Introduction to Signal Detection and Estimation (SDE)

Dirk Dahlhaus

Communications Laboratory [ComLab]

Summer Semester 2019

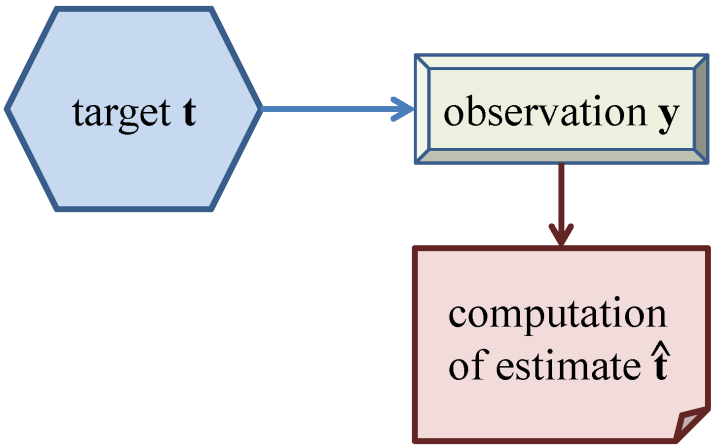
Organisation

- time: summer semester (SS), annually, Friday: 13:00-16:30
- place: HS 0446
- workload: 45 hours course attendance,135 hours self-study
- language: English, oral exam (30 minutes either in English or in German)
- exercises: are integrated in the lecture (on demand)
- regular attendance of the lecture *and* the exercises is mandatory to pass the exam
- the lecture is based on the books (well-known in communications), see the Reference
- upon passing the exam, you obtain **6 credit points for lecture and exercises**

Table of Contents

Table of Contents

What is this Lecture About?



observation $\mathbf{y}$:

- represented by a scalar or a vector
- somehow depending on the target $\mathbf{t}$
- subject to some uncertainty

target $\mathbf{t}$ may be:

- state of a system or environment
- information contained in a signal, picture, etc.
- location of an object
- time of an event
- ...

objective: detection/estimation of $\mathbf{t}$

- on the basis of $\mathbf{y}$
- making use of side information

Hypothesis Testing

In hypothesis testing problems we are confronted with a finite number of "states of nature". Based on an observation, which is modeled as a random vector $\mathbf{Y}$ with a probability distribution that depends on the state (i.e., the hypothesis), our objective is to decide which of the given hypotheses is true. We shall restrict our attention to problems involving two hypotheses H_0 and H_1. Hence, here the target t has the form of a binary variable $t \in \{0, 1\}$.
The formulation of a detection rule may be facilitated by

- the known probability distributions of $\mathbf{Y}$ under H_0 and H_1
- given costs of choosing $\hat{t}$ when hypothesis H_t is true.

Example: smoke detector

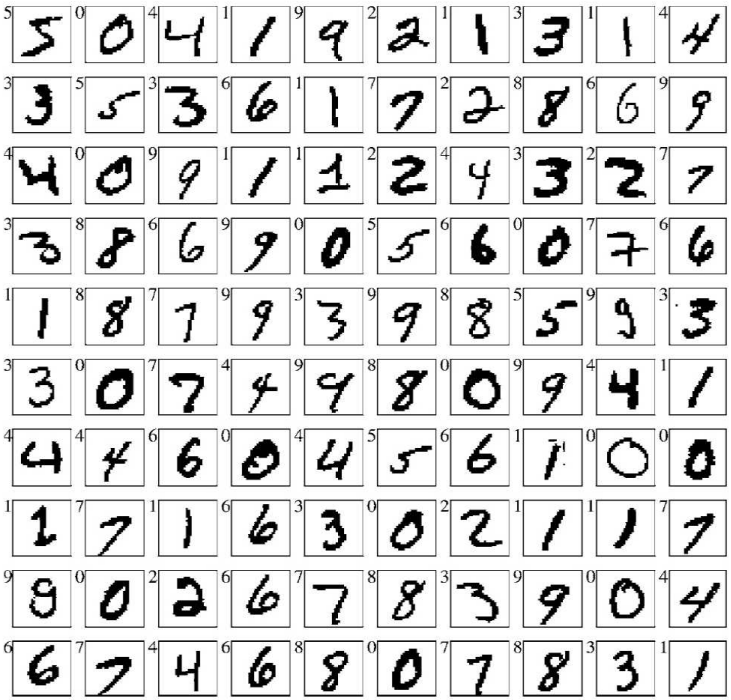
Optical smoke detectors measure the amount of particles by evaluating the scattering of the light from a light source. We may let a real-valued random variable Y describe the amount of particles at a certain time. The probability distributions of Y for both the case of no fire (hypothesis H_0) and the case of a fire (hypothesis H_1) are assumed known. Furthermore, we know the cost of triggering a (false) fire alarm when there is no fire, and the cost of failing to trigger an alarm when there is a fire.

At what amount of particles in the air should the detector trigger an alarm?

Classification

In many detection problems an optimal detection rule cannot be derived because of a difficulty to assign probability distributions to the hypotheses, possibly because $\mathbf{y}$ has a high dimension. More heuristic approaches are required in such situations. In the presence of training data, adaptive so-called *supervised learning* techniques may be employed. A number of different methods are known under this term for *classification* tasks, that is, for the association of observations with a number of given classes. The training sets typically consist of pairs of samples (i.e., observations), and associated class labels (i.e., targets).

Example: MNIST training images with class labels for handwritten digit recognition.



Parameter Estimation

Rather than making decisions between two hypotheses or between a finite number of classes to which a sample belongs, we sometimes need to make a choice among a continuum of possible states behind an observation. The state may be represented by a parameter vector $\boldsymbol{\theta}$. Hence, we wish to estimate $\boldsymbol{\theta}$ on the basis of the observation y. In such a situation we need a function which maps y onto an estimate $\hat{\boldsymbol{\theta}}$ of the parameter vector. The formulation of this function may be facilitated by

- known probability distributions of the random observation given $\boldsymbol{\theta}$
- the prior probability distribution of $\boldsymbol{\theta}$
- a function $C(\mathbf{a}, \boldsymbol{\theta})$ defining the cost of estimating a true state $\boldsymbol{\theta}$ as $\mathbf{a}$.

Example: coherent signal demodulation

In wireless radio systems, to enable a coherent demodulation of quadrature amplitude modulated signals for instance, the receivers need to first estimate the amplitude and phase of a baseband signal. For this purpose, preambles are embedded in the transmitted signal bursts. The observation of the preamble enables the receiver to estimate the signal amplitude and phase for the following demodulation procedure.

Smoothing, Filtering, and Prediction

Sometimes we are given a sequence of observations, in the form of noisy measurements, of a signal or dynamic system over time. We may want to perform

- *smoothing*, that is, to eliminate the noise from the past measurements
- *filtering*, that is, to estimate the current state of the system as accurately as possible in real time
- *prediction*, that is, to forecast the future course of the signal or system.

Depending on the assumed model there are different approaches for the above tasks.

Kalman filter:

The Kalman filter, for example, builds on a linear dynamic system model and offers an efficient recursive estimation of the internal system state from a series of noisy measurements. The Kalman filter has a multitude of applications from object position tracking to time-variant channel parameter estimation.

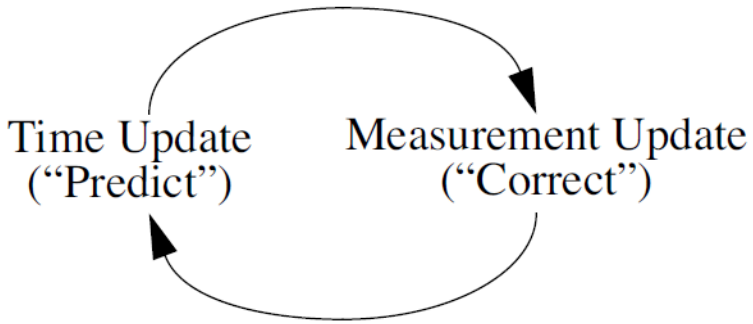


Table of Contents

What is probability theory good for?

The Three Doors Problem

In a popular television show in the 70's, a finalist was given a choice of three doors of which only one contains the big price. Behind the other two doors there was nothing ...

After the finalist chose one of the doors, the quizmaster then revealed one "empty door" among the two doors that were not chosen. After that, the quizmaster asked if the finalist would like to switch to the other door to open.

The question: What should you do? Do you open the door you initially chose or do you switch to the other unopened door before? Do the odds of winning the game increase by switching to the remaining door?

Copyright Archive Photos

Probability Space and Random Variables

A *probability space* $(\Omega, \mathcal{A}, \mathbb{P})$ consists of:

- a sample space Ω, which is a non-empty set
- a set of events $\mathcal{A}$, which is a σ-algebra over Ω and contains all the subsets of Ω
- a probability measure $\mathbb{P} : \mathcal{A} \to [0, 1]$, which is a countably additive function, subject to $\mathbb{P}(\Omega) = 1$, associating every event with a probability

A *random variable* Y represents a measurable mapping from Ω to $\mathbb{R}$. A random variable thus has a real value which is, however, subject to uncertainty.

Given a subset B of $\mathbb{R}$, we may associate the *event* $(Y \in B)$ the *probability*

$$P(Y \in B) = \mathbb{P}\left(Y^{-1}(B)\right).$$

Remark: The measurability of Y ensures that $Y^{-1}(B) \in \mathcal{A}$.

Discrete and Continuous Random Variables

Discrete random variable Y:

- Y is a mapping from Ω to a finite or a countably infinite subset of $\mathbb{R}$
- in case of $Y \in \{y_1, \ldots, y_L\}$,

$$\sum_{i=1}^{L} P(Y = y_i) = 1$$

- we will write $P(Y = y_i)$ simply as $P(y_i)$ and $P(Y \in B)$ simply as $P(B)$ whenever the involvement of Y is clear

Continuous random variable Y:

- Y is a mapping from Ω to an uncountable subset of $\mathbb{R}$
- if the cumulative distribution function $F_Y(y) = P(Y \leq y)$ is differentiable over $\mathbb{R}$, the distribution of Y may be described by a *probability density function* (PDF) $p_Y(y)$, which is subject to

$$P(Y \in B) = \int_B p_Y(y)\mathrm{d}y \quad \text{for any } B \subset \mathbb{R}$$

- we will omit the subscript and simply write $p(Y)$ whenever the involvement of Y is clear

Probability Densities

Joint Probability

Joint probability:

- $P(X = x, Y = y)$ denotes the joint probability that X takes the value x and Y takes the value y
- under certain conditions, the discussion of which lies outside the scope of this course, the distribution of two continuous random variables X and Y can be described by the joint PDF $p_{X,Y}(x, y)$

Random vectors:

- a random vector comprises multiple random variables
- for a K-dimensional random vector $\mathbf{X}$ with joint PDF $p_{\mathbf{X}}(\mathbf{x})$,

$$P(\mathbf{X} \in B) = \int_B p_{\mathbf{X}}(\mathbf{x}) \mathrm{d}\mathbf{x} \quad \text{for any } B \subset \mathbb{R}^K$$

Marginal and Conditional Probability: Discrete Random Variables

Consider two discrete random variables $X \in \{x_1, \ldots, x_L\}$ and $Y \in \{y_1, \ldots, y_M\}$ with the joint probability $P(X = x_i, Y = y_j)$:

- the *marginal* probability $P(X = x_i)$ is given as

$$P(X = x_i) = \sum_{j=1}^{M} P(X = x_i, Y = y_j)$$

(above formula is sometimes called the *sum rule* of probability)

- the *conditional* probability of Y given $(X = x_i)$ is written as

$$P(Y = y_j | X = x_i),$$

which can be calculated using

$$P(X = x_i, Y = y_j) = P(Y = y_j | X = x_i) P(X = x_i)$$

(above formula is sometimes called the *product rule* of probability)

Marginal and Conditional Probability: Continuous Random Variables

Given the joint PDF $p_{X,Y}(x,y)$ of two continuous random variables X and Y:

- the marginal distribution of X is given by the PDF

$$p_X(x) = \int_{-\infty}^{\infty} p_{X,Y}(x,y)\mathrm{d}y$$

- the conditional probability of Y given $(X = x)$ can expressed as the conditional PDF

$$p_{Y|X}(y|X=x) \qquad \text{(or simply } p(y|X=x) \text{ or } p(y|x)\text{)},$$

  which can be calculated using

$$p_{X,Y}(x,y) = p_{X|Y}(y|X=x)p_X(x)$$

Bayes' theorem has a central role in this course:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Expectation and Covariance

Given a random variable X with PDF $p_X(x)$ and a function $f : \mathbb{R} \to \mathbb{R}$:

- the *expectation* (or *mean*) of X and the expectation of $f(X)$ are defined as

$$E(X) = \int_{-\infty}^{\infty} p_X(x) x \mathrm{d}x \quad \text{and} \quad E(f(X)) = \int_{-\infty}^{\infty} p_X(x) f(x) \mathrm{d}x$$

- expectations of discrete random variables are defined in a similar way
- the *variance* of X is defined as $\mathrm{Var}(X) = E\left((X - E(X))^2\right)$
- Jensen's inequality: if $f(\cdot)$ is a convex function then $f(E(X)) \leq E(f(X))$

Given a random vector $\mathbf{X}$ with probability density $p_{\mathbf{X}}(\mathbf{x})$:

- the expectation $E(\mathbf{X})$ of $\mathbf{X}$ is defined as the vector holding the expectations of the random variables contained by $\mathbf{X}$
- the *covariance matrix* of $\mathbf{X}$ is a symmetric, *non-negative definite* matrix defined as

$$\mathbf{\Sigma_X} = E\left((\mathbf{X} - E(\mathbf{X}))(\mathbf{X} - E(\mathbf{X}))^{\mathrm{T}}\right)$$

- if $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$ for constant $\mathbf{A}$ and $\mathbf{b}$, then $E(\mathbf{Y}) = \mathbf{A}E(\mathbf{X}) + \mathbf{b}$ and $\mathbf{\Sigma_Y} = \mathbf{A}\mathbf{\Sigma_X}\mathbf{A}^{\mathrm{T}}$

Conditional Expectation

Given two continuous random variables X and Y and the conditional PDF $p_{Y|X}(y|x)$, the *conditional expectation* of Y given $(X = x)$ is written as $E(Y|X = x)$:

$$E(Y|X = x) = \int_{-\infty}^{\infty} p_{Y|X}(y|X = x) y \mathrm{d}y$$

Remarks:

- conditional expectations are similarly defined for discrete random variables and for random vectors
- $E(Y|X = x)$ is deterministic, whereas $E(Y|X)$ represents a random variable
- $E(E(Y|X)) = E(Y)$

Independence

Independent random variables:

- the random variables $X_1, \ldots, X_M$ are independent if, for any $B_1 \subset \mathbb{R}, \ldots, B_M \subset \mathbb{R}$,

$$P(X_1 \in B_1, \ldots, X_M \in B_M) = \prod_{m=1}^{M} P(X_m \in B_m).$$

- the continuous random variables $X_1, \ldots, X_M$ with the respective PDFs $p_{X_1}(x_1), \ldots, p_{X_M}(x_M)$ are independent if

$$p_{X_1,\ldots,X_M}(x_1, \ldots, x_M) = \prod_{m=1}^{M} p_{X_m}(x_m).$$

Remarks:

- if X and Y are independent, then $E(XY) = E(X)E(Y)$ and $E(X|Y) = E(X)$
- the covariance matrix of a random vector with independent elements is a diagonal matrix

Multivariate Gaussian Distribution

Gaussian random variable X with mean m and variance σ^2:

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-m)^2\right)$$

Definition: An M-dimensional random vector $\mathbf{X}$ has a *multivariate Gaussian distribution* if, for every $\mathbf{a} \in \mathbb{R}^M$, $\mathbf{a}^{\mathrm{T}}\mathbf{X}$ is (univariate) Gaussian distributed.

M-variate Gaussian random vector $\mathbf{X}$ with mean vector $\mathbf{m_X}$ and covariance matrix $\mathbf{\Sigma_X}$:

$$\mathbf{X} \sim \mathcal{N}(\mathbf{m_X}, \mathbf{\Sigma_X}), \qquad p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{M/2}(\det \mathbf{\Sigma_X})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m_X})^{\mathrm{T}}\mathbf{\Sigma_X}^{-1}(\mathbf{x}-\mathbf{m_X})\right)$$

Remarks:

- the multivariate Gaussian distribution is fully determined by the mean vector $\mathbf{m_X}$ and the covariance matrix $\mathbf{\Sigma_X}$
- if $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$ for constant $\mathbf{A}$ and $\mathbf{b}$, then $\mathbf{Y} \sim \mathcal{N}(\mathbf{A}\mathbf{m_X} + \mathbf{b}, \mathbf{A}\mathbf{\Sigma_X}\mathbf{A}^{\mathrm{T}})$
- if $\mathbf{X}$ and $\mathbf{Y}$ are independent M-variate Gaussian random vectors, $\mathbf{X} \sim \mathcal{N}(\mathbf{m_X}, \mathbf{\Sigma_X})$ and $\mathbf{Y} \sim \mathcal{N}(\mathbf{m_Y}, \mathbf{\Sigma_Y})$, then $\mathbf{X} + \mathbf{Y} \sim \mathcal{N}(\mathbf{m_X} + \mathbf{m_Y}, \mathbf{\Sigma_X} + \mathbf{\Sigma_Y})$

Marginal and Conditional Distributions

Let

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_a \\ \mathbf{X}_b \end{bmatrix}, \quad \mathbf{m}_\mathbf{X} = \begin{bmatrix} \mathbf{m}_a \\ \mathbf{m}_b \end{bmatrix}, \quad \mathbf{\Sigma}_\mathbf{X} = \begin{bmatrix} \mathbf{\Sigma}_{aa} & \mathbf{\Sigma}_{ab} \\ \mathbf{\Sigma}_{ab}^{\mathrm{T}} & \mathbf{\Sigma}_{bb} \end{bmatrix}.$$

Given that $\mathbf{X} \sim \mathcal{N}(\mathbf{m}_\mathbf{X}, \mathbf{\Sigma}_\mathbf{X})$:

- $\mathbf{X}_a$ and $\mathbf{X}_b$ are multivariate Gaussian with the marginal distributions

  $$\mathcal{N}(\mathbf{m}_a, \mathbf{\Sigma}_{aa}) \quad \text{and} \quad \mathcal{N}(\mathbf{m}_b, \mathbf{\Sigma}_{bb}),$$

  respectively

- the conditional distribution of $\mathbf{X}_a$ given $(\mathbf{X}_b = \mathbf{x}_b)$ is

  $$\mathcal{N}\left(\mathbf{m}_a + \mathbf{\Sigma}_{ab}\mathbf{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \mathbf{m}_b), \mathbf{M}\right)$$

  with $\mathbf{M} = \mathbf{\Sigma}_{aa} - \mathbf{\Sigma}_{ab}\mathbf{\Sigma}_{bb}^{-1}\mathbf{\Sigma}_{ab}^{\mathrm{T}}$ the *Schur complement* of the matrix $\mathbf{\Sigma}_\mathbf{X}$

- if $\mathbf{\Sigma}_{ab} = \mathbf{0}$ then $\mathbf{X}_a$ and $\mathbf{X}_b$ are independent

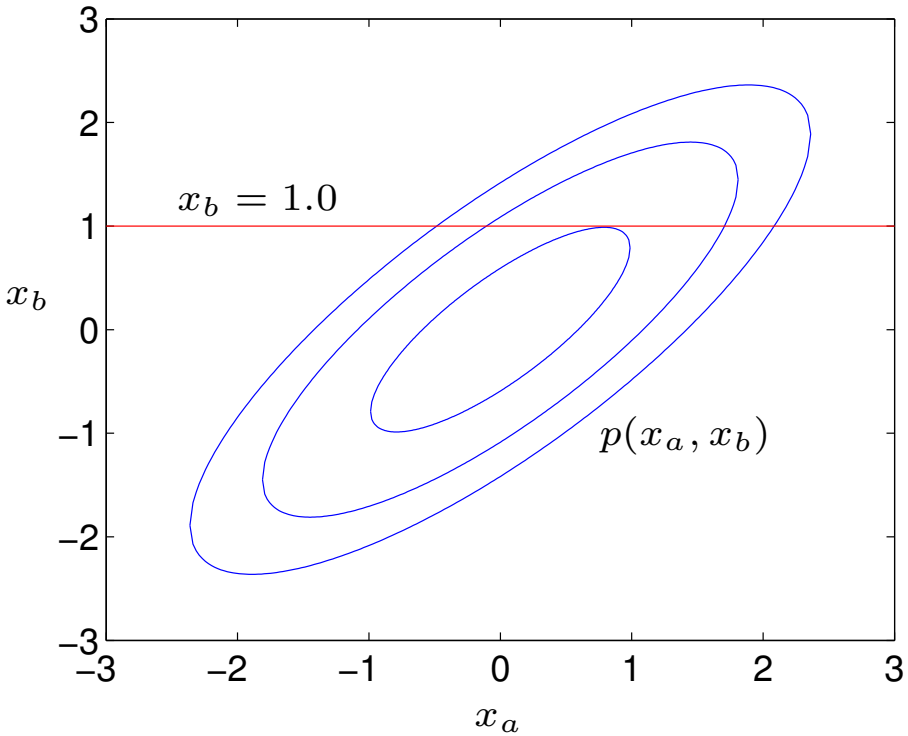
Example

Assume

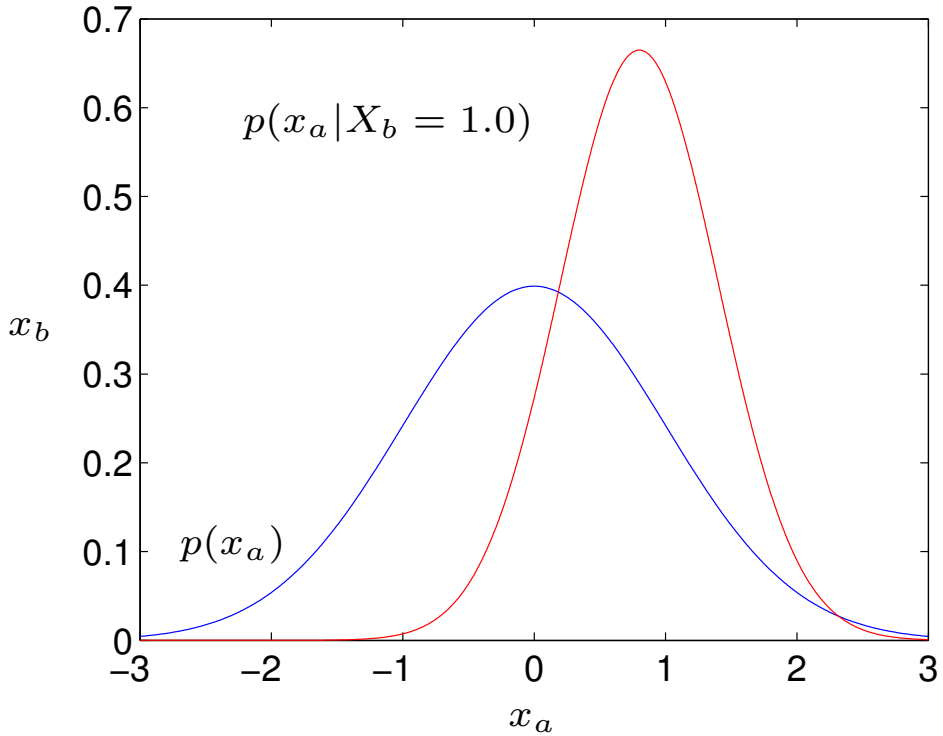
$$\mathbf{m} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} X_a \\ X_b \end{bmatrix} \sim \mathcal{N}(\mathbf{m}, \mathbf{\Sigma}).$$

It follows, for example, that $X_a|(X_b = 1.0) \sim \mathcal{N}(0.8, 0.36)$.

contours of $p(x_a, x_b)$:



marginal and conditional PDFs:



Asymptotic Distribution of Independent Random Vectors

Multivariate central limit theorem:

Let $\mathbf{X}_1, \mathbf{X}_2, \ldots$ be a sequence of independent and identically distributed (i. i. d.) random vectors with mean $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$ and let

$$\bar{\mathbf{X}}_K = \frac{1}{K} \sum_{i=1}^{K} \mathbf{X}_i, \qquad K \geq 1.$$

Then, as $K \to \infty$, the distribution of $\sqrt{K}\left(\bar{\mathbf{X}}_K - \mathbf{m}\right)$ tends to $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$.

Table of Contents

Hypotheses and Observation

Two hypotheses:

- representing two possible states of nature
- denoted as H_0 and H_1

Observation:

- modeled as an M-dimensional random vector $\mathbf{Y}$
- element of an observation set $\Gamma \subset \mathbb{R}^M$, i.e., $\mathbf{Y} \in \Gamma$
- subject to either of two probability distributions $\mathcal{P}_0$ and $\mathcal{P}_1$:

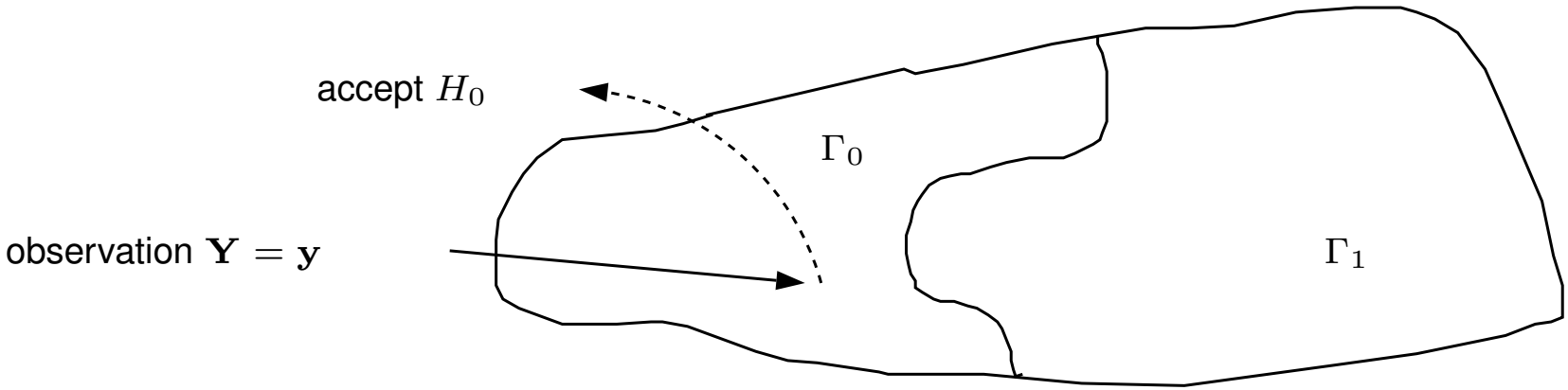
$$\begin{aligned} H_0: &\quad \mathbf{Y} \sim \mathcal{P}_0 \\ \text{versus} & \\ H_1: &\quad \mathbf{Y} \sim \mathcal{P}_1 \end{aligned}$$

Decision rule

A *decision rule* is:

- a hypothesis test for H_0 versus H_1
- a partition of the observation set Γ into sets Γ_0 and Γ_1 such that we choose hypothesis H_j when $\mathbf{y} \in \Gamma_j$ for $j = 0, 1$
- formulated as a function $\delta : \Gamma \to \{0, 1\}$, i.e.,

$$\delta(\mathbf{y}) = \begin{cases} 1 & \text{if } \mathbf{y} \in \Gamma_1 \\ 0 & \text{if } \mathbf{y} \in \Gamma_0 \end{cases}$$



How can we partition Γ in an optimum way?

Costs

- How can we choose Γ_1 in an optimum way?
- Intuition: Consider a military RADAR application, where upon detection of a target, a missile is launched to bring down the target. How do we set up the decision region? Note that it might be costly to detect a target although there is none (false alarm) as well as to miss a target (miss).
- One way of taking into account the different costs is to define an overall risk R which depends on both the statistics of Y for the true hypothesis as well as the decision prescribed by δ.

Classification of errors:

		valid hypothesis	
		H_0	H_1
outcome of test	accept H_0	✓	miss (type II error)
	reject H_0	false alarm (type I error)	✓

Recall the smoke detector example: it may be costly to trigger a fire alarm when there is no fire (i.e., a *false alarm*), and even more costly to miss a fire event (i.e., a *miss*).

Cost assignment:

- we let C_{ij} denote the *cost* incurred by choosing hypothesis H_i when H_j is true
- a commonly used cost assignment is the *uniform cost* assignment

$$C_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases}$$

- the *conditional risk* for hypothesis H_j, denoted as $R_j(\delta)$, represents the expected cost incurred by decision rule δ when that hypothesis is true:

$$R_j(\delta) = C_{0j} P(\Gamma_0 | H_j) + C_{1j} P(\Gamma_1 | H_j), \qquad j = 0, 1$$

How to find a decision rule that is optimal with respect to $R_0(\delta)$ and $R_1(\delta)$?

Bayesian Hypothesis Testing

Bayes Rule

Expected cost of a rule:

- assume we can assign the *prior* (or *a priori*) probabilities

$$\pi_0 = P(H_0) \qquad \text{and} \qquad \pi_1 = P(H_1) = 1 - \pi_0$$

  to the occurrences of hypotheses H_0 and H_1

- the *average risk* or *Bayes risk* is the overall expected cost incurred by decision rule δ and defined by

$$r(\delta) = \pi_0 R_0(\delta) + \pi_1 R_1(\delta)$$

Rule with minimal risk:

- a *Bayes rule* δ_{B} is a decision rule for H_0 versus H_1 that minimizes, over all possible decision rules, the Bayes risk

- a Bayes rule is defined by appropriate decision regions $\widetilde{\Gamma}_0$ or $\widetilde{\Gamma}_1$

Constructing a Bayes Rule

Formulating Bayes rule through decision region $\widetilde{\Gamma}_1$:

- express $R_j(\delta)$ depending on Γ_1:

$$\begin{aligned} R_j(\delta) &= C_{0j}\,P(\Gamma_0|H_j) + C_{1j}\,P(\Gamma_1|H_j) \\ &= C_{0j}(1 - P(\Gamma_1|H_j)) + C_{1j}\,P(\Gamma_1|H_j) \\ &= C_{0j} + (C_{1j} - C_{0j})P(\Gamma_1|H_j), \qquad\qquad j = 0, 1 \end{aligned}$$

- express $r(\delta)$ depending on Γ_1:

$$\begin{aligned} r(\delta) &= \pi_0 R_0(\delta) + \pi_1 R_1(\delta) \\ &= \sum_{j=0}^{1} \pi_j \left(C_{0j} + (C_{1j} - C_{0j})P(\Gamma_1|H_j)\right) \\ &= \sum_{j=0}^{1} \pi_j C_{0j} + \sum_{j=0}^{1} \pi_j (C_{1j} - C_{0j})P(\Gamma_1|H_j) \end{aligned}$$

Constructing a Bayes Rule (cont.)

- if $\Gamma = \{\mathbf{y}_1, \mathbf{y}_2, \dots\}$,

$$r(\delta) = \sum_{j=0}^{1} \pi_j C_{0j} + \sum_{\{i \in \mathbb{N} \,:\, \mathbf{y}_i \in \Gamma_1\}} \left(\sum_{j=0}^{1} \pi_j (C_{1j} - C_{0j}) P(\mathbf{y}_i | H_j) \right),$$

whereas if we are given the conditional densities $p(\mathbf{y}|H_0)$ and $p(\mathbf{y}|H_1)$,

$$r(\delta) = \sum_{j=0}^{1} \pi_j C_{0j} + \int_{\Gamma_1} \left(\sum_{j=0}^{1} \pi_j (C_{1j} - C_{0j}) p(\mathbf{y}|H_j) \right) \mathrm{d}\mathbf{y}$$

- a Bayes rule is thus defined by

$$\begin{aligned} \widetilde{\Gamma}_1 &= \left\{ \mathbf{y} \in \Gamma \;:\; \sum_{j=0}^{1} \pi_j (C_{1j} - C_{0j}) P(\mathbf{y}|H_j) \leq 0 \right\} \\ &= \{ \mathbf{y} \in \Gamma \;:\; \pi_1 (C_{11} - C_{01}) P(\mathbf{y}|H_1) \leq \pi_0 (C_{00} - C_{10}) P(\mathbf{y}|H_0) \} \end{aligned}$$

(or accordingly using conditional densities $p(\mathbf{y}|H_0)$ and $p(\mathbf{y}|H_1)$)

Likelihood Ratio Test

- assuming $C_{11} < C_{01}$ (i.e., the cost of correctly choosing H_1 is less than the cost of incorrectly rejecting H_1),

$$\widetilde{\Gamma}_1 = \{\mathbf{y} \in \Gamma \ : \ P(\mathbf{y}|H_1) \geq \tau P(\mathbf{y}|H_0)\}$$

  with the decision threshold

$$\tau = \frac{\pi_0(C_{10} - C_{00})}{\pi_1(C_{01} - C_{11})}$$

- the *likelihood ratio* is given by

$$L(\mathbf{y}) = \frac{P(\mathbf{y}|H_1)}{P(\mathbf{y}|H_0)},$$

  with the value $+\infty$ if $P(\mathbf{y}|H_0) = 0$

- a *likelihood ratio test* is a decision rule defined in the form

$$\widetilde{\Gamma}_1 = \{\mathbf{y} \in \Gamma \ : \ L(\mathbf{y}) \geq \tau\}$$

- and the corresponding Bayes rule reads

$$\delta_{\mathrm{B}}(\mathbf{y}) = \begin{cases} 1 & \text{if } L(\mathbf{y}) \geq \tau \\ 0 & \text{if } L(\mathbf{y}) < \tau \end{cases}$$

Posterior Probabilities

Formulating Bayes rule through posterior probabilities:

- using Bayes formula,

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

  where $P(\mathbf{y}) = \pi_0 P(\mathbf{y}|H_0) + \pi_1 P(\mathbf{y}|H_1)$ is the overall probability of $\mathbf{y}$

- the probabilities $P(H_0|\mathbf{y})$ and $P(H_1|\mathbf{y})$ are called the *posterior* or *a posteriori* probabilities of the two hypotheses

- Bayes rule:

$$\widetilde{\Gamma}_1 = \{\mathbf{y} \in \Gamma \ : \ C_{10}P(H_0|\mathbf{y}) + C_{11}P(H_1|\mathbf{y}) \leq C_{00}P(H_0|\mathbf{y}) + C_{01}P(H_1|\mathbf{y})\}$$

- the quantity

$$C_{j0}P(H_0|\mathbf{y}) + C_{j1}P(H_1|\mathbf{y})$$

  represents the *posterior cost* of choosing H_j for observation $\mathbf{y}$

- the Bayes rule chooses the hypothesis with minimum posterior cost

Minimum-Probability-of-Error Decision

Choosing uniform cost assignment:

- that is

$$C_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases}$$

- Bayes risk

$$r(\delta) = \pi_0 P(\Gamma_1|H_0) + \pi_1 P(\Gamma_0|H_1)$$

represents the *average probability of error* incurred by decision rule δ

- Bayes rule becomes

$$\delta_{\mathrm{B}}(\mathbf{y}) = \begin{cases} 1 & \text{if } L(\mathbf{y}) \geq \tau \\ 0 & \text{if } L(\mathbf{y}) < \tau \end{cases}$$

with $\tau = \pi_0/\pi_1$, which is called *minimum-probability-of-error decision* scheme

- formulated using posterior probabilities, Bayes rule

$$\delta_{\mathrm{B}}(\mathbf{y}) = \begin{cases} 1 & \text{if } P(H_1|\mathbf{y}) \geq P(H_0|\mathbf{y}) \\ 0 & \text{if } P(H_1|\mathbf{y}) < P(H_0|\mathbf{y}) \end{cases}$$

is also referred to as *maximum a posteriori probability* (MAP) decision rule

Example 1: The Binary Channel

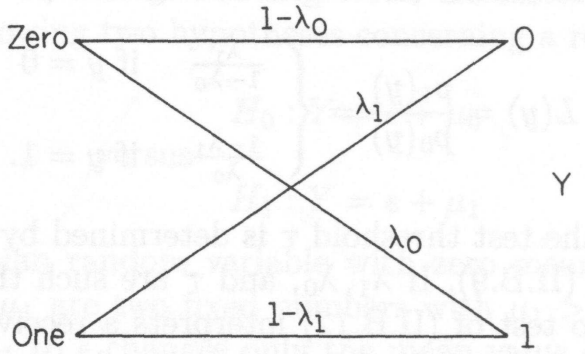


FIGURE II.B.1. The binary channel.

- Suppose a binary digit (*zero* or *one*) is to be transmitted over a communication channel.
- The observation Y is the channel output, which can be either 0 or 1 (e.g. hard decision in a receiver).
- Due to errors in the receiver (synchronization, channel estimation, noise), a transmitted *zero* is received as a 1 with probability λ_0 and as a 0 with probability $(1 - \lambda_0)$, where $0 < \lambda_0 < 1$.

Example 1: The Binary Channel (cont.)

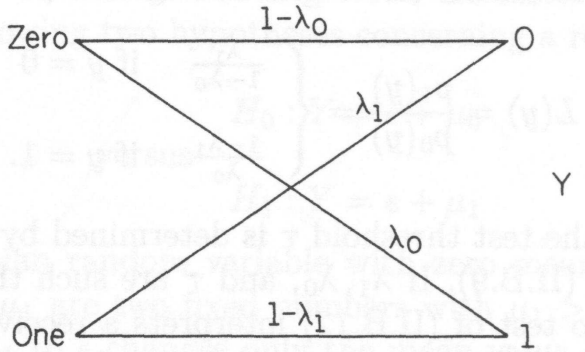


FIGURE II.B.1. The binary channel.

- Correspondingly, a transmitted *one* is received as a 0 with probability λ_1 and as a 0 with probability $(1-\lambda_1)$, where $0 < \lambda_1 < 1$.
- Intuition: The decision whether or not we trust in the received signal depends obviously on λ_0 and λ_1. For example, if $\lambda_0 = \lambda_1 = 0$, the observation y would give us a perfect decision. On the other hand, for $\lambda_0 = \lambda_1 = 1$, we should choose $1 - y$ as our estimate. What is the optimum choice for the general case $0 < \lambda_0 < 1$ and $0 < \lambda_1 < 1$?

Example 1: The Binary Channel (cont.)

Formulation as a binary hypothesis testing problem:

- The hypothesis H_j, $j = 0, 1$, means that a j has been transmitted, the observation set Γ is $\{0, 1\}$, and the observation Y has densities (i.e., probability mass functions)

$$p_j(y) = \begin{cases} \lambda_j & \text{if } y \neq j \\ (1 - \lambda_j) & \text{if } y = j \end{cases}$$

  for $j = 0, 1$.

- The likelihood ratio is given by

$$L(y) = \frac{p_1(y)}{p_0(y)} = \begin{cases} \frac{\lambda_1}{1-\lambda_0} & \text{if } y = 0 \\ \frac{1-\lambda_1}{\lambda_0} & \text{if } y = 1. \end{cases}$$

Example 1: The Binary Channel (cont.)

For a Bayes test, the threshold τ is determined by the costs and prior probabilities. Suppose we have calculated τ.

- Interpretation of the likelihood ratio test $\delta_{\mathrm{B}}(y) = \left\{ \begin{array}{ll} 1 & \text{if } L(y) \geq \tau \\ 0 & \text{if } L(y) < \tau \end{array} \right\}$ with

$$L(y) = \frac{p_1(y)}{p_0(y)} = \left\{ \begin{array}{ll} \frac{\lambda_1}{1-\lambda_0} & \text{if } y = 0 \\ \\ \frac{1-\lambda_1}{\lambda_0} & \text{if } y = 1. \end{array} \right\}:$$

  - Received $y = 0$: if $\lambda_1 \geq \tau(1 - \lambda_0)$, decide a *one* was transmitted, otherwise decide *zero* was transmitted.
  - Received $y = 1$: if $(1 - \lambda_1) \geq \tau\lambda_0$, decide a *one* was transmitted, otherwise decide *zero* was transmitted.

Example 1: The Binary Channel (cont.)

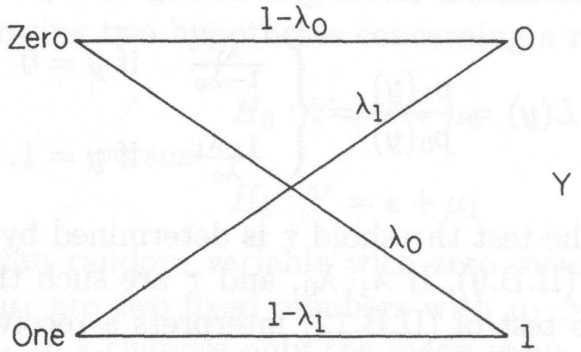


FIGURE II.B.1. The binary channel.

- Assuming uniform costs and equal priors $\pi_0 = \pi_1 = 1/2$, the Bayes rule reads

$$\delta_{\mathrm{B}}(0) = \begin{cases} 1 & \text{if } \lambda_1 \geq 1 - \lambda_0 \\ 0 & \text{if } \lambda_1 < 1 - \lambda_0 \end{cases}$$

$$\delta_{\mathrm{B}}(1) = \begin{cases} 1 & \text{if } 1 - \lambda_1 \geq \lambda_0 \\ 0 & \text{if } 1 - \lambda_1 < \lambda_0 \end{cases}$$

Example 1: The Binary Channel (cont.)

- Reformulation of Bayes' rule:

$$\delta_{\mathrm{B}}(0) = \begin{cases} 0 & \text{if } 1-\lambda_1 > \lambda_0 \\ 1 & \text{if } 1-\lambda_1 \leq \lambda_0 \end{cases}$$

$$\delta_{\mathrm{B}}(1) = \begin{cases} 1 & \text{if } 1-\lambda_1 \geq \lambda_0 \\ 0 & \text{if } 1-\lambda_1 < \lambda_0 \end{cases}$$

- Since boundary points $L(y) = \tau$ can be assigned to either Γ_0 or Γ_1 without changing the risk, we have the equivalent rule

$$\delta_{\mathrm{B}}(y) = \begin{cases} y & \text{if } 1-\lambda_1 \geq \lambda_0 \\ 1-y & \text{if } 1-\lambda_1 < \lambda_0 \end{cases}$$

- Assuming a symmetric channel $\lambda_0 = \lambda_1 = \lambda$, we have

$$\delta_{\mathrm{B}}(y) = \begin{cases} y & \text{if } \lambda \leq 1/2 \quad \Rightarrow \quad \text{keep the bit, since } \lambda \leq 1/2 \\ 1-y & \text{if } \lambda > 1/2 \quad \Rightarrow \quad \text{invert the bit, since } \lambda > 1/2. \end{cases}$$

Example 1: The Binary Channel (cont.)

- What is the risk for the symmetric channel? From

$$\delta_{\mathrm{B}}(y) = \begin{cases} y & \text{if } \lambda \leq 1/2 \quad \Rightarrow \quad \text{keep the bit, since } \lambda \leq 1/2 \\ 1-y & \text{if } \lambda > 1/2 \quad \Rightarrow \quad \text{invert the bit, since } \lambda > 1/2. \end{cases}$$

we first observe, that

$$\Gamma_1 = \begin{cases} \{y=1\} & \text{if } \lambda \leq 1/2 \quad \Rightarrow \quad \text{keep the bit, since } \lambda \leq 1/2 \\ \{y=0\} & \text{if } \lambda > 1/2 \quad \Rightarrow \quad \text{invert the bit, since } \lambda > 1/2. \end{cases}$$

- Thus, we obtain the risk

$$\begin{aligned} r(\delta) &= \pi_0 P_0(\Gamma_1) + \pi_1 P_1(\Gamma_0) \\ &= \begin{cases} (P_0(y=1) + P_1(y=0))/2 & \text{if } \lambda \leq 1/2 \\ (P_0(y=0) + P_1(y=1))/2 & \text{if } \lambda > 1/2 \end{cases} \\ &= \left\{ \begin{array}{ll} \lambda & \text{if } \lambda \leq 1/2 \\ 1-\lambda & \text{if } \lambda > 1/2 \end{array} \right\} = \min\{\lambda, 1-\lambda\}. \end{aligned}$$

Example 1: The Binary Channel (cont.)

To consider some more cases, we reformulate the likelihood ratio

$$L(y) = \frac{p_1(y)}{p_0(y)} = \begin{cases} \frac{\lambda_1}{1-\lambda_0} & \text{if } y = 0 \\ \frac{1-\lambda_1}{\lambda_0} & \text{if } y = 1. \end{cases}$$

for the different observations y as

$$L(y) = \left(\frac{\lambda_{1-y}}{1-\lambda_y} \right)^{1-2y}.$$

and consider the set $\Gamma_1 = \{y \in \{0,1\} \,|L(y) \geq \tau\}$ for $C_{00} = C_{11} = 0$, i.e. $\tau = \frac{\pi_0 C_{10}}{\pi_1 C_{01}}$.

Example 1: The Binary Channel (cont.)

case 1: $\pi_0 = \pi_1 = 1/2, C_{01} = C_{10} = 1, C_{00} = C_{11} = 0, \lambda_0 = \lambda_1 = 1/2, \tau = 1$

$$
\begin{aligned}
\Gamma_1 &= \left\{ y \in \{0,1\} \,\middle|\, \lambda_{1-y}^{1-2y} \geq (1-\lambda_y)^{1-2y} \right\} \\
&= \{y = 0 \wedge \lambda_1 \geq 1-\lambda_0\} \cup \left\{ y = 1 \wedge \frac{1}{\lambda_0} \geq \frac{1}{1-\lambda_1} \right\} \\
&= \left\{ y = 0 \wedge \frac{1}{2} \geq \frac{1}{2} \right\} \cup \left\{ y = 1 \wedge \frac{1}{2} \geq \frac{1}{2} \right\} \\
&= \{y = 0\} \cup \{y = 1\} \\
&= \{0,1\} = \Gamma \\
\Rightarrow \Gamma_0 &= \emptyset.
\end{aligned}
$$

The risk is

$$r(\delta) = \pi_0 C_{10} P_0(\Gamma_1) + \pi_1 C_{01} P_1(\Gamma_0) = \frac{1}{2} P_0(\Gamma_1) = \frac{1}{2} P_0(\Gamma) = \frac{1}{2}.$$

Example 1: The Binary Channel (cont.)

case 2: $\pi_0 = \pi_1 = 1/2, C_{01} = 1, C_{10} = 2, C_{00} = C_{11} = 0, \lambda_0 = \lambda_1 = 1/2, \tau = 2$

a) We first assume the test from case 1 (which is not a Bayes test for the above parameters) with $\Gamma_1 = \Gamma$ and calculate the risk

$$r(\delta) = \pi_0 C_{10} P_0(\Gamma_1) = \frac{1}{2} \times 2 \times 1 = 1.$$

b) Bayes test:

$$\begin{aligned} \Gamma_1 &= \left\{ y \in \{0,1\} \,\middle|\, \lambda_{1-y}^{1-2y} \geq \tau(1-\lambda_y)^{1-2y} \right\} \\ &= \left\{ y = 0 \wedge \frac{1}{2} \geq 1 \right\} \cup \left\{ y = 1 \wedge \frac{1}{2} \geq 1 \right\} = \emptyset. \end{aligned}$$

The risk is

$$r(\delta) = \pi_0 C_{10} P_0(\Gamma_1) + \pi_1 C_{01} P_1(\Gamma_0) = \pi_1 C_{01} P_1(\Gamma_0) = \frac{1}{2} \times 1 \times 1 = \frac{1}{2}.$$

Example 1: The Binary Channel (cont.)

case 2: $\pi_0 = \pi_1 = 1/2,\ C_{01} = 1,\ C_{10} = 2,\ C_{00} = C_{11} = 0,\ \lambda_0 = \lambda_1 = 1/2,\ \tau = 2$

b) Obviously, the optimum test provides only half the risk of the non-optimum test.

Intuition: Increasing C_{10} from $C_{10} = 1$ to $C_{10} = 2$ leads to a reduced set Γ_1 (in the optimum case $\Gamma_1 = \emptyset$ for $\lambda_0 = \lambda_1 = 1/2$). In the general case of arbitrary λ_0 and λ_1, we expect that we have to be *pretty sure* about deciding $\delta = 1$ (i.e. $y \in \Gamma_1$) for high costs C_{1j}.

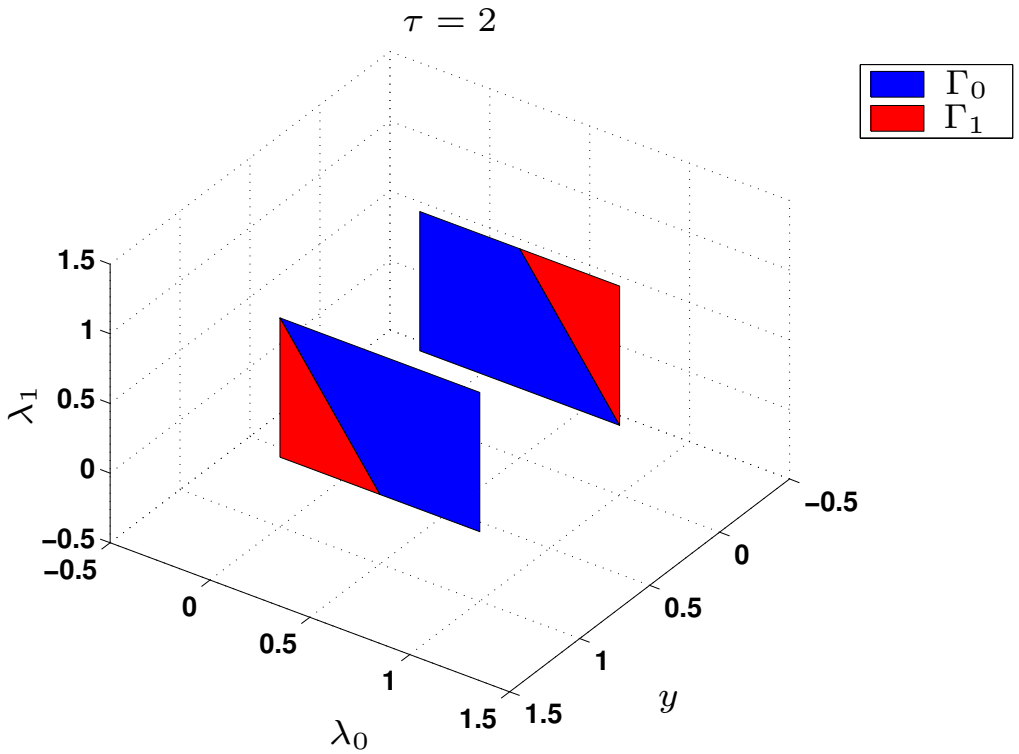
case 3: $\pi_0 = \pi_1 = 1/2,\ C_{01} = 1,\ C_{10} = 2,\ C_{00} = C_{11} = 0,$ arbitrary $\lambda_0, \lambda_1,\ \tau = 2$

The Bayes test provides the decision regions

$$\begin{aligned}
\Gamma_1 &= \left\{ y \in \{0,1\} \,\middle|\, \lambda_{1-y}^{1-2y} \geq \tau(1-\lambda_y)^{1-2y} \right\} \\
&= \{y = 0 \wedge \lambda_1 \geq 2(1-\lambda_0)\} \cup \{y = 1 \wedge \lambda_1 \leq 1 - 2\lambda_0\} \\
\Gamma_0 &= \{y = 0 \wedge \lambda_1 < 2(1-\lambda_0)\} \cup \{y = 1 \wedge \lambda_1 > 1 - 2\lambda_0\}
\end{aligned}$$

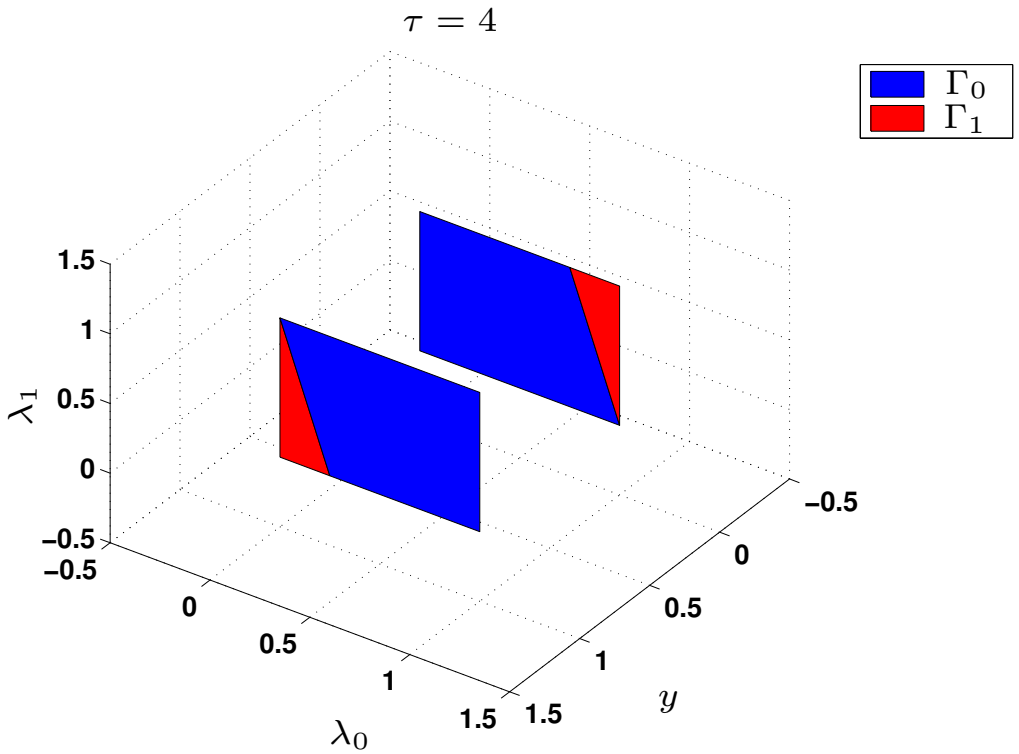
Example 1: The Binary Channel (cont.)

case 3: $\pi_0 = \pi_1 = 1/2$, $C_{01} = 1$, $C_{10} = 2$, $C_{00} = C_{11} = 0$, arbitrary λ_0,λ_1, $\tau = 2$



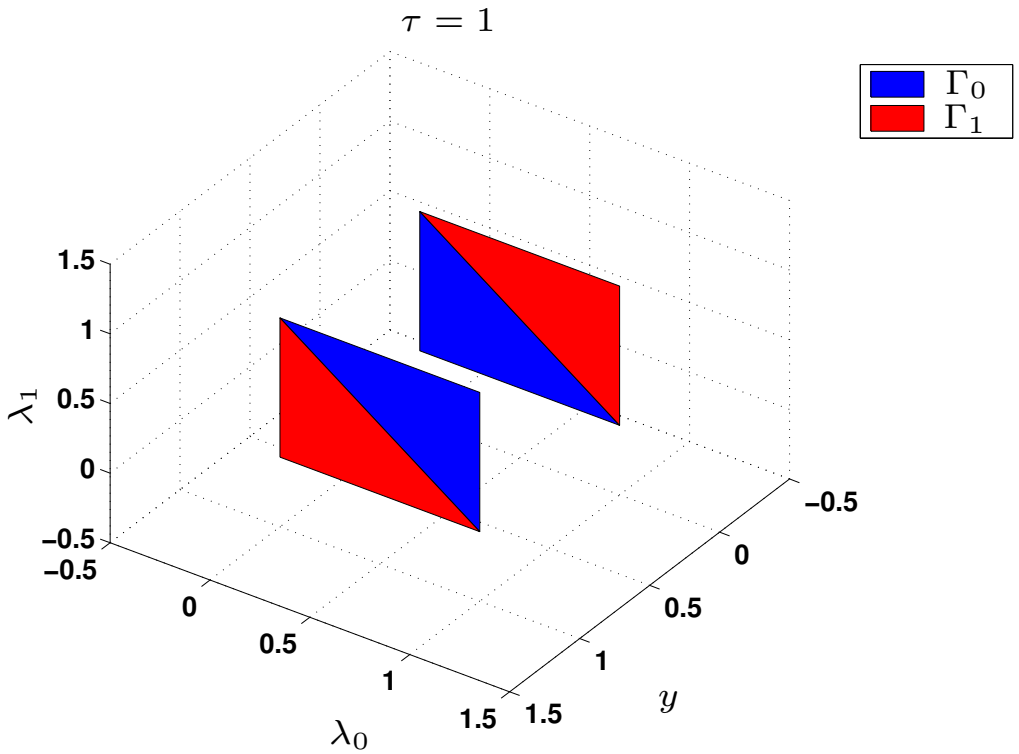
Example 1: The Binary Channel (cont.)

case 3: $\pi_0 = \pi_1 = 1/2$, $C_{01} = 1$, $C_{10} = 4$, $C_{00} = C_{11} = 0$, arbitrary λ_0,λ_1, $\tau = 4$



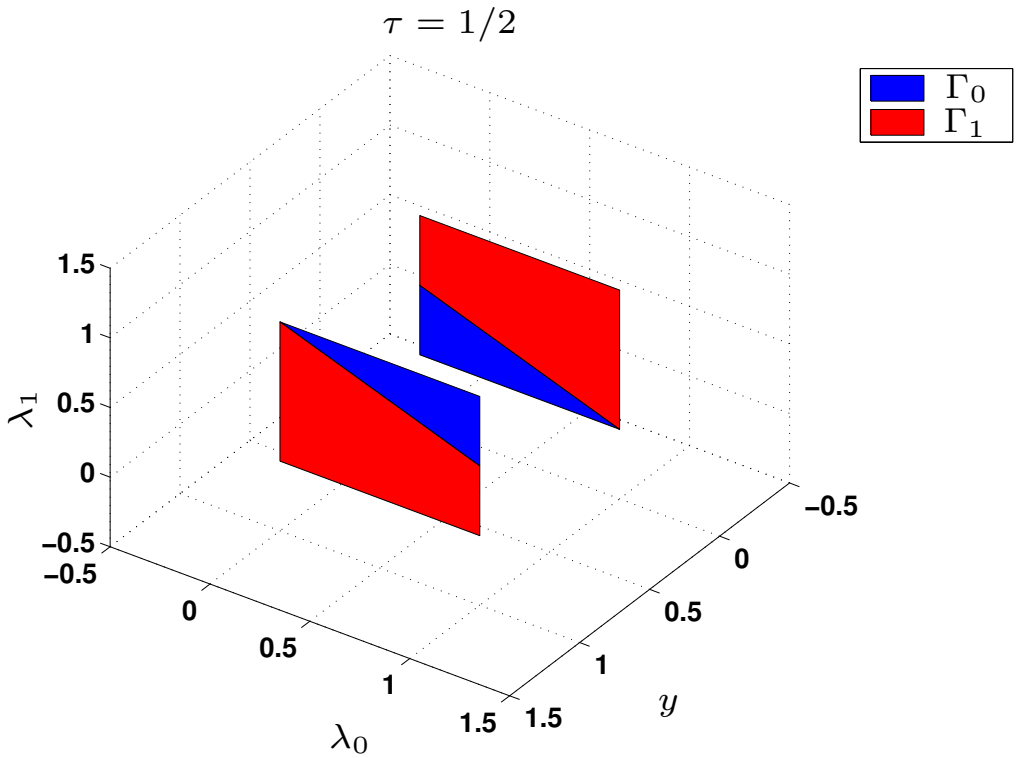
Example 1: The Binary Channel (cont.)

case 3: $\pi_0 = \pi_1 = 1/2$, $C_{01} = 1$, $C_{10} = 1$, $C_{00} = C_{11} = 0$, arbitrary λ_0,λ_1, $\tau = 1$



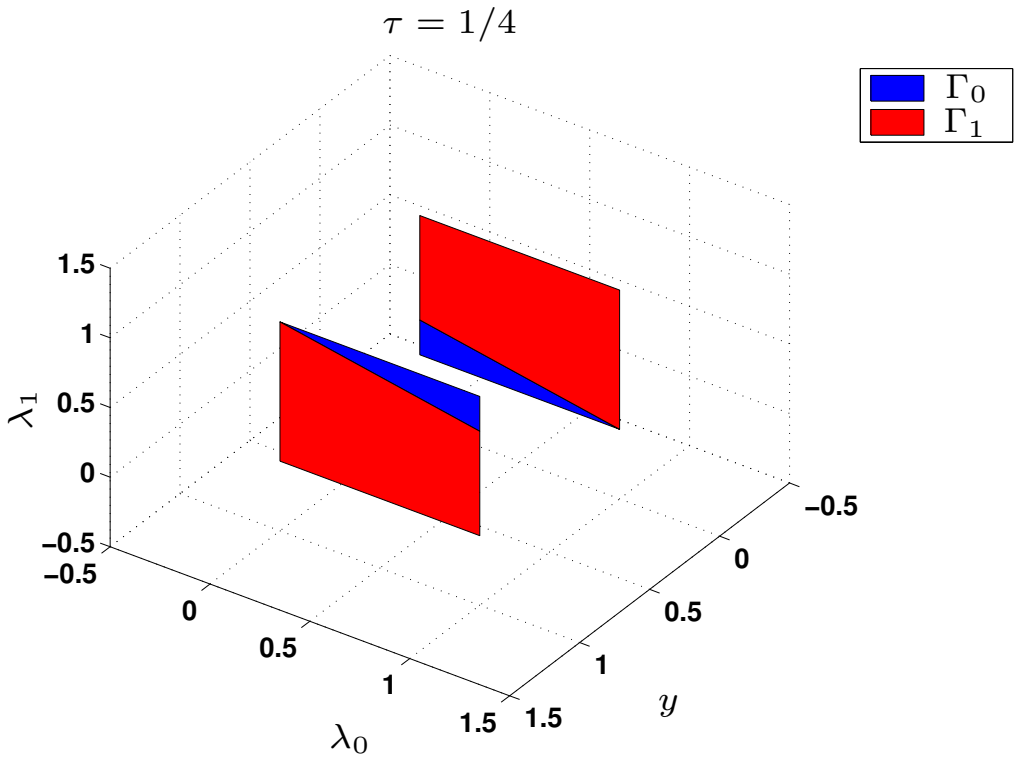
Example 1: The Binary Channel (cont.)

case 3: $\pi_0 = \pi_1 = 1/2$, $C_{01} = 2$, $C_{10} = 1$, $C_{00} = C_{11} = 0$, arbitrary λ_0,λ_1, $\tau = 1/2$



Example 1: The Binary Channel (cont.)

case 3: $\pi_0 = \pi_1 = 1/2$, $C_{01} = 4$, $C_{10} = 1$, $C_{00} = C_{11} = 0$, arbitrary λ_0,λ_1, $\tau = 1/4$



Example 2: Location Testing with Gaussian Error

- Consider the following two hypotheses concerning a real-valued observation Y with real numbers $\mu_1 > \mu_0$:

$$\begin{aligned} H_0: &\quad Y = \epsilon + \mu_0 \\ \text{versus} & \\ H_1: &\quad Y = \epsilon + \mu_1 \end{aligned}$$

  where ϵ is a Gaussian random variable with zero mean and variance σ^2.

- The probability density function (PDF) of $\epsilon = X$ is

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad x \in \mathbb{R}.$$

- The likelihood ratio is given by

$$L(y) = \frac{p_1(y)}{p_0(y)} = \frac{\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu_1)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu_0)^2}{2\sigma^2}\right)} = \exp\left(\frac{\mu_1 - \mu_0}{\sigma^2}\left(y - \frac{\mu_1 + \mu_0}{2}\right)\right).$$

Example 2: Location Testing with Gaussian Error (cont.)

- A Bayes test is defined by

$$\delta_{\mathrm{B}}(y) = \begin{cases} 1 & \text{if } \exp\left(\frac{\mu_1-\mu_0}{\sigma^2}\left(y - \frac{\mu_1+\mu_0}{2}\right)\right) \geq \tau \\ 0 & \text{otherwise,} \end{cases}$$

  where τ is an appropriate threshold.

- Since $\mu_1 > \mu_0$, the likelihood ratio is a strictly increasing function of the observation y (i.e. $\mathrm{d}L(y)/\mathrm{d}y = (\mu_1 - \mu_0)L(y)/\sigma^2 > 0$). Thus, an equivalent test is given by

$$\delta_{\mathrm{B}}(y) = \begin{cases} 1 & \text{if } y \geq \tau' \\ 0 & \text{if } y < \tau', \end{cases}$$

  with $\tau' = \dfrac{\sigma^2}{\mu_1 - \mu_0} \ln(\tau) + \dfrac{\mu_1 + \mu_0}{2}$.

Example 2: Location Testing with Gaussian Error (cont.)

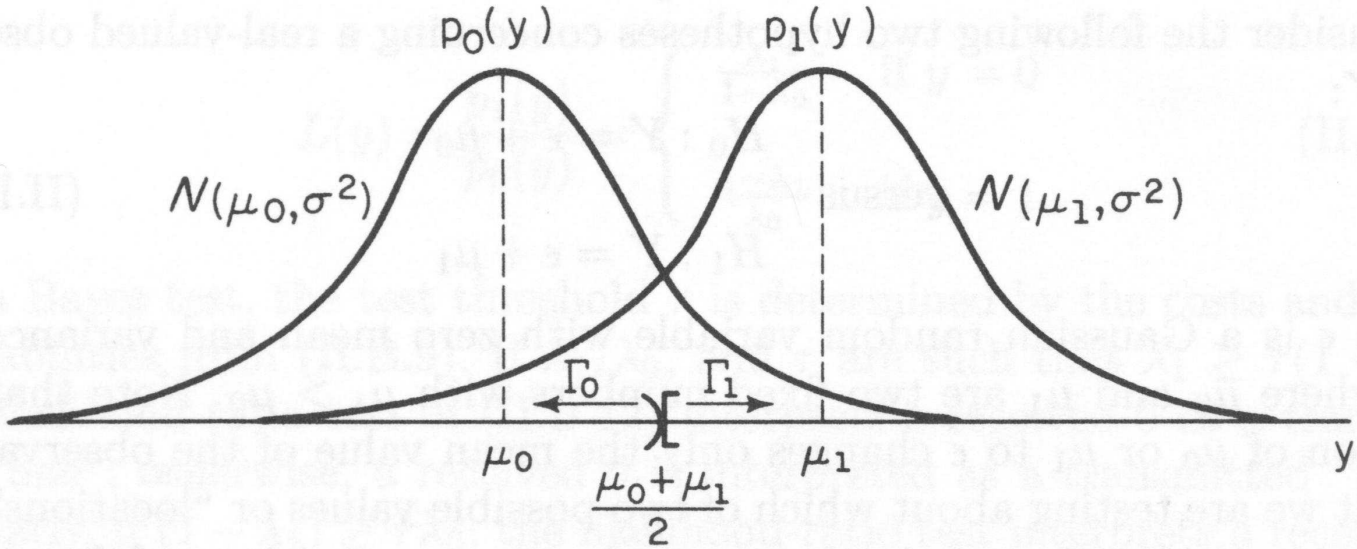


FIGURE II.B.2. Illustration of location testing with Gaussian errors, uniform costs, and equal priors.

$$\tau' = \frac{\sigma^2}{\mu_1 - \mu_0} \ln(1) + \frac{\mu_1 + \mu_0}{2} = \frac{\mu_1 + \mu_0}{2}$$

Example 2: Location Testing with Gaussian Error (cont.)

- The minimum Bayes risk $r(\delta_{\mathrm{B}})$ requires the expressions for $P_j\left(\tilde{\Gamma}_1\right) = P\left(\tilde{\Gamma}_1 \middle| H_j\right)$ for $j = 0, 1$.
- Since $\tilde{\Gamma}_1 = \{y \in \mathbb{R} \mid y \geq \tau'\}$, we have

$$\begin{aligned} P_j\left(\tilde{\Gamma}_1\right) &= \int_{\tau'}^{\infty} p_j(y)\mathrm{d}y = 1 - \Phi\left(\frac{\tau' - \mu_j}{\sigma}\right) \\ &= \begin{cases} 1 - \Phi\left(\frac{\ln(\tau)}{d} + \frac{d}{2}\right), & j = 0 \\ 1 - \Phi\left(\frac{\ln(\tau)}{d} - \frac{d}{2}\right), & j = 1, \end{cases} \end{aligned}$$

where $\Phi(\cdot)$ denotes the cumulative probability distribution function (CDF) of a $\mathcal{N}(0,1)$ random variable (RV) and $d = (\mu_1 - \mu_0)/\sigma$ defined by

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} \exp\left(-\frac{y^2}{2}\right)\mathrm{d}y.$$

Example 2: Location Testing with Gaussian Error (cont.)

- The minimum Bayes risk $r(\delta_{\mathrm{B}})$ requires the expressions for $P_j\left(\tilde{\Gamma}_1\right) = P\left(\tilde{\Gamma}_1 \middle| H_j\right)$ for $j = 0, 1$.
- Since $\tilde{\Gamma}_1 = \{y \in \mathbb{R} \mid y \geq \tau'\}$, we have

$$\begin{aligned} P_j\left(\tilde{\Gamma}_1\right) &= \int_{\tau'}^{\infty} p_j(y)\mathrm{d}y = 1 - \Phi\left(\frac{\tau' - \mu_j}{\sigma}\right) \\ &= \begin{cases} 1 - \Phi\left(\frac{\ln(\tau)}{d} + \frac{d}{2}\right), & j = 0 \\ 1 - \Phi\left(\frac{\ln(\tau)}{d} - \frac{d}{2}\right), & j = 1, \end{cases} \end{aligned}$$

where $\Phi(\cdot)$ denotes the cumulative probability distribution function (CDF) of a $\mathcal{N}(0,1)$ random variable (RV) and $d = (\mu_1 - \mu_0)/\sigma$ defined by

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} \exp\left(-\frac{y^2}{2}\right)\mathrm{d}y.$$

Example 2: Location Testing with Gaussian Error (cont.)

For uniform costs and equal priors, the Bayes risk results to $r(\delta_{\mathrm{B}}) = 1 - \Phi(d/2)$.

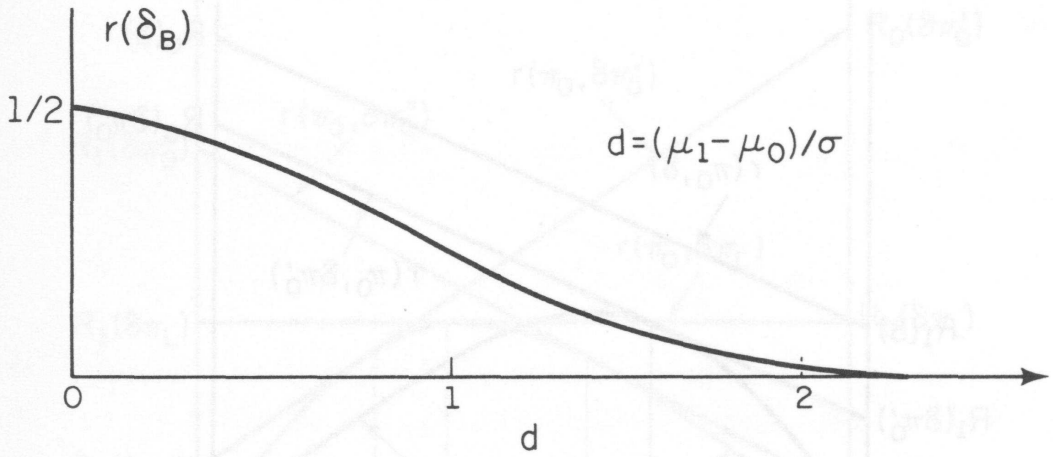


FIGURE II.B.3. Bayes risk in location testing with Gaussian error.

The quantity d is a simple version of a signal-to-noise ratio (SNR).

Summary

- observation set Γ, observation $\mathbf{Y} \in \Gamma$
- H_0, H_1 : two hypotheses with prior probabilities π_0 and π_1, respectively
- decision rule δ : partition of Γ into Γ_0 and Γ_1, and

$$\delta(\mathbf{y}) = \begin{cases} 0 & \text{if } \mathbf{y} \in \Gamma_0 \\ 1 & \text{if } \mathbf{y} \in \Gamma_1 \end{cases}$$

- $R_j(\delta)$: risk when hypothesis H_j is true and decision rule δ applied
- $r(\delta) = \pi_0 R_0(\delta) + \pi_1 R_1(\delta)$: *Bayes risk* for decision rule δ
- *Bayes rule* δ_{B} : decision rule minimizing Bayes risk over all possible rules
- likelihood ratio : $L(\mathbf{y}) = P(\mathbf{y}|H_1)/P(\mathbf{y}|H_0)$
- formulation of Bayes rule using likelihood ratio:

$$\delta_{\mathrm{B}}(\mathbf{y}) = \begin{cases} 1 & \text{if } L(\mathbf{y}) \geq \tau \\ 0 & \text{if } L(\mathbf{y}) < \tau \end{cases}$$

- threshold τ : for uniform cost assignment $\tau = \pi_0/\pi_1$

Minimax Hypothesis Testing

Conditional Bayes Risk

What do we do if the prior probabilities π_0 and π_1 are unknown?

Smoke detector example: the probability π_1 that a fire occurs may be difficult to quantify.

Bayes risk as a function of the prior probability:

- let
$$r(\pi_0, \delta) = \pi_0 R_0(\delta) + (1 - \pi_0) R_1(\delta)$$
denote Bayes risk for a certain decision rule δ for given prior probability $\pi_0 \in [0, 1]$
- $r(\pi_0, \delta)$ is an affine function of the prior π_0, hence
$$\max_{0 \le \pi_0 \le 1} r(\pi_0, \delta) = \max\{R_0(\delta), R_1(\delta)\}$$

Idea of *minimax hypothesis testing*:

minimize, over all δ, the maximal risk $\max_{0 \le \pi_0 \le 1} r(\pi_0, \delta)$

Minimax Rule

- a decision rule δ_{L} satisfying

$$\max_{0\leq\pi_0\leq 1} r(\pi_0, \delta_{\mathrm{L}}) = \min_{\delta} \max_{0\leq\pi_0\leq 1} r(\pi_0, \delta)$$

  is a *minimax rule*

- alternative formulation of the condition:

$$\max\{R_0(\delta_{\mathrm{L}}), R_1(\delta_{\mathrm{L}})\} = \min_{\delta} \max\{R_0(\delta), R_1(\delta)\}$$

How can we construct a minimax rule?

Minimal Bayes Risk

Let

- δ_{π_0} denote a Bayes rule corresponding to prior π_0
- $V(\pi_0) = r(\pi_0, \delta_{\pi_0})$ represent the minimum possible Bayes risk for prior π_0

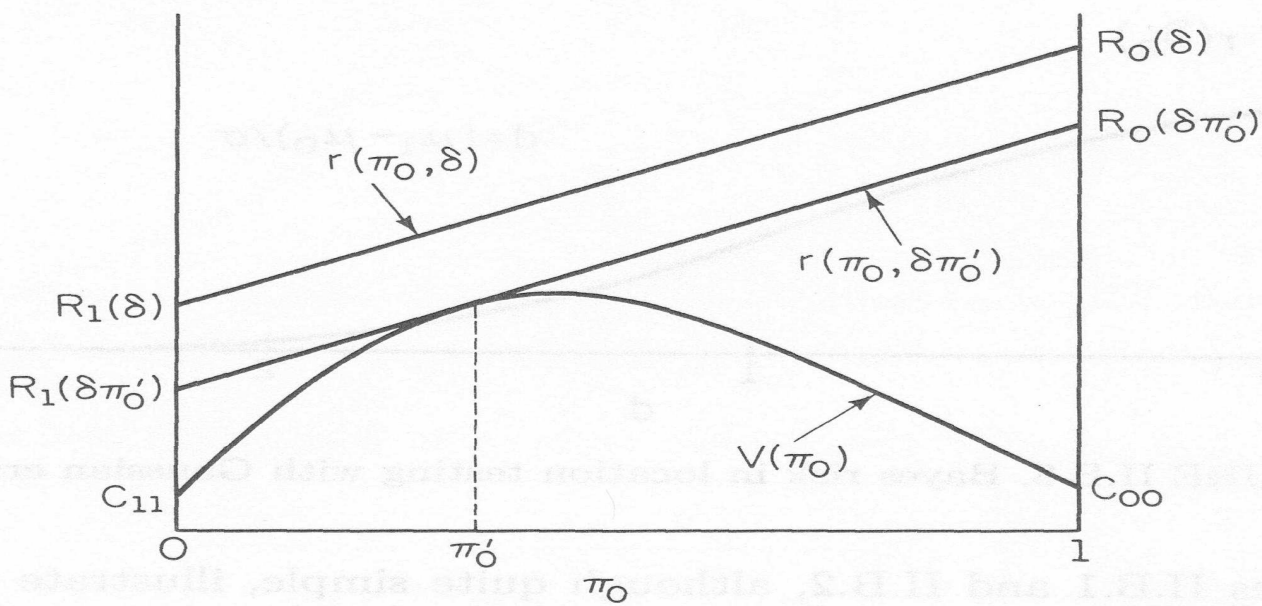


FIGURE II.C.1. Illustration of the functions $r(\pi_0, \delta)$ and $V(\pi_0)$.

What are the properties of $V(\pi_0)$?

Properties of Function $V(\pi)$

Property 1: if $C_{11} < C_{01}$ and $C_{00} < C_{10}$ then $V(0) = C_{11}$ and $V(1) = C_{00}$.

Proof:

- if hypothesis H_1 was true with probability 1 (i.e., $\pi_0 = 0$) then a minimal Bayes risk of C_{11} is attained by always choosing H_1
- if hypothesis H_0 was true with probability 1 (i.e., $\pi_0 = 1$) then a minimal Bayes risk of C_{00} is attained by always choosing H_0

Property 2: for any given rule δ:

$$r(\pi_0, \delta) \geq V(\pi_0) \qquad \text{for all } \pi_0 \in [0, 1].$$

Proof: obvious.

Properties of Function $V(\pi)$ (cont.)

Property 3: $V(\pi_0)$ is a concave function over $[0,1]$.

Recall: a real-valued function $f(x)$ is *concave* if for any x_0, x_1 in the domain of f

$$f(\alpha x_0 + (1-\alpha)x_1) \geq \alpha f(x_0) + (1-\alpha) f(x_1) \quad \text{for all } \alpha \in [0,1]$$

Proof: suppose $V(\pi_0)$ was not concave:

- then there exist $\pi_a, \pi_b \in [0,1]$ and $\pi_0' = \alpha\pi_a + (1-\alpha)\pi_b$ with $\alpha \in (0,1)$ such that

$$V(\pi_0') < \alpha V(\pi_a) + (1-\alpha) V(\pi_b)$$

- consider now Bayes rule $\delta_{\pi_0'}$, and the Bayes risk $r(\pi_0, \delta_{\pi_0'})$ of that rule for prior π_0
- since $r(\pi_0, \delta_{\pi_0'})$ is an affine function of π_0, and since

$$r(\pi_0', \delta_{\pi_0'}) = V(\pi_0') < \alpha V(\pi_a) + (1-\alpha) V(\pi_b),$$

  we conclude that either $r(\pi_a, \delta_{\pi_0'}) < V(\pi_a)$ or $r(\pi_b, \delta_{\pi_0'}) < V(\pi_b)$

- this violates property 2, hence, $V(\pi_0)$ must be concave!

Properties of Function $V(\pi)$ (cont.)

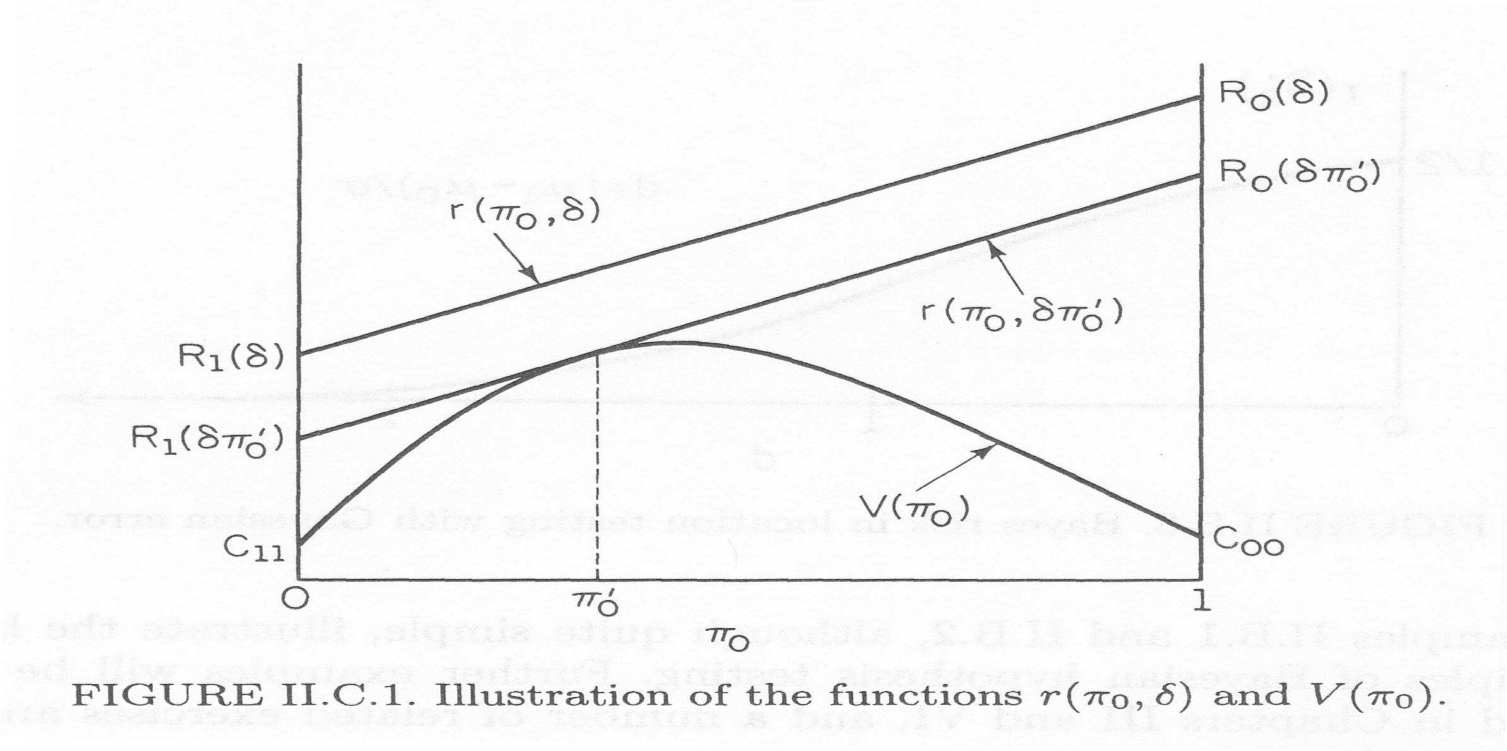


FIGURE II.C.1. Illustration of the functions $r(\pi_0, \delta)$ and $V(\pi_0)$.

Property 4: a rule δ for which $r(\pi_0, \delta) > V(\pi_0)$ for all $\pi_0 \in [0, 1]$ cannot be a Bayes rule.

Proof: obvious.

Equalizer Rule

Let the *least favorable prior* π_{L} represent the prior probability at which $V(\cdot)$ is maximal.

Case 1: $0 < \pi_{\mathrm{L}} < 1$ and $V(\pi_0)$ is differentiable at $\pi_0 = \pi_{\mathrm{L}}$

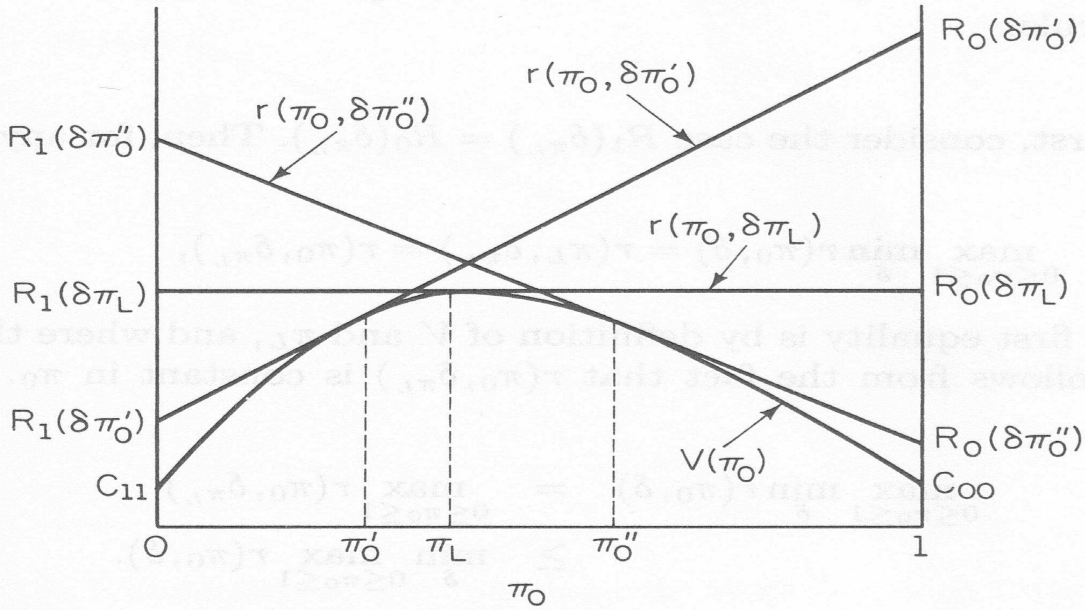


FIGURE II.C.2. Illustration of the minimax rule when V has an interior maximum.

- $\delta_{\pi_{\mathrm{L}}}$ is called an *equalizer rule* because $r(\pi_0, \delta_{\pi_{\mathrm{L}}})$ is constant over $\pi_0 \in [0, 1]$
- $\delta_{\pi_{\mathrm{L}}}$ is a minimax rule

Endpoint Maximum

Case 2: $\pi_{\mathrm{L}} = 0$ or $\pi_{\mathrm{L}} = 1$

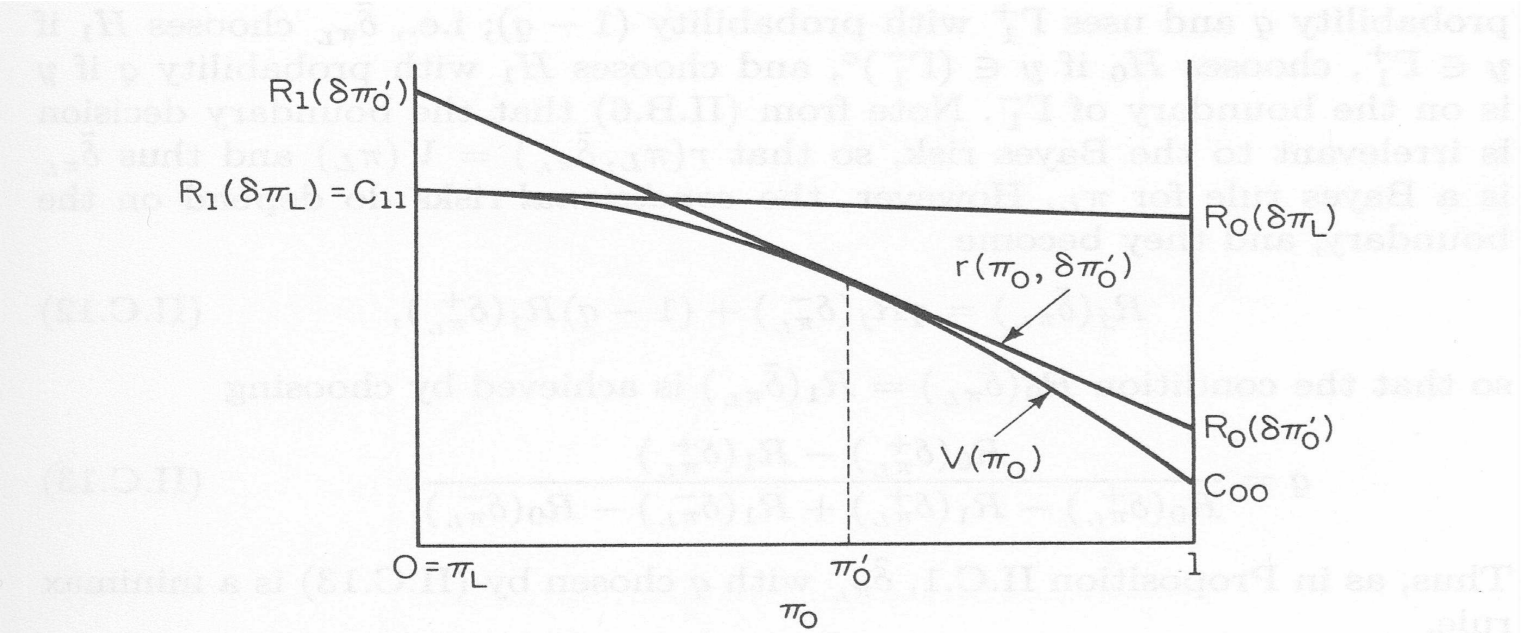


FIGURE II.C.3. Depiction of the minimax rule when V has an endpoint maximum.

- $r(\pi_0, \delta_{\pi_{\mathrm{L}}})$ has a maximum at $\pi_0 = \pi_{\mathrm{L}}$
- $\delta_{\pi_{\mathrm{L}}}$ is a minimax rule

Randomized Minimax Rule

Case 3: $V(\pi)$ is not differentiable at $\pi_{\mathrm{L}} \in (0,1)$

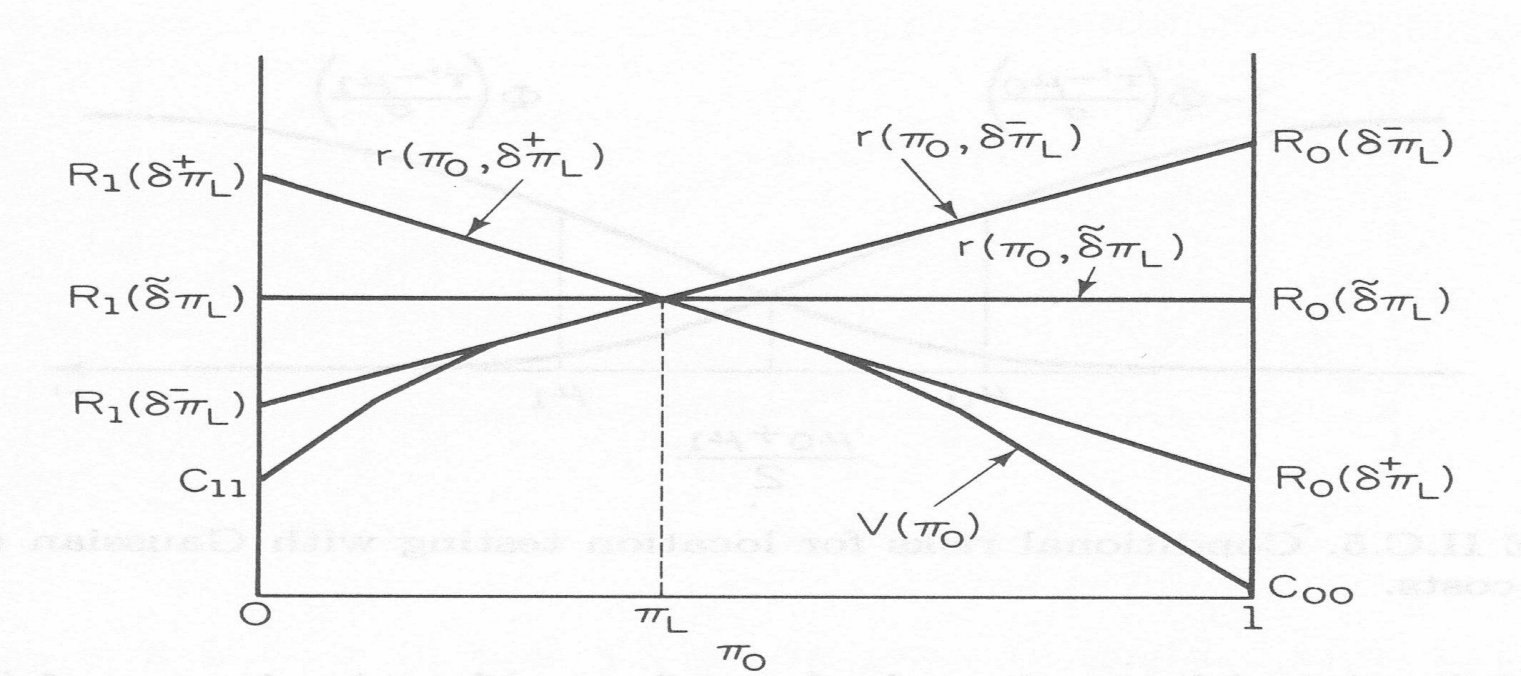


FIGURE II.C.4. Depiction of a randomized decision rule.

- define a *randomized decision rule* $\tilde{\delta}_{\pi_{\mathrm{L}}}$ according to which $\delta^{-}_{\pi_{\mathrm{L}}}$ is applied with probability $q \in [0,1]$ and $\delta^{+}_{\pi_{\mathrm{L}}}$ is applied with probability $(1-q)$
- choose q such that $r(\pi_0, \tilde{\delta}_{\pi_{\mathrm{L}}}) = qr(\pi_0, \delta^{-}_{\pi_{\mathrm{L}}}) + (1-q)r(\pi_0, \delta^{+}_{\pi_{\mathrm{L}}})$ is constant in π_0
- $\tilde{\delta}_{\pi_{\mathrm{L}}}$ is then a minimax rule

Summary

- $r(\pi_0, \delta)$: Bayes risk for a certain decision rule δ for given prior probability $\pi_0 \in [0, 1]$

$$r(\pi_0, \delta) = \pi_0 R_0(\delta) + (1 - \pi_0) R_1(\delta) \quad \text{is affine function of } \pi_0$$

$$\rightarrow \max_{0 \leq \pi_0 \leq 1} r(\pi_0, \delta) = \max\{R_0(\delta), R_1(\delta)\}$$

- minimax rule δ_{L} : decision rule satisfying

$$\max_{0 \leq \pi_0 \leq 1} r(\pi_0, \delta_{\mathrm{L}}) = \min_{\delta} \max_{0 \leq \pi_0 \leq 1} r(\pi_0, \delta)$$

or

$$\max\{R_0(\delta_{\mathrm{L}}), R_1(\delta_{\mathrm{L}})\} = \min_{\delta} \max\{R_0(\delta), R_1(\delta)\}$$

- δ_{π_0} : Bayes rule corresponding to prior probability π_0
- $V(\pi_0) = r(\pi_0, \delta_{\pi_0})$: concave function reflecting minimum Bayes risk versus π_0
- if $V(\pi_0)$ has a maximum at $\pi_{\mathrm{L}} \in [0, 1]$ and if either $\pi_{\mathrm{L}} \in \{0, 1\}$ or $V(\pi_0)$ is differentiable at π_{L}, then $\delta_{\pi_{\mathrm{L}}}$ is a minimax rule with risk $V(\pi_{\mathrm{L}})$
- if $V(\pi_0)$ is not differentiable at $\pi_{\mathrm{L}} \in (0, 1)$, randomization leads to a minimax rule

Neyman-Pearson Hypothesis Testing

Detection, False Alarm, and Miss

A decision rule is often employed to detect an "exceptional condition" (represented by H_1).

For many asymmetrical scenarios – for instance in radar systems – the following terms are appropriate:

- *detection*: correct acceptance of H_1
- *false alarm*: false acceptance of H_1
- *miss*: false acceptance of H_0

Instead of imposing a specific cost structure, it is sometimes more appropriate to design a decision rule δ on the basis of

- *detection probability*: $P_{\mathrm{D}}(\delta) = P(\delta(\mathbf{Y}) = 1 \mid H_1)$
- *false alarm probability*: $P_{\mathrm{F}}(\delta) = P(\delta(\mathbf{Y}) = 1 \mid H_0)$
- *miss probability*: $P_{\mathrm{M}}(\delta) = 1 - P_{\mathrm{D}}(\delta)$

How can we address the trade-off between false alarm probability and miss probability?

Randomized Decision Rules

Randomized decision rule $\tilde{\delta} : \Gamma \to [0, 1]$:

- when $\tilde{\delta}(\mathbf{y}) = 0$ choose hypothesis H_0
- when $\tilde{\delta}(\mathbf{y}) = 1$ choose hypothesis H_1
- when $\tilde{\delta}(\mathbf{y}) = q \in (0, 1)$ choose
  - hypothesis H_1 with probability q
  - hypothesis H_0 with probability $1 - q$

Using $\tilde{\delta}(\mathbf{y})$,

- detection probability: $P_{\mathrm{D}}\left(\tilde{\delta}\right) = E\left(\tilde{\delta}(\mathbf{Y}) \mid H_1\right)$
- false alarm probability: $P_{\mathrm{F}}\left(\tilde{\delta}\right) = E\left(\tilde{\delta}(\mathbf{Y}) \mid H_0\right)$
- miss probability: $P_{\mathrm{M}}\left(\tilde{\delta}\right) = 1 - P_{\mathrm{D}}\left(\tilde{\delta}\right)$

Neyman-Pearson Criterion

Neyman-Pearson hypothesis testing:

- placing a bound α on the false alarm probability $P_{\mathrm{F}}\left(\tilde{\delta}\right)$
- minimizing the miss probability $P_{\mathrm{M}}\left(\tilde{\delta}\right)$ subject to $P_{\mathrm{F}}\left(\tilde{\delta}\right) \leq \alpha$
- a decision rule $\tilde{\delta}_{\mathrm{NP}}$ with $P_{\mathrm{F}}\left(\tilde{\delta}_{\mathrm{NP}}\right) = \alpha$ and a detection probability $P_{\mathrm{D}}\left(\tilde{\delta}_{\mathrm{NP}}\right)$ equal to

$$\max_{\tilde{\delta}} P_{\mathrm{D}}\left(\tilde{\delta}\right) \quad \text{subject to} \quad P_{\mathrm{F}}\left(\tilde{\delta}\right) \leq \alpha$$

  is called an α-*level Neyman-Pearson* test

Neyman-Pearson lemma: for every $\alpha \in (0,1)$ there is a Neyman-Pearson test of the form

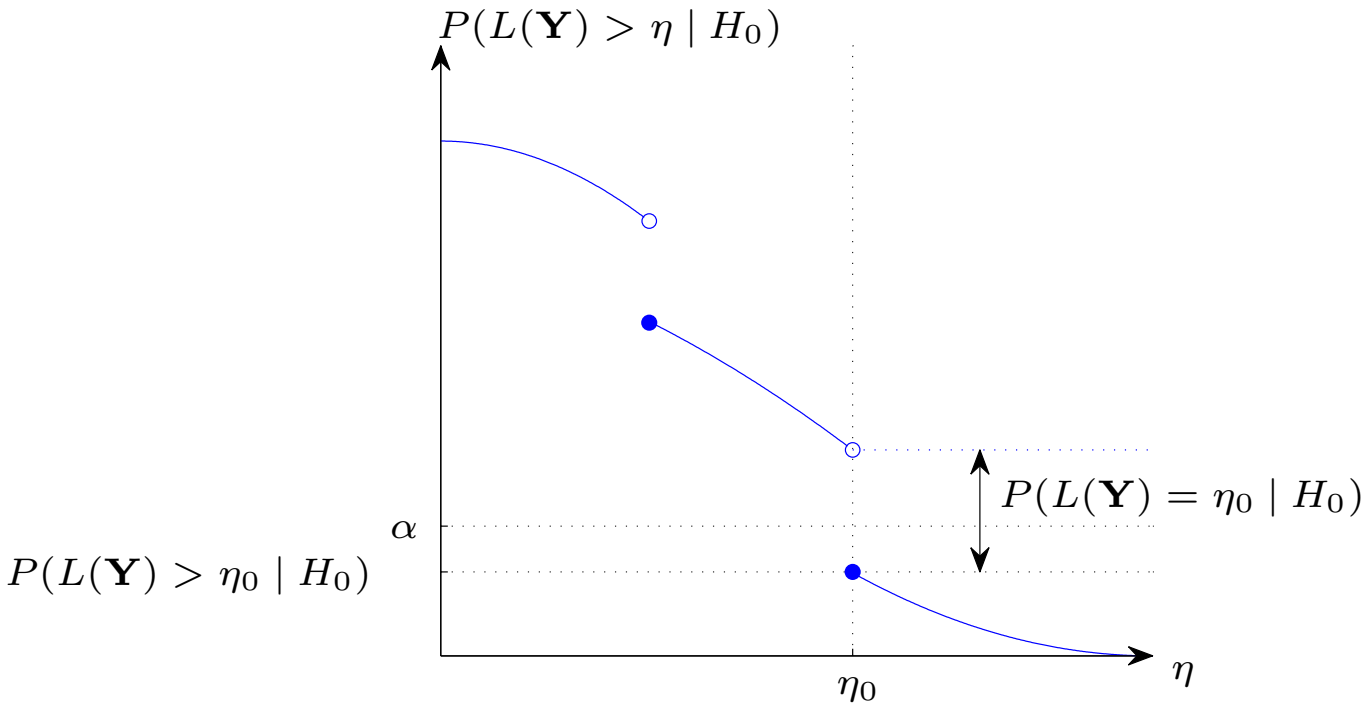
$$\tilde{\delta}_{\mathrm{NP}}(\mathbf{y}) = \begin{cases} 1 & \text{if } L(\mathbf{y}) > \eta_0 \\ \gamma_0 & \text{if } L(\mathbf{y}) = \eta_0 \\ 0 & \text{if } L(\mathbf{y}) < \eta_0 \end{cases}$$

with likelihood ratio threshold η_0 and randomization constant γ_0, satisfying $P_{\mathrm{F}}\left(\tilde{\delta}_{\mathrm{NP}}\right) = \alpha$.

Construction of Neyman-Pearson Test

Construction of a Neyman-Pearson decision rule $\tilde{\delta}_{\mathrm{NP}}$:

1. choose η_0 as the smallest number $\eta \in \mathbb{R}$ for which $P(L(\mathbf{Y}) > \eta \mid H_0) \leq \alpha$



2. choose the randomization constant

$$\gamma_0 = \frac{\alpha - P(L(\mathbf{Y}) > \eta_0 \mid H_0)}{P(L(\mathbf{Y}) = \eta_0 \mid H_0)}$$

Summary

- Neyman-Pearson test has form of a randomized rule

$$\tilde{\delta}_{\mathrm{NP}}(\mathbf{y}) = \begin{cases} 1 & \text{if } L(\mathbf{y}) > \eta_0 \\ \gamma_0 & \text{if } L(\mathbf{y}) = \eta_0 \\ 0 & \text{if } L(\mathbf{y}) < \eta_0 \end{cases}$$

  with appropriate likelihood ratio threshold η_0 and randomization constant $\gamma_0 \in [0, 1]$

- false-alarm probability : $P_{\mathrm{F}}\left(\tilde{\delta}\right) = E\left(\tilde{\delta}(\mathbf{Y}) \mid H_0\right)$
- detection probability : $P_{\mathrm{M}}\left(\tilde{\delta}\right) = E\left(\tilde{\delta}(\mathbf{Y}) \mid H_1\right)$
- miss probability : $P_{\mathrm{M}}\left(\tilde{\delta}\right) = 1 - P_{\mathrm{M}}\left(\tilde{\delta}\right)$
- decision rule $\tilde{\delta}_{\mathrm{NP}}$ is an α-level Neyman-Pearson test if $P_{\mathrm{D}}\left(\tilde{\delta}_{\mathrm{NP}}\right)$ equals

$$\max_{\tilde{\delta}} P_{\mathrm{D}}\left(\tilde{\delta}\right) \quad \text{subject to} \quad P_{\mathrm{F}}\left(\tilde{\delta}\right) \leq \alpha$$

- construction of a Neyman-Pearson rule $\tilde{\delta}_{\mathrm{NP}}$ with $P_{\mathrm{F}}\left(\tilde{\delta}_{\mathrm{NP}}\right) = \alpha$:
  1. choose for threshold η_0 smallest number $\eta \in \mathbb{R}$ for which $P(L(\mathbf{Y}) > \eta \mid H_0) \leq \alpha$
  2. choose for randomization constant $\gamma_0 = \frac{\alpha - P(L(\mathbf{Y}) > \eta_0 \mid H_0)}{P(L(\mathbf{Y}) = \eta_0 \mid H_0)}$

Signal Detection in Discrete Time

Introduction

- Here, we apply the aforementioned approaches to the detection of signals in discrete time.
- Again, our basic physical observation model that we wish to consider is that of an observed continuous-time waveform that consists of one of two possible signals corrupted by **additive** noise.
- We consider a finite number (say n) of samples taken from the observed waveform.
- We can formulate the detection problem by the hypothesis pair for the observation space $(\Gamma, \mathcal{G}) = (\mathbb{R}^n, \mathcal{B}^n)$:

$$\begin{aligned} H_0: \quad & Y_k = N_k + S_{0k}, \qquad k = 1, \ldots, n \\ \text{versus} \\ H_1: \quad & Y_k = N_k + S_{1k}, \qquad k = 1, \ldots, n. \end{aligned}$$

Introduction (cont.)

- Here, $\mathbf{Y} = [Y_1, \ldots, Y_n]^{\mathrm{T}}$ is an observation column vector consisting of the samples of the observed waveform, $\mathbf{N} = [N_1, \ldots, N_n]^{\mathrm{T}}$ is a vector of noise samples, and $\mathbf{S}_j = [S_{j1}, \ldots, S_{jn}]^{\mathrm{T}}$, $j = 0, 1$, are vectors of samples from the two possible signals.
- Clearly, the index k can be interpreted both as a discrete-time index as well as a discrete-space index (e.g. indicating the output of spatially seperated signal sensors) or an index enumerating the outputs of the bank of n parallel filters (employed e.g. in a broadband wireless system receiver with signal transmission over a multipath channel).
- The signal vectors $\mathbf{S}_j$ can be completely known (i.e. deterministic)

Detection of Deterministic Signals in Gaussian Noise

- If the noise samples N_k are not mutually independent, then the optimum test has no particular structure that would allow for a simple implementation and a characterization of the resulting detector.
- An important exception is the situation where the noise vector $\mathbf{N}$ has a multivariate Gaussian distribution. The assumption of Gaussian noise is often justifiable in practice, and the detectors which can be derived are intuitively reasonable systems to use even when the noise is not Gaussian.
- Without restriction of generality, suppose that $\mathbf{N}$ is a Gaussian random vector with zero mean vector $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma}_N$.
- The probability density function of an n-dimensional Gaussian random vector $\mathbf{X} \in \mathbb{R}^n$ with mean vector $\boldsymbol{\mu} = E\{\mathbf{X}\} \in \mathbb{R}^n$ and $(n \times n)$-dimensional covariance matrix $\mathbf{\Sigma} = E\left\{(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^{\mathrm{T}}\right\}$ is given by

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}\,|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}, \qquad \mathbf{x} \in \mathbb{R}^n.$$

Detection of Deterministic Signals in Gaussian Noise (cont.)

- Here, $|\mathbf{\Sigma}|$ denotes the determinant of $\mathbf{\Sigma}$ and $\mathbf{\Sigma}^{-1}$ is its inverse. We denote a Gaussian distributed vector as $\mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$.
- Note that a covariance matrix is always non-negative definite. Clearly, for arbitrary $\mathbf{a} \in \mathbb{R}^n$, we have for a random vector $\mathbf{X}$ with mean $\boldsymbol{\mu}$ that $E\left\{(\mathbf{a}^{\mathrm{T}}(\mathbf{X}-\boldsymbol{\mu}))^2\right\} = E\left\{\mathbf{a}^{\mathrm{T}}(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{a}\right\} = \mathbf{a}^{\mathrm{T}}\mathbf{\Sigma}\mathbf{a} \geq 0$. Furthermore, from the definition of the covariance matrix, it follows directly that the covariance matrix as well as its inverse are symmetric.
- In the aforementioned Gaussian multivariate PDF, we assume that the covariance matrix $\mathbf{\Sigma}$ is indeed positive definite, which implies that $|\mathbf{\Sigma}| > 0$ and that $\mathbf{\Sigma}^{-1}$ exists. If $\mathbf{\Sigma}$ is not positive definite, at least one of the components of $\mathbf{X}$ can be written as a linear combination of the others and is thus redundant. If not stated otherwise, we will assume that $\mathbf{\Sigma}$ is positive definite.
- Obviously, the observation under hypothesis H_j, $j = 0, 1$, is Gaussian distributed with $\mathbf{Y} \sim \mathcal{N}(\mathbf{s}_j, \mathbf{\Sigma}_N)$.

Detection of Deterministic Signals in Gaussian Noise (cont.)

- We obtain for the log-likelihood ratio

$$\begin{aligned}
\log L(\mathbf{y}) &= \log \frac{p_1(\mathbf{y})}{p_0(\mathbf{y})} \\
&= \log \frac{\frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_N|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{s}_1)^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}(\mathbf{y}-\mathbf{s}_1)\right)}{\frac{1}{(2\pi)^{n/2}|\mathbf{\Sigma}_N|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{s}_0)^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}(\mathbf{y}-\mathbf{s}_0)\right)} \\
&= \mathbf{s}_1^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}\mathbf{y} - \mathbf{s}_0^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}\mathbf{y} - \frac{1}{2}\mathbf{s}_1^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}\mathbf{s}_1 + \frac{1}{2}\mathbf{s}_0^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}\mathbf{s}_0 \\
&= (\mathbf{s}_1-\mathbf{s}_0)^{\mathrm{T}}\mathbf{\Sigma}_N^{-1}\left(\mathbf{y} - \frac{\mathbf{s}_0+\mathbf{s}_1}{2}\right), \qquad \mathbf{y} \in \mathbb{R}^n.
\end{aligned}$$

(Comparing the log-likelihood expression with the Gaussian location testing problem (cf. slide #51), we observe that the both expressions are identical if we replace μ_j by $\mathbf{s}_j$, $j = 0, 1$, and σ^2 by $\mathbf{\Sigma}_N$.

Detection of Deterministic Signals in Gaussian Noise (cont.)

- To simplify the test, we can absorb the term $\frac{1}{2}(\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \boldsymbol{\Sigma}_N^{-1}(\mathbf{s}_0 + \mathbf{s}_1)$ into the threshold.
- Thus, the optimum test for deciding between

$$H_0: \quad Y_k = N_k + S_{0k}, \qquad k = 1, \ldots, n$$

versus

$$H_1: \quad Y_k = N_k + S_{1k}, \qquad k = 1, \ldots, n$$

with $\mathbf{N} = [N_1, \ldots, N_n]^{\mathrm{T}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_N)$ is given by

$$\tilde{\delta}_0(\mathbf{y}) = \begin{cases} 1 & \\ \gamma & \text{if } (\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \boldsymbol{\Sigma}_N^{-1} \mathbf{y} \quad \begin{matrix} > \\ = \\ < \end{matrix} \quad \tau' \\ 0 & \end{cases}$$

with $\tau' = \ln \tau + \frac{1}{2}(\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \boldsymbol{\Sigma}_N^{-1}(\mathbf{s}_0 + \mathbf{s}_1)$.

Detection of Deterministic Signals in Gaussian Noise (cont.)

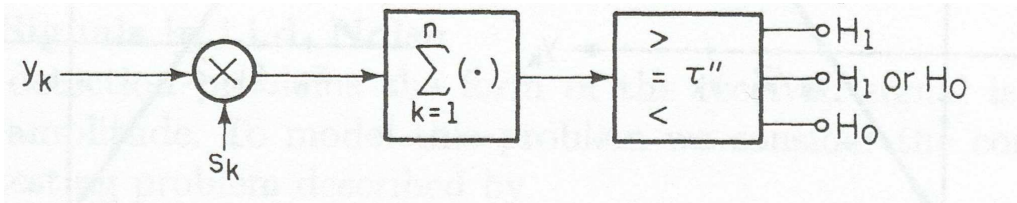
- Upon defining

$$\tilde{\mathbf{s}} = \mathbf{\Sigma}_N^{-1}(\mathbf{s}_1 - \mathbf{s}_0)$$

the test is equivalent to

$$\tilde{\delta}_0(\mathbf{y}) = \begin{cases} 1 & \\ \gamma & \text{if } \tilde{\mathbf{s}}^{\mathrm{T}}\mathbf{y} \quad \begin{matrix} > \\ = \\ < \end{matrix} \quad \tau'. \\ 0 & \end{cases}$$

- Thus, the structure of the optimum detector is identical to the i.i.d. case with s_k replaced by $\tilde{s}_k$ and τ'' replaced by τ':



- Therefore for this Gaussian case, detector implementation is no more difficult for dependent noise than for independent noise.

Detection of Deterministic Signals in Gaussian Noise (cont.)

- The performance of the decision procedure can be characterized by considering the linear transformation
$$T(Y) = \tilde{\mathbf{s}}^{\mathrm{T}} Y,$$
which is a Gaussian random variable under both hypotheses.
- Thus, the mean and variances under H_0 and H_1 determine the PDFs completely.
- We obtain for the means
$$E\left\{T(Y)\,|H_j\right\} = E\left\{\tilde{\mathbf{s}}^{\mathrm{T}} Y\,|H_j\right\} = \tilde{\mathbf{s}}^{\mathrm{T}} E\left\{Y\,|H_j\right\} = \tilde{\mathbf{s}}^{\mathrm{T}} \mathbf{s}_j = \tilde{\mu}_j.$$
- The variances are given by
$$\begin{aligned}\mathrm{Var}(T(Y)\,|H_j) &= E\left\{\left(\tilde{\mathbf{s}}^{\mathrm{T}} Y - \tilde{\mathbf{s}}^{\mathrm{T}} \mathbf{s}_j\right)^2 |H_j\right\} = E\left\{\left(\tilde{\mathbf{s}}^{\mathrm{T}} \mathbf{N}\right)^2\right\} \\ &= E\left\{\tilde{\mathbf{s}}^{\mathrm{T}} \mathbf{N}\,\mathbf{N}^{\mathrm{T}} \tilde{\mathbf{s}}\right\} = (\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \boldsymbol{\Sigma}_N^{-1} (\mathbf{s}_1 - \mathbf{s}_0) = d^2.\end{aligned}$$

Detection of Deterministic Signals in Gaussian Noise (cont.)

- The variances d^2 are independent of H_j and, in view of the positive definiteness of $\mathbf{\Sigma}_N^{-1}$, positive for $\mathbf{s}_1 \neq \mathbf{s}_0$.
- Thus, we have that $T(Y)\,|H_j \sim \mathcal{N}\left(\tilde{\mu}_j, d^2\right)$ for $j = 1, 2$, so that the randomization γ is irrelevant.
- The probability of choosing H_1 under H_j is given by

$$P_j(\Gamma_1) = \frac{1}{\sqrt{2\pi}d} \int_{\tau'}^{\infty} \mathrm{e}^{-\frac{\left(x-\tilde{\mu}_j\right)^2}{2d^2}} \,\mathrm{d}x = 1 - \Phi\left(\frac{\tau' - \tilde{\mu}_j}{d}\right)$$

with $\tau' = \ln\tau + \frac{1}{2}(\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \mathbf{\Sigma}_N^{-1}(\mathbf{s}_0 + \mathbf{s}_1)$ and d being the positive square root of d^2.
(Note that τ', $\tilde{\mu}_j$ and d^2 contain terms of the form $(\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{s}_j$, which makes it possible to express the argument of Φ by τ and d.)

Detection of Deterministic Signals in Gaussian Noise (cont.)

We will consider three aspects of the deterministic signal detection in Gaussian noise in more detail:

1. Interpretation of d^2
2. Reduction to the i.i.d. noise case
3. Optimum signal selection.

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- The structural properties of the covariance matrix play a central role for the structure of the optimum detection system.
- First of all, $\mathbf{\Sigma}_N$ is a positive definite matrix, i.e. for arbitrary n-dimensional vectors $\mathbf{a} \in \mathbb{R}^n$, we have $\mathbf{a}^{\mathrm{T}} \mathbf{\Sigma}_N \mathbf{a} > 0$.
- The eigenvalues $\lambda_1, \ldots, \lambda_n$ and the corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of an $(n \times n)$-dimensional matrix $\mathbf{\Sigma}_N$ are the solutions to the equation $\mathbf{\Sigma}_N \mathbf{v}_k = \lambda_k \mathbf{v}_k$.
- Since $\mathbf{\Sigma}_N$ in our case is symmetric and positive definite, all of its eigenvalues are real and positive and its eigenvectors can be chosen to be orthonormal (i.e., $\mathbf{v}_k^{\mathrm{T}} \mathbf{v}_\ell = 0$ if $k \neq \ell$ and $\mathbf{v}_k^{\mathrm{T}} \mathbf{v}_k = 1$ if for all $\ell, k = 1, \ldots, n$).
- With this choice of eigenvectors we have the spectral decomposition of $\mathbf{\Sigma}_N$ according to

$$\mathbf{\Sigma}_N = \sum_{k=1}^{n} \lambda_k \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}}.$$

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- Proof of the spectral decomposition: For any $\mathbf{x} \in \mathbb{R}^n$, we can write $\mathbf{x} = \sum_{k=1}^{n} c_k \mathbf{v}_k$ with $c_k = \mathbf{v}_k^{\mathrm{T}} \mathbf{x}$, so we have

$$\mathbf{\Sigma}_N \mathbf{x} = \sum_{k=1}^{n} c_k \mathbf{\Sigma}_N \mathbf{v}_k = \sum_{k=1}^{n} \lambda_k \mathbf{v}_k c_k = \sum_{k=1}^{n} \lambda_k \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}} \mathbf{x} = \left(\sum_{k=1}^{n} \lambda_k \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}} \right) \mathbf{x}.$$

- Note that the matrix $\mathbf{v}_k \mathbf{v}_k^{\mathrm{T}}$, when multiplied by a vector $\mathbf{x}$, gives the projection of $\mathbf{x}$ onto $\mathbf{v}_k$.

- From the spectral decomposition, we conclude that $\mathbf{\Sigma}_N^{-1} = \sum_{k=1}^{n} \lambda_k^{-1} \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}}$, so that the optimum detection statistic $T(\mathbf{y})$ is given by

$$\begin{aligned} T(\mathbf{y}) &= \tilde{\mathbf{s}}^{\mathrm{T}} \mathbf{y} = (\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{y} = \sum_{k=1}^{n} \lambda_k^{-1} (\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}} \mathbf{y} \\ &= \sum_{k=1}^{n} \frac{\mathbf{v}_k^{\mathrm{T}} (\mathbf{s}_1 - \mathbf{s}_0)}{\sqrt{\lambda_k}} \frac{\mathbf{v}_k^{\mathrm{T}} \mathbf{y}}{\sqrt{\lambda_k}} = \sum_{k=1}^{n} (\hat{s}_{1k} - \hat{s}_{0k}) \hat{y}_k = (\hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_0)^{\mathrm{T}} \hat{\mathbf{y}}. \end{aligned}$$

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- Here, we have

$$\begin{aligned} \hat{y}_k &= \frac{\mathbf{v}_k^{\mathrm{T}}\mathbf{y}}{\sqrt{\lambda_k}} \qquad k = 1, \ldots, n \\ \hat{s}_{jk} &= \frac{\mathbf{v}_k^{\mathrm{T}}\mathbf{s}_j}{\sqrt{\lambda_k}} \qquad k = 1, \ldots, n \text{ and } j = 0, 1. \end{aligned}$$

- If we interpret these quantities as components of a vector in a vector space of dimension n, we can define the vectors

$$\begin{aligned} \hat{\mathbf{y}} &= \sum_{k=1}^{n} \left(\sqrt{\lambda_k^{-1}}\mathbf{v}_k^{\mathrm{T}}\mathbf{y} \right) \mathbf{e}_k \\ \hat{\mathbf{s}}_j &= \sum_{k=1}^{n} \left(\sqrt{\lambda_k^{-1}}\mathbf{v}_k^{\mathrm{T}}\mathbf{s}_j \right) \mathbf{e}_k \qquad \text{for } j = 0, 1, \end{aligned}$$

  where $\{\mathbf{e}_k\}_{k=1}^{n}$ represents the standard basis for $\mathbb{R}^n$.

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- This actually represents a change of coordinates for the problem. To investigate this in more detail, we can represent the vector in the new coordinates as a linear mapping from the vector in the old coordinates. To this end, we write

$$\hat{\mathbf{Y}} = \mathbf{A}\mathbf{Y},$$

where the matrix $\mathbf{A}$ satisfies

$$\begin{aligned} \mathbf{A} &= \sum_{k=1}^{n} \sqrt{\lambda_k^{-1}}\mathbf{e}_k\mathbf{v}_k^{\mathrm{T}}, \\ \mathbf{A}^{-1} &= \sum_{k=1}^{n} \sqrt{\lambda_k}\mathbf{v}_k\mathbf{e}_k^{\mathrm{T}}. \end{aligned} \tag{1}$$

- Obviously, we change the base vectors from $\{\mathbf{e}_k\}_{k=1}^n$ to $\left\{\sqrt{\lambda_k}\mathbf{v}_k\right\}_{k=1}^n$.

- Since this represents an invertible linear transformation from $\mathbf{Y}$ to $\hat{\mathbf{Y}}$, the original problem can be rewritten equivalently in terms of $\hat{\mathbf{Y}}$:

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

$$H_0: \quad \hat{\mathbf{Y}} = \hat{\mathbf{N}} + \hat{\mathbf{s}}_0,$$

versus

$$H_1: \quad \hat{\mathbf{Y}} = \hat{\mathbf{N}} + \hat{\mathbf{s}}_1,$$

where the noise vector $\hat{\mathbf{N}} = \mathbf{A}\mathbf{N}$ is still Gaussian with mean $\mathbf{0}$. However, the covariance matrix of $\hat{\mathbf{N}}$ is given by

$$\begin{aligned}
E\left\{\hat{\mathbf{N}}\,\hat{\mathbf{N}}^{\mathrm{T}}\right\} &= E\left\{\mathbf{A}\mathbf{N}\,\mathbf{N}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\right\} = \left(\sum_{k=1}^{n}\sqrt{\lambda_k^{-1}}\mathbf{e}_k\mathbf{v}_k^{\mathrm{T}}\right) E\left\{\mathbf{N}\,\mathbf{N}^{\mathrm{T}}\right\} \left(\sum_{j=1}^{n}\sqrt{\lambda_j}\mathbf{v}_j\mathbf{e}_j^{\mathrm{T}}\right) \\
&= \left(\sum_{k=1}^{n}\sqrt{\lambda_k^{-1}}\mathbf{e}_k\mathbf{v}_k^{\mathrm{T}}\right)\left(\sum_{j=1}^{n}\sqrt{\lambda_j}(\mathbf{\Sigma}_N\mathbf{v}_j)\mathbf{e}_j^{\mathrm{T}}\right) \\
&= \left(\sum_{k=1}^{n}\sqrt{\lambda_k^{-1}}\mathbf{e}_k\mathbf{v}_k^{\mathrm{T}}\right)\left(\sum_{j=1}^{n}\sqrt{\lambda_j}(\lambda_j\mathbf{v}_j)\mathbf{e}_j^{\mathrm{T}}\right) \\
&= \sum_{k=1}^{n}\sum_{j=1}^{n}\sqrt{\lambda_k^{-1}}\sqrt{\lambda_j}\mathbf{e}_k\mathbf{v}_k^{\mathrm{T}}\mathbf{v}_j\mathbf{e}_j^{\mathrm{T}} = \sum_{k=1}^{n}\mathbf{e}_k\mathbf{e}_k^{\mathrm{T}} = \mathbf{I}.
\end{aligned}$$

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- Result: We have transformed the problem with dependent Gaussian noise into an equivalent problem with i.i.d. Gaussian noise, where $T(\mathbf{y}) = \sum_{k=1}^{n} (\hat{s}_{1k} - \hat{s}_{0k})\hat{y}_k = (\hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_0)^{\mathrm{T}}\hat{\mathbf{y}}$ provides the optimum detection statistic for the transformed problem. Note that the signals $\mathbf{s}_j$ for $j = 0, 1$ are transformed in the same way as the observation $\mathbf{y}$.
- The filtering (or correlation) is being carried out in the transformed domain.
- A similar approach can be derived using, however, a more direct way. To this end, denote the covariance matrix of $\mathbf{N}$ as $\mathbf{\Sigma}_N = \mathbf{B}^2$, where the matrix

$$\mathbf{B} = \sum_{k=1}^{n} \lambda_k^{1/2} \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}}$$

  is called the square root of $\mathbf{\Sigma}_N$.
- This matrix has an inverse $\mathbf{B}^{-1} = \sum_{k=1}^{n} \lambda_k^{-1/2} \mathbf{v}_k \mathbf{v}_k^{\mathrm{T}} = \left(\mathbf{B}^{-1}\right)^{\mathrm{T}}$.

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- Then, the inverse covariance matrix is given by $\mathbf{\Sigma}_N^{-1} = \mathbf{B}^{-2} = \left(\mathbf{B}^{-1}\right)^2$.
- As a result, we can represent the decision variable as

$$(\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} \mathbf{y} = (\mathbf{s}_1^* - \mathbf{s}_0^*)^{\mathrm{T}} \mathbf{y}^*.$$

- Now, the transformed vectors are for $j = 0, 1$

$$\mathbf{Y}^* = \sum_{k=1}^{n} \hat{Y}_k \mathbf{v}_k = \mathbf{N}^* + \mathbf{s}_j^* = \mathbf{B}^{-1}\mathbf{N} + \mathbf{B}^{-1}\mathbf{s}_j.$$

- Again, the covariance matrix of the transformed problem is

$$E\left\{\mathbf{N}^* \left(\mathbf{N}^*\right)^{\mathrm{T}}\right\} = E\left\{\mathbf{B}^{-1}\mathbf{N}\,\mathbf{N}^{\mathrm{T}}\mathbf{B}^{-1}\right\} = \mathbf{B}^{-1}\mathbf{\Sigma}_N\mathbf{B}^{-1} = \mathbf{B}^{-1}\mathbf{B}\mathbf{B}\mathbf{B}^{-1} = \mathbf{I}.$$

- Thus, the $\mathbf{N}^*$ has i.i.d. components, which is clear since $\mathbf{Y}^*$ and $\hat{\mathbf{Y}}$ are the same random vectors in two different coordinate systems.

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- The observation vector $\mathbf{Y}$ can be transformed in another interesting way to give an equivalent observation with i.i.d. noise. To this end, we can write

$$\mathbf{\Sigma}_N = \mathbf{C}\mathbf{C}^{\mathrm{T}},$$

  where $\mathbf{C}$ is an $(n \times n)$-dimensional invertible, lower triangular matrix (i.e., all above-diagonal elements of $\mathbf{C}$ are zero).

- This is called the Cholesky decomposition of $\mathbf{\Sigma}_N$ and there are several standard algorithms for finding $\mathbf{C}$ from $\mathbf{\Sigma}_N$.

- As above, we have to investigate $\mathbf{\Sigma}_N^{-1}$, which is given by

$$\mathbf{\Sigma}_N^{-1} = \left(\mathbf{C}^{\mathrm{T}}\right)^{-1}\mathbf{C}^{-1} = \left(\mathbf{C}^{-1}\right)^{\mathrm{T}}\mathbf{C}^{-1}.$$

- On defining new observables $\bar{\mathbf{Y}} = \mathbf{C}^{-1}\mathbf{Y} = \mathbf{C}^{-1}\mathbf{N} + \mathbf{C}^{-1}\mathbf{s}_j = \bar{\mathbf{N}} + \bar{\mathbf{s}}_j$, we have straightforwardly that $\bar{\mathbf{N}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

- Again, we have an i.i.d. noise situation and the optimum detection statistic is $\left(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}_0\right)^{\mathrm{T}}\bar{\mathbf{Y}}$.

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- What is special about the transformation $\mathbf{C}^{-1}\mathbf{Y}$?
- The lower triangularity of $\mathbf{C}$ implies that $\mathbf{C}^{-1}$ is also lower triangular. This in turn implies that we can write $\bar{y}_k = \sum_{\ell=1}^{k} h_{k,\ell} y_\ell$, where $h_{k\ell}$ is the (k,ℓ)-th element of $\mathbf{C}^{-1}$. Note that the expression above represents a causal operation. Indeed, it is a causal, but possibly time-varying linear filtering operation.
- Since the noise at the output of the filter is white (i.e. i.i.d.), this filter is sometimes known as whitening filter.

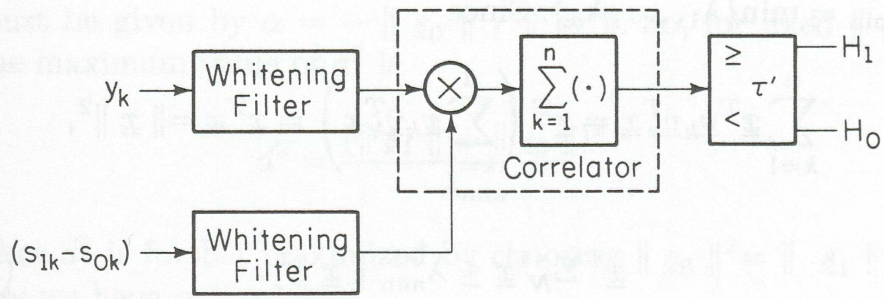


FIGURE III.B.8. Optimum detector for coherent signals in dependent Gaussian noise.

Detection of Deterministic Signals in Gaussian Noise (cont.): Reduction to i.i.d. Noise

- Note that the signal-to-noise ratio $d^2 = (\mathbf{s}_1 - \mathbf{s}_0)^{\mathrm{T}} \mathbf{\Sigma}_N^{-1} (\mathbf{s}_1 - \mathbf{s}_0)$ can be written in terms of any of the transformed signal pairs as

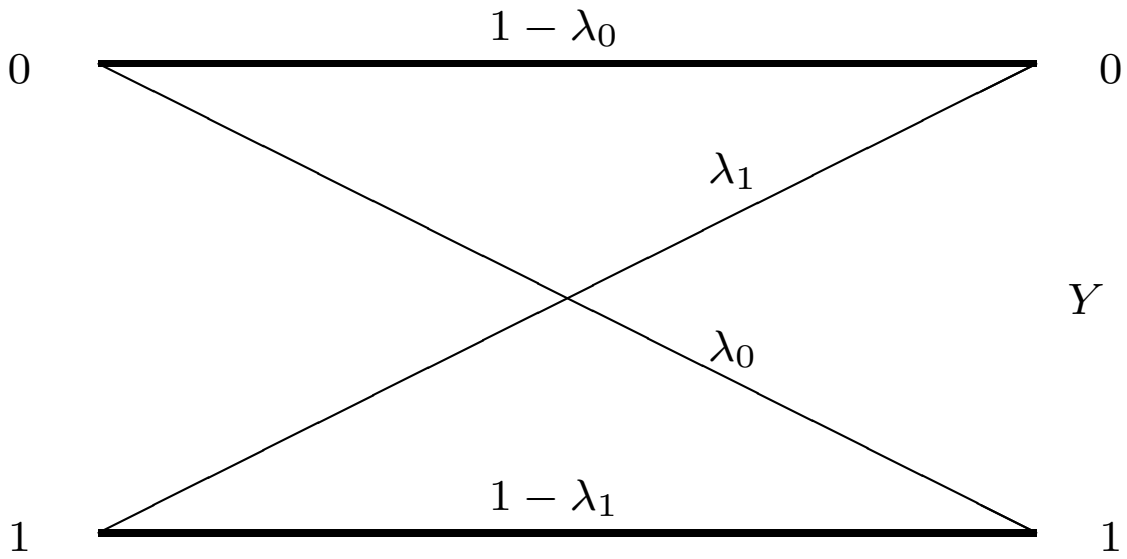
$$d^2 = ||\hat{\mathbf{s}}_1 - \hat{\mathbf{s}}_0||^2 = ||\mathbf{s}_1^* - \mathbf{s}_0^*||^2 = ||\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}_0||^2 .$$

  Thus, the performance of coherent detection in dependent noise depends on how far apart the signals are when transformed to a coordinate system in which the noise components are i.i.d.

- All three signal pairs are the same distance apart since they are all representations of the same pair of vectors in different coordinate systems that are simple rotations of one another.

Exercises

1. Consider the transmission of a binary digit (a '0' or a '1') over an erroneous communication channel. At the output of the channel, which includes a hard decision in the receiver, we observe $Y \in \{0, 1\}$. As illustrated below, a transmitted '0' appears as a '1' with probability λ_0 and as a '1' with probability $1 - \lambda_0$, where $\lambda_0 \in (0, 1)$. Correspondingly, a transmitted '1' appears as a '0' with probability λ_1 and as a '1' with probability $1 - \lambda_1$, where $\lambda_1 \in (0, 1)$. Let hypothesis H_0 stand for the transmission of a bit '0', and hypothesis H_1 for the transmission of a bit '1'.



(a) Formulate the likelihood ratio, and derive Bayes rule assuming that '0' and '1' are equally likely at the channel input (i.e., equal priors) and uniform cost assignment. What is the minimal risk in the case of a binary *symmetric* channel where $\lambda_0 = \lambda_1 = \lambda$?

(b) Construct a (randomized?) minimax test for the binary symmetric channel without information about the prior probability π_0 of a transmitted '0'. Plot the minimal attainable bit-error probability $V(\pi_0)$ versus π_0.

2. Suppose Y is a random variable that, under hypothesis H_0, has probability density

$$p(y\,|H_0) = \begin{cases} \frac{2}{3}(y+1) & \text{for } 0 \le y \le 1 \\ 0 & \text{otherwise} \end{cases}$$

and, under hypothesis H_1, has probability density

$$p(y\,|H_1) = \begin{cases} 1 & \text{for } 0 \le y \le 1 \\ 0 & \text{otherwise.} \end{cases}$$

(a) Find the Bayes rule and minimum Bayes risk for testing H_0 versus H_1 with uniform costs and equal priors.

(b) Find the minimax rule and minimax risks for uniform costs.

(c) Find the Neyman-Pearson rule and the corresponding detection probability for false-alarm probability $\alpha \in (0, 1)$.

Table of Contents

Supervised Learning

techniques:

- techniques developed within the fields of *pattern recognition* and *machine learning*
- suitable for problems where observations contain information about an unknown state or parameter vector in a rather involved fashion
- adequate decision rules or model parameters are to be learned from available *training data*

Classification problems:

- assignment of input vectors to a finite number of classes
- example: handwritten digit recognition

Regression problems:

- prediction of a continuous target vector from an observed input vector using a model of adjustable parameters to be learned

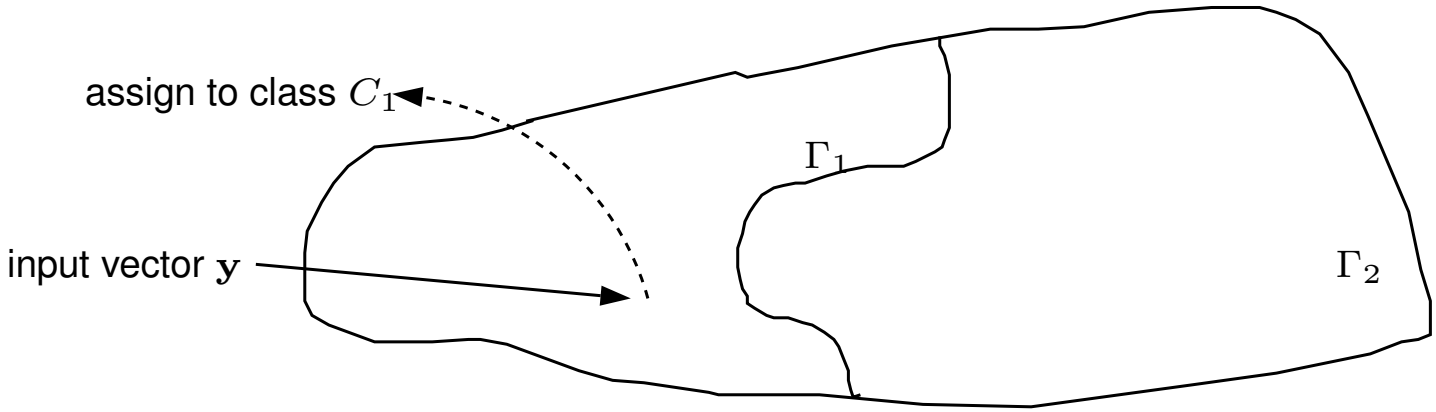
		Predicted		
		Cat	**Dog**	**Rabbit**
Actual	**Cat**	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

Example: Confusion matrix

Decision Regions for Classification

General definition of a classification rule:

- partition of input (i. e., observation) set Γ into decision regions $\Gamma_1, \ldots, \Gamma_M$ corresponding to classes $C_1, \ldots, C_M$



Specification of decision regions:

- through hyperplanes / linear discriminant functions
- through nonlinear discriminant functions
- induced by partitions in a feature space

Linear Discriminant Functions

Separation of two decision regions by a hyperplane:

- a *linear discriminant function* $f(\mathbf{y})$ in a K-dimensional space $\mathbb{R}^K$ has the form

$$f(\mathbf{y}) = \mathbf{w}^{\mathrm{T}}\mathbf{y} + b$$

  with $\mathbf{w} \in \mathbb{R}^K$ a weight vector and $b \in \mathbb{R}$ a bias term

- two decision regions Γ_1 and Γ_2 for a binary classification can be defined as

$$\Gamma_1 = \left\{\mathbf{y} \in \mathbb{R}^K \;:\; \mathbf{w}^{\mathrm{T}}\mathbf{y} + b \geq 0\right\} \quad \text{and} \quad \Gamma_2 = \left\{\mathbf{y} \in \mathbb{R}^K \;:\; \mathbf{w}^{\mathrm{T}}\mathbf{y} + b < 0\right\}$$

- the *decision boundary* $\{\mathbf{y} \in \mathbb{R}^K \;:\; \mathbf{w}^{\mathrm{T}}\mathbf{y} + b = 0\}$ is an (affine) *hyperplane*
- the vector $\mathbf{w}$ is normal to the hyperplane
- the distance of a point $\mathbf{y}$ from the hyperplane is given by

$$\frac{\left|\mathbf{w}^{\mathrm{T}}\mathbf{y} + b\right|}{||\mathbf{w}||}$$

Non-Binary Classification

Approaches for partitioning the input set in the scope of classification problems with M classes ($M > 2$):

1. use of $M-1$ parallel hyperplanes defined by a common linear discriminant function with $M-1$ distinct bias values
   - offering limited degree of freedom
2. *one-versus-the-rest* classifier:
   - initially, each class is trained to separate from one to the rest, and combine them by doing the multi-class classification according to the maximal output.

- M linear discriminant functions, defining the subsets $\Gamma_1, \ldots, \Gamma_M$ according to

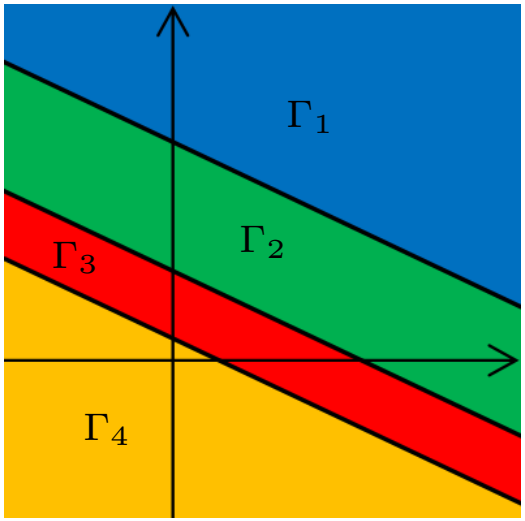
$$\Gamma_m = \left\{ \mathbf{y} \in \mathbb{R}^K : \begin{array}{ll} \mathbf{w}_m^{\mathrm{T}}\mathbf{y} + b_m > \mathbf{w}_\ell^{\mathrm{T}}\mathbf{y} + b_\ell & \text{for } \ell = 1, \ldots, m-1 \\ \text{and} & \\ \mathbf{w}_m^{\mathrm{T}}\mathbf{y} + b_m \geq \mathbf{w}_\ell^{\mathrm{T}}\mathbf{y} + b_\ell & \text{for } \ell = m+1, \ldots, M \end{array} \right\}$$

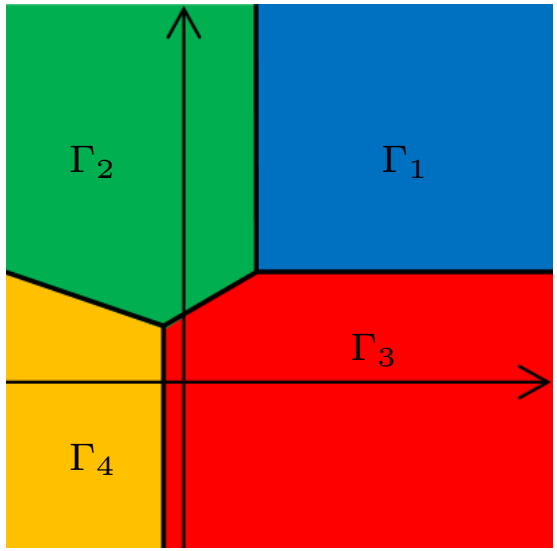
- binary classifiers are obtained by training on different binary classification problems, thus it is unclear whether their real-valued outputs are on comparable scales. i.e. several binary classifiers assign the pattern to their respective or *none*.

Non-Binary Classification (cont.)

3. *one-versus-one* classifier, where each of $M(M-1)/2$ discriminant functions focusses on two specific classes, combined with majority voting
   - may result in ambiguous decision regions

Examples: A classification (non-binary) into four classes





- left: partitioning of the input space by means of a common discriminant function $\mathbf{w}^{\mathrm{T}}\mathbf{y}$ along with three different bias values
- right: input space partitioning resulting from one-versus-the-rest classifier

Non-Binary Classification by the Least Squares Method

Casting a classification problem into a form suitable for the method of least squares:

- given a training set comprising L K-dimensional input vectors $\mathbf{y}_1, \ldots, \mathbf{y}_L$, each assigned to one out of M classes
- let the class to which input vector $\mathbf{y}_\ell$ belongs to be identified by the position of the "1" in the M-dimensional vector $\mathbf{t}_\ell = [0 \cdots 0\ 1\ 0 \cdots 0]^{\mathrm{T}}$
- let further

$$\mathbf{Q} = \begin{bmatrix} [\mathbf{y}_1^{\mathrm{T}}\ 1] \\ \vdots \\ [\mathbf{y}_L^{\mathrm{T}}\ 1] \end{bmatrix}, \quad \mathbf{W} = \left[\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} \cdots \begin{bmatrix} \mathbf{w}_M \\ b_M \end{bmatrix} \right], \quad \text{and } \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{t}_L^{\mathrm{T}} \end{bmatrix}$$

- find the coefficient matrix $\mathbf{W}$ for which the residual matrix $(\mathbf{QW} - \mathbf{T})$ becomes minimal in the least squares sense
- hence, with $\mathrm{tr}(\cdot)$ denoting the trace of a square matrix,

$$\widetilde{\mathbf{W}} = \arg\min_{\mathbf{W}} \mathrm{tr}\left((\mathbf{QW} - \mathbf{T})^{\mathrm{T}} (\mathbf{QW} - \mathbf{T}) \right)$$

- assuming that $L > K$ and $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ being non-singular: $\widetilde{\mathbf{W}} = \mathbf{Q}^{+}\mathbf{T}$ (see #147)

Classification by the Least Squares Method: Properties

Having computed $\widetilde{\mathbf{W}}$, a new input vector $\mathbf{y} \in \mathbb{R}^K$ can be classified on the basis of

$$\mathbf{t} = \widetilde{\mathbf{W}}^{\mathrm{T}} \begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix}$$

Properties of $\mathbf{t}$:

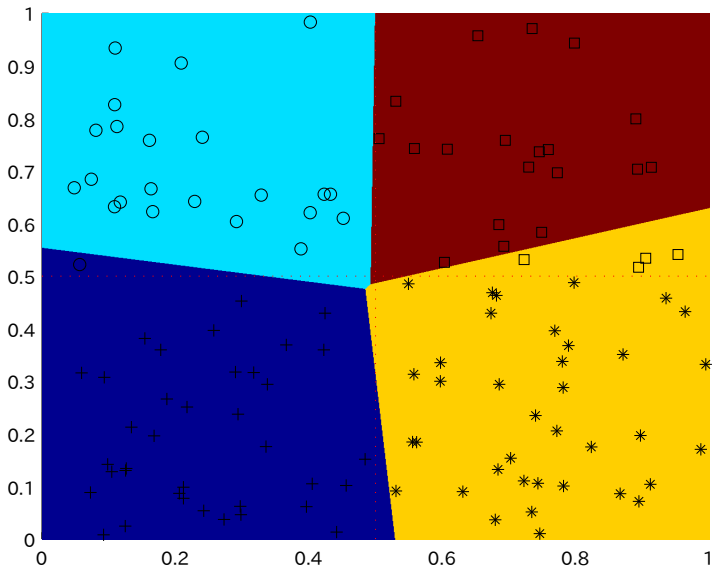
- generally contains multiple non-zero elements, some of which may be negative
- sum of the L elements equals 1

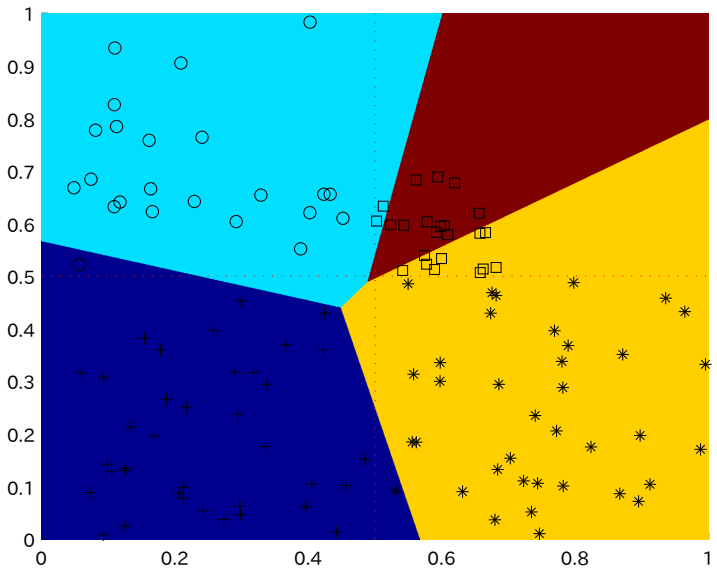
One-versus-the-rest type of classification:

- choose as class identifier the position of the largest element of $\mathbf{t}$
- resulting in decision boundaries expressed by affine functions

Classification by the Least Squares Method: Properties (cont.)

Examples of one-versus-the-rest type of classifications ($M = 4$) of points in $\Gamma = [0, 1] \times [0, 1]$ by the least squares method:





- left: training set and resulting partition with only a few misclassified samples
- right: training set and resulting partition with a number of misclassified samples

Binary Classification by the Perceptron Algorithm

Binary classification problem:

- given a training set with L samples $(\mathbf{y}_1, t_1), \ldots, (\mathbf{y}_L, t_L) \in \mathbb{R}^K \times \{-1, 1\}$
- separating hyperplane $\{\mathbf{y} \in \mathbb{R}^K : \mathbf{w}^{\mathrm{T}}\mathbf{y} = 0\}$ defined by K-dimensional vector $\mathbf{w}$
  (the role of the bias b can be given to some element of $\mathbf{w}$ whilst the corresponding elements of $\mathbf{y}_1, \ldots, \mathbf{y}_L$ are appointed to equal 1)
- find $\mathbf{w}$ such that, if possible,

$$t_\ell = f(\mathbf{y}_\ell, \mathbf{w}) = \begin{cases} +1 & \text{if } \mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell \geq 0 \\ -1 & \text{if } \mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell < 0 \end{cases} \qquad \text{for all } \ell = 1, \ldots, L \qquad (*)$$

A training set for which a hyperplane satisfying $(*)$ exists is called (linearly) *separable*.

Cost function:

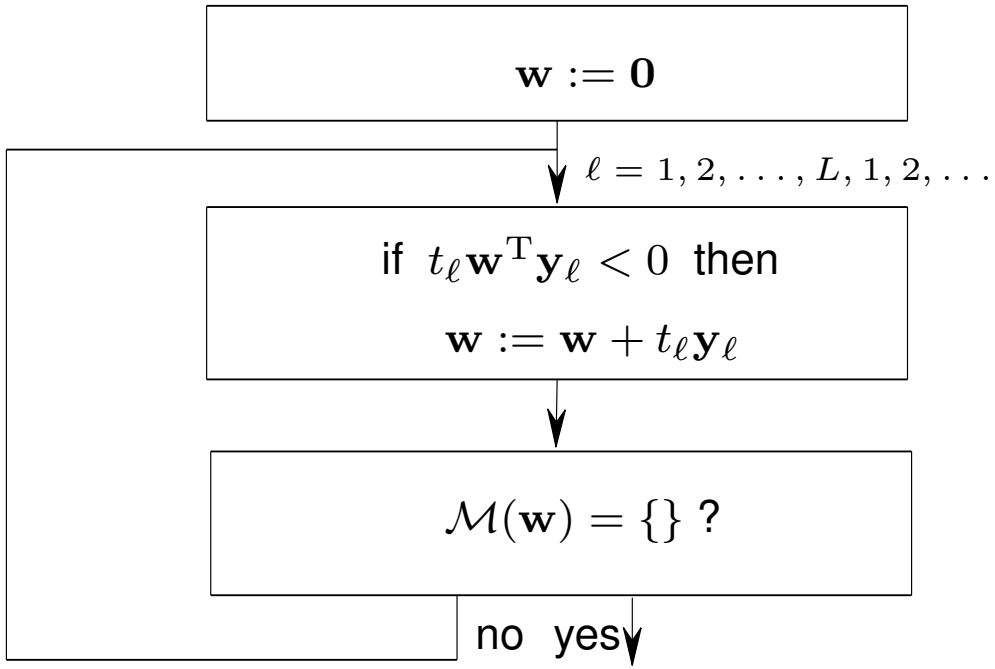
- *perceptron criterion* $\epsilon_{\mathrm{P}}(\mathbf{w}) = -\sum_{\ell \in \mathcal{M}(\mathbf{w})} t_\ell \mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell$
  with $\mathcal{M}(\mathbf{w}) = \{\ell \in \{1, \ldots, L\} : t_\ell \mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell < 0\}$ reflecting the misclassified points
- function $\epsilon_{\mathrm{P}}(\mathbf{w})$ is linear within the regions of $\mathbb{R}^K$ where $\mathcal{M}(\mathbf{w})$ is invariant

The Perceptron Algorithm

Gradient of $\epsilon_{\mathrm{P}}(\mathbf{w})$:

$$\nabla\epsilon_{\mathrm{P}}(\mathbf{w}) = -\sum_{\ell\in\mathcal{M}(\mathbf{w})} t_\ell \mathbf{y}_\ell$$

Iterative *perceptron algorithm*:



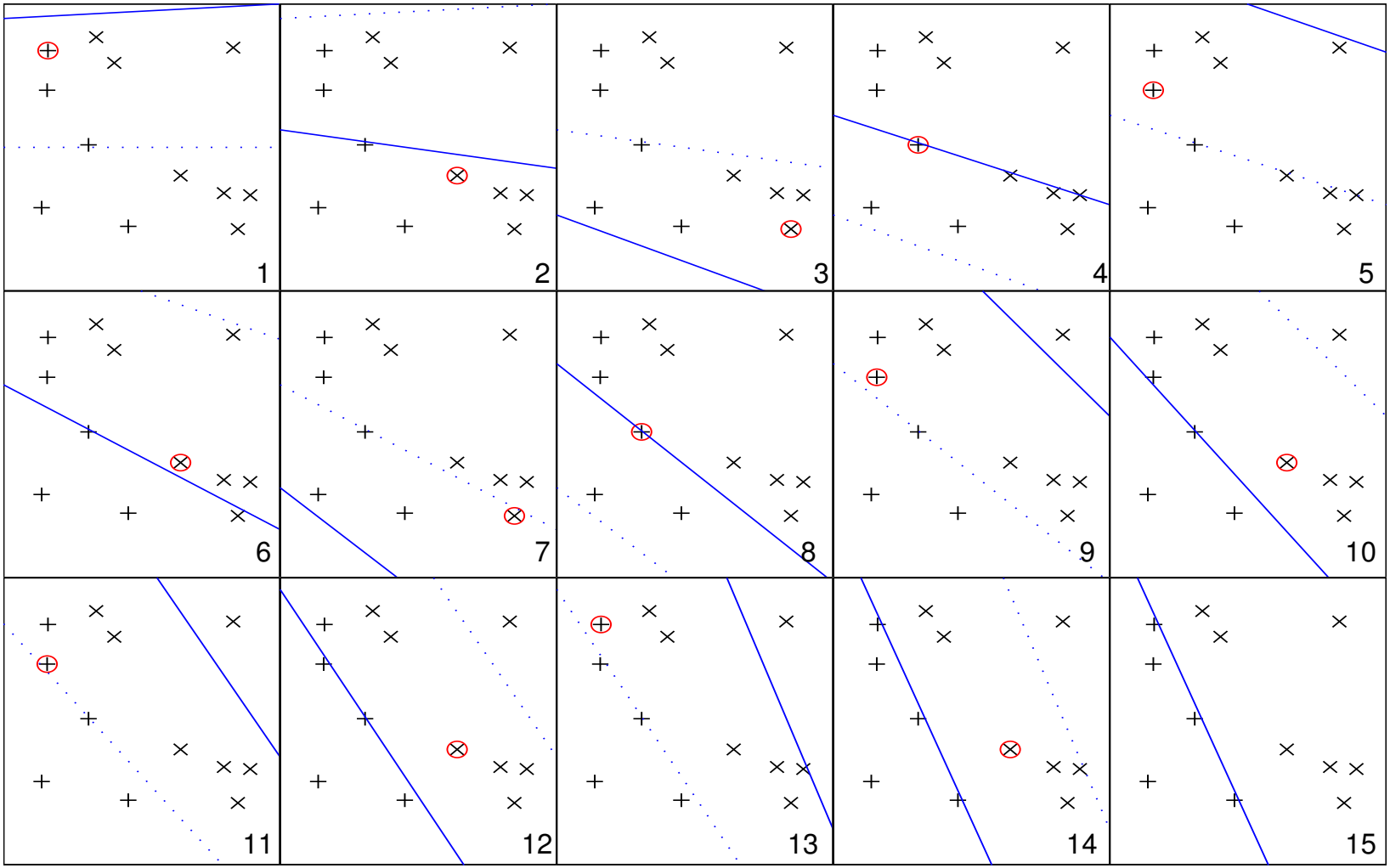
Frank Rosenblatt

(1928–1971)

Convergence:

- for separable training sets the perceptron algorithm ends after a finite number of iterations

Example: The Perceptron Algorithm



Support Vector Machines

Support vector machines are supervised learning methods which use hyperplanes for classification and regression tasks. An important property of support vector machines is that the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum.

Possible types of hyperplanes:

- maximum margin hyperplanes
- soft margin hyperplanes

Constituents of typical support vector machines:

- use of hyperplanes in high or infinite dimensional feature space
- nonlinear map from input space to feature space
- representation of the hyperplanes by using only a subset of the samples of the training set, called the *support vectors*

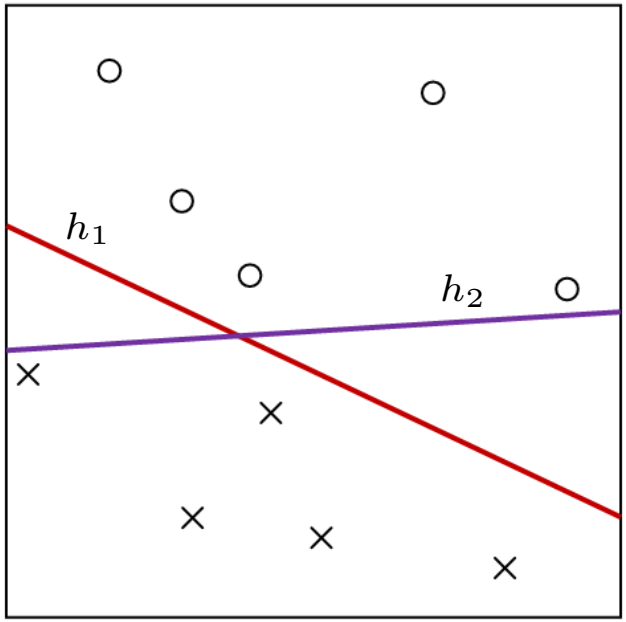
Towards an Optimal Hyperplane

Given the training set is linearly separable,

- the least squares method comes with no guarantee to yield a valid hyperplane (i.e., one which perfectly classifies the training points)
- the perceptron algorithm ends with an arbitrary valid hyperplane

Can we find a hyperplane which separates the points in the training set in some optimal way?

Which of the two below hyperplanes is "better", h_1 or h_2?



The Margin

Given a hyperplane defined by $\mathbf{w}$ and b,

- the *margin* of a point $\mathbf{y}$ from class $t \in \{-1, 1\}$ is given by

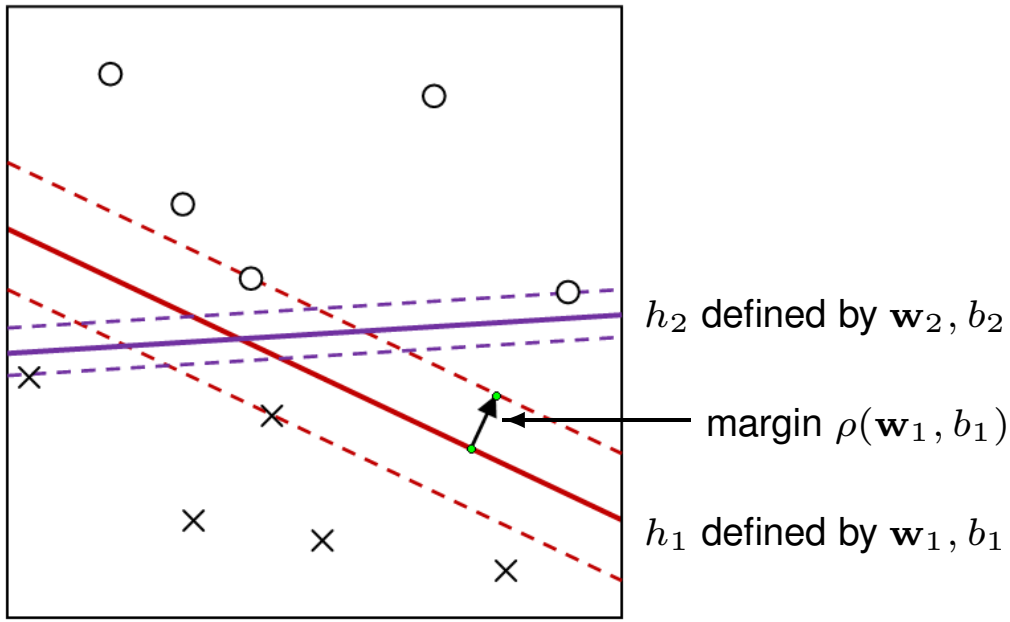
$$\frac{t_\ell \left(\mathbf{w}^{\mathrm{T}} \mathbf{y}_\ell + b\right)}{||\mathbf{w}||}$$

- the absolute value of the margin reflects the distance of the point from the hyperplane
- misclassified points have negative margins
- the margin of a training set $\{(\mathbf{y}_1, t_1), \ldots, (\mathbf{y}_L, t_L)\}$ is defined as

$$\rho(\mathbf{w}, b) = \min_{\ell=1,\ldots,L} \frac{t_\ell \left(\mathbf{w}^{\mathrm{T}} \mathbf{y}_\ell + b\right)}{||\mathbf{w}||}$$

Maximum Margin Hyperplane

The margin of a separating hyperplane, and thus the length of the weight vector $\mathbf{w}$, plays a fundamental role in support vector type algorithms (large margin leads to correct classification of *all* test points).



Problem of finding $(\mathbf{w}, b) \in \mathbb{R}^K \times \mathbb{R}$ defining the hyperplane with maximal margin:

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \quad \text{subject to} \quad t_\ell \left(\mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell + b\right) \geq 1, \quad \ell = 1, \ldots, L \qquad (\mathrm{P})$$

For a (strictly) separable training set a solution $\left(\widetilde{\mathbf{w}}, \tilde{b}\right)$ of (P) exists with $\rho\left(\widetilde{\mathbf{w}}, \tilde{b}\right) = 1/\left|\left|\widetilde{\mathbf{w}}\right|\right|$.

Maximum Margin Hyperplane: The Lagrange Dual Function

Solution of the *primal problem* (P) via the Lagrange dual function:

- the *Lagrangian L* has to be minimized with respect to the *primal variables* $\mathbf{w}, b$ and maximized with respect to the *dual variables* $\boldsymbol{\lambda}$
- the *Lagrangian*, which includes the inequality constraints, reads

$$L(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \sum_{\ell=1}^{L} \lambda_\ell \left(1 - t_\ell \left(\mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell + b\right)\right)$$

  with the factors in $\boldsymbol{\lambda} = [\lambda_1 \cdots, \lambda_L]^{\mathrm{T}}$ referred to as the *Lagrange multipliers*

- the *Lagrange dual function* reads

$$g(\boldsymbol{\lambda}) = \inf_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\lambda}), \qquad \boldsymbol{\lambda} \succeq \mathbf{0}$$

  with $\boldsymbol{\lambda} \succeq \mathbf{0}$ standing for $\lambda_\ell \geq 0$ for all $\ell = 1, \ldots, L$

Maximum Margin Hyperplane: Properties of the Lagrange Dual Function

Properties of the Lagrange dual Function:

- for any $\boldsymbol{\lambda} \succeq \mathbf{0}$

$$g(\boldsymbol{\lambda}) \leq \frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}}$$

  (lower bounding property of $g(\boldsymbol{\lambda})$)

- the function $g(\boldsymbol{\lambda})$ is concave
- $g(\boldsymbol{\lambda}) = -\infty$ for $\boldsymbol{\lambda}$ for which $L(\mathbf{w}, b, \boldsymbol{\lambda})$ is unbounded below
- conditions for $L(\mathbf{w}, b, \boldsymbol{\lambda})$ to attain a minimum over $(\mathbf{w}, b)$:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\lambda})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{\ell=1}^{L} \lambda_\ell t_\ell \mathbf{y}_\ell = \mathbf{0}, \qquad \frac{\partial L(\mathbf{w}, b, \boldsymbol{\lambda})}{\partial b} = -\sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0$$

- consequently,

$$g(\boldsymbol{\lambda}) = \begin{cases} \sum_{\ell=1}^{L} \lambda_\ell - \frac{1}{2} \sum_{\ell=1}^{L} \sum_{k=1}^{L} \lambda_\ell \lambda_k t_\ell t_k \mathbf{y}_\ell^{\mathrm{T}} \mathbf{y}_k & \text{for } \boldsymbol{\lambda} \text{ satisfying } \sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0 \\ -\infty & \text{otherwise} \end{cases}$$

Maximum Margin Hyperplane: The Lagrange Dual Problem

For the closest possible lower bound to $\frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}}$ we consider the *Lagrange dual problem*

$$\text{maximize} \quad g(\boldsymbol{\lambda}) \quad \text{subject to} \quad \boldsymbol{\lambda} \succeq \mathbf{0} \qquad (\mathrm{D})$$

The difference between

- the optimal value $\frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}}$ of the primal problem (P) and
- the optimal value $g\left(\widetilde{\boldsymbol{\lambda}}\right)$ of the dual problem (D)

is known as *duality gap*.

Strong duality:

- referring to the condition under which the duality gap equals zero
- (P) qualifies for strong duality because, first, the objective function $\mathbf{w}^{\mathrm{T}}\mathbf{w}/2$ is convex, and second, the constraints satisfy *Slater's condition*
- hence,

$$\frac{1}{2}\widetilde{\mathbf{w}^{\mathrm{T}}}\widetilde{\mathbf{w}} = g\left(\widetilde{\boldsymbol{\lambda}}\right)$$

Maximum Margin Hyperplane: Complementary Slackness and Support Vectors

We note that

$$\frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}} = g\left(\widetilde{\boldsymbol{\lambda}}\right) = \inf_{\mathbf{w},b} L\left(\mathbf{w}, b, \widetilde{\boldsymbol{\lambda}}\right) \leq \frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}} + \sum_{\ell=1}^{L} \underbrace{\widetilde{\lambda}_\ell}_{\geq 0} \underbrace{\left(1 - t_\ell\left(\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{y}_\ell + \tilde{b}\right)\right)}_{\leq 0}$$

strong duality

$\mathbf{w} := \widetilde{\mathbf{w}}$, $b := \tilde{b}$

Consequently,

$$\widetilde{\lambda}_\ell\left(1 - t_\ell\left(\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{y}_\ell + \tilde{b}\right)\right) = 0, \qquad \ell = 1, \ldots, L \qquad \text{(a)}$$

referred to as *complementary slackness*

Indices of the *support vectors*:

$$\mathcal{I}_{\mathrm{SV}} = \left\{\ell \in \{1, \ldots, L\} \ : \ \widetilde{\lambda}_\ell > 0\right\}$$

- according to (a), *support vectors* lie exactly on the margin
- remaining examples in the training set are irrelevant: their constraints are satisfied automatically and do not appear in the expansion, since their multipliers satisfy $\widetilde{\lambda}_\ell = 0$

Maximum Margin Hyperplane: Computation

Procedure for computing maximum margin hyperplane:

1. solve the dual problem

   maximize $\displaystyle\sum_{\ell=1}^{L} \lambda_\ell - \frac{1}{2}\sum_{\ell=1}^{L}\sum_{k=1}^{L} \lambda_\ell \lambda_k t_\ell t_k \mathbf{y}_\ell^{\mathrm{T}} \mathbf{y}_k$ subject to $\begin{cases} \boldsymbol{\lambda} \succeq \mathbf{0} \\ \displaystyle\sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0 \end{cases}$

   by means of *convex optimization* methods

2. compute $\widetilde{\mathbf{w}}$ according to

$$\widetilde{\mathbf{w}} = \sum_{\ell=1}^{L} \widetilde{\lambda}_\ell t_\ell \mathbf{y}_\ell = \sum_{\ell \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_\ell t_\ell \mathbf{y}_\ell$$

3. compute $\tilde{b}$ according to

$$\tilde{b} = t_\ell - \widetilde{\mathbf{w}}^{\mathrm{T}} \mathbf{y}_\ell = t_\ell - \sum_{k \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_k t_k \mathbf{y}_k^{\mathrm{T}} \mathbf{y}_\ell \quad \text{for any } \ell \in \mathcal{I}_{\mathrm{SV}}$$

Convex Optimization

- Given the set of convex functions $f, c_1, \ldots, c_n$ on the convex set $\mathfrak{X}$, the problem

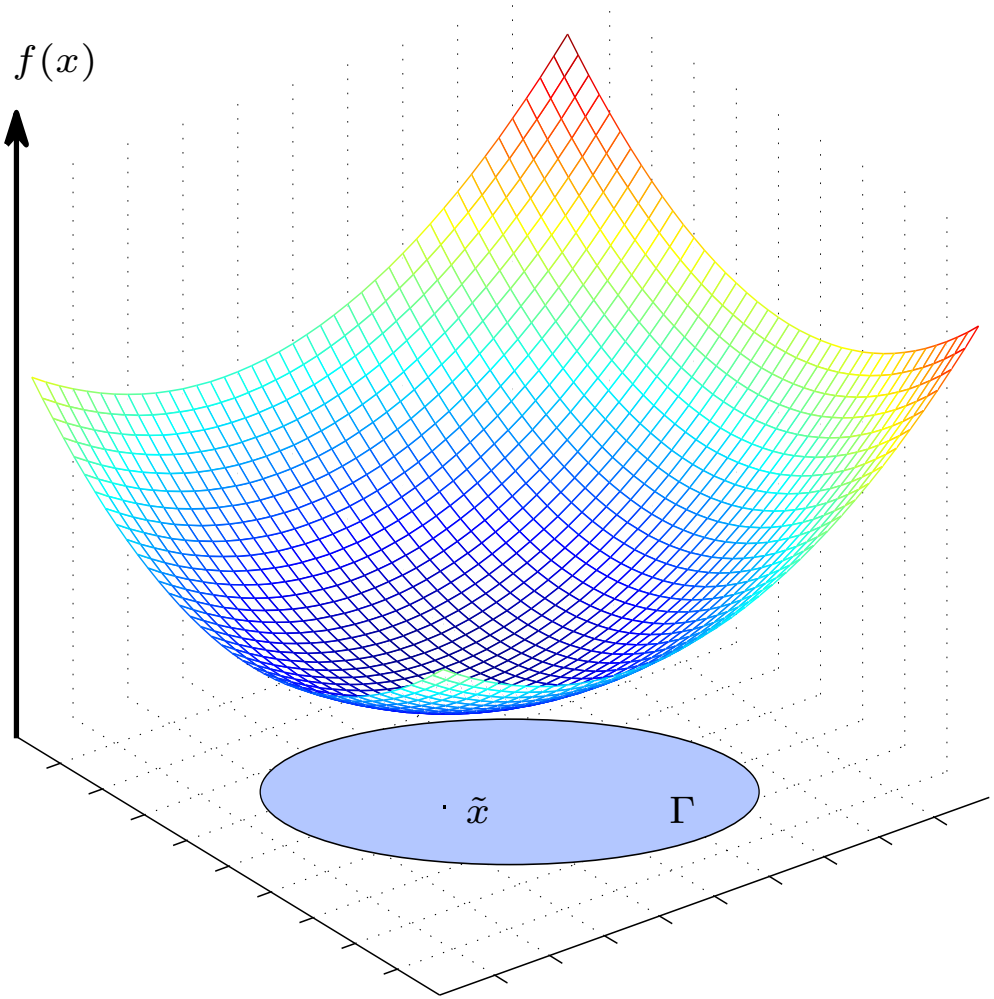
$$\min_{x : c_i(x) \leq 0 \ \forall i \ \in [n]} f(x)$$

  has as its solution a convex set, if a solution exists. This solution is unique if f is strictly convex. Many problems in Mathematical Programming or Support Vector Machines can be cast into this formulation. This means either they all have unique solutions (if f is strictly convex), or all solutions are equally good and form a convex set (of f is merely convex).

- The problem of convex minimization on a convex set is typically a hard problem, in the sense that the maximum can only be found at one of the extreme points of the constraining set.

- Learning (statistical estimation) implies the minimization of risk function. Minimizing an arbitrary function on a set of arguments is a difficult task, and will most likely exhibit many local minima. Minimization of a convex objective function on a convex set exhibits exactly one global minima.

Convex Optimization (cont.)



Problem:

- given a *convex function*

$$f \ : \ \Gamma \rightarrow \mathbb{R}$$

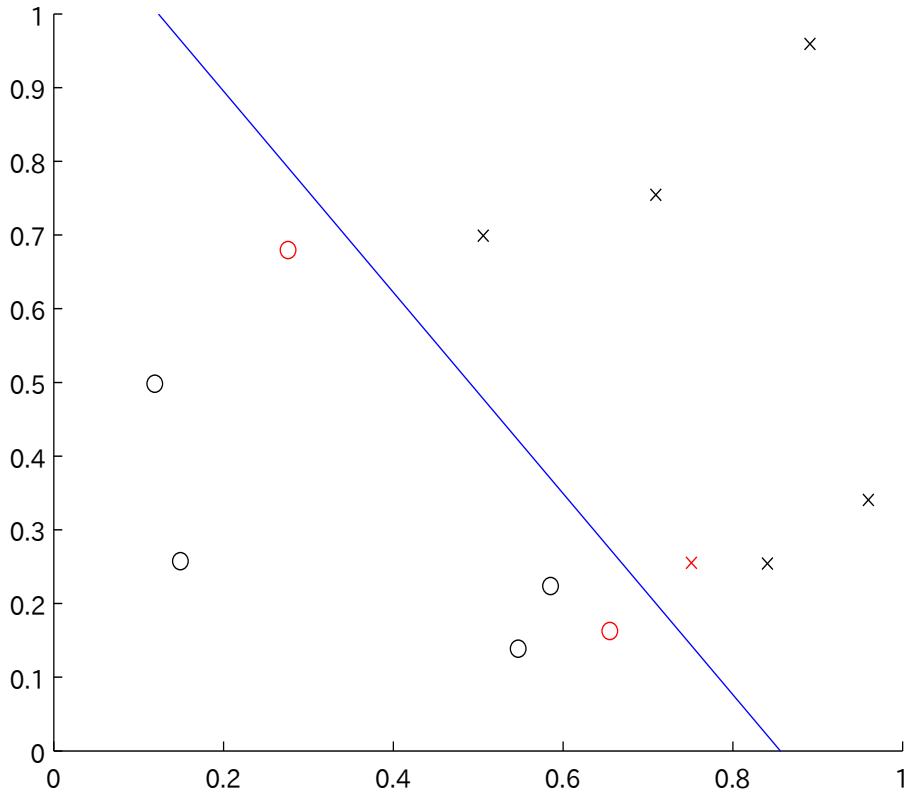
- defined on a *convex subset* Γ of a vector space $\mathbb{R}^K$
- find a point $\tilde{x} \in \Gamma$ at which $f(x)$ achieves a minimum

Properties:

- every *local* minimum is a *global* minimum
- efficient numerical methods are known for solving many convex optimization problems like *Newton's Methods* and *Interior Point Methods*

Example: Maximum Margin Hyperplane

Example of a maximum margin hyperplane in a two-dimensional input space:



- $\Gamma = [0,1] \times [0,1]$
- points of training set are plotted as "x" (class C_1) or "o" (class C_2)
- $\widetilde{\mathbf{w}} = \begin{bmatrix} 12.19 \\ 8.94 \end{bmatrix}$
- $\tilde{b} = -10.44$
- support vectors are plotted in red

Non-Separable Training Sets

If the training set is linearly non-separable, the afore formulated optimization problem has no solution.

How can we modify the problem statement to be applicable to non-separable training sets?

Possible approach to allow some points to violate the margin:

- introduction of *slack variables* $\xi_1, \ldots, \xi_L$, one variable per point
- allowing the possibility of examples violating $t_\ell \left(\mathbf{w}^{\mathrm{T}} \mathbf{y}_\ell + b\right) \geq 1, \quad \ell = 1, \ldots, L$, and use relaxed separation constraints is the reason they are introduced
- $\xi_\ell = 0$ indicates that ℓth point maintains the margin
- positive variable ξ_ℓ reflects the value by which the ℓth point violates the margin
- $C \frac{1}{L} \sum_{\ell=1}^{L} \xi_\ell$, with C a certain constant (determines the trade-off between minimizing the training error, and maximizing the margin), serves as penalty in the primal problem

$$\text{minimize} \quad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \frac{C}{L}\sum_{\ell=1}^{L}\xi_\ell \quad \text{subject to} \quad \begin{cases} t_\ell \left(\mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell + b\right) \geq 1 - \xi_\ell, & \ell = 1, \ldots, L \\ \xi_\ell \geq 0, & \ell = 1, \ldots, L \end{cases} \qquad \text{(P)}$$

- result known as *soft margin hyperplane*

Soft Margin Hyperplane: The Lagrange Dual Function

Solution of the primal problem (P) via the Lagrange dual function:

- the Lagrangian reads

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \frac{C}{L}\sum_{\ell=1}^{L}\xi_\ell + \sum_{\ell=1}^{L}\lambda_\ell\left(1 - t_\ell\left(\mathbf{w}^{\mathrm{T}}\mathbf{y}_\ell + b\right) - \xi_\ell\right) - \sum_{\ell=1}^{L}\mu_\ell\xi_\ell$$

  with $\boldsymbol{\xi} = [\xi_1 \cdots \xi_L]^{\mathrm{T}}$ and the Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_L]^{\mathrm{T}}$, $\boldsymbol{\mu} = [\mu_1 \cdots \mu_L]^{\mathrm{T}}$

- the Lagrange dual function reads

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\mathbf{w}, b, \boldsymbol{\xi}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}), \qquad \begin{matrix}\boldsymbol{\lambda} \succeq \mathbf{0} \\ \boldsymbol{\mu} \succeq \mathbf{0}\end{matrix}$$

Soft Margin Hyperplane: Reformulation of the Lagrange Dual Function

Conditions for $L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ to attain a minimum over $(\mathbf{w}, b, \boldsymbol{\xi})$:

- $$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{\ell=1}^{L} \lambda_\ell t_\ell \mathbf{y}_\ell = \mathbf{0}$$
- $$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})}{\partial b} = -\sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0$$
- $$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})}{\partial \boldsymbol{\xi}} = \frac{C}{L}\mathbf{1} - \boldsymbol{\lambda} - \boldsymbol{\mu} = \mathbf{0}$$

It follows that

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \begin{cases} \sum_{\ell=1}^{L} \lambda_\ell - \frac{1}{2} \sum_{\ell=1}^{L} \sum_{k=1}^{L} \lambda_\ell \lambda_k t_\ell t_k \mathbf{y}_\ell^{\mathrm{T}} \mathbf{y}_k & \text{for } \boldsymbol{\lambda} \text{ satisfying } \sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0 \text{ and } \boldsymbol{\lambda} + \boldsymbol{\mu} = \frac{C}{L}\mathbf{1} \\ -\infty & \text{otherwise} \end{cases}$$

Soft Margin Hyperplane: The Lagrange Dual Problem

Lagrange dual problem:

$$\text{maximize} \quad \sum_{\ell=1}^{L} \lambda_\ell - \frac{1}{2} \sum_{\ell=1}^{L} \sum_{k=1}^{L} \lambda_\ell \lambda_k t_\ell t_k \mathbf{y}_\ell^{\mathrm{T}} \mathbf{y}_k \quad \text{subject to} \quad \begin{cases} \frac{C}{L}\mathbf{1} \succeq \boldsymbol{\lambda} \succeq \mathbf{0} \\ \sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0 \end{cases} \tag{D}$$

Strong duality:

- again, (P) qualifies for strong duality
- with $\left(\widetilde{\mathbf{w}}, \tilde{b}, \widetilde{\boldsymbol{\xi}}\right)$ the solution of (P), $\widetilde{\boldsymbol{\lambda}}$ the solution of (D) and $\widetilde{\boldsymbol{\mu}} = \frac{C}{L}\mathbf{1} - \widetilde{\boldsymbol{\lambda}}$, we have

$$\frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}} + \frac{C}{L}\sum_{\ell=1}^{L}\widetilde{\xi}_\ell = g\left(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}}\right)$$

Soft Margin Hyperplane: Complementary Slackness and Support Vectors

We note that

$$\frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}} + \frac{C}{L}\sum_{\ell=1}^{L}\widetilde{\xi}_\ell = g\left(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}}\right) = \inf_{\mathbf{w},b,\boldsymbol{\xi}} L\left(\mathbf{w}, b, \boldsymbol{\xi}, \widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}}\right)$$

$$\leq \frac{1}{2}\widetilde{\mathbf{w}}^{\mathrm{T}}\widetilde{\mathbf{w}} + \frac{C}{L}\sum_{\ell=1}^{L}\widetilde{\xi}_\ell + \sum_{\ell=1}^{L}\underbrace{\widetilde{\lambda}_\ell}_{\geq 0}\underbrace{\left(1 - t_\ell\left(\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{y}_\ell + \tilde{b}\right) - \widetilde{\xi}_\ell\right)}_{\leq 0} - \sum_{\ell=1}^{L}\underbrace{\widetilde{\mu}_\ell}_{\geq 0}\widetilde{\xi}_\ell$$

Consequently, for $\ell = 1, \ldots, L$,

- $\widetilde{\lambda}_\ell\left(1 - t_\ell\left(\widetilde{\mathbf{w}}^{\mathrm{T}}\mathbf{y}_\ell + \tilde{b}\right) - \widetilde{\xi}_\ell\right) = 0$
- $\widetilde{\mu}_\ell\widetilde{\xi}_\ell = 0$
- if $\widetilde{\lambda}_\ell < \frac{C}{L}$ then $\widetilde{\xi}_\ell = 0$

Indices of the *support vectors*:

$$\mathcal{I}_{\mathrm{SV}} = \left\{\ell \in \{1, \ldots, L\} \ : \ \widetilde{\lambda}_\ell > 0\right\}$$

Soft Margin Hyperplane: Computation

Procedure for computing soft margin hyperplane:

1. solve the dual problem

   maximize $\sum_{\ell=1}^{L} \lambda_\ell - \frac{1}{2} \sum_{\ell=1}^{L} \sum_{k=1}^{L} \lambda_\ell \lambda_k t_\ell t_k \mathbf{y}_\ell^{\mathrm{T}} \mathbf{y}_k$ subject to $\begin{cases} \frac{C}{L}\mathbf{1} \succeq \boldsymbol{\lambda} \succeq \mathbf{0} \\ \sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0 \end{cases}$

   by means of *convex optimization* methods

2. compute $\widetilde{\mathbf{w}}$ according to

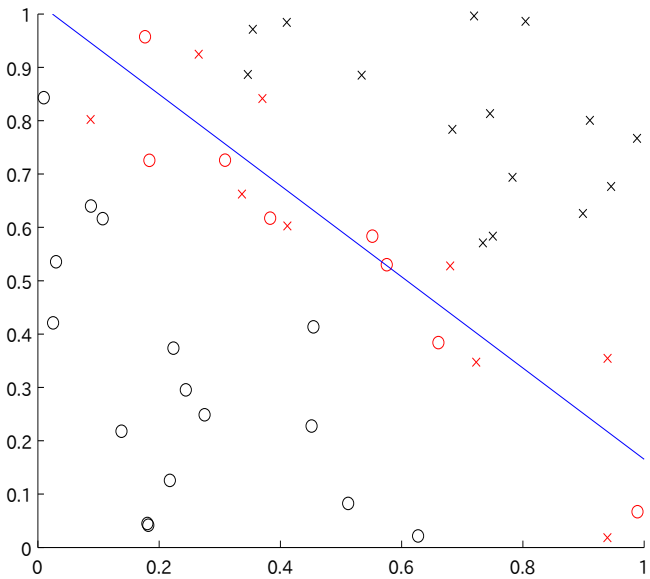
$$\widetilde{\mathbf{w}} = \sum_{\ell=1}^{L} \widetilde{\lambda}_\ell t_\ell \mathbf{y}_\ell = \sum_{\ell \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_\ell t_\ell \mathbf{y}_\ell$$

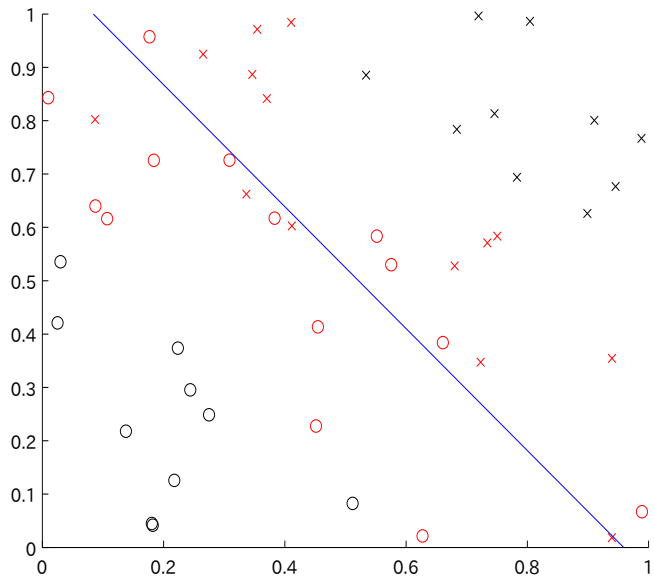
3. compute $\tilde{b}$ according to

$$\tilde{b} = t_\ell - \widetilde{\mathbf{w}}^{\mathrm{T}} \mathbf{y}_\ell = t_\ell - \sum_{k \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_k t_k \mathbf{y}_k^{\mathrm{T}} \mathbf{y}_\ell \quad \text{for any } \ell \text{ for which } 0 < \widetilde{\lambda}_\ell < \frac{C}{L}$$

Examples: Soft Margin Hyperplane

Examples of soft margin hyperplanes in a two-dimensional input space:





- left: training points and resulting hyperplane for $C = 10000$
- right: training points and resulting hyperplane for $C = 100$

Nonlinear Feature Map

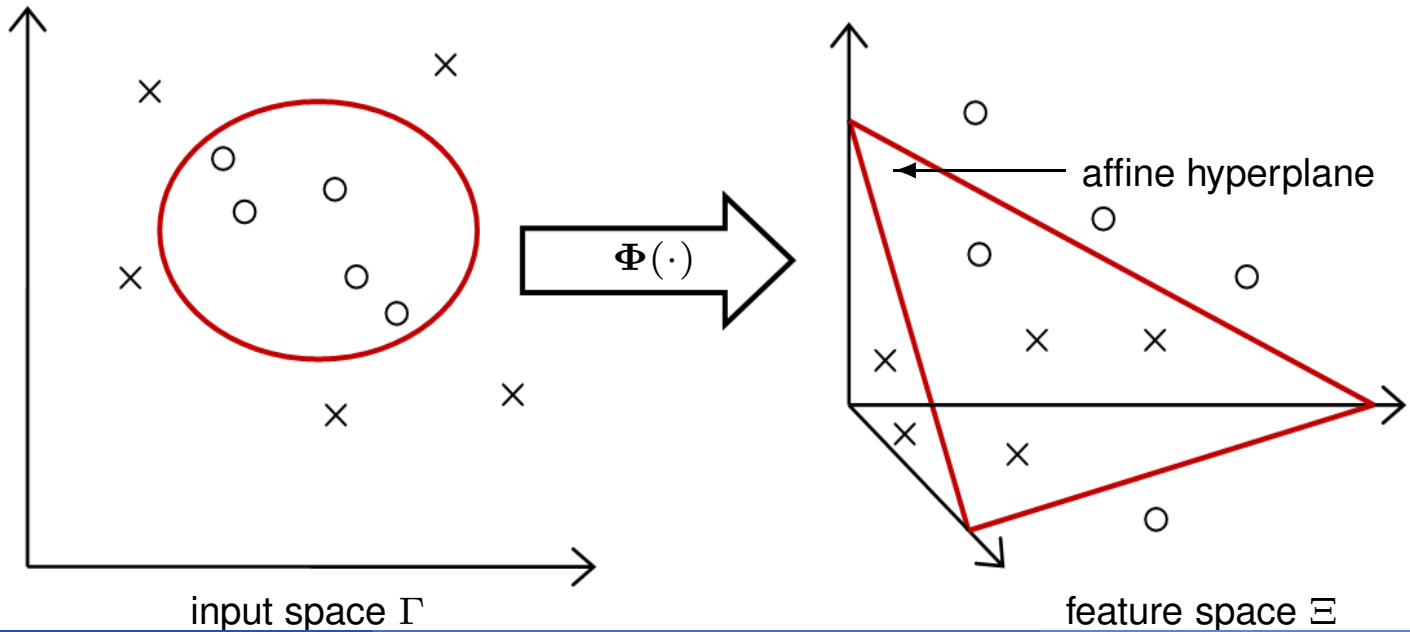
Hyperplanes may be an inappropriate means for the partitioning of the input space.
Remedy: substitution of the input vector $\mathbf{y}$ in the linear discriminant function $\mathbf{w}^{\mathrm{T}}\mathbf{y} + b$

by

$$\mathbf{\Phi}(\mathbf{y}) = \begin{bmatrix} \phi_1(\mathbf{y}) \\ \vdots \\ \phi_K(\mathbf{y}) \end{bmatrix}$$

, where $\phi_1(\cdot), \ldots, \phi_K(\cdot)$ are suitable nonlinear real-valued functions.

Nonlinear *feature map* from the input space to the *feature space*:



Classification by the Least Squares Method Revisited

Incorporating a feature map into the classification procedure based on the least squares method is straightforward:

- replace matrix $\mathbf{Q} = \begin{bmatrix} (\mathbf{y}_1^{\mathrm{T}}\ 1) \\ \vdots \\ (\mathbf{y}_L^{\mathrm{T}}\ 1) \end{bmatrix}$ by

  $$\mathbf{Q} = \begin{bmatrix} \phi_1(\mathbf{y}_1) & \phi_2(\mathbf{y}_1) & \cdots & \phi_K(\mathbf{y}_1) \\ \phi_1(\mathbf{y}_2) & \phi_2(\mathbf{y}_2) & \cdots & \phi_K(\mathbf{y}_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(\mathbf{y}_L) & \phi_2(\mathbf{y}_L) & \cdots & \phi_K(\mathbf{y}_L) \end{bmatrix}$$ with suitable $\phi_1(\cdot), \ldots, \phi_{K-1}(\cdot)$ and $\phi_K(\cdot) = 1$

- assuming that $L > K$ and $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ being non-singular: $\widetilde{\mathbf{W}} = \mathbf{Q}^{+}\mathbf{T}$
- a new input vector $\mathbf{y}$ can be classified on the basis of

  $$\mathbf{t} = \widetilde{\mathbf{W}}^{\mathrm{T}} \begin{bmatrix} \phi_1(\mathbf{y}) \\ \vdots \\ \phi_K(\mathbf{y}) \end{bmatrix}$$

- *one-versus-the-rest* type of classification results in more general decision boundaries in input space not representable by affine functions

Classification by Soft Margin Hyperplane in Feature Space

Binary classification by soft margin hyperplane in feature space:

1. solve the dual problem

$$\text{maximize} \quad \sum_{\ell=1}^{L} \lambda_\ell - \frac{1}{2} \sum_{\ell=1}^{L} \sum_{k=1}^{L} \lambda_\ell \lambda_k t_\ell t_k \underbrace{\boldsymbol{\Phi}(\mathbf{y}_\ell)^{\mathrm{T}} \boldsymbol{\Phi}(\mathbf{y}_k)}_{=k(\mathbf{y}_\ell, \mathbf{y}_k)} \quad \text{subject to} \quad \begin{cases} \frac{C}{L}\mathbf{1} \succeq \boldsymbol{\lambda} \succeq \mathbf{0} \\ \sum_{\ell=1}^{L} \lambda_\ell t_\ell = 0 \end{cases}$$

2. compute $\tilde{b}$ according to

$$\tilde{b} = t_\ell - \sum_{k \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_k t_k \underbrace{\boldsymbol{\Phi}(\mathbf{y}_k)^{\mathrm{T}} \boldsymbol{\Phi}(\mathbf{y}_\ell)}_{=k(\mathbf{y}_\ell, \mathbf{y}_k)} \quad \text{for any } \ell \text{ for which } 0 < \widetilde{\lambda}_\ell < \frac{C}{L}$$

3. classify new input vector $\mathbf{y}$ according to

$$t = \operatorname{sgn}\left(\sum_{\ell \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_\ell t_\ell \underbrace{\boldsymbol{\Phi}(\mathbf{y}_\ell)^{\mathrm{T}} \boldsymbol{\Phi}(\mathbf{y})}_{=k(\mathbf{y}_\ell, \mathbf{y})} + \tilde{b} \right)$$

Note: all steps can be accomplished based on the *kernel function* $k(\mathbf{y}, \mathbf{y}') = \boldsymbol{\Phi}(\mathbf{y})^{\mathrm{T}} \boldsymbol{\Phi}(\mathbf{y}')$

The Gram Matrix

The *Gram matrix* w. r. t. $\mathbf{\Phi}(\mathbf{y}_1), \ldots, \mathbf{\Phi}(\mathbf{y}_L)$ is an $L \times L$ matrix, the mth element of the ℓth row of which is given by

$$k(\mathbf{y}_\ell, \mathbf{y}_m) = \langle \mathbf{\Phi}(\mathbf{y}_\ell), \mathbf{\Phi}(\mathbf{y}_m) \rangle \qquad (*)$$

- in linear algebra Gram matrices are sometimes used to verify the linear independence of a set of vectors in an inner product space
- by definition Gram matrices are symmetric and nonnegative definite

A kernel function $k : \Gamma \times \Gamma \to \mathbb{R}$ is called

- *symmetric* if $k(\mathbf{y}, \mathbf{y}') = k(\mathbf{y}', \mathbf{y})$ for all $\mathbf{y}, \mathbf{y}' \in \Gamma$
- *nonnegative definite* if for all $M \in \mathbb{N}$ and $\mathbf{y}_1, \ldots, \mathbf{y}_M \in \Gamma$ the $M \times M$ matrix with the mth element of the ℓth row given by $k(\mathbf{y}_\ell, \mathbf{y}_m)$ is nonnegative definite

Any kernel function defined as an inner product in a feature space according to $(*)$ is symmetric nonnegative definite. How about an alternative kernel function not defined as an inner product in a feature space?

Mercer's Theorem

Suppose the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric, continuous, and nonnegative definite, and $\mathcal{X}$ a compact subset[1)] of $\mathbb{R}^K$. Then there exists an orthonormal basis $\psi_1, \psi_2, \cdots \in L^2(\mathcal{X})$ [2)] along with nonnegative eigenvalues $\lambda_1, \lambda_2, \ldots$ such that

$$k(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{\infty} \lambda_j \psi_j(\mathbf{x}) \psi_j(\mathbf{y}) \qquad \text{for all } \mathbf{x}, \mathbf{y} \in \mathcal{X}$$

Corollary of Mercer's theorem:
Under the above assumptions, $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{y}) \rangle$ where

$$\begin{aligned} \mathbf{\Phi} \; : \; \mathcal{X} &\to \ell^2 \text{ [3)]} \\ \mathbf{x} &\mapsto \left(\sqrt{\lambda_j} \psi_j(\mathbf{x}) \right)_{j=1,2,\ldots} \end{aligned}$$

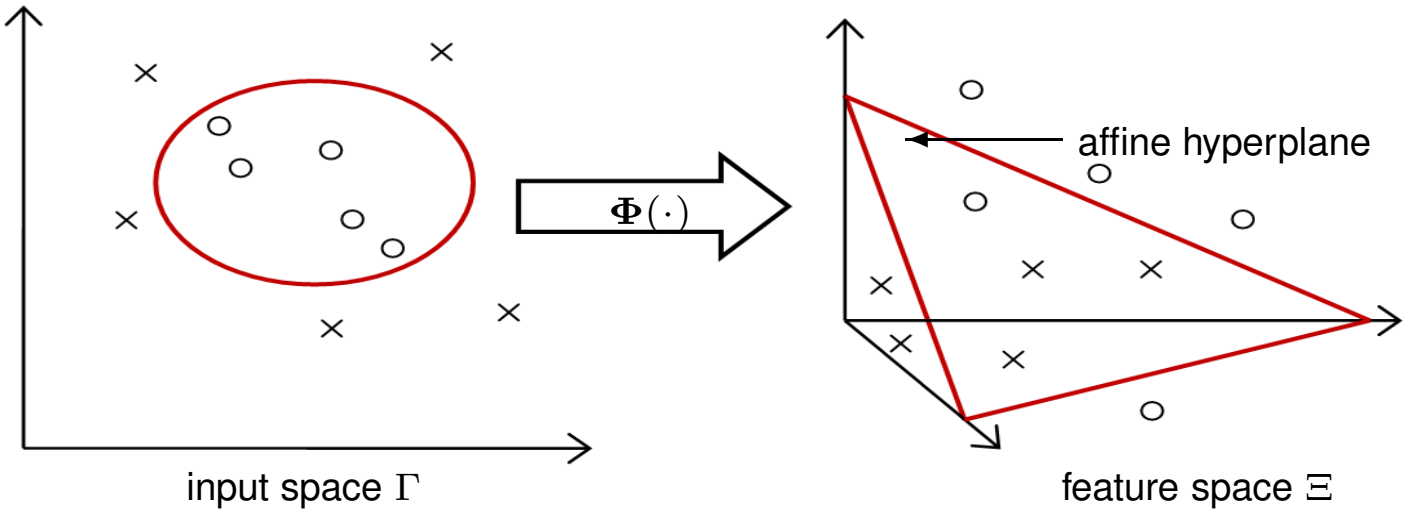
---

1) an example of a compact subset of $\mathbb{R}^K$ would be $[a, b]^K$ with $a < b$
2) $L^2(\mathcal{X})$ represents the vector space of the square integrable functions defined on $\mathcal{X}$
3) ℓ^2 denotes the Hilbert space of the square summable real-valued sequences $x = (x_j)_{j=1,2,\ldots}$ endowed with the inner product $\langle x, y \rangle = \sum_{j=1}^{\infty} x_j y_j$

The Kernel Trick

As a consequence of Mercer's theorem,

- any valid *kernel* (i. e., symmetric, continuous, and nonnegative definite kernel function) $k(\mathbf{y}, \mathbf{y}')$ can be viewed as an inner product in some feature space
- complex computation of inner products in high-dimensional feature space may be evaded by direct computation of nonlinear $k(\mathbf{y}, \mathbf{y}')$ in input space
- given an algorithm based on a certain kernel $k(\mathbf{y}, \mathbf{y}')$, an alternative algorithm is obtained by simply replacing $k(\mathbf{y}, \mathbf{y}')$ by another kernel $k'(\mathbf{y}, \mathbf{y}')$
- using a *kernel* typically amounts to using a larger function class, thus increasing the capacity of the learning machine, and rendering problems separable that are not linearly separable to start with.



Example: The Kernel Trick

Consider the feature map

$$\begin{array}{rccc} \boldsymbol{\Phi} & : & \mathbb{R}^2 & \to & \mathbb{R}^3 \\ & & \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} & \mapsto & \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2} y_1 y_2 \end{bmatrix} \end{array}$$

Given the two arguments $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ and $\mathbf{y}' = \begin{bmatrix} y_1' \\ y_2' \end{bmatrix}$ the kernel function defined as inner product in the feature space reads

$$\begin{aligned} k(\mathbf{y}, \mathbf{y}') &= \langle \boldsymbol{\Phi}(\mathbf{y}), \boldsymbol{\Phi}(\mathbf{y}') \rangle \\ &= y_1^2 {y_1'}^2 + y_2^2 {y_2'}^2 + 2 y_1 y_1' y_2 y_2' \\ &= \left(\mathbf{y}^{\mathrm{T}} \mathbf{y}' \right)^2 \end{aligned}$$

(*polynomial kernel* of degree 2 in two-dimensional input space)

Similar feature maps exist for higher degree polynomial kernels in K-dimensional input spaces.

Valid Kernel Functions

Common symmetric, nonnegative definite kernel functions:

- polynomial kernels: $k(\mathbf{y}, \mathbf{y}') = \left(\mathbf{y}^{\mathrm{T}}\mathbf{y}'\right)^{D}$ with $D \in \mathbb{N}$
- *Gaussian* (radial basis function) kernels: $k(\mathbf{y}, \mathbf{y}') = \exp\left(-\frac{\|\mathbf{y}-\mathbf{y}'\|^2}{2\sigma^2}\right)$ with $\sigma > 0$
- *sigmoid* kernels: $k(\mathbf{y}, \mathbf{y}') = \tanh\left(\kappa\mathbf{y}^{\mathrm{T}}\mathbf{y}' + b_0\right)$ with $\kappa > 0$ and $b_0 \in \mathbb{R}$

Construction of New Kernels

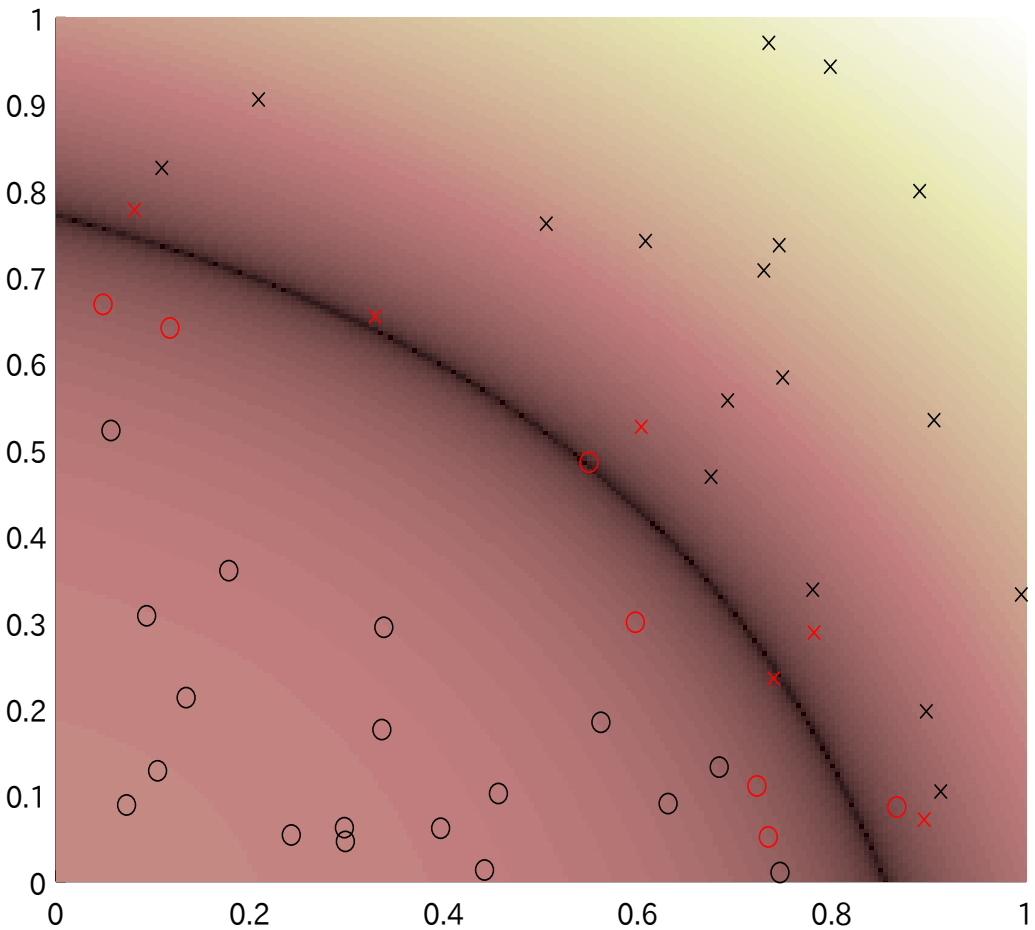
There are many possibilities for combining given valid kernels into new kernels.
For example, constructed from the valid kernels $k_1(\mathbf{y}, \mathbf{y}')$ and $k_2(\mathbf{y}, \mathbf{y}')$, the kernel functions

- $ck_1(\mathbf{y}, \mathbf{y}')$ with $c > 0$
- $\exp(k_1(\mathbf{y}, \mathbf{y}'))$
- $k_1(\mathbf{y}, \mathbf{y}') + k_2(\mathbf{y}, \mathbf{y}')$
- $k_1(\mathbf{y}, \mathbf{y}')k_2(\mathbf{y}, \mathbf{y}')$

are all valid kernels.

Examples: Nonlinear Support Vector Machines

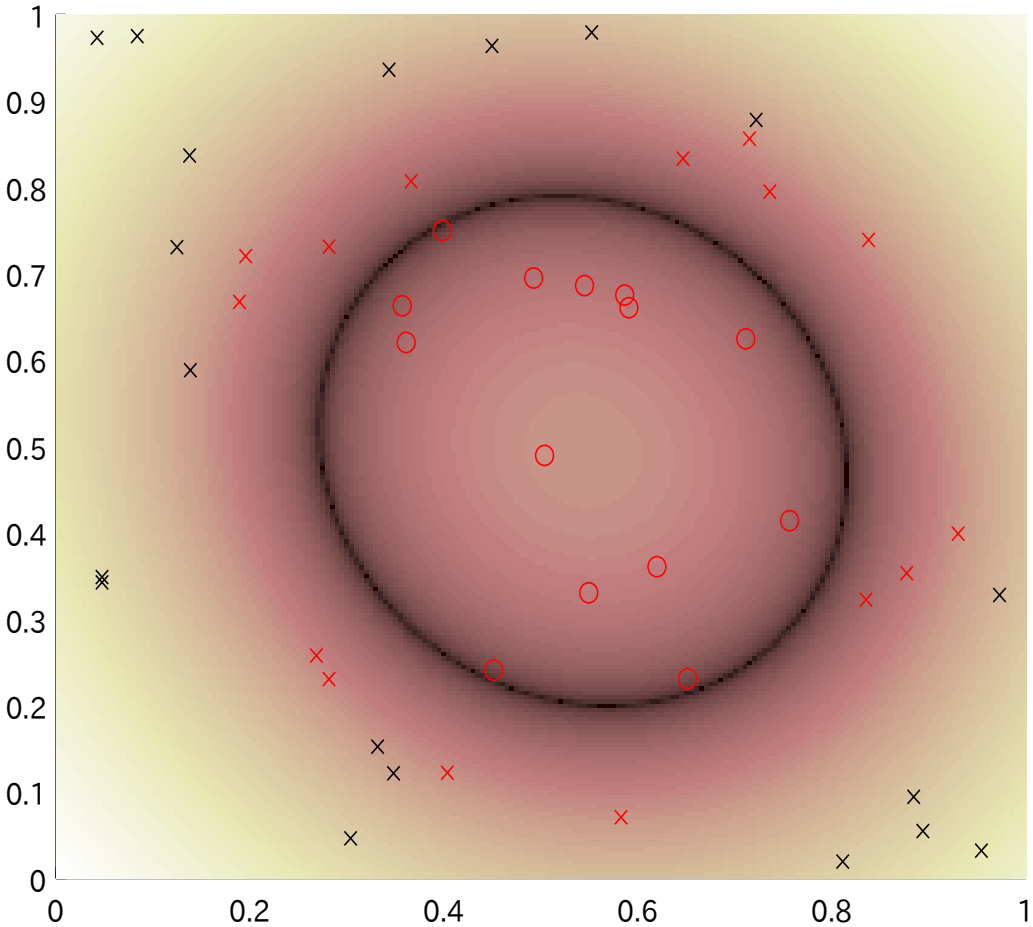
Example of a support vector machine based on a polynomial kernel:



- $\Gamma = [0,1] \times [0,1]$
- $k(\mathbf{y}, \mathbf{y}') = \left(\mathbf{y}^{\mathrm{T}} \mathbf{y}'\right)^3$
- points of training set are plotted as "x" (class C_1) or "o" (class C_2)
- binary classification using soft margin hyperplane ($C = 1000$)
- support vectors are plotted in red
- background color reflects absolute value of discriminant function $\sum_{\ell \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_\ell t_\ell k(\mathbf{y}_\ell, \mathbf{y}) + \tilde{b}$ over $\mathbf{y} \in \Gamma$

Examples: Nonlinear Support Vector Machines (cont.)

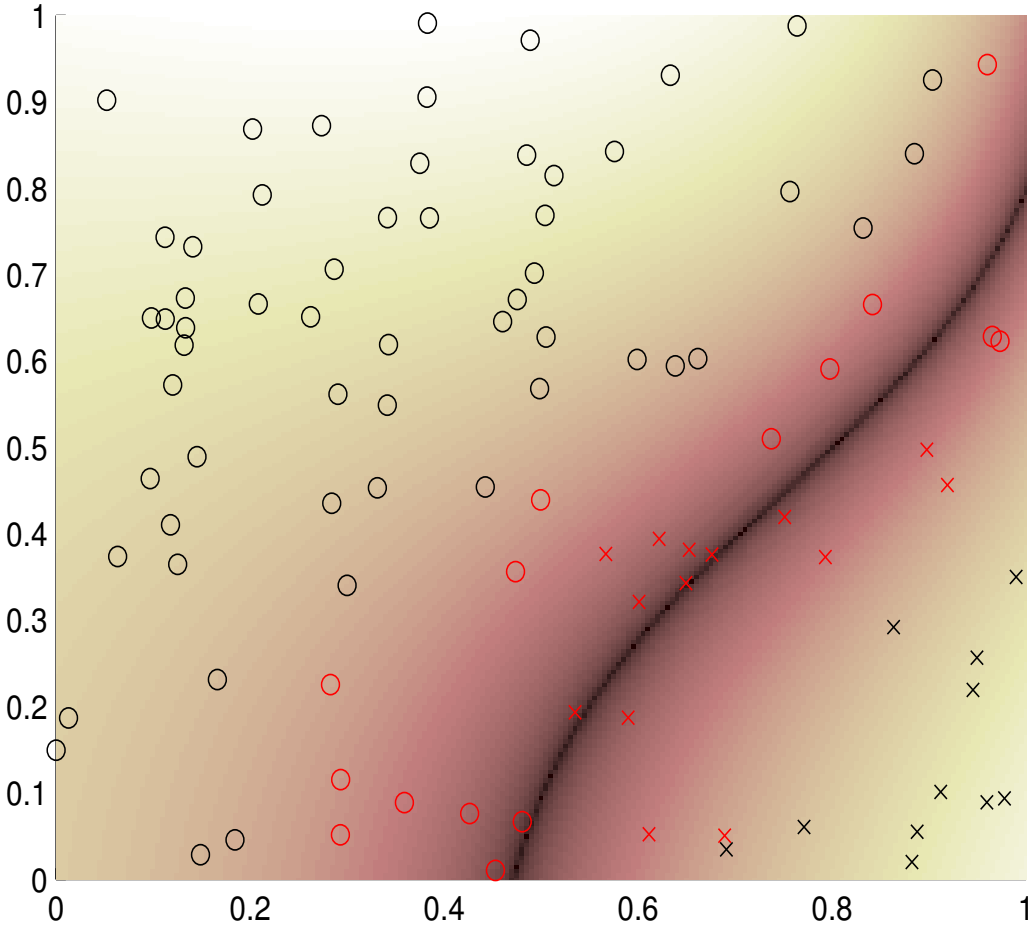
Example of a support vector machine based on a Gaussian kernel:



- $\Gamma = [0,1] \times [0,1]$
- $k(\mathbf{y}, \mathbf{y}') = \exp\left(-\frac{||\mathbf{y}-\mathbf{y}'||^2}{2}\right)$
- points of training set are plotted as "x" (class C_1) or "o" (class C_2)
- binary classification using soft margin hyperplane ($C = 1000$)
- support vectors are plotted in red
- background color reflects absolute value of discriminant function $\sum_{\ell \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_\ell t_\ell k(\mathbf{y}_\ell, \mathbf{y}) + \tilde{b}$ over

Examples: Nonlinear Support Vector Machines (cont.)

Example of a support vector machine based on a sigmoid kernel:



- $\Gamma = [0,1] \times [0,1]$
- $k(\mathbf{y}, \mathbf{y}') = \tanh\left(\mathbf{y}^{\mathrm{T}}\mathbf{y}' - 1.5\right)$
- points of training set are plotted as "x" (class C_1) or "o" (class C_2)
- binary classification using soft margin hyperplane ($C = 1000$)
- support vectors are plotted in red
- background color reflects absolute value of discriminant function $\sum_{\ell \in \mathcal{I}_{\mathrm{SV}}} \widetilde{\lambda}_\ell t_\ell k(\mathbf{y}_\ell, \mathbf{y}) + \tilde{b}$ over $\mathbf{y} \in \Gamma$

Handwritten Digit Recognition by Support Vector Machines

Experiment[1)]: recognition of handwritten digits by one-versus-the-rest classification using (10) soft margin hyperplanes trained by *US Postal Service* database, which includes 7291 samples in 16×16 pixel resolution.

1. Polynomial kernels $k(\mathbf{y}, \mathbf{y}') = \left(\mathbf{y}^{\mathrm{T}}\mathbf{y}'/256\right)^{D}$:

D	1	2	3	4	5	6	7
raw error in %	8.9	4.7	4.0	4.2	4.5	4.5	4.7
average number of SVs	282	237	274	321	374	422	491

2. Gaussian kernels $k(\mathbf{y}, \mathbf{y}') = \exp\left(-\left|\left|\mathbf{y} - \mathbf{y}'\right|\right|^2 / (256\ C_0)\right)$:

C_0	4.0	2.0	1.2	0.8	0.5	0.2	0.1
raw error in %	5.3	5.0	4.9	4.3	4.4	4.4	4.5
average number of SVs	266	240	233	235	251	366	722

[1)] All the shown results are reported in the book *Learning with Kernels – Support Vector Machines, Regularization, Optimization, and Beyond*, by B. Schölkopf and A. J. Smola, MIT, 2000.

Handwritten Digit Recognition by Support Vector Machines (cont.)

3. Sigmoid kernels $k(\mathbf{y}, \mathbf{y}') = \tanh\left(2\mathbf{y}^{\mathrm{T}}\mathbf{y}'/256 + B\right)$:

$-B$	0.8	0.9	1.0	1.1	1.2	1.3	1.4
raw error in %	6.3	4.8	4.1	4.3	4.3	4.4	4.8
average number of SVs	206	242	254	267	278	289	296

Conclusions:

- all three kernel types are able to achieve similarly good results
- error rates of down to 4.0% can be achieved as compared to 2.5% human error rate
- typically around 250 support vectors (SVs) are used per two-class-classifier, which is less than 4% of the samples in the training set

Exercises

1. For a given data set of 3-D input data, we apply the SVM learning algorithm and therefore achieve an optimal decision plane:

$$H(X) = x_1 + 2x_2 + 2x_3 + 3$$

What is the margin of this SVM? For three examples in the training set A:(-1,-1,-2) with label -1, B: (2,-2,-1) with label -1, and C: (-2,-2,2) with label +1, tell whether they are support vectors of the decision plane.

2. Prove that
   (a) Dot products are positive definite kernels.
   (b) Positive definite kernels are dot product kernels.
   hint: transformation $\Phi : \mathcal{H} = \Phi(\chi)$
   - construct a vector space $\mathcal{H}$
   - define the dot-product $< ., . >$ on $\mathcal{H}$
   - show that $k(x, y) =< \Phi(x), \Phi(y) >$ holds

3. Given the input space $\mathbb{R}^K$, show that the kernel function $k(\mathbf{y}, \mathbf{y}') = (\mathbf{y}^{\mathrm{T}}\mathbf{y}')^D$ with $D \in \mathbb{N}$ corresponds to an inner product in the feature space spanned by all possible Dth degree monomials composed of the elements of $\mathbf{y} = (y_1 \cdots y_K)^{\mathrm{T}}$, that is, all possible $y_1^{d_1} \cdot y_2^{d_2} \cdot \ \ldots \ \cdot y_K^{d_K}$ with $d_1, \ldots, d_K \in \{0, \ldots, D\}$ such that $\sum_{j=1}^{K} d_j = D$.

Table of Contents

Method of Least squares

Fitting of a model to a data set:

- given L data pairs $(\mathbf{y}_1, t_1), \ldots, (\mathbf{y}_L, t_L)$
- and a model $f(\mathbf{y}_\ell, \mathbf{w})$ for the determination of t_ℓ from $\mathbf{y}_\ell$, with the vector $\mathbf{w} = [w_1 \cdots w_K]^{\mathrm{T}}$ containing the model parameters
- assuming $L > K$ (i.e., overdetermined system)

Carl Friedrich Gauß

(1777–1855)

Sum of the squares of the residuals:

$$\epsilon(\mathbf{w}) = \sum_{\ell=1}^{L} (f(\mathbf{y}_\ell, \mathbf{w}) - t_\ell)^2$$

Least squares solution for parameter vector $\mathbf{w}$:

$$\widetilde{\mathbf{w}} = \arg \min_{\mathbf{w} \in \mathbb{R}^K} \epsilon(\mathbf{w})$$

General Linear Least Squares Problem

Linear basis function model:

$$f(\mathbf{y}, \mathbf{w}) = \sum_{k=1}^{K} w_k \phi_k(\mathbf{y})$$

with $\phi_1(\cdot), \ldots, \phi_K(\cdot)$ denoting K basis functions

Optimal parameter vector $\widetilde{\mathbf{w}}$ results from solving the *normal equation*

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\widetilde{\mathbf{w}} = \mathbf{Q}^{\mathrm{T}}\mathbf{t},$$

where

- $\mathbf{t} = [t_1 \cdots t_L]^{\mathrm{T}}$
- $\mathbf{q}_k = [\phi_k(\mathbf{y}_1) \cdots \phi_k(\mathbf{y}_L)]^{\mathrm{T}}$
- $\mathbf{Q} = [\mathbf{q}_1 \cdots \mathbf{q}_K] = \begin{bmatrix} \phi_1(\mathbf{y}_1) & \phi_2(\mathbf{y}_1) & \cdots & \phi_K(\mathbf{y}_1) \\ \phi_1(\mathbf{y}_2) & \phi_2(\mathbf{y}_2) & \cdots & \phi_K(\mathbf{y}_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(\mathbf{y}_L) & \phi_2(\mathbf{y}_L) & \cdots & \phi_K(\mathbf{y}_L) \end{bmatrix}$

Principle of Orthogonality

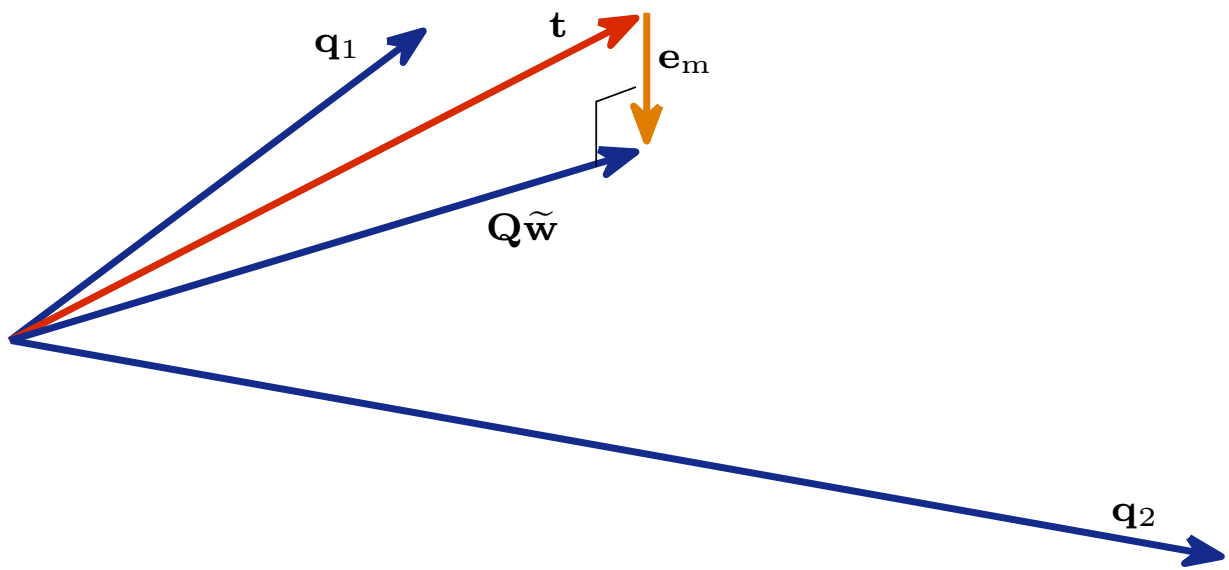
Optimal parameter vector $\widetilde{\mathbf{w}}$ leads to:

- $\mathbf{Q}\widetilde{\mathbf{w}}$: vector with versions of the data samples $t_1, \ldots, t_L$ fitted to the model
- $\mathbf{e}_\mathrm{m} = \mathbf{Q}\widetilde{\mathbf{w}} - \mathbf{t}$: vector with the residuals

Residual vector is normal to the range of $\mathbf{Q}$:

$$\mathbf{Q}^\mathrm{T}\left(\mathbf{Q}\widetilde{\mathbf{w}} - \mathbf{t}\right) = \begin{bmatrix} \mathbf{q}_1^\mathrm{T} \\ \vdots \\ \mathbf{q}_K^\mathrm{T} \end{bmatrix} \mathbf{e}_\mathrm{m} = 0$$

Example ($L = 3, K = 2$):



Computation of Linear Least Squares Solution

Solution of the normal equation:

$$\widetilde{\mathbf{w}} = \underbrace{\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right)^{-1}\mathbf{Q}^{\mathrm{T}}}_{=\mathbf{Q}^{+}}\mathbf{t}$$

$\mathbf{Q}^{+}$: *pseudoinverse* of $\mathbf{Q}$ under the assumption that $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ is non-singular

Efficient methods for solving the normal equation:

1. *Cholesky decomposition*
2. *singular value decomposition* (SVD)

Cholesky Decomposition

Cholesky decomposition of a symmetric, positive definite matrix $\mathbf{A}$:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathrm{T}} \quad \text{where} \quad \mathbf{L} = \begin{bmatrix} \blacksquare & 0 & \cdots & 0 \\ \blacksquare & \blacksquare & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \blacksquare & \blacksquare & \cdots & \blacksquare \end{bmatrix}$$

Solving of normal equation $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\widetilde{\mathbf{w}} = \mathbf{Q}^{\mathrm{T}}\mathbf{t}$:

1. Cholesky decomposition $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$
2. forward substitution step: solving the equation

   $$\mathbf{L}\mathbf{z} = \mathbf{Q}^{\mathrm{T}}\mathbf{t}$$

   for $\mathbf{z} = \mathbf{L}^{\mathrm{T}}\widetilde{\mathbf{w}}$
3. backward substitution step: solving the equation

   $$\mathbf{L}^{\mathrm{T}}\widetilde{\mathbf{w}} = \mathbf{z}$$

   for $\widetilde{\mathbf{w}}$

Mean-Squared Parameter Estimation

Estimation of a random parameter vector:

- random parameter vector $\boldsymbol{\Theta} \in \mathbb{R}^K$
- observation $\mathbf{y}$ of a random vector $\mathbf{Y} \in \Gamma$
- estimator of $\boldsymbol{\Theta}$ to be designed: function

$$\widehat{\theta} \; : \; \Gamma \to \mathbb{R}^K$$

mapping every element $\mathbf{y}$ of the observation set Γ to a parameter vector $\widehat{\theta}(\mathbf{y})$

Conditional *mean-squared error* given observation $\mathbf{y}$:

$$\begin{aligned} E\left(\left\|\widehat{\theta}(\mathbf{Y}) - \boldsymbol{\Theta}\right\|^2 \mid \mathbf{Y} = \mathbf{y}\right) &= E\left(\left\|\widehat{\theta}(\mathbf{Y})\right\|^2 \mid \mathbf{Y} = \mathbf{y}\right) \\ &\quad + E\left(\|\boldsymbol{\Theta}\|^2 \mid \mathbf{Y} = \mathbf{y}\right) - 2E\left(\left(\widehat{\theta}(\mathbf{Y})\right)^{\mathrm{T}} \boldsymbol{\Theta} \mid \mathbf{Y} = \mathbf{y}\right) \\ &= \left\|\widehat{\theta}(\mathbf{y})\right\|^2 - 2\left(\widehat{\theta}(\boldsymbol{y})\right)^{\mathrm{T}} E\left(\boldsymbol{\Theta} \mid \mathbf{Y} = \mathbf{y}\right) + E\left(\|\boldsymbol{\Theta}\|^2 \mid \mathbf{Y} = \mathbf{y}\right) \end{aligned}$$

Minimum mean-squared error (MMSE) estimator:

$$\widehat{\theta}_{\mathrm{MMSE}}(\mathbf{y}) = E\left(\boldsymbol{\Theta} \mid \mathbf{Y} = \mathbf{y}\right)$$

Linear MMSE Estimator

Linear MMSE (i.e., LMMSE) estimators:

- estimators of the form
$$\widehat{\theta}(\mathbf{y}) = \mathbf{A}\mathbf{y} + \mathbf{b}$$
are often easier to design than nonlinear MMSE estimators
- an LMMSE estimator $\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{y})$ fulfills
$$E\left(\left\|\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{Y}) - \mathbf{\Theta}\right\|^2\right) = \min_{\mathbf{A},\mathbf{b}} E\left(\|\mathbf{A}\mathbf{Y} + \mathbf{b} - \mathbf{\Theta}\|^2\right)$$

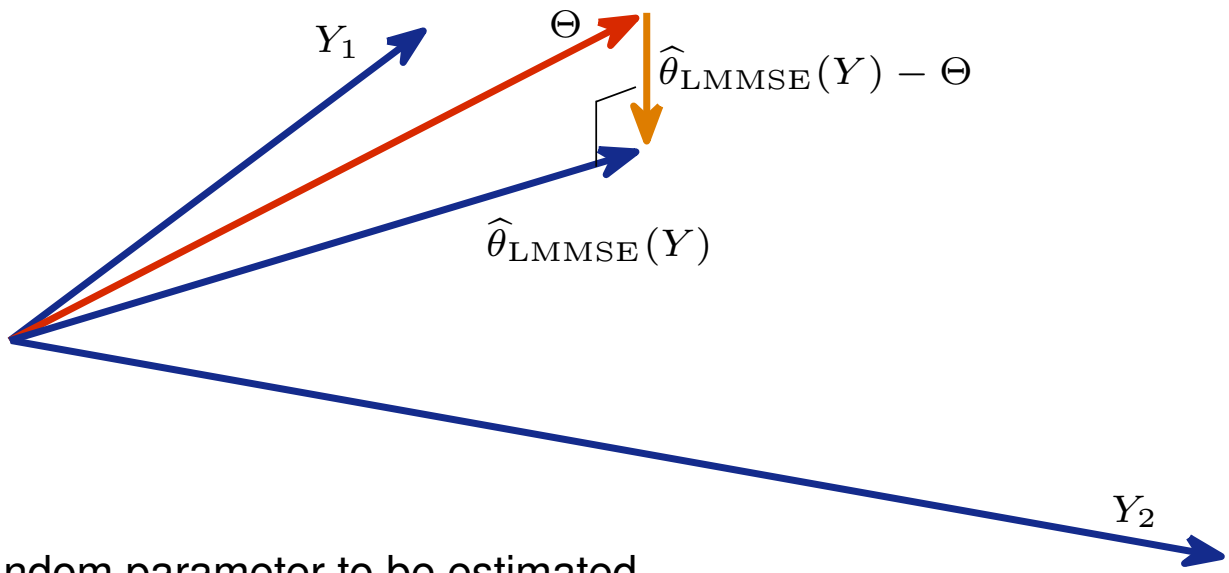
Problem: show that $\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{y})$ satisfies

$$E\left(\left(\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{Y}) - \mathbf{\Theta}\right)\left(f_{\mathrm{a}}(\mathbf{Y})\right)^{\mathrm{T}}\right) = 0$$

for any vector-valued affine function $f_{\mathrm{a}}(\cdot)$ (i.e., any $f_{\mathrm{a}}(\cdot)$ of the form $f_{\mathrm{a}}(\mathbf{y}) = \mathbf{A}_0\mathbf{y} + \mathbf{b}_0$).

Principle of Orthogonality for LMMSE Estimators

Example: vector space of zero-mean random variables, with inner product $\langle X, Y \rangle = E(XY)$



- Θ: unobservable random parameter to be estimated
- $Y_1, Y_2, \ldots$: observable random variables
- $\widehat{\theta}_{\mathrm{LMMSE}}(Y)$: linear combination of $Y_1, Y_2, \ldots$ such that distance to Θ, i.e.,

$$\sqrt{E\left(\widehat{\theta}_{\mathrm{LMMSE}}(Y) - \Theta\right)^2}$$

becomes minimal

Observation with Measurement Error

Consider scenarios where the observation is given by

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\Theta} + \mathbf{V}$$

where

- the random parameter vector $\boldsymbol{\Theta}$ has mean $\mathbf{m}_{\boldsymbol{\Theta}}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}$
- $\mathbf{H}$ is a given measurement matrix
- the random measurement error $\mathbf{V}$ is zero-mean with covariance matrix $\boldsymbol{\Sigma}_{\mathbf{V}}$
- $\boldsymbol{\Theta}$ and $\mathbf{V}$ are independent

Applying the principle of orthogonality with $f_{\mathrm{a}}(\mathbf{Y}) = 1$ and $f_{\mathrm{a}}(\mathbf{Y}) = \mathbf{Y}$ leads to

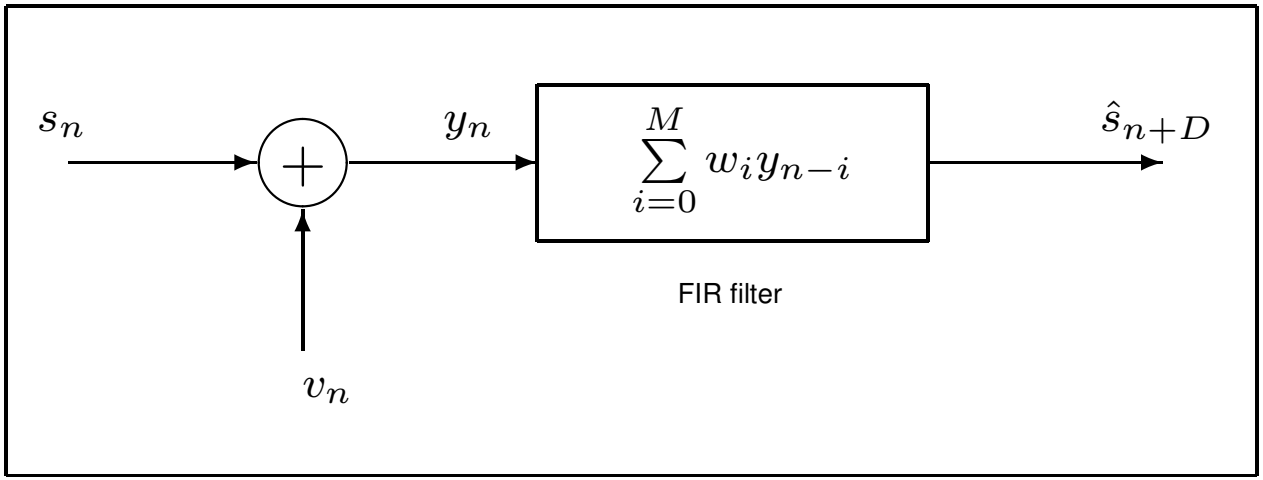
- the LMMSE estimator

$$\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{y}) = \boldsymbol{\Sigma}_{\boldsymbol{\Theta}}\mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}\mathbf{H}^{\mathrm{T}} + \boldsymbol{\Sigma}_{\mathbf{V}}\right)^{-1}(\mathbf{y} - \mathbf{H}\mathbf{m}_{\boldsymbol{\Theta}}) + \mathbf{m}_{\boldsymbol{\Theta}}$$

- with the error covariance matrix

$$E\left(\left(\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{Y}) - \boldsymbol{\Theta}\right)\left(\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{Y}) - \boldsymbol{\Theta}\right)^{\mathrm{T}}\right) = \left(\boldsymbol{\Sigma}_{\boldsymbol{\Theta}}^{-1} + \mathbf{H}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{V}}^{-1}\mathbf{H}\right)^{-1}$$

Wiener Filter



Norbert Wiener
(1894-1964)

Andrei Nikolajewitsch
Kolmogorov
(1903-1987)

Input of finite impulse response (FIR) filter:

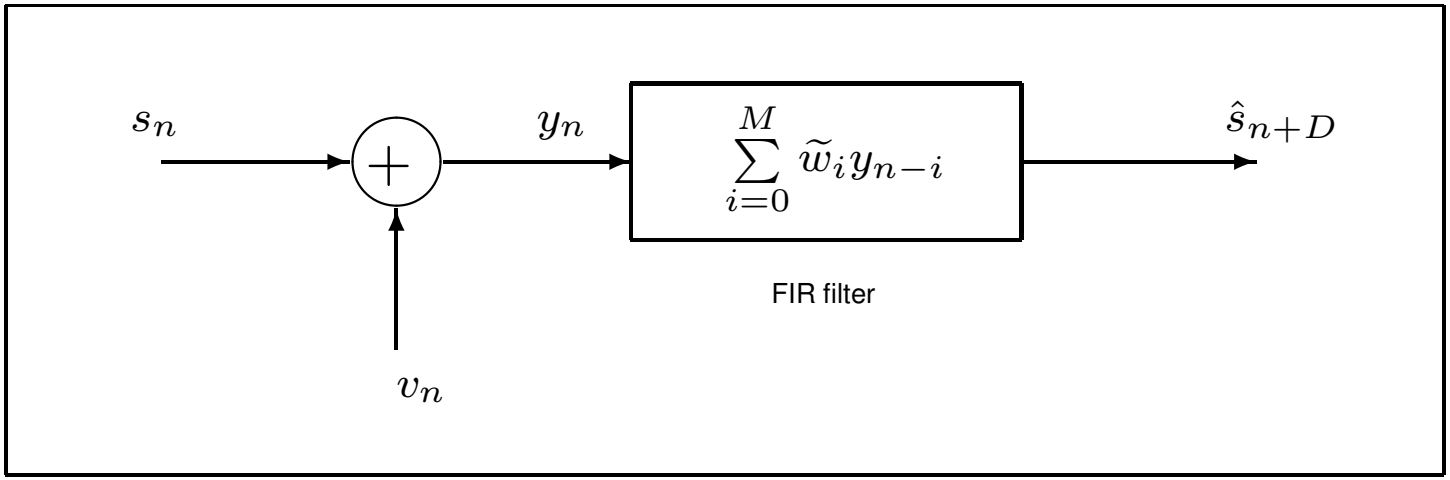
- superposition of signal $(s_n \ : \ n \in \mathbb{Z})$ and noise $(v_n \ : \ n \in \mathbb{Z})$
- viewed as two independent, zero-mean wide-sense stationary random processes

Wiener filter of order M:

- constituted by impulse response $\widetilde{w}_0, \ldots, \widetilde{w}_M$ chosen such that filter output achieves minimal mean-squared error

$$E\left(\left(\hat{s}_{n+D} - s_{n+D}\right)^2\right)$$

Smoothing, Filtering, and Prediction



Possible tasks of the Wiener filter:

- *smoothing* ($D < 0$): retrospective elimination of noise from the observed signal
- *filtering* ($D = 0$): recreation of signal s_n from the noisy observation y_n in real time
- *prediction* ($D > 0$): forecast of future course of the signal

Wiener-Hopf Equations

Second-order properties of the wide-sense stationary random processes:

- $r_y(d) = E(y_{n+d}\, y_n)$: autocorrelation function of the filter input
- $r_{sy}(d) = E(s_{n+d}\, y_n)$: cross-correlation between desired response and filter input

The impulse response vector $\widetilde{\mathbf{w}} = [\widetilde{w}_0 \cdots \widetilde{w}_M]^{\mathrm{T}}$ of the Wiener filter results from solving

$$\mathbf{R}_y \widetilde{\mathbf{w}} = \mathbf{r}_{sy},$$

known as the *Wiener-Hopf equations* with

$$\mathbf{R}_y = \begin{bmatrix} r_y(0) & r_y(1) & \cdots & r_y(M) \\ r_y(1) & r_y(0) & \cdots & r_y(M-1) \\ \vdots & \vdots & & \vdots \\ r_y(M) & r_y(M-1) & \cdots & r_y(0) \end{bmatrix} \quad \text{and} \quad \mathbf{r}_{sy} = \begin{bmatrix} r_{sy}(D) \\ r_{sy}(D+1) \\ \vdots \\ r_{sy}(D+M) \end{bmatrix}.$$

Efficient recursive solution of the Wiener-Hopf equations: *Levinson-Durbin recursion*.

Performance of Wiener Filter

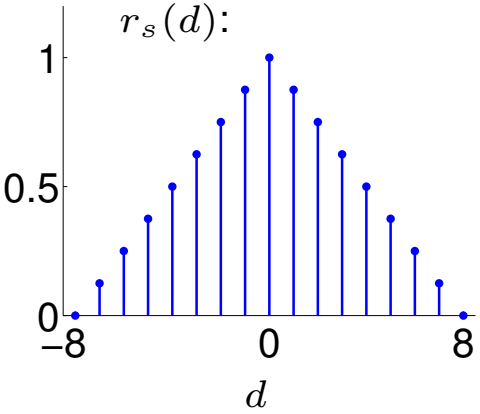
With $\hat{s}_{n+D} = \sum_{i=0}^{M} \widetilde{w}_i y_{n-i}$ and $\varepsilon_s = E(s_n^2)$, the mean-squared error achieved by the Wiener filter results as

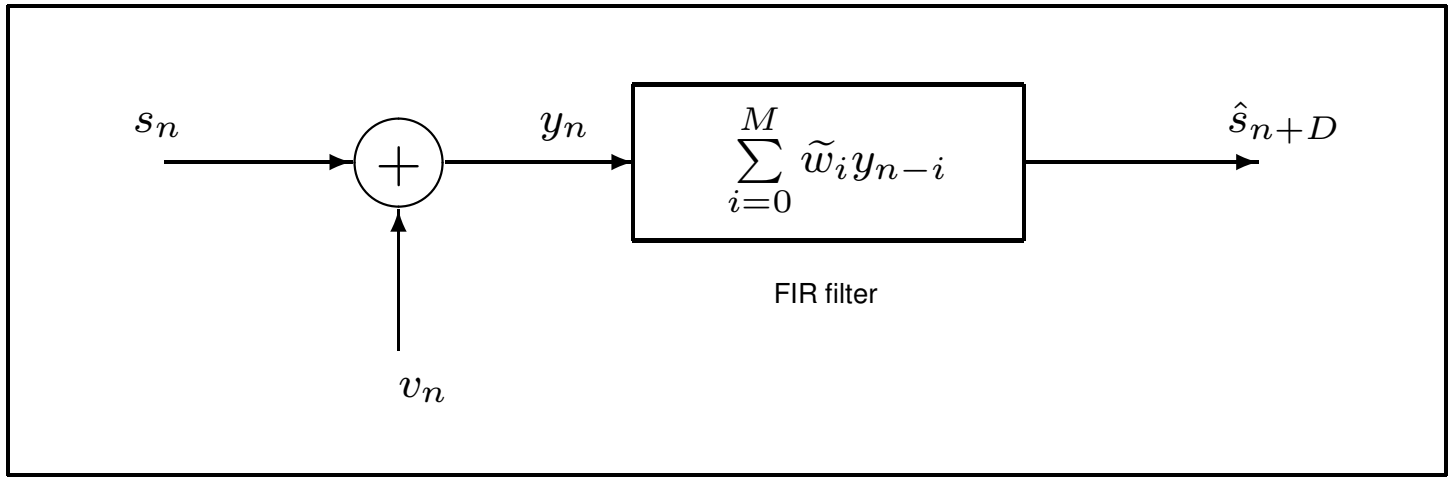
$$\begin{aligned} E\left((\hat{s}_{n+D} - s_{n+D})^2\right) &= \widetilde{\mathbf{w}}^{\mathrm{T}} \mathbf{R}_y \widetilde{\mathbf{w}} - \widetilde{\mathbf{w}}^{\mathrm{T}} \mathbf{r}_{sy} - \mathbf{r}_{sy}^{\mathrm{T}} \widetilde{\mathbf{w}} + \varepsilon_s \\ &= \varepsilon_s - \widetilde{\mathbf{w}}^{\mathrm{T}} \mathbf{R}_y \widetilde{\mathbf{w}} \\ &= \varepsilon_s - \mathbf{r}_{sy}^{\mathrm{T}} \mathbf{R}_y^{-1} \mathbf{r}_{sy}. \end{aligned}$$

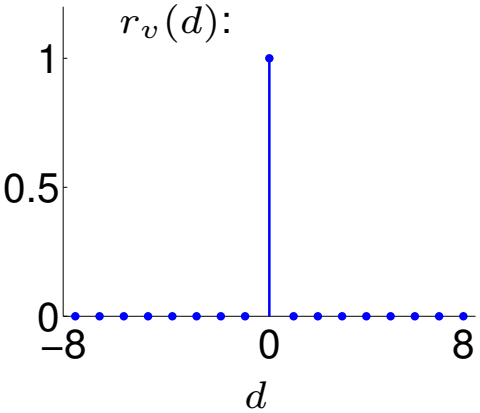
Remarks:

- the Wiener filter and its performance depend only on $r_y(d)$, $r_{sy}(d)$, and ε_s
- the Wiener filter is applicable in the general case where filter input and desired signal are both wise-sense stationary
- as e. g. in echo cancellation in digital communication systems, the desired signal part may appear linearly distorted at the Wiener filter input

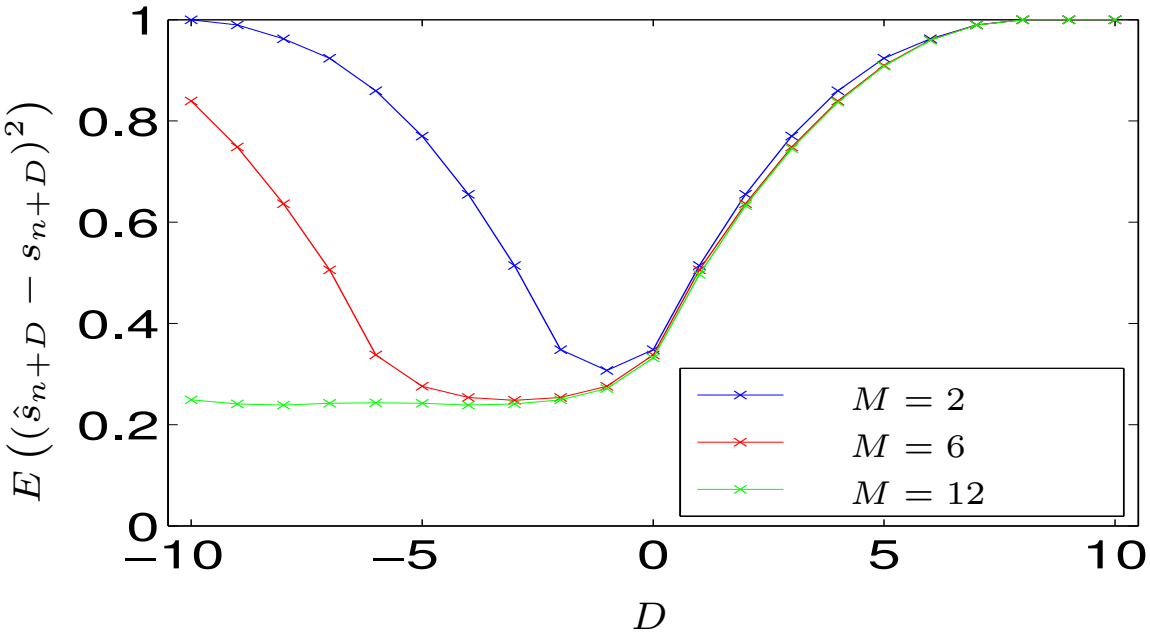
Example







$$r_y(d) = r_s(d) + r_v(d)$$
$$r_{sy}(d) = r_s(d)$$



Levinson-Durbin Recursion

Suppose: $\mathbf{A}^{(n+1)} = \begin{bmatrix} \mathbf{A}^{(n)} & \mathbf{g} \\ \mathbf{h}^{\mathrm{T}} & d \end{bmatrix}$ is an $(n+1)\times(n+1)$ *Toeplitz* matrix, $\mathbf{y}^{(n+1)} = \begin{bmatrix} \mathbf{y}^{(n)} \\ y \end{bmatrix}$,

$\boxed{\mathbf{A}^{(n)}\mathbf{x}^{(n)} = \mathbf{y}^{(n)},}$ and additionally $\mathbf{A}^{(n)}\mathbf{f}^{(n)} = \begin{bmatrix} 1 \\ \mathbf{0}_{n-1} \end{bmatrix}$, $\mathbf{A}^{(n)}\mathbf{b}^{(n)} = \begin{bmatrix} \mathbf{0}_{n-1} \\ 1 \end{bmatrix}$

then:

$$\mathbf{A}^{(n+1)} \begin{bmatrix} \mathbf{x}^{(n)} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{y}^{(n)} \\ \varepsilon_x \end{bmatrix}, \; \mathbf{A}^{(n+1)} \begin{bmatrix} \mathbf{f}^{(n)} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{0}_{n-1} \\ \varepsilon_f \end{bmatrix}, \; \mathbf{A}^{(n+1)} \begin{bmatrix} 0 \\ \mathbf{b}^{(n)} \end{bmatrix} = \begin{bmatrix} \varepsilon_b \\ \mathbf{0}_{n-1} \\ 1 \end{bmatrix}$$

define:

$$\mathbf{f}^{(n+1)} = \frac{1}{1-\varepsilon_b\varepsilon_f} \begin{bmatrix} \mathbf{f}^{(n)} \\ 0 \end{bmatrix} - \frac{\varepsilon_f}{1-\varepsilon_b\varepsilon_f} \begin{bmatrix} 0 \\ \mathbf{b}^{(n)} \end{bmatrix}, \quad \mathbf{b}^{(n+1)} = \frac{1}{1-\varepsilon_b\varepsilon_f} \begin{bmatrix} 0 \\ \mathbf{b}^{(n)} \end{bmatrix} - \frac{\varepsilon_b}{1-\varepsilon_b\varepsilon_f} \begin{bmatrix} \mathbf{f}^{(n)} \\ 0 \end{bmatrix}$$

and

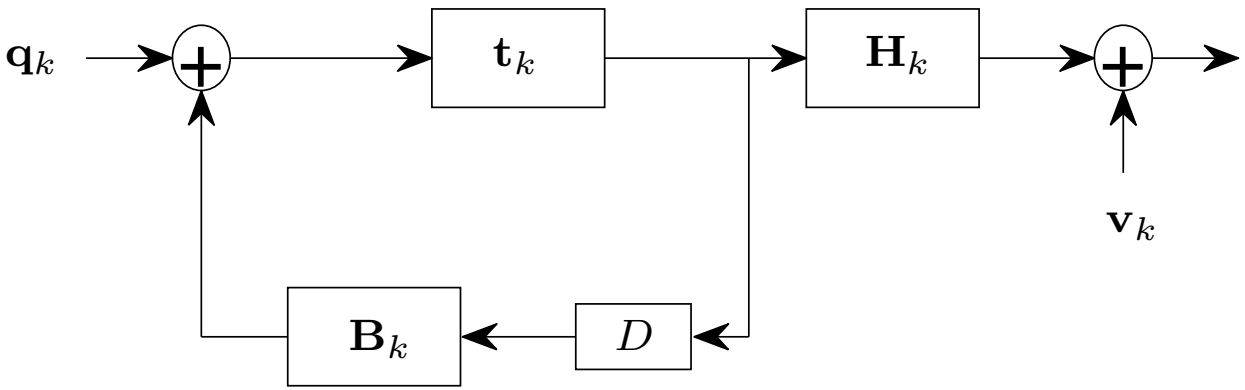
$$\mathbf{x}^{(n+1)} = \begin{bmatrix} \mathbf{x}^{(n)} \\ 0 \end{bmatrix} + (y - \varepsilon_x)\mathbf{b}^{(n+1)}$$

then:

$\boxed{\mathbf{A}^{(n+1)}\mathbf{x}^{(n+1)} = \mathbf{y}^{(n+1)},}$ $\mathbf{A}^{(n+1)}\mathbf{f}^{(n+1)} = \begin{bmatrix} 1 \\ \mathbf{0}_n \end{bmatrix}$, $\mathbf{A}^{(n+1)}\mathbf{b}^{(n+1)} = \begin{bmatrix} \mathbf{0}_n \\ 1 \end{bmatrix}$

Kalman Filter: State Space Representation

Discrete-time state space model of a dynamical system:



Rudolf Emil Kalman
(born 1930)

- $\mathbf{t}_k$: internal state vector (with subscript k representing time)
- $\mathbf{H}_k$: deterministic *measurement matrix*
- $\mathbf{v}_k$: random zero-mean measurement error vector with covariance matrix $\mathbf{W}_k$
- $\mathbf{y}_k$: vector-valued observation
- D: delay, $\mathbf{t}_k \mapsto \mathbf{t}_{k+1}$
- $\mathbf{B}_k$: deterministic *state transition matrix*
- $\mathbf{q}_k$: random vector with mean $\mathbf{u}_k$ (i.e., the system input) and covariance matrix $\mathbf{P}_k$

  (random vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{q}_1, \mathbf{q}_2, \ldots$ are independent)

Prediction Step

Based on the observations $\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$, suppose we know (for some $k \geq 2$)

- the LMMSE estimate $\hat{\mathbf{t}}_{k-1}$ of the state vector $\mathbf{t}_{k-1}$
- the corresponding error covariance matrix

$$\mathbf{R}_{k-1} = E\left(\left(\hat{\mathbf{t}}_{k-1} - \mathbf{t}_{k-1}\right) \left(\hat{\mathbf{t}}_{k-1} - \mathbf{t}_{k-1}\right)^{\mathrm{T}} \,\middle|\, \mathbf{y}_1, \ldots, \mathbf{y}_{k-1} \right)$$

Principle of orthogonality: $E\left(\left(\hat{\mathbf{t}}_{k-1} - \mathbf{t}_{k-1}\right) \left(f_{\mathrm{a}}(\mathbf{y}_1, \ldots, \mathbf{y}_{k-1})\right)^{\mathrm{T}} \right) = \mathbf{0}$

Prediction of system state and observation at time k:

- LMMSE estimate of the internal state: $\bar{\mathbf{t}}_k = \mathbf{B}_k \hat{\mathbf{t}}_{k-1} + \mathbf{u}_k$
- corresponding error covariance matrix: $\bar{\mathbf{R}}_k = \mathbf{B}_k \mathbf{R}_{k-1} \mathbf{B}_k^{\mathrm{T}} + \mathbf{P}_k$
- LMMSE estimate of the observation: $\bar{\mathbf{y}}_k = \mathbf{H}_k \bar{\mathbf{t}}_k$

Correction Step

Update of the estimate of state vector $\mathbf{t}_k$ taking latest observation $\mathbf{y}_k$ into account:

- the conditional mean and covariance matrix of the random vector $\mathbf{t}_k$ are

$$E\left(\mathbf{t}_k \,\middle|\, \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}\right) = \bar{\mathbf{t}}_k, \qquad E\left(\left(\mathbf{t}_k - \bar{\mathbf{t}}_k\right)\left(\mathbf{t}_k - \bar{\mathbf{t}}_k\right)^{\mathrm{T}} \,\middle|\, \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}\right) = \bar{\mathbf{R}}_k$$

- applying the formula from LMMSE parameter estimation to

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{t}_k + \mathbf{v}_k,$$

the LMMSE estimate of $\mathbf{t}_k$ follows as
$\hat{\mathbf{t}}_k = \bar{\mathbf{R}}_k \mathbf{H}_k^{\mathrm{T}} \left(\mathbf{H}_k \bar{\mathbf{R}}_k \mathbf{H}_k^{\mathrm{T}} + \mathbf{W}_k\right)^{-1} \left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right) + \bar{\mathbf{t}}_k$

- the LMMSE estimate of the internal state is thus given by

$$\hat{\mathbf{t}}_k = \bar{\mathbf{t}}_k + \mathbf{G}_k(\mathbf{y}_k - \bar{\mathbf{y}}_k)$$

with the *Kalman gain*

$$\mathbf{G}_k = \bar{\mathbf{R}}_k \mathbf{H}_k^{\mathrm{T}} \left(\mathbf{H}_k \bar{\mathbf{R}}_k \mathbf{H}_k^{\mathrm{T}} + \mathbf{W}_k\right)^{-1}$$

- and the corresponding error covariance matrix $\mathbf{R}_k$

Derivation of Error Covariance Matrix

To obtain a formula for $\mathbf{R}_k = E\left(\left(\mathbf{t}_k - \hat{\mathbf{t}}_k\right)\left(\mathbf{t}_k - \hat{\mathbf{t}}_k\right)^{\mathrm{T}} \,\middle|\, \mathbf{y}_1, \ldots, \mathbf{y}_k\right)$

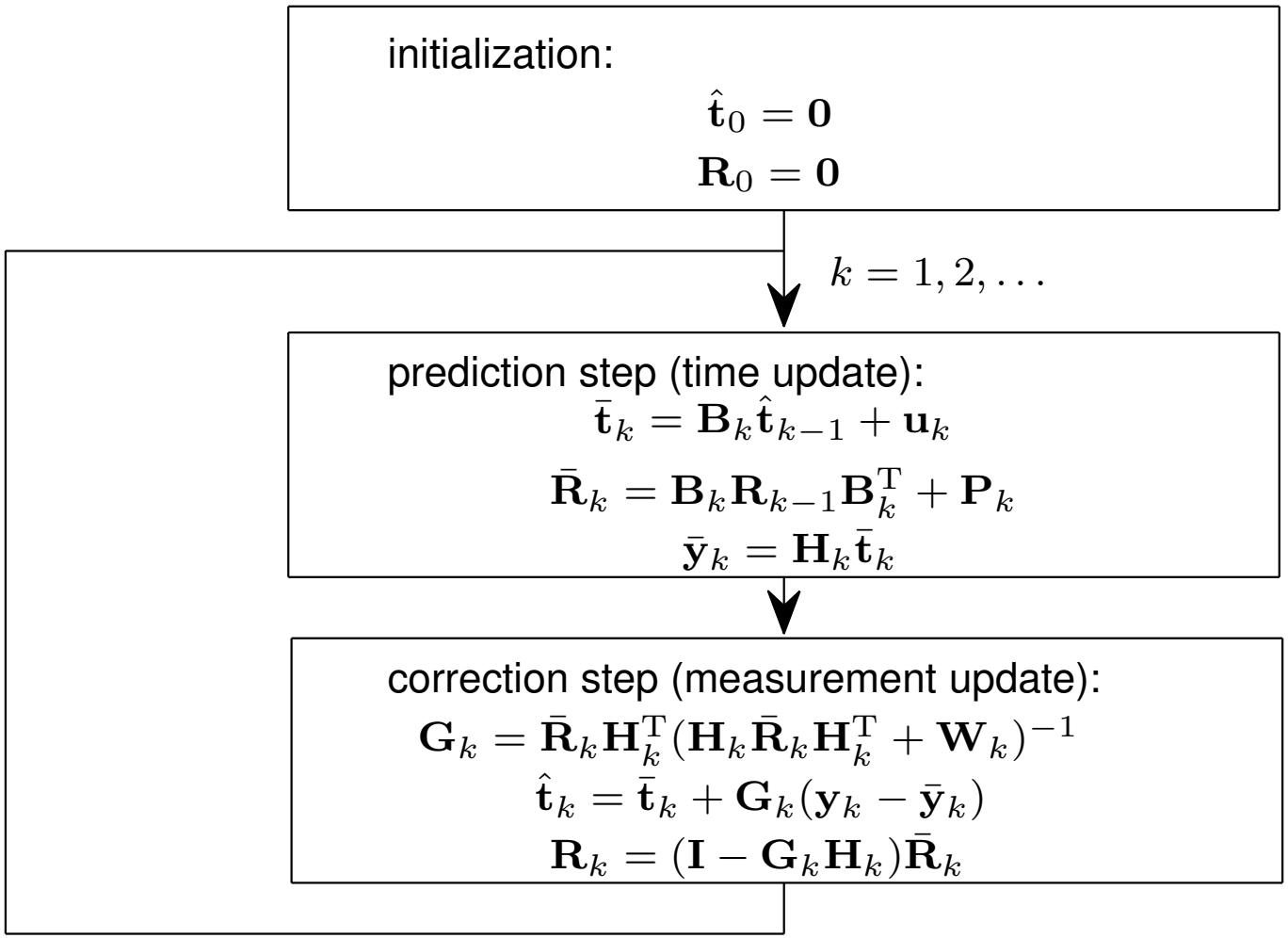
- write $\mathbf{t}_k - \hat{\mathbf{t}}_k$ as

$$\begin{aligned}
\mathbf{t}_k - \hat{\mathbf{t}}_k &= \mathbf{B}_k\mathbf{t}_{k-1} + \mathbf{q}_k - \bar{\mathbf{t}}_k - \mathbf{G}_k\left(\mathbf{y}_k - \bar{\mathbf{y}}_k\right) \\
&= \mathbf{B}_k\mathbf{t}_{k-1} + \mathbf{q}_k - \mathbf{B}_k\hat{\mathbf{t}}_{k-1} - \mathbf{u}_k \\
&\quad -\mathbf{G}_k\left(\mathbf{H}_k\left(\mathbf{B}_k\mathbf{t}_{k-1} + \mathbf{q}_k\right) + \mathbf{v}_k - \mathbf{H}_k\left(\mathbf{B}_k\hat{\mathbf{t}}_{k-1} + \mathbf{u}_k\right)\right) \\
&= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\mathbf{B}_k\left(\mathbf{t}_{k-1} - \hat{\mathbf{t}}_{k-1}\right) + \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\left(\mathbf{q}_k - \mathbf{u}_k\right) - \mathbf{G}_k\mathbf{v}_k
\end{aligned}$$

- and then

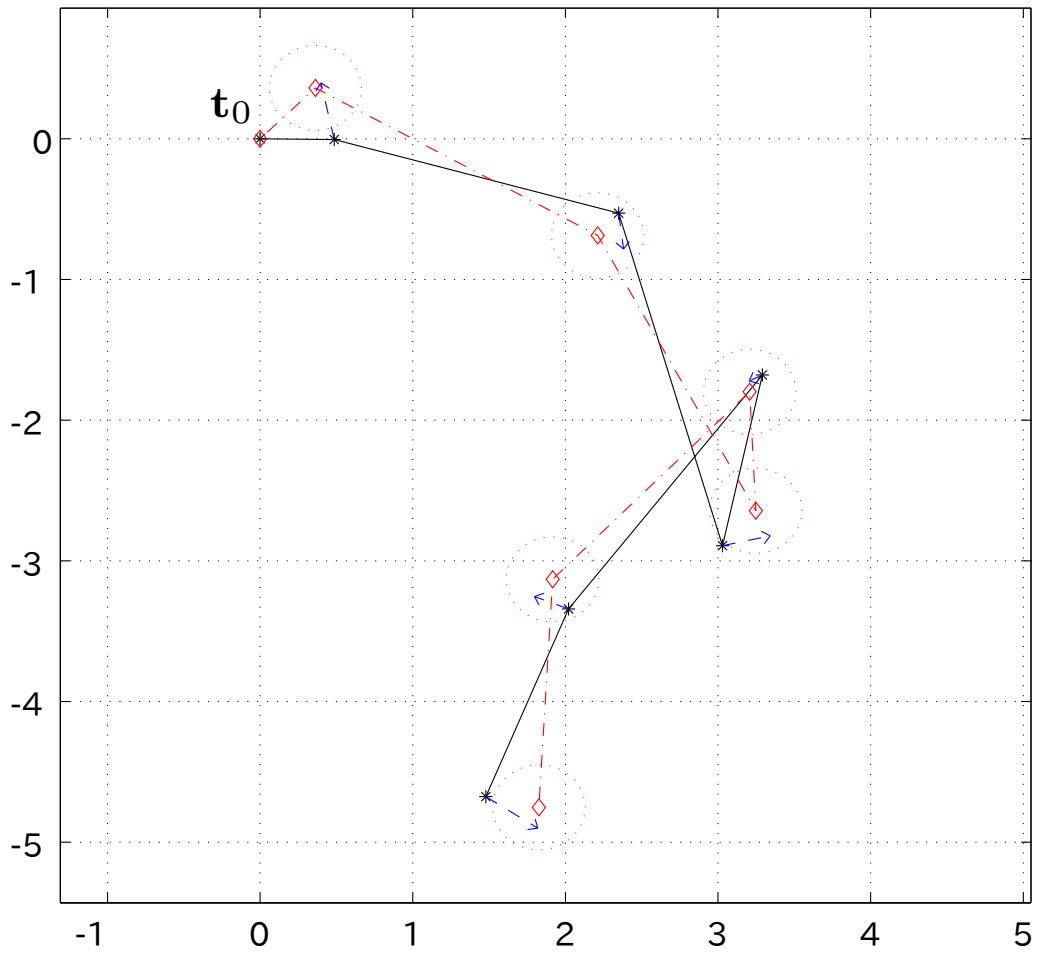
$$\begin{aligned}
\mathbf{R}_k &= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\left(\mathbf{B}_k\mathbf{R}_{k-1}\mathbf{B}_k^{\mathrm{T}} + \mathbf{P}_k\right)\left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)^{\mathrm{T}} + \mathbf{G}_k\mathbf{W}_k\mathbf{G}_k^{\mathrm{T}} \\
&= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\bar{\mathbf{R}}_k\left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)^{\mathrm{T}} + \mathbf{G}_k\mathbf{W}_k\mathbf{G}_k^{\mathrm{T}} \\
&= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\bar{\mathbf{R}}_k - \bar{\mathbf{R}}_k\mathbf{H}_k^{\mathrm{T}}\mathbf{G}_k^{\mathrm{T}} + \mathbf{G}_k\mathbf{H}_k\bar{\mathbf{R}}_k\mathbf{H}_k^{\mathrm{T}}\mathbf{G}_k^{\mathrm{T}} + \mathbf{G}_k\mathbf{W}_k\mathbf{G}_k^{\mathrm{T}} \\
&= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\bar{\mathbf{R}}_k - \bar{\mathbf{R}}_k\mathbf{H}_k^{\mathrm{T}}\mathbf{G}_k^{\mathrm{T}} + \mathbf{G}_k\left(\mathbf{H}_k\bar{\mathbf{R}}_k\mathbf{H}_k^{\mathrm{T}} + \mathbf{W}_k\right)\mathbf{G}_k^{\mathrm{T}} \\
&= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\bar{\mathbf{R}}_k - \bar{\mathbf{R}}_k\mathbf{H}_k^{\mathrm{T}}\mathbf{G}_k^{\mathrm{T}} + \bar{\mathbf{R}}_k\mathbf{H}_k^{\mathrm{T}}\mathbf{G}_k^{\mathrm{T}} \\
&= \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\bar{\mathbf{R}}_k
\end{aligned}$$

Algorithm Summary



Example

Tracking of Brownian motion-like 2-D random process:



- starting from $\mathbf{t}_0 = [0\ 0]^{\mathrm{T}}$, course $\mathbf{t}_1, \mathbf{t}_2, \ldots$ is shown in black
- every $\mathbf{q}_k$ is Gaussian distributed with $\mathbf{u}_k = \mathbf{0}$ and $\mathbf{P}_k = \mathbf{I}_2$
- measurement errors $\mathbf{v}_1, \mathbf{v}_2, \ldots$ (shown in blue) are jointly Gaussian distributed with covariance matrix $\mathbf{W}_k = 0.1 \cdot \mathbf{I}_2$
- estimates $\hat{\mathbf{t}}_0, \hat{\mathbf{t}}_1, \ldots$ are shown in red
- error covariance matrices have the form $\mathbf{R}_k = \sigma_k^2 \mathbf{I}_2$, represented by magenta-colored circles with radius σ_k

Comparison of Kalman Filter and Wiener Filter

Wiener filter:

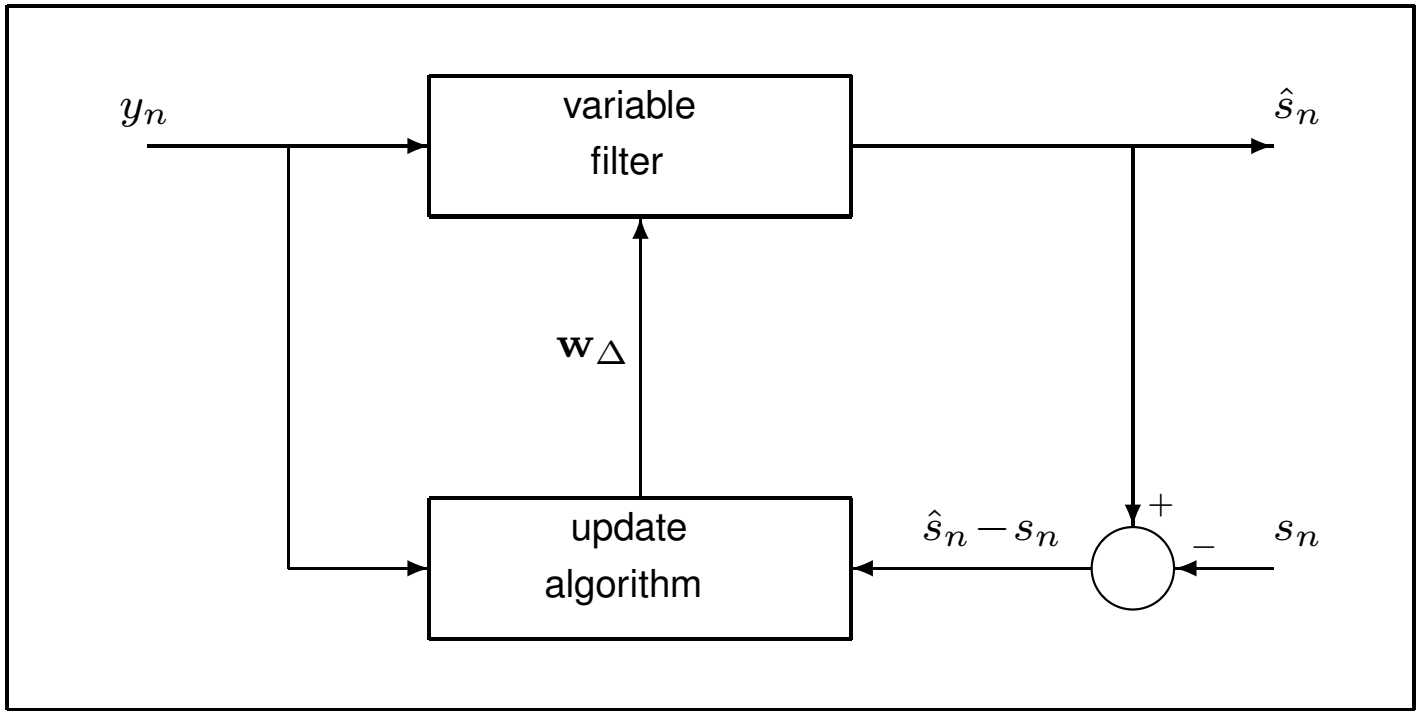
- builds on autocorrelation and cross-correlation information
- static finite-length filter
- applicable only in stationary systems
- acting directly on observation

Kalman filter:

- builds on state space representation with second-order information on random processes
- recursive computation of infinite-length filter
- applicable in dynamic systems complying with first-order Markov model
- prediction/correction procedure

Adaptive Filters

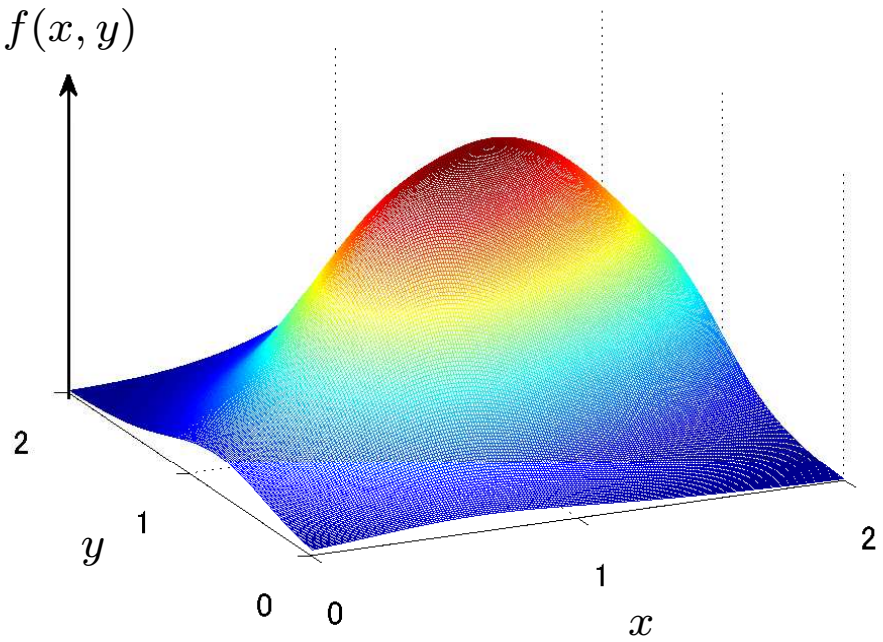
General structure of an adaptive filter:

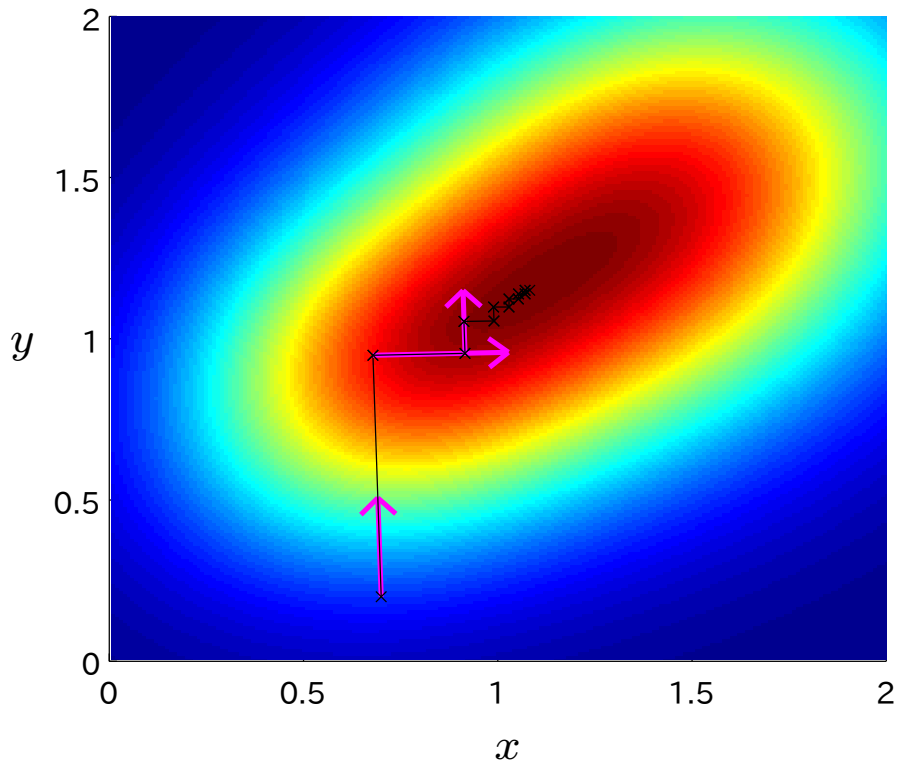


Adaptive filter algorithms:

- least mean squares (LMS)
- recursive least squares (RLS)
- etc.

Example: Method of Steepest Descent:





- left: 3D-plot of a function $f(x,y)$
- right: iterative procedure for finding a maximum of $f(x,y)$, with first three gradients shown in magenta

Least Mean Squares (LMS) Filter

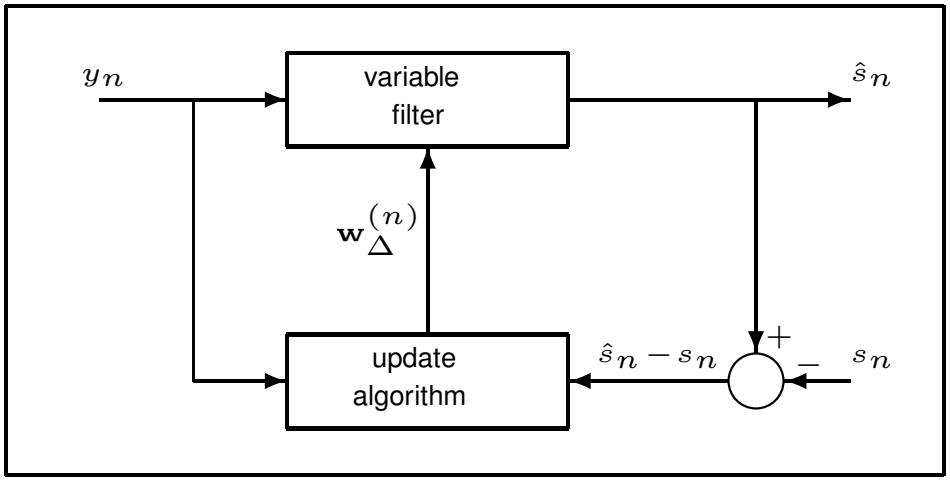
- cost function for LMS filter of order M:

$$\varepsilon_{\mathrm{MSE}}(\mathbf{w}) = E\left(\left(\mathbf{y}_n^{\mathrm{T}}\mathbf{w} - s_n\right)^2\right)$$

with $\mathbf{y}_n = (y_n \; y_{n-1} \cdots y_{n-M})^{\mathrm{T}}$

- gradient:

$$\nabla\varepsilon_{\mathrm{MSE}}(\mathbf{w}) = E\left(2\left(\mathbf{y}_n^{\mathrm{T}}\mathbf{w} - s_n\right)\mathbf{y}_n\right)$$



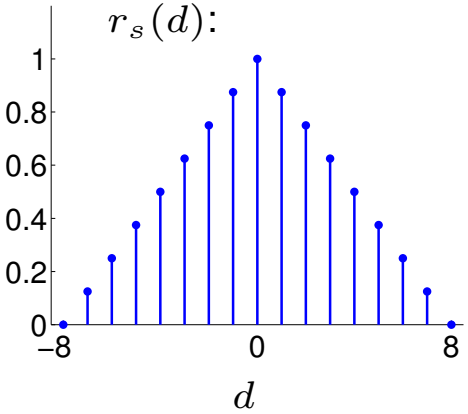
Iterative LMS algorithm:

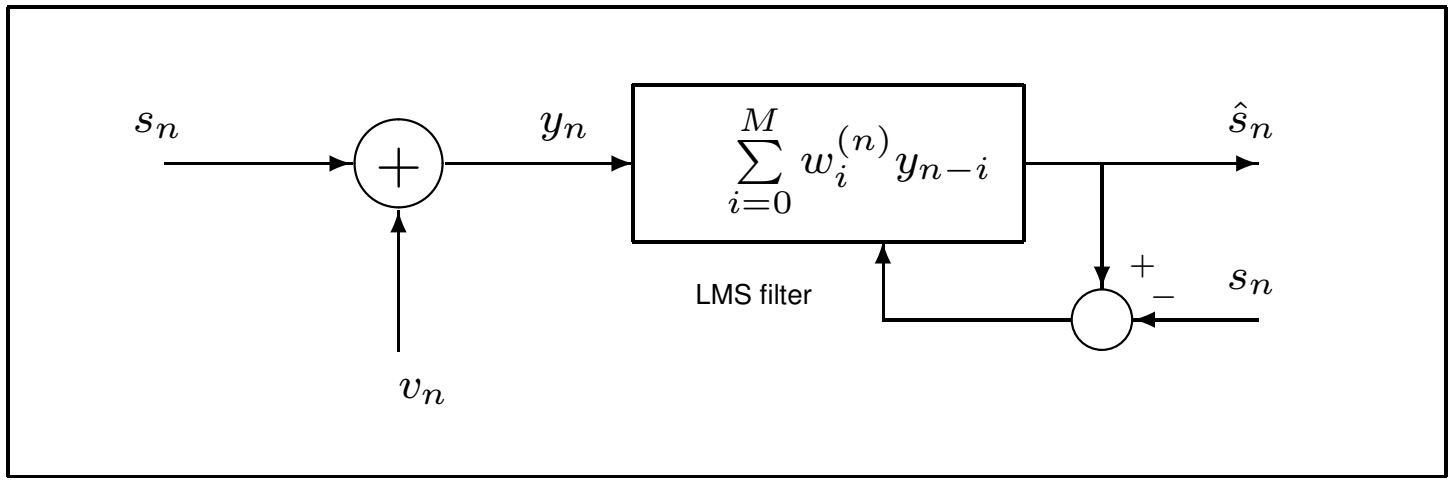
- initialization: $\mathbf{w}^{(0)} = \mathbf{0}_{M+1}$
- update at $n = 1, 2, \ldots$:

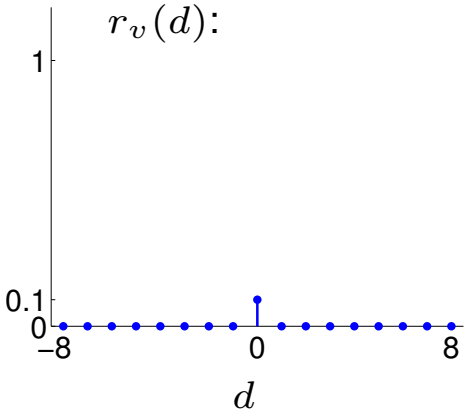
$$\mathbf{w}^{(n)} = \mathbf{w}^{(n-1)} \underbrace{-\mu\left(\mathbf{y}_n^{\mathrm{T}}\mathbf{w}^{(n-1)} - s_n\right)\mathbf{y}_n}_{\mathbf{w}_\Delta^{(n)}}$$

- μ: step size (or *learning rate*)

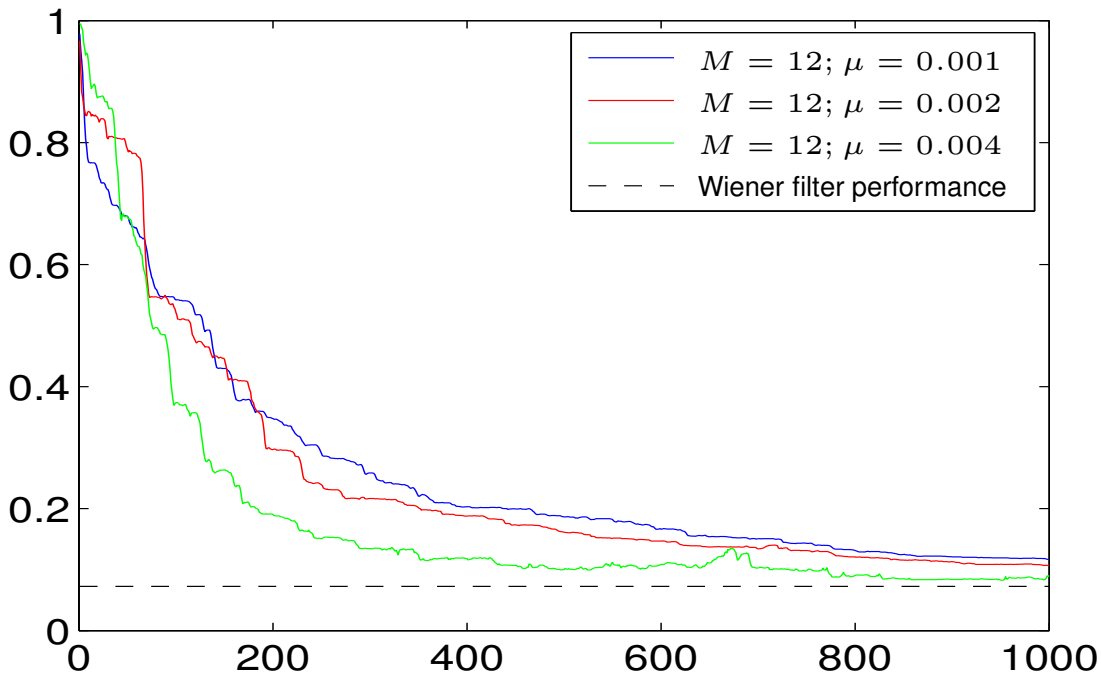
Example: LMS Filter:







$$\varepsilon_{\mathrm{MSE}}(\mathbf{w}) = \mathbf{w}^{\mathrm{T}}\mathbf{R}_y\mathbf{w} - 2\mathbf{w}^{\mathrm{T}}\mathbf{r}_{sy} + \varepsilon_s$$



Exercises

1. Show that the *linear* minimum mean-squared error (LMMSE) estimator $\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{y})$ of the random vector $\boldsymbol{\Theta}$ on the basis of the observation vector $\mathbf{Y} = \mathbf{y}$ fulfills

$$E\left(\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{Y}) - \boldsymbol{\Theta}\right) = \mathbf{0}$$

and

$$E\left(\left(\widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{Y}) - \boldsymbol{\Theta}\right)\mathbf{Y}^{\mathrm{T}}\right) = \mathbf{0}.$$

2. Assume a random parameter $\Theta \in \{-1, 1\}$ with $P(\Theta = -1) = P(\Theta = 1) = 0.5$, and an observation Y with the conditional probability densities

$$p(y \mid \Theta = -1) = \begin{cases} 1 & \text{if } y \in [-1, 0] \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad p(y \mid \Theta = 1) = \begin{cases} 1 & \text{if } y \in [0, 1] \\ 0 & \text{otherwise} \end{cases} .$$

   (a) Formulate the (non-linear) minimum mean-squared error (MMSE) estimator $\widehat{\theta}_{\mathrm{MMSE}}(y)$ of Θ, and quantify the MMSE $E\left(\left(\widehat{\theta}_{\mathrm{MMSE}}(Y) - \Theta\right)^2\right)$.

   (b) Formulate the LMMSE estimator $\widehat{\theta}_{\mathrm{LMMSE}}(y)$ of Θ and derive the mean-squared error $E\left(\left(\widehat{\theta}_{\mathrm{LMMSE}}(Y) - \Theta\right)^2\right)$.

   (c) Illustrate the principle of orthogonality by drawing Θ, Y, and $\widehat{\theta}_{\mathrm{LMMSE}}(Y)$ in the vector space of zero-mean random variables with inner product $\langle X, Y \rangle = E(XY)$.

3. In Kalman filter theory the process $\alpha_k = \mathbf{y}_k - \bar{\mathbf{y}}_k$, $k = 1, 2, \ldots$ is known as *innovations process*. Show that $\alpha_1, \alpha_2, \ldots$ has the following properties:

   (a)
   $$E\left(\alpha_k \mathbf{y}_n^{\mathrm{T}}\right) = \mathbf{0}, \qquad n = 1, \ldots, k-1$$

   (b)
   $$E\left(\alpha_k \alpha_n^{\mathrm{T}}\right) = \mathbf{0}, \qquad n = 1, \ldots, k-1.$$

Table of Contents

Non-Random Parameter Estimation

Recall the parameter estimation problem:

- unknown parameter $\theta \in \Lambda$ (where, for simplicity, we assume $\Lambda \subset \mathbb{R}$)
- observation $\mathbf{y}$ of a random vector $\mathbf{Y} \in \Gamma$
- find an estimator for θ in the form of a function

$$\widehat{\theta} \; : \; \Gamma \to \Lambda$$

  which maps every element $\mathbf{y}$ of the observation set Γ to a parameter $\widehat{\theta}(\mathbf{y})$

Concept of *non-random* parameter estimation:

- taking a *non-Bayesian* approach, the parameter θ is regarded as unknown *non-random* (unlike in e. g. MMSE estimation, where Θ is governed by some probability distribution)
- parameter estimation is based on the known probability density of $\mathbf{Y}$ under θ, which is assumed to exist and denoted here as $p(\mathbf{y} \mid \theta)$ even though θ is non-random

Estimator Properties

An estimator $\widehat{\theta}(\mathbf{y})$ is termed

- *unbiased* if
$$E\left(\widehat{\theta}(\mathbf{Y})\right) = \theta \quad \text{for all } \theta \in \Lambda$$
- a *minimum-variance unbiased estimator* (MVUE) if it is unbiased and minimizes the mean-squared error $E\left(\left(\widehat{\theta}(\mathbf{Y}) - \theta\right)^2\right)$ for each $\theta \in \Lambda$
- *efficient* if it is unbiased and achieves the *Cramér-Rao lower bound* for each $\theta \in \Lambda$ (every efficient estimator is an MVUE)

The Cramér-Rao lower bound is derived via the *log-likelihood function* $\log p(\mathbf{y} \mid \theta)$, which is in the following assumed to be a sufficiently smooth function w. r. t. to both $\mathbf{y}$ and θ.

Fisher Information and Cramér-Rao Lower Bound

Fisher information:

$$I_\theta = E\left(\left(\frac{\partial \log p(\mathbf{Y} \mid \theta)}{\partial \theta}\right)^2\right) = \int_\Gamma \left(\frac{\partial \log p(\mathbf{y} \mid \theta)}{\partial \theta}\right)^2 p(\mathbf{y} \mid \theta) \mathrm{d}\mathbf{y}$$

Remarks:

- Fisher information I_θ reflects the information about θ in $\mathbf{Y}$
- using integration by parts, one can show that

$$I_\theta = -E\left(\frac{\partial^2 \log p(\mathbf{Y} \mid \theta)}{\partial \theta^2}\right)$$

Cramér-Rao lower bound: for any unbiased estimator $\widehat{\theta}(\mathbf{y})$,

$$\mathrm{Var}\left(\widehat{\theta}(\mathbf{Y})\right) \geq \frac{1}{I_\theta}$$

Cramér-Rao Lower Bound

Proof:

$$\frac{\partial}{\partial\theta}E\left(\widehat{\theta}(\mathbf{Y})\right) = \int_{\Gamma}\widehat{\theta}(\mathbf{y})\frac{\partial p(\mathbf{y}\mid\theta)}{\partial\theta}\mathrm{d}\mathbf{y}$$

$$= \int_{\Gamma}\left(\widehat{\theta}(\mathbf{y}) - E\left(\widehat{\theta}(\mathbf{Y})\right)\right)\frac{\partial p(\mathbf{y}\mid\theta)}{\partial\theta}\mathrm{d}\mathbf{y}$$

$$\left(\text{since } E\left(\widehat{\theta}(\mathbf{Y})\right) \text{ does not depend on } \mathbf{y} \text{ and } \int_{\Gamma}\frac{\partial p(\mathbf{y}\mid\theta)}{\partial\theta}\mathrm{d}\mathbf{y} = \frac{\partial}{\partial\theta}\int_{\Gamma}p(\mathbf{y}\mid\theta)\mathrm{d}\mathbf{y} = 0\right)$$

$$= \int_{\Gamma}\left(\widehat{\theta}(\mathbf{y}) - E\left(\widehat{\theta}(\mathbf{Y})\right)\right)\frac{\partial \log p(\mathbf{y}\mid\theta)}{\partial\theta}p(\mathbf{y}\mid\theta)\mathrm{d}\mathbf{y}$$

$$= E\left(\left(\widehat{\theta}(\mathbf{Y}) - E\left(\widehat{\theta}(\mathbf{Y})\right)\right)\frac{\partial \log p(\mathbf{Y}\mid\theta)}{\partial\theta}\right)$$

$$\leq \sqrt{E\left(\left(\widehat{\theta}(\mathbf{Y}) - E\left(\widehat{\theta}(\mathbf{Y})\right)\right)^2\right)E\left(\left(\frac{\partial \log p(\mathbf{Y}\mid\theta)}{\partial\theta}\right)^2\right)} = \sqrt{\mathrm{Var}\left(\widehat{\theta}(\mathbf{Y})\right)I_\theta}$$

↑ Cauchy-Schwarz inequality

$$E\left(\widehat{\theta}(\mathbf{Y})\right) = \theta \quad\rightarrow\quad \mathrm{Var}\left(\widehat{\theta}(\mathbf{Y})\right) \geq \frac{1}{I_\theta}$$

Maximum-Likelihood Estimation

Maximum-likelihood (ML) estimator:

$$\widehat{\theta}_{\mathrm{ML}}(\mathbf{y}) = \arg\max_{\theta\in\Lambda} p(\mathbf{y} \mid \theta)$$

Remarks:

- $\widehat{\theta}_{\mathrm{ML}}(\mathbf{y})$ is a non-random (non-Bayesian) parameter estimator
- $p(\mathbf{y} \mid \theta)$ as a function of θ is known as the *likelihood function*
- $\widehat{\theta}_{\mathrm{ML}}(\mathbf{y})$ also maximizes the log-likelihood function $\log p(\mathbf{y} \mid \theta)$ due to monotonicity of the logarithm
- if $\widehat{\theta}_{\mathrm{ML}}(\mathbf{y})$ exists and $\partial \log p(\mathbf{y} \mid \theta)/\partial\theta$ is a continuous function of θ, then $\widehat{\theta}_{\mathrm{ML}}(\mathbf{y})$ solves the *likelihood equation*

$$\frac{\partial \log p(\mathbf{y} \mid \theta)}{\partial\theta} = 0$$

- any estimator $\widehat{\theta}(\mathbf{y})$ achieving the Cramér-Rao lower bound is a solution of the likelihood equation
- a solution of the likelihood equation can be an MVUE

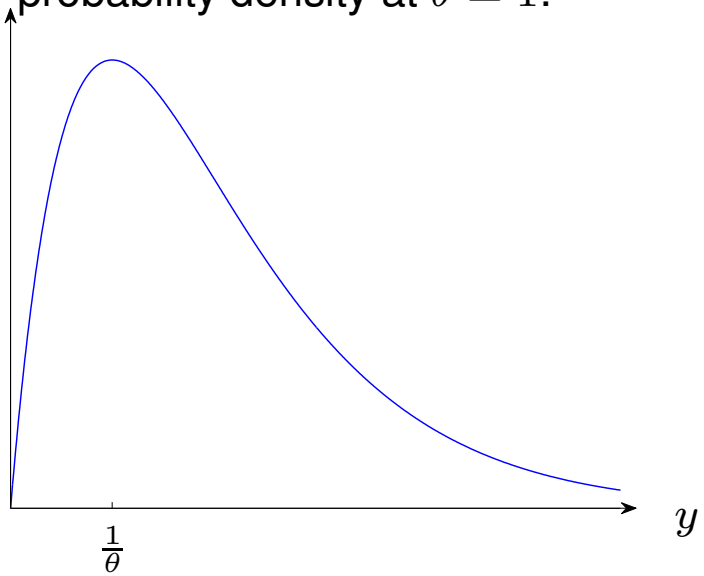
Example: Erlang Distributed Observation

- suppose

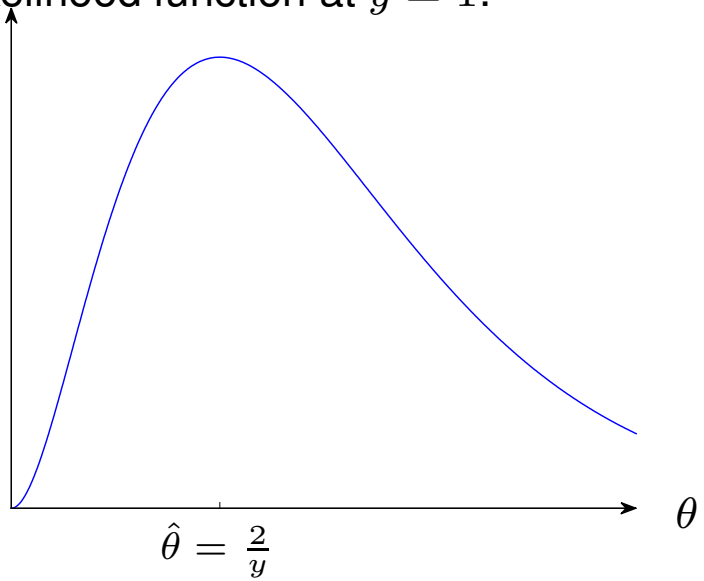
$$p(y \mid \theta) = \theta^2 y e^{-\theta y}, \qquad y, \theta \geq 0$$

(Erlang-2 distribution with *rate parameter* θ)

- probability density at $\theta = 1$:



likelihood function at $y = 1$:



- log-likelihood function:

$$\log p(y \mid \theta) = 2 \log \theta + \log y - \theta y, \qquad y, \theta \geq 0$$

Example: Erlang Distributed Observation (cont.)

- from the likelihood equation

$$\frac{\partial \log p(y \mid \theta)}{\partial \theta} = \frac{2}{\theta} - y = 0$$

we obtain the estimator

$$\widehat{\theta}_{\mathrm{ML}}(y) = \frac{2}{y}$$

- since

$$E\left(\widehat{\theta}_{\mathrm{ML}}(Y)\right) = \int_0^\infty \frac{2}{y}\theta^2 y e^{-\theta y}\mathrm{d}y = 2\theta$$

the ML estimator is biased

Observation of Independent Samples

The maximum-likelihood approach is particularly efficient for the model parameter estimation on the basis of a large number of independent observations.

Given the entries $Y_1, \ldots, Y_L$ of the random vector $\mathbf{Y}$ are independent and identically distributed (i. i. d.) depending on the parameter θ:

- $$p_{\mathbf{Y}}(\mathbf{y} \mid \theta) = \prod_{j=1}^{L} p_{\mathbf{Y}}(y_j \mid \theta) \quad \text{with } \mathbf{y} = [y_1 \ \cdots \ y_L]^{\mathrm{T}}$$
- $$\log p_{\mathbf{Y}}(\mathbf{y} \mid \theta) = \sum_{j=1}^{L} \log p_{\mathbf{Y}}(y_j \mid \theta)$$
- the Fisher information increases linearly with L, i.e.,

$$I_\theta = E\left(\left(\frac{\partial \log p_{\mathbf{Y}}(\mathbf{Y} \mid \theta)}{\partial \theta}\right)^2\right) = LE\left(\left(\frac{\partial \log y_Y(Y_j \mid \theta)}{\partial \theta}\right)^2\right)$$

Independent Samples: Asymptotic Properties

For i. i. d. observations the following asymptotic properties hold under certain conditions that are met in a large class of problems:

- $\widehat{\theta}_{\mathrm{ML}}(\mathbf{y})$ is asymptotically unbiased, that is, $E\left(\widehat{\theta}_{\mathrm{ML}}(\mathbf{Y}) - \theta\right)$ tends to zero as $L \to \infty$
- $\widehat{\theta}_{\mathrm{ML}}(\mathbf{y})$ is asymptotically efficient, that is, it asymptotically achieves the Cramér-Rao lower bound
- the distribution of the normalized estimation error $\sqrt{L}\left(\widehat{\theta}_{\mathrm{ML}}(\mathbf{Y}) - \theta\right)$ tends to the Gaussian distribution $\mathcal{N}(0, LI_{\theta}^{-1})$ as $L \to \infty$

Example: Estimation of Mean, Variance of Gaussian Distributed Observations

Suppose $Y_1, \ldots, Y_L$ are i. i. d. subject to $\mathcal{N}(\mu, v)$ and either μ or v to be estimated:

- the log-likelihood functions w. r. t. μ and w. r. t. v read

$$\log p_{\mathbf{Y}}(\mathbf{y} \mid \mu) = \sum_{j=1}^{L} \log p_Y(y_j \mid \mu) = -\frac{L}{2} \log(2\pi v) - \frac{1}{2v} \sum_{j=1}^{L} (y_j - \mu)^2,$$

$$\log p_{\mathbf{Y}}(\mathbf{y} \mid v) = \sum_{j=1}^{L} \log p_Y(y_j \mid v) = -\frac{L}{2} \log(2\pi v) - \frac{1}{2v} \sum_{j=1}^{L} (y_j - \mu)^2$$

- from

$$\frac{\partial \log p_{\mathbf{Y}}(\mathbf{y} \mid \mu)}{\partial \mu} = \frac{1}{v} \sum_{j=1}^{L} (v_j - \mu) = 0 \quad \text{and} \quad \frac{\partial \log p_{\mathbf{Y}}(\mathbf{y} \mid v)}{\partial v} = -\frac{L}{2v} + \frac{1}{2v^2} \sum_{j=1}^{L} (y_j - \mu)^2 = 0$$

  we obtain

$$\hat{\mu}_{\mathrm{ML}}(\mathbf{y}) = \frac{1}{L} \sum_{j=1}^{L} y_j \quad \text{and} \quad \hat{v}_{\mathrm{ML}}(\mathbf{y}) = \frac{1}{L} \sum_{j=1}^{L} (y_j - \mu)^2$$

Example: Estimation of Mean, Variance of Gaussian Distributed Observations (cont.)

- the estimators $\hat{\mu}_{\mathrm{ML}}(\mathbf{y})$ and $\hat{v}_{\mathrm{ML}}(\mathbf{y})$ are unbiased since

$$E\left(\hat{\mu}_{\mathrm{ML}}(\mathbf{Y})\right) = \mu \quad \text{and} \quad E\left(\hat{v}_{\mathrm{ML}}(\mathbf{Y})\right) = v$$

- for the variance of the estimators we find

$$\mathrm{Var}\left(\hat{\mu}_{\mathrm{ML}}(\mathbf{Y})\right) = E\left(\left(\hat{\mu}_{\mathrm{ML}}(\mathbf{Y}) - \mu\right)^2\right) = \frac{v}{L}$$

and

$$\mathrm{Var}\left(\hat{\mu}_{\mathrm{ML}}(\mathbf{Y})\right) = E\left(\left(\hat{v}_{\mathrm{ML}}(\mathbf{Y}) - v\right)^2\right) = \frac{2v^2}{L}$$

- the information about μ and v in $\mathbf{Y}$ is given by

$$I_\mu = E\left(\left(\frac{\partial \log p_{\mathbf{Y}}(\mathbf{y} \mid \mu)}{\partial \mu}\right)^2\right) = \frac{L}{v} \quad \text{and} \quad I_v = E\left(\left(\frac{\partial \log p_{\mathbf{Y}}(\mathbf{y} \mid v)}{\partial v}\right)^2\right) = \frac{L}{2v^2}$$

- since $\mathrm{Var}\left(\hat{\mu}_{\mathrm{ML}}(\mathbf{Y})\right) = I_\mu^{-1}$ and $\mathrm{Var}\left(\hat{v}_{\mathrm{ML}}(\mathbf{Y})\right) = I_v^{-1}$, both estimators are efficient

Extension to Multivariate Gaussian Distributed Observations

Suppose $\mathbf{Y}_1, \ldots, \mathbf{Y}_L$ are i. i. d. random *vectors* subject to $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$:

- the corresponding log-likelihood functions w. r. t. $\mathbf{m}$ and w. r. t. $\mathbf{\Sigma}$ read

$$\log p(\mathbf{y}_1, \ldots, \mathbf{y}_L \mid \mathbf{m}) = -\frac{LM}{2} \log(2\pi) - \frac{L}{2} \log\det \mathbf{\Sigma} - \frac{1}{2} \sum_{j=1}^{L} (\mathbf{y}_j - \mathbf{m})^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{y}_j - \mathbf{m}),$$

$$\log p(\mathbf{y}_1, \ldots, \mathbf{y}_L \mid \mathbf{\Sigma}) = -\frac{LM}{2} \log(2\pi) - \frac{L}{2} \log\det \mathbf{\Sigma} - \frac{1}{2} \sum_{j=1}^{L} (\mathbf{y}_j - \mathbf{m})^{\mathrm{T}} \mathbf{\Sigma}^{-1} (\mathbf{y}_j - \mathbf{m})$$

- it can be shown that the ML estimators for $\mathbf{m}$ and $\mathbf{\Sigma}$ are given by

$$\widehat{\mathbf{m}}_{\mathrm{ML}}(\mathbf{y}_1, \ldots, \mathbf{y}_L) = \frac{1}{L} \sum_{j=1}^{L} \mathbf{y}_j \quad \text{and} \quad \widehat{\mathbf{\Sigma}}_{\mathrm{ML}}(\mathbf{y}_1, \ldots, \mathbf{y}_L) = \frac{1}{L} \sum_{j=1}^{L} (\mathbf{y}_j - \mathbf{m})(\mathbf{y}_j - \mathbf{m})^{\mathrm{T}}$$

Excursus: Maximum A Posteriori Probability Estimation

If we know the prior probability density $p(\theta)$ for the parameter vector $\boldsymbol{\Theta}$, we may be able to express the *a posteriori* probability density $p(\theta \mid \mathbf{y})$ using Bayes' theorem:

$$p(\theta \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \theta)p(\theta)}{p(\mathbf{y})}$$

Maximum a posteriori probability (MAP) estimation concept:

- Bayesian approach aiming at the parameter vector which maximizes $p(\theta \mid \mathbf{y})$
- the MAP estimator is given by

$$\widehat{\theta}_{\mathrm{MAP}}(\mathbf{y}) = \arg\max_{\theta \in \Lambda} p(\theta \mid \mathbf{y})$$

- there is usually no need to compute $p(\mathbf{y})$ for solving the maximization problem
- MAP-equation for eligible $p(\mathbf{y} \mid \theta)$ and $p(\theta)$:

$$\frac{\partial}{\partial \theta} \log p(\mathbf{y} \mid \theta)\Big|_{\theta = \widehat{\theta}_{\mathrm{MAP}}(\mathbf{y})} = -\frac{\partial}{\partial \theta} \log p(\theta)\Big|_{\theta = \widehat{\theta}_{\mathrm{MAP}}(\mathbf{y})}$$

Jointly Gaussian Case

If $\boldsymbol{\Theta}$ and $\mathbf{Y}$ are *jointly Gaussian* random vectors, that is, if

$$\begin{bmatrix} \boldsymbol{\Theta} \\ \mathbf{Y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}_{\boldsymbol{\Theta}} \\ \mathbf{m}_{\mathbf{Y}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{V}} & \boldsymbol{\Sigma}_{\boldsymbol{\Theta}\mathbf{Y}} \\ \boldsymbol{\Sigma}_{\boldsymbol{\Theta}\mathbf{Y}}^{\mathrm{T}} & \boldsymbol{\Sigma}_{\mathbf{Y}} \end{bmatrix} \right),$$

the conditional (a posteriori) probability distribution of $\boldsymbol{\Theta}$ given $(\mathbf{Y} = \mathbf{y})$ becomes (see #21)

$$\mathcal{N}\left(\boldsymbol{\mu}(\mathbf{y}), \boldsymbol{\Sigma}_{\mathbf{V}} - \boldsymbol{\Sigma}_{\boldsymbol{\Theta}\mathbf{Y}} \boldsymbol{\Sigma}_{\mathbf{Y}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{\Theta}\mathbf{Y}}^{\mathrm{T}} \right) \quad \text{with} \quad \boldsymbol{\mu}(\mathbf{y}) = \mathbf{m}_{\boldsymbol{\Theta}} + \boldsymbol{\Sigma}_{\boldsymbol{\Theta}\mathbf{Y}} \boldsymbol{\Sigma}_{\mathbf{Y}}^{-1} \left(\mathbf{y} - \mathbf{m}_{\mathbf{Y}} \right).$$

It follows that

$$\begin{aligned} \widehat{\theta}_{\mathrm{MAP}}(\mathbf{y}) &= \boldsymbol{\mu}(\mathbf{y}) && \text{since at the mean value the density function has its maximum} \\ &= E\left(\boldsymbol{\Theta} \,\middle|\, \mathbf{Y} = \mathbf{y} \right) \\ &= \widehat{\theta}_{\mathrm{MMSE}}(\mathbf{y}) && \text{according to the definition of } \widehat{\theta}_{\mathrm{MMSE}}(\mathbf{y}), \text{ see \#149} \\ &= \widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{y}) && \text{since } \boldsymbol{\mu}(\mathbf{y}) \text{ is an affine function} \end{aligned}$$

The Linear Gaussian Model

Recall the linear model $\mathbf{Y} = \mathbf{H\Theta} + \mathbf{V}$, and suppose that

- $\mathbf{\Theta} \sim \mathcal{N}(\mathbf{m_\Theta}, \mathbf{\Sigma_V})$
- $\mathbf{H}$ is a deterministic matrix
- $\mathbf{V} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma_V})$
- $\mathbf{\Theta}$ and $\mathbf{V}$ are independent

Since

$$\begin{bmatrix} \mathbf{\Theta} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{H} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{\Theta} \\ \mathbf{V} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m_\Theta} \\ \mathbf{H m_\Theta} \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma_V} & \mathbf{\Sigma_V H}^{\mathrm{T}} \\ \mathbf{H\Sigma_V} & \mathbf{H\Sigma_V H}^{\mathrm{T}} + \mathbf{\Sigma_V} \end{bmatrix} \right)$$

the MAP estimator results as

$$\widehat{\theta}_{\mathrm{MAP}}(\mathbf{y}) = \mathbf{m_\Theta} + \mathbf{\Sigma_V H}^{\mathrm{T}} \left(\mathbf{H\Sigma_V H}^{\mathrm{T}} + \mathbf{\Sigma_V} \right)^{-1} (\mathbf{y} - \mathbf{H m_\Theta})$$

Obviously,

- $\widehat{\theta}_{\mathrm{MAP}}(\mathbf{y}) = \widehat{\theta}_{\mathrm{LMMSE}}(\mathbf{y})$, see #152
- in linear Gaussian models (including e. g. the state space model underlying the Kalman filter with Gaussian $\mathbf{v}_1, \mathbf{v}_2, \ldots$ and $\mathbf{q}_1, \mathbf{q}_2, \ldots$) the LMMSE and MAP estimates coincide

The Gaussian Mixture Model

The (multivariate) Gaussian model is often too simple and thus not adequate for modeling a certain physical phenomena.

More versatile and often adopted is the *Gaussian mixture model*, which builds on the probability density

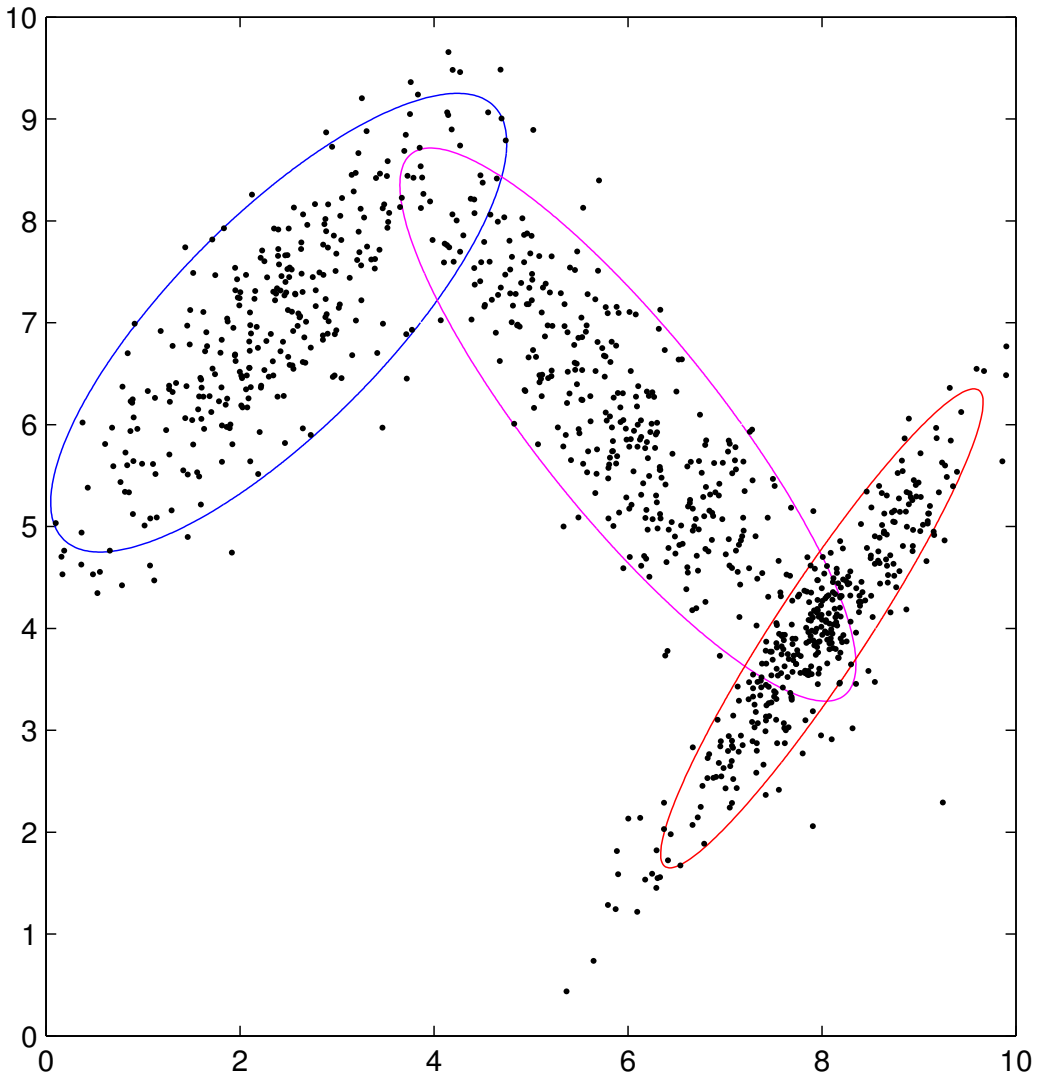
$$p_{\mathbf{Y}}(\mathbf{y} \mid \mathbf{m}_1, \ldots, \mathbf{m}_N, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_N, \pi_1, \ldots, \pi_N) = \sum_{n=1}^{N} \frac{\pi_n}{(2\pi)^{K/2}(\det \mathbf{\Sigma}_n)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{m}_n)^{\mathrm{T}} \mathbf{\Sigma}_n^{-1}(\mathbf{y}-\mathbf{m}_n)\right)$$

Model parameters in case of N clusters:

- mixture weights $\pi_1, \ldots, \pi_N$, where $0 \leq \pi_n \leq 1$ and $\sum_{n=1}^{N} \pi_n = 1$
- cluster mean vectors $\mathbf{m}_1, \ldots, \mathbf{m}_N$
- cluster covariance matrices $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_N$

Example

1000 random points drawn from a 2-D mixed Gaussian distribution with 3 clusters



- $\pi_1 = \pi_2 = \pi_3 = \frac{1}{3}$
- Gaussian distributions underlying the three clusters:

$$\mathcal{N}\left(\begin{bmatrix} 2.4 \\ 7.0 \end{bmatrix}, \begin{bmatrix} 1.2 & 0.9 \\ 0.9 & 1.1 \end{bmatrix}\right),$$

$$\mathcal{N}\left(\begin{bmatrix} 6 \\ 6 \end{bmatrix}, \begin{bmatrix} 1.2 & -1.2 \\ -1.2 & 1.6 \end{bmatrix}\right),$$

$$\mathcal{N}\left(\begin{bmatrix} 8.0 \\ 4.0 \end{bmatrix}, \begin{bmatrix} 0.6 & 0.8 \\ 0.8 & 1.2 \end{bmatrix}\right)$$

- 10%-contours of the Gaussian distributions are plotted

ML Estimation of Model Parameters

Model parameter estimation:

- parameter set $\theta = \{\mathbf{m}_1, \ldots, \mathbf{m}_N, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_N, \pi_1, \ldots, \pi_N\}$
- L i. i. d. K-dimensional observations $\mathbf{y}_1, \ldots, \mathbf{y}_L$
- log-likelihood function

$$\log p(\mathbf{y}_1, \ldots, \mathbf{y}_L \mid \theta) = \sum_{\ell=1}^{L} \log \sum_{n=1}^{N} \frac{\pi_n}{(2\pi)^{K/2} (\det \mathbf{\Sigma}_n)^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{y}_\ell - \mathbf{m}_n)^{\mathrm{T}} \mathbf{\Sigma}_n^{-1} (\mathbf{y}_\ell - \mathbf{m}_n) \right)$$

- ML estimate

$$\widehat{\theta}_{\mathrm{ML}} = \arg \max_{\theta} \log p(\mathbf{y}_1, \ldots, \mathbf{y}_L \mid \theta)$$

Analytical maximization of the log-likelihood function is not feasible.

Expectation-Maximization Algorithm

- Very often, direct maximization of the log-likelihood function can be replaced by an iterative Algorithmus with substantially lower complexity, where in each Integration the log-likelihood increases (no proof given here), but convergence of the scheme cannot be guaranteed for arbitrary initialization of the scheme.
- The scheme is known as Expectation-Maximization Algorithm (EM algorithm).
- Assume data $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_L\}$ being observed and generated by some distribution where $\mathcal{Y}$ is called incomplete data. We assume that a complete data set exists given by $\mathcal{X} = (\mathcal{Y}, \mathcal{Z})$ and a density $p(\mathbf{x}|\theta) = p(\mathbf{y}, \mathbf{z}|\theta) = p(\mathbf{z}|\mathbf{y}, \theta)\, p(\mathbf{y}|\theta)$, where $\mathcal{Z}$ is unobservable and thus called *hidden data*.
- With this assignment where $\mathcal{Z}$ is random and yet unknown, we can define a new likelihood function by

$$\mathcal{L}(\mathcal{X}|\theta) = \mathcal{L}(\mathcal{Y}, \mathcal{Z}|\theta) = p(\mathcal{Y}, \mathcal{Z}|\theta)$$

  called the complete-data likelihood.
- Clearly, $p(\mathcal{Y}, \mathcal{Z}|\theta) = h_{\mathcal{Y},\theta}(\mathcal{Z})$ is a random variable for some function $h_{\mathcal{Y},\theta}(\cdot)$ since it depends on the constants $\mathcal{Y}$ and θ as well as the random $\mathcal{Z}$.
- The choice of $\mathcal{Z}$ is specific to the problem at hand for a given application.
- To formulate the algorithm, denote by $\theta^{(i)}$ the estimate of θ in the ith algorithm iteration.

- The EM algorithm finds the expected value (E-step) of the complete-data log-likelihood $\log p\left(\mathcal{Y}, \mathcal{Z} \middle| \theta\right)$ with respect to the unknown data $\mathcal{Z}$ given the observed data $\mathcal{Y}$ and the current parameter estimates. Then, the found function E-step

$$Q\left(\theta, \theta^{(i)}\right) = E\left(\log p\left(\mathcal{Y}, \mathcal{Z} \middle| \theta\right) \middle| \mathcal{Y}, \theta^{(i)}\right) = \int_{\mathbf{z} \in \Upsilon} \log p\left(\mathcal{Y}, \mathbf{z} \middle| \theta\right) p\left(\mathbf{z} \middle| \mathcal{Y}, \theta^{(i)}\right) \mathrm{d}\mathbf{z} \quad \text{(E)}$$

  is maximized with respect to θ in the maximization step (M-step)

$$\theta^{(i+1)} = \arg\max_{\theta} Q\left(\theta, \theta^{(i)}\right). \quad \text{(M)}$$

- The resulting scheme can be shown to provide monotonically increasing values of the original log-likelihood function $\Lambda\left(\mathcal{Y} \middle| \theta^{(i)}\right) = \Lambda\left(\mathbf{y} \middle| \theta^{(i)}\right)$.

- A modified form of the M-step is to find a value $\theta^{(i+1)}$ such that, instead of finding the maximum of $Q\left(\theta, \theta^{(i)}\right)$, we have only an increase of $Q\left(\theta, \theta^{(i)}\right)$ for the given parameter estimate $\theta^{(i)}$, that is

$$Q\left(\theta^{(i+1)}, \theta^{(i)}\right) > Q\left(\theta^{(i)}, \theta^{(i)}\right).$$

  This form of the algorithm is called *Generalized EM* (GEM) and is also guaranteed to converge.

Expectation-Maximization Algorithm for the Gaussian Mixture Model

Introduction of L independent discrete random variables $Z_1, \ldots, Z_L$:

- random variable $Z_\ell \in \{1, \ldots, N\}$ selects the cluster to which the ℓth point belongs
- the conditional probability density of the ℓth observation given $(Z_\ell = n)$ reads

  $$p(\mathbf{y}_\ell \mid Z_\ell = n, \theta) = \frac{1}{(2\pi)^{K/2}(\det \mathbf{\Sigma}_n)^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{y}_\ell - \mathbf{m}_n)^{\mathrm{T}} \mathbf{\Sigma}_n^{-1} (\mathbf{y}_\ell - \mathbf{m}_n)\right)$$

  with prior probability distribution of Z_ℓ given by $P(Z_\ell = n) = \pi_n$ for $n = 1, \ldots, N$
- the marginal distribution $p(\mathbf{y}_\ell \mid \theta)$ thus conforms with the Gaussian mixture model
- the unobservable random variables $Z_1, \ldots, Z_L$ are the hidden data or *latent* variables
- note that the integral in (E) becomes a sum due to the discrete nature of $\Upsilon \ni \mathbf{z}$
- for calculating $Q\left(\theta, \theta^{(i)}\right)$, we need the conditional distribution

  $$p\left(\mathbf{z} | \mathcal{Y}, \theta^{(i)}\right) = \prod_{\ell=1}^{L} p\left(z_\ell | \mathbf{y}_\ell, \theta^{(i)}\right) = \prod_{\ell=1}^{L} \sum_{n=1}^{N} P\left(Z_\ell = n | \mathbf{y}_\ell, \theta^{(i)}\right) \delta(z_\ell - n)$$

  with $\gamma_{\ell,n}(\theta) = P(Z_\ell = n \mid \mathbf{y}_\ell, \theta) = \dfrac{\pi_n p(\mathbf{y}_\ell \mid Z_\ell = n, \theta)}{\sum_{j=1}^{N} \pi_j p(\mathbf{y}_\ell \mid Z_\ell = j, \theta)}$ as follows from

Expectation-Maximization Algorithm for the Gaussian Mixture Model (cont.)

- Furthermore, $Q\left(\theta, \theta^{(i)}\right)$ requires the joint density $p\left(\mathcal{Y}, \mathbf{z}|\,\theta\right)$ which in view of the i.i.d. observations $\mathbf{y}_\ell$ is given by

$$p\left(\mathcal{Y}, \mathbf{z}|\,\theta\right) = \prod_{\ell=1}^{L} p\left(\mathbf{y}_\ell, z_\ell|\,\theta\right).$$

With $\Upsilon = \Upsilon_1 \times \ldots \times \Upsilon_L$ (E) takes the following form:

$$\begin{aligned} Q\left(\theta, \theta^{(i)}\right) &= E\left(\log p\left(\mathcal{Y}, \mathcal{Z}|\,\theta\right)|\,\mathcal{Y}, \theta^{(i)}\right) = \int_{\mathbf{z}\in\Upsilon} \log p\left(\mathcal{Y}, \mathbf{z}|\,\theta\right) p\left(\mathbf{z}|\,\mathcal{Y}, \theta^{(i)}\right) \mathrm{d}\mathbf{z} \\ &= \int_{z_1\in\Upsilon_1} \cdots \int_{z_L\in\Upsilon_L} \sum_{\ell=1}^{L} \log\left(p\left(\mathbf{y}_\ell, z_\ell|\,\theta\right)\right) \prod_{\lambda=1}^{L} \sum_{n=1}^{N} \gamma_{\lambda,n}\left(\theta^{(i)}\right) \delta(z_\lambda - n)\, \mathrm{d}z_1 \ldots \mathrm{d}z_L. \end{aligned}$$

Since $\displaystyle\prod_{\lambda=1}^{L} \sum_{n=1}^{N} \gamma_{\lambda,n}\left(\theta^{(i)}\right) \delta(z_\lambda - n) = \sum_{n_1=1}^{N} \cdots \sum_{n_L=1}^{N} \gamma_{1,n_1}\left(\theta^{(i)}\right) \cdots$

$$\gamma_{L,n_L}\left(\theta^{(i)}\right) \delta(z_1 - n_1) \cdots \delta(z_L - n_L)$$

$$= \sum_{n_1=1}^{N} \cdots \sum_{n_L=1}^{N} \prod_{\lambda=1}^{L} \gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right) \delta(z_\lambda - n_\lambda),$$

we have with to the sifting property of the Dirac delta function $\delta(\cdot)$

Expectation-Maximization Algorithm for the Gaussian Mixture Model (cont.)

$$Q\left(\theta,\theta^{(i)}\right)=\sum_{n_1=1}^{N}\cdots\sum_{n_L=1}^{N}\int_{z_1\in\Upsilon_1}\cdots\int_{z_L\in\Upsilon_L}\sum_{\ell=1}^{L}\log\left(p\left(\mathbf{y}_\ell,z_\ell|\,\theta\right)\right)\prod_{\lambda=1}^{L}\gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right)\delta(z_\lambda-n_\lambda)\,\mathrm{d}z_1$$
$$\ldots\mathrm{d}z_L$$
$$=\sum_{n_1=1}^{N}\cdots\sum_{n_L=1}^{N}\sum_{\ell=1}^{L}\log\left(p\left(\mathbf{y}_\ell,z_\ell=n_\ell|\,\theta\right)\right)\prod_{\lambda=1}^{L}\gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right)$$
$$=\sum_{n_1=1}^{N}\cdots\sum_{n_L=1}^{N}\sum_{\ell=1}^{L}\log\left(\pi_{n_\ell}p\left(\mathbf{y}_\ell|\,z_\ell=n_\ell,\theta\right)\right)\prod_{\lambda=1}^{L}\gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right).$$

With the Kronecker delta δ_ℓ, we can write

$$Q\left(\theta,\theta^{(i)}\right)=\sum_{n_1=1}^{N}\cdots\sum_{n_L=1}^{N}\sum_{\ell=1}^{L}\sum_{n=1}^{N}\delta_{n-n_\ell}\log\left(\pi_n p\left(\mathbf{y}_\ell|\,z_\ell=n,\theta\right)\right)\prod_{\lambda=1}^{L}\gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right)$$
$$=\sum_{n=1}^{N}\sum_{\ell=1}^{L}\log\left(\pi_n p\left(\mathbf{y}_\ell|\,z_\ell=n,\theta\right)\right)\sum_{n_1=1}^{N}\cdots\sum_{n_L=1}^{N}\delta_{n-n_\ell}\prod_{\lambda=1}^{L}\gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right).$$

Expectation-Maximization Algorithm for the Gaussian Mixture Model (cont.)

Although the right-hand side of the equation looks complicated, it can be greatly simplified:

$$\begin{aligned}
& \sum_{n_1=1}^{N} \cdots \sum_{n_L=1}^{N} \delta_{n-n_\ell} \prod_{\lambda=1}^{L} \gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right) \\
= & \left(\sum_{n_1=1}^{N} \cdots \sum_{n_{\ell-1}=1}^{N} \sum_{n_{\ell+1}=1}^{N} \cdots \sum_{n_L=1}^{N} \prod_{\lambda=1,\lambda\neq\ell}^{L} \gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right) \right) \gamma_{\ell,n}\left(\theta^{(i)}\right) \\
= & \prod_{\lambda=1,\lambda\neq\ell}^{L} \left(\sum_{n_\lambda=1}^{N} \gamma_{\lambda,n_\lambda}\left(\theta^{(i)}\right) \right) \gamma_{\ell,n}\left(\theta^{(i)}\right) \\
= & \gamma_{\ell,n}\left(\theta^{(i)}\right)
\end{aligned}$$

since $\sum_{n=1}^{N} \gamma_{\ell,n}\left(\theta^{(i)}\right) = 1$ for $\ell \in \{1,\ldots,L\}$. Thus, we finally obtain

$$\begin{aligned}
Q\left(\theta,\theta^{(i)}\right) &= \sum_{n=1}^{N}\sum_{\ell=1}^{L} \log\left(\pi_n p\left(\mathbf{y}_\ell \middle| z_\ell = n,\theta\right)\right) \gamma_{\ell,n}\left(\theta^{(i)}\right) \\
&= \sum_{n=1}^{N}\sum_{\ell=1}^{L} \log\left(\pi_n\right) \gamma_{\ell,n}\left(\theta^{(i)}\right) + \sum_{n=1}^{N}\sum_{\ell=1}^{L} \log\left(p\left(\mathbf{y}_\ell \middle| z_\ell = n,\theta\right)\right) \gamma_{\ell,n}\left(\theta^{(i)}\right).
\end{aligned}$$

Expectation-Maximization Algorithm for the Gaussian Mixture Model (cont.)

As can be seen, the maximization step (M) can now be implemented with considerably reduced complexity in comparison with the original problem (cf. above) given by

$$\widehat{\theta}_{\mathrm{ML}} = \arg\max_{\theta} \log p(\mathbf{y}_1, \ldots, \mathbf{y}_L \mid \theta)$$

$$= \arg\max_{\theta} \log \sum_{\ell=1}^{L} \log \sum_{n=1}^{N} \frac{\pi_n}{(2\pi)^{K/2} (\det \mathbf{\Sigma}_n)^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{y}_\ell - \mathbf{m}_n)^{\mathrm{T}} \mathbf{\Sigma}_n^{-1} (\mathbf{y}_\ell - \mathbf{m}_n) \right).$$

Instead, we can carry out the maximization in (M) (cf. above) given by $\theta^{(i+1)} = \arg\max_\theta Q\left(\theta, \theta^{(i)}\right)$, where we consider as an example the updating of the estimate $\pi_n^{(i)}$. Clearly, in maximizing $Q\left(\theta, \theta^{(i)}\right)$ with respect to π_n, we have to consider the constraint $\sum_{n=1}^{N} \pi_n = 1$ in each iteration. Thus, considering a Lagrange multiplier λ, our objective is to maximize the function

$$Q\left(\theta, \theta^{(i)}\right) + \lambda \left(\sum_{n=1}^{N} \pi_n - 1 \right)$$

$$= \sum_{n=1}^{N} \sum_{\ell=1}^{L} \log\left(\pi_n\right) \gamma_{\ell,n}\left(\theta^{(i)}\right) + \sum_{n=1}^{N} \sum_{\ell=1}^{L} \log\left(p\left(\mathbf{y}_\ell \middle| z_\ell = n, \theta\right)\right) \gamma_{\ell,n}\left(\theta^{(i)}\right) + \lambda \left(\sum_{n=1}^{N} \pi_n - 1 \right)$$

with respect to all π_n.

The Gaussian Mixture Model: Iterative Likelihood Maximization

Setting the partial derivative of the objective function w.r.t. π_n equal to zero, we obtain

$$\frac{1}{\pi_n} \sum_{\ell=1}^{L} \gamma_{\ell,n} \left(\theta^{(i)} \right) + \lambda = 0$$

or

$$\pi_n = - \frac{\sum_{\ell=1}^{L} \gamma_{\ell,n} \left(\theta^{(i)} \right)}{\lambda}.$$

After summing over all n and taking into account $\sum_{n=1}^{N} \gamma_{\ell,n} \left(\theta^{(i)} \right) = 1$ and $\sum_{n=1}^{N} \pi_n = 1$, we obtain

$$\sum_{n=1}^{N} \pi_n = - \frac{\sum_{\ell=1}^{L} \sum_{n=1}^{N} \gamma_{\ell,n} \left(\theta^{(i)} \right)}{\lambda} = - \frac{\sum_{\ell=1}^{L} 1}{\lambda} = - \frac{L}{\lambda} \stackrel{!}{=} 1.$$

Thus, $\lambda = -L$ and we obtain

$$\pi_n^{(i+1)} = \frac{\sum_{\ell=1}^{L} \gamma_{\ell,n} \left(\theta^{(i)} \right)}{L}. \tag{M.1}$$

The Gaussian Mixture Model: Iterative Likelihood Maximization (cont.)

After similar algebra taking into account the Gaussian mixture components, we obtain

$$\mathbf{m}_n^{(i+1)} = \frac{\sum_{\ell=1}^{L} \gamma_{\ell,n}\left(\theta^{(i)}\right) \mathbf{y}_\ell}{\sum_{\ell=1}^{L} \gamma_{\ell,n}\left(\theta^{(i)}\right)}. \tag{M.2}$$

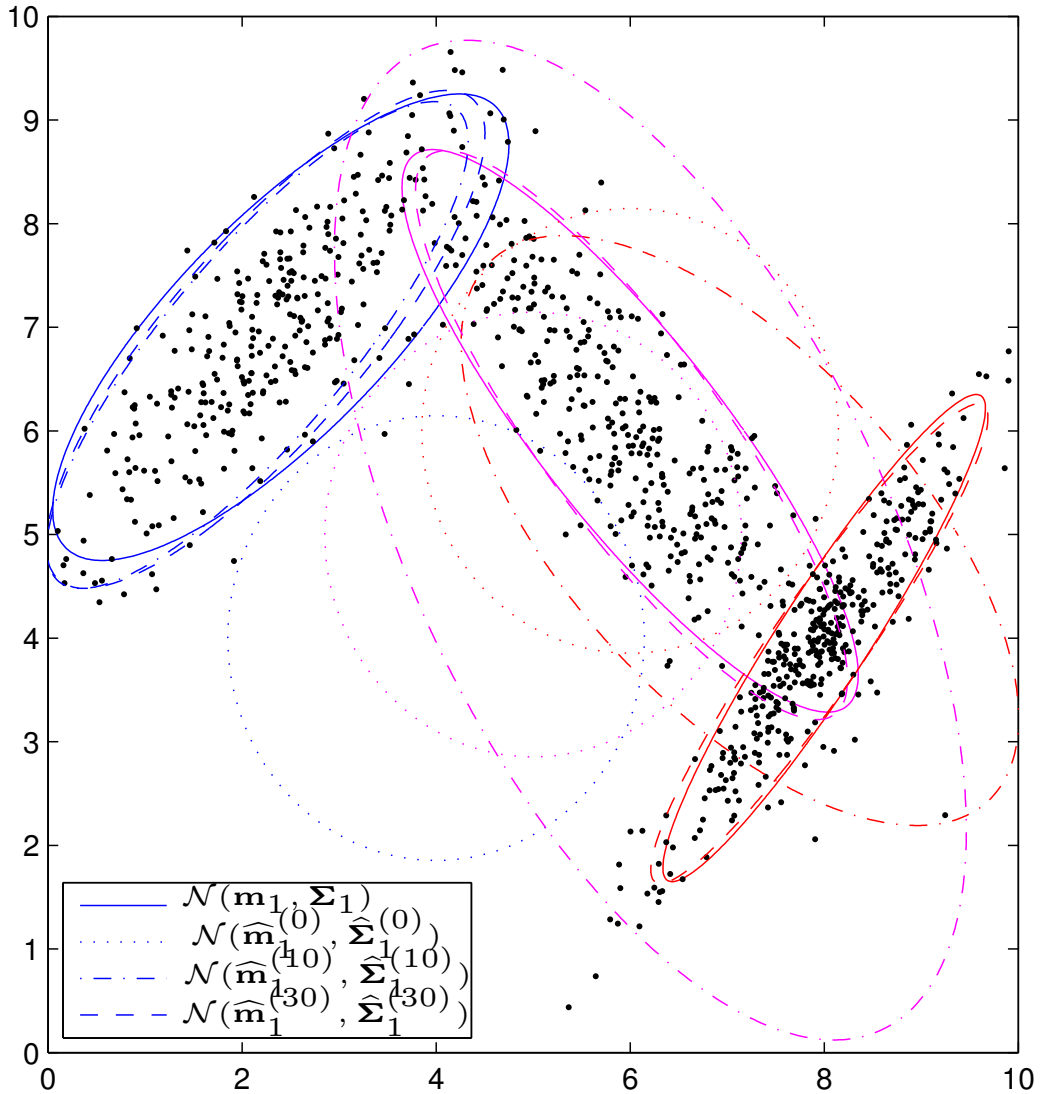
$$\mathbf{\Sigma}_n^{(i+1)} = \frac{\sum_{\ell=1}^{L} \gamma_{\ell,n}\left(\theta^{(i)}\right) \left(\mathbf{y}_\ell - \mathbf{m}_n^{(i)}\right)\left(\mathbf{y}_\ell - \mathbf{m}_n^{(i)}\right)^{\mathrm{T}}}{\sum_{\ell=1}^{L} \gamma_{\ell,n}\left(\theta^{(i)}\right)}. \tag{M.3}$$

Summary of iterative likelihood maximization procedure:

1. choose for $i = 0$ initial estimate
$$\theta^{(0)} = \{\mathbf{m}_1^{(0)}, \ldots, \mathbf{m}_N^{(0)}, \mathbf{\Sigma}_1^{(0)}, \ldots, \mathbf{\Sigma}_N^{(0)}, \pi_1^{(0)}, \ldots, \pi_N^{(0)}\}$$
2. E-step: compute $\{\gamma_{\ell,n}\left(\theta^{(i)}\right)\}$ on the basis of $\theta^{(i)}$
3. M-step: compute $\theta^{(i+1)}$ on the basis of $\{\gamma_{\ell,n}\left(\theta^{(i)}\right)\}$ according to (M.1-M.3)
4. unless $\theta^{(i+1)}$ and $\theta^{(i)}$ are sufficiently close (check e.g. corresponding log-likelihood value estimates), increase i by one and continue with step 2

Example: Iterative Likelihood Maximization:

1000 random points drawn from a 2-D mixed Gaussian distribution with 3 clusters



- solid lines indicate 10%-contours of underlying Gaussian distributions
- dotted lines indicate 10%-contours of Gaussian distributions corresponding to initial estimates
- dash-dotted lines indicate estimated distributions after 10 iterations
- dashed lines indicate estimated distributions after 30 iterations

Convergence Analysis

Write the likelihood function as

$$\log p(\mathbf{y} \mid \theta) = g(q, \theta) + d_{\mathrm{KL}}(q, r)$$

with

$$\begin{aligned} g(q,\theta) &= \sum_{n=1}^{N} q(\mathbf{z}_n) \log \frac{p(\mathbf{y}, \mathbf{Z} = \mathbf{z}_n \mid \theta)}{q(\mathbf{z}_n)} \\ d_{\mathrm{KL}}(q,r) &= -\sum_{n=1}^{N} q(\mathbf{z}_n) \log \frac{r(\mathbf{z}_n)}{q(\mathbf{z}_n)} \\ r(\mathbf{z}) &= P(\mathbf{Z} = \mathbf{z} \mid \mathbf{y}, \theta) \end{aligned}$$

and $q(\mathbf{z})$ a function subject to $q(\mathbf{z}_n) > 0$ for all $n \in \{1, \ldots, N\}$ and $\sum_{n=1}^{N} q(\mathbf{z}_n) = 1$.

Remarks:

- value of $\log p(\mathbf{y} \mid \theta)$ invariant w. r. t. $q(\cdot)$
- the *Kullback-Leibler divergence* $d_{\mathrm{KL}}(q, r)$ reflects the difference between the two probabilities represented by q and r, and $d_{\mathrm{KL}}(q, r)$ is positive unless $q = r$

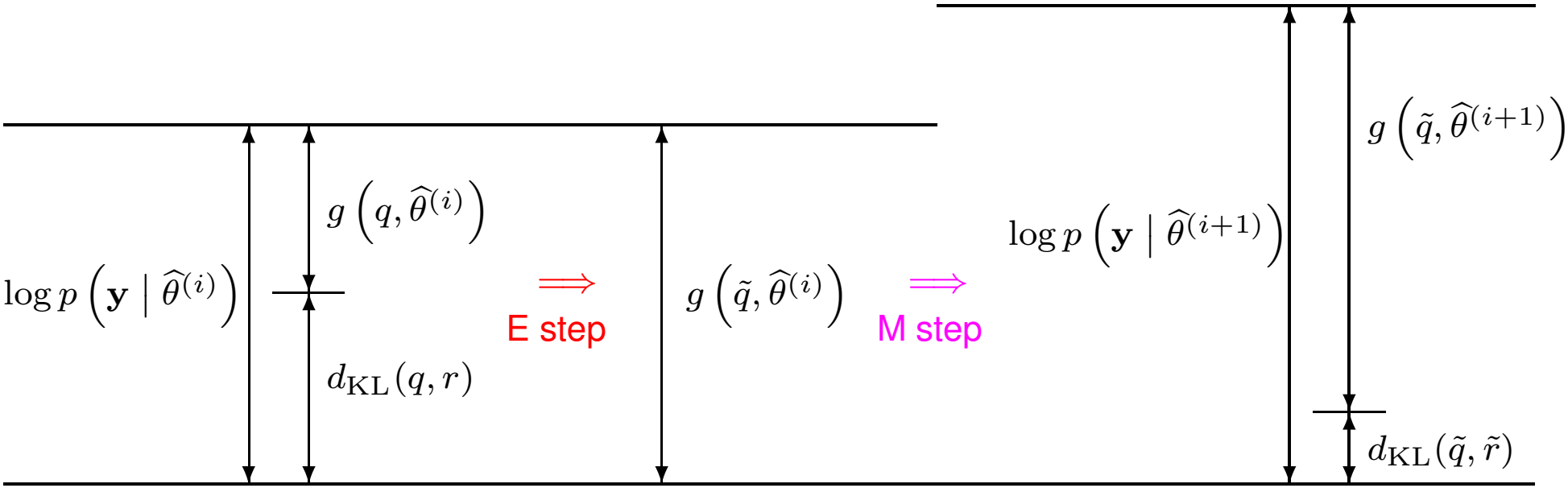
E Step and M Step

E Step: maximize $g\left(q, \widehat{\theta}^{(i)}\right)$ w. r. t. $q(\cdot)$

$\rightarrow$ $\tilde{q}(\mathbf{z}_n) = P\left(\mathbf{Z} = \mathbf{z}_n \mid \mathbf{y}, \widehat{\theta}^{(i)}\right)$ since this choice results in $d_{\mathrm{KL}}(\tilde{q}, r) = 0$

M step: maximize $g(\tilde{q}, \theta)$ w. r. t. θ

$\rightarrow$ $g\left(\tilde{q}, \widehat{\theta}^{(i+1)}\right)$



Summary

In general, the EM algorithm can be employed when

- a maximization of the likelihood function $p(\mathbf{y} \mid \theta)$ is difficult
- a maximization of the *complete-data* likelihood function $p(\mathbf{y}, \mathbf{z} \mid \theta)$ is feasible

Features:

- over the iterations the log-likelihood function values form a non-decreasing sequence
- the parameter estimates converge to a *local* maximum of the log-likelihood function, which however cannot be guaranteed to be a *global* maximum

Applications:

- in digital communications the EM algorithm can be employed e. g. for multi-path channel parameter estimation and for multiuser detection
- tasks in other fields for which the EM algorithm can be used include data clustering in machine learning, risk management in finance, and medical image reconstruction

Exercises

1. Suppose we toss a coin n independent times and define an observation sequence

$$Y_k = \begin{cases} 1 & \text{if the } k\text{th outcome is heads} \\ 0 & \text{if the } k\text{th outcome is tails} \end{cases}$$

$k = 1, 2, \ldots, n$. Let $\theta = P(Y_k = 1), k = 1, \ldots, n$.

   (a) Find an MVUE of θ.

   (b) Find the ML estimate of θ. Find its bias and variance.

   (c) Compute the Cramér-Rao lower bound and compare with results from (a) and (b).

2. Suppose we observe two jointly Gaussian random variables Y_1 and Y_2, each of which has zero mean and unit variance. We want to estimate the correlation coefficient $\rho = E\{Y_1 Y_2\}$.

   (a) Find the equation for the maximum-likelihood estimate of ρ based on observation of (Y_1, Y_2).

   (b) Compute the Cramér-Rao lower bound for unbiased estimates of ρ.

3. Suppose that, given $\Theta = \theta, Y_1, \ldots, Y_n$ are i.i.d. real observations with marginal densities

$$Y_\theta(y) = \begin{cases} \theta^{-1} e^{-y/\theta}, & y \geq 0 \\ 0, & y < 0 \end{cases}$$

   (a) Find the maximum-likelihood estimate of θ based on $Y_1, \ldots, Y_n$. Compute the mean and variance.

   (b) Compute the Ceamér-Rao lower bound for the variance of unbiased estimates of θ.

References

- H. Vincent Poor. *An Introduction to Signal Detection and Estimation*. 2^{nd}. Springer-Verlag,1994. ISBN: 0-387-94173-8 or 3-540-94173-8.
- J. M. Mendel. *Lessons in Estimation Theory for Signal Processing, Communications, and Control*. Prentice Hall PTR, 1995. ISBN: 0131209817.
- Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006. ISBN: 0387310738.
- David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. ISBN: 0521642981.
- B. Sch?olkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. ISBN: 0262194759.
- Jeff A Bilmes et al. *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*. In: International Computer Science Institute 4.510 (1998), p. 126.