Modelling 1

SUMMER TERM 2020

LECTURE 14

Least-Squares

Michael Wand · Institut für Informatik · Michael.Wand@uni-mainz.de

Least-Squares Fitting

Approximation

Common Situation:

- Many data points
  - Noisy data
  - Measurements, 3D scans, …
- Approximate with simple curve / surface

What we need:

- What is a good approximation?
- How to compute?

Approximation Techniques

Agenda:

- Least-squares approximation
  (and why/when this makes sense)
- Iteratively reweighted least-squares
  (for nasty noise distributions)
- Total least-squares linear approximation
  (get rid of the coordinate system)

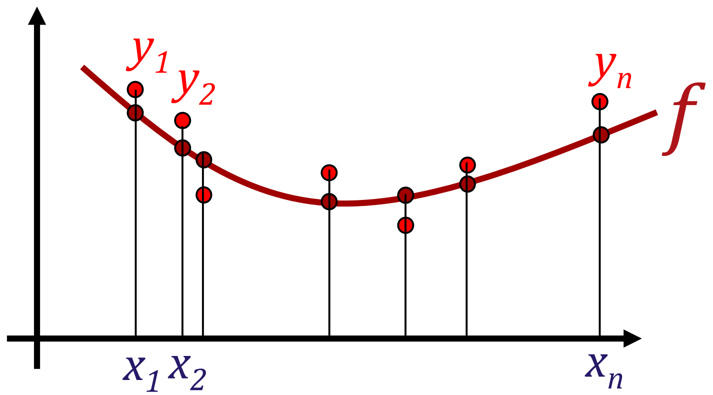
Least-Squares

Scenario for now:

- Function values y_i at positions x_i (1D $\rightarrow$ 1D)
  - Independent variables x_i known exactly.
  - Dependent variables y_i with some error.
- Error Gaussian, i.i.d.
  - *normal distributed*
  - *independent*
  - *same distribution* at every point
- Class of functions (basis) known

i.i.d. = "independent, indentically distributed"

Situation



Situation:

- Sample points taken at x_i from original f.
- Unknown Gaussian i.i.d. noise added to each y_i.
- Reconstruct $\tilde{f}$.

Summary

Statistical model: least-squares criterion

$$\arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2$$

Linear ansatz: quadratic objective

$$\tilde{f}_{\lambda_1,\dots,\lambda_k}(x) = \sum_{j=1}^{k} \lambda_j b_j(x) \longrightarrow \arg\min_{\lambda_1,\dots,\lambda_k} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{k} \lambda_j b_j(x_i) \right) - y_i \right)^2$$

Critical point: linear system

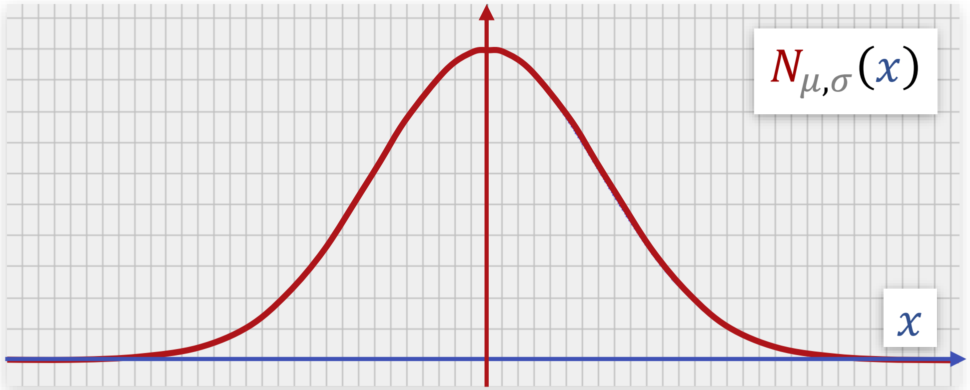
$$\begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle \mathbf{y}, \mathbf{b}_1 \rangle \\ \vdots \\ \langle \mathbf{y}, \mathbf{b}_k \rangle \end{pmatrix} \quad \text{with} \quad \begin{cases} \langle \mathbf{b}_i, \mathbf{b}_j \rangle := \sum_{t=1}^{n} b_i(x_t) \cdot b_j(x_t) \\ \langle \mathbf{y}, \mathbf{b}_i \rangle := \sum_{t=1}^{n} y_t \cdot b_i(x_t) \end{cases}$$

Maximum Likelihood Estimation

Goal:

- Maximize probability of $\tilde{f}$
  - Probability that measured data originated from $\tilde{f}$
- "Maximum likelihood estimation"

Error Model



Gaussian normal distribution

$$N_{\mu,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Assumption

- Error normal distributed
  - Independent for each data point
- Gaussian noise: maximum entropy for given variance
  - Unstructured noise

Maximum Likelihood Estimation

$$\arg\max_{\tilde{f}} \prod_{i=1}^{n} N_{0,\sigma}\left(\tilde{f}(x_i) - y_i\right) = \arg\max_{\tilde{f}} \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)$$

$$= \arg\max_{\tilde{f}} \ln\left[\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)\right]$$

$$= \arg\max_{\tilde{f}} \sum_{i=1}^{n} \left(\left(\ln\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)$$

$$= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(+\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)$$

$$= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2$$

Maximum Likelihood Estimation

$$\arg\max_{\tilde{f}} \prod_{i=1}^{n} N_{0,\sigma}\left(\tilde{f}(x_i) - y_i\right) = \arg\max_{\tilde{f}} \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)$$

$$= \arg\max_{\tilde{f}} \ln\left[\prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)\right]$$

$$= \arg\max_{\tilde{f}} \sum_{i=1}^{n} \left(\left(\ln\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}\right)$$

$$= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \frac{\left(\tilde{f}(x_i) - y_i\right)^2}{2\sigma^2}$$

$$= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2$$

Least-Squares Approximation

We have shown

- *Maximum likelihood estimate minimizes sum of squared errors*

Next: Compute optimal coefficients

- Linear ansatz: $\tilde{f}(x) = \sum_{j=1}^{k} \lambda_j b_j(x)$
- Determine optimal λ_i

Maximum Likelihood Estimation

Notation

$$\boldsymbol{\lambda} = \underbrace{\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix}}_{\updownarrow\ k \text{ entries}}, \qquad \mathbf{b}(\mathbf{x}) = \underbrace{\begin{pmatrix} b_1(\mathbf{x}) \\ \vdots \\ b_k(\mathbf{x}) \end{pmatrix}}_{\updownarrow\ k \text{ entries}}, \qquad \mathbf{b}_i = \underbrace{\begin{pmatrix} b_i(x_1) \\ \vdots \\ b_i(x_n) \end{pmatrix}}_{\updownarrow\ n \text{ entries}}, \qquad \mathbf{y} = \underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}}_{\updownarrow\ n \text{ entries}}$$

Maximum Likelihood Estimation

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix}, \quad \mathbf{b}(\mathbf{x}) = \begin{pmatrix} b_1(\mathbf{x}) \\ \vdots \\ b_k(\mathbf{x}) \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} b_i(x_1) \\ \vdots \\ b_i(x_n) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

↕ k entries ↕ k entries ↕ n entries ↕ n entries

$$\arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2 = \arg\min_{\boldsymbol{\lambda}} \sum_{i=1}^{n} \left(\sum_{j=1}^{k} \lambda_j b_j(x_i) - y_i \right)^2$$

$$= \arg\min_{\boldsymbol{\lambda}} \sum_{i=1}^{n} \left(\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{b}(x_i) - y_i\right)^2$$

$$= \arg\min_{\boldsymbol{\lambda}} \boldsymbol{\lambda}^{\mathrm{T}} \left[\sum_{i=1}^{n} \mathbf{b}(x_i) \mathbf{b}^{\mathrm{T}}(x_i) \right] \boldsymbol{\lambda} - 2 \sum_{i=1}^{n} y_i \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{b}(x_i) + \sum_{i=1}^{n} y_i^2$$

Maximum Likelihood Estimation

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix}, \quad \mathbf{b}(\mathbf{x}) = \begin{pmatrix} b_1(\mathbf{x}) \\ \vdots \\ b_k(\mathbf{x}) \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} b_i(x_1) \\ \vdots \\ b_i(x_n) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

↕ k entries ↕ k entries ↕ n entries ↕ n entries

$$\arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2 = \arg\min_{\boldsymbol{\lambda}} \sum_{i=1}^{n} \left(\sum_{j=1}^{k} \lambda_j b_j(x_i) - y_i \right)^2$$

$$= \arg\min_{\boldsymbol{\lambda}} \sum_{i=1}^{n} \left(\boldsymbol{\lambda}^{\mathrm{T}} \mathbf{b}(x_i) - y_i\right)^2$$

$$= \arg\min_{\boldsymbol{\lambda}} \underbrace{\boldsymbol{\lambda}^{\mathrm{T}} \left[\sum_{i=1}^{n} \mathbf{b}(x_i) \mathbf{b}^{\mathrm{T}}(x_i) \right] \boldsymbol{\lambda}}_{\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}} \underbrace{- 2 \sum_{i=1}^{n} y_i \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{b}(x_i)}_{\mathbf{b}\mathbf{x}} + \underbrace{\sum_{i=1}^{n} y_i^2}_{c}$$

→ quadratic optimization problem

Critical Point

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix}, \quad \mathbf{b}(\mathbf{x}) = \begin{pmatrix} b_1(\mathbf{x}) \\ \vdots \\ b_k(\mathbf{x}) \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} b_i(x_1) \\ \vdots \\ b_i(x_n) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

↕ k entries ↕ k entries ↕ n entries ↕ n entries

$$\nabla_{\boldsymbol{\lambda}} \left(\boldsymbol{\lambda}^{\mathrm{T}} \left[\sum_{i=1}^{n} \mathbf{b}(x_i) \mathbf{b}^{\mathrm{T}}(x_i) \right] \boldsymbol{\lambda} - 2 \sum_{i=1}^{n} y_i \boldsymbol{\lambda}^{\mathrm{T}} \mathbf{b}(x_i) + \sum_{i=1}^{n} y_i^2 \right)$$

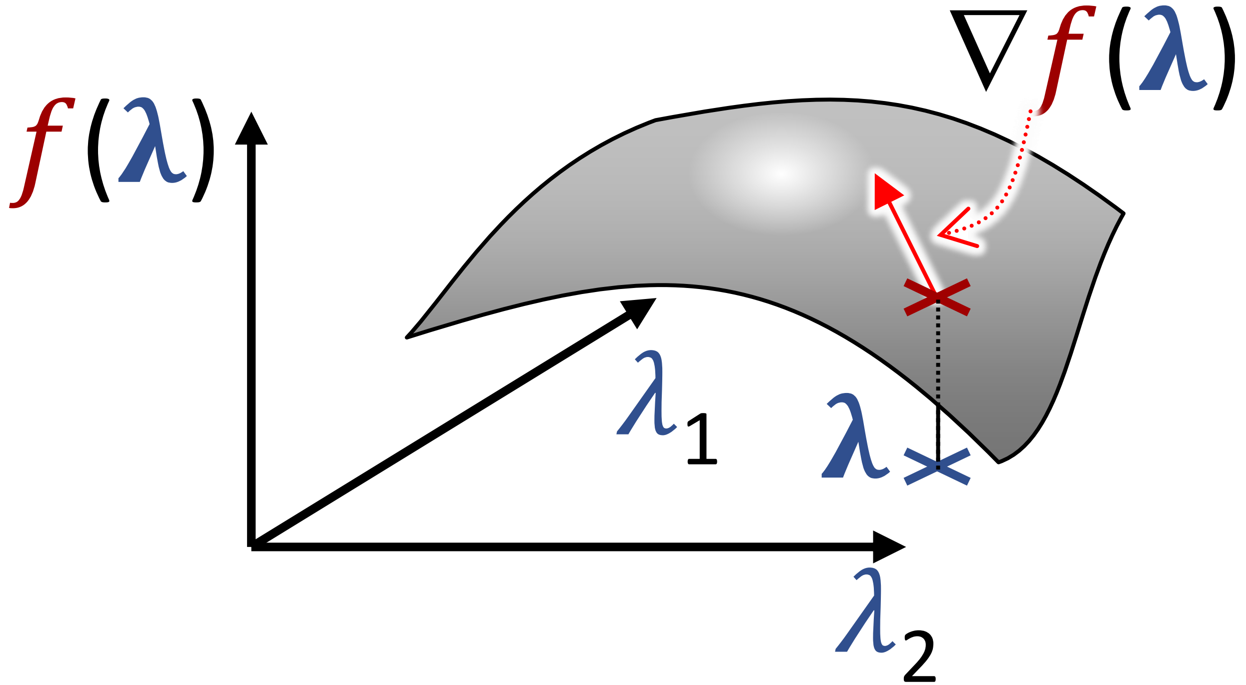
← Find minimum (critical point)

$$= 2 \left[\sum_{i=1}^{n} \mathbf{b}(x_i) \mathbf{b}^{\mathrm{T}}(x_i) \right] \boldsymbol{\lambda} - 2 \sum_{i=1}^{n} y_i \mathbf{b}(x_i)$$

Linear System:

$$\left[\sum_{i=1}^{n} \mathbf{b}(x_i) \mathbf{b}^{\mathrm{T}}(x_i) \right] \boldsymbol{\lambda} = \begin{pmatrix} \mathbf{y}^{\mathrm{T}} \mathbf{b}_1 \\ \vdots \\ \mathbf{y}^{\mathrm{T}} \mathbf{b}_k \end{pmatrix}$$

Gradient



$f(\boldsymbol{\lambda})$
$\nabla f(\boldsymbol{\lambda})$
λ_1
$\boldsymbol{\lambda}$
λ_2

Critical Point

$$\boldsymbol{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix}, \quad \mathbf{b}(\mathbf{x}) = \begin{pmatrix} b_1(\mathbf{x}) \\ \vdots \\ b_k(\mathbf{x}) \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} b_i(x_1) \\ \vdots \\ b_i(x_n) \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

↕ k entries ↕ k entries ↕ n entries ↕ n entries

Linear System

$$\left[\sum_{i=1}^{n} \mathbf{b}(x_i)\mathbf{b}^{\mathrm{T}}(x_i)\right]\boldsymbol{\lambda} = \begin{pmatrix} \mathbf{y}^{\mathrm{T}}\mathbf{b}_1 \\ \vdots \\ \mathbf{y}^{\mathrm{T}}\mathbf{b}_k \end{pmatrix}$$

Can be written as

$$\begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle \mathbf{y}, \mathbf{b}_1 \rangle \\ \vdots \\ \langle \mathbf{y}, \mathbf{b}_k \rangle \end{pmatrix} \text{ with } \begin{cases} \langle \mathbf{b}_i, \mathbf{b}_j \rangle := \sum_{t=1}^{n} b_i(x_t) \cdot b_j(x_t) \\ \langle \mathbf{y}, \mathbf{b}_i \rangle := \sum_{t=1}^{n} y_t \cdot b_i(x_t) \end{cases}$$

Summary (again)

Statistical model: least-squares criterion

$$\arg\min_{\tilde{f}} \sum_{i=1}^{n} \left(\tilde{f}(x_i) - y_i\right)^2$$

Linear ansatz: quadratic objective

$$\tilde{f}_{\lambda_1,\dots,\lambda_k}(x) = \sum_{j=1}^{k} \lambda_j b_j(x) \quad \longrightarrow \quad \arg\min_{\lambda_1,\dots,\lambda_k} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{k} \lambda_j b_j(x_i) \right) - y_i \right)^2$$

Critical point: linear system

$$\begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle \mathbf{y}, \mathbf{b}_1 \rangle \\ \vdots \\ \langle \mathbf{y}, \mathbf{b}_k \rangle \end{pmatrix} \quad \text{with} \quad \begin{cases} \langle \mathbf{b}_i, \mathbf{b}_j \rangle := \sum_{t=1}^{n} b_i(x_t) \cdot b_j(x_t) \\ \langle \mathbf{y}, \mathbf{b}_i \rangle := \sum_{t=1}^{n} y_t \cdot b_i(x_t) \end{cases}$$

Variants

Weighted least squares:

- Varying noise level
  - Varying standard deviations σ_i
- Weighted least squares problem
- Noisier points have smaller influence

Same procedure as prev. slides...

$$\arg\max_{\tilde{f}} \prod_{i=1}^{n} N_{0,\sigma_i}(\tilde{f}(x_i) - y_i) = \arg\max_{\tilde{f}} \prod_{i=1}^{n} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(\tilde{f}(x_i) - y_i)^2}{2\sigma_i^2}\right)$$

$$= \arg\max_{\tilde{f}} \ln\left[\prod_{i=1}^{n} \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(\tilde{f}(x_i) - y_i)^2}{2\sigma_i^2}\right)\right]$$

$$= \arg\max_{\tilde{f}} \sum_{i=1}^{n} \left(\left(\ln\frac{1}{\sigma_i\sqrt{2\pi}}\right) - \frac{(\tilde{f}(x_i) - y_i)^2}{2\sigma_i^2}\right)$$

$$= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \frac{(\tilde{f}(x_i) - y_i)^2}{2\sigma_i^2}$$

$$= \arg\min_{\tilde{f}} \sum_{i=1}^{n} \frac{1}{\sigma_i^2}(\tilde{f}(x_i) - y_i)^2$$

difference:

$\sigma \rightarrow \sigma_i$

Result

Linear system for the general case:

$$\begin{pmatrix} \langle \mathbf{b}_1, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_1, \mathbf{b}_k \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{b}_k, \mathbf{b}_1 \rangle & \cdots & \langle \mathbf{b}_k, \mathbf{b}_k \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle \mathbf{y}, \mathbf{b}_1 \rangle \\ \vdots \\ \langle \mathbf{y}, \mathbf{b}_k \rangle \end{pmatrix} \quad \text{with} \quad \begin{cases} \langle \mathbf{b}_i, \mathbf{b}_j \rangle := \sum_{t=1}^{n} b_i(x_t) \cdot b_j(x_t) \cdot \omega^2(x_t) \\ \langle \mathbf{y}, \mathbf{b}_i \rangle := \sum_{t=1}^{n} y_t \cdot b_i(x_t) \cdot \omega^2(x_t) \end{cases}$$

$\omega^2(x_t) = \frac{1}{\sigma_t^2}$, i.e. $\omega(x_t) = \frac{1}{\sigma_t}$

Larger $\omega \rightarrow$ larger influence of data point

Least-Squares Linear Systems

Least-squares solution to general linear system

- Consider

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

- Least-squares formulation

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\arg\min} (\mathbf{A}\mathbf{x} - \mathbf{b})^2$$

$$= \underset{\mathbf{x} \in \mathbb{R}^d}{\arg\min} \left(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} - 2\mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\mathbf{b} \right)$$

- Critical point: gradient = zero

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}$$

  - "System of normal equations"

SVD

Problem with normal equations:

- *Condition number* for normal equations = (*condition number* of $\mathbf{A}$)^2
- Proof
  - SVD: $\mathbf{A} = \mathbf{U}\,\mathbf{D}\,\mathbf{V}^T$
  - $\mathbf{A}^T\mathbf{A} = \mathbf{V}\,\mathbf{D}\,\mathbf{U}^T\,\mathbf{U}\,\mathbf{D}\,\mathbf{V}^T = \mathbf{V}^T\mathbf{D}^2\mathbf{V}$
- More stable (for bad problems)
  - Use SVD:
  - $\mathbf{A}^{-1} \approx \mathbf{A}^+ = \mathbf{V}\,\mathbf{D}^+\,\mathbf{U}^T$

    ("+" = pseudo-inverse, do not invert zero singular values)
- Effect: Pick smallest solution to normal Equations

Connection to Least-Squares Approx.

Equivalent results:

- Least-squares fitting of basis functions to data

same as

- Setting up over-constrained interpolation problem
- Then solve system of normal equations
  - Or pseudoinverse

Proof

- Elementary: Compare resulting equations

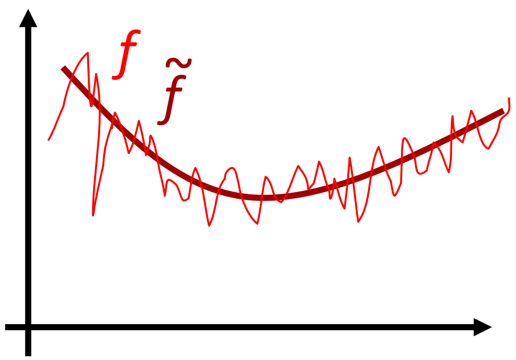
One more Variant...

Function Approximation

- Function given
  - $f: \Omega \supseteq \mathbb{R}^n \to \mathbb{R}$
  - Approximate by

$$\tilde{f}_{\lambda} = \sum_{i=1}^{d} \lambda_i b_i$$

- Difference: Continuous function as "data"
- Almost the same solution...



Function Approximation

Objective function:

- $\left\|\tilde{f}(\mathbf{x}) - f\right\|^2 \to \min$
- Solution

$$\left\|\sum_{i=1}^{k} \lambda_i\, b_i - f\right\|^2 = \left\langle \sum_{i=1}^{k} \lambda_i\, b_i - f, \sum_{i=1}^{k} \lambda_i\, b_i - f \right\rangle$$

$$= \boldsymbol{\lambda}^{\mathrm{T}} \begin{pmatrix} \langle b_1, b_1 \rangle & \cdots & \langle b_k, b_1 \rangle \\ \vdots & & \vdots \\ \langle b_1, b_k \rangle & \cdots & \langle b_k, b_k \rangle \end{pmatrix} \boldsymbol{\lambda} - \sum_{i=1}^{k} \lambda_i \langle b_i, f \rangle + \langle f, f \rangle$$

Function Approximation

Critical point (i.e., solution):

$$\begin{pmatrix} \langle b_1, b_1\rangle & \cdots & \langle b_k, b_1\rangle \\ \vdots & & \vdots \\ \langle b_1, b_k\rangle & \cdots & \langle b_k, b_k\rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle f, b_1\rangle \\ \vdots \\ \langle f, b_k\rangle \end{pmatrix}$$

with:

$$\langle f, g\rangle = \int_\Omega f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}, (\Omega \subseteq \mathbb{R}^D)$$ (unweighted version)

or

$$\langle f, g\rangle = \int_\Omega f(\mathbf{x}) g(\mathbf{x}) \omega^2(\mathbf{x}) d\mathbf{x}, (\Omega \subseteq \mathbb{R}^D)$$ (weighted version)

Galerkin Approximation

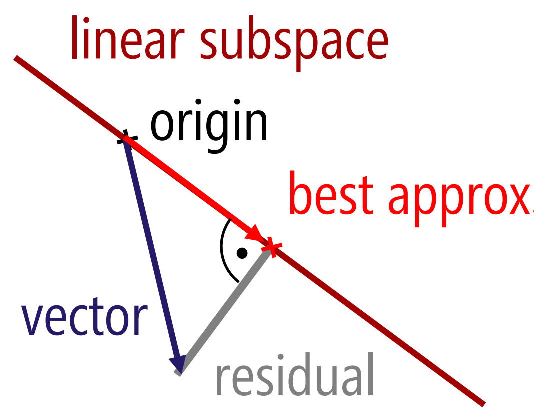
Least-squares criterion (here) equivalent to:

residual — each basis function

$$\forall i \in \{1, \ldots, k\}: \left\langle \sum_{j=1}^{k} \lambda_j b_j - f, b_i \right\rangle = 0$$

$$\Leftrightarrow \forall i \in \{1, \ldots, k\}: \left\langle \sum_{j=1}^{k} \lambda_j b_j, b_i \right\rangle = \langle f, b_i \rangle$$

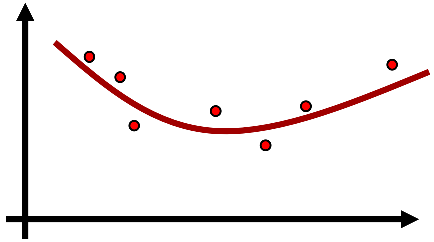
$$\Leftrightarrow \begin{pmatrix} \langle b_1, b_1 \rangle & \cdots & \langle b_k, b_1 \rangle \\ \vdots & & \vdots \\ \langle b_1, b_k \rangle & \cdots & \langle b_k, b_k \rangle \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix} = \begin{pmatrix} \langle f, b_1 \rangle \\ \vdots \\ \langle f, b_k \rangle \end{pmatrix}$$

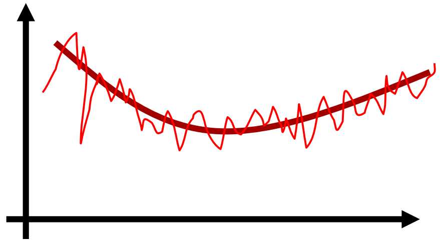


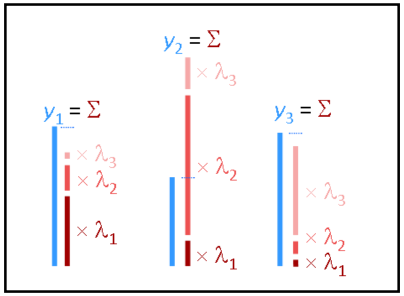
Summary

What we can do so far:

- Least-squares approximation:
  - Fit linear combination to data points
- Variants
  - Solve linear systems approximately
  - Fit functions to functions
- Extensions
  - Weights model varying uncertainty
  - The multi-dimensional case is similar







Remaining problems

What is missing:

- Gaussian noise only

  → Iteratively reweighted least-squares (M-estimators)

- Errors in $\mathbf{x}$-direction are ignored

  → Total least-squares