Non-Quadratic Losses

Jong-Han Kim

EE787 Machine learning
Kyung Hee University

Penalty functions and error histograms

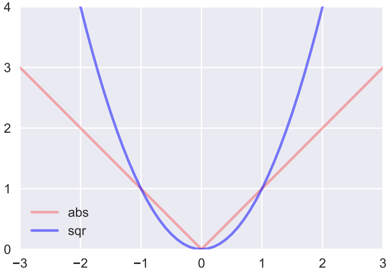
Loss and penalty functions

- empirical risk (or average loss) is $\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta^{\mathsf{T}} x^i, y^i)$
- the loss function $\ell(\hat{y}, y)$ penalizes deviation between the predicted value $\hat{y}$ and the observed value y
- common form for loss function: $\ell(\hat{y}, y) = p(\hat{y} - y)$
- p is the *penalty function*
- *e.g.*, the square penalty $p^{\mathsf{sqr}}(r) = r^2$
- $r = \hat{y} - y$ is the *prediction error* or *residual*

Penalty functions

- the penalty function tells us how much we object to different values of prediction error
- usually $p(0) = 0$ and $p(r) \geq 0$ for all r
- if p is *symmetric*, *i.e.*, $p(-r) = p(r)$, we care only about the magnitude (absolute value) of prediction error
- if p is *asymmetric*, *i.e.*, $p(-r) \neq p(r)$, it bothers us more to over- or under-estimate

Square versus absolute value penalty



- for square penalty $p^{\text{sqr}}(r) = r^2$
  - for small prediction errors, penalty is very small (small squared)
  - for large prediction errors, penalty is very large (large squared)
- for absolute penalty $p^{\text{abs}}(r) = |r|$
  - for small prediction errors, penalty is large (compared to square)
  - for large prediction errors, penalty is small (compared to square)

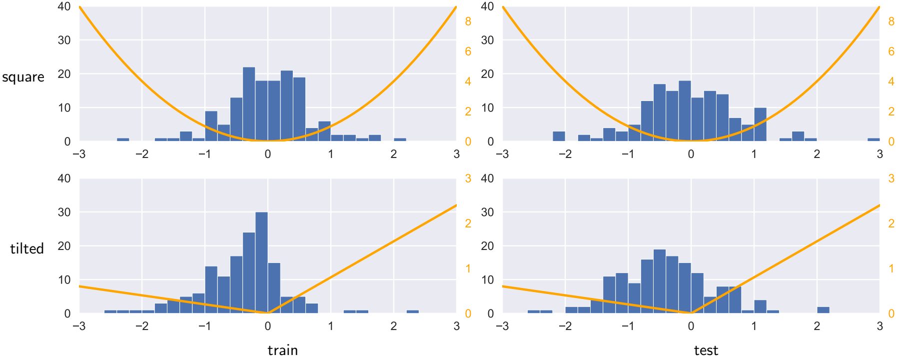
Predictors and choice of penalty function

- choice of penalty function depends on how you feel about large, small, positive, or negative prediction errors
- different choices of penalty function yield different predictor parameters
- choice of penalty function *shapes* the histogram of prediction errors, *i.e.*,

  $$r^1, \ldots, r^n$$

  (usually divided into bins and displayed as bar graph distribution)

Histogram of residuals



- artificial data with $n = 300$ and $d = 30$, using 50/50 test/train split
- plots show histogram of residuals $r^1, \ldots, r^n$
- tilted loss results in distribution with most residuals $r^i < 0$, *i.e.*, predictor prefers $\hat{y}^i < y^i$

Robust fitting

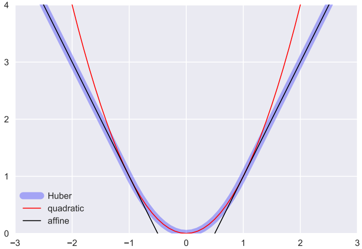
Outliers

- in some applications, a few data points are ‘way off’, or just ‘wrong’
- occurs due to transcription errors, error in decimal point position, *etc.*
- these points are called *outliers*
- even a few outliers in a data set can result in a poor predictor
- several standard methods are used to remove outliers, or reduce their impact
- one simple method:
  - create predictor from data set
  - flag data points with large prediction errors as outliers
  - remove them from the data set and repeat

Robust penalty functions

- we say a penalty function is *robust* if it has low sensitivity to outliers
- robust penalty functions grow more slowly for large prediction error values than the square penalty
- and so ‘allow’ the predictor to have a few large prediction errors (presumably for the outliers)
- so they handle outliers more gracefully
- a *robust predictor* might fit, *e.g.*, 98% of the data very well

Huber loss



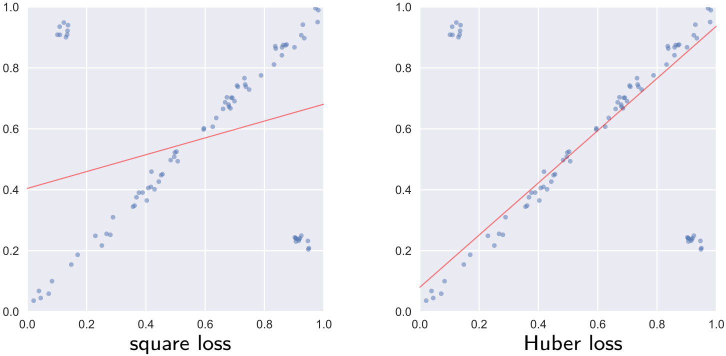
- the *Huber* penalty function is

$$p^{\text{hub}}(r) = \begin{cases} r^2 & \text{if } |y| \leq \alpha \\ \alpha(2|r| - \alpha) & \text{if } |r| > \alpha \end{cases}$$

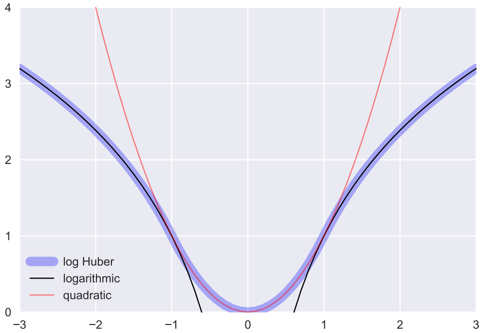
- α is a parameter
- quadratic for small r, affine for large r

Huber loss

- linear growth for large r makes fit less sensitive to outliers
- ERM with Huber loss is called a *robust* prediction method



Log Huber

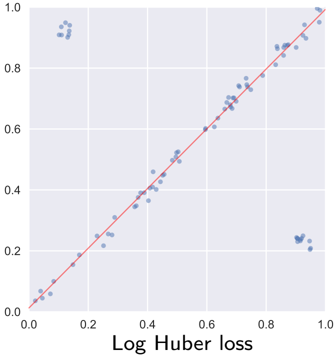


- quadratic for small y, logarithmic for large y

$$p^{\mathrm{dh}}(y) = \begin{cases} y^2 & \text{if } |y| \leq \alpha \\ \alpha^2(1 - 2\log(\alpha) + \log(y^2)) & \text{if } |y| > \alpha \end{cases}$$

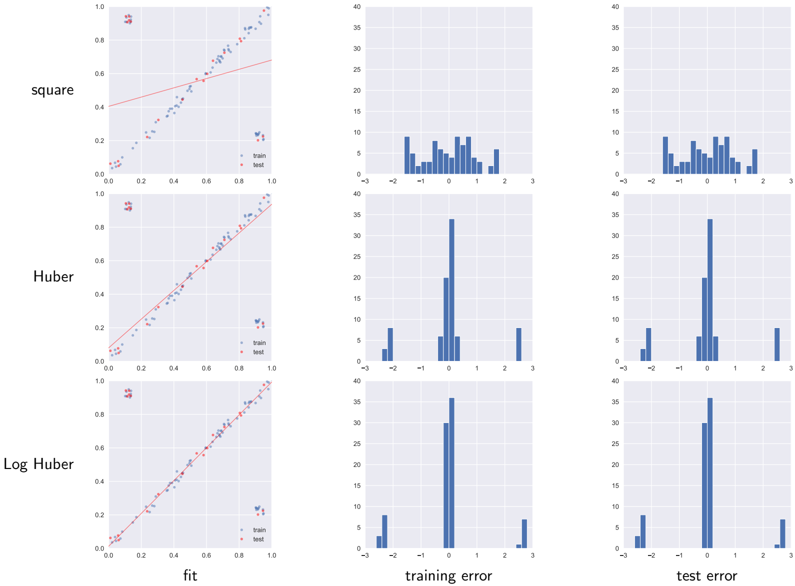
- diminishing incremental penalty at large y

Log Huber



- even less sensitive to outliers than Huber

Error distribution



Quantile regression

Absolute penalty

- absolute penalty $p^{\text{abs}}(r) = |r|$
- the best constant predictor ($d = 1$, $x_1 = 1$) minimizes $\frac{1}{n}\sum_{i=1}^{n}|\theta_1 - y^i|$
- solution is $\hat{y} = \theta_1 = \text{median}\{y^1, \ldots, y^n\}$
- (*cf.* best constant predictor with square loss, which is the average)
- rough idea:

$$\frac{d}{d\theta_1}\sum_{i=1}^{n}|\theta_1 - y^i| = \big(\text{number of } y^i\text{s} < \theta_1\big) - \big(\text{number of } y^i\text{s} > \theta_1\big)$$

- in general case, with no regularization on constant feature, median of errors is zero

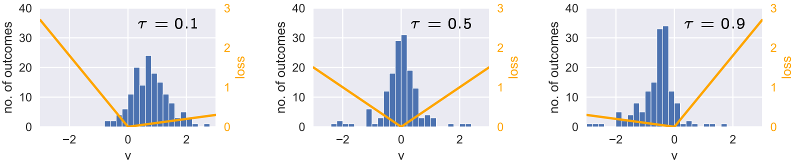
Tilted absolute penalty

- *tilted absolute penalty*: for $0 < \tau < 1$,

$$p^{\text{tlt}}(z) = \tau(z)_+ + (1-\tau)(z)_- = (1/2)|z| + (\tau - 1/2)z$$

- $\tau = 0.5$: equal penalty for over- and under-estimating
- $\tau = 0.1$: $9\times$ more penalty for under-estimating
- $\tau = 0.9$: $9\times$ more penalty for over-estimating

Tilted absolute penalty

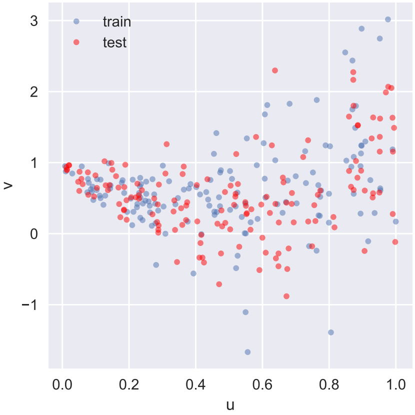


- best constant predictor for τ minimizes $\frac{1}{n}\sum_{i=1}^{n} p^{\text{tlt}}(\theta_1 - y^i)$
- fraction τ of training data satisfies $\theta_1 < y^i$
- τ-quantile of training residuals is zero
- solution is $\hat{y} = \theta_1 =$ the $(1-\tau)$-*quantile* of $\{y^1, \ldots, y^n\}$
- plots show histogram of residuals for training data

Quantile regression

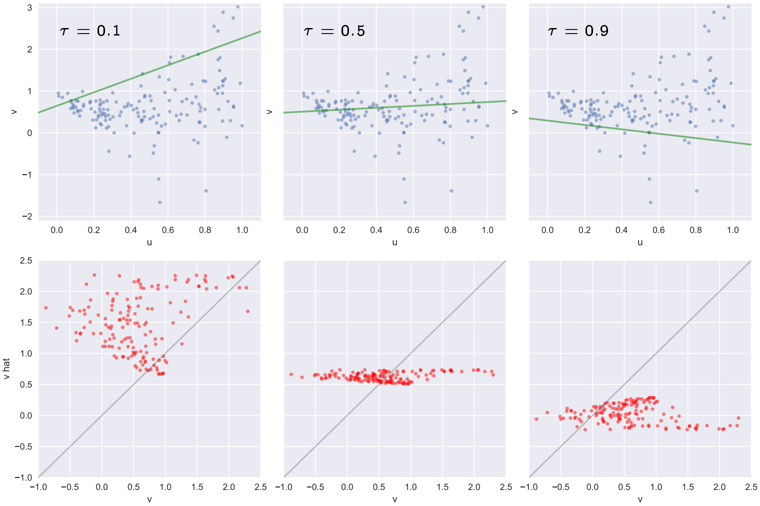
- *quantile regression* uses penalty p^{tlt}
- in general case, with no regularization on constant feature, τ-quantile of optimal errors is zero
- hence the name quantile regression

Example: Quantile regression



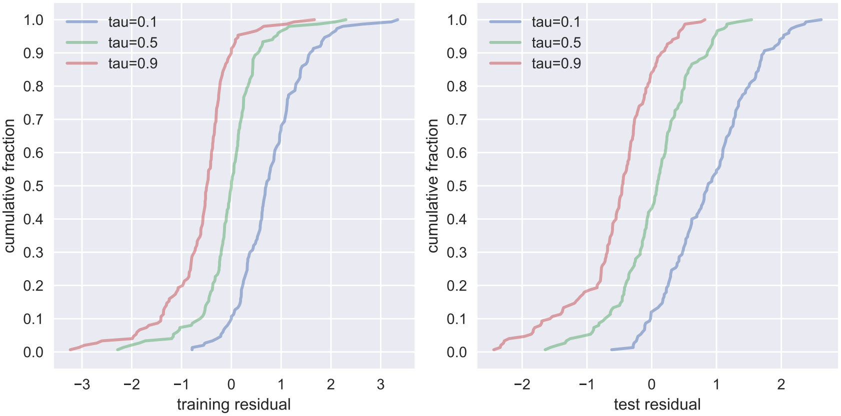
- fit training data with loss $l(\hat{y}, y) = p^{\text{tlt}}(\hat{y} - y)$
- consider τ values 0.1, 0.5, 0.9

Example: Quantile regression



- three quite different predictors

Example: Quantile regression



- τ-quantile of training residuals is zero