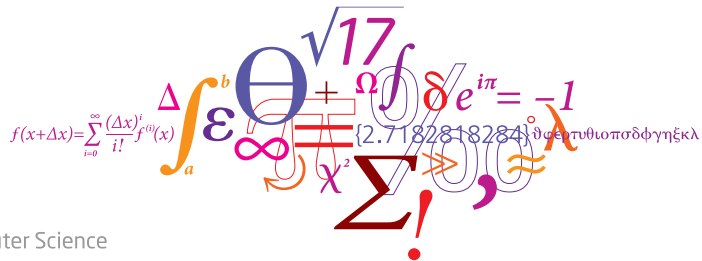
02471 Machine Learning for Signal Processing

Parameter Estimation

Tommy Sonne Alstrøm

Cognitive Systems section



DTU Compute
Department of Applied Mathematics and Computer Science

Outline

- Administrative
- Last Week
- Tools needed for parameter estimation
  - Cross validation
  - Matrix derivatives
- Parameter Estimation
- Biased and unbiased estimation
- Maximum likelihood and Bayesian inference
- Next week

Material: ML 3.1–3.3, 3.5, 3.8–3.11

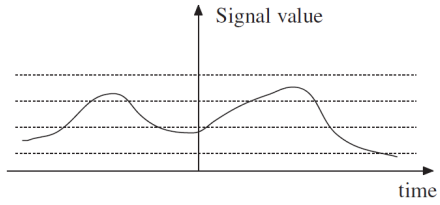
Feedback

- Feedback from you is a critical component for improving both the course and my teaching.
- Most significant feedback
  - Preliminary videos worked great
- Type of feedback
  - Mention one thing that worked?
  - Mention one thing should be improved (both in current lecture and last weeks exercise)?
  - Mention one thing you would change if you gave the lecture.

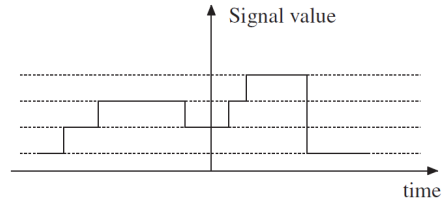
Last Week

From analog to digital signals

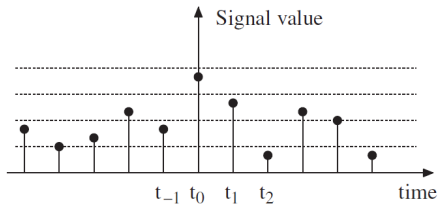
Continuous-time signal
Continuous time
Continuous signal values



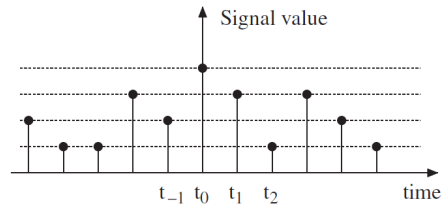
Continuous-time signal
Continuous time
Discrete signal values



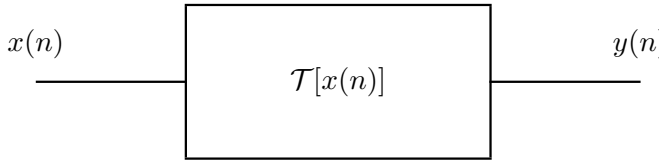
Discrete-time signal
Discrete time
Continuous signal values



Discrete-time signal
Discrete time
Discrete signal values



Traditional DSP system



This system is restricted to be a linear time-invariant (LTI) system.

General linear system

$$y(n) = \underbrace{-\sum_{k=1}^{N} a_k y(n-k)}_{\text{Recursive (AR)}} + \underbrace{\sum_{k=0}^{M} b_k x(n-k)}_{\text{Non-recursive (MA)}}$$

Impulse response and convolution sum of time-invariant linear system (LTI)

If

$$h(n) = \mathcal{T}[\delta(n)]$$

where

$$\delta(n) = 1 \quad \text{for } n = 0 \tag{1}$$
$$\delta(n) = 0 \quad \text{otherwise} \tag{2}$$

then the output of the system is a convolution sum.

Response of a MA-LTI system

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) = h(n) * x(n)$$

The System function

The time–domain expression

$$y(n) = h(n) * x(n)$$

Can be proven to be, in the z–domain

$$Y(Z) = H(Z)X(Z)$$

The z-transform and Fourier transforms are equal when

$$X(f) = X(\mathsf{z})_{\mathsf{z}=e^{j2\pi f}}$$
$$Y(f) = H(f)X(f)$$

Common operations from probability

Expectation (mean)

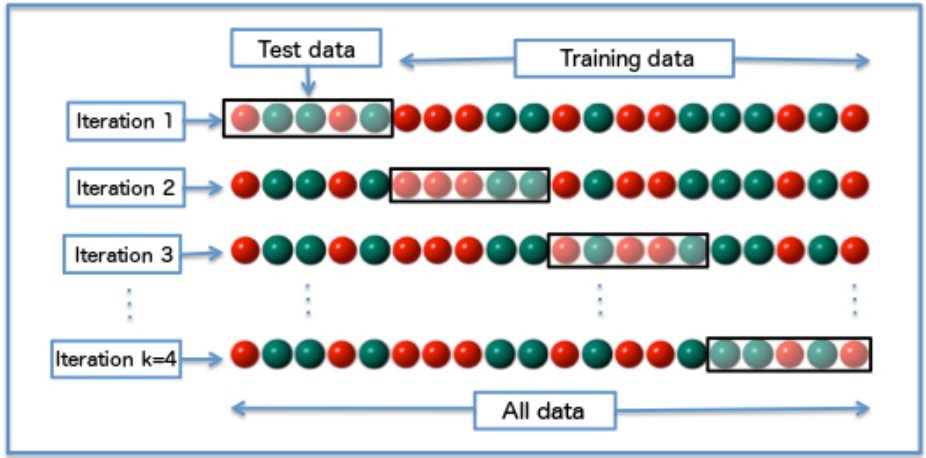
$$\mathbb{E}[x] := \int_{-\infty}^{+\infty} x p_x(x) dx$$

Last weeks summary

- Digital signal processing is processing of quantized discrete-time signals, the two primary tools being:
  - Filtering of signals (changing the content of the signals) using linear time-invariant (LTI) systems.
- The impulse response of a LTI filter fully describes the filter functionality, and the filter is applied by convolution of the input signal with the impulse response.
- We need tools from probability, especially expectations and probability density functions.
- Machine learning is often characterized by data-driven approaches of learning models, as opposed to filter design in DSP.
- We will largely assume you know the basics, such as cross validation, classification (k-nearest neighbor) and k-means.

Tools needed for parameter estimation

Cross validation



Do you see any problems here if the data is a time-series?

Cross validation for time-series

74 *C. Bergmeir et al. / Computational Statistics and Data Analysis 120 (2018) 70–83*

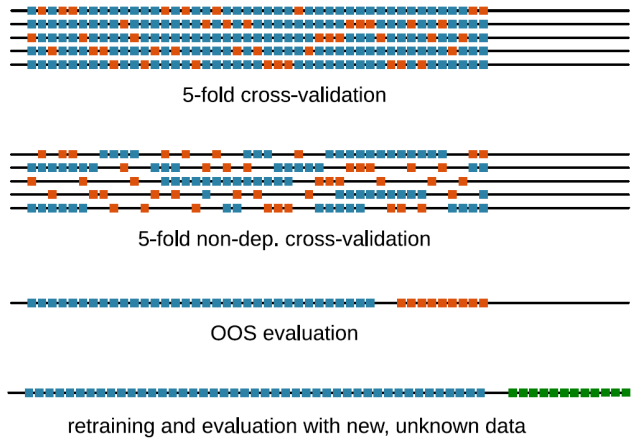


Fig. 2. Training and test sets used for the experiments. The blue and orange dots represent values in the training and test set, respectively. The green dots represent future data not available at the time of model building. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Vector layout I

The book assumes vectors are column vectors, this is contrary to e.g. DTU Mathematics 1 (01005) where vectors are usually row vectors

$$\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \dots & x_l \end{bmatrix}^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \end{bmatrix}$$

In this layout, the gradient of a multivariate function is also a column vector

$$\nabla f(\boldsymbol{x}) := \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} := \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \dots & \frac{\partial f(\boldsymbol{x})}{\partial x_l} \end{bmatrix}^T$$

In other literature, often the gradient is defined as a row vector:

$$\nabla f(\boldsymbol{x}) := \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} := \begin{pmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \dots & \frac{\partial f(\boldsymbol{x})}{\partial x_l} \end{pmatrix} = \begin{bmatrix} \frac{\partial f(\boldsymbol{x})}{\partial x_1} & \frac{\partial f(\boldsymbol{x})}{\partial x_2} & \dots & \frac{\partial f(\boldsymbol{x})}{\partial x_l} \end{bmatrix}$$

Vector layout II

This layout is called denominator layout (also called "Hessian formulation").

For vector derivatives this layout is important, e.g for denominator layout:

$$\frac{\partial \mathbf{b}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{b}^\top \mathbf{A}$$

But for numerator layout (also called "Jacobian formulation") we get

$$\frac{\partial \mathbf{b}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{A}^\top \mathbf{b}$$

Read more about layouts at

`https://en.wikipedia.org/wiki/Matrix_calculus`

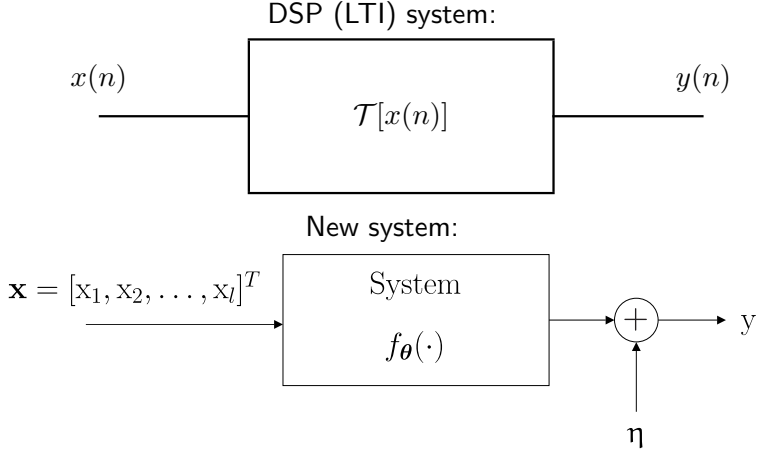
If you are uncomfortable with matrix derivatives, check the video on the course website.

Summary

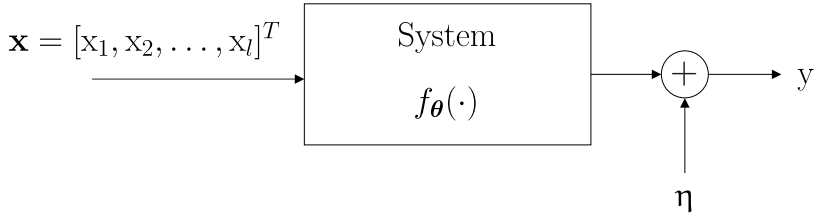
- Cross-validation for time-series requires extra attention to avoid using information from test in train.
- Matrix calculus is equivalent to ordinary calculus and actually makes your life much easier (once learned).
- If you look-up rules outside the book, it is important to be aware of the vector layout convention.
- We will use matrix calculus regularly in the course to derive update rules.
- Check Appendix A for useful rules and identities when doing exercises.

Parameter Estimation

Our new system



What is parameter estimation



This system can be written formally as

$$\mathrm{y} = f_{\boldsymbol{\theta}}(\mathbf{x}) + \eta$$

Two tasks must be carried out to use this system:

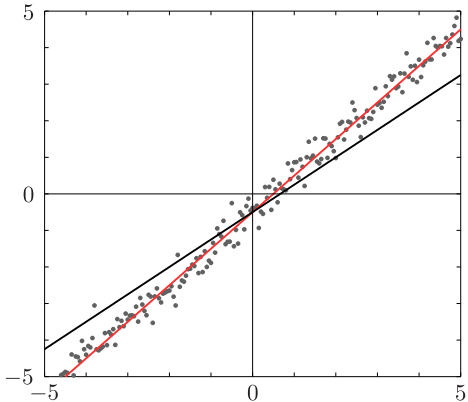
- Choose the function $f_{\boldsymbol{\theta}}(\cdot) := f(\cdot, \boldsymbol{\theta})$.
- Once chosen, estimate the parameters $\boldsymbol{\theta}$. This is the topic for today!!.

Remarks on notation:

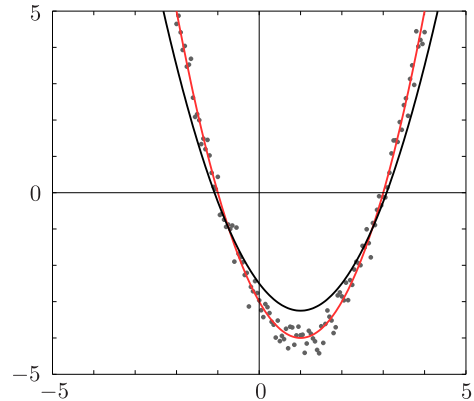
- y is a random variable, y is a scalar.
- $\mathbf{x}$ is a random vector, $\boldsymbol{x}$ is a vector
- $\boldsymbol{\theta}$ is a parameter vector.
- η is a random variable (noise).

Two examples

$$y = \theta_0 + \theta_1 x$$



$$y = \theta_0 + \theta_1 x + \theta_2 x^2$$



Warm-up: What is the value of $\boldsymbol{\theta}$ for the two cases (red lines)?

Relation to existing material

Let us rewrite the second order polynomial, $y(x) = \theta_0 + \theta_1 x + \theta_2 x^2$, as follows:

Let

$$\boldsymbol{\phi}(x) := \begin{bmatrix} x^0 & x^1 & x^2 \end{bmatrix}^T \in \mathbb{R}^3$$

Then

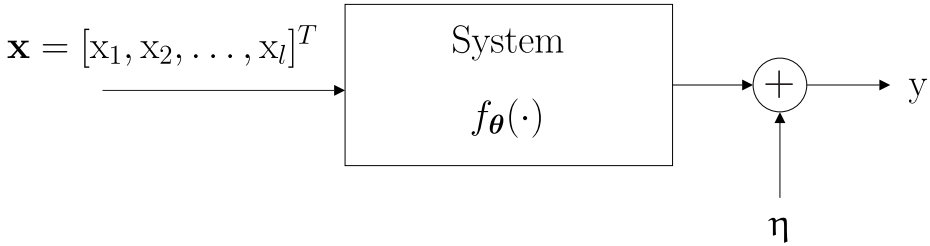
$$y(x) = \sum_{k=0}^{2} \theta_k \phi_k(x)$$

The Fourier series

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t}$$

Polynomials was one choice of basis, complex exponential was another choice of basis

Linear Regression



Let

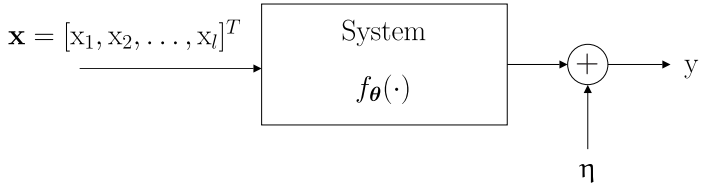
$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \theta_0 + \theta_1 \mathrm{x}_1 + \cdots + \theta_l \mathrm{x}_l$$

This leads to the linear regression model, written as

$$\mathrm{y} = \boldsymbol{\theta}^T \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} + \eta$$

Often the "1" is implicit, and the model is written as $\mathrm{y} = \boldsymbol{\theta}^T \mathbf{x} + \eta$

Linear regression and relation to filters



Linear regression

$$\mathrm{y} = \boldsymbol{\theta}^T \mathbf{x} + \eta$$

Is the system now a LTI filter? Discuss for a few minutes

Response of a LTI system (from earlier slide)

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Loss functions – how do we actually estimate θ from data?

Loss function

$$J(\boldsymbol{\theta}) := \sum_{n=1}^{N} \mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))$$

To do parameter estimation, we minimize the loss

$$\boldsymbol{\theta}_* = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

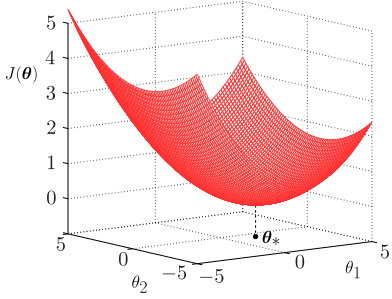
A popular choice is the Least-Squares loss function

Least-Squares (LS) loss function

$$\mathcal{L}(y_n, f_{\boldsymbol{\theta}}(\boldsymbol{x}_n)) = (y_n - f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))^2$$

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} (y_n - f_{\boldsymbol{\theta}}(\boldsymbol{x}_n))^2$$

The LS cost function



From Appendix A.1, we have the following

$$\frac{\partial \boldsymbol{a}^T \boldsymbol{x}}{\partial \boldsymbol{x}} = \frac{\partial \boldsymbol{x}^T \boldsymbol{a}}{\partial \boldsymbol{x}} = \boldsymbol{a}$$

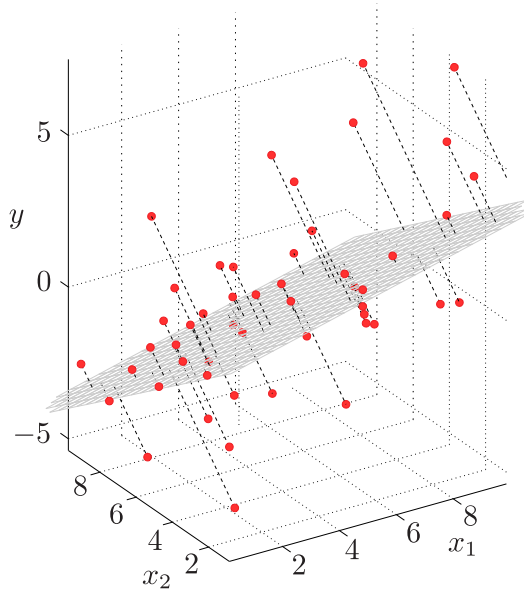
$$\frac{\partial \boldsymbol{x}^T A \boldsymbol{x}}{\partial \boldsymbol{x}} = \left(A + A^T\right) \boldsymbol{x}$$

LS Estimate

$$\boldsymbol{\theta}_* = (X^T X)^{-1} X^T \boldsymbol{y}$$

To obtain the LS estimate, put the derivative, $\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} J(\boldsymbol{\theta}) = 0$ and isolate for $\boldsymbol{\theta}$.

Example



Summary

- Parameter estimation is about the quality of the model parameter, and how to compute them.
- We use loss functions to measure how well we estimate the parameters.
- To keep it simple, we will work with a linear model.
- Linear regression in statistics/ML corresponds to linear filtering in DSP.
- The estimated parameters are uncertain, if we sample a new i.i.d dataset, the parameter estimate change, hence the parameters are themselves stochastic.

Biased and unbiased estimation

The estimator is random

Consider; the observations are random; our dataset $\mathcal{D} = (\boldsymbol{y}, X)$ will change, our observed data is only one realization of the underlying stochastic process (we will get back to this next week). Even if x is held constant, y will still change every time we "measure":

$$\mathrm{y} = \boldsymbol{\theta}^T \mathbf{x} + \eta$$

Our estimator is random in itself

Estimator of the unknown vector θ

$$\hat{\boldsymbol{\theta}} = g(\mathbf{y}, \mathrm{X})$$

Definitions

$\hat{\theta}$ is the estimate of the optimal value θ_o.

Bias definition

$$\mathrm{Bias}[\hat{\theta}] = \mathbb{E}[\hat{\theta}] - \theta_o$$

That means an unbiased estimater ($\mathrm{Bias}[\hat{\theta}] = 0$) attains the value $\mathbb{E}[\hat{\theta}] = \theta_o$.

Variance definition

$$\mathrm{Var}[\hat{\theta}] = \mathbb{E}\left[\left(\hat{\theta} - \mathbb{E}[\hat{\theta}]\right)^2\right]$$

Mean Squared Error definition

$$\mathrm{MSE}[\hat{\theta}] = \mathbb{E}\left[\left(\hat{\theta} - \theta_o\right)^2\right]$$

Which of these should we minimize?

Bias–Variance decomposition

The mean squared error can be decomposed as follows

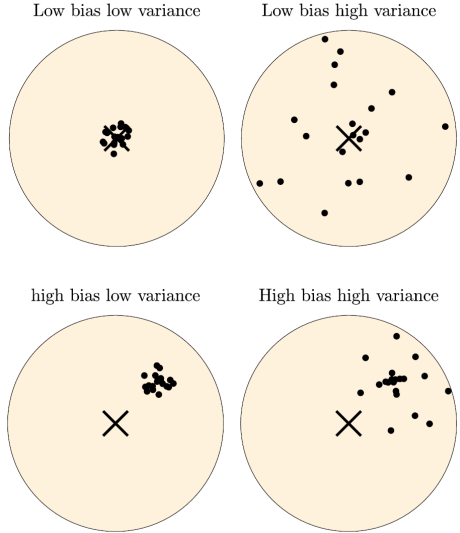
$$\begin{aligned}\mathrm{MSE}[\hat{\theta}] &= \mathbb{E}\left[\left(\hat{\theta} - \theta_o\right)^2\right] \\ &= \mathbb{E}\left[\left(\left(\hat{\theta} - \mathbb{E}[\hat{\theta}]\right) + \left(\mathbb{E}[\hat{\theta}] - \theta_o\right)\right)^2\right]\end{aligned}$$

The cross–terms vanish, hence we are left with the following result

Bias–Variance decomposition

$$\mathrm{MSE}[\hat{\theta}] = \underbrace{\mathbb{E}\left[\left(\hat{\theta} - \mathbb{E}[\hat{\theta}]\right)^2\right]}_{\text{Variance}} + \underbrace{\left(\mathbb{E}[\hat{\theta}] - \theta_o\right)^2}_{\text{Bias}^2}$$

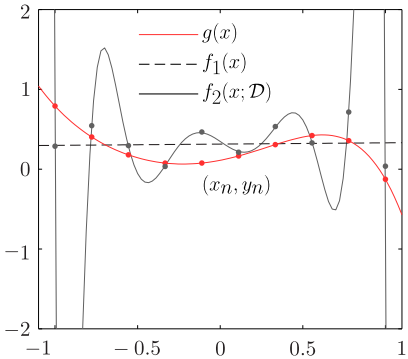
Examples of estimations



Bias–Variance dilemma

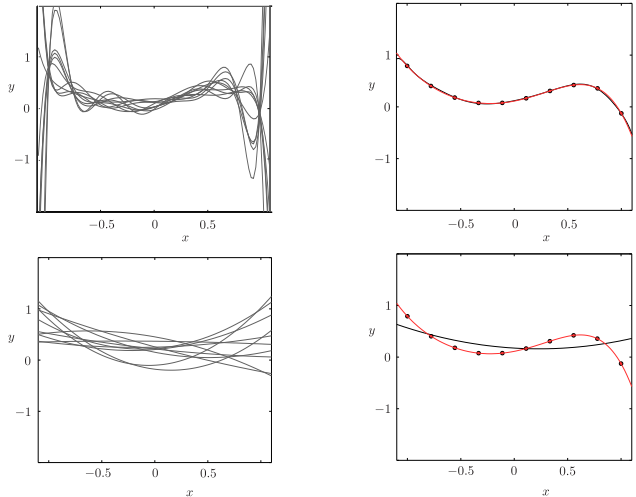
Bias-Variance tradeoff

$$\mathbb{E}_{\mathcal{D}}\left[(f(\mathbf{x};\mathcal{D}) - \mathbb{E}[y|\mathbf{x}])^2\right] = \underbrace{\mathbb{E}_{\mathcal{D}}\left[(f(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})])^2\right]}_{\text{Variance}} + \underbrace{(\mathbb{E}_{\mathcal{D}}[f(\mathbf{x};\mathcal{D})] - \mathbb{E}[y|\mathbf{x}])^2}_{\text{Bias}^2}$$



- Which model has lowest/highest bias?
- Which model has lowest/highest variance?

Bias Variance tradeoff examples (figure 3.8)



Unbiased estimation

How can we improve our estimation ?

Let us assume we have L datasets and consider an unbiased estimator $\hat{\theta}$, where, for each dataset we have the estimate $\hat{\theta}_i$.

$$\hat{\theta}^{(L)} = \frac{1}{L}\sum_{i=1}^{L}\hat{\theta}_i$$

Then $\hat{\theta}^{(L)}$ is an unbiased estimator with MSE

$$\mathrm{MSE}[\hat{\theta}^{(L)}] = \frac{1}{L}\mathrm{MSE}[\hat{\theta}_i]$$

You will derive this result in the exercise.

What is the "issue" with this result?

Biased estimation – an alternative route to reduce the MSE

Define the following biased estimator, where $\hat{\theta}_u$ is an unbiased estimator, $\alpha \in \mathbb{R}$

$$\hat{\theta}_b = (1 + \alpha)\hat{\theta}_u$$

Then, if we require $\mathrm{MSE}(\hat{\theta}_b) < \mathrm{MSE}(\hat{\theta}_u)$, we arrive at

$$-\frac{2\mathrm{MSE}(\hat{\theta}_u)}{\mathrm{MSE}(\hat{\theta}_u) + \theta_o^2} < \alpha < 0$$

And in addition, we can show that the norm of the biased estimator is smaller than the norm of the unbiased estimator

$$|\mathrm{MSE}(\hat{\theta}_b)| < |\mathrm{MSE}(\hat{\theta}_u)|$$

You will derive this result on your own in the exercise.

Ridge regression

The Ridge regression loss function is a norm shrinking loss:

Ridge regression

$$\mathcal{L}(\boldsymbol{\theta}, \lambda) = \sum_{n=1}^{N} \left(y_n - \boldsymbol{\theta}^T \boldsymbol{x}_n\right)^2 + \lambda \|\boldsymbol{\theta}\|_2^2$$

It can be shown (ML problem 3.12) that $\mathrm{MSE}(\hat{\theta}_b) < \mathrm{MSE}(\hat{\theta}_u)$ if

$$\lambda \in [0, \infty[, \qquad \theta_o^2 \leq \frac{\sigma_\eta^2}{N}$$

$$\lambda \in \left[0, \frac{2\sigma_\eta^2}{\theta_o^2 - \frac{\sigma_\eta^2}{N}}\right], \qquad \theta_o^2 > \frac{\sigma_\eta^2}{N}$$

Summary

- Bias-variance trade-off is crucial to understand, and offers direct interpretation of the source of errors and behavior of models.
- We have theoretic evidence that bias will lower MSE.
- The more data we have, the less regularization we need (the smaller the λ).

Maximum likelihood and Bayesian inference

Maximum likelihood estimation and Bayesian inference

Bayes theorem

$$p(\boldsymbol{\theta}|\mathcal{X}) = \frac{p(\mathcal{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{X})}$$

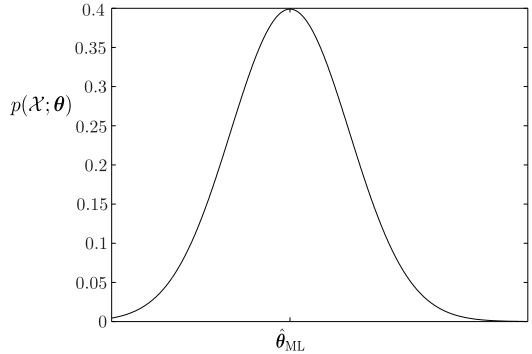
Taking the log we get

$$\ln p(\boldsymbol{\theta}|\mathcal{X}) = \ln p(\mathcal{X}|\boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(\mathcal{X})$$

$$\underbrace{\ln p(\boldsymbol{\theta}|\mathcal{X})}_{\text{log posterior}} \propto \underbrace{\ln p(\mathcal{X}|\boldsymbol{\theta})}_{\text{log likelihood}} + \underbrace{\ln p(\boldsymbol{\theta})}_{\text{log prior}}$$

- Using a normal distribution as likelihood leads to LS Estimate (ML example 3.7).
- Using a normal distribution as prior leads to Ridge regression (ML example 3.8).

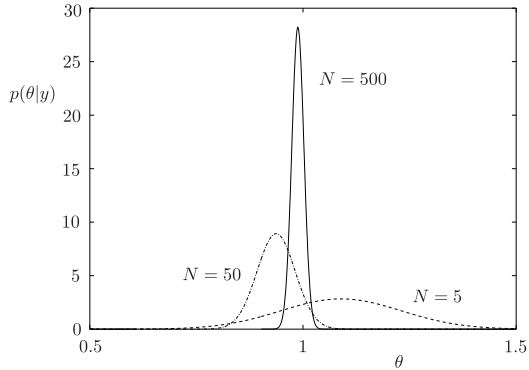
Maximum likelihood illustrated



Maximum likelihood estimation

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{X}|\boldsymbol{\theta})$$

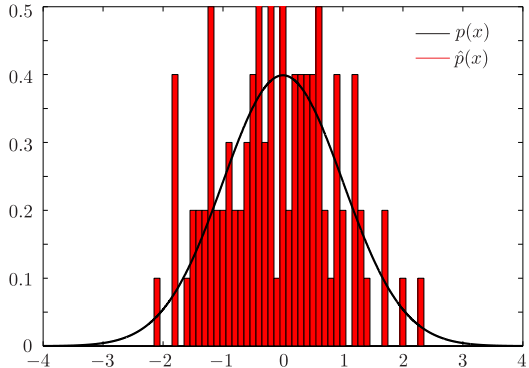
Bayesian inference illustrated

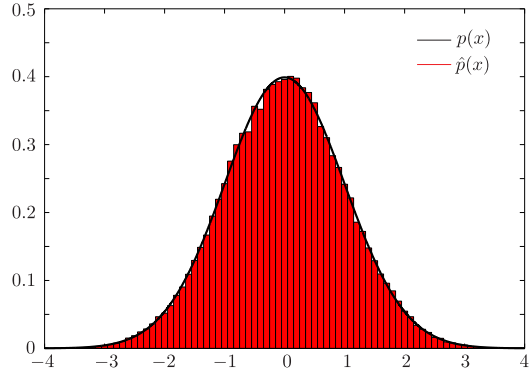


Maximum posterior estimation

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{X})$$

Bayesian inference using sampling





Outside the scope of this course – taught in

- 02443 Stochastic Simulation.
- 02477 Bayesian machine learning

Lecture Summary

- Matrix calculus is equivalent to ordinary calculus, but can make your life easier.
- Parameter estimations is concerned with estimating parameters of the function, using cost functions. But the estimations are random as well.
- We can do parameter estimation by loss functions, maximum likelihood, or Bayesian inference.
- Maximum likelihood offers an elegant way to understand the noise model.
- Bayesian inference offers an elegant way to learn the uncertainty of the parameter estimates.
- Bias-variance trade-off is crucial to understand, and offers direct interpretation of the source of errors and behavior of models.

Next week

Week 3 material; 2.4, 4.1–4.3, 4.5–4.7

Linear filtering:

- Stochastic Processes.
- The Wiener filter (linear filtering).
- Typical applications of the Wiener filter.