Introduction to Machine Learning

Lecture Notes

Bengt Lennartson

2023

Department of Electrical Engineering
Division of Systems and Control
Chalmers University of Technology

1

Probability Theory

1.1 Probability

Probability interpretations

Probability can be defined in two different ways. One is called frequentist probability, and the other one Bayesian probability.

Frequentist probability The most common interpretation of probability is called frequentist probability. For a random experiment, the relative frequency of an event A after many trials is then defined as the probability $\mathbb{P}(A)$. More specifically, assume that N trials are performed in an experiment. If the event A occurs in N_A of these trials, then $\mathbb{P}(A)$ is defined as

$$\mathbb{P}(A) = \lim_{N \to \infty} \frac{N_A}{N} \tag{1.1}$$

where a finite N results in an approximative estimate of $\mathbb{P}(A)$ that is improved when N increases. A well known random experiment is coin tossing, where the outcome is either heads (H) or tails (T), and $\mathbb{P}(H) = \mathbb{P}(T) = 0.5$ if the coin is fair. After $N = 50$, 100, 200, 500, and 1000 trials, the standard deviations of the corresponding estimates N_A/N decreases surprisingly slowly as 0.071, 0.050, 0.035, 0.022, and 0.016.

Bayesian probability An alternative to this classical probability interpretation is Bayesian probability, which includes prior knowledge or a hypothesis together

with knowledge from experimental data. The prior knowledge or hypothesis is called *prior probability*, while the updated probability when new data (evidence) is added from experiments is called *posterior probability*. The prior knowledge in the Bayesian approach is often expressed as *degree of belief*. The degree of belief may be based on results from previous experiments, physical modelling and simulation, or earlier personal experience.

Going back to the coin tossing example, assume that the coin is not perfect, meaning that $\mathbb{P}(H) \neq 0.5$. We could then start with $\mathbb{P}(H) = \mathbb{P}(T) = 0.5$ as prior knowledge based on earlier experience, and then achieve a probability fine-tuning from experiments with the non-perfect coin.

Most of the methods presented in this text are based on frequentist probability, but the Bayesian approach has become more and more popular in machine learning, and a few methods are based on this approach.

Probability Space

The starting point in the formal definition of probability and random variables is to define a probabilty space. A probability space includes a sample space, an event space and a probability measure. These concepts are now mathematically defined.

Event and sample space An outcome ω is a possible result of an action or trial in a random experiment, and an *event* is defined as a set of outcomes. The set of all possible outcomes Ω is called the *outcome set*, but also the *sample space*. Thus, an event A is a subset of the sample space Ω, i.e. $A \subseteq \Omega$. In the coin example mentioned above, $\Omega = \{H, T\}$ and examples of events are $\{H\}$, $\{T\}$, and $\{H, T\}$.

Definition 1.1 (Probability space). The elements in a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ are

1. a *sample space* Ω, including all possible outcomes of an experiment;
2. an *event space* $\mathcal{F}$, where all events $A \in \mathcal{F}$ are sets of outcomes, i.e. $A \subseteq \Omega$ and $\mathcal{F} \subseteq 2^{\Omega}$. More specifically, $\Omega \in \mathcal{F}$, called the *certain event*, and if $A \in \mathcal{F}$ also the complement $\sim A \in \mathcal{F}$. Furthemore, if $A_1, A_2, \ldots \in \mathcal{F}$ then the countable union $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$;
3. a *probability measure* $\mathbb{P} : \mathcal{F} \to [0, 1]$ is a function from a set $A \in \mathcal{F}$ to a real number $\mathbb{P}(A) \in [0, 1]$, which expresses the probability that the event A occurs. This function has the properties

   i) $\mathbb{P}(A) \geq 0$ for all $A \in \mathcal{F}$, and $\mathbb{P}(\Omega) = 1$;

   ii) If $A_1, A_2, \ldots$ are all disjoint events, i.e. $A_i \cap A_j = \varnothing$ for all $i \neq j$, then

$$\mathbb{P}\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} \mathbb{P}(A_i). \tag{1.2}$$

□

Example 1.1 — Coin probability spaces

Tossing a fair coin, the two possible outcomes, heads (H) and tails (T), result in the sample space $\Omega = \{H, T\}$ and the maximal event set $\mathcal{F} = 2^{\Omega} = \{\varnothing, \{H\}, \{T\}, \{H, T\}\}$. The corresponding probability measures are $\mathbb{P}(\varnothing) = 0$, $\mathbb{P}(\{H\}) = \mathbb{P}(\{T\}) = 0.5$, and $\mathbb{P}(\{H, T\}) = \mathbb{P}(\{H\} \cup \{T\}) = \mathbb{P}(\{H\}) + \mathbb{P}(\{T\}) = 1$.

When two coins are tossed, the sample space is $\Omega = \{HH, HT, TH, TT\}$, and the probability measures for the events in the event space $\mathcal{F} = \{\{HH\}, \{HT, TH\}, \{TT\}\}$ are $\mathbb{P}(\{HH\}) = 0.25$, $\mathbb{P}(\{HT, TH\}) = 0.5$, and $\mathbb{P}(\{TT\}) = 0.25$. □

Example 1.2 — Infinite event set

Tossing a fair coin repeatedly until a head occurs, and using the notation $T^{i-1}H$ for a sequence of $i-1$ tails before a head is obtained, the sample space can be expressed as $\Omega = \{H, TH, T^2H, T^3H, \ldots\}$. Obviously, the size of Ω is infinite, as well as relevant events such as the event where the first head occurs after an odd number of trials, i.e. $A = \{H, T^2H, T^4H, \ldots\}$.

Considering the singleton event $A_i = \{T^{2i-2}H\}$, it is clear from Def. 1.1 that also $A = \{H, T^2H, T^4H, \ldots\} = \bigcup_{i=1}^{\infty} A_i$ is an event. □

Related to Def. 1.1 on probability space, the following observations are made:

- Since the event space $\mathcal{F}$ is [1]closed under complement $\sim A$ and countable unions $\bigcup_{i=1}^{\infty} A_i$, it means that $\mathcal{F}$ is also closed under set difference and countable intersections, where countable intersections can be rewritten as countable unions using deMorgan's rule.
- Since events are sets, and sets are closed under the ordinary set operations union, intersection, complement, and set difference, it is also clear that these operations applied on events results in events, often illustrated by Venn diagrams.
- For any event $A \in \mathcal{F}$, $A \cup \sim A = \Omega$. This implies that $\mathbb{P}(A \cup \sim A) = \mathbb{P}(A) + \mathbb{P}(\sim A) = \mathbb{P}(\Omega) = 1$, and therefore $\mathbb{P}(\sim A) = 1 - \mathbb{P}(A)$ and $\mathbb{P}(\varnothing) = 1 - \mathbb{P}(\Omega) = 0$. Furthermore, $0 \leq \mathbb{P}(A) \leq 1$.
- An *elementary event*, also called an atomic event or a sample point, is an event that only contains a single outcome $\omega \in \Omega$. Thus, an elementary event $A = \{\omega\}$ is a singleton set.
- When an outcome ω occurs, all events $A \in \mathcal{F}$ that include $\omega \in \Omega$ are said to occur.

Discrete probability measure In discrete probability theory, the sample space Ω is countable and probabilities can be expressed by the *probability mass function* $p : \Omega \to [0, 1]$ such that $\sum_{\omega \in \Omega} p(\omega) = 1$. All subsets of the sample space Ω can be treated as events, i.e. $\mathcal{F} = 2^{\Omega}$, and the probability measure for an event A may therefore be expressed as

$$\mathbb{P}(A) = \sum_{\omega \in A} p(\omega) \quad \text{for all } A \in 2^{\Omega}. \tag{1.3}$$

[1] An event $A \in \mathcal{F}$ is closed under a specific operation, for instance the complement $\sim A$, if this is also an event, i.e. $\sim A \in \mathcal{F}$.

Example 1.3 — Dice probability space

Rolling a dice, the sample space is $\Omega = \{1, 2, 3, 4, 5, 6\}$. In this example we consider three events, the first event getting number 3, the second event getting an even number, and the third event getting any number larger than 1. This corresponds to the event space $\mathcal{F} = \{\{3\}, \{2, 4, 6\}, \Omega \setminus \{1\}\}$.

The corresponding probability measures are $\mathbb{P}(\{3\}) = 1/6$, $\mathbb{P}(\{2, 4, 6\}) = 3/6$, and $\mathbb{P}(\Omega \setminus \{1\}) = 5/6$. We note generally that the probability of an event A is $|A|/|\Omega|$, since Ω is a discrete sample space, and every atomic event $\omega \in \Omega$ has in this example equal probability $1/|\Omega|$. □

Conditional probabilities and independent events

Conditional probability The probability that an event A occurs, given that an event B has occurred, denoted $\mathbb{P}(A|B)$, is called *conditional probability*. This can also be expressed as the fraction of event B for which A also occurs, i.e. the probability that both A and B occur, $\mathbb{P}(A \cap B)$, normalized with the probability $\mathbb{P}(B)$. Thus,

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{P(B)}. \tag{1.4}$$

Since the conditional probability $\mathbb{P}(A|B)$ obviously implies that both A and B occur, the normalization with $\mathbb{P}(B)$ can also be motivated by the fact that $\mathbb{P}(A|B)$ must be proportional to $\mathbb{P}(A \cap B)$, i.e. $\mathbb{P}(A|B) = \alpha\mathbb{P}(A \cap B)$. Hence, we need to show that $\alpha = 1/\mathbb{P}(B)$. This follows because the conditional probability that any event in Ω occurs, given event B, is $\mathbb{P}(\Omega|B) = 1$. On the other hand, $\mathbb{P}(\Omega|B) = \alpha\mathbb{P}(\Omega \cap B) = \alpha\mathbb{P}(B)$, which gives $\alpha = 1/\mathbb{P}(B)$.

Bayes' law Now assume that the sample space Ω is partitioned into n disjoint events $A_1, \ldots, A_n$, such that $\Omega = \bigcup_{i=1}^n A_i$ and $A_i \cap A_j = \varnothing$ for all $i \neq j$. Then $B = B \cap \Omega = B \cap \left(\bigcup_{i=1}^n A_i\right) = (B \cap A_1) \cup \cdots \cup (B \cap A_n)$, and according to (1.4), $\mathbb{P}(B \cap A_i) = \mathbb{P}(B|A_i)\mathbb{P}(A_i)$. Together with Example 1.2, we then obtain $\mathbb{P}\left(\bigcup_1^n (B \cap A_i)\right) = \sum_1^n \mathbb{P}(B \cap A_i) = \sum_1^n \mathbb{P}(B|A_i)\mathbb{P}(A_i)$, and the probability $\mathbb{P}(B)$ can be expressed as

$$\mathbb{P}(B) = \sum_1^n \mathbb{P}(B|A_i)\mathbb{P}(A_i). \tag{1.5}$$

This expression is called the *total probability formula*. Moreover, since

$$\mathbb{P}(A_i|B)\mathbb{P}(B) = \mathbb{P}(A_i \cap B) = \mathbb{P}(B \cap A_i) = \mathbb{P}(B|A_i)\mathbb{P}(A_i),$$

we also obtain the famous *Bayes' law*

$$\mathbb{P}(A_i|B) = \frac{\mathbb{P}(B|A_i)\mathbb{P}(A_i)}{\mathbb{P}(B)} = \frac{\mathbb{P}(B|A_i)\mathbb{P}(A_i)}{\sum_1^n \mathbb{P}(B|A_i)\mathbb{P}(A_i)}, \tag{1.6}$$

where the last denominator comes from (1.5). This expression was formulated by Thomas Bayes 1701–1761, who was an English statistician, philosopher and Presbyterian minister.

Example 1.4 — State and transition probabilities

For a discrete system with two states, state 1 (operating state) and state 2 (broken state), the event staying in state i has state probability $\mathbb{P}(i)$. The conditional probability, making a transition to state i from the current state j, is $\mathbb{P}(i|j)$. Applying the total probability formula (1.5) implies that

$$\begin{aligned}\mathbb{P}(1) &= \mathbb{P}(1|1)\mathbb{P}(1) + \mathbb{P}(1|2)\mathbb{P}(2)\\ \mathbb{P}(2) &= \mathbb{P}(2|1)\mathbb{P}(1) + \mathbb{P}(2|2)\mathbb{P}(2).\end{aligned}$$

Given the transition probabilities $\mathbb{P}(1|1) = 0.9$, $\mathbb{P}(1|2) = 0.4$, $\mathbb{P}(2|1) = 0.1$, and $\mathbb{P}(2|2) = 0.6$, and the fact that the state probabilities together must be equal 1, meaning that $\mathbb{P}(1) + \mathbb{P}(2) = 1$, the solution to these equations generates the state probabilities $\mathbb{P}(1) = 0.8$ and $\mathbb{P}(2) = 0.2$. □

Joint probability The probability that both event A and event B occur is called *joint probability*, and is here denoted $\mathbb{P}(A \cap B)$. In the literature it is also often written as $\mathbb{P}(A, B)$. In the first part of the coin example in Example 1.1 where only one coin is tossed, the event set $\mathcal{F} = 2^{\Omega} = \{\varnothing, \{H\}, \{T\}, \{H, T\}\}$. The joint probability $\mathbb{P}(\{H\}, \{T\}) = \mathbb{P}(\{H\} \cap \{T\}) = 0$, since only one of the two outcomes may appear tossing one coin. The result is either $\{H\}$ or $\{T\}$. On the other hand, $\mathbb{P}(\{H, T\}) = \mathbb{P}(\{H\} \cup \{T\}) = \mathbb{P}(\Omega) = 1$, where the union operator means that either $\{H\}$ or $\{T\}$ occurs, corresponding to the whole sample space. To avoid the confusion between $\mathbb{P}(\{H\}, \{T\}) \neq \mathbb{P}(\{H, T\})$, we recommend to use the intersection operator for the joint probability.

Independent events Independent events mean that the joint probability of a set of events $A_1, \ldots, A_n$ is equal to the product of the probabilities if the individual events, i.e.

$$\mathbb{P}\Big(\bigcap_{i=1}^{n} A_i\Big) = \prod_{i=1}^{n} \mathbb{P}(A_i).$$

When n coins are tossed the events are independent, meaning for instance that if a head is achieved for one coin, it does not influence the outcome of the other coins. The probability to get only heads for all coins $\mathbb{P}(\{H \ldots H\}) = \mathbb{P}(\{H^n\}) = 0.5^n$, and to get tails for all coins $\mathbb{P}(\{T^n\}) = 0.5^n$. In the same way, rolling n dices the probability to get the same number every time is $1/6^n$.

When two events A and B are independent, the conditional probability $\mathbb{P}(A|B) = \mathbb{P}(A \cap B)/\mathbb{P}(B)$ is also simplified to

$$\mathbb{P}(A|B) = \mathbb{P}(A),$$

since the outcome of event A is not influenced by the fact that the event B has occurred.

Disjoint and not disjoint events We conclude this subsection by some remarks on disjoint and not disjoint events. Already in the definition of probability measure in Def. 1.1 we observed that when events $A_1, A_2, \ldots$ are all disjoint events, i.e. $A_i \cap A_j = \varnothing$ for all $i \neq j$, the probability of unions of events can be expressed as the sum of the probability of the individual events. Disjoint events are also called *mutually exclusive events*, meaning that they cannot happen at the same time.

On the other hand, if two events A and B are not mutually exclusive and therefore not disjoint, they can happen at the same time, and $\mathbb{P}(A \cap B)$ is not restricted to be zero. The probability of the union of A and B then becomes

$$\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B).$$

This avoids that the outcomes in $A \cap B$ are counted two times when the outcomes in A and B are added together.

1.2 Probability Density and Mass Functions

Random variables and distribution functions

For a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a random variable is a function $X : \Omega \to \mathbb{R}$ such that for any real number $x \in \mathbb{R}$, the set

$$A(x) = \{\omega \in \Omega \,|\, X(\omega) \leq x\} \subseteq \Omega$$

is an event $A(x) \in \mathcal{F}$ with a probability $\mathbb{P}(A(x))$. This probability is often written in shortened form as $\mathbb{P}(X \leq x)$, and the *distribution function* of X, denoted $F(x)$, is an alternative notion for the probability $\mathbb{P}(A(x))$, i.e.

$$F(x) = \mathbb{P}(A(x)) = \mathbb{P}(X \leq x).$$

This function expresses the relative frequency (fraction of times) the variable $X \leq x$. The distribution function $F(x)$ is nondecreasing, i.e. if $x < y$ then $F(x) \leq F(y)$, and the lower and upper limits are $\lim_{x \to -\infty} F(x) = 0$ and $\lim_{x \to \infty} F(x) = 1$. We write $F_X(x)$ when multiple variables are involved.

Probability Density Functions

When X is a continuous random variable, the *probability density function* (pdf) $p(x)$ is the derivative of the distribution function $F(x)$, i.e. $p(x) = \frac{dF(x)}{dx}$, and the probability that X assumes a value between a and b is expressed as

$$\mathbb{P}(a \leq X < b) = \int_a^b p(x)\, dx \tag{1.7}$$

Furthermore, $p(x) \geq 0$ and $\int_{-\infty}^{\infty} p(x)\, dx = 1$.

Example 1.5 — Uniform and Gaussian distributions

Two common probability density functions are the *uniform distribution*

$$p(x) = \begin{cases} \dfrac{1}{b-a} & a < x \leq b \quad (b > a) \\ 0 & \text{elsewhere,} \end{cases}$$

and the *Gaussian*, or *normal, distribution*

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\sigma > 0),$$

also expressed as $X \sim \mathcal{N}(\mu, \sigma)$, where μ is the mean value and σ is the standard deviation. □

Probability mass functions

A discrete random variable X assumes values x that map the elements ω in the sample space Ω to particular quantities of interest defined by a set $\mathcal{T} \subseteq \mathbb{R}$, i.e. X is now a function $X : \Omega \to \mathcal{T}$. Tossing two coins where the sample space $\Omega = \{HH, HT, TH, TT\}$, the quantity of interest can be the number of heads. Then $X(HH) = 2$, $X(HT) = 1$, $X(TH) = 1$, and $X(TT) = 0$, which results in the quantity of interest set $\mathcal{T} = \{0, 1, 2\}$

Rolling two dices, the outcome of the individual dices are $n_i \in \{1, 2, 3, 4, 5, 6\}$, $i = 1, 2$ and the sample space for the two dices includes all combinations (n_1, n_2). The quantity of interest can be sum of the numbers $n_1 + n_2$ on the two dices. Thus, the variable X takes values in the set $\mathcal{T} = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$.

Based on the possible values $x \in \mathcal{T}$ of the random variable X, a probability function can now be defined as $p : \mathcal{T} \to [0, 1]$, such that $\sum_{x \in \mathcal{T}} p(x) = 1$. This function is also called *probability mass function* (pmf)

Example 1.6 — Bernoulli distribution

The most simple and basic pmf is the *Bernoulli distribution*

$$p(x|q) = q^x (1-q)^{1-x} \quad x \in \{0, 1\},$$

where tossing a coin, see Example 1.1, has the pmf $p(x|0.5)$. Another important application of this distribution is the probability that a machine or device is working with probability $p(1) = q$ and broken with probability $p(0) = 1 - q$. □

Expectations

The expectations of a continuous and a discrete random variable X are

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} xp(x)\,dx \qquad\qquad \mathbb{E}[X] = \sum_{x:p(x)>0} xp(x)$$

A shorthand for the expectation, also called *mean* is $\mu = \mathbb{E}[X]$. For an arbitrary function $g : \mathbb{R} \to \mathbb{R}$, the mapping $Y = g(X)$ is also a random variable. The expected values of the corresponding continuous and discrete random variable $g(X)$ are

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)p(x)\,dx \qquad\qquad \mathbb{E}[g(X)] = \sum_{x:p(x)>0} g(x)p(x)$$

Variance and moments A common special case is the variance $\text{var}(X)$ that is obtained for $g(X) = (X-\mu)^2$, i.e.

$$\text{var}(X) = \mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2] - \mu^2$$

and the standard deviation $\sigma = \sqrt{\text{var}(X)}$. More generally, for $k \in \mathbb{N}^+ = \{1, 2, \ldots\}$ the kth moment m_k is defined as

$$m_k = \mathbb{E}[X^k],$$

and the kth central moment σ_k is

$$\sigma_k = \mathbb{E}[(X-\mu)^k].$$

Obviously, the mean $\mu = m_1$ and the variance $\text{var}(X) = \sigma_2 = \sigma^2$.

Example 1.7 — Mean and variance

For the uniform distribution with pdf $p(x) = 1/(b-a)$ for $a < x \leq b$ and $p(x) = 0$ elsewhere, the mean μ and variance σ^2 are obtained based on the definitions above. More specifically,

$$\mu = \mathbb{E}[X] = \frac{1}{b-a}\int_a^b x\,dx = \frac{1}{b-a}\Big(\frac{b^2}{2} - \frac{a^2}{2}\Big) = \frac{(b-a)(b+a)}{2(b-a)} = \frac{b+a}{2}$$

$$\mathbb{E}[X^2] = \frac{1}{b-a}\int_a^b x^2\,dx = \frac{1}{b-a}\Big(\frac{b^3}{3} - \frac{a^3}{3}\Big) = \frac{(b^2+ab+a^2)}{3} = \frac{(b+a)^2 - ab}{3}$$

$$\sigma^2 = \mathbb{E}[X^2] - \mu^2 = \frac{(b+a)^2 - ab}{3} - \frac{(b+a)^2}{4} = \frac{(b+a)^2 - 4ab}{12} = \frac{(b-a)^2}{12}$$

For the discrete Bernoulli distribution with pmf $p(x|q) = q^x(1-q)^{1-x}$, the mean and variance σ^2 are obtained by replacing the integral above with a sum as follows:

$$\mu = \mathbb{E}[X] = \sum_{x:p(x|q)>0} xp(x|q) = 0 \cdot q^0(1-q)^1 + 1 \cdot q^1(1-q)^0 = q$$
$$\mathbb{E}[X^2] = \sum_{x:p(x|q)>0} x^2p(x|q) = 0^2 \cdot q^0(1-q)^1 + 1^2 \cdot q^1(1-q)^0 = q$$
$$\sigma^2 = \mathbb{E}[X^2] - \mu^2 = q - q^2 = q(1-q)$$

□

Covariance The covariance between two random variables X and Y with mean values μ_X and μ_Y is defined as

$$\text{cov}(X,Y) = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)] = \mathbb{E}[XY] - \mu_X\mu_Y$$

The random variables are said to be *uncorrelated* if $\text{cov}(X,Y) = 0$.

Stochastic processes

A sequence of stochastic variables is called a *stochastic process*. Such a sequence often represents a signal at time points $k = 0, 1, 2, \ldots$ in a dynamic system. Since signals are normally denoted by small letters, such as a system output signal $y(k)$, a control signal $u(k)$, or a disturbance signal $e(k)$, the common notation in probability theory that a stochastic variable is denoted by a capital letter, see above, is now replaced by small letters for stochastic variables in stochastic processes, such as

$$\{y(k), k = 0, 1, 2, \ldots\},$$

in shorter form written as $\{y(k)\}$. For stochastic processes, it is of interest to express how signals are correlated at different time instances. This can be determined by the expectation

$$r_y(k,\ell) = \mathbb{E}[y(k+\ell)y(k)]$$

Stationary stochastic processes When a stochastic process is stationary, the expectation above normally does not depend on the time k, only the time difference ℓ [2]. This results in the simplified and symmetric expectation

$$r_y(\ell) = \mathbb{E}[y(k+\ell)y(k)] = \mathbb{E}[y(k)y(k-\ell)] = \mathbb{E}[y(k-\ell)y(k)] = r_y(-\ell)$$

In the same way, the correlation between two stochastic processes $\{y(k)\}$ and $\{u(k)\}$ depends on the expression

$$r_{yu}(\ell) = \mathbb{E}[y(k+\ell)u(k)] = \mathbb{E}[y(k)u(k-\ell)] = \mathbb{E}[u(k-\ell)y(k)] = r_{uy}(-\ell)$$

[2] To be time independent, it is also required that a stationary stochastic process is ergodic

Auto-correlation and cross-correlation When the means $\mu_y = \mathbb{E}[y(k] = 0$ and $\mu_u = \mathbb{E}[u(k] = 0$, the expectations $r_y(\ell)$ and $r_{yu}(\ell)$ can be considered as covariance functions, and when they are normalized with $r_y(0)$ and $r_{yu}(0)$, respectively, the expression

$$\frac{r_y(\ell)}{r_y(0)}$$

is called *auto-correlation*, and

$$\frac{r_{yu}(\ell)}{r_{yu}(0)}$$

is called *cross-correlation*. Here we observe that for slowly time-varying signals, $y(k+\ell)$ is close to $y(k)$ for small time differences ℓ. This results in a stronger correlation between $y(k+\ell)$ and $y(k)$, and a higher value of the auto-correlation $r_y(\ell)/r_y(0)$, see Example 1.9, compared to a more rapidly time-varying signal, where the auto-correlation for the same time difference ℓ is smaller.

When u is the input to a system, y is the corresponding output, and the input does not include any feedback from the output, it is also clear that

$$r_{uy}(\ell) = \mathbb{E}[u(k+\ell)y(k)] = 0, \quad \ell > 0.$$

The reason is that the output $y(k)$ cannot depend on a future independent input signal $u(k+\ell)$ that has not happened at time k. Furthermore, without feedback control, the output $y(k)$ normally only depends on past input signals $u(k-\ell)$ where $\ell > 0$, and in that case also

$$r_{uy}(0) = \mathbb{E}[u(k)y(k)] = \mathbb{E}[y(k)u(k)] = 0$$

White noise A zero-mean stationary stochastic process $e(k)$ that is uncorrelated in time is called *white noise*, which means that the covariance

$$r_e(\ell) = \mathbb{E}[e(k+\ell)e(k)] = \begin{cases} r_e & \ell = 0 \\ 0 & \ell \neq 0 \end{cases}$$

and since the mean $\mathbb{E}e(k) = 0$, r_e is also the variance of the white noise $e(k)$. Furthermore,

$$r_{ey}(\ell) = \mathbb{E}[e(k+\ell)y(k)] = \mathbb{E}[e(k)y(k-\ell)] = 0, \quad \ell > 0.$$

since any future white noise $e(k+\ell)$ is uncorrelated with the current output $y(k)$, indeed also the current input $u(k)$.

Example 1.8 — Covariance functions for dynamic system

For a dynamic system given by the discrete-time model

$$y(k) = ay(k-1) + bu(k-1) + e(k), \qquad k = 1, 2, 3, \ldots,$$

the input u and the noise e are assumed to be mutually uncorrelated white noises with variances r_u and r_e, respectively. First, we observe that the average $\mu_y = \mathbb{E}[y(k)] = \mathbb{E}[y(k-1)] = 0$, since the expectation on the discrete-time model with the zero-mean inputs $\mu_u = \mu_e = 0$ gives

$$\mu_y - a\mu_y = b\mu_u + \mu_e = 0$$

Taking the expectation on the system equation multiplied with $y(k-1)$, results in the one-step output covariance

$$\begin{aligned} r_y(1) = \mathbb{E}[y(k)y(k-1)] &= \mathbb{E}[\big(ay(k-1) + bu(k-1) + e(k)\big)y(k-1)] \\ &= a\mathbb{E}[y^2(k-1)] + b\mathbb{E}[u((k-1)y(k-1)] + \mathbb{E}[e(k)y(k-1)] = ar_y(0). \end{aligned}$$

This follows since $\mathbb{E}[e(k)y(k-1)] = 0$ and $\mathbb{E}[u(k-1)y(k-1)] = \mathbb{E}[y(k)u(k)] = 0$, because $y(k)$ does not depend on $u(k)$, only $u(k-1)$.

In the same way, $r_y(\ell) = \mathbb{E}[y(k)y(k-\ell)] = a\mathbb{E}[y(k-1)y(k-\ell)] + b\mathbb{E}[u(k-1)y(k-\ell)] + \mathbb{E}[e(k)y(k-\ell)] = ar_y(\ell-1)$. Iterating this expression $\ell-1$ times gives

$$r_y(\ell) = a^\ell r_y(0).$$

Finally, taking the expectation on the system equation, multiplied by itself, generates the variance

$$\begin{aligned} r_y(0) =& \mathbb{E}[y^2(k)] = \mathbb{E}[\big(ay(k-1) + bu(k-1) + e(k)\big)^2] \\ =& a^2\mathbb{E}[y^2(k-1)] + b^2\mathbb{E}[u^2(k-1)] + \mathbb{E}[e^2(k)] + 2ab\mathbb{E}[y(k-1)u(k-1)] + \\ & 2a\mathbb{E}[y(k-1)e(k)] + 2b\mathbb{E}[u(k-1)e(k)] = a^2r_y(0) + b^2r_u + r_e, \end{aligned}$$

since $\mathbb{E}[y(k-1)u(k-1)] = 0$, $\mathbb{E}[y(k-1)e(k)] = 0$, and $\mathbb{E}[u(k-1)e(k)] = 0$. Thus, the ℓ-step covariance

$$r_y(\ell) = a^\ell r_y(0) = \frac{a^\ell(b^2r_u + r_e)}{1-a^2},$$

and the auto-correlation $r_y(\ell)/r_y(0) = a^\ell$.

The expectation of the output signal $\mu_y = \mathbb{E}[y(k)] = \mathbb{E}[y(k-1)] = 0$, since the expectation on the discrete-time model with the zero-mean inputs $\mu_u = \mu_e = 0$ (u and e are white noises) gives

$$\mu_y - a\mu_y = \mu_e = 0$$

Taking the expectation on the system equation multiplied with $y(k-1)$, results in the one-step output covariance

$$\begin{aligned} r_y(1) = \mathbb{E}[y(k)y(k-1)] &= \mathbb{E}[\big(ay(k-1) + e(k)\big)y(k-1)] \\ &= a\mathbb{E}[y^2(k-1)] + \mathbb{E}[e(k)y(k-1)] = ar_y(0). \end{aligned}$$

This follows since $e(k)$ is white noise and happens after $y(k-1)$, and therefore $\mathbb{E}[e(k)y(k-1)] = 0$. In the same way,

$$r_y(\ell) = \mathbb{E}[y(k)y(k-\ell)] = a\mathbb{E}[y(k-1)y(k-\ell)] + \mathbb{E}[e(k)y(k-\ell)] = ar_y(\ell-1).$$

Iterating this expression for $\ell = 2, \ldots, \ell$ gives

$$r_y(2) = ar_y(1) = a^2r_y(0),\ r_y(3) = ar_y(2) = a^3r_y(0), \ldots, r_y(\ell) = a^\ell r_y(0).$$

Finally, taking the expectation on the system equation, multiplied by itself, generates the variance

$$\begin{aligned} r_y(0) =& \mathbb{E}[y^2(k)] = \mathbb{E}[\big(ay(k-1) + e(k)\big)^2] \\ =& a^2\mathbb{E}[y^2(k-1)] + \mathbb{E}[e^2(k)] + 2a\mathbb{E}[y(k-1)e(k)]] \\ = a^2 r_y(0) + r_e, \end{aligned}$$

since again $\mathbb{E}[y(k-1)e(k)] = 0$, and therefore $r_y(0) = r_e/(1-a^2)$. Thus, the ℓ-step covariance

$$r_y(\ell) = a^\ell r_y(0) = \frac{a^\ell r_e}{1-a^2},$$

and the auto-correlation $r_y(\ell)/r_y(0) = a^\ell$.

□

A similar example follows directly, but now with the white noise $e(k)$ extended to the non-white noise $e(k) + ce(k-1)$, also called *colored noise*, and only focus on the auto-correlation as a function of the variance $r_y(0)$.

Example 1.9 — Auto-correlation for a dynamic system with colored noise

For a dynamic system given by the discrete-time model

$$y(k) = ay(k-1) + bu(k-1) + e(k) + ce(k-1), \qquad k = 1, 2, 3, \ldots,$$

also here the input signals u and e are assumed to be mutually uncorrelated white noises with variances r_u and r_e, respectively. In the same way as in the last example, it follows that the average $\mu_y = \mathbb{E}[y(k)] = 0$. First, taking the expectation on the system equation multiplied with $e(k)$, results in the cross-covariance

$$\begin{aligned} r_{ye}(0) &= \mathbb{E}[y(k)e(k)] = \mathbb{E}[\big(ay(k-1) + bu(k-1) + e(k) + ce(k-1)\big)e(k)] \\ &= a\mathbb{E}[y(k-1)e(k)] + b\mathbb{E}[u((k-1)e(k)] + \mathbb{E}[e(k)e(k) + c\mathbb{E}[e(k-1)e(k)] = r_e, \end{aligned}$$

since $e(k)$ is white noise, and therefore $\mathbb{E}[y(k-1)e(k)] = 0$, $\mathbb{E}[u(k-1)e(k)] = 0$, and $\mathbb{E}[e(k-1)e(k)] = 0$. In the same way, taking the expectation on the system equation multiplied with $y(k-1)$, results in the one-step covariance

$$\begin{aligned} r_y(1) =& \mathbb{E}[y(k)y(k-1)] = \mathbb{E}[\big(ay(k-1) + bu(k-1) + e(k) + ce(k-1)\big)y(k-1)] \\ =& a\mathbb{E}[y^2(k-1)] + b\mathbb{E}[u(k-1)y(k-1)] + \mathbb{E}[e(k)y(k-1) + \\ & c\mathbb{E}[e(k-1)y(k-1)] = ar_y(0) + cr_e. \end{aligned}$$

This follows, since $\mathbb{E}[u(k-1)y(k-1)] = \mathbb{E}[y(k)u(k)] = 0$, $\mathbb{E}[e(k)y(k-1)] = 0$, and $\mathbb{E}[e(k-1)y(k-1)] = r_{ye}(0) = r_e$.

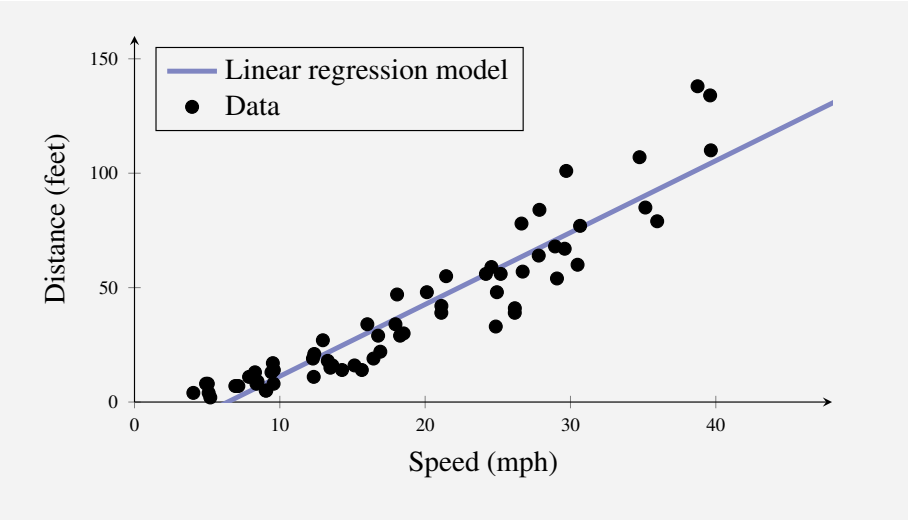
In the same way $r_y(\ell) = \mathbb{E}[y(k)y(k-\ell)] = a\mathbb{E}[y(k-1)y(k-\ell)] + b\mathbb{E}[u(k-1)y(k-\ell)] + \mathbb{E}[e(k)y(k-\ell)] + c\mathbb{E}[e(k-1)y(k-\ell)] = ar_y(\ell-1)$. Iterating this expression $\ell-1$ times gives $r_y(\ell) = a^{\ell-1}r_y(1)$. Thus, the auto-correlation

$$\frac{r_y(\ell)}{r_y(0)} = \frac{a^{\ell-1}r_y(1)}{r_y(0)} = a^\ell + a^{\ell-1}c\frac{r_e}{r_y(0)}.$$

This result illustrates that when the white noise $e(k)$ is replaced by the colored noise $e(k) + e(k-1)$, an additional term is added in the auto-correlation function, proportional to the parameter c, but also the inverse signal-to-noise relation $r_e/r_y(0)$. □

2

Linear Regression



Machine Learning – A First Course for Engineers and Scientists
Online draft version July 8, 2022, `http://smlbook.org`
© *Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön* 2022.

Linjär regression

$$y_i = \theta_0 + x_1\theta_1 + \dots + x_n\theta_n + \varepsilon_i$$

N experiments $i = 1, 2, \dots, N$

$$x_i^T = [1 \; x_{i1} \; x_{i2} \; \dots \; x_{in}]$$

$$\theta^T = [\theta_0 \; \theta_1 \; \theta_2 \; \dots \; \theta_n]$$

$$y_i = x_i^T\theta + \varepsilon_i \qquad i = 1, \dots, N$$

Välj en skattning $\hat{\theta}$ så att $J(\theta) = \frac{1}{2}\sum_{i=1}^{N} \varepsilon_i^2$ minimeras

Minimum erhålls då

$$\frac{dJ(\theta)}{d\theta} = \sum_{i=1}^{N} \frac{d\varepsilon_i}{d\theta}\varepsilon_i = 0$$

$$\frac{d}{d\theta} a^T\theta = a \quad \Rightarrow$$

$$\frac{d\varepsilon_k}{d\theta} = \frac{d}{d\theta}(y_i - x_i^T\theta) = -x_i$$

$$-\frac{dJ}{d\theta} = \sum_{i=1}^{N} x_i(y_i - x_i^T\theta) =$$

$$= \sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i x_i^T \theta = 0$$

Detta ger normalekvationen

$$\boxed{\left(\sum_{i=1}^{N} x_i x_i^T\right)\hat{\theta} = \sum_{i=1}^{N} x_i y_i \quad (1)}$$

där θ ersatts med $\hat{\theta}$, eftersom lösningen till detta ekvationssystem ger den optimala skattningen $\hat{\theta}$ som minimerar $J(\theta)$. Normera (1) med $\frac{1}{N}$ och låt $N \to \infty$, vilket ger vänte värdet med antagandet om ergodicitet

dvs

$$\frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{ersätts}$$

med $E\,x$

normalekvationen kan formuleras som $(E x x^T)\hat{\theta} = E x y$

$$\hat{\theta} = (E x x^T)^{-1} E x y$$

En alternativ formulering som ges i boken är att införa matriserna

$$X = \begin{bmatrix} x_1^T \\ x_2^T \\ \\ x_N^T \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}$$

Eftersom

$$\sum_{i=1}^{N} x_i x_i^T = n \left\{ \underbrace{[x_1 \; \dots \; x_N]}_{N} \underbrace{\begin{bmatrix} x_1^T \\ \\ x_N^T \end{bmatrix}}_{n} \right\} N$$

$$= X^T X$$

$$\sum_{i=1}^{N} x_i y_i = [x_1 \dots x_N] \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = X^T Y$$

så kan även normalekvationen formuleras som

$$(X^T X)\hat{\theta} = X^T Y$$

$$\Rightarrow \hat{\theta} = (X^T X)^{-1} X^T Y$$

Nedan följer ett antal exempel på hur linjära regressions-modeller kan formuleras.

EX.1 polynommodell i x

$$y = \theta_0 + x\theta_1 + x^2\theta_2 + \dots + x^n\theta_n + \varepsilon \qquad y = \begin{bmatrix} 1 & x & x^2 & \dots & x^n \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix} + \varepsilon$$

EX.2 Diff.ekv. för ett dynamiskt system

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_1 u(k-1) + \dots + b_n u(k-n) + \varepsilon(k)$$

$$y(k) = \underbrace{\begin{bmatrix} -y(k-1) & \dots & -y(k-n) & u(k-1) & \dots & u(k-n) \end{bmatrix}}_{X^T(k)} \underbrace{\begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\theta} + \varepsilon(k)$$

EX.3 Fourier serie

$$y = \theta_0 + \theta_1 \sin(x) + \theta_2 \cos(x) + \theta_3 \sin(2x) + \theta_4 \cos(2x) + \theta_5 \sin(3x) + \theta_6 \cos(3x) = + \dots + \varepsilon$$

$$= \begin{bmatrix} 1 & \sin(x) & \cos(x) & \sin(2x) & \cos(2x) & \dots \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \vdots \end{bmatrix} + \varepsilon$$

EX.4 $z = u v^a$

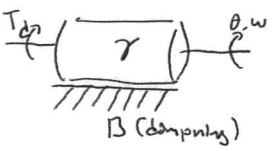
a är den okända parametern
z, u och v är kända signaler.

$$\ln z = \ln u + a \ln v$$

$$\underbrace{\ln z - \ln u}_{y} = \underset{\theta}{a} \underbrace{\ln v}_{x}$$

$$y = x\theta + \varepsilon$$

Transformation från diff.ekv. till regressionsmodell



$$\gamma\dot{\omega} + B\omega = T_d$$

Givet $\omega(kh)$ $k = 1, 2, 3 \dots$
h = samplingsintervallet

$$\dot{\omega}(kh) \approx \frac{\omega(kh) - \omega(kh-h)}{h} \quad (*) \qquad \left(\text{Kan få } \ddot{\omega}(kh) \approx \frac{\dot{\omega}(kh) - \dot{\omega}(kh-h)}{h} \text{ där } \dot{\omega}(kh) \text{ fås av } * \right)$$

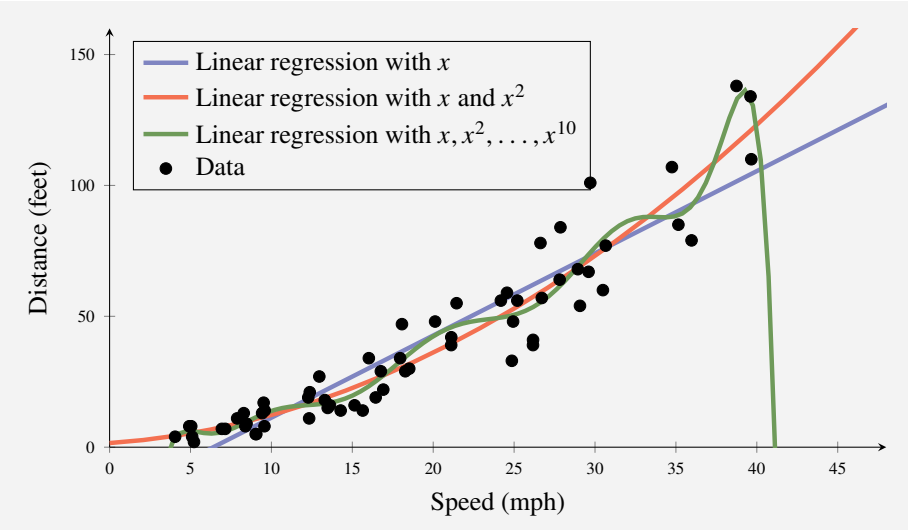
Denna approx. av en derivata kallas för differenskapproximation men också Euler-approx

$$\gamma = \frac{\omega(kh+h) - \omega(kh)}{h} + B\omega(kh) = T_d(kh)$$

$k=0$: $\omega(0) = 0$
$k=1$: $\omega(h) = \omega(0) + \frac{h}{\gamma}(T_d(0) - B\omega(0))$
$k=2$: $\omega(2h) = \omega(h) + \frac{h}{\gamma}(T_d(h) - B\omega(h))$
$\vdots$

Givet en diff.approx. av tidsderivatan $\dot{\omega}$ erhålls följande regressionsmodell:

$$\underset{y}{T_d} = \underbrace{\begin{bmatrix} \dot{\omega} & \omega \end{bmatrix}}_{X^T} \underbrace{\begin{bmatrix} \gamma \\ B \end{bmatrix}}_{\theta} + \varepsilon$$



Machine Learning – A First Course for Engineers and Scientists
Online draft version July 8, 2022, `http://smlbook.org`
© *Andreas Lindholm, Niklas Wahlström, Fredrik Lindsten, and Thomas B. Schön* 2022.

Farthillare

$$m\dot{v} = u - bv^2$$

$$\underset{\uparrow \; y}{u} = m\dot{v} + bv^2 + \varepsilon = \underbrace{\begin{bmatrix} \dot{v} & v \end{bmatrix}}_{X^T} \underbrace{\begin{bmatrix} m \\ b \end{bmatrix}}_{\theta} + \varepsilon$$

Servomotor med ström som insignal

$$\gamma\dot{\omega} + B\omega = T_d = K_m i_a$$

Regressionsmodell:

$$\begin{bmatrix} \dot{\omega} & \omega & -i_a \end{bmatrix} \begin{bmatrix} \gamma \\ B \\ K_m \end{bmatrix} + \varepsilon = 0$$

Alternativ formulering:

$$i_a = \frac{\gamma}{K_m}\dot{\omega} + \frac{B}{K_m}\omega + \varepsilon = \begin{bmatrix} \dot{\omega} & \omega \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \varepsilon \quad \text{där} \quad \theta_1 = \frac{\gamma}{K_m}, \; \theta_2 = \frac{B}{K_m}$$

$$i_a = \theta_1\dot{\omega} + \theta_2\omega$$

Laplace:

$$I_a(s) = (s\theta_1 + \theta_2)\,\Omega(s)$$

$$G(s) = \frac{\Omega(s)}{I_a(s)} = \frac{1}{\theta_2 + s\theta_1} = \frac{(1/\theta_2)\,K}{1 + \underbrace{\left(\frac{\theta_1}{\theta_2}\right)}_{T \text{ (Km bytes mot)}} s} = \frac{K}{1 + Ts}$$

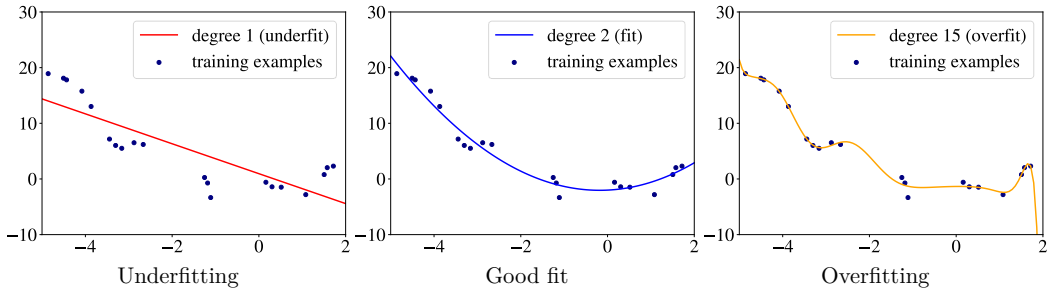


Figure 9: Examples of underfitting (linear model), good fit (quadratic model), and overfitting (polynomial of degree 15).

Andriy Burkov, Machine Learning Engineering

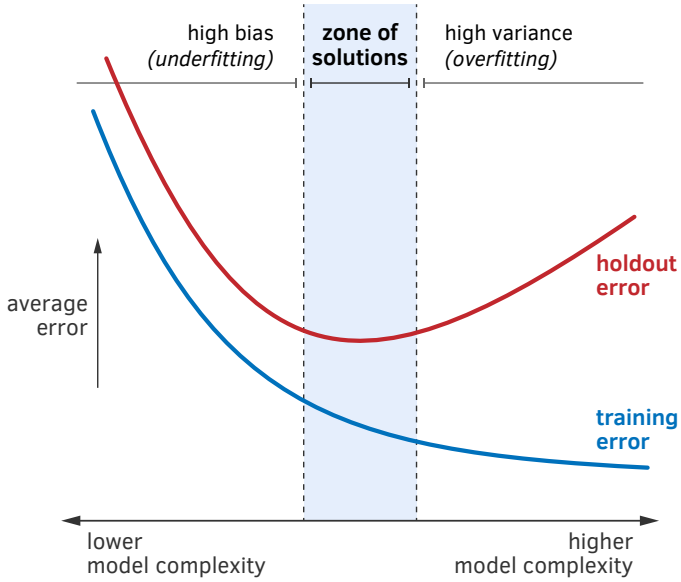


Figure 10: Bias-variance tradeoff.

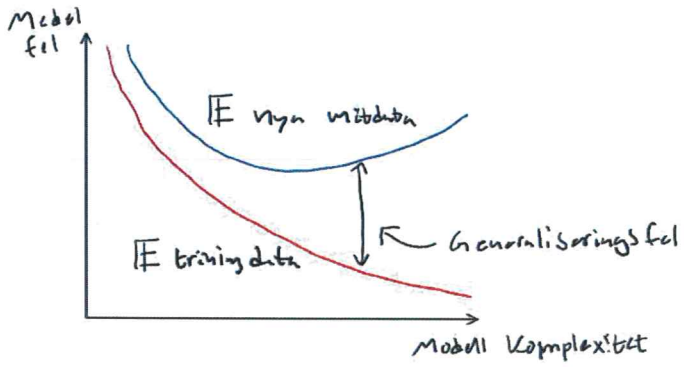
Andriy Burkov, Machine Learning Engineering

Utvärdering av skattade modeller

Korsvalidering

Datamängd

Parameter skattning $\hat{\theta}$	Utvärdering av $\hat{\theta}$



Ökad modellkomplexitet innebär typiskt fler parametrar i vektorn θ

Bias vid regressionsmodell skattning

regressions modell: $y_i = x_i^T \theta_0 + e_i \qquad i = 1, \dots, N$

$$\hat{\theta} = \underbrace{\left(\sum_{i=1}^{N} x_i x_i^T\right)^{-1}}_{R_N^{-1}} \sum_{i=1}^{N} x_i y_i = R_N^{-1} \left(\underbrace{\sum_{i=1}^{N} x_i x_i^T}_{R_N} \theta_0 + x_i e_i \right) = \theta_0 + \sum_{i=1}^{N} x_i e_i$$

$$\mathbb{E}(\hat{\theta}) = \theta_0 + \sum_{i=1}^{N} \mathbb{E}\, x_i e_i$$

Bias undviks dvs. $\mathbb{E}(\hat{\theta}) = \theta_0$ då x_i och e_i är okorrelerade.

Kovarians för skattning $\hat{\theta}$

$$\mathbb{E}(\hat{\theta} - \mathbb{E}(\hat{\theta})(\hat{\theta} - \mathbb{E}(\hat{\theta}))^T$$

Antag att bias undviks dvs. $\mathbb{E}(\hat{\theta}) = \theta_0$

$$\hat{\theta} = \theta_0 + R_N^{-1} \sum_{k=1}^{N} x(k) e(k)$$

$$\operatorname{cov}(\hat{\theta}) = \mathbb{E}(\hat{\theta} - \theta_0)(\hat{\theta} - \theta_0)^T = \mathbb{E}\, R_N^{-1} \sum_{k=1}^{N} x(k) e(k) \sum_{l=1}^{N} e(l) x^T(l) R_N^{-1}$$

$e(k)$ antas vara vitt brus dvs. $\mathbb{E}\, e(k) e(l) = 0 \quad , \; k \neq l$

$$\Rightarrow \operatorname{cov} \hat{\theta} = R_N^{-1} \sum_{k=1}^{N} x(k) \underbrace{\mathbb{E}(e^2(k))}_{r_e} x^T(k) R_N^{-1} = R_N^{-1} R_N R_N^{-1} r_e = R_N^{-1} r_e$$

Slutsats:

$$\operatorname{cov} \hat{\theta} = R_N^{-1} r_e = \left(\sum_{k=1}^{N} x(k) x^T(k) \right)^{-1} \mathbb{E}(e^2(k))$$

Regularisering är ett bra sätt att minska varians i skattningen.

Optimeringskriterium: $J(\theta) = \sum_{k=1}^{N} \underbrace{(y(k) - \hat{y}(k))^2}_{\varepsilon^2(k)} + \lambda \theta^T \theta$

$$\hat{\theta}_R = \left(\sum_{k=1}^{N} X(k) X^T(k) + I_n \lambda \right)^{-1} \left(\sum_{k=1}^{N} X(k) y(k) \right)$$

(Regularisering)

$\dim \theta = n$

$$y(k) = X^T(k) \theta_0 + e(k)$$

$$\hat{\theta}_R = R_{(N+\lambda)}^{-1} R_N \theta_0 + R_{(N+\lambda)}^{-1} \sum_{k=1}^{N} X(k) e(k) \quad \text{där} \quad R_{N+\lambda} = \sum_{k=1}^{N} X(k) X^T(k) + I_n \lambda$$

$$\mathrm{cov}(\hat{\theta}_R) = R_{N+\lambda}^{-1} R_N R_{N+\lambda}^{-1} r_e$$

$$\mathrm{Var}(y) = \mathbb{E}(y - \underbrace{\mathbb{E}y}_{m_y})^2$$

Då y är en vektor ingår istället:

$$\mathrm{cov}(y) = \mathbb{E}\left[(y - \underbrace{\mathbb{E}(y)}_{m_y})(y - \underbrace{\mathbb{E}(y)}_{m_y}) \right]$$

Korrelationen mellan två signaler $y(k)$ och $z(k)$:

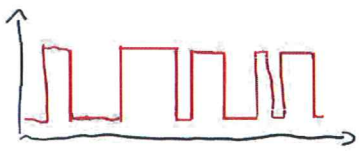
$$\mathbb{E}(y(k) z^T(k))$$

För skalära signaler:

$$\mathbb{E}(y(k) z(k))$$

PRBS-signal

En insignal som ofta används vid parameterskattning är PRBS-signalen (Pseudo-Random Binary Signal)
Denna signalen är approximativt lika med vitt brus men ändrar endast slumpmässigt mellan två signalnivåer.



Linjär regression med flera utsignaler

Regressionsmodell

$$y_i = \Theta^T x_i + \varepsilon_i \qquad i = 1, \dots, N$$

där

$$y_i = \begin{bmatrix} y_{1i} \\ \vdots \\ y_{pi} \end{bmatrix} \quad \varepsilon_i = \begin{bmatrix} \varepsilon_{1i} \\ \vdots \\ \varepsilon_{1p} \end{bmatrix} \quad x_i = [x_{1i} \dots x_{pi}]$$

$$\dim x_i = n \times p$$

Kriterium

$$J(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \|\varepsilon_i\|^2 = \frac{1}{N} \sum_{i=1}^{N} \left(\varepsilon_{1i}^2 + \dots + \varepsilon_{pi}^2 \right)$$

Normalekvation

$$\hat{\Theta} = \left(\sum_{i=1}^{N} x_i x_i^T \right)^{-1} \sum_{i=1}^{N} x_i y_i =$$

$$= \left(\sum_{i=1}^{N} x_{1i} x_{1i}^T + \dots + x_{pi} x_{pi}^T \right)^{-1} \sum_{i=1}^{N} x_{1i} y_{1i} + \dots + x_{pi} y_{pi}$$

3

Optimization

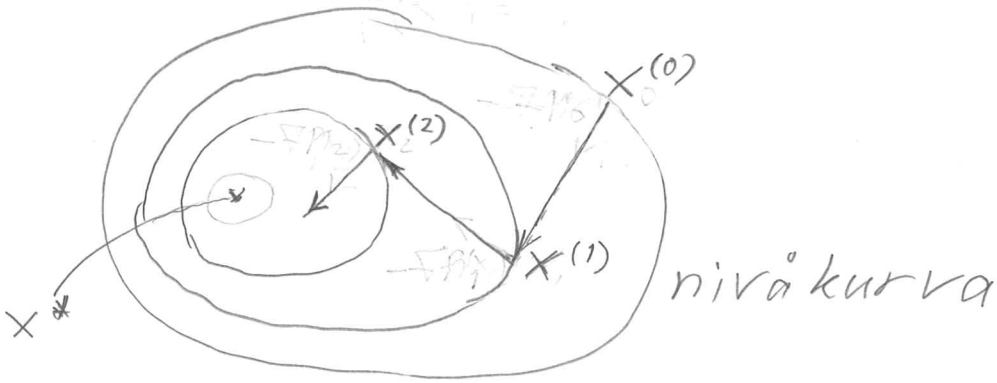
Optimering

Gradient

$$\nabla f(x_1, \dots, x_n) = \operatorname{grad} f(x_1, \dots, x_n) = \left[\frac{\partial f}{\partial x_1} \dots \frac{\partial f}{\partial x_n}\right]^T$$

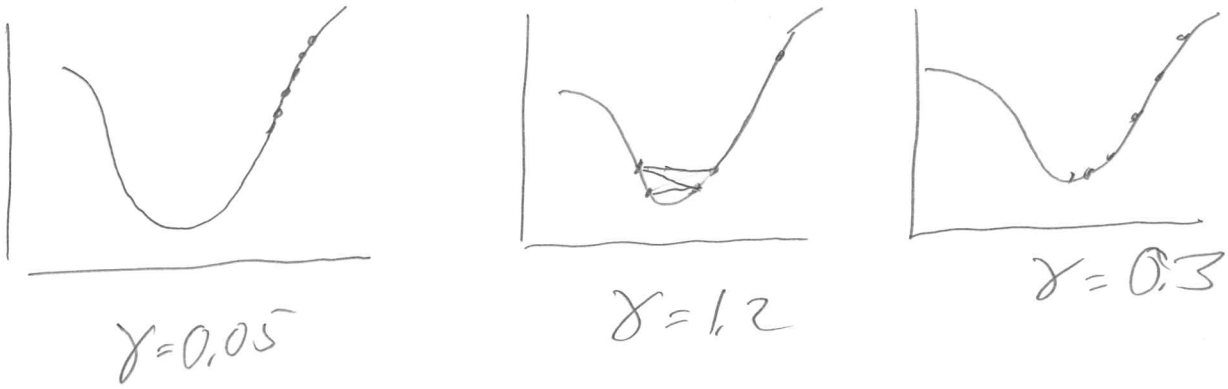
Iterativ optimering

Steepest descent (nedstigande lutning)



$$x^{(k+1)} = x^{(k)} - \gamma \underbrace{\nabla f(x^{(k)})}_{\text{normal vektor}}$$

Learning rate γ



Second order method

Taylor serieutveckling av $f(x+\Delta x)$

$$f(x+\Delta x) \approx f(x) + \nabla f(x)^T \Delta x + \frac{1}{2} \Delta x^T \nabla^2 f(x) \Delta x$$

$$\nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & & \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

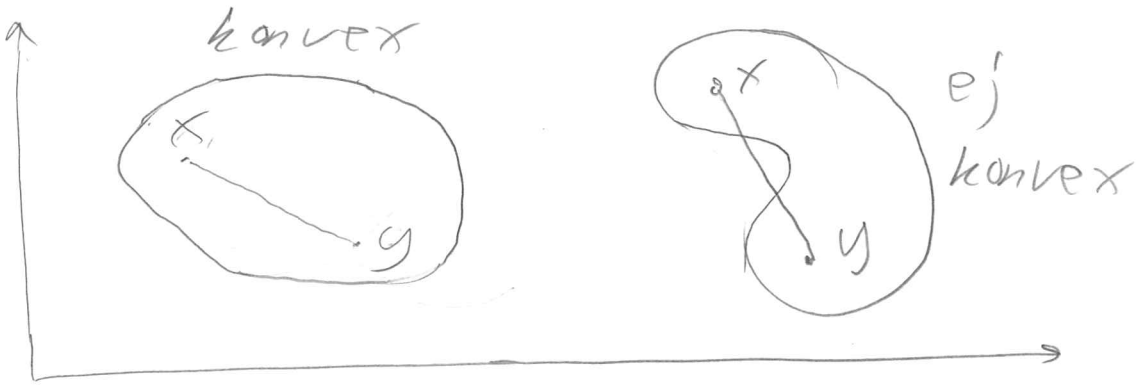
= Hessian matrix

$$\frac{\partial f(x+\Delta x)}{\partial \Delta x} = \nabla f(x) + \nabla^2 f(x) \Delta x = 0$$

at the optimum

Convex set

S is convex if for all $x, y \in S$ and all $\theta \in [0,1]$ we have $\theta x + (1-\theta) y \in S$



För en konvex funktion $f(x)$ är $f''(x) \geq 0$

Då $x \in \mathbb{R}^n$ är $f(x)$ konvex då $\nabla^2 f(x) \geq 0$, i.e. Hessianen är positivt semidefinit.

$$\Delta x = -\left[\nabla^2 f(x)\right]^{-1} \nabla f(x)$$

$$x^{(k+1)} = x^{(k)} - \gamma \left[\nabla^2 f(x^{(k)})\right]^{-1} \nabla f(x^{(k)})$$

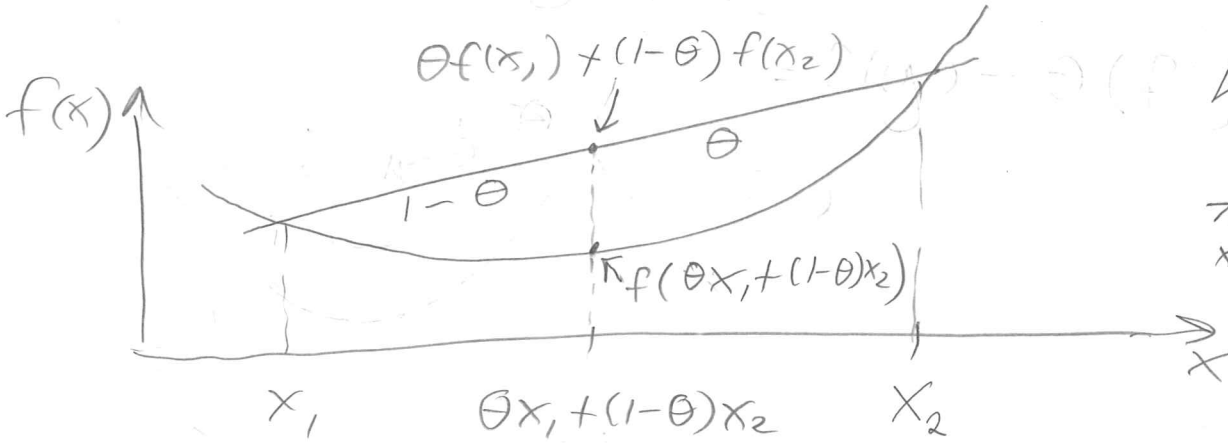
This is called Newton's method

Convex optimization.

Objective function is convex
Feasible set is a convex set

A function f is convex if for all $\theta \in [0,1]$ and all x, y in its domain

$$f(\theta x_1 + (1-\theta) x_2) \leq \theta f(x_1) + (1-\theta) f(x_2)$$

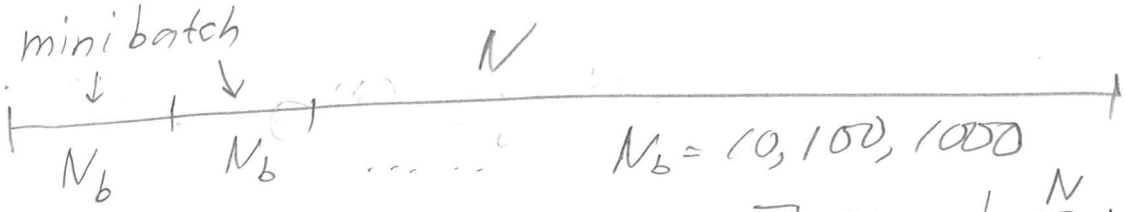


konkav

$\leq$ ersätts med $\geq$

Konvexa funktioner: x^2, x^4, $|x|$, e^x, $\frac{1}{x^2}$ $-\infty, 0$ och $0, \infty$ ej $-\infty, \infty$

Stochastic Gradient Descent



$N_b = 10, 100, 1000$

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(x_i, y_i, \theta)$$

Inner loop

for $j = 1, 2, \ldots, \frac{N}{N_b}$

$$\hat{d}^{(k)} = \frac{1}{N_b} \sum_{i=(j-1)N_b+1}^{jN_b} \nabla L(x_i, y_i, \theta^{(k)})$$

$$\theta^{(k+1)} = \theta^{(k)} - \gamma^{(k)} \hat{d}^{(k)}$$

$k := k+1$

end

Batch Gradient Descent

$$\hat{d}^{(k)} = \frac{1}{N} \sum_{i=1}^{N} \nabla L(x_i, y_i, \theta^{(k)})$$

Optimering med bivillkor

Ex $\min f(x_1, x_2) = x_1^2 + x_2$

bivillkor $h(x_1, x_2) = 2 - x_1 - 0{,}5x_2 = 0$

studera gradienterna

$$\nabla f(x_1, x_2) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right) = (2x_1, 1)$$

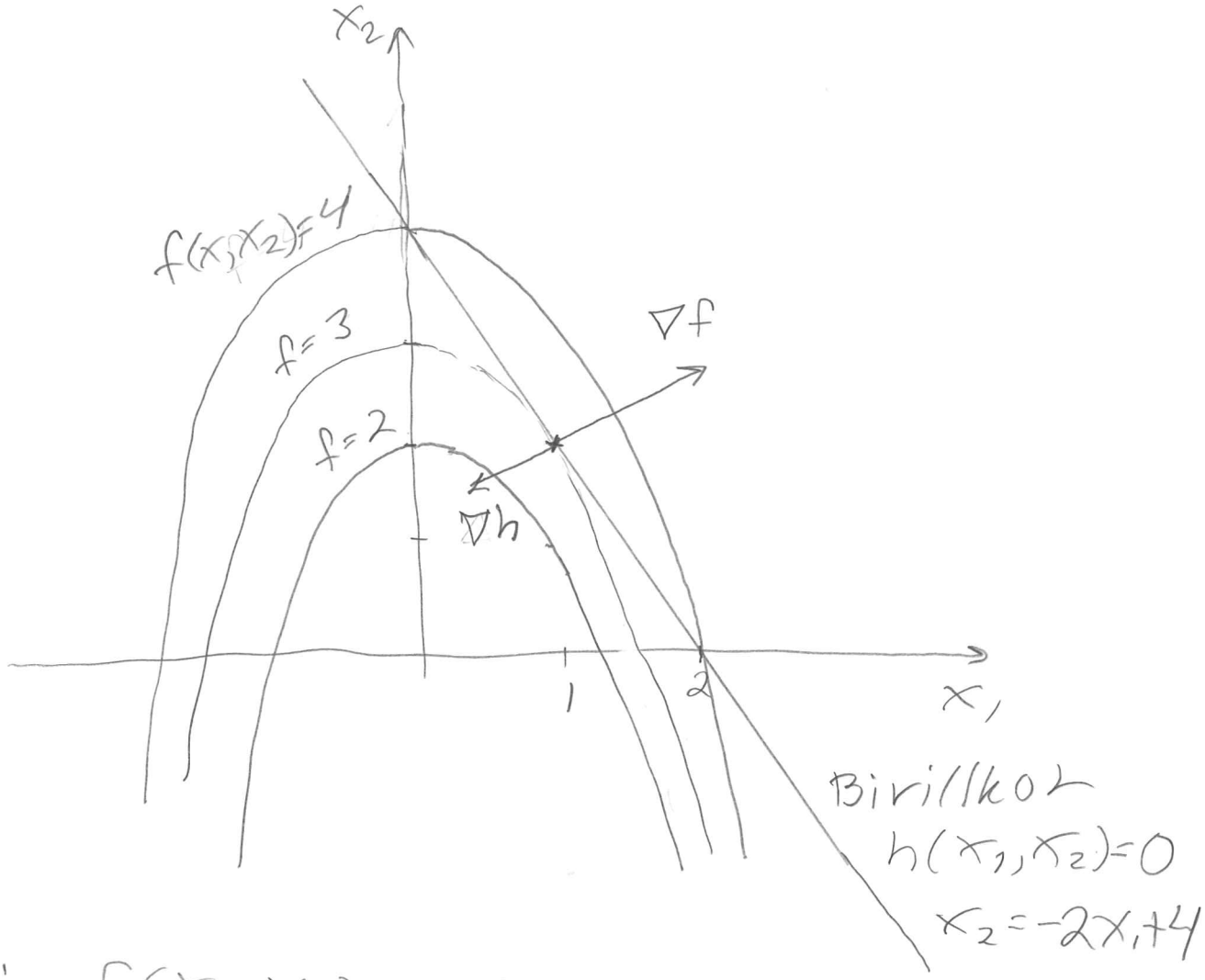
$$\nabla h(x_1, x_2) = \left(\frac{\partial h}{\partial x_1}, \frac{\partial h}{\partial x_2}\right) = (-1, -0{,}5)$$

∇f blir parallell med ∇h för $x_1 = 1$ eftersom

$$\nabla f = (2, 1) = -2(-1, -0{,}5) = -2\nabla h$$

vid extrempunkter, min eller max, gäller alltid att $\nabla f = -\lambda \nabla h$ λ = skalär

dvs $\boxed{\nabla f + \lambda \nabla h = 0}$



Min $f(x_1,x_2)$ som

tangerar bivillkoret $h(x_1,x_2)=0$

ges i detta exempel för

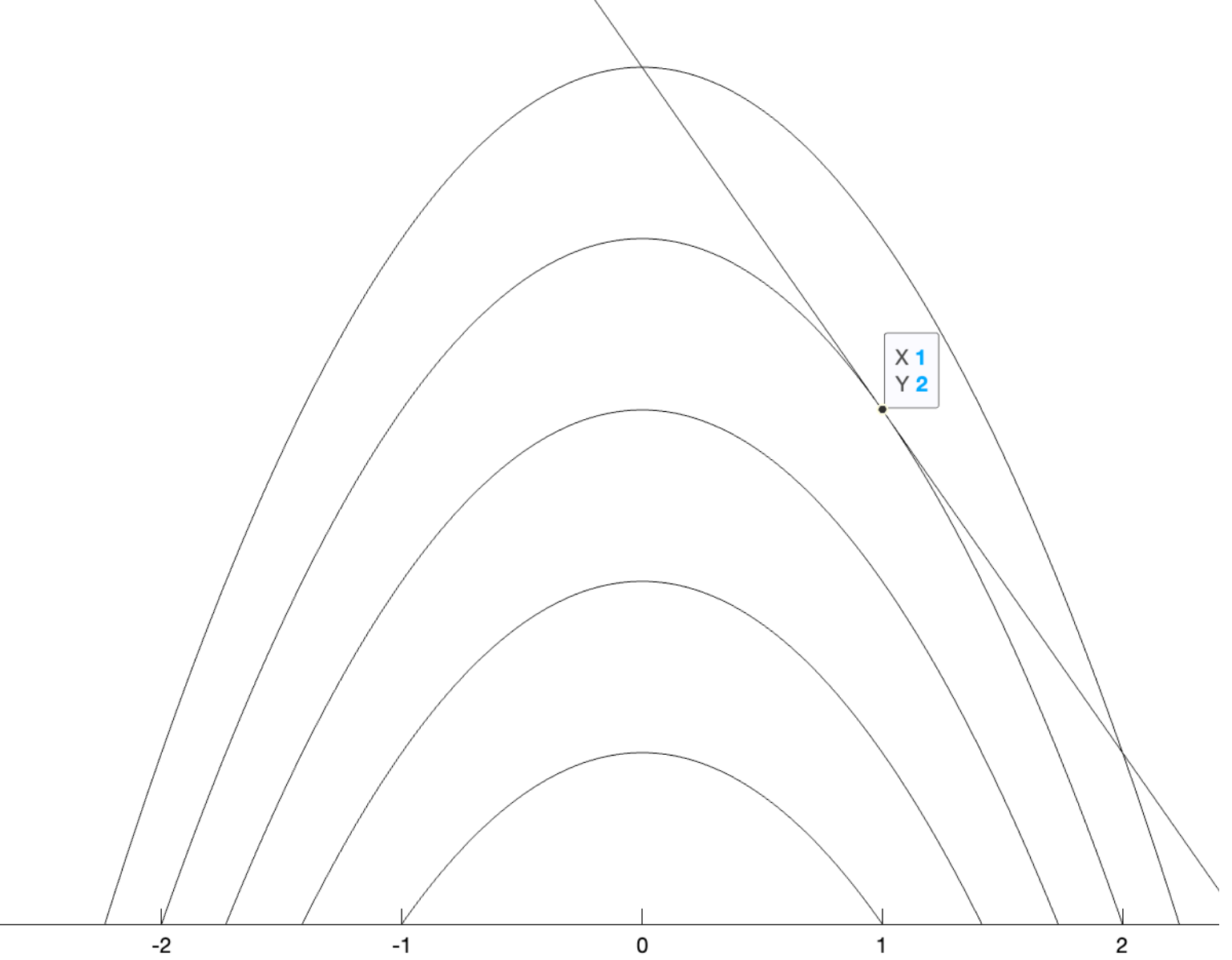
$x_1=1$ enligt villkoret

$\nabla f+\lambda\nabla h=0$ vilket för

$\lambda=2 \Rightarrow (2x_1,1)+2(-1,-0.5)=0$

för $x_1=1$

$h(x_1,x_2)=2-x_1-0.5x_2=0 \Rightarrow x_2=2$


X 1
Y 2
-2
-1
0
1
2

Detta resultat fås också med införande av Lagrange-multiplikator. Optimerings-problemet $\min f(x)$

bivillkor $h(x)=0$

Löses genom att minimera Lagrange funktionen

$$\boxed{L(x,\lambda) = f(x) + \lambda h(x)}$$

$$\nabla_x L = \nabla_x f(x) + \lambda \nabla_x h(x) = 0 \qquad (1)$$

$$\nabla_\lambda L = h(x) = 0 \qquad (2)$$

(1) innebär att ∇f & ∇h är parallella

(2) ger bivillkoret.

Ex $\quad f = x_1^2 + x_2$

$\quad h = 2 - x_1 - 0.5x_2$

$$L = f + \lambda h = x_1^2 + x_2 + \lambda(2 - x_1 - 0.5x_2)$$

$$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda = 0 \Rightarrow \lambda = 2x_1$$

$$\frac{\partial L}{\partial x_2} = 1 - 0.5\lambda = 0 \Rightarrow \lambda = 2 \Rightarrow x_1 = 1$$

$$\frac{\partial L}{\partial \lambda} = h = 2 - x_1 - 0.5x_2 = 2 - 1 - 0.5x_2 = 0$$

$$\Rightarrow x_2 = 2$$

Minimum eftersom

$$\nabla^2 f = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{pmatrix} \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \end{pmatrix} [2x_1 \quad 1]$$

$$= \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} \geq 0 \qquad \text{Max} \Rightarrow \nabla^2 f \leq 0$$

4

Classification

An example of feature-based data

Hair	Height	Weight	Lotion	Result
blonde	average	light	no	sunburned
blonde	tall	average	yes	none
brown	short	average	yes	none
blonde	short	average	no	sunburned
red	average	heavy	no	sunburned
brown	tall	heavy	no	none
brown	average	heavy	no	none
blonde	short	light	yes	none

Table 1.1. An example of feature-based data.

We use a dataset in (Winston, 1992) to explain what are features, different types of features, classes, and instances. The slightly modified dataset can be seen in Table 1.1. There are a total of five features, one of which is of class *Result*. The class feature has two values: sunburned or none. Each row in the table is an instance. All features are of discrete values. However, features Height and Weight can be readings of measurements. In that case, the two features are of continuous values. The discrete values of features *Height* and *Weight* can be the result of manual discretization of the measurements. For example, we can define an adult's height as *short* if the measurement is less than or equal to 1.5 meters, as *average* if it is greater than 1.5 meters and less than or equal to 1.8 meters, and as *tall* if it is greater than 1.8 meters.

Huan Liu and Hiroshi Motoda. Feature selection for knowledge discovery and data mining. Kluwer international series in engineering and computer science, 1998.

Klassificering

Klassiskt exempel: Känna igen handskrivna siffror (MNIST)
Motsvarar Hello World som standard programmeringsexempel

Förenkla teckenigenkänning:
Klassificera bokstaven E eller F
Antag 5×4 pixlar

E

```
x x x x
x
x x x
x
x x x x
```

F

```
x x x x
x
x x x
x
x
```

Feature vektor

E $x_1 = [1\ 1\ 1\ 1\ 1\ 0\ 0\ 0 \ldots 1\ 1\ 1\ 1]$

F $x_2 = [$ — " — $1000]$

Testdata

1) $x = [$ — " — $1100] \Rightarrow \hat{y} = F$

2) $x = [$ — " — $1110] \Rightarrow \hat{y} = E$

1) $\|x_1 - x\| = \sqrt{0^2 + \dots + 0^2 + 1^2 + 1^2} = \sqrt{2}$

$\|x_2 - x\| = \sqrt{0^2 + \dots + 0^2 + (-1)^2 + 0^2 + 0^2} = 1$

x närmare x_2 än $x_1 \Rightarrow \hat{y} = F$

2) $\|x_1 - x\| = \sqrt{0^2 + \dots + 0^2 + 1^2} = 1$

$\|x_2 - x\| = \sqrt{0^2 + \dots + 0^2 + (-1)^2 + (-1)^2 + 0^2} = \sqrt{2}$

x närmare x_1 än $x_2 \Rightarrow \hat{y} = E$

Feature vektorer

Bildbehandling: varje pixel motsvarar ett element i x

2000 pixlar $\leftrightarrow$ dim $x = 2000$

Färg $\Rightarrow$ gånger 3 (Intensitet (röd, grön, blå))

Ordbehandling exempelvis

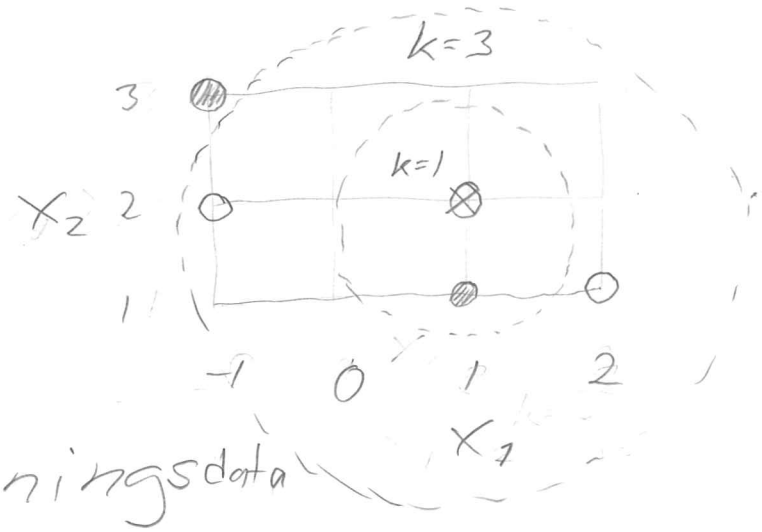
Översättning, ChatGPT

varje ord repr. av en vektor med 512 reella tal $\Rightarrow$

mening med 20 ord dim $x = 10240$

Klassificering: k-närmaste granna

Engelska: k-nearest neighbours (k-NN)



$\hat{y}(x) = ?$

Test $x^T = [1 \;\; 2]$

$\sqrt{(x_{i1}-x_1)^2+(x_{i2}-x_2)^2}$

Träningsdata

i	x_i^T	y_i	$\|x_i - x\|_2$
1	$[-1 \;\; 3]$	Svart	$\sqrt{5}$
2	$[-1 \;\; 2]$	Vit	$\sqrt{4}$
3	$[1 \;\; 1]$	Svart	$\sqrt{1}$
4	$[2 \;\; 1]$	Vit	$\sqrt{2}$

$k=1 \Rightarrow \hat{y}(x) =$ Svart

$k=3 \Rightarrow \hat{y}(x) =$ Vit 2 vita, 1 svart av de 3 närmaste $\Rightarrow$ majoritet vita

Träningsdata $\{x_i, y_i\}_{i=1}^{N}$

Bestäm $\hat{y}(x)$

Algoritm: k-närmaste grannar
k-nearest neighbours (k-NN)

1. Beräkna avståndet $d_i = \|x_i - x\|$
för alla träningsdata $i \in [1, N]$

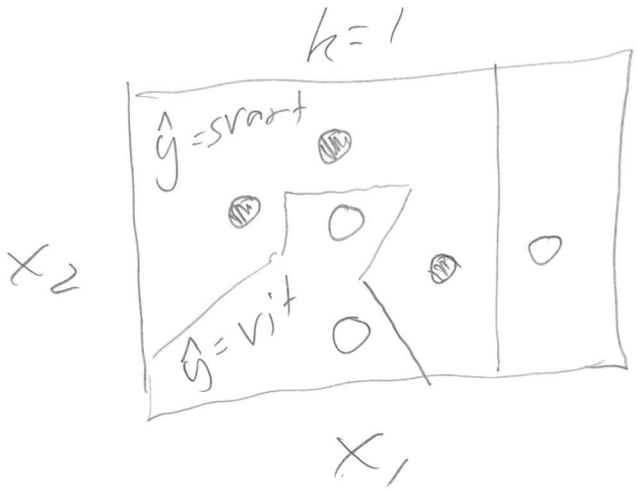
2. Sortera avstånden d_i till
testdata x efter storleksordning
(minst först)

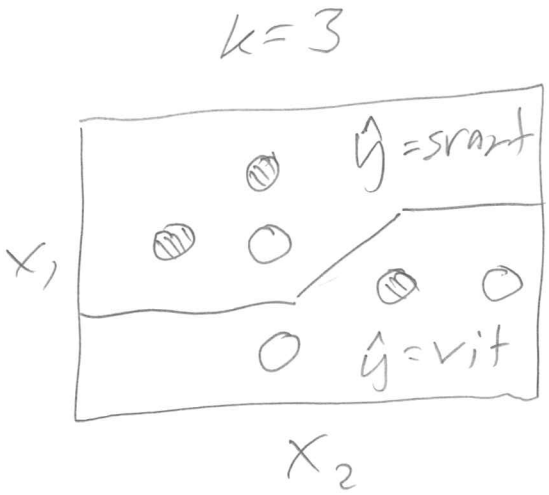
3. $NN = \{ i \mid d_i$ är en de k
kortaste avstånden$\}$

dvs. Välj de k första avstånden
i den sorterade listan

4. $\hat{y}(x) = \text{Majoriteten}\{ y_i \mid i \in NN \}$
dvs Välj den mest frekventa
klassen bland de k första
avstånden.

k-NN är en olinjär klassificerare





Ökat k ger bias men mindre varians, se Fig 2.5 i boken

Naiv Bayesiansk klassificering

skattning av en klass y givet data $x = \{x_1, \dots x_n\}$

prior class prob

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

(tidigare sannolikhet)

Posterior class prob (sannolikhet inkluderande ny data)

Naiv Bayesiansk klassificering förenklar problemet genom att anta att features $x = [x_1 \dots x_n]$ för en klass y_m är oberoende av varandra

dvs

$$p(x|y_m) = \prod_{j=1}^{n} p(x_j|y_m)$$

$$p(y_m|x_1,\dots,x_n) = \frac{\prod_{j=1}^{n} p(x_j|y_m)\,p(y_m)}{\sum_{m=1}^{M} \prod_{j=1}^{n} p(x_j|y_m)\,p(y_m)}$$

$$\hat{y} = \underset{y_m}{\operatorname{argmax}}\; p(y_m|x_1,\dots,x_n)$$

Play golf: Yes (Y) or No (N)

Classes: $P(Y) = 9/14$ $P(N) = 5/14$

Features:

Outlook
$$\begin{cases} p(\text{sunny}|Y) = 2/9 & p(\text{sunny}|N) = 3/5 \\ p(\text{overcast}|Y) = 4/9 & p(\text{overcast}|N) = 0/5 \\ p(\text{Rainy}|Y) = 3/9 & p(\text{Rainy}|N) = 2/5 \end{cases}$$

Temp
$$\begin{cases} p(\text{Hot}|Y) = 2/9 & p(\text{Hot}|N) = 2/5 \\ p(\text{Mild}|Y) = 4/9 & p(\text{Mild}|N) = 2/5 \\ p(\text{Cold}|Y) = 3/9 & p(\text{Cold}|N) = 1/5 \end{cases}$$

Humidity
$$\begin{cases} p(\text{Normal}|Y) = 6/9 & p(\text{Normal}|N) = 1/5 \\ p(\text{High}|Y) = 3/9 & p(\text{High}|N) = 4/5 \end{cases}$$

Wind
$$\begin{cases} p(\text{NoWind}|Y) = 6/9 & p(\text{NoWind}|N) = 2/5 \\ p(\text{Wind}|Y) = 3/9 & p(\text{Wind}|N) = 3/5 \end{cases}$$

Today1 = {Sunny, Hot, Normal, No wind}

Today2 = {Sunny, Hot, High, No wind}

Today3 = {Sunny, Hot, Normal, Wind}

$$p(Y|Today1) = \frac{2\cdot2\cdot6\cdot6/9^4\cdot 9/14}{2\cdot2\cdot6\cdot6/9^4\cdot 9/14 + 3\cdot2\cdot1\cdot2/5^4\cdot 5/14}$$

$$= 0.673$$

$$p(N|Today1) = \frac{3\cdot2\cdot1\cdot2/5^4\cdot 5/14}{2\cdot2\cdot6\cdot6/9^4\cdot 9/14 + 3\cdot2\cdot1\cdot2/5^4\cdot 5/14}$$

$$= 0.327$$

$\hat{y}(Today1) = Y$ since $0.673 > 0.327$

$p(Y|Today2) = 0.205$

$p(N|Today2) = 0.795$

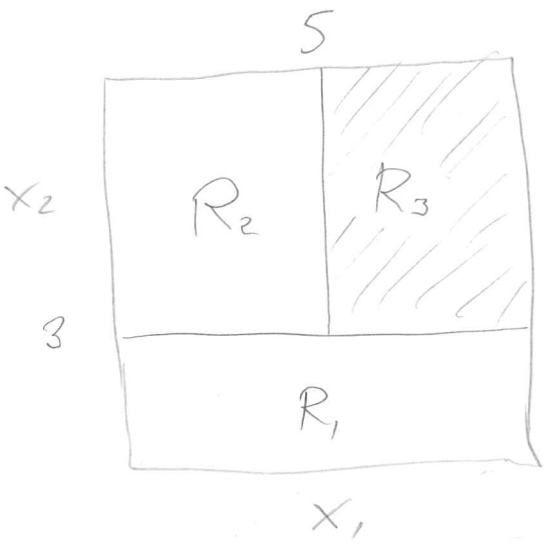
$\hat{y}(Today2) = N$ since $0.205 < 0.795$

$p(Y|Today3) = 0.407$

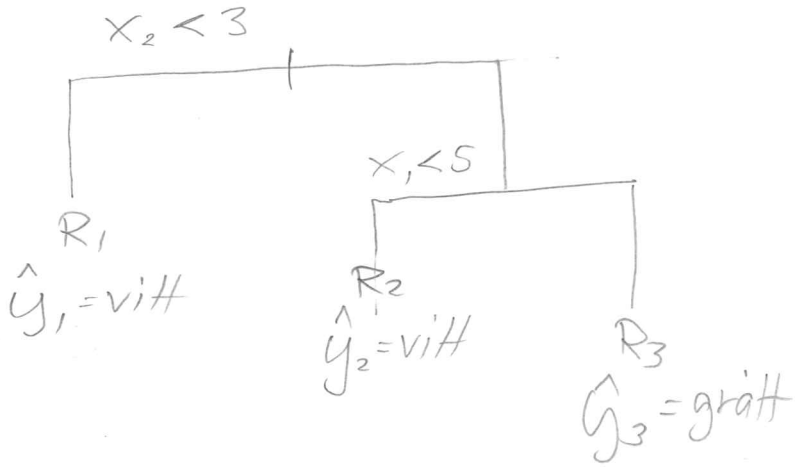
$p(N|Today3) = 0.593$

$\hat{y}(Today3) = N$ since $0.407 < 0.593$

Klassificering baserat på beslutsträd



Beslutsträd



Interna noder: $x_2 < 3$, $x_1 < 5$

Lövnoder: R_1, R_2, R_3

Antag n insignal variabler

$\{x_j \mid j = 1, \dots, n\}$ $\quad \mathbf{x}^T = (x_1, \dots, x_j, \dots, x_n)$

För varje intern nod införs två halvmängder

$$R_1(j,s) = \{\mathbf{x} \mid x_j < s\}$$

$$R_2(j,s) = \{\mathbf{x} \mid x_j \geq s\}$$

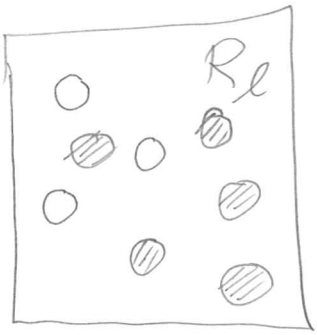
j = index insignalkomponent x_j

s = insignal värde

$$\hat{y}_\ell = \text{majoritet}\{y_i \mid \mathbf{x}_i \in R_\ell\}$$

lövnod

i = index för alla n-dim punkter $\mathbf{x}_i$ som tillhör lövnoden ℓ.



$\hat{y}_\ell$ = svart

Vid varje uppdelning i vänster och höger halvmängder välj

$$(j,s) = \arg\min_{(j,s)} (n_1 Q_1 + n_2 Q_2)$$

där n_1, n_2 = antalet träningsdata i vänstra respektive högra regionen och Q_1, Q_2 är kostnadsfunktionen för respektive region

Litet Q-värde $\Leftrightarrow$ få punkter av fel klass

$$\hat{\pi}_{\ell m} = \frac{1}{n_\ell} \sum_{i:\, \mathbf{x}_i \in R_\ell} \mathbb{I}\{y_i = m\} =$$

$= 1$ då $y_i = m$
0 då $y_i \neq m$

= procentuella andelen av observationerna i lövnod ℓ som tillhör klass m

Tre olika kriterier

1) Missklassificering

$$Q_\ell = 1 - \max_m \hat{\pi}_{\ell m}$$ vanligaste klassen

= andelen datapunkter i region R_ℓ som inte tillhör den vanligaste klassen

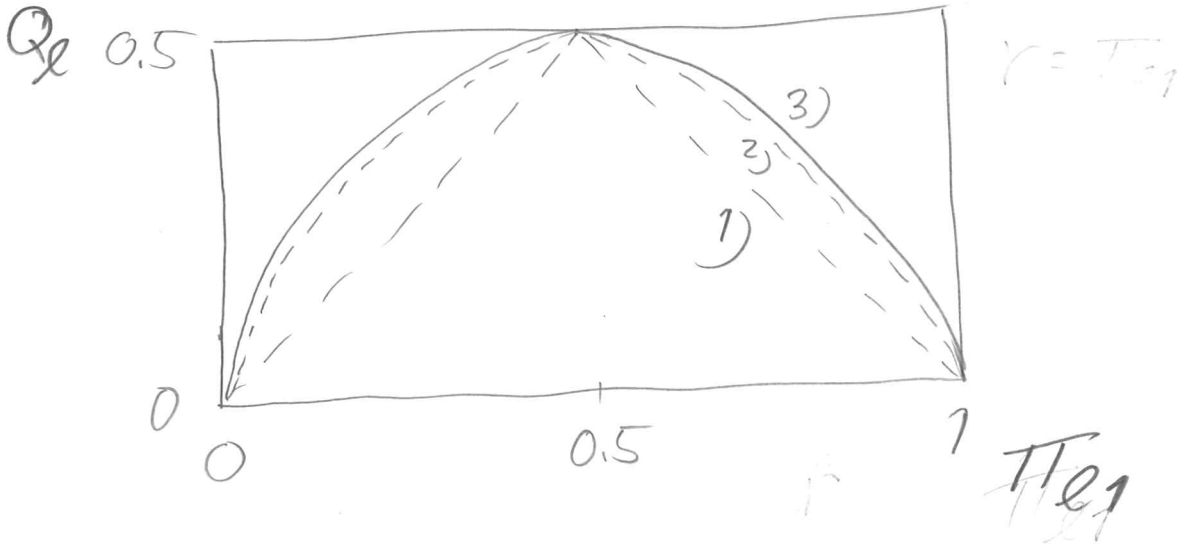
2) Gini index

$$Q_\ell = \sum_{m=1}^{M} \hat{\pi}_{\ell m}(1-\hat{\pi}_{\ell m})$$

3) Entropi index

$$Q_\ell = -\sum_{m=1}^{M} \hat{\pi}_{\ell m} \ln \hat{\pi}_{\ell m}$$

Consider only two classes $m = 1, 2$



2) & 3) gynnar regioner med fler punkter av samma klass ("renare" klasser)

	x_1	x_2	y
R_1	1	4	svart
	2	2	svart
R_2	1	3	vit
	3	3	svart
	3	4	vit

Undersök uppdelningen

$R_1(2, 2.5) = \{x_2 | < 2.5\}$ $\quad R_2(2, 2.5) = \{x_2 \geq 2.5\}$

$\hat{\pi}_{1\,svart} = 1 \qquad \hat{\pi}_{1\,vit} = 0$

$\hat{\pi}_{2\,svart} = 1/3 \qquad \hat{\pi}_{2\,vit} = 2/3$

Gini index

$$Q_1 = \hat{\pi}_{1\,svart}(1 - \hat{\pi}_{1\,svart}) + \hat{\pi}_{1\,vit}(1 - \hat{\pi}_{1\,vit})$$

$$= 1 \cdot 0 + 0 \cdot 1 = 0$$

$$Q_2 = \hat{\pi}_{2\,svart}(1 - \hat{\pi}_{2\,svart}) + \hat{\pi}_{2\,vit}(1 - \hat{\pi}_{2\,vit})$$

$$= 1/3 \cdot 2/3 + 2/3 \cdot 1/3 = 4/9$$

$$n_1 Q_1 + n_2 Q_2 = 2 \cdot Q_1 + 3 \cdot Q_2 = 2 \cdot 0 + 3 \cdot \frac{4}{9} = 1.33$$

$x_2 = 3.5 \qquad Q_1 = \frac{3}{4} \cdot \frac{1}{4} \cdot 2 = \frac{6}{16} \qquad Q_2 = 0$

$$n_1 Q_1 + n_2 Q_2 = 4 \frac{6}{16} = \frac{6}{4} = 1.5$$

$x_2 = 1.5 \qquad Q_1 = 0 \qquad Q_2 = \frac{1}{2} \cdot \frac{1}{2} \cdot 2 = \frac{1}{2}$

$$n_1 Q_1 + n_2 Q_2 = 4 \cdot \frac{1}{2} = 2$$

$x_1 = 2.5 \qquad Q_1 = \frac{1}{4} + \frac{1}{4} = \frac{1}{2} \qquad Q_2 = \frac{2}{3} \frac{1}{3} \cdot 2 = \frac{4}{9}$

$$n_1 Q_1 + n_2 Q_2 = 2 \frac{1}{2} + 3 \cdot \frac{4}{9} = 1 + \frac{4}{3} = \frac{7}{3}$$

	Q_1	Q_2	$n_1 Q_1 + n_2 Q_2$
$x_1 < 1.5$	0	3/8	1.5
$x_1 < 2.5$	1/2	4/9	2.33
$x_1 < 3.5$	1/2	0	2
$x_2 < 1.5$	0	1/2	2
$x_2 < 2.5$	0	4/9	1.33 minsta värde
$x_2 < 3.5$	3/8	0	1.5

$\therefore$ optimal första indelning är $x_2 < 2.5$ enligt Gini index

uppdelningen i nya noder fortsätter tills ett stopp-kriterium aktiveras antingen ett bestämt djup i trädet eller att en region bara innehåller ett visst mindre antal datapunkter.

För djupt träd motsvarar ett litet k i k-NN dvs hög varians och overfitting.

Logistisk regression

Antag M klasser, där klassen $m = 1, \dots, M$

Skatta den betingade sannolikhets-fördelningen

$p(y=m|x)$ givet data (y_i, x_i) $i = 1, \dots, N$

För varje $p(y=m|x)$ skattas en funktion $g_m(x)$ som beror på en linjär regression

$$\theta^T x = \theta_0 + \theta_1 x_1 + \dots + \theta_n x_n$$

Här behövs ingen osäkerhetsterm ε eftersom slumpmässigheten modelleras med fördelningsfunktionen $p(y=m|x)$.

specialisering till binär klassificering innebär att

$p(y=1|x) = g(x)$ och $p(y=-1|x) = 1 - g(x)$

Eftersom en fördelningsfunktion har värdemängden $[0, 1]$ väljs

$g(x) = \dfrac{e^{\theta^T x}}{1 + e^{\theta^T x}}$ som för alla $\theta^T x$ ger ett värde $\in [0, 1]$

För $y=1$ gäller dessutom att

$$p(y=1|x) = g(x) = \frac{e^{y\theta^T x}}{1+e^{y\theta^T x}} = \frac{1}{1+e^{-y\theta^T x}}$$

medan

$$p(y=-1|x) = 1-g(x) = 1 - \frac{e^{\theta^T x}}{1+e^{\theta^T x}} = \frac{1}{1+e^{\theta^T x}}$$

$$= \frac{1}{1+e^{-(-1)\theta^T x}} = \frac{1}{1+e^{-y\theta^T x}}$$

dvs både för $y=1$ o $y=-1$ kan fördelningsfunktionen uttryckas som

$$p(y|x) = \frac{1}{1+e^{-y\theta^T x}}$$

och dessutom gäller att

$$-\ln p(y|x) = -\ln \frac{1}{1+e^{-y\theta^T x}} = \ln(1+e^{-y\theta^T x}) \quad (1)$$

Maximum likelihood är en generell metod för att maximera en fördelningsfunktion mha en parameter-vektor θ. Här gäller då datapunkterna x_i är oberoende av varandra att

$$\hat{\theta} = \arg\max_{\theta} p(y|X;\theta) =$$

$$= \arg\max_{\theta} \prod_{i=1}^{N} p(y_i|x_i;\theta)$$

Eftersom logaritmfunktionen är en monotont växande funktion gäller att $\arg\max_\theta f(\theta) = \arg\max_\theta \ln f(\theta)$

$$\therefore \hat{\theta} = \arg\max_\theta \ln \prod_{i=1}^{N} p(y_i | x_i; \theta) =$$

$$= \arg\max_\theta \sum_{i=1}^{N} \ln p(y_i | x_i; \theta) =$$

$$= \arg\min_\theta \sum_{i=1}^{N} -\ln p(y_i | x_i; \theta)$$

$$= \text{enligt } (1) = \arg\min_\theta \sum_{i=1}^{N} \ln(1 + e^{-y_i \theta^T x_i})$$

$$\boxed{\hat{\theta} = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \ln(1 + e^{-y_i \theta^T x_i}) \quad (2)}$$

Ingen analytisk lösning men effektiv numerisk optimering löser detta problem.

Prediktor givet $\hat{g}(x_*) = \frac{1}{1+e^{-\hat{\theta}^T x}}$

$$\hat{y}(x_*) = \begin{cases} 1 & \text{då } \hat{g}(x_*) \geq 0.5 \\ -1 & \text{då } \hat{g}(x_*) < 0.5 \end{cases}$$

Klassifiseringsgränsen erhålls då

$g(x) = 1 - g(x)$ dvs då

$$g(x) = \frac{1}{1+e^{-\theta^T x}} = \frac{1}{1+e^{\theta^T x}} = 1 - g(x)$$

vilket erhålls då $\theta^T x = 0$

dvs ett linjärt hyperplan

och därmed en linjär klassifiserare

En binär klassificerare

baserat på logistisk regression

kan därför uttryckas som

$$\hat{y}(x_*) = \text{sign}(\theta^T x_*)$$

där $\hat{\theta}$ bestäms av (2)

M klasser innebär att

M stycken parameter vektorer θ_m

$m \in [1, M]$ införs, dvs $\theta = \{\theta_1, \dots \theta_M\}$

och $$g_m(x;\theta) = \frac{e^{\theta_m^T x}}{\sum_{j=1}^{M} e^{\theta_j^T x}}$$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \underbrace{\left(-\theta_{y_i}^T x_i + \ln \sum_{j=1}^{M} e^{\theta_j^T x_i} \right)}_{= -\ln g_{y_i}(x_i;\theta)}$$

$$\hat{y}(x_*) = \arg\max_{m} g_m(x_*)$$

Ex se wikipedia Logistic regression

Utvärdering av binära klassificerare

Confusion (förvirrings) matrix

		predikterat utfall	
		$\hat{y}=1$	$\hat{y}=-1$
Faktiskt utfall	$y=1$	TP = true positive	FN = false negative
	$y=-1$	FP = false positive	TN = true negative

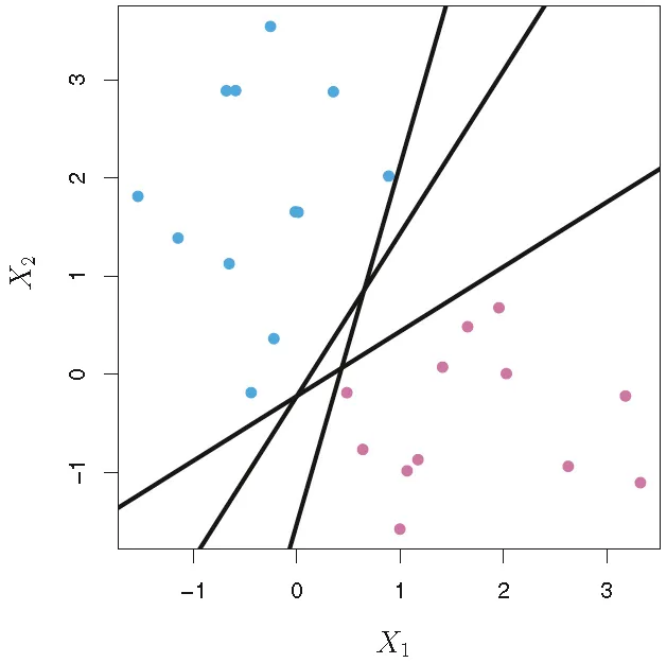
vid sjukdomsundersökning är prognos frisk $\hat{y}=1$ men i verkligheten sjuk patient $y=-1$ dvs FP mycket allvarligare än tvärt om (FN)

Ex

	1	2	3	4	5	6	7	8
Faktiskt utfall	1	1	1	1	1	-1	0	0
Predikterat utfall	-1	-1	1	1	1	1	-1	-1
Resultat	FN	FN	TP	TP	TP	FP	TN	TN

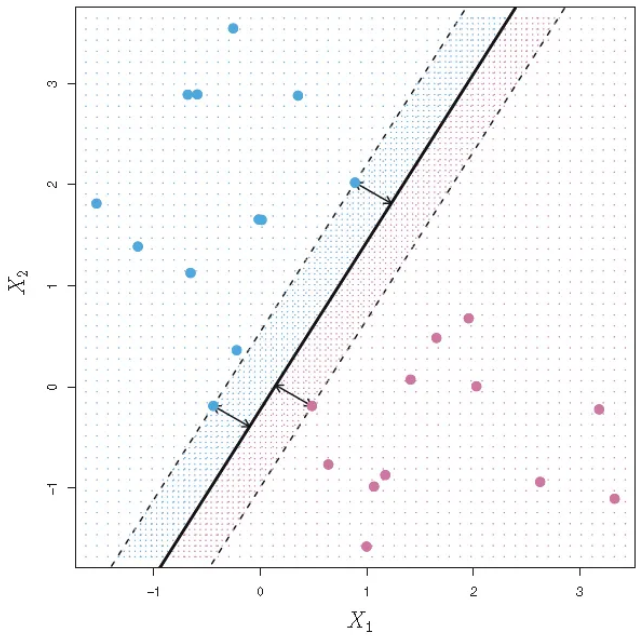
Support Vector Machines (SVM)

Linear classification



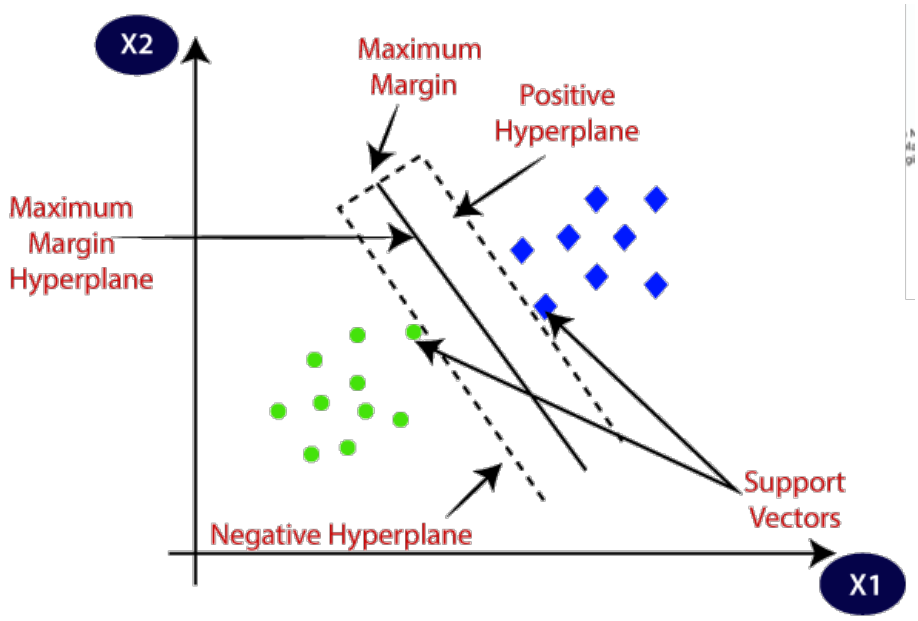
An example dataset showing classes that can be linearly separated.

Support vectors



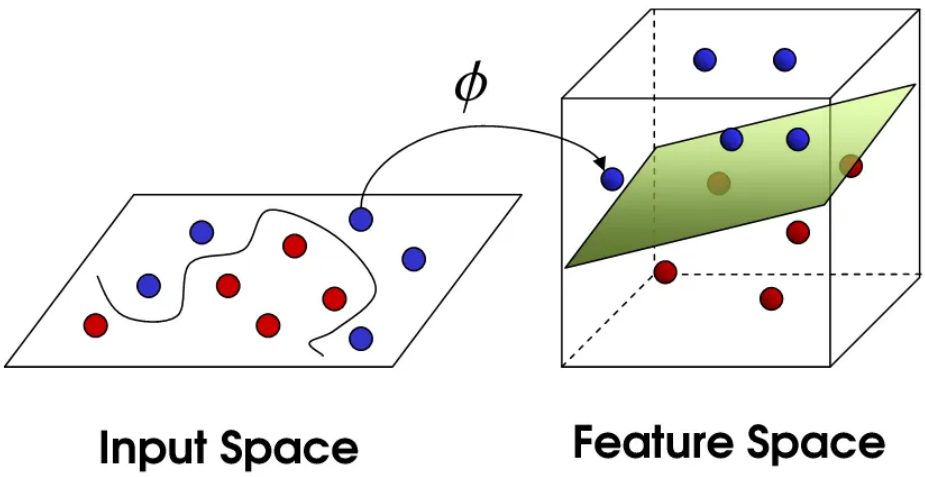
The support vectors are the points on the dashed lines. The distance from the dashed line to the solid line is the margin, represented by the arrows.

SVM means that this margin is maximized.
Solved by quadratic optimization.



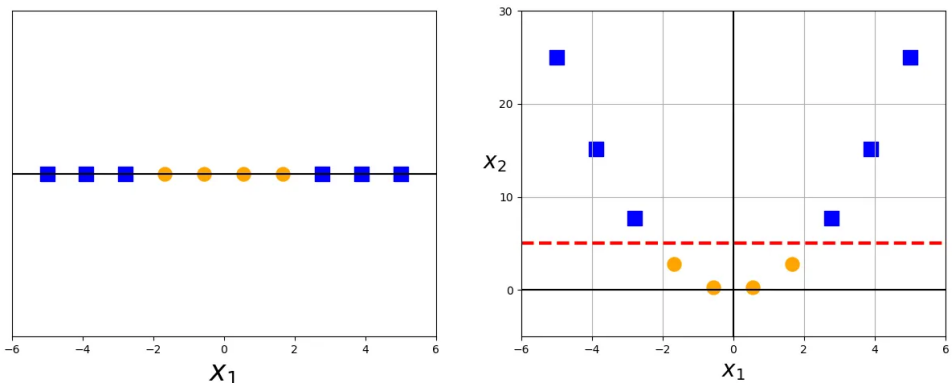
Nonlinear Classification

- Introduced when data are not linearly separable.
- Regression model $y = \theta^T \varphi(\boldsymbol{x})$
- $\varphi(\boldsymbol{x})$ is a nonlinear feature transformation

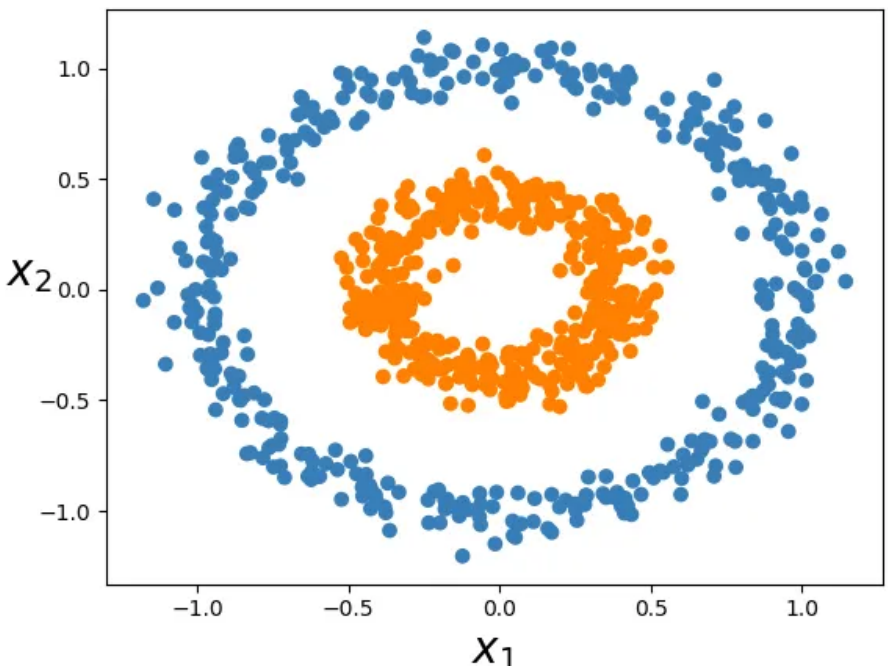


Classes can be linearly separated in a higher dimensional feature space.

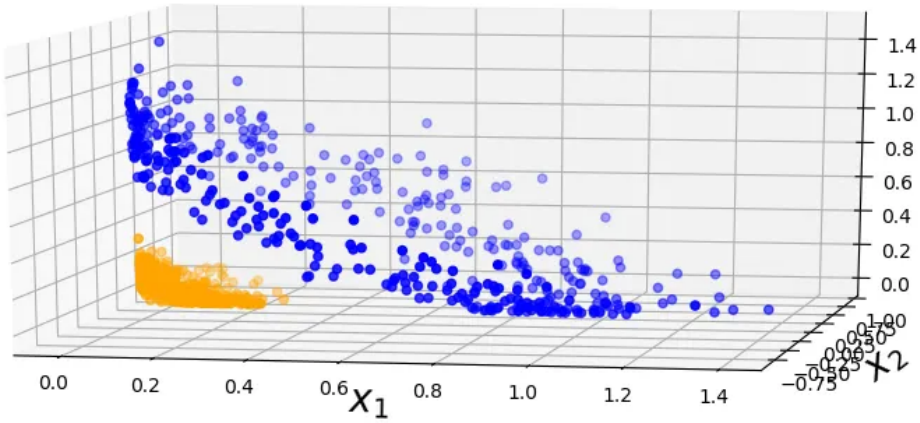
Example 1. Introduce the transformation $\phi(\mathrm{x}) = \mathrm{x}^2$



Example 2. Two-D input to three-D feature space



$$\phi(\mathbf{x}) = \phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} {x_1}^2 \\ \sqrt{2}\, x_1 x_2 \\ {x_2}^2 \end{pmatrix}$$



Linearly separable data in 3-d after applying the 2nd-degree polynomial transformation.

The kernel trick

Main idea: a high dimensional linear classifier or regression curve becomes a non-linear classifier or regression curve in a lower dimension.

Replace the feature transformation $\varphi: n \to n$ with a *kernel function* $k: n \times n \to 1$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \varphi^T(\boldsymbol{x})\varphi(\boldsymbol{x}')$$

and a *kernel matrix*

$$K(i,j) = k(\boldsymbol{x}_i, \boldsymbol{x}'_j), \qquad i,j \in [1, N]$$

where dim(K) = $N \times N$

In Example 2

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T \boldsymbol{x}')^2$$

This is a polynomial kernel. The ordinary regression predictor

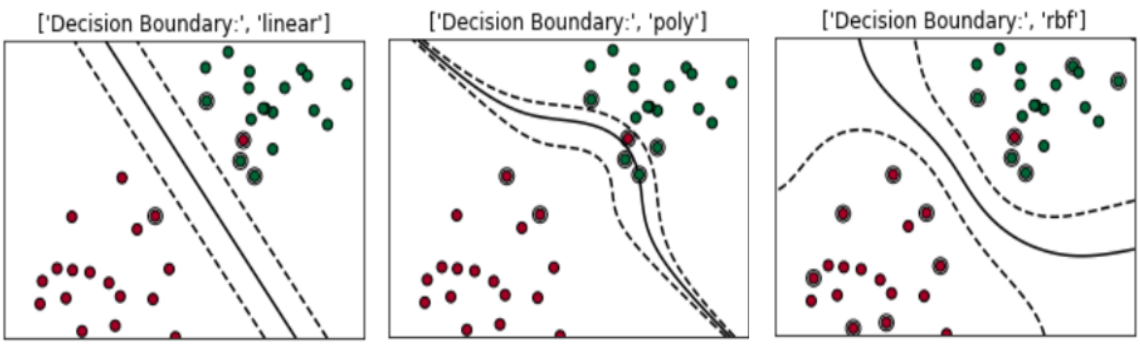
$$\hat{y} = \theta^T \varphi(\boldsymbol{x})$$

can also be expressed as a function of the kernel function. See separate derivation.

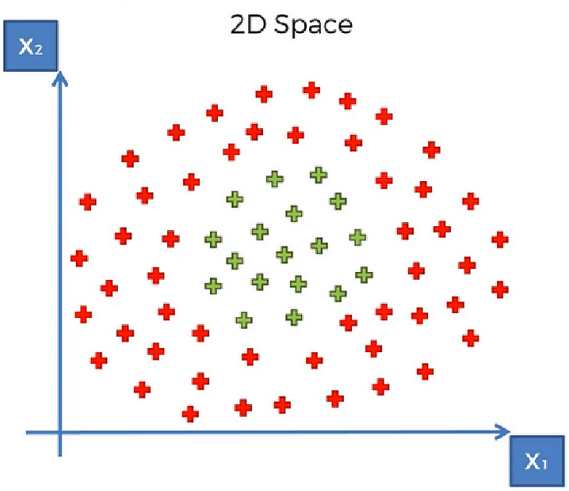
Radial basis function

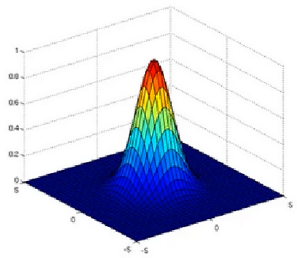
The most common kernel function is the radial basis function (RBF)

$$k(\boldsymbol{x}, \boldsymbol{x}') = exp(-\| \mathbf{x} - \boldsymbol{x}' \|_2^2 / \ell^2)$$



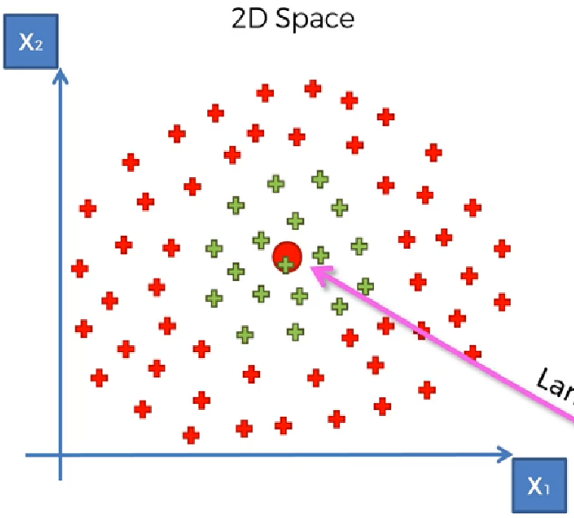
Example 3

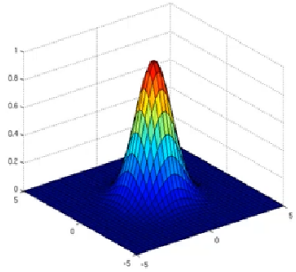




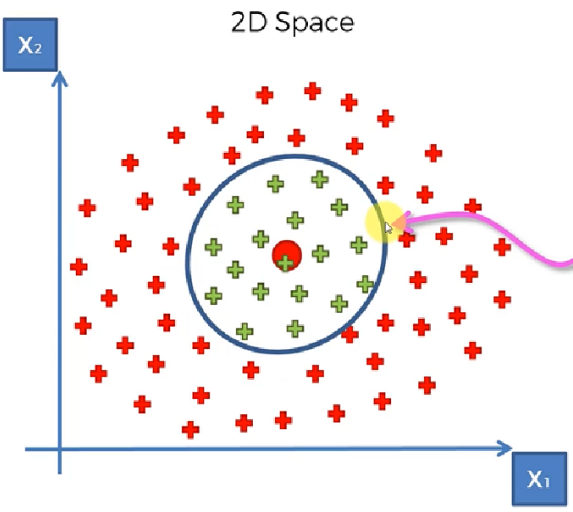
$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

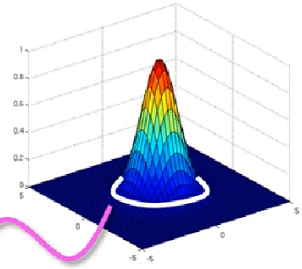
Now, put the landmark and do hyperplane separation.





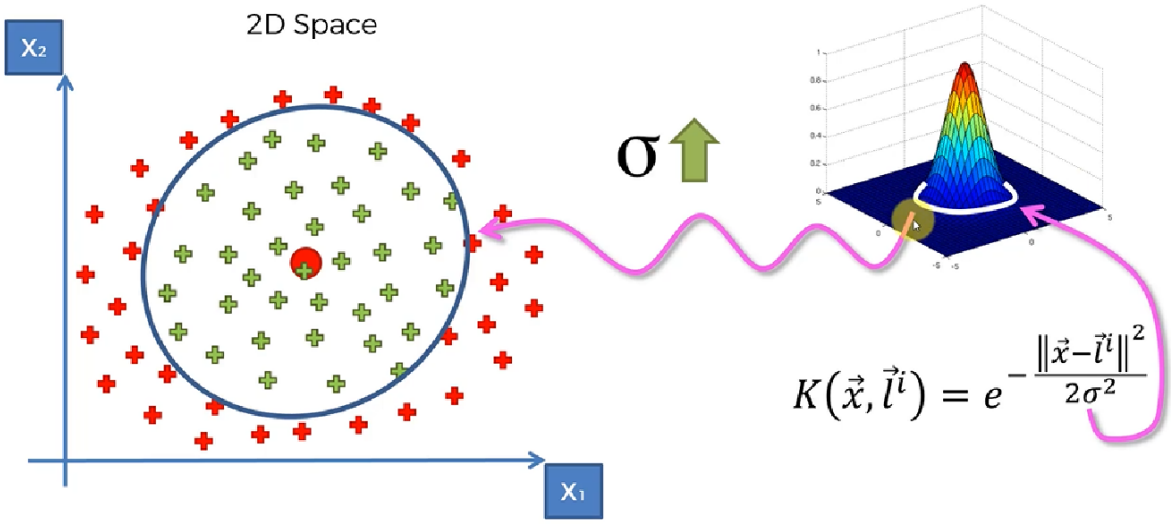
$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$



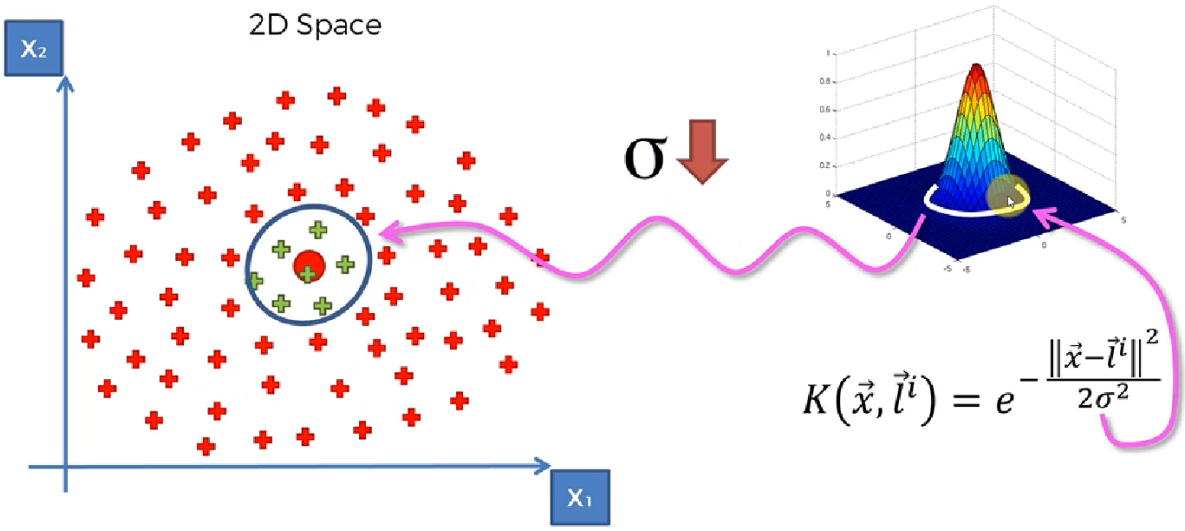


$$K(\vec{x}, \vec{l}^i) = e^{-\frac{\|\vec{x}-\vec{l}^i\|^2}{2\sigma^2}}$$

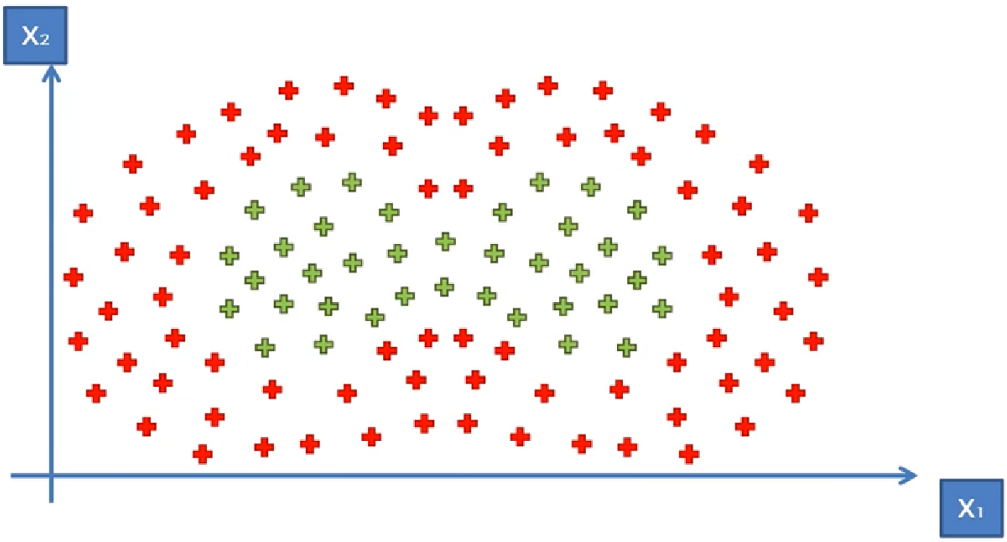
If the value of ‘σ’ is big then the circumference is bigger.

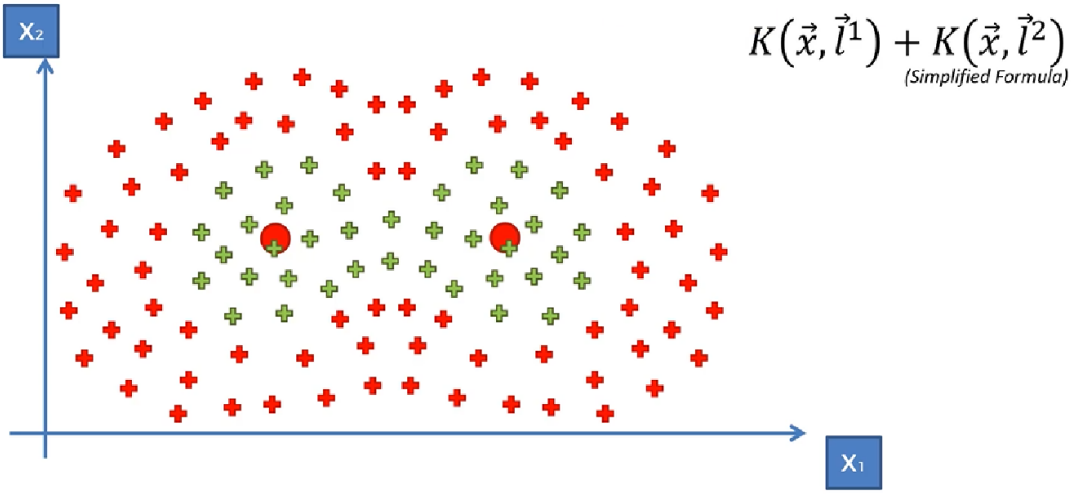


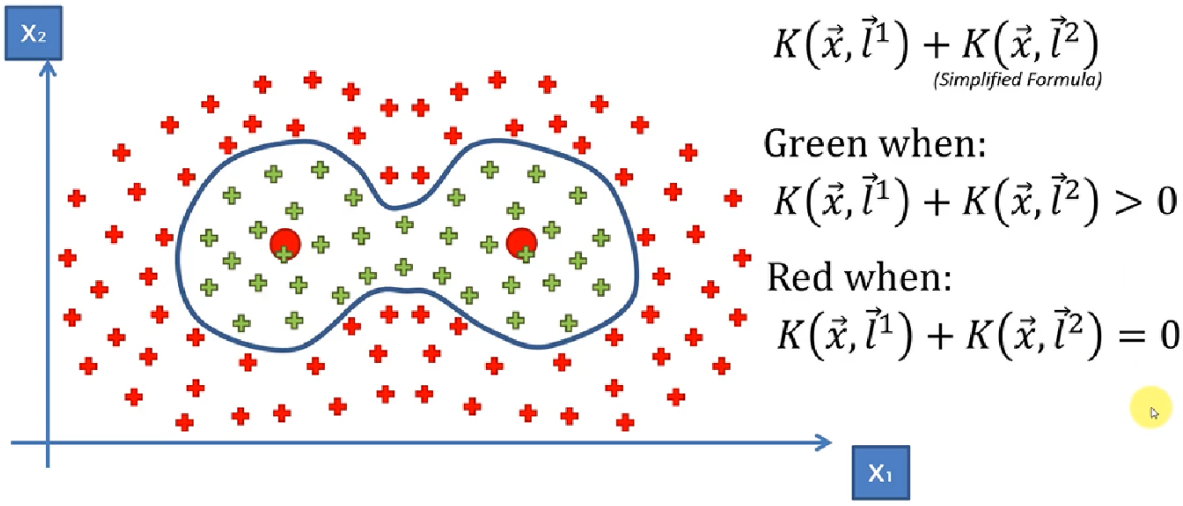
If the value of ‘σ’ is small then the circumference is smaller.



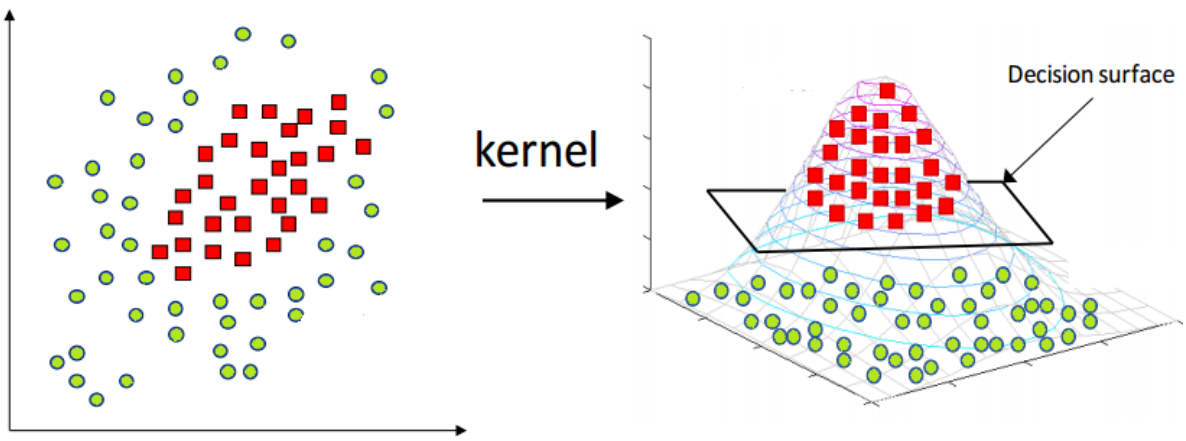
Now we can solve any problem such as,



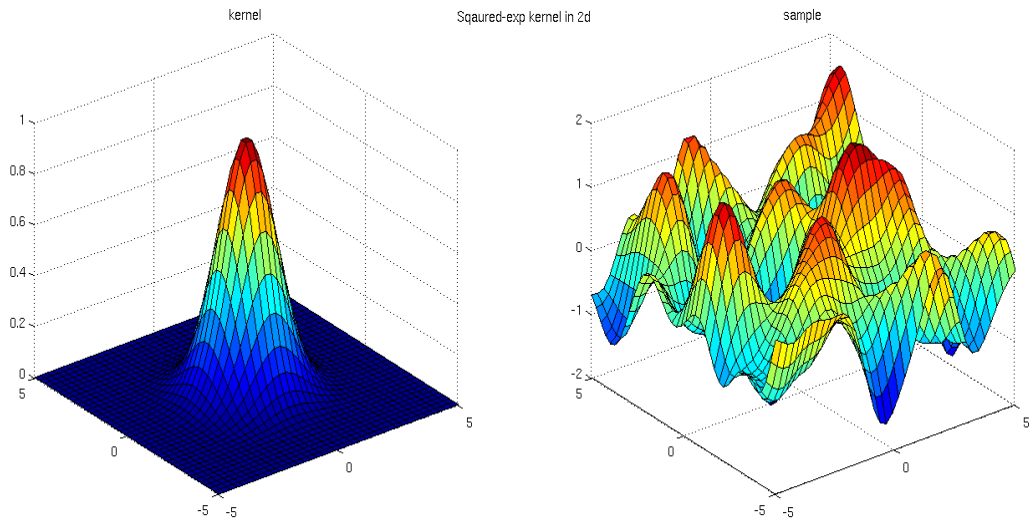




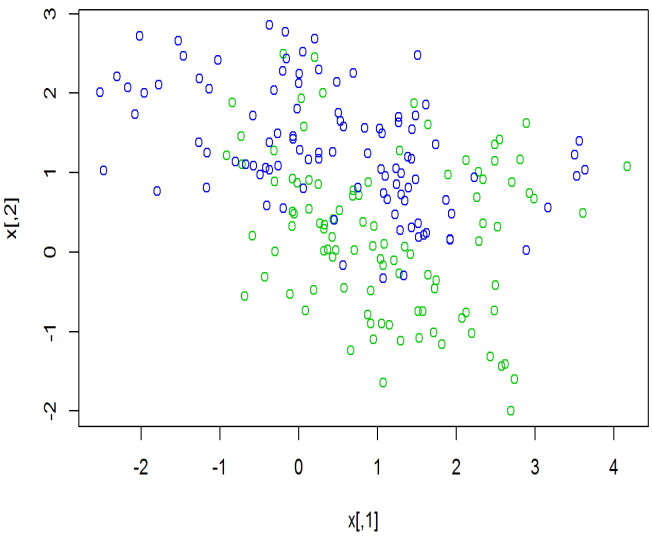
An alternative non-zero decision surface using an RBF kernel

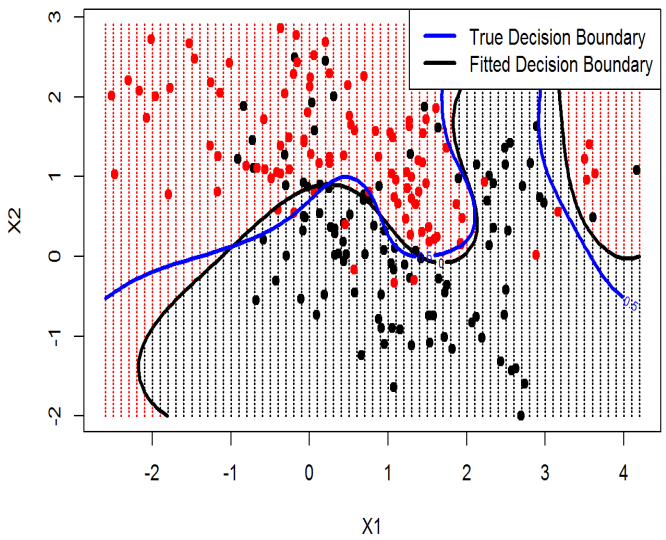


Multiple RBFs



Complex boundaries using RBFs





Fitted decision boundary obtained by RBFs

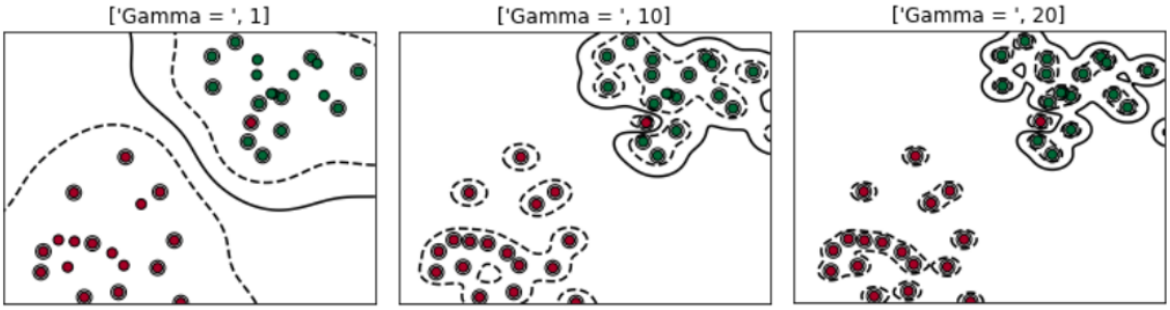
Why is RBF so powerful?

Kernels are used to perform calculations in a higher dimensional space to be able to perform linear separation in the feature space.

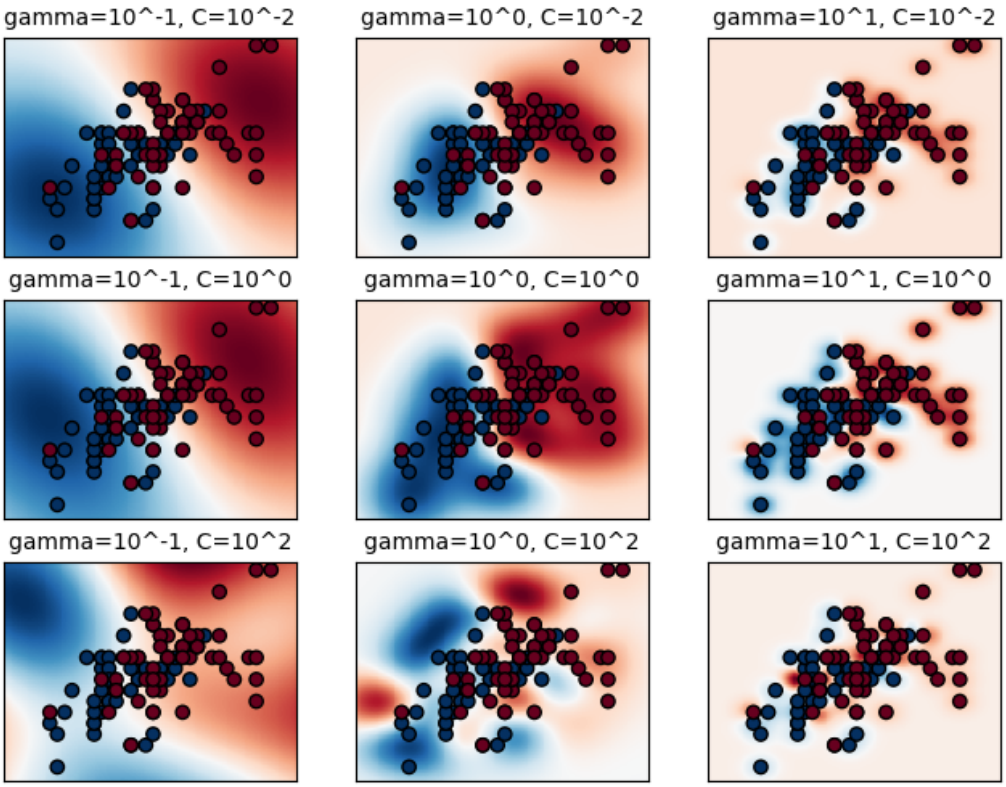
Since RBF uses exponent and as we know the expansion of exp(x) gives a polynomial equation of infinite power, using this kernel gives an *infinite dimensional feature space.*

Regularization

Smaller gamma (larger ℓ), less variance, more bias, more regularization.



The bigger the C (less λ), the more penalty SVM gets when it makes misclassification. More variance, less bias, less regularization.



Kernels och support-vektor maskin

Linjär regression kan formuleras som

$$y = \underbrace{[\theta_1 \ \theta_2 \ \dots \ \theta_n]}_{\theta^T} \underbrace{\begin{bmatrix} \varphi_1(x) \\ \varphi_2(x) \\ \vdots \\ \varphi_n(x) \end{bmatrix}}_{\varphi(x)} + \varepsilon$$

$\varphi(x)$: nonlinear feature transformation

Exempel på $\varphi(x)$

$$\varphi(x) = [1 \ x \ x^2 \ \dots \ x^{n-1}]^T$$

$$\varphi(x) = [1 \ \sin(x) \ \cos(x) \ \sin(2x) \ \cos(2x) \dots]^T$$

$\varphi(x) = [x_1 \ x_2 \ x_1 x_2]^T$ se Ex 8.1 boken

Med regularisering samt

$y = \begin{bmatrix} y_1 \\ \\ y_N \end{bmatrix}$ och $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix}$ ersatt med $\Phi(X) = \underbrace{\begin{bmatrix} \varphi^T(x_1) \\ \vdots \\ \varphi^T(x_N) \end{bmatrix}}_{n}$

fås skattningen

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \underbrace{(\theta^T \varphi(x_i) - y_i)^2}_{\varepsilon_i^2} + \lambda \underbrace{\left(\sum_{i=1}^{n} \theta_i^2\right)}_{\|\theta\|_2^2}$$

normaleq.

$$= (\Phi^T(X)\Phi(X) + N\lambda I_n)^{-1} \Phi^T(X) y$$

$$\hat{y}(x) = \hat{\theta}^T \varphi(x) = y^T \Phi(X)(\Phi^T(X)\Phi(X) + N\lambda I_n)^{-1} \varphi(x)$$

$$= y^T (\underbrace{\Phi(X)\Phi^T(X)}_{K(X,X)} + N\lambda I_N) \underbrace{\Phi(X)\varphi(x)}_{K(X,x)}$$

där

$$K(X,X) = \underbrace{\begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_N) \\ \vdots & & \vdots \\ k(x_N,x_1) & \cdots & k(x_N,x_N) \end{bmatrix}}_{N} \Big\} N$$

$$K(X,x) = \begin{bmatrix} k(x_1,x) \\ \vdots \\ k(x_N,x) \end{bmatrix} \quad \text{och}$$

$$k(x,x') = \varphi^T(x)\,\varphi(x') = \left[\varphi_1(x) \ldots \varphi_n(x)\right] \begin{bmatrix} \varphi_1(x') \\ \vdots \\ \varphi_n(x') \end{bmatrix} = \left(= \sum_{\ell=1}^{n} \varphi_\ell(x)\varphi_\ell(x') \right)$$

benämns kernel där allmänt $k(x,x') = k(x',x)$ är en skalär funktion.

prediktorn $\hat{y}(x)$ kan nu med införandet av den duala skattningen $\hat{\alpha}^T = y^T(K(X,X) + N\lambda I_N)^{-1}$ formuleras på två sätt

$$\hat{y}(x) = \underbrace{\hat{\theta}^T\varphi(x)}_{\text{primär skattning}} = \underbrace{\hat{\alpha}^T K(X,x)}_{\text{dual skattning}} = \hat{\alpha}^T\Phi(X)\varphi(x)$$

$\dim\hat{\theta} = n \qquad \dim\hat{\alpha} = N \qquad \because\ \hat{\theta} = \Phi^T(X)\hat{\alpha}$

Exemper på kernel funktioner

Linjär: $k(x, x') = x^T x'$

Polynomisk: $k(x, x') = (c + x^T x')^{n-1}$

Squared exponential: $k(x, x') = \exp\left(-\frac{\|x - x'\|_2^2}{2\ell^2}\right)$

also called radial basis function (RBF)

exponential quadratic or Gaussian kernel

The hyperparameter ℓ corresponds to the length scale of the kernel, i.e. the distance over which the difference $x - x'$ matters.

ℓ is also known as the bandwidth

Combining kernels

$k(x, x') = k_1(x, x') + k_2(x, x')$

$k(x, x') = k_1(x, x') \cdot k_2(x, x')$

Meaning of a kernel

1) The kernel defines how close/similar any two data points are

$k(x_i, x) > k(x_j, x) \Rightarrow x$ is closer more similar to x_i than x_j.

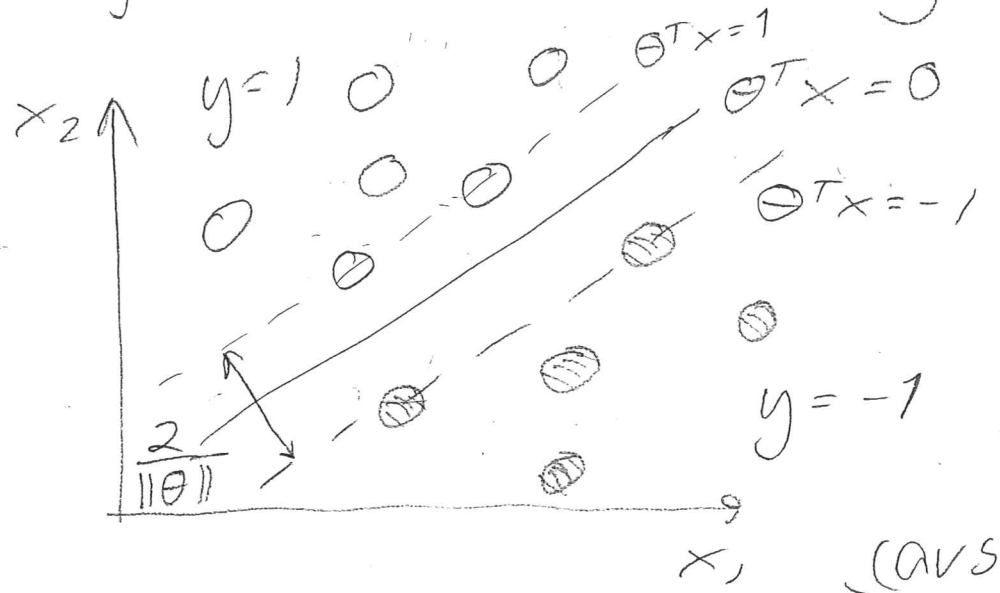
2) $\hat{y}(x)$ is most influenced by the training data points that are closest to x.

3) In the same way as regulization reduces the size of $\hat{\theta}$ due to the cost term $\lambda \|\theta\|_2^2$ and therefore the fluctuation of $\hat{y}$ (less variance but some bias), the choice of kernel also influences the shape of $\hat{y}(x)$. For instance, the RBF kernel implies a preference for smooth functions.

Support vector klassificering

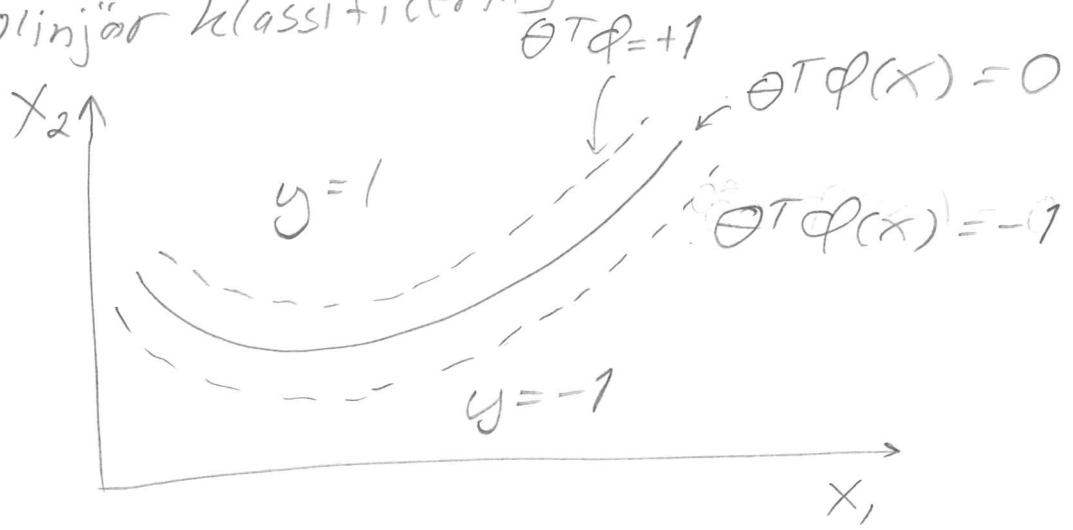
Binär klassificering $\Rightarrow y \in \{-1, 1\}$

Linjär klassificering



Maximera marginalen (avståndet) mellan de två klasserna $\Rightarrow \max \frac{1}{\|\theta\|}$ och $\min \|\theta\|$

Olinjär klassificering



Antag att data går att separera i två områden, ett för $y=1$ o ett för $y=-1$. Punkterna x_i som ligger närmast klassificeringsgränsen $\theta^T\varphi(x)=0$ antas ligga på den övre gränsen $\theta^T\varphi(x_i)=1$ då $y_i=1$ och den undre gränsen $\theta^T\varphi(x_i)=-1$ då $y_i=-1$.

Därmed uppfyller alla datapunkter

$\theta^T\varphi(x_i) \geq 1$ för $y_i=1$

$\theta^T\varphi(x_i) \leq -1$ för $y_i=-1$

vilket kan sammanfattas i villkoret

$\boxed{y_i\,\theta^T\varphi(x_i) \geq 1}$ för alla (x_i, y_i)

samtidigt kan man visa att minimering av $\|\theta\|$ ger maximal marginal mellan undre och övre streckade linjerna ovan.

1) Hård marginal

$$\min_{\theta} \|\theta\| \quad \text{med bivillkoret}$$

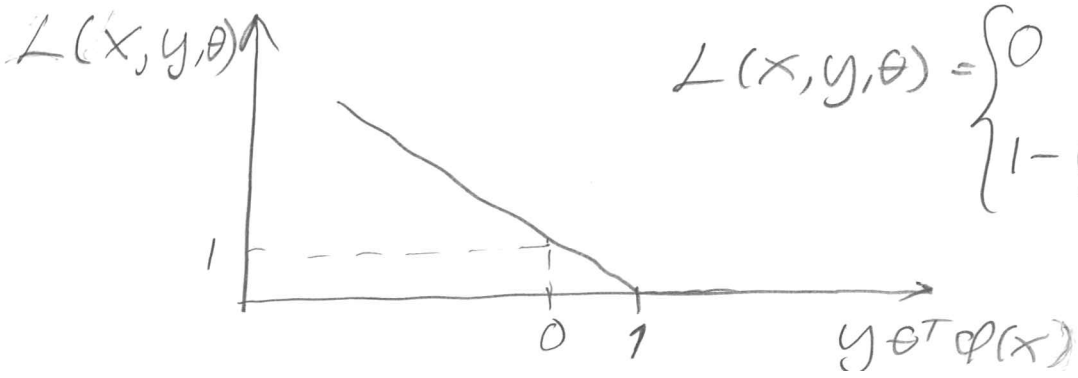
$$y_i \theta^T \varphi(x_i) \geq 1$$

$$i = 1, \dots, N$$

Klassificerare
$\hat{y}(x_*) = \text{sign}(\theta^T \varphi(x^*))$

2) Mjuk (soft) marginal, vilket innebär att data innanför streckade området accepteras t.o.m. på fel sida om klassificeringsgränsen

Inför Hinge förlustfunktion



$$L(x,y,\theta) = \begin{cases} 0 & y\theta^T x \geq 1 \\ 1 - y\theta^T x & \text{övrigt} \end{cases}$$

$$L(x, y, \theta) = \max\{0, 1 - y\theta^T \varphi(x)\}$$

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \max\{0, 1 - y_i \theta^T \varphi(x_i)\} + \lambda \|\theta\|_2^2$$

Lagrange formulering och införande av kernels ger

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{2} \alpha^T K(X, X) \alpha - \alpha^T y$$

med birillkoren $|\alpha_i| \leq \frac{1}{2N\lambda}$

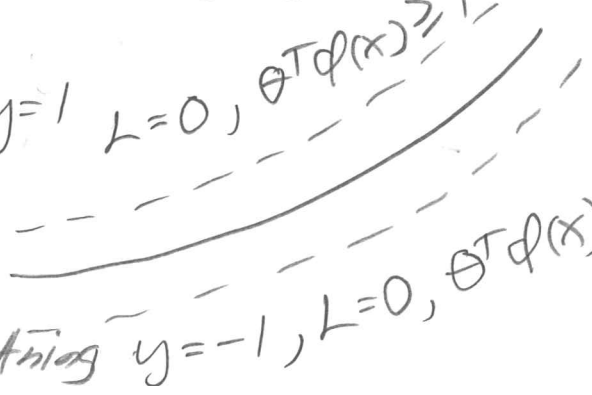
$$\alpha_i y_i \geq 0$$

Klassificerare

$$\hat{y}(x_*) = \text{sign}(\underbrace{\hat{\alpha}^T K(X, x_*)}_{= \hat{y} \text{ för svm regression}})$$

$\alpha_i = 0$ då $L = 0$ dvs då

$y\theta^T \varphi(x) \geq 1$



$y\theta^T \varphi(x) < 1$ påverkar optimeringen med linjärt ökande omfattning

Summary Support Vector Machines

Advantages:

- based on nice theory,
- excellent generalization properties,
- objective function has no local minima,
- can be used to find non-linear discriminant functions,
- Complexity of the classifier is characterized by the number of support vectors rather than the dimensionality of the transformed space.

Disadvantages:

- tends to be slower than other methods,
- quadratic programming is computationally expensive.