Optimization and Data Analytics, Course Notes

Alex Justesen Karlsen, 201404623

January 9, 2019

Contents

Optimization

Optimization is an important tool in making decisions and in analyzing physical systems. In mathematical terms, an optimization problem is the problem of finding the best solution from among the set of all feasible solutions. [1] Problems can be solved analytically (symbolic), when solutions do exist. Numerical methods can solve (some) problems, that can not be solved analytically by approximation. Optimization can be classified in to categories: Constrained or Unconstrained, Continuous or Discrete and Deterministic or Stochastic Optimization.

Most algorithm do not guarantee to find global optimum, there exist method that does, however these are computationally heavy and time exhaustive. Local search are often prefered, as the local solution are *“good enough”* and are not as time consuming.

Constrained Optimization vs. Unconstrained Optimization

Unconstrained optimization problems arise directly in many practical applications; they also arise in the reformulation of constrained optimization problems in which the constraints are replaced by a penalty term in the objective function. Constrained optimization problems arise from applications in which there are explicit constraints on the variables. The constraints on the variables can vary widely from simple bounds to systems of equalities and inequalities that model complex relationships among the variables. Constrained optimization problems can be furthered classified according to the nature of the constraints (e.g., linear, nonlinear, convex) and the smoothness of the functions (e.g., differentiable or non-differentiable) [2].

Continuous Optimization vs. Discrete Optimization

Models with discrete variables are discrete optimization problems; models with continuous variables are continuous optimization problems. Continuous optimization problems tend to be easier to solve than discrete optimization problems; the smoothness of the functions means that the objective function and constraint function values at a point x can be used to deduce information about points in a neighborhood of x [3].

The travelling salesman problem is a discrete optimization problem. The travelling salesman and simulated annealing are likely to come in the exam.

[1] NEOS. Accessed December 20, 2018. https://neos-guide.org/optimization-tree.

[2] NEOS. Accessed December 20, 2018. https://neos-guide.org/optimization-tree.

[3] NEOS. Accessed December 20, 2018. https://neos-guide.org/optimization-tree.

Deterministic Optimization vs. Stochastic Optimization

Deterministic Optimization *Always produces the same result from initial input (e.g. for same candidate solution x_0)* – Carl

In deterministic optimization, it is assumed that the data for the given problem are known accurately. However, for many actual problems, the data cannot be known accurately for a variety of reasons. The first reason is due to simple measurement error. The second and more fundamental reason is that some data represent information about the future (e. g., product demand or price for a future time period) and simply cannot be known with certainty.[4]

Stochastic Optimization Does not always produce the same result. Randomness is added to the algorithm e.g. random selection of candidate solutions or neighborhoods. Stochastic methods are presumably faster and more robust, as they have mechanism to avoid getting stuck in local minimums.

In optimization under uncertainty, or stochastic optimization, the uncertainty is incorporated into the model. Robust optimization techniques can be used when the parameters are known only within certain bounds; the goal is to find a solution that is feasible for all data and optimal in some sense. Stochastic programming models take advantage of the fact that probability distributions governing the data are known or can be estimated; the goal is to find some policy that is feasible for all (or almost all) the possible data instances and optimizes the expected performance of the model. [5]

Words to look up!: The language: Heuristic, Converge,

[4] NEOS. Accessed December 20, 2018. https://neos-guide.org/optimization-tree.

[5] NEOS. Accessed December 20, 2018. https://neos-guide.org/optimization-tree.

Linear Programming

Matrix Games

Geometric Method

Simplex

Constrained Optimization

Solving Linear Equations

Non-linear Constrained Optimization

Unconstrained Optimization

Mathematical Preliminaries

One-dimension

$$f : R^1 \rightarrow R^1$$

$FONC$:

$$f'(x) = 0$$

$FOSC$:

For a maximum (not entirely sure)

$$f''(x) > 0$$

For a minimum (not entirely sure)

$$f''(x) < 0$$

N-dimensions

$$f : R^n \to R^1$$

$SONC$:

$$\nabla f(x_1, x_2) = 0$$

∇f is the gradient of f, which is a vector of the partial derivatives of f.

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

$SOSC$:

Considering the definiteness of the Hessian matrix reveals optimum as max, min or saddle point, using an eigen value analysis.

The hessian matrix is a matrix of the second order derivatives of f.

$$D^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 x_2} \\ \frac{\partial^2 f}{\partial x_2 x_1} \end{bmatrix}$$

1D Line Search Methods

Golden Section Search

Fibonacci Search

Newtons Method

Gradient Methods

Steepest Descent

Conjugate Descent

Newton's Method Variants

N-Dimensional Newton

Levenberg-Macquard's modification

Newton's for non-linear least-square

Quasi-Newton

Global Search Methods

Simulated Annealing

Particle Swarm Optimization

Genetic Algorithms

Data Analytics (Machine Learning)

Machine Learning (ML) is a branch within Data Science to construct algorithms, that are self-improving based on statistical models. ML approaches can be classified into 3 main categories;

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Supervised Learning

Is the learning task of mapping an input vector to an output response. Supervised learning uses data labelled by human experts. The learning phase tries to optimize the model to correctly map the input to the labeled output. The model are then used to predict the response of unlabeled data. Model are either used for regression or classification, depending on wether the data is continuous or discrete.

Unsupervised Learning

Is the learning task of finding commonalities in forms of structure and patterns in unlabeled raw test data. The algorithm are unsupervised i.e. without human help.

Reinforcement Learning

The training process is iteratively the model is presented an input and guesses an output. An expert corrects the guess only by binary feedback. Depending on the feedback the training either updates the model or continue the process.

Data Analytics Machine Learning is also known as Data analytics. Data Analytics is the discovery, interpretation and communication of meaningful patterns in data. Typically the analysis is four-fold; data preprocessing, data representation, representation preprocessing and model selection/training.

An example using image data: 1. Data preprocessing → image segmentation 2. Data representation → vectorizing 3. Representation preprocessing → centering 4. Model selection → classification

Regression

Classification

Multi-class Classification

Unsupervised Learning Techniques

Dimensional Reduction Techniques

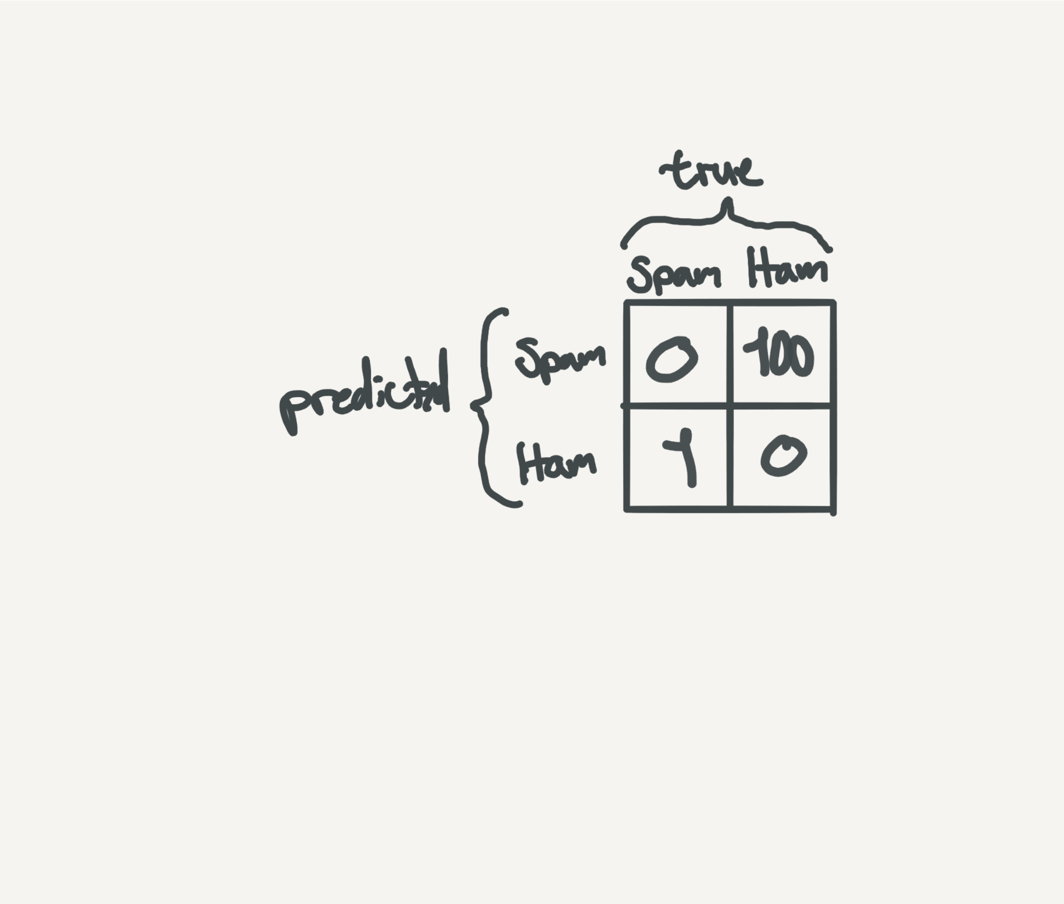
Principal Component Analysis

Decision Theory

Decision Theory is about *Minimize expected loss* of a decision e.g. classification or regression.

An Email Spam filter is a good example. We can classify mails as ham or spam. The loss of getting a mail, that was actually spam is not very costlt, however filtering an important mail can be potentially harmful.

A loss matrix could look like:



		true	
		Spam	Ham
predicted	Spam	0	100
	Ham	1	0

The are different kind of loss functions e.g.

- “1-0” loss function, that either classifies correctly (l=0) or incorrectly (l=1).
- Squared loss function, that squared the loss, typically used in regressions.

Statistically we wish the smallest loss on average. Using “1-0” loss function we classify based on the conditional probability $p(y|x)$. The optimization problem can be described as;

$$\hat{y} = \arg\min_{y} p(y|x)$$

Is also called maximum likelihood classification.

State of nature

Probability-Based Learning

Assuming a classification problem with K possible outcomes;

$$C = \{c_1...c_k\}$$

C is the set of classes.

$P(c_k)$ denotes the prior probability of the outcomes.

The probability of all comes are denoted as;

$$\sum_{k=1}^{K} P(c_k) = 1$$

We do NOT classify based on the priori, if $P(c_1) > P(c_2)$, then all would be classified as the c_1.

$P(c_k|x)$ denotes the conditional probability, that states the probability of class c_k given the observation x.

The joint probability of $c_k 1$ and x is

$$P(c_k, x) = P(c_k|x)P(x) = P(x|c_k)P(c_k)$$

From this we can obtain Bayes' formula

$$P(c_k|x) = \frac{p(x|c_k)P(c_k)}{p(x)}$$

The error can then be defined as

$$P(error|x) = \begin{cases} P(c_1|x), if\ x\ is\ misclassified\ to\ c_2 \\ P(c_2|x), if\ x\ is\ misclassified\ to\ c_1 \end{cases}$$

Which is given by;

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

Thus, we can define Bayes' rule as decision rule:

Decide c_1 if $P(c_1|x) > P(c_2|x)$, else decide c_2.

The decision function is obtained by finding x for $P(c_1|x) = P(c_2|x)$.

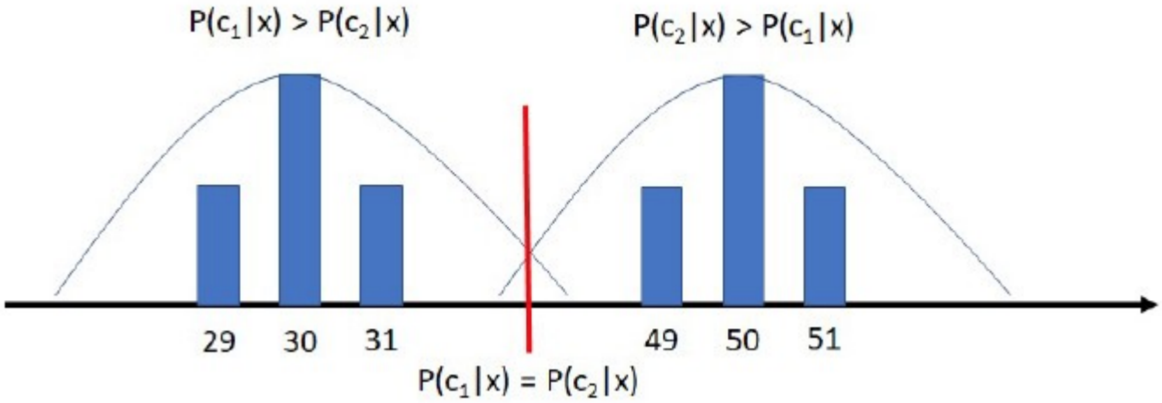


Figure 1: Bayes Decision Function

More than two classes??

Risk-Based Decision Functions

Suppose given observation x , we take the action α_i classifying the sample to class i.

We define the loss function $\lambda(\alpha_i|c_k)$, which expresses the loss incurred taking action α_i, given the correct class is c_k.

The risk of α_i for observation x can be defined as:

$$R(\alpha|x) = \sum_{k=1}^{K} \lambda(\alpha_i|c_k)P(c_k|x)$$

Gaussian Decision Functions

We make assumptions about the distribution of samples data is Gaussian. In reality it can take any distribution, however many has proven to be gaussian. The central limit theorem tells, that aggregating a large number from a lot of small independent disturbance will turn out to be gaussian.

When sampling a variable a lot of disturbance is collected as well. The expected value of a continuous variable can be determined as;

$$\varepsilon[f(x)] = \int_{-\infty}^{\infty} f(x)p(x)dx$$

The normal distribution $p(x)$ is given by:

$$p(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

For discrete variables of set D, it can be expressed as;

$$\varepsilon[f(x)] = \sum_{x \in D} f(x) P(x) dx$$

where $P(x)$ is the Probability Mass Function (PMF) of x.

$$p(x) = \frac{1}{\sqrt{2\pi} \cdot \sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

μ is the mean and σ is the standard deviation.

$$\mu = \varepsilon[x] = \int_{-\infty}^{\infty} x \cdot p(x) dx$$

$$\sigma^2 = \varepsilon[(x-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 \cdot p(x) dx$$

The following model visualizes the normal distribution. 95% are within 95% of the interval of $2\mu + \sigma$ (i.e. confidence interval).

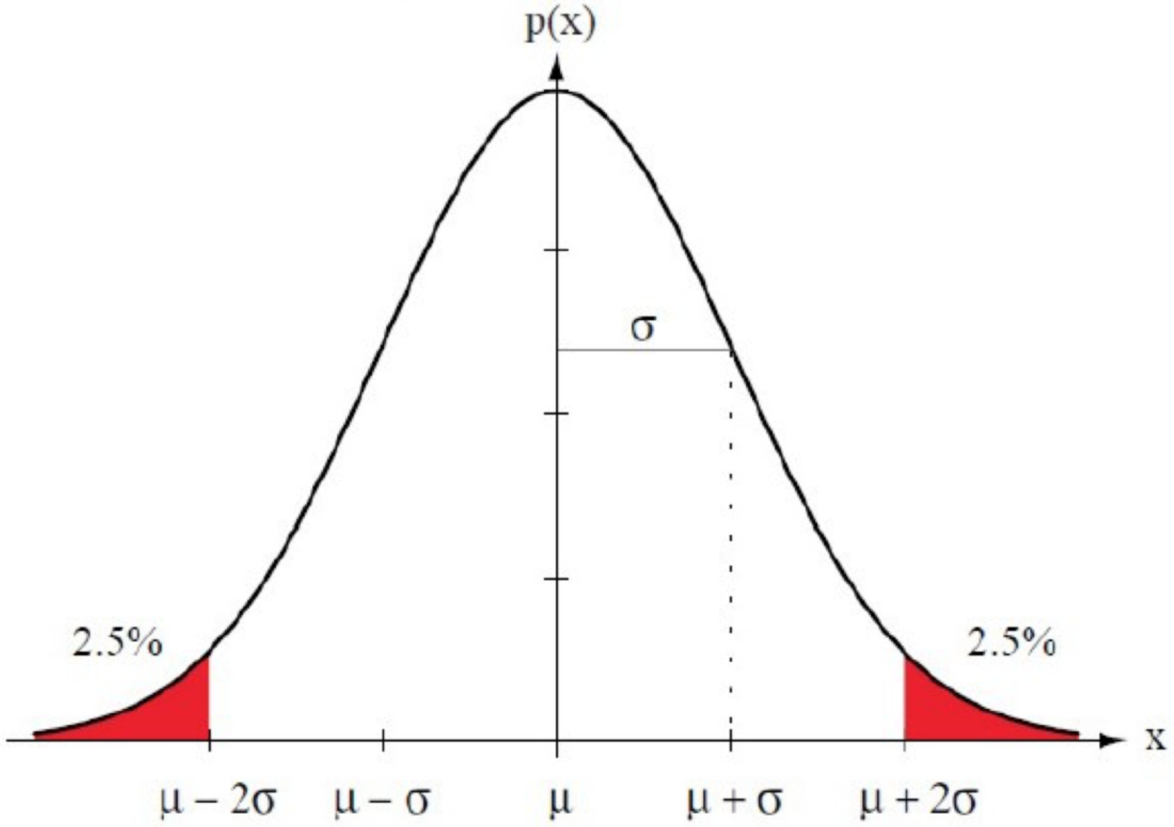


Figure 2: Normal Distribution

Normal distribution are also denoted $\mathcal{N}(\mu, \sigma^2)$.

Multi-variate normal distribution

In higher dimensions, we have a D-dimensional vector x;

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix}$$

The normal distribution is denoted $\mathcal{N}_k(\mu, \sigma^2)$, where $p(x)$ can be expressed as;

$$p(x) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$$

where Σ is the covariance matrix, that is defined as;

$$\Sigma = \varepsilon[(x-\mu)(x-\mu)^T] = \int (x-\mu)(x-\mu)^T p(x)dx$$

Σ can be calculated using 1D operations.

$$\Sigma_{ij} = \varepsilon[(x_i-\mu_i)(x_j-\mu_j)]$$

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \dots & \Sigma_{1D} \\ \vdots & \ddots & \vdots \\ \Sigma_{D1} & \dots & \Sigma_{DD} \end{bmatrix}$$

The covariance matrix have some important properties;

- Σ_{ii} is the variance of dimension i.
- Σ_{ij} is the covariance of i and j.

1. If $\Sigma_{ij} = 0$ for $i \neq j$, then i and j are statistically independent.
2. If $\Sigma_{ij} = 0$ for $i \neq j$, then the multi-variate normal distribution degenerates to the product of k normal distributions.

It can often be used to define a decision function taking into account the different scaling of the various dimensions and theri co-variance.

$$d_m(x-\mu) = (x-\mu)^T\Sigma^{-1}(x-\mu)$$

Data Whitening Is a linear transformation with a whitetning matrix W, that transforms a vector of random variables X with a known covariance matrix into a nex vector Y whose covariance matrix is the identity matrix, meaning the data are uncorrelated and each have a variance of 1.

Eigenvalues decomposition:

$$\Sigma = U\Lambda U^T$$

Where U is a matrix whose column are formed by the eigenvectors and Λ is a diagonal matrix with the corresponding eigenvalues.

There are infinitely many option of W. Commonly used are ZCA i.e. Mahalanobis whitening;

$$W = U\Lambda^{-\frac{1}{2}}$$

The transformation can be denoted; Wikipedia:

$$Y = WX$$

Alexandros:

$$Y = W^T X$$

The transformations has much resemblance with PCA. It is an linear transformation using eigen vectors. PCA is an orthogonal transformation, thus rotating the data. PCA uses $W = U\Lambda$.

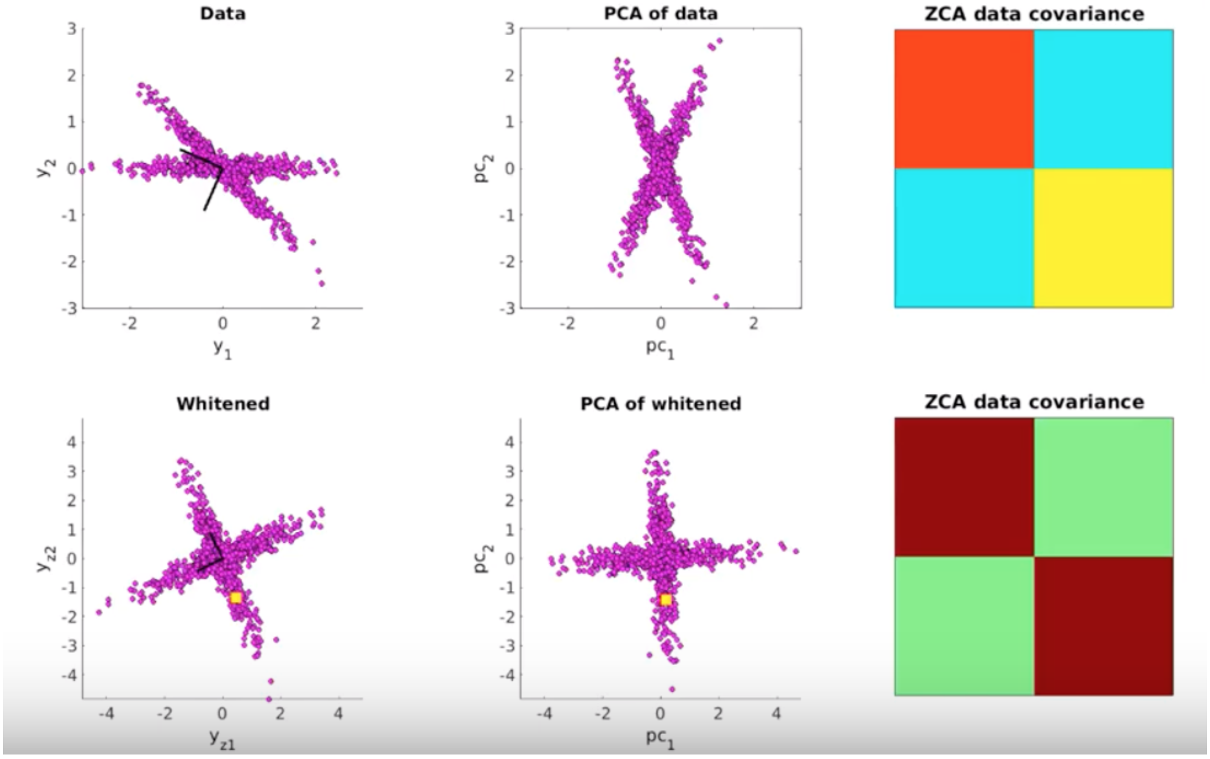


Figure 3: Whitetning

PCA can be used to reduce the dimensionality. Whitening is used to remove correlation in data by shrinking large data direction and expanding small directions. Large data directions tend to reflect low spatial frequencies, thus whitening can increase spatial precision. Whitening separates data by expanding small dimension assuming all the dimensions are of equal importance.

Decision Functions

Consider a two-class classification problem:

$$p(x|c_1) \sim \mathcal{N}(\mu_1, \Sigma_1)$$

$$p(x|c_2) \sim \mathcal{N}(\mu_2, \Sigma_2)$$

Using Bayes' rule, we will;

Decide c_1 if $P(c_1|x) > P(c_2|x)$, else decide c_2.

If $g(\cdot)$ is a monotonic function, then

$$P(c_1|x) > P(c_2|x) \rightarrow g(P(c_1|x)) > g(P(c_2|x))$$

We do this to ease our calculations. We replace using Bayes formula;

Decide c_1 if $P(c_1|x)P(c_1) > P(c_2|x)P(c_2)$, else decide c_2. (note: divide by p(x) dissappears on both sides.) or Decide c_1 if $f(x|c_1) > f(x|c_2)$, else decide c_2.

We simplify the computation of the exponential function by $g(x) = ln(x)$, then we have;

$$f(x|c_k) = -\frac{1}{2}(x-\mu)^T \Sigma_k^{-1}(x-\mu) - \frac{D}{2}ln(2\pi) - \frac{1}{2}ln(|\Sigma_k|) + ln(P(c_k))$$

Special Case In case $\Sigma_k = \sigma^2 I$, the deteminant $|\Sigma_k| = \sigma^{2D}$ and the inverse $\Sigma_k^{-1} = \frac{1}{\sigma^2}I$. The decision rule then becomes;

$$-\frac{1}{2\sigma^2}(x-\mu_1)^T(x-\mu_1) + ln(P(c_1)) > -\frac{1}{2\sigma^2}(x-\mu_2)^T(x-\mu_2) + ln(P(c_2))$$

Discrete Values Integrals become summation;

$$\int p(x|c_k) \rightarrow \sum_x P(x|c_k)$$

Bayes' formula involves propalities instead of propability densities;

$$P(c_k|x) = \frac{P(x|c_k)P(c_k)}{P(x)}$$

Where

$$P(x) = \sum_{k=1}^{K} P(x|c_k)P(c_k)$$

Maximum Likelihood Estimation

We assume our collection of samples to be of a gaussian distribution and that each training sample drawn are idenpendent identical distributed (i.i.d.) random variables.
Suppose the training set D contains n samples, and θ is the sample we wish to estimate. The likelihood is defined as;

$$p(D|\theta) = \prod_{k=1}^{n} p(x_k|\theta)$$

The estimate $\hat{\theta}$ is by definition the value, that maximizes $p(D|\theta)$.

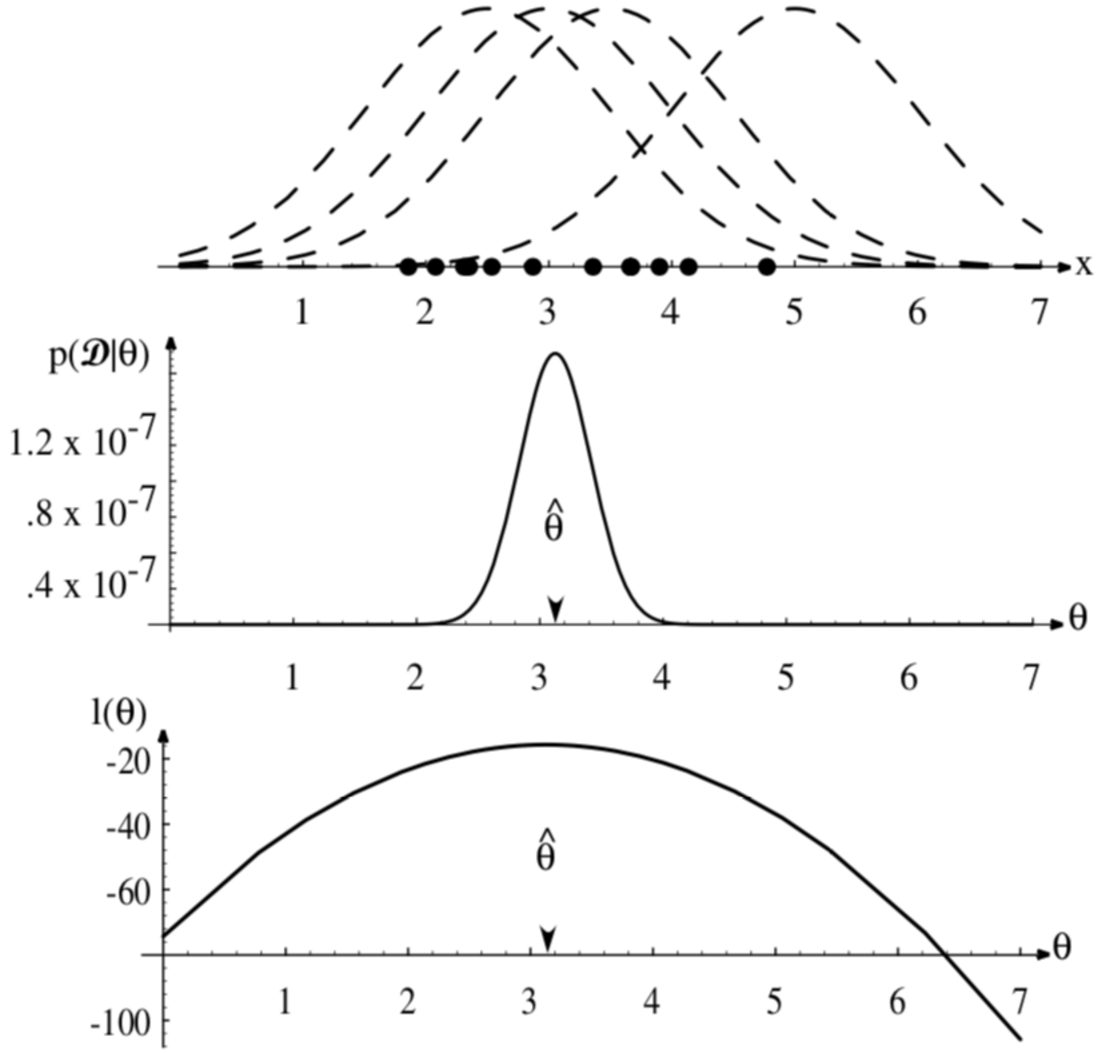


Figure 4: likelihood

Figure shows all the possible distributions of the sampled data, an optimal solution and the same optimal solution option by the log-likelihood.

Taking the natural logarithm generate a monotonic function, that simplifies computations. The operation can be written as;

$$\hat{\theta} = \arg\max_{\theta} ln(p(D|\theta))$$

It can be derived, that the mean μ and standard deviation σ can be estimated from the collection by the general formulas.

Univariate

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \mu)^2$$

Multivariate

$$\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(x_k - \mu)^2$$

Typically we do not know these variables and must estimate them.

Baysian Estimation

We do not seek a true value for θ, but a random variable instead. Training allow us to convert a distribution into a posterior probability density.

Linear Methods

Fischer Discriminant Analysis

Linear Discriminant Analysis

Linear Discriminant Functions

Generalized Discriminant Function

Kernel-Based Learning

Support Vector Machines

Least-Mean-Square Regression

Kernel Discriminant Analysis

Neural Networks