Overview

Regression

Definition, Loss function, Quadratic Loss, Linear Loss, Robustness

Risk Functionals

Expected, Empirical, and Regularized Risk

Maximum Likelihood Estimation

Equivalence to Risk Functionals, Common Noise Models, Problems and Advantages

Regularization

Basic Idea, Linear and Quadratic Terms, Matrices

Bayesian Estimation

Prior Probabilities, Posterior Probabilities, MAP Estimate, Inference

Regression

Data

Observations $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \subset \mathcal{X} \times \mathcal{Y}$ drawn from some underlying probability distribution $\Pr(\mathbf{x}, y)$.

Goal

We want to find a function $f : \mathcal{X} \longrightarrow \mathbb{R}$ which will tell us the value of f at $\mathbf{x}$.

The aim is to **predict** $f(\mathbf{x})$ at a **new** location $\mathbf{x}$.

Usual Assumption

No time correlation or order between observations.

Warning: this is not true for many cases. For instance, stock market, body height (people get taller over the ages), process control in factories (parameter drift and device degradation), meteorology (global warming).

Loss

Cost for Misprediction

- The cost for mispredicting the value of a stock (we invest and speculate that the shares will be worth y\$ at day $\mathbf{x}$).
- Penalty that a power company pays for misprediction of the power consumption the next day (they have to decide whether to turn on an extra power plant and whether to buy power from other companies)
- Friction of a manufactured device (excess of tolerances).
- Yield of a wafer fab.
- Quality of recommendation based on the prediction (social sciences).
- Efficiency of subsidies (e.g. we want to increase public spending and try to find a model for the expected spending spree before we commit the money).

Flashback: Risk

Loss Function

A mapping c from $\mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ into $\mathbb{R}$ with the property that for all $\mathbf{x}, y, f(\mathbf{x})$

1. $c(\mathbf{x}, y, f(\mathbf{x})) \geq 0$ (we can't win by getting things right)
2. $c(\mathbf{x}, y, y) = 0$ (we don't lose if we get it exactly right)

Empirical Risk

The average loss we incur on the training data

$$R_{\text{emp}}[f] := \frac{1}{m} \sum_{i=1}^{m} c(\mathbf{x}_i, y_i, f(\mathbf{x}_i))$$

Expected Risk

The average loss we incur on new (unseen) data

$$R[f] := \mathbf{E}\left[c(\mathbf{x}, y, f(\mathbf{x})\right] = \int_{\mathcal{X} \times \mathcal{Y}} c(\mathbf{x}, y, f(\mathbf{x})) d \Pr(\mathbf{x}, y)$$

Quadratic Loss

Loss

$c(\mathbf{x}, y, f(\mathbf{x})) = \frac{1}{2}(y - f(\mathbf{x}))^2$ This is the most commonly used loss function.

Optimization Problem $\underset{f}{\text{minimize}}\, R_{\text{emp}}[f] = \underset{f}{\text{minimize}}\, \frac{1}{2m}\sum_{i=1}^{m}(y_i - f(\mathbf{x}_i))^2$

Linear Model

We assume $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$. Hence we have to minimize

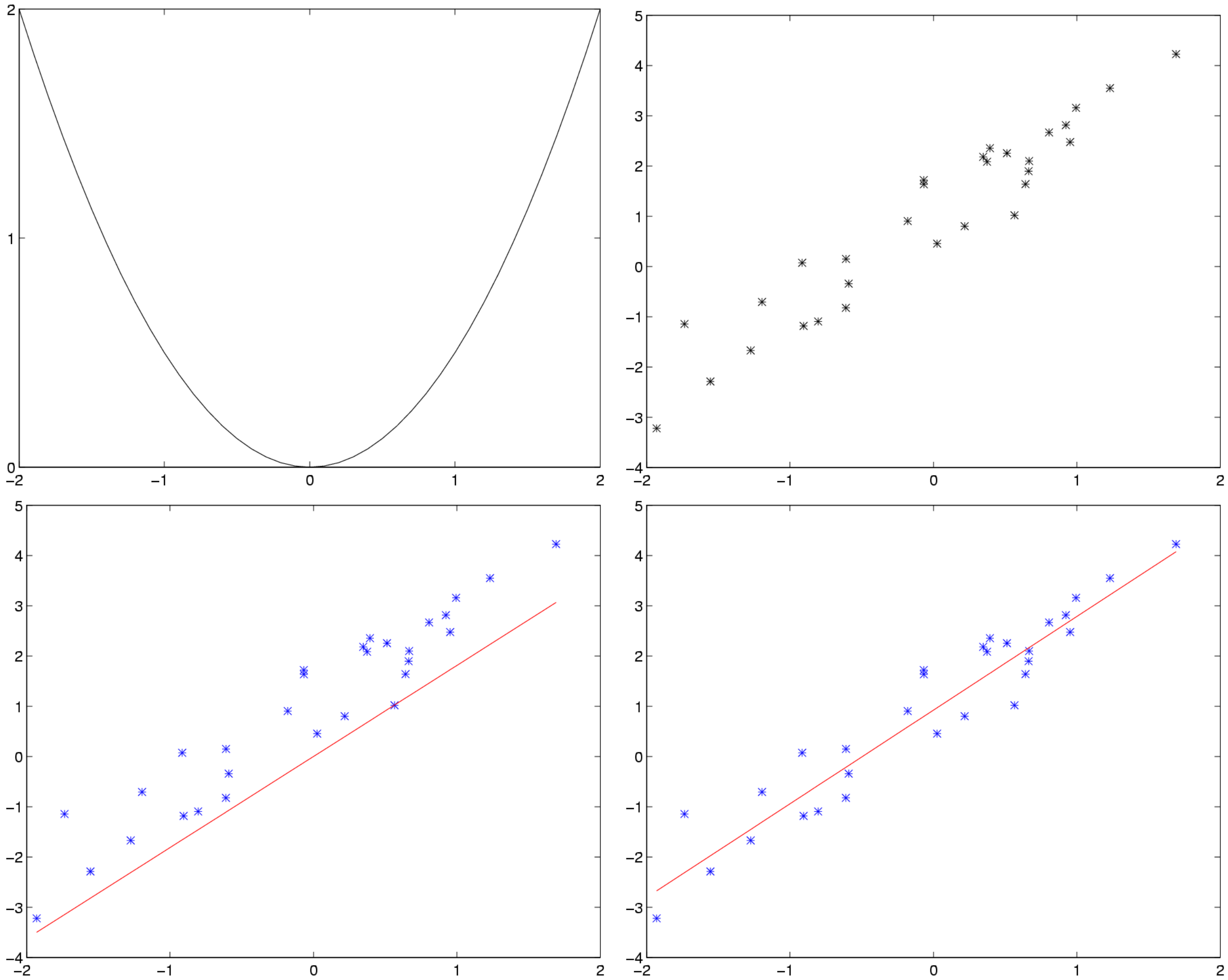
$$R_{\text{emp}}[f] = \frac{1}{2m}\sum_{i=1}^{m}(y_i - f(\mathbf{x}_i))^2 = \frac{1}{2m}\sum_{i=1}^{m}(y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2 = \frac{1}{2m}\|\mathbf{y} - X\mathbf{w}\|^2$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m) \in \mathcal{X}^m$ and $\mathbf{y}$ is the matrix of all y_i. This means

$$\partial_w R_{\text{emp}}[f] = \partial_w \left[\frac{1}{2m}\|\mathbf{y} - X\mathbf{w}\|^2\right] = \partial_w \left[\frac{1}{2m}\left(\mathbf{y}^\top \mathbf{y} - 2\mathbf{y}^\top X\mathbf{w} + \mathbf{w}^\top X^\top X\mathbf{w}\right)\right] = 0.$$

This is satisfied for $\mathbf{w} = (X^\top X)^{-1} X^\top \mathbf{y}$.

Example



- Quadratic loss
- Data in 1 dimension
- Linear approximation

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$$

- Linear and constant approximation

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

Pseudo Inverse

Problem

What happens if $(X^\top X)$ is not invertible?

Idea

After all, we only have to find $\mathbf{w}$ such that $X^\top = (X^\top X)\mathbf{w}$, so we have to invert $X^\top X$ only in the **image of** x under X.

Pseudo Inverse (also Moore-Penrose inverse)

Given $M \in \mathbb{R}^{m \times n}$ the pseudo inverse M^* of M has to satisfy

$MM^*Mx = Mx$ for all x.

Flashback: Inverse

Recall that for the inverse matrix we have

$M^{-1}Mx = x$ for all x.

Note that for matrices of full rank the two definitions are equivalent.

How to Compute the Pseudo Inverse

Singular Value Decomposition

We decompose $M \in \mathbb{R}^{m \times n}$ with $m \geq n$ into $U \Lambda V$ where $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{n \times n}$ have orthogonal rows / columns and Λ is diagonal. Then M^* is given by

$$M^* = V^\top \Lambda^* U^\top \text{ where } \Lambda^{-1} = \operatorname{diag}(\lambda_1^*, \ldots, \lambda_n^*)$$

where $\lambda^* = \lambda^{-1}$ for $\lambda \neq 0$ and $\lambda^* = 0$ otherwise.

Proof $M M^* M x = U \Lambda V V^\top \Lambda^* U^\top U \Lambda V x = U \Lambda \Lambda^* \Lambda V x = U \Lambda V x = M x$

Skinny Matrices

We assume that M has rank n (recall $m \geq n$). Consequently $M^\top M$ has full rank and therefore it can be inverted. Now we may compute M^* via

$$M^* = (M^\top M)^{-1} M^\top$$

Proof $M M^* M x = M (M^\top M)^{-1} M^\top M x = M x.$

Regression, Interpolation, Pseudo Inverse

Linear Model

We expand f into a set of basis functions $f_i : \mathfrak{X} \to \mathbb{R}$ where $i \in [n]$, hence

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i f_i(\mathbf{x}) \text{ and on training data } \mathbf{f} = F\alpha \text{ where } F_{ij} = f_j(\mathbf{x}_i)$$

Interpolation

Find function f for which $f(\mathbf{x}_i) = y_i$ for all i. This implies in general that $n = m$ and we obtain α via $\alpha = F^{-1}\mathbf{y}$.

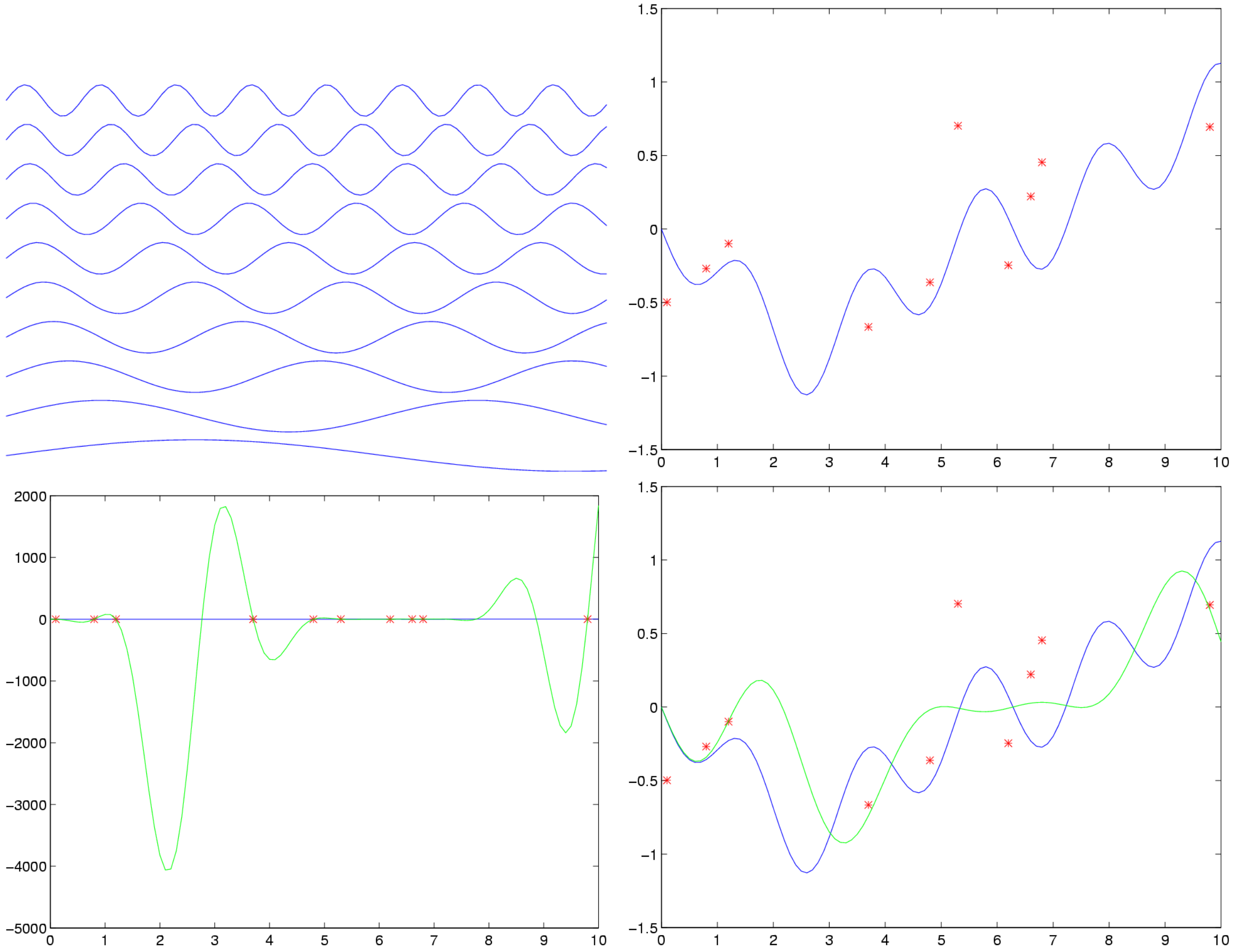
Regression

Find function f for which $f(\mathbf{x}) \approx y$ for all $\mathbf{x}, y$ drawn from $\Pr(\mathbf{x}, y)$. For mean squared loss this implies that we obtain α by minimizing

$$\frac{1}{m} \sum_{i=1}^{m} \frac{1}{2}(f(\mathbf{x}_i) - y_i)^2 = \frac{1}{2m} \|\mathbf{y} - F\alpha\|^2$$

The minimum is obtained for $\alpha = (F^\top F)^{-1} F^\top \mathbf{y}$. Typically $\text{rank} F < m$.

Example

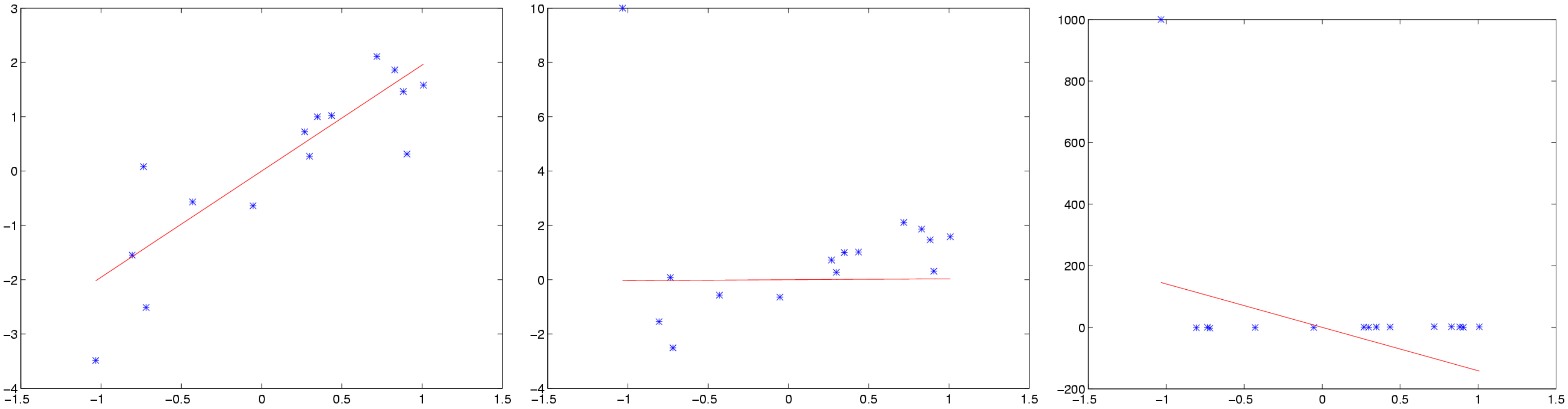


- Basis functions
- Noisy data and underlying function
- Interpolation with correct basis
- Regression with the first six terms of the basis

Problems with Squared Loss

Robustness

Single outliers (e.g., measurement errors) may destroy the estimate.



Linear regression with quadratic loss.

Squared loss may not be what we want

- linear cost of errors
- minimum error bounds (corridor estimator)

Linear Loss (L_1-loss)

Loss Function

$$c(\mathbf{x}, y, f(\mathbf{x})) = |y - f(\mathbf{x})|$$

This means that large losses are much les penalized than with squared loss. The estimator focuses more on the close by observations.

Optimization Problem

The loss function is not everywhere differentiable and we will obtain a linear program (see more on next slide)

$$\underset{f}{\text{minimize}} \frac{1}{m} \sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|$$

Linear Model

$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ where $\mathbf{x}, \mathbf{w} \in \mathbb{R}^n$.

$$R_{\text{emp}} = \frac{1}{m} \sum_{i=1}^{m} |y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle|$$

How to Solve It

Idea

Transform into linear optimization problem with constraints and feed into off-the-shelf optimizer.

Trick

Transform the absolute $|y|$ into two variables plus a constraint (this trick is quite useful for other problems, too).

$$|y| \Longleftrightarrow \left\{ \begin{array}{ll} \text{minimize} & \xi + \xi^* \\ \text{subject to} & y \leq \xi \\ & -y \leq \xi^* \end{array} \right\}$$

Linear Program

$$\begin{array}{ll} \text{minimize} & \displaystyle\frac{1}{m}\sum_{i=1}^{m} \xi_i + \xi_i^* \\ \text{subject to} & y_i - f(\mathbf{x}_i) \leq \xi_i \\ & y_i - f(\mathbf{x}_i) \geq -\xi_i^* \end{array}$$

Optionally substitute $f(\mathbf{x}_i)$ with $\langle \mathbf{w}, \mathbf{x}_i \rangle$.

Robustness of L_1-loss

Optimal Solution

At the solution the lhs and rhs derivatives of $R_{\text{emp}}[f]$ with respect to the parameters (of f) must satisfy

$$\partial_{w_i}|_{0-0} R_{\text{emp}}[f] \leq 0 \text{ and } \partial_{w_i}|_{0+0} R_{\text{emp}}[f] \geq 0$$

Derivative for L_1 Loss

The derivative only depends on the sign of $y_i - f(\mathbf{x}_i)$. Therefore changing a point via $y_i \to y_i + \Delta$ will not change the optimality of a solution unless the sign changes.

Properties of the Optimal Solution

An equal number of points will have $y_i - f(\mathbf{x}_i) > 0$ and $y_i - f(\mathbf{x}_i) < 0$.

Problems with Linear Loss

Problems

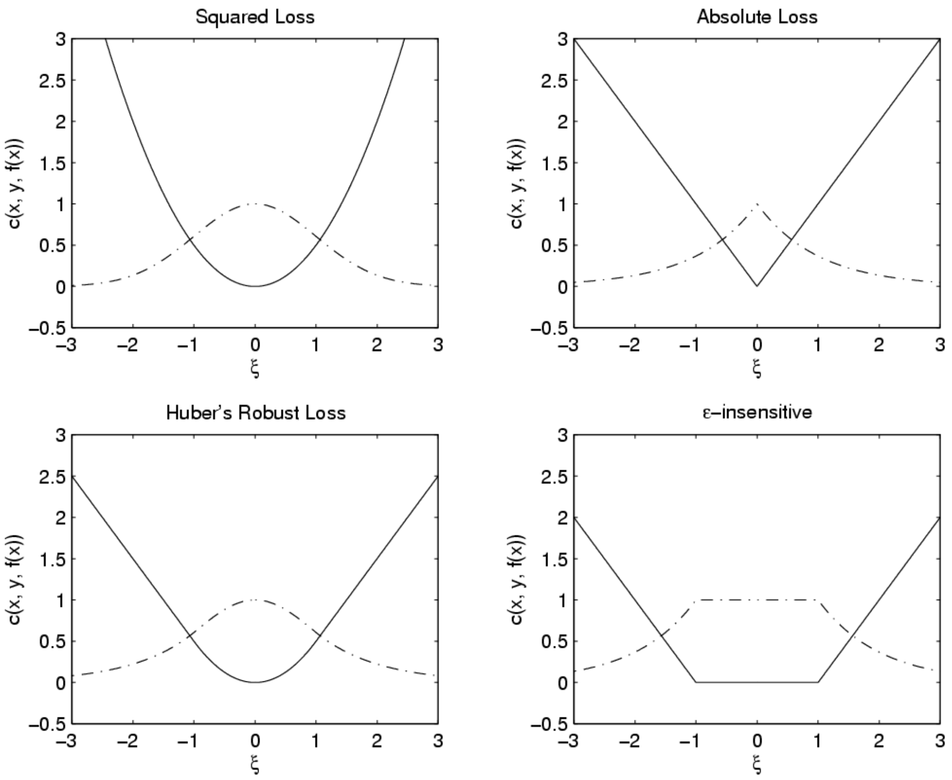
- Larger variance than with ℓ_2 loss
- Symmetric loss but **median** rather than **mean**
- More difficult to compute
- Quite often not exactly what we want

Improved Choices

- Combine properties of both loss functions (quadratic and linear) to obtain a robust and good regressor (Huber's Loss).
- We want some tolerance (ε-insensitive Loss)

$$c(\mathbf{x}, y, f(\mathbf{x})) = \max(0, |y - f(\mathbf{x})| - \varepsilon)$$

More Losses



- Quadratic Loss

$$c(\mathbf{x}, \mathbf{y}, f(\mathbf{x})) = \frac{1}{2}(y - f(\mathbf{x}))$$

- Linear Loss

$$c(\mathbf{x}, \mathbf{y}, f(\mathbf{x})) = |y - f(\mathbf{x})|$$

- Huber's Robust Loss

$$c(\mathbf{x}, \mathbf{y}, f(\mathbf{x})) = \begin{cases} \frac{1}{2\sigma}(y - f(\mathbf{x}))^2 & \text{for } |y - f(\mathbf{x})| \leq \sigma \\ |y - f(\mathbf{x})| + \frac{\sigma}{2} & \text{otherwise} \end{cases}$$

- ε-insensitive Loss

$$c(\mathbf{x}, y, f(\mathbf{x})) = \max(0, |y - f(\mathbf{x})| - \varepsilon)$$