Interpolation and Regression

1) Interpolation

- General basis functions
- Special class of polynomials

} system of linear equations

↳ problematic because of strong asymptotic behaviour of polynomials (problem of class of functions)

↳ problem of ill-conditioned linear systems of equations

- numerical problem due to very different numbers within the same matrix
- check condition number (cond(V))

  if cond(V) is "big" it is difficult to treat problem numerically

↳ there exist high level functions in Matlab for interpolation

e.g. polyfit (fitting a polynomial)

polyval (evaluating a ")

2) Regression

↪ fit approximate functions
(→ simple functions) to data

Goal: Minimize error!

- there exist different models for errors (error functionals)

Def: $F[f(x)] = z$ functional
mapping from space of functions to real numbers

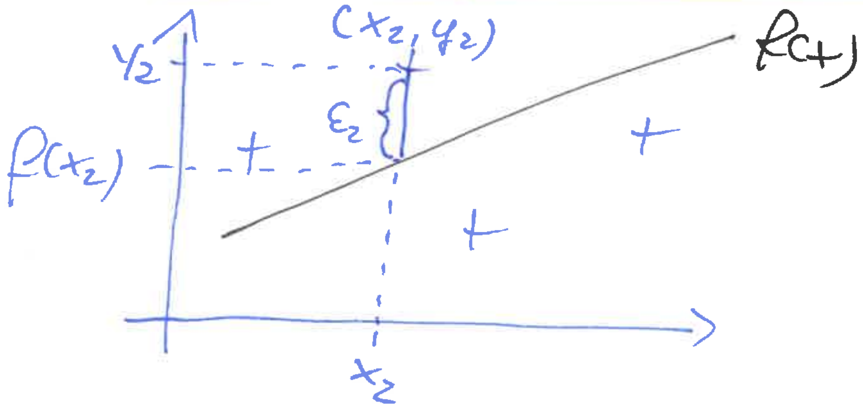
Choices in a regression problem:

- error functional
- class of functions
  here: $f(x) = a_1 x + a_0$
  a_1: slope, a_0: intercept
  → two parameters

Goal: Minimize error by choosing optimal values for a_1 and a_0

Linear Regression

(1) Error functional



Minimize total error (sum over all errors ε_i)

$$\varepsilon_i = y_i - f(x_i)$$

Error functional:
Mean squared error (MSE)

$$E = \sum_{i=1}^{N} [y_i - f(x_i)]^2$$

f depends on parameters

- no compensation of pos/neg contributions to total error
- don't use absolute value (not differentiable at $x=0$)

Trick for optimization:

Swap the role of parameters and variables

	Variables	Parameters
normal	x, $y = f(x)$	a_1, a_0
fitting	a_1, a_0	data (x_i, y_i)

Minimize error for linear regression

$$E(a_1, a_0) = \sum_{i=1}^{N} [y_i - (a_1 x_i + a_0)]^2$$

$$\frac{\partial E}{\partial a_1} = \sum_{i=1}^{N} 2[y_i - (a_1 x_i + a_0)](-x_i) \overset{!}{=} 0$$

$$\frac{\partial E}{\partial a_0} = \sum_{i=1}^{N} 2[y_i - (a_1 x_i + a_0)](-1) \overset{!}{=} 0$$

$$\underbrace{\begin{pmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} 1 \end{pmatrix}}_{A} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \underbrace{\begin{pmatrix} \sum_{i=1}^{N} y_i x_i \\ \sum_{i=1}^{N} y_i \end{pmatrix}}_{\vec{b}}$$

↳ efficient algorithm

Regression with arbitrary functions

- Error functional $E = \sum_{i=1}^{N} [y_i - f(t_i)]^2$
- Superposition of basis functions

$$f(t) = \sum_{i=1}^{M} \alpha_i \, g_i(t)$$

N: # data points
M: # basis functions

Set up a generalized Vandermonde matrix

$$A = \begin{pmatrix} g_1(t_1) & g_2(t_1) & \cdots & g_M(t_1) \\ g_1(t_2) & & & \\ g_1(t_3) & \vdots & \cdots & \vdots \\ \vdots & & & \\ g_1(t_N) & g_2(t_N) & & g_M(t_N) \end{pmatrix} \quad (N \times M)$$

Normally, $M \ll N$, i.e. matrix not square

Trick: Normal condition (Gauss)

$$A \vec{\alpha} = \vec{y}$$

$$A^T A \vec{\alpha} = A^T \vec{y}$$

multiplication from left with transpose matrix

$$\underbrace{(M \times N)(N \times M)}_{(M \times M)} \quad (M,1) \qquad (M \times N)(N,1)$$

Compare with linear regression

$g_1(x) = 1, \quad g_2(x) = x$

$$A = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & \vdots \\ \vdots & x_N \end{pmatrix} \quad A^T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 \dots & x_N \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \dots \\ x_1 & x_2 & x_3 & x_4 & \dots x_N \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} = \begin{pmatrix} N & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} x_i^2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & \dots \\ x_1 & x_2 & x_3 & \dots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \sum_i^N y_i \\ \sum_{i=1}^{N} x_i y_i \end{pmatrix}$$