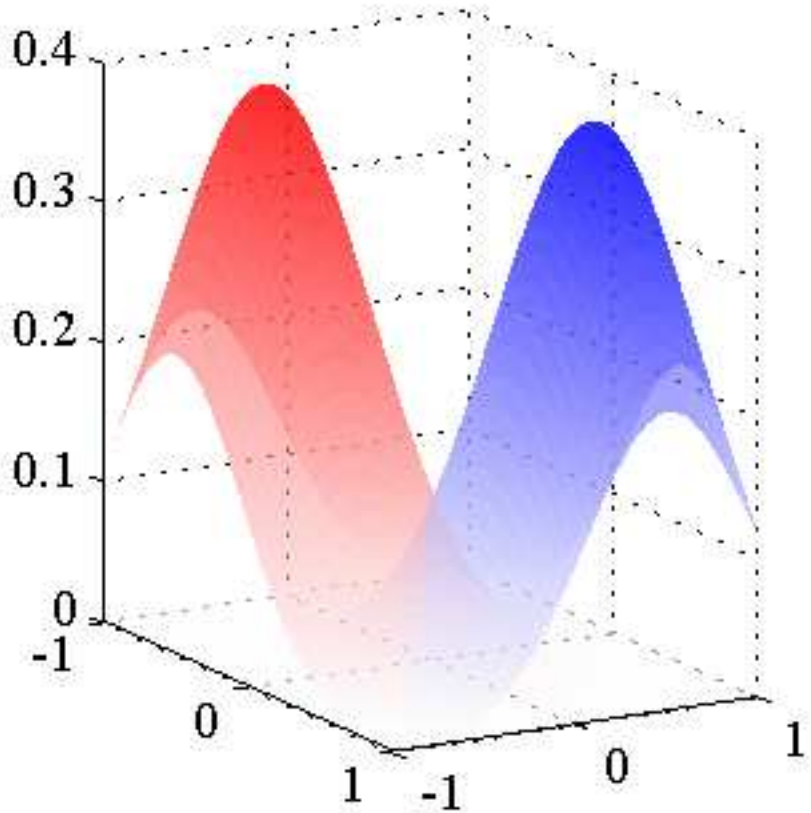
- Let us assume that the class-conditional densities are Gaussian and then explore the resulting form for the posterior probabilities.
- assume that all classes share the same covariance matrix. Thus the density for class $\mathcal{C}_k$is given by

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

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

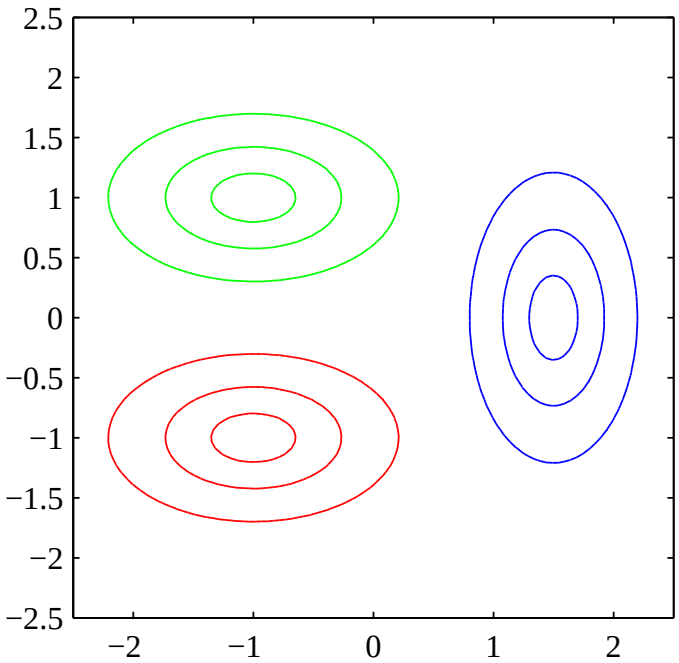
Linear Models for Classification

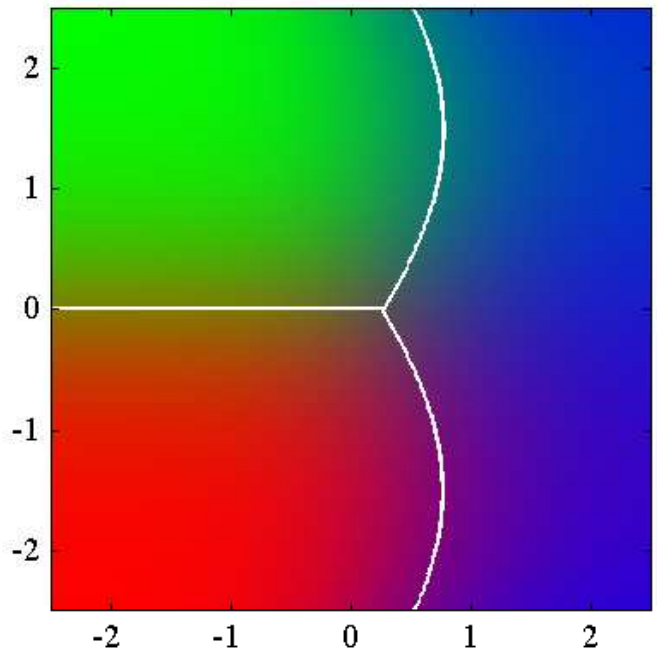
Bayes Linear Classifier



Quadratic discriminant Model

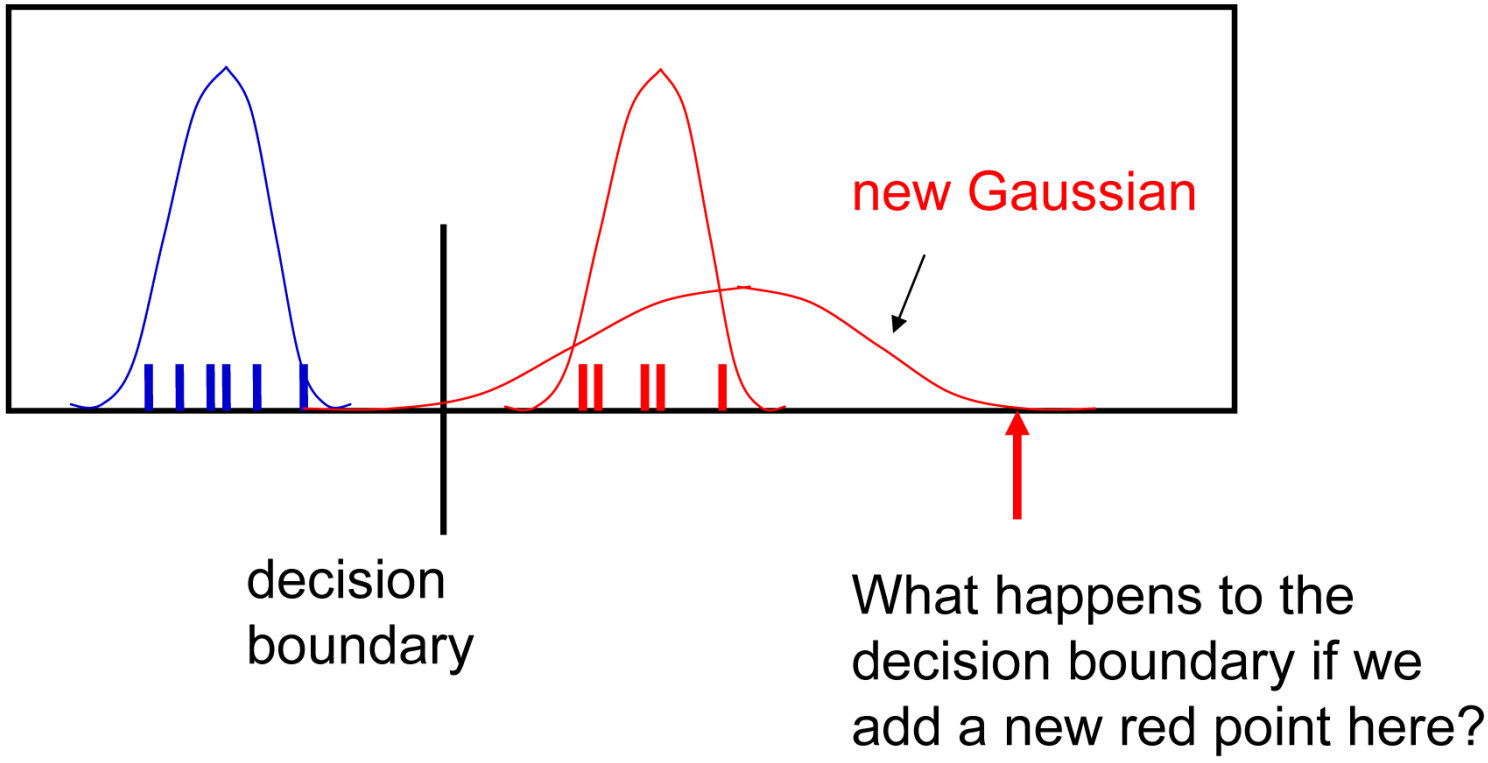
The decision surface is planar when the covariance matrices are the same and quadratic when they are not.





Maximum likelihood solution

Gaussian fitting using maximum likelihood estimation is not robust



For generative fitting, the red mean moves rightwards but the decision boundary moves leftwards! If you really believe it's Gaussian data this is sensible.

Naive Bayes

Let's learn classifiers by learning $P(y|\mathbf{x})$

Suppose y =Wealth, $\mathbf{x}$ =<Gender, HoursWorked>

Gender	HrsWorked	P(rich | G,HW)	P(poor | G,HW)
F	<40.5	.09	.91
F	>40.5	.21	.79
M	<40.5	.23	.77
M	>40.5	.38	.62

Naive Bayes

How many parameters must we estimate?

Suppose $\mathbf{x} =< x_1, \ldots x_n >$ where x_i and y are boolean RV's

To estimate $P(y|x_1, x_2, \ldots x_n)$

Gender	HrsWorked	P(rich | G,HW)	P(poor | G,HW)
F	<40.5	.09	.91
F	>40.5	.21	.79
M	<40.5	.23	.77
M	>40.5	.38	.62

And if we have 30 x_i's instead of 2?

Naive Bayes

Can we reduce params by using Bayes Rule?

Suppose $\mathbf{x} =< x_1, \ldots x_n >$ where x_i and y are boolean RV's.
$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$

Naive Bayes

Bayes Rule

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})}$$

Which is shorthand for:

$$(\forall i, j) P(y = y_i | \mathbf{x} = \mathbf{x}_j) = \frac{P(\mathbf{x} = \mathbf{x}_j | y = y_i) P(y_i)}{P(\mathbf{x} = \mathbf{x}_j)}$$

Equivalently:

$$(\forall i, j) P(y = y_i | \mathbf{x} = \mathbf{x}_j) = \frac{P(\mathbf{x} = \mathbf{x}_j | y = y_i) P(y_i)}{\sum_k P(\mathbf{x} = \mathbf{x}_j | y = y_k) P(y_k)}$$

Naive Bayes

Assumption

Naïve Bayes assumes

$$P(x_1, x_2, \cdots, x_n|y) = P(x_1|y)P(x_2|y)\cdots P(x_n|y)$$

i.e., that x_i and x_j are conditionally independent given y, for all $i \neq j$

Naive Bayes

Conditional Independence

Definition: X is conditionally independent of Y given Z, if the probability distribution governing X is independent of the value of Y, given the value of Z

$$(\forall i, j, k) P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

Which we often write

$$P(X|Y, Z) = P(X|Z)$$

e.g $P(thunder|rain, lightning) = P(thunder|lightning)$

Naive Bayes

Naïve Bayes uses assumption that the x_i are conditionally independent, given y.
Given this assumption, then:

$$P(x_1, x_2|y) = P(x_1|x_2, y)P(x_2|y) = P(x_1|y)P(x_2|y)P$$

in general: $P(x_1 \cdots x_n|y) = \prod_i P(x_i|y)$
How many parameters to describe $P(x_1 \cdots x_n|y)$? $P(y)$?

- Without conditional indep assumption?
- With conditional indep assumption?

Naive Bayes

Naïve Bayes in a Nutshell

Bayes rule:

$$P(y = y_k|x_1 \cdots x_n) = \frac{P(y = y_k)P(x_1 \cdots x_n|y = y_k)}{\sum_j P(y = y_j)P(x_1 \cdots x_n|y = y_j)}$$

Assuming conditional independence among x_i's:

$$P(y = y_k|x_1 \cdots x_n) = \frac{P(y = y_k)\prod_i P(x_i|y = y_k)}{\sum_j P(y = y_j)\prod_i P(x_i|y = y_j)}$$

So, classification rule for $\mathbf{x}^{new} =< x_1 x_2 \cdots x_n >$ is

$$y^{new} \leftarrow \arg\max_{y_k} P(y = y_k)\prod_i P(x_i^{new}|y = y_k)$$

Naive Bayes

Naïve Bayes Algorithm – discrete x_i

- Train Naïve Bayes (examples) for each* value y_k
  estimate $\pi_k = P(y = y_k)$
  for each* value x_{ij} of each attribute x_i
  estimate $\theta_{ijk} = P(x_i = x_{ij}|y = y_k)$
- Classify $\mathbf{x}^{new}$
  $y^{new} \leftarrow \arg\max_{y_k} P(y = y_k) \prod_i P(x_i^{new}|y = y_k)$
  $y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i \theta_{ijk}$

* probabilities must sum to 1, so need estimate only $n - 1$ parameters...

Naive Bayes

Estimating Parameters: y, x_i discrete-valued

Maximum likelihood estimates (MLE's):

$$\hat{\pi} = \hat{P}(y = y_k) = \frac{\#D\{y = y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}(x_{ij}|y = y_k) = \frac{\#D\{x_i = x_{ij} \wedge y = y_k\}}{\#D\{y = y_k\}}$$

Naive Bayes

Naïve Bayes: Subtlety #1

If unlucky, our MLE estimate for $P(x_i|y)$ might be zero. (e.g., $x_{373} =$ Birthday_Is_January_30_1990)

- Why worry about just one parameter out of many?
- What can be done to avoid this?

Naive Bayes

Estimating Parameters

- Maximum Likelihood Estimate (MLE): choose θ that maximizes probability of observed data $\mathcal{D}$

$$\hat{\theta} = \arg\max_{\theta} P(\mathcal{D}|\theta)$$

- Maximum a Posteriori (MAP) estimate: choose θ that is most probable given prior probability and the data

$$\hat{\theta} = \arg\max_{\theta} P(\theta|\mathcal{D}) = \arg\max_{\theta} \frac{\mathcal{P}(\mathcal{D}|\theta)\mathcal{P}(\theta)}{\mathcal{P}(\mathcal{D})}$$

Naive Bayes

Estimating Parameters

Maximum likelihood estimates (MLE's):

$$\hat{\pi} = \hat{P}(y = y_k) = \frac{\#D\{y = y_k\}}{|D|}$$

$$\hat{\theta}_{ijk} = \hat{P}(x_{ij}|y = y_k) = \frac{\#D\{x_i = x_{ij} \wedge y = y_k\}}{\#D\{y = y_k\}}$$

MAP estimates (Beta, Dirichlet priors):

$$\hat{\pi} = \hat{P}(y = y_k) = \frac{\#D\{y = y_k\} + \alpha_k}{|D| + \sum_m \alpha_m}$$

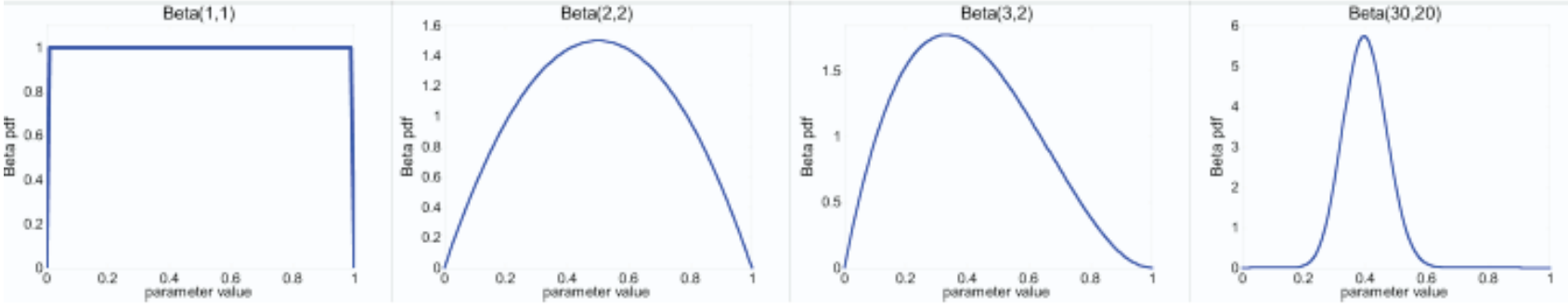
$$\hat{\theta}_{ijk} = \hat{P}(x_{ij}|y = y_k) = \frac{\#D\{x_i = x_{ij} \wedge y = y_k\} + \alpha'_k}{\#D\{y = y_k\} + \sum_m \alpha'_m}$$

Beta prior distribution – P(θ)

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

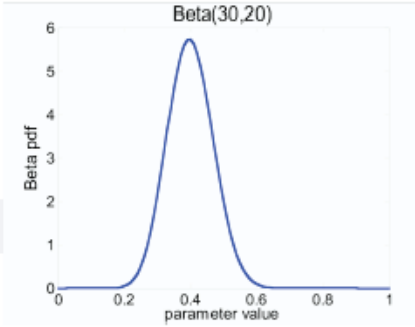
Mean:

Mode:



- Likelihood function: $P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$
- Posterior: $P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)P(\theta)$

MAP for Beta distribution



$$P(\theta \mid \mathcal{D}) = \frac{\theta^{\beta_H+\alpha_H-1}(1-\theta)^{\beta_T+\alpha_T-1}}{B(\beta_H+\alpha_H, \beta_T+\alpha_T)} \sim Beta(\beta_H+\alpha_H, \beta_T+\alpha_T)$$

- MAP: use most likely parameter:

$$\widehat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D}) =$$

- Beta prior equivalent to extra thumbtack flips
- As $N \to \infty$, prior is "forgotten"
- But, for small sample size, prior is important!

Naive Bayes

Dirichlet distribution

- number of heads in N flips of a two-sided coin
  - follows a binomial distribution
  - Beta is a good prior (conjugate prior for binomial)
- what it's not two-sided, but k-sided?
  - follows a multinomial distribution
  - Dirichlet distribution is the conjugate prior

$$P(\theta_1, \theta_2, ...\theta_K) = \frac{1}{B(\alpha)} \prod_i^K \theta_i^{(\alpha_1 - 1)}$$

Lejeune Dirichlet

Johann Peter Gustav Lejeune Dirichlet

Born	13 February 1805 Düren, French Empire
Died	5 May 1859 (aged 54) Göttingen, Hanover
Residence	Germany
Nationality	German
Fields	Mathematician
Institutions	University of Berlin University of Breslau University of Göttingen
Alma mater	University of Bonn
Doctoral advisor	Simeon Poisson Joseph Fourier
Doctoral students	Ferdinand Eisenstein Leopold Kronecker Rudolf Lipschitz Carl Wilhelm Borchardt
Known for	Dirichlet function Dirichlet eta function

Naive Bayes

Naïve Bayes: Subtlety #2

Often the x_i are not really conditionally independent

- We use Naïve Bayes in many cases anyway, and it often works pretty well
  - often the right classification, even when not the right probability (see [Domingos&Pazzani, 1996])
- What is effect on estimated $P(y|\mathbf{x})$?
  - Special case: what if we add two copies: $x_i = x_k$

Naive Bayes

Learning to classify text documents

- Classify which emails are spam?
- Classify which emails promise an attachment?
- Classify which web pages are student home pages?

How shall we represent text documents for Naïve Bayes?

Naive Bayes

Baseline: Bag of Words Approach

aardvark	0
about	2
all	2
Africa	1
apple	0
anxious	0
...	
gas	1
...	
oil	1
...	
Zaire	0

Naive Bayes

Learning to classify text documents

Target concept $Interesting? : Document \rightarrow \{+, -\}$

1. Represent each document by vector of words
   - one attribute per word position in document
2. Learning: Use training examples to estimate
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Naive Bayes conditional independence assumption

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

where $P(a_i = w_k|v_j)$ is probability that word in position i is w_k, given v_j

one more assumption:
$P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

Naive Bayes

Learning to classify text documents

Twenty NewsGroups

Given 1000 training documents from each group
Learn to classify new documents according to which newsgroup it came from

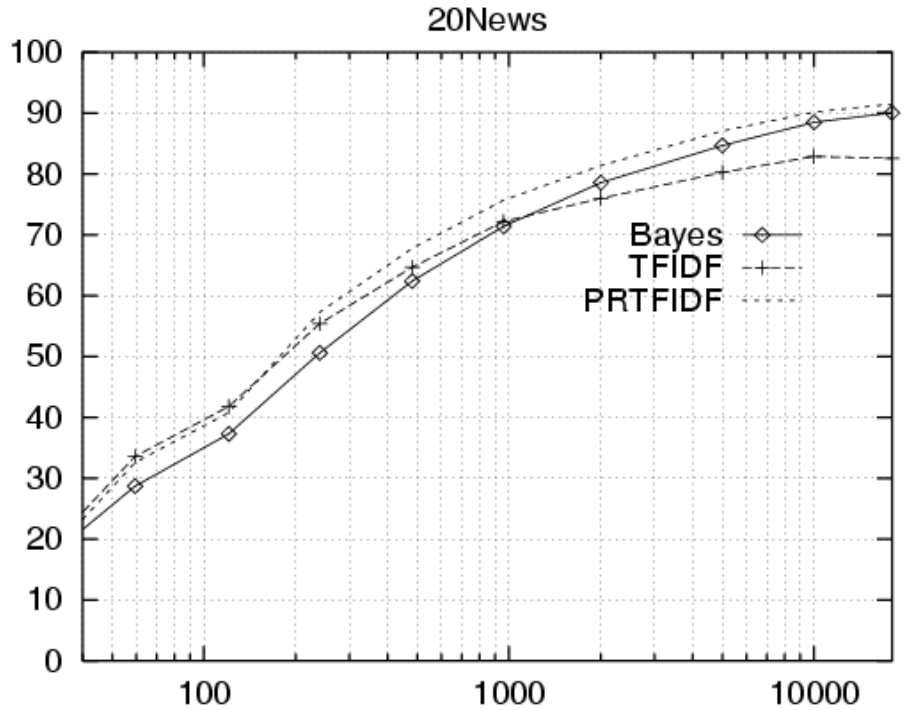
comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% classification accuracy

Naive Bayes

Learning to classify text documents

Learning Curve for 20 Newsgroups

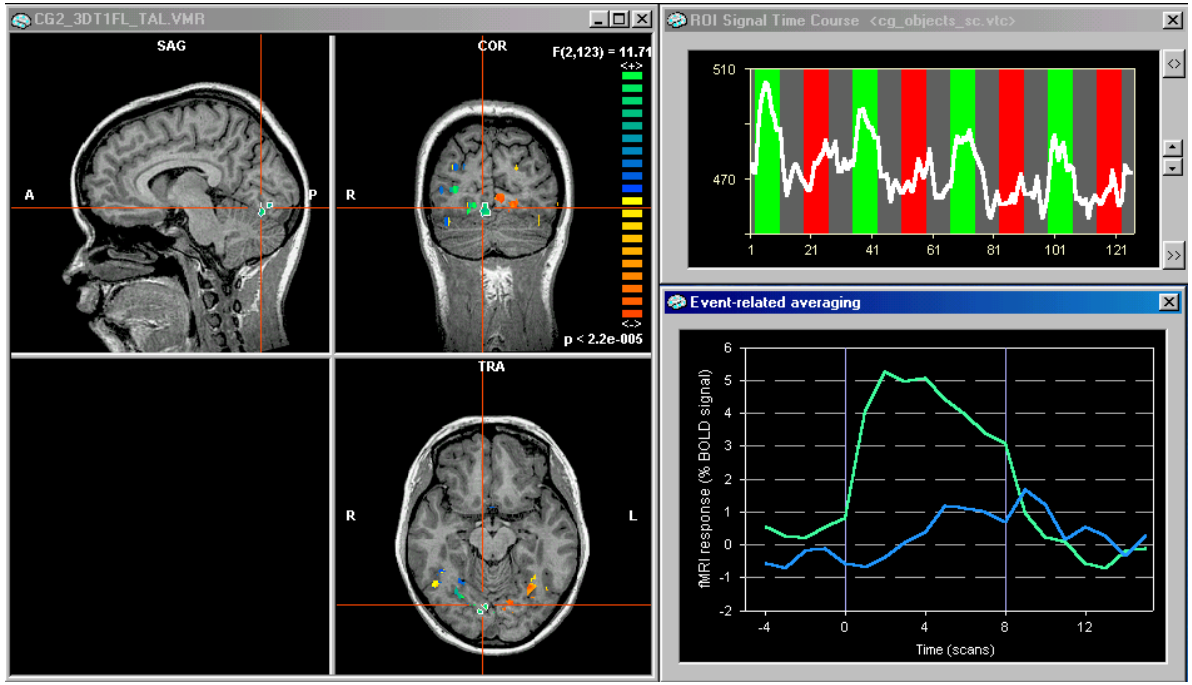


Accuracy vs. Training set size (1/3 withheld for test)

Naive Bayes

What if we have continuous x_i ?

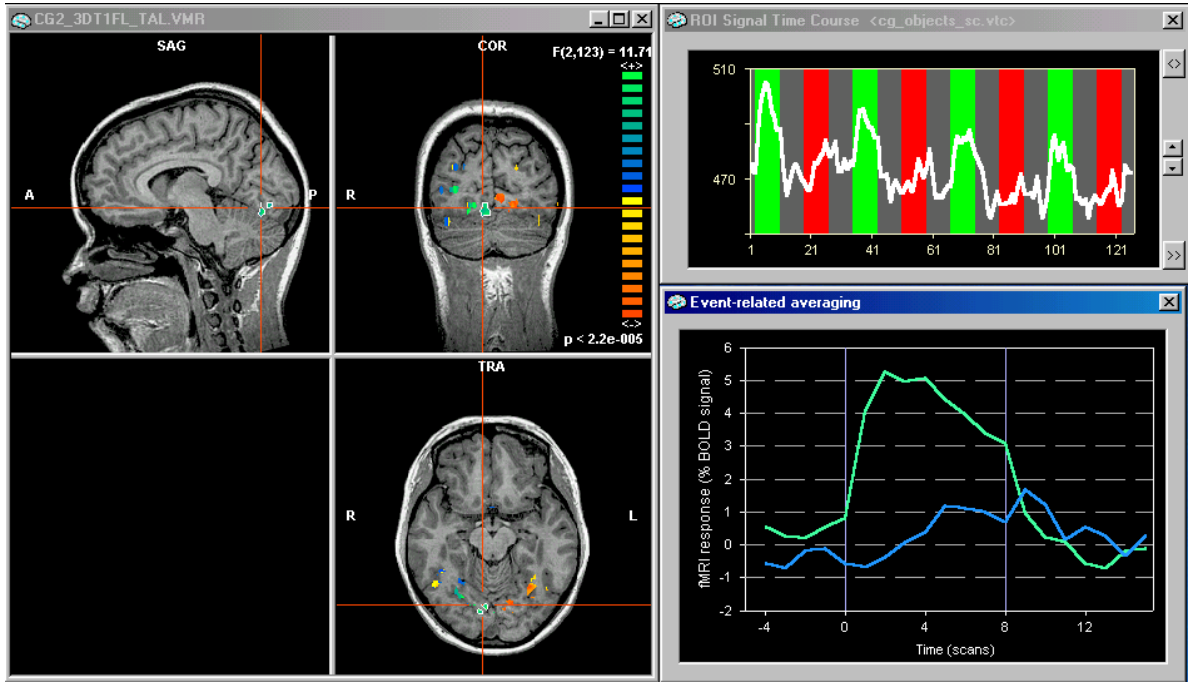
E.g., image classification: x_i is ith pixel



Naive Bayes

What if we have continuous x_i ?

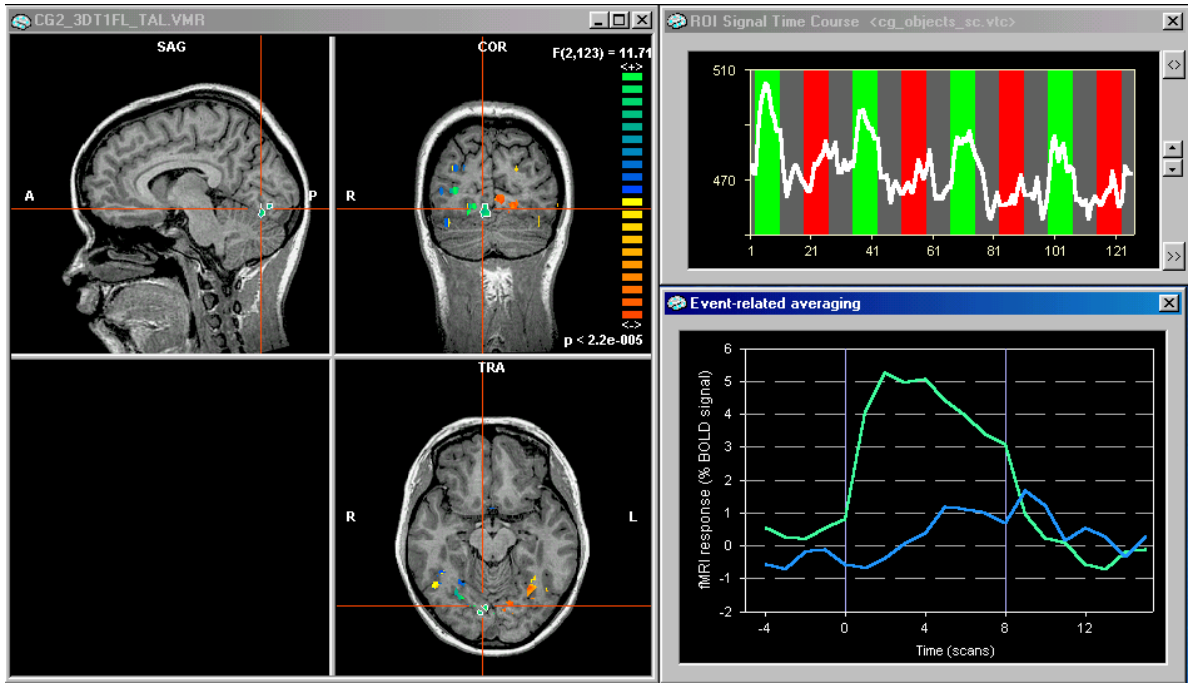
E.g., image classification: x_i is ith pixel



Naive Bayes

What if we have continuous x_i ?

E.g., image classification: x_i is ith pixel, $y =$ mental state



Still have:

$$P(y = y_k|x_1 \cdots x_n) = \frac{P(y = y_k) \prod_i P(x_i|y = y_k)}{\sum_j P(y = y_j) \prod_i P(x_i|y = y_j)}$$

Just need to decide how to represent $P(x_i|y)$

Naive Bayes

What if we have continuous x_i ?

E.g., image classification: x_i is ith pixel
Gaussian Naïve Bayes (GNB): assume

$$P(x_i = x | y = y_k) = \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{(x-\mu_{ik})^2}{2\sigma_{ik}^2}}$$

Sometimes assume variance

- is independent of y (i.e., σ_i),
- or independent of x_i (i.e., σ_k)
- or both (i.e., σ)

Naive Bayes

Gaussian Naïve Bayes Algorithm – continuous x_i (but still discrete y)

- Train Naïve Bayes (examples) for each* value y_k
  estimate $\pi_k = P(y = y_k)$
  for each attribute x_i estimate
  class conditional mean μ_{ik}, variance σ_{ik}
- Classify $\mathbf{x}^{new}$
  $y^{new} \leftarrow \arg\max_{y_k} P(y = y_k) \prod_i P(x_i^{new} | y = y_k)$
  $y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i Normal(x_i^{new}, \mu_{ik}, \sigma_{ik})$

Naive Bayes

Estimating Parameters: y discrete, x_i continuous

Maximum likelihood estimates:

$$\hat{\mu}_{ik} = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j X_i^j \delta(Y^j = y_k)$$

jth training example

ith feature

kth class

$\delta(z)=1$ if z true, else 0

$$\hat{\sigma}_{ik}^2 = \frac{1}{\sum_j \delta(Y^j = y_k)} \sum_j (X_i^j - \hat{\mu}_{ik})^2 \delta(Y^j = y_k)$$

Naive Bayes

What you should know:

- Training and using classifiers based on Bayes rule
- Conditional independence
  - What it is
  - Why it's important
- Naïve Bayes
  - What it is
  - Why we use it so much
  - Training using MLE, MAP estimates
  - Discrete variables and continuous (Gaussian)

Naive Bayes

Questions:

- What is the error will classifier achieve if Naïve Bayes assumption is satisfied and we have infinite training data?
- Can you use Naïve Bayes for a combination of discrete and real-valued x_i?
- How can we easily model just 2 of n attributes as dependent?
- What does the decision surface of a Naïve Bayes classifier look like?

Density Estimation

- Parametric techniques
  - Maximum Likelihood
  - Maximum A Posteriori
  - Bayesian Inference
  - Gaussian Mixture Models (GMM)
    - EM-Algorithm
- Non-parametric techniques
  - Histogram
  - Parzen Windows
  - k-nearest-neighbor rule

Density Estimation

Non-parametric Techniques

- Common parametric forms rarely fit the densities encountered in practice.
- Classical parametric densities are unimodal, whereas many practical problems involve multimodal densities.
- Non-parametric procedures can be used with arbitrary distributions and without the assumption that the form of the underlying densities are known

Histograms

- Conceptually most simple and intuitive method to estimate a p.d.f. is a histogram.
- The range of each dimension x_i of vector $\mathbf{x}$ is divided into a fixed number m of intervals
- The resulting M boxes (bins) of identical volume V count the number of points falling into each bin
- Assume we have N samples $(\mathbf{x}_i)_{i=1..n}$ and the number of points $\mathbf{x}_l$ in the $i-$th bin, b_i, is n_i. Then the histogram estimate of the density is $\hat{p}(\mathbf{x}) = \frac{n_i}{NV}$, $\mathbf{x} \in b_i$

Histograms

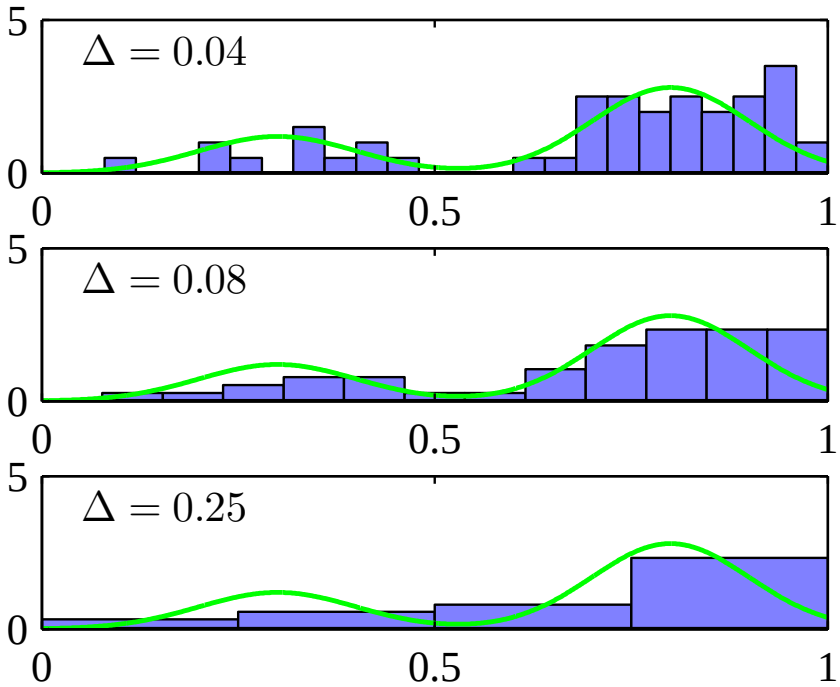
$\hat{p}(\mathbf{x})$

- ... is constant over every bin b_i.
- ... is a density function

$$\int \hat{p}(\mathbf{x})dx = \sum_{i=1}^{M} \int_{b_i} \frac{n_i}{NV} dx = 1$$

- The number of bins M and their starting positions are 'parameters'. However only the choice of M is critical. It plays the role of a smoothing parameter.

Histograms
Example



Density Estimation

Non-parametric Techniques

Histogram Approach

- Histogram p.d.f. estimator is very efficient since it can be computed online (only update counters, no need to keep all data)
- One obvious problem is that the estimated density has discontinuities that are due to the bin edges rather than any property of the underlying distribution that generated the data.
- Usefulness is limited to low dimensional vectors, since number of bins, M, grows exponentially with data's dimensionality If we divide each variable in a D-dimensional space into M bins, then the total number of bins will be M^D.
  - This exponential scaling with D is an example of the curse of dimensionality. In a space of high dimensionality, the quantity of data needed to provide meaningful estimates of local probability density would be prohibitive.

Parzen Windows: Motivation

- Let us suppose that observations are being drawn from some unknown probability density $p(\mathbf{x})$ in some D-dimensional space, and we wish to estimate the value of $p(\mathbf{x})$.
- Consider some small region $\mathcal{R}$ containing $\mathbf{x}$.
- Collect a data set comprising N observations drawn from $p(\mathbf{x})$.
- Each data point has a probability P of falling within $\mathcal{R}$
- the number of points falling inside the region is $K \approx NP$
- If, however, we also assume that the region $\mathcal{R}$ is sufficiently small that the probability density $p(\mathbf{x})$ is roughly constant over the region, then we have $P \approx p(\mathbf{x})V \Rightarrow p(\mathbf{x}) = \frac{K}{NV}$

Parzen Windows: Motivation

- Note that the approach depends on two contradictory assumptions
  - the region $\mathcal{R}$ is sufficiently small so that the density is approximately constant over the region
  - the region $\mathcal{R}$ is sufficiently large (in relation to the value of that density) so that the number K of points falling inside the region is sufficient for a good approximation

Density Estimation

Non-parametric Techniques

Parzen Windows: Motivation

- In order to make sure that we get a good estimate of p(x) at each point, we have to have lots of data points (instances) for any given R (or volume V). This can be done in two ways
  - We can fix V and take more and more samples in this volume. Then $K/N \to P$; however, we then estimate only an average of $p(\mathbf{x})$, not the $p(\mathbf{x})$ itself, because P can change in any region of nonzero volume
  - Alternatively, we can fix N and make $V \to 0$, so that $p(\mathbf{x})$ is constant in that region. However, in practice we have a finite number of training data, so as $V \to 0$, V will be so small that it will eventually contain no samples: $k = 0 \to p(\mathbf{x}) = 0$, a useless result!
- Therefore, $V \to 0$ is not feasible and one has to live with the fact that there will always be some variance in K/N and hence some averaging in $p(\mathbf{x})$ within the finite non-zero volume V. A compromise need to be found for V so that
  - It will be large enough to contain sufficient number of samples
  - It will be small enough to justify our assumption of $p(\mathbf{x})$ be constant within the chosen volume/region.

Parzen Windows: Motivation

- We can exploit these results in two different ways.
  - Either we can fix K and determine the value of V from the data, which gives rise to the K-nearest-neighbour technique to be discussed shortly,
  - we can fix V and determine K from the data, giving rise to the kernel approach or Parzen Windows (PW).
- It can be shown that both the K-nearest-neighbour density estimator and the kernel density estimator converge to the true probability density in the limit $N \to \infty$ provided V shrinks suitably with N, and K grows with N
  - $\lim_{i\to\infty} V_i = 0$
  - $\lim_{i\to\infty} K_i = \infty$
  - $\lim_{i\to\infty} K_i/n = 0$

Density Estimation

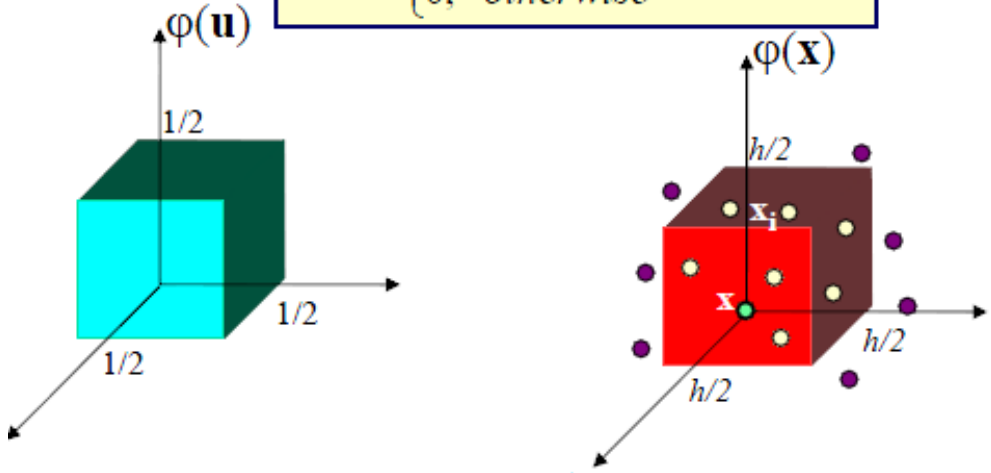
Non-parametric Techniques

Parzen Windows

- The number of samples falling into the a specified region is obtained by the help of a windowing function, hence the name Parzen windows.
  - We first assume that $\mathcal{R}$ is a d-dimensional hypercube of each side h, whose volume is then $V = (h)^d$
  - Then define a window function $\phi(u)$, called a kernel function, to count the number of samples K that fall into $\mathcal{R}$.

$$\varphi(\mathbf{u}) = \begin{cases} 1, & |\mathbf{u}_j| \le 1/2 \quad j = 1,2,\cdots,d \\ 0, & otherwise \end{cases} \Rightarrow \varphi\left(\frac{\mathbf{x}-\mathbf{x_i}}{h}\right) = 1$$

if the point $\mathbf{x_i}$ falls inside the hypercube centered at $\mathbf{x}$ with each side of h_n, 0 otherwise.



The number of samples in this hypercube is then

$$k = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

Density Estimation

Non-parametric Techniques

Parzen Windows

$$p(\mathbf{x}) \approx \frac{k_n/n}{V_n} \qquad \& \quad k = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

$$\tilde{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

Estimate of the density: number of samples falling into R, where R is centered at $\mathbf{x}$ with a width of h

The volume covered by the window function

Window function (kernel)

The width of the window function

Density Estimation

Non-parametric Techniques

Parzen Windows

- Now consider $\phi()$ as a general function, typically a smooth and continuous function, instead of a hypercube. The general expression of $p(\mathbf{x})$ remains unchanged

$$\tilde{p}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) = \frac{1}{nh^d}\sum_{i=1}^{n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

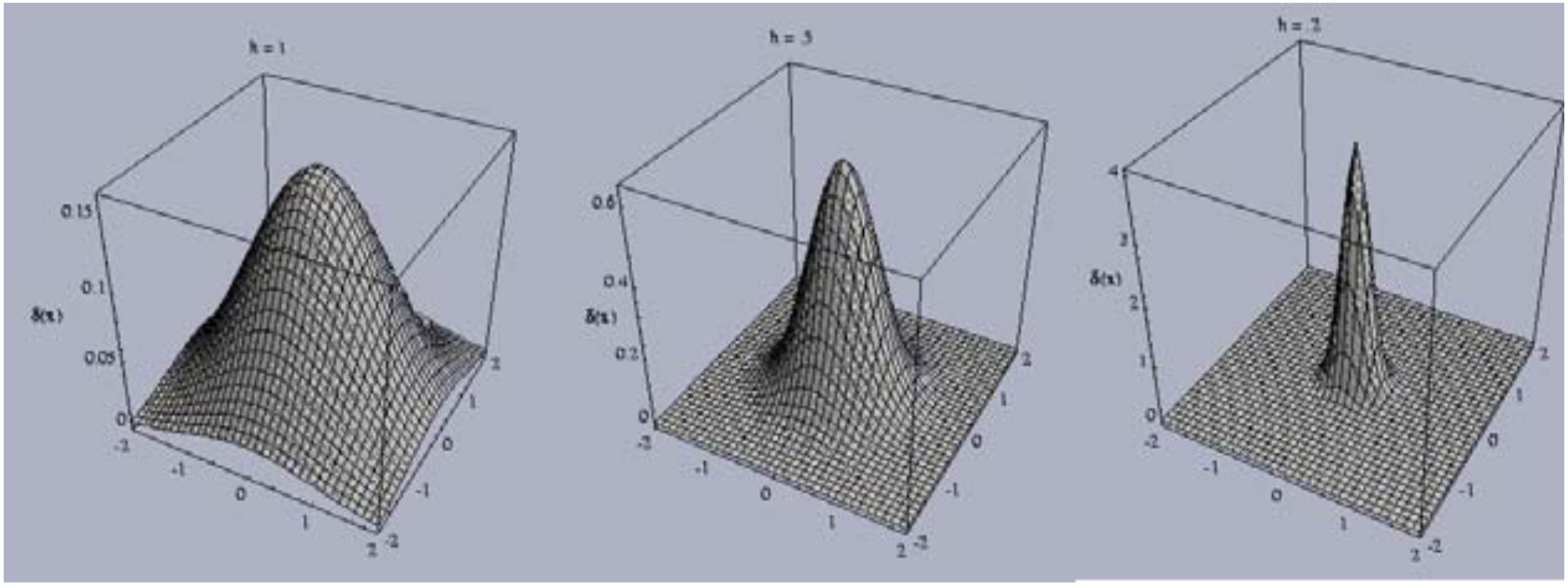
- Then $\hat{p}(\mathbf{x})$ is a superposition (interpolation) of $\phi()$s, where each $\phi()$ measures how far a given $\mathbf{x}_i$ is from $\mathbf{x}$!
  - In practice, $\mathbf{x}_i$ are the training data points and we estimate $\hat{p}(\mathbf{x})$ by interpolating the contributions of each sample data point $\mathbf{x}_i$ based on its distance from $\mathbf{x}$, the point at which we want to estimate the density. The kernel function $\phi()$ provides the numerical value of this distance
  - If $\phi()$ is itself a distribution, then $\hat{p}(\mathbf{x})$ will converge to $p(\mathbf{x})$ as N increases. A typical choice for $\phi()$ is ... the Gaussian!
- The density $p(\mathbf{x})$ is then estimated simply by a superposition of Gaussians, where each Gaussian is centered at the training data instances. The parameter h is then the variance of the Gaussian

Density Estimation

Non-parametric Techniques

Parzen Windows: Bandwidth

The choice of bandwidth is critical !



Density Estimation

Non-parametric Techniques

Parzen Windows: Bandwidth

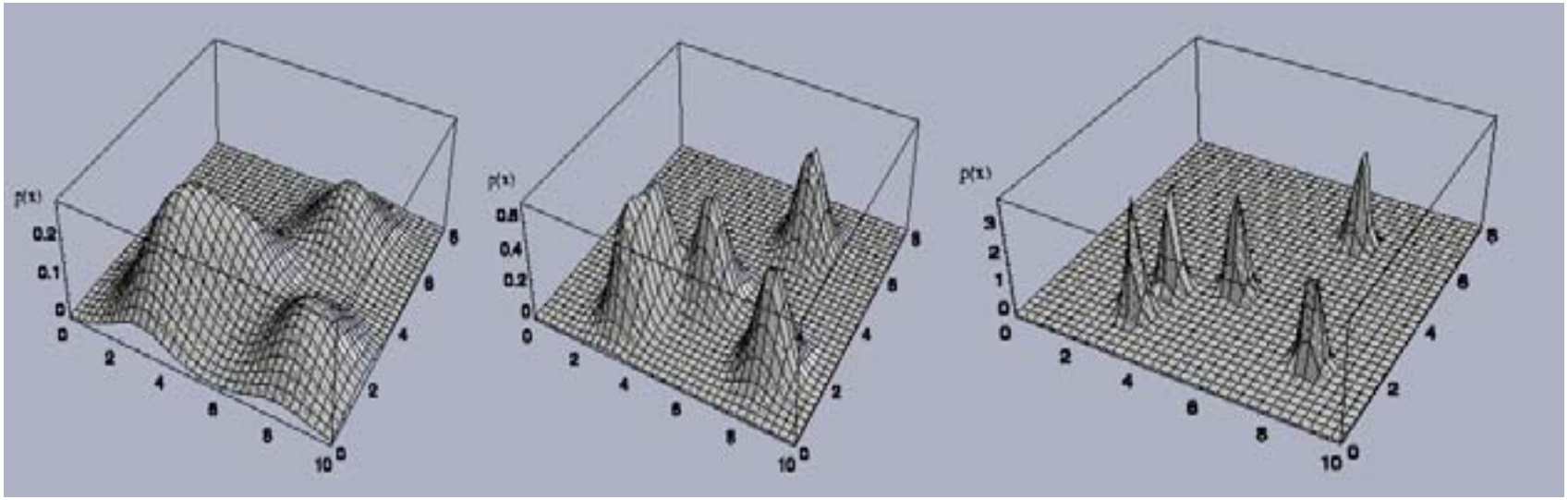


Figure: 3 Parzen-window density estimates based on the same set of 5 samples, using windows from previous figure

- If h is too large the estimate will suffer from too little resolution
- If h is too small the estimate will suffer from too much statistical variability.

Density Estimation

Non-parametric Techniques

Parzen Windows: Bandwidth

- The decision regions of a PW-classifier also depend on bandwidth (and of course kernel)

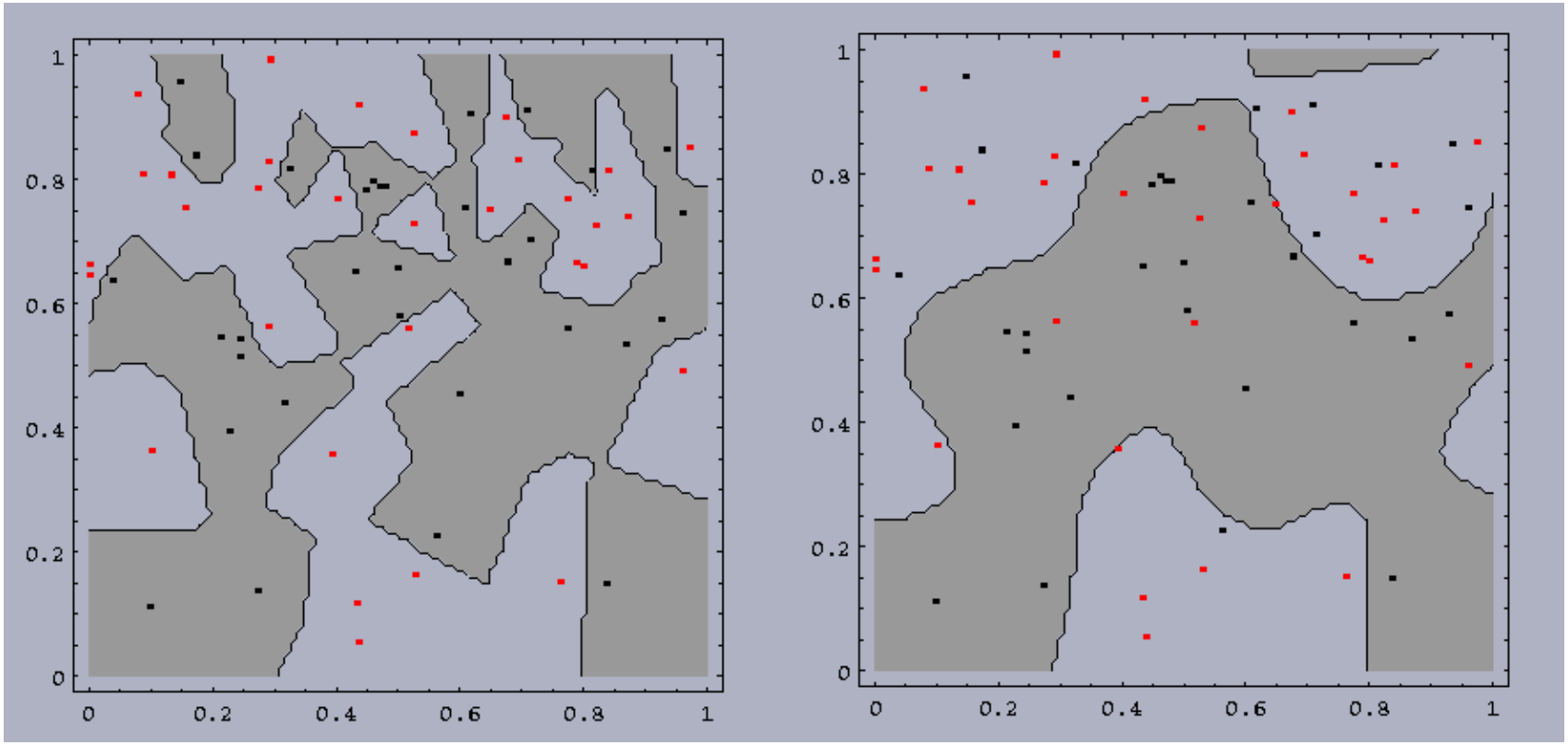


Figure: a) Small h: more complicated boundaries; Large h: less complicated boundaries

k-Nearest-Neighbor Estimation

- Estimate from N training samples by centering a volume V around $\mathbf{x}$ and letting it grow until it captures K samples.
- These samples are the K nearest neighbors of $\mathbf{x}$.
- In regions of high density (around $\mathbf{x}$) the volume will be relatively small.
- K plays a similar role as the bandwidth parameter in PW.

k-NN Decision Rule (Classifier)

- Let N be the total number of samples and V the volume around x which contains K samples $\hat{p}(\mathbf{x}) = \frac{K}{NV(\mathbf{x})}$
- Suppose that in the K samples we find K_m from class $\mathcal{C}_m$ (so that $\sum_{m=1}^{C} K_m = K$)
- Let the total number of samples in class $\mathcal{C}_m$ be N_m (so that $\sum_{m=1}^{C} N_m = N$)

k-NN Decision Rule (Classifier)

- Then we may estimate the class-conditional density $p(\mathbf{x}|\mathcal{C}_m)$ as

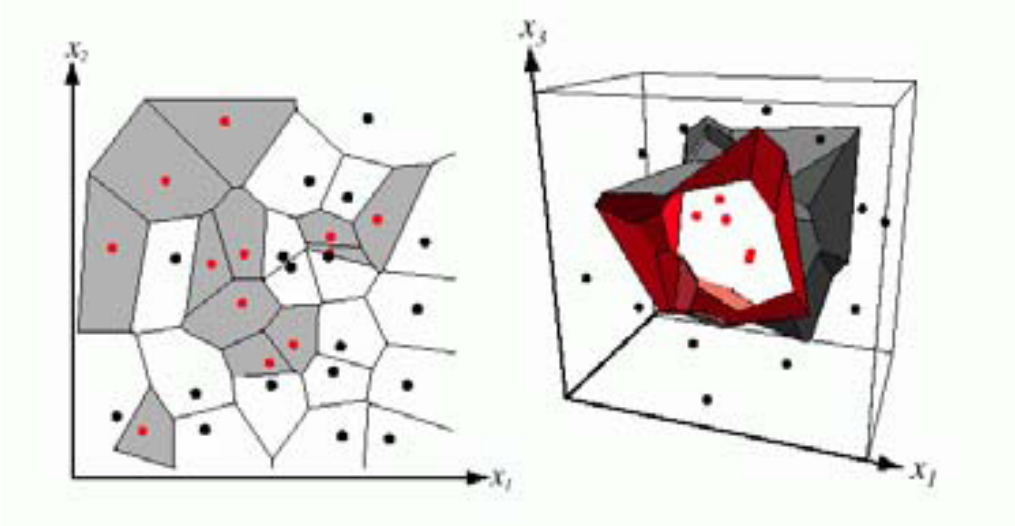
$$\hat{p}(\mathbf{x}|\mathcal{C}_m) = \frac{K_m}{N_m V}$$

  and the prior probability $p(\mathcal{C}_m)$ as $\hat{p}(\mathcal{C}_m) = N_m/N$
- Using these estimates the decision rule:
  assign $\mathbf{x}$ to $\mathcal{C}_m$ if $\forall i : \hat{p}(\mathcal{C}_m|\mathbf{x}) \geq \hat{p}(\mathcal{C}_i|\mathbf{x})$
  translates (Bayes' theorem) to:
  assign $\mathbf{x}$ to $\mathcal{C}_m$ if $\forall i : K_m \geq K_i$

k-NN Decision Rule (Classifier)

- The decision rule is to assign $\mathbf{x}$ to the the class that receives the largest vote amongst the K nearest neighbors

For $K = 1$ this is the nearest neighbor rule producing a Voronoi tesselation of the training space



- This rule is sub-optimal, but when the number of prototypes is large its error is never worse then twice the Bayes rate.

Non-parametric comparison

- Parzen window estimates require storage of all observations and N evaluations of the kernel function for each estimate, which is computationally expensive!
- Nearest neighbor requires the storage of all the observations.
- Histogram estimates do not require storage for all the observations, they require storage for description of the bins. But for simple histogram the number of the bins grows exponentially with dimension of the observation space.

Advantages

- Generality: same procedure for unimodal, normal and bimodal mixture
- No assumption about the distribution required ahead of time
- With enough samples we can converge to an arbitrarily complicated target density

Disadvantages

- Number of required samples may be very large (much larger than would be required if we knew the form of the unknown density)
- Curse of dimensionality
- In case of PW and KNN computationally expensive (storage & processing)
- Sensitivity to choice of bin size, bandwidth, ...

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.

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