Convex Optimization and Modeling

Modeling

13th lecture, 07.07.2010

Jun.-Prof. Matthias Hein

Program of today

Modeling:

- What is Modeling ?
- Loss/Penalty Functions for Approximation/Regression
- Properties

Theory is done $\Longrightarrow$ All tools available

How to formulate optimization problems ?

$$\Downarrow$$

Modeling

What is Modeling ? Transition of the practical problem into the mathematical formulation

- differential equation,
- optimization problem,
- ...

Central question: How to enforce certain desired properties of the solution ?

- sparsity of the solution,
- robustness against small changes,
- ...

Properties of the mathematical formulation:

- Integration of all prior knowledge about the problem into the mathematical formulation. Examples:
  - desired solution is periodic,
  - desired solution is non-negative.
- The objective function corresponds to the criterion we are interested to minimize.

  But: for some problems true objective is unknown or difficult to grasp into mathematical formulation.

  Example: visual appealing reconstruction of noisy images.
- Is the problem of statistical nature:
  - Robustness against noise,
  - Worst-case versus average case.

Approximation/Regression: The norm approximation problem,

$$\min_{x \in \mathbb{R}^n} \|Ax - b\| .$$

General:

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \phi((Ax-b)_i) = \min_x \sum_{i=1}^{m} \phi\Big(\sum_{j=1}^{n} A_{ij}x_j - b_i\Big),$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ and $\phi : \mathbb{R}_+ \to \mathbb{R}_+$ is a **loss/penalty function**.

- **Projection** of b onto the subspace S spanned by the columns of A,

$$S = \Big\{ \sum_{i=1}^{n} a_i x_i \,|\, a_i \in \mathbb{R}^m, \quad A = (a_1, \ldots, a_n) \Big\}.$$

- find linear function $f_w(x) = \langle w, x \rangle$ which fits m data points (x_i, y_i)

Regression problem:

$$\min_x \sum_{i=1}^{m} \phi\Big(\langle w, x_i \rangle - y_i \Big).$$

Loss functions:

	$\phi(\lvert x\rvert)$
squared loss	$\lvert x\rvert^2$,
L_p - loss	$\lvert x\rvert^p$,
σ-insensitive	$(\lvert x\rvert-\sigma)\mathbb{1}_{\lvert x\rvert>\sigma}$,
Huber's robust loss	$\begin{cases} \frac{1}{2\sigma}\lvert x\rvert^2 & \text{if} \quad \lvert x\rvert \le \sigma \\ \lvert x\rvert - \frac{\sigma}{2} & \text{if} \quad \lvert x\rvert > \sigma, \end{cases}$,
log-barrier	$\begin{cases} -a^2 \log\left(1-\left(\frac{\lvert x\rvert}{a}\right)^2\right) & \text{if} \quad \lvert x\rvert \le a \\ \infty & \text{if} \quad \lvert x\rvert > a. \end{cases}$.

Table 1: Loss functions for regression. The σ-insensitive loss is called deadzone-linear penalty function in BV.

Examples of loss/penalty functions II

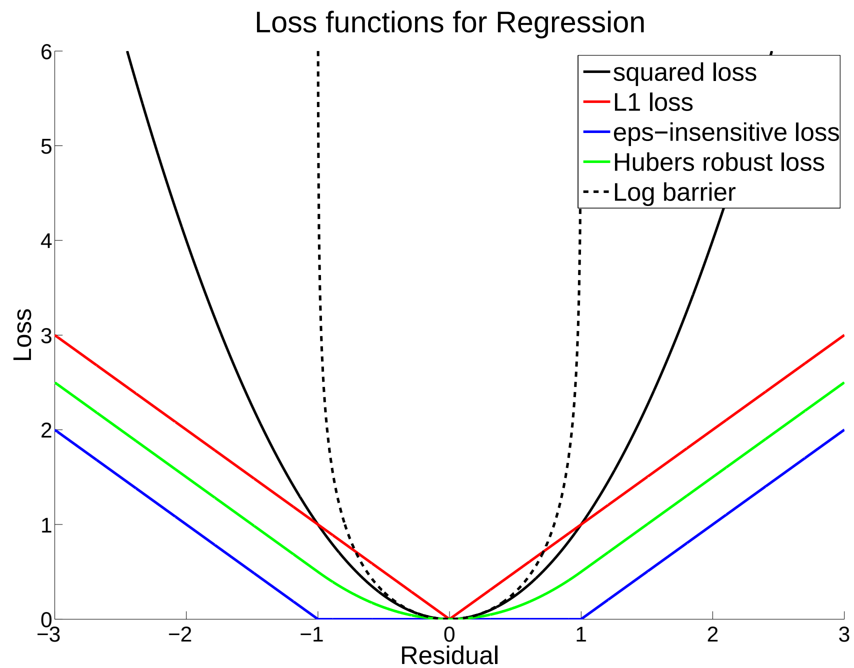


Figure 1: The log-barrier loss plotted for $a = 1 \Longrightarrow$ unbounded for $|x| \geq 1$. Huber and σ-insensitive loss are both plotted for $\sigma = 1$.

What is the influence of the loss function on the solution ?: The loss function quantifies how much we "dislike" the individual errors of each component,

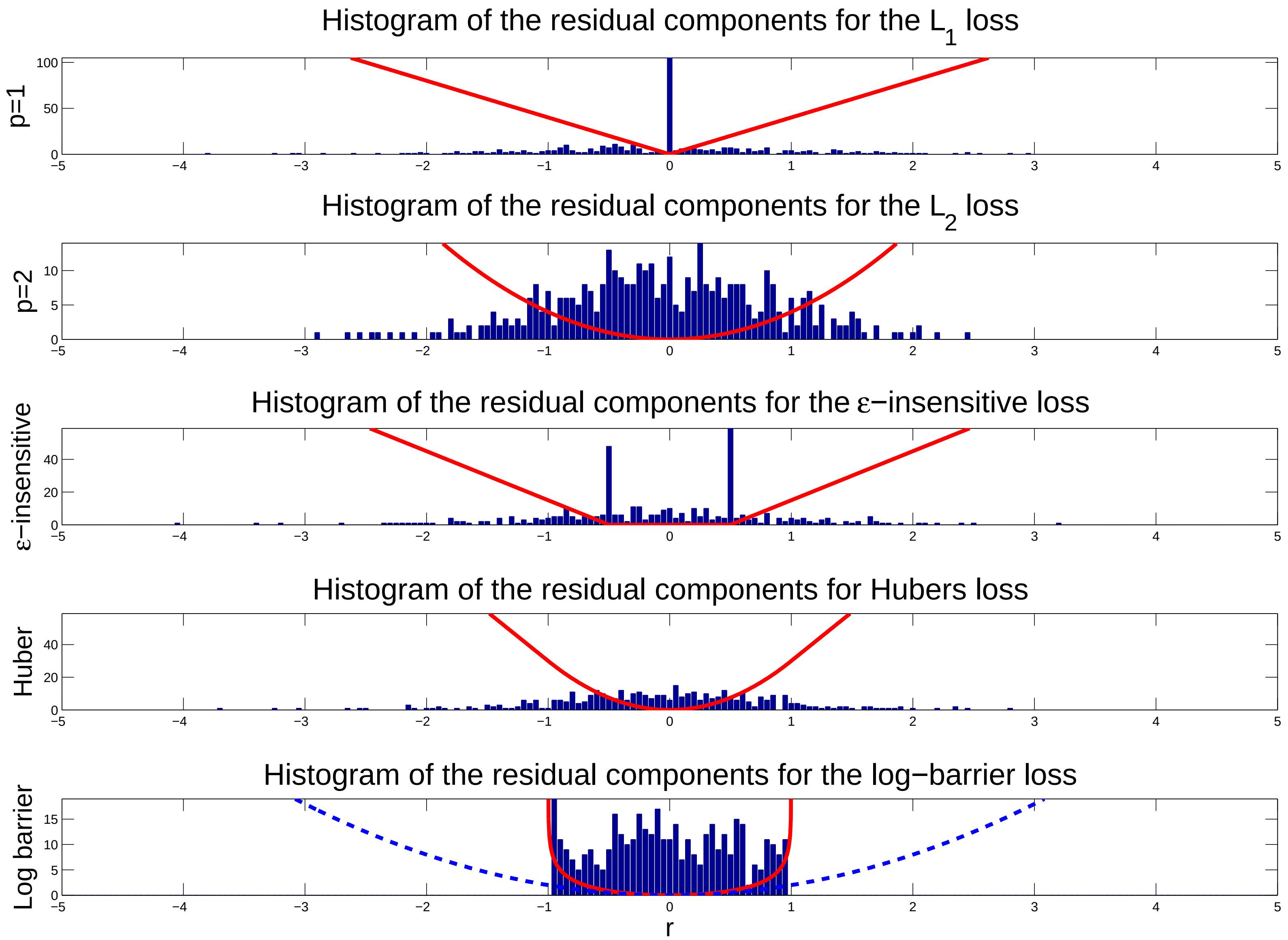
$$r_i = \sum_{j=1}^{n} A_{ij} x_j - b_i.$$

Given: solution $x^* = \arg\min_x \sum_{i=1}^m \phi(r_i)$.

How does the **residual** $r = Ax^* - b$ change when we use different loss functions ?

Histogram of the components of the residual, $r_i = (Ax^*)_i - b_i$.

Problem: $min_x\ \phi(Ax - b),\ A \in R^{400\times100},\ b \in R^{400}$



Properties of loss functions:

- The **L_2-loss** avoids large residuals - error is more evenly distributed,
- The **L_1-loss** penalizes proportional to the size of the residuals, large residuals can counter small residuals
- The **ε-insensitive loss** ignores small residuals and behaves for large residuals as the L_1-loss $\Longrightarrow$ huge fraction of residuals is in $[-\varepsilon, \varepsilon]$,
- The **Huber loss** grows as the L_2-loss for small residuals and as the L_1-loss for large residuals $\Longrightarrow$ histogram of residuals is a mixture of the histograms of L_2-loss and L_1-loss,
- The **log-barrier loss** behaves as the L_2-loss for small residuals,

$$\text{Taylor-approximation at } u = 0: \qquad \phi(u) \approx 2\frac{u^2}{a^2},$$

and no residuals larger than a are allowed.

Influence of loss/penalty functions III

	L_1-loss /Rel.	L_2-loss/Rel.	ε-ins./Rel.	Huber-loss/Rel.
Solution $x^*_{L_1}$	239.7 /1.00	323.7 /1.18	123.0 /1.19	263.5 /1.09
Solution $x^*_{L_2}$	262.0 /1.09	274.6 /1.00	111.3 /1.08	247.9 /1.02
Solution $x^*_{\varepsilon-\text{ins.}}$	266.6 /1.11	294.5 /1.07	103.3 /1.00	250.7 /1.03
Solution x^*_{Huber}	253.0 /1.06	282.9 /1.03	108.1 /1.05	242.5 /1.00

Table 2: The loss of the optimal solution x^* with respect to the other loss function. The left column gives the absolute loss and the second column gives the relative difference to the best.

Sparsity:

- compression setting - fit the data with less coefficients,
- regression setting - non-zero weights correspond to important features.

The L_1-norm is the best **convex** approximation of the cardinality function

$$\mathrm{card}(x) = \lim_{p \to 0} \|x\|_p^p = \sum_{i=1}^{n} \mathbb{1}_{|x_i|>0},$$

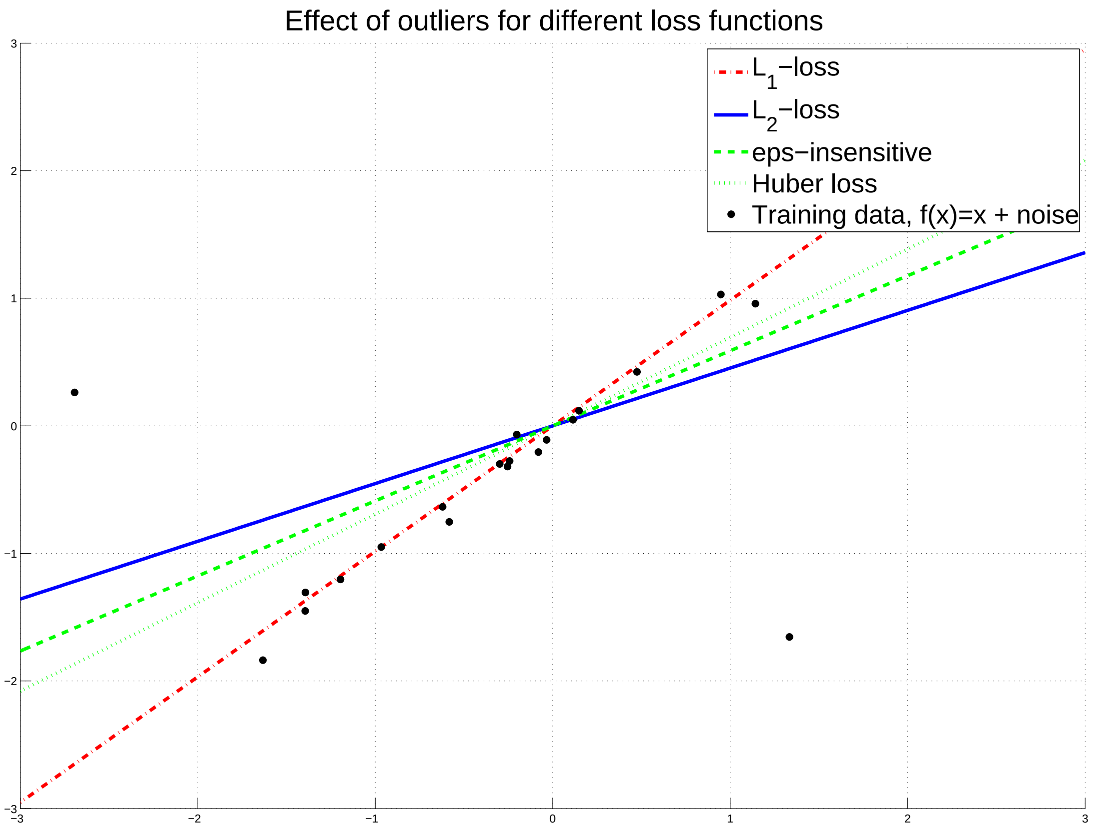
among all L_p-norms,

$$\|x\|_p^p = \sum_{i=1}^{n} |x_i|^p.$$

Sensitivity to outliers in a regression problem:

Model for data generation: output y_i is equal to true underlying function $f(x_i)$ plus an additive noise term ε_i (measurement noise),

$$y_i = f(x_i) + \varepsilon_i.$$



A regression problem in $\mathbb{R}$, where the true function is linear. The outputs are disturbed by small Gaussian noise and two outliers have been added. The L_1-loss is insensitive to the outliers whereas the L_2-loss is very much affected.

Sensitivity to outliers in a regression problem:

- the L_2**-loss** is very much affected by outliers since large residuals are heavily penalized,
- the L_1**-loss** is much less affected by outliers than the L_2-loss,
- the ε**-insensitive loss** forces the fitted linear function away from the "regular" training data towards the outliers. Basically, so that at all residuals of the "regular" training data are at the ε-boundary,,
- the **Huber loss** is between L_1-loss and L_2-loss
- the **log-barrier loss** produces in this case no feasible solution

Regularization:

Not always a good idea just to minimize the loss $\Longrightarrow$ bicriterion problem

$$\min\{\|Ax-b\|\,,\ \|x\|\},$$

Justification of the new criterion:

- limits the influence of a single dimension/feature x_i,
- sparse solution x e.g. by using $\|x\|_1$ as regularizer.

Pareto-optimal solutions of multi-criterion problems via scalarization,

$$\min_x \lambda_1 \|Ax-b\| + \lambda_2 \|x\|\,,$$

Since $\lambda_1, \lambda_2 > 0$ we can eliminate λ_1 and get

$$\min_x \|Ax-b\| + \lambda \|x\|\,,$$

where $\lambda = \frac{\lambda_2}{\lambda_1} \Longrightarrow \lambda$ is called **regularization parameter**.

Trade-off curve for different regularizers: L_2-loss for the term $Ax - b$ and L_1- or L_2-regularizer,

$$\min_x \|Ax - b\|_2 + \lambda \|x\|,$$

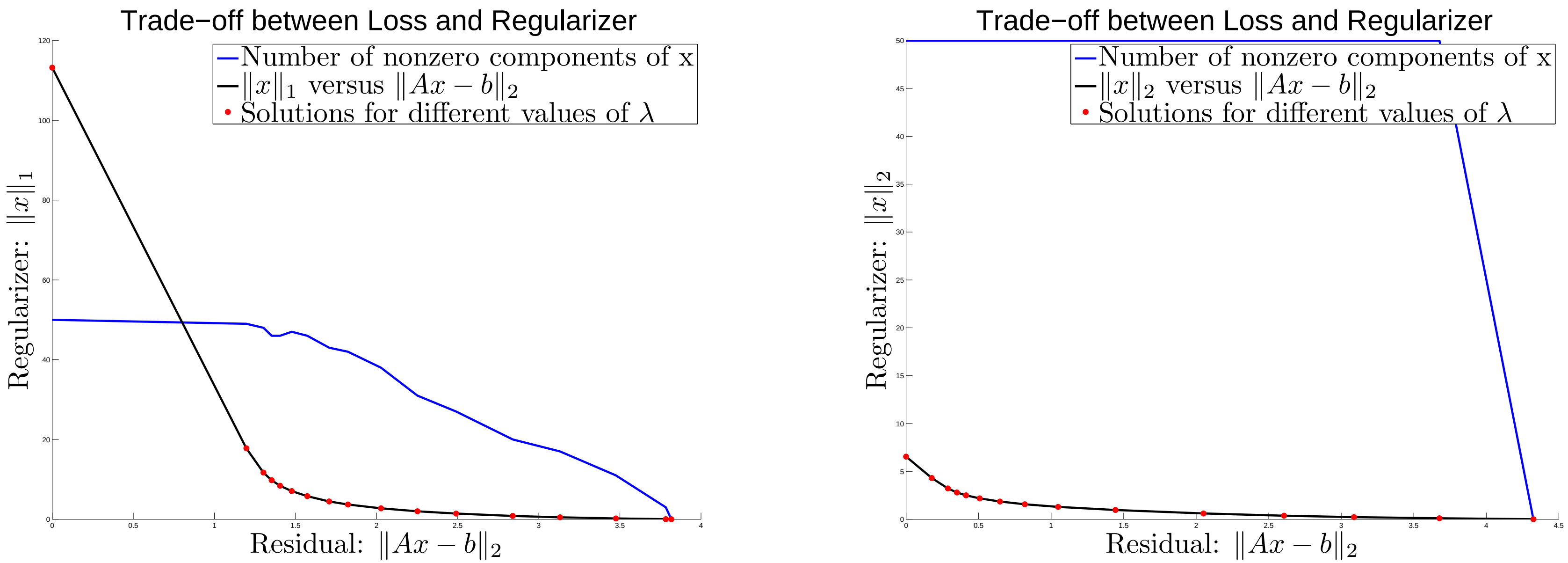


Figure 3: The optimization problem is solved for different values of λ, basically $\lambda \in \{2^k \,|\, k = -12, \ldots, 12\}$, which gives the red markers. The whole trade-off curves is obtained by linear interpolation. $x \in \mathbb{R}^{50}$ and $A \in \mathbb{R}^{50\times 50}$ has full rank $\Rightarrow x^* = A^{-1}b$ minimizes $\|Ax - b\|$.

How to represent functions: $f : \mathbb{R}^d \to \mathbb{R}$

Problem: All functions are an infinite-dimensional space !

- Parameterized function class: $f(x) = \sum_{i=1}^{d} w_i \phi_i(x)$, where ϕ_i is a set of basis functions (usually linearly independent),

  1. **linear functions** $\phi_i(x) = x_i$,
  2. **polynomials** (second order $\phi_{ij}(x) = x_i x_j$),
  3. **trigonometric basis** on $[0, 2\pi]$, $\phi = \{1, \sin(x), \cos(x), \sin(2x), \ldots\}$,

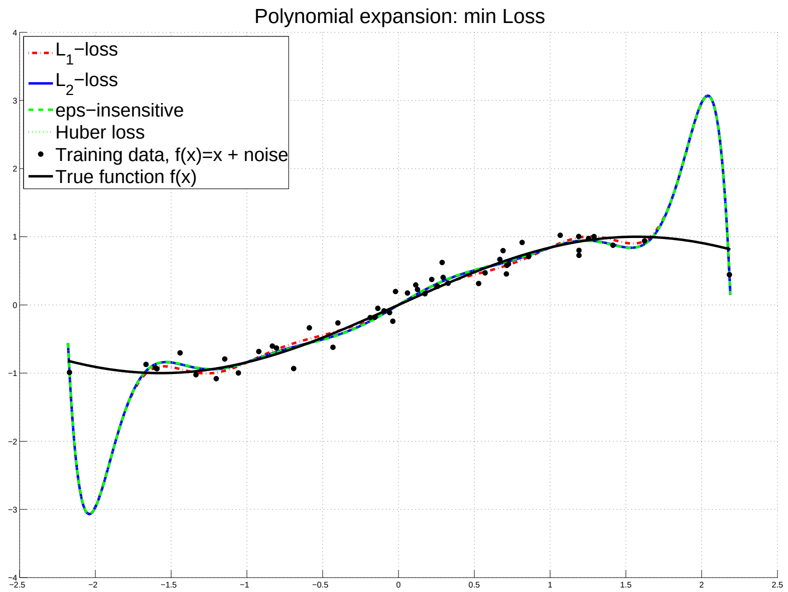
  In **RBF networks** one has for each training point X_i, $i = 1, \ldots, N$,

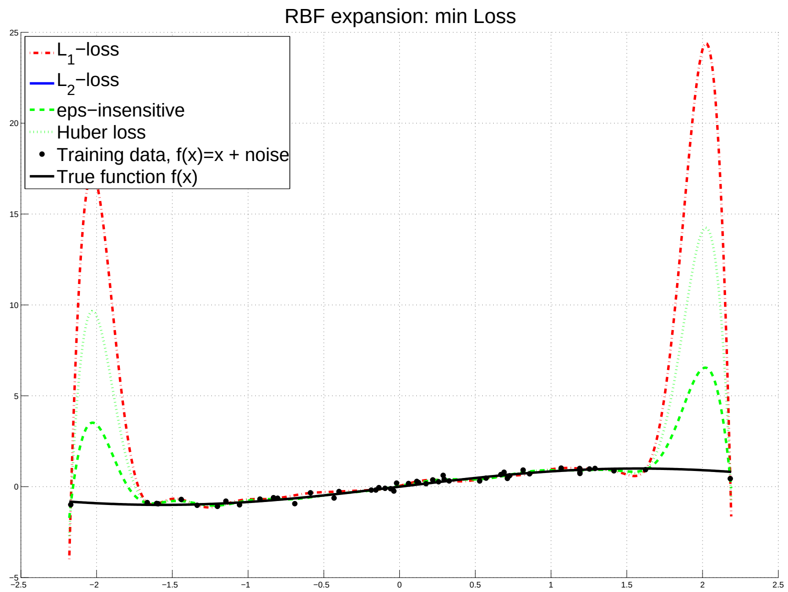
$$\phi_i(x) = e^{-\|x - X_i\|^2/(2\sigma^2)}, \qquad i = 1, \ldots, N.$$

- Discretize space (e.g. grid) $\Longrightarrow$ function is defined by values on the grid. $[0,1]^d$ discretized with spacing h yields $\left(\frac{1}{h}\right)^d$ points
  $\Rightarrow$ only possible in low dimension d.

Parameterized function classes:

More degrees of freedom $\implies$ data can be fitted much better !





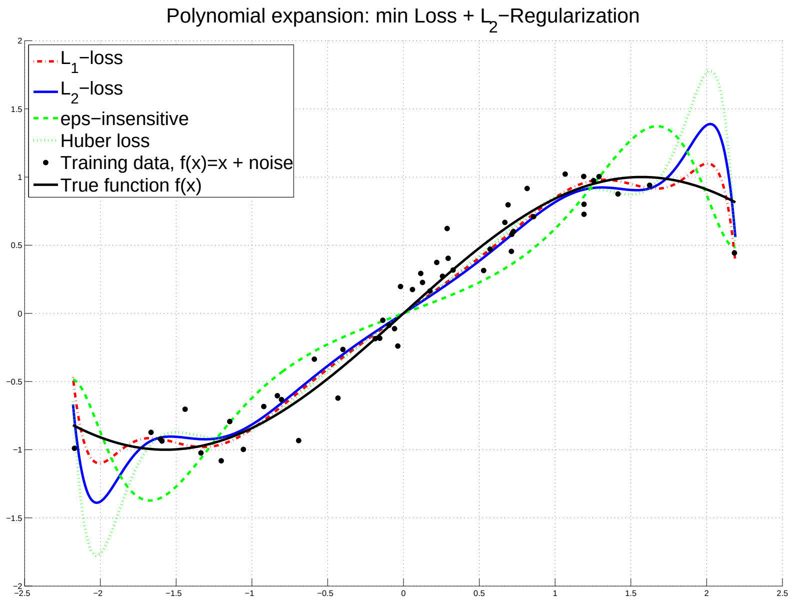
Left: Polynomials up to 7th order, Right: RBF network.

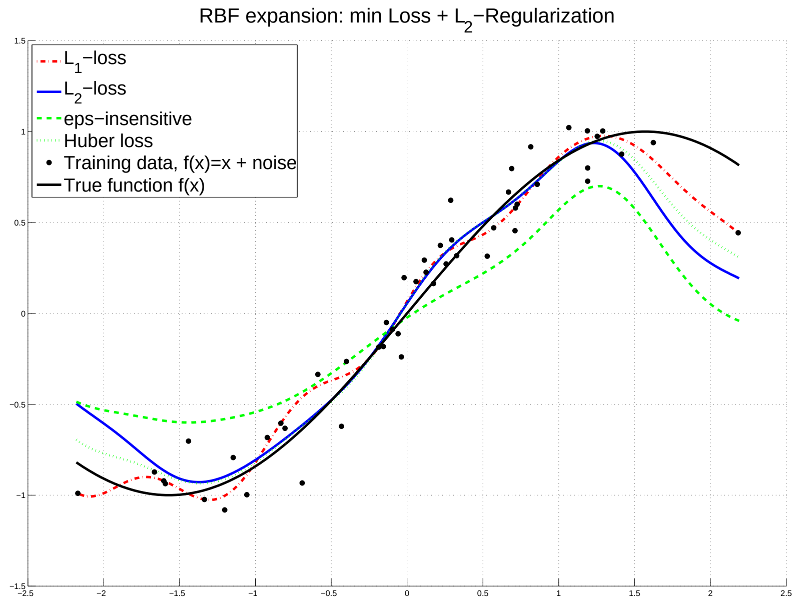
Single data points affect largely the solution - Why ?

$\implies$ use regularization to limit influence of one single parameter.

Parameterized function classes:

Regularization with $\|x\|_2 \quad \Longrightarrow \quad$ solution less influenced by outliers !





Left: Polynomials up to 7th order, Right: RBF network. The regularizer limits the influence of single outliers on the solution.

Problems with regularization of the weights:

- Regularization just on the parameters is unintuitive,
- requires that weights are comparable (scale-problem),

Solution:

- we want basically a smooth solution !
- enforce smoothness using regularization,
- use $\|Df\|$ as regularizer, where D is a differential operator.

Why do we want a smooth solution ?

The observed outputs/measurements Y are contaminated by noise ε_i

$$Y = f(X_i) + \varepsilon_i,$$

where

- f is the "true" underlying function,
- ε_i is a noise distribution e.g. Gaussian, shot-noise,...

Exact fit of the data means that we fit the noise but not the true function !

Overfitting !!!

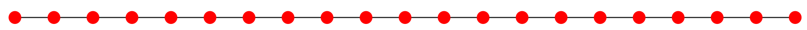
A smooth function will not fit the noise !

The level of smoothness depends on the noise level.

Discretized function class

Functions on a grid:

- discretization of $[-1,1]$ $x = -1, -1+h, -1+2h, \ldots, 1-2h, 1-h, 1,$



- function values are determined on the grid (apart from that no restrictions !)
- for simplicity: linear interpolation of f between grid points

How can we incorporate smoothness ?

- approximate derivatives using finite differences,
- different regularizers lead to different behavior,

Finite differences:

- $\frac{\partial f}{\partial x}\Big|_i = \frac{1}{h}\Big(f(i) - f(i-1)\Big)$ or $\frac{\partial f}{\partial x}\Big|_i = \frac{1}{2h}\Big(f(i+1) - f(i-1)\Big)$,
- $\frac{\partial^2 f}{\partial x^2}\Big|_i = \frac{1}{h^2}\Big(f(i+1) - 2f(i) + f(i-1)\Big)$

$\Longrightarrow$ approximated derivatives are just matrix multiplications with the discretized f

$$D = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \qquad Df = \begin{pmatrix} \text{f(2)-f(1)} \\ \text{f(3)-f(2)} \\ \text{f(4)-f(3)} \\ \text{f(5)-f(4)} \end{pmatrix}.$$

$\Rightarrow$ Regularize with: $\|Df\|_2^2$ (linear splines), $\|Df\|_1$ (total variation).

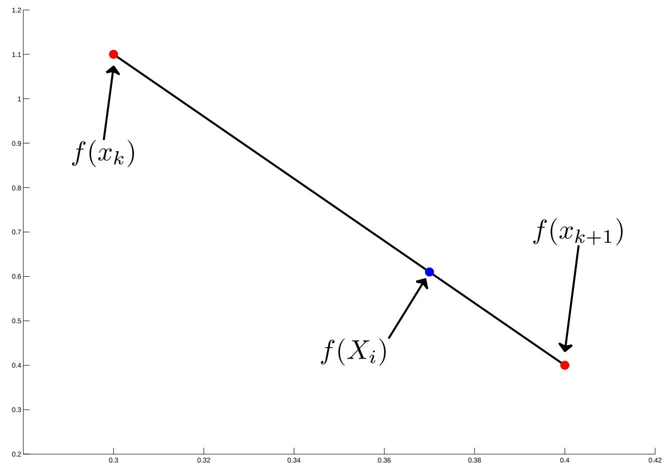
What is the effect of the different norms ?

$$\min_f \|If - Y\|_2^2 + \lambda\, \phi(Df).$$

Note that the training data (X_i, Y_i) need not lie on grid points.
$\Longrightarrow$ Interpolate f linearly at training data points.

$$f(X_i) = \frac{1}{h}\Big((x_{k+1} - X_i) f(x_k) + (X_i - x_k) f(x_{k+1})\Big),$$

where x_k is largest grid point smaller than X_i.

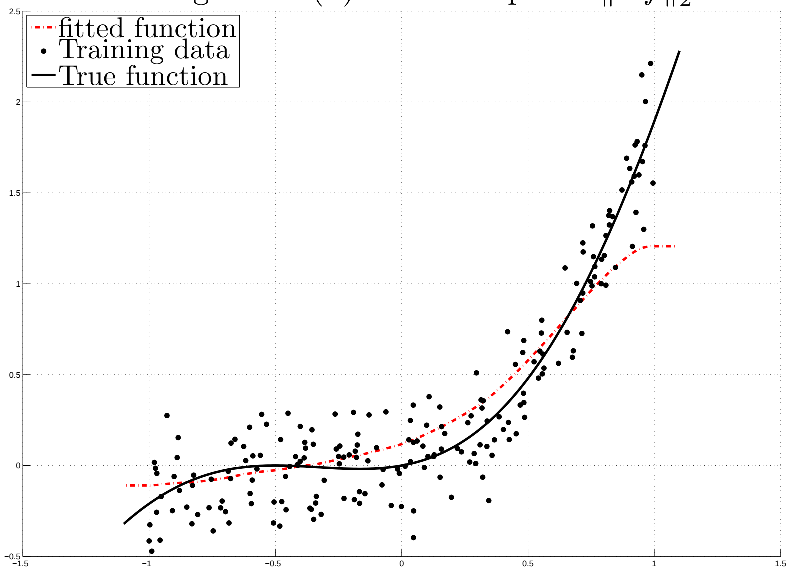


Total Variation versus Linear Splines

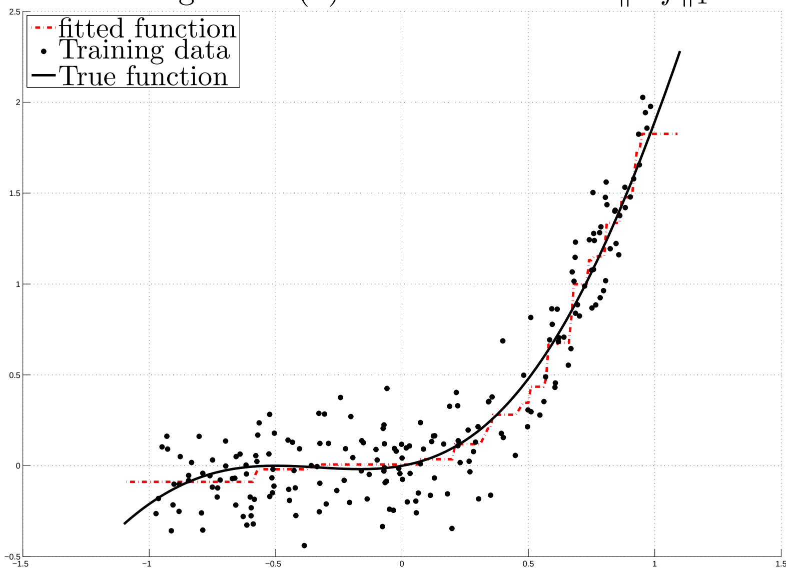
What is the effect of the different norms ?

$$\min_f \|If - Y\|_2^2 + \lambda\, \phi(Df).$$

Fitting: $x^2 \sin(x)$ - Linear Spline: $\|\nabla f\|_2^2$



Fitting: $x^2 \sin(x)$ - Total Variation: $\|\nabla f\|_1$



Setting: 200 samples of true function $x^2 \sin(x)$ with Gaussian noise $\sigma = 0.2$

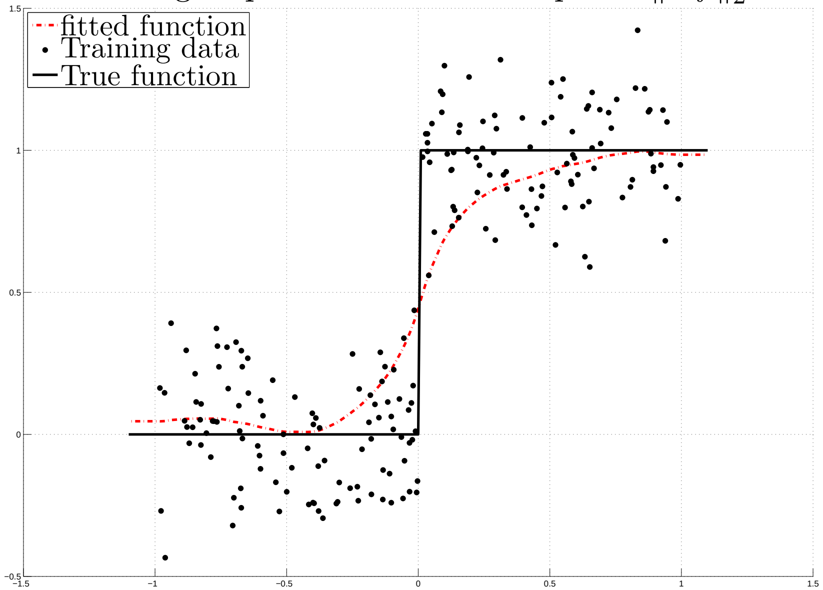
Left: Linear Splines $\|Df\|_2^2$, Right: Total Variation $\|Df\|_1$.

Total Variation versus Linear Splines II

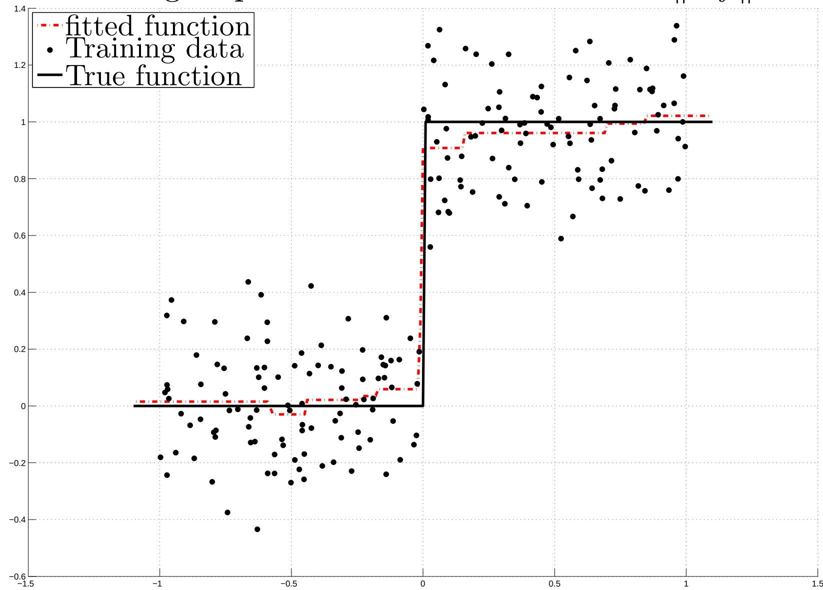
What is the effect of the different norms ?

$$\min_f \|If - Y\|_2^2 + \lambda\,\phi(Df).$$

Fitting step function - Linear Spline: $\|\nabla f\|_2^2$



Fitting step function - Total Variation: $\|\nabla f\|_1$

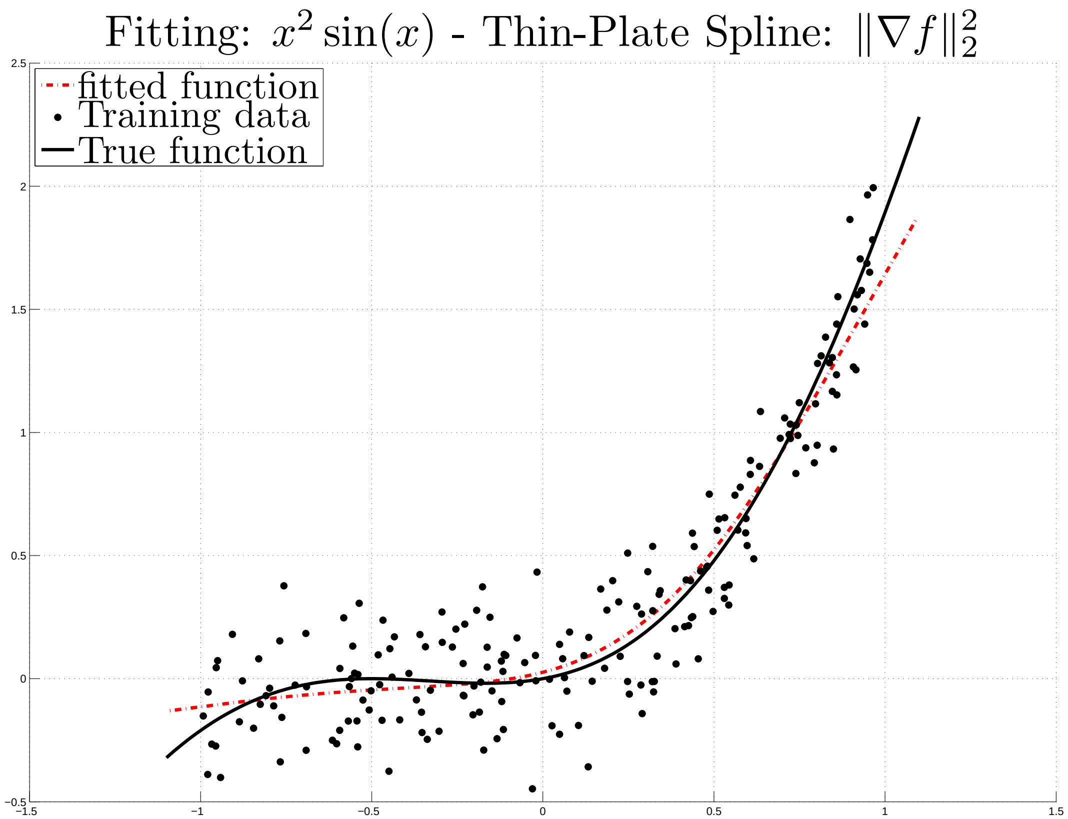


Setting: 200 samples of true function $\mathbb{1}_{x>0}$ with Gaussian noise $\sigma = 0.2$

Left: Linear Splines $\|Df\|_2^2$, Right: Total Variation $\|Df\|_1$.

Effect of the order of the differential operator

Higher order derivatives: Penalization of second-order derivatives leads to thin-plate splines.



Setting: 200 samples of true function $x^2 \sin(x)$ with Gaussian noise $\sigma = 0.2$

Thin-Plate-Regularizer penalizes second derivative (change of the first one) !

Effect of the order of the differential operator II

The null space of a differential operator D:

$$\{f \mid Df = 0\},$$

is the set of functions **not** penalized by the regularizer.

It does not cost anything to fit the data with functions from the null space !
Any deviation from the null space is penalized !

Null space of:

- first-order derivative: constant functions,
- second-order derivative: linear functions,
- k-th order derivative: $k-1$-order polynomials.