Penalty functions and error histograms

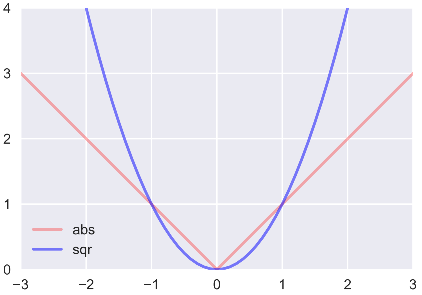
Loss and penalty functions

- empirical risk (or average loss) is $\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\hat{y}^i, y^i)$, with $\hat{y}^i = g_\theta(x^i)$
- 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 $p^{\mathsf{sqr}}(r) = r^2$ (for scalar y)
- $r = \hat{y} - y$ is the *prediction error* or *residual*
- for scalar y, $r > 0$ is *over-estimating*; $r < 0$ is *under-estimating*

Penalty functions

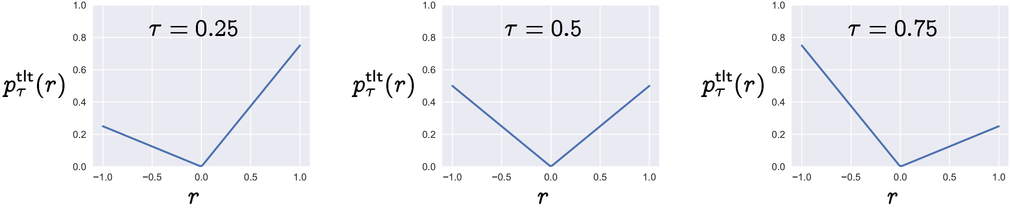
- the penalty function tells us how much we object to different values of prediction error
- usually $p(0) = 0$ and $p(r) \geq 0$ for all r
- if p is *symmetric*, *i.e.*, $p(-r) = p(r)$, we care only about the magnitude (absolute value) of prediction error
- if p is *asymmetric*, *i.e.*, $p(-r) \neq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

Square versus absolute value penalty



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

Tilted absolute penalty function



- tilted absolute penalty, with $\tau \in [0,1]$, is $p_\tau^{\mathrm{tlt}}(r) = \begin{cases} -\tau r & r < 0 \\ (1-\tau)r & r \geq 0 \end{cases}$

- for $\tau = 1/2$, same as absolute penalty (scaled by $1/2$); same penalty for under-estimating and over-estimating

- for $\tau > 1/2$, worse (higher penalty) to under-estimate than over-estimate

- for $\tau < 1/2$, worse (higher penalty) to over-estimate than under-estimate

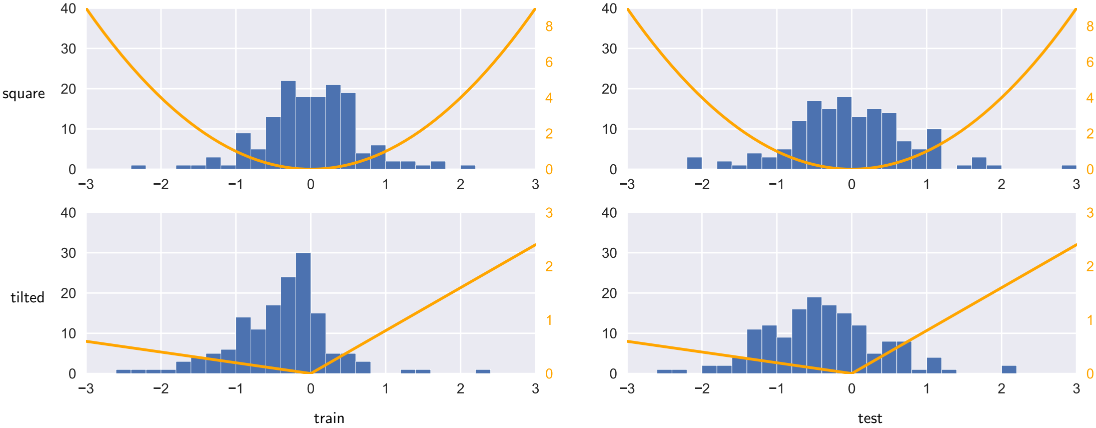
Predictors and choice of penalty function

- penalty function expresses 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 $n = 300$, $m = 1$, and $d = 31$, using 50/50 test/train split
- $r^i = \theta^\mathsf{T} x^i - y^i$, first feature is constant; 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

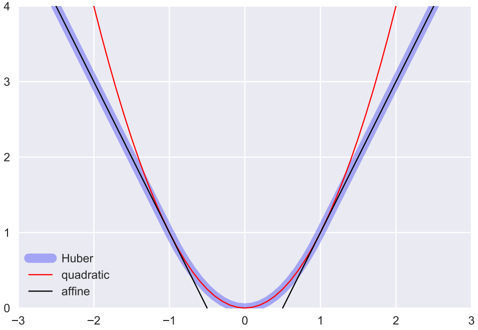
Outliers

- 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 ERM picking 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
- it's also possible to use a penalty function that is less sensitive to outlier data points

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



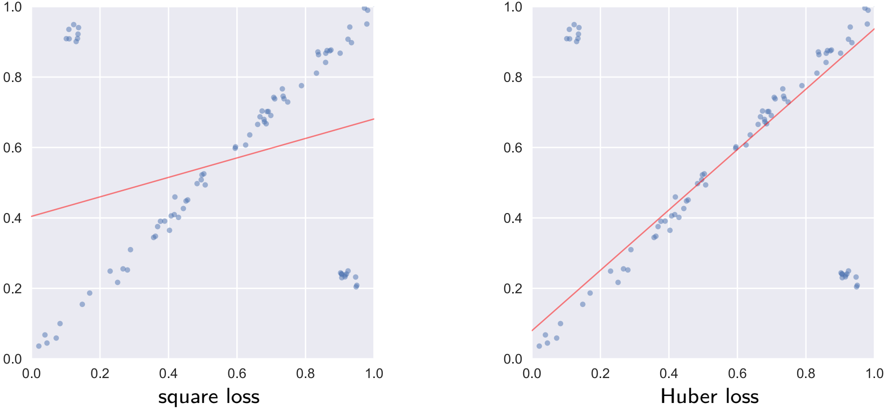
- the *Huber* penalty function is

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

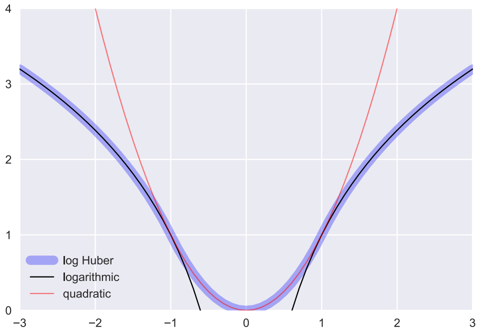
- α is a positive parameter
- quadratic for small r, affine for large r, with transition at value $r = \pm\alpha$

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

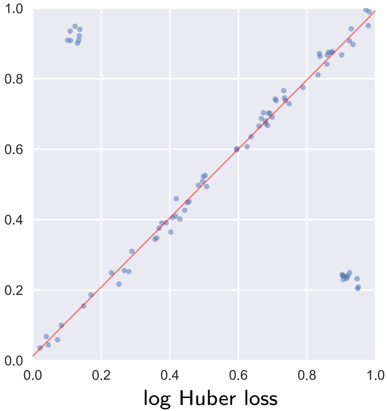


- quadratic for small y, logarithmic for large y

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

- diminishing incremental penalty at large y

Log Huber



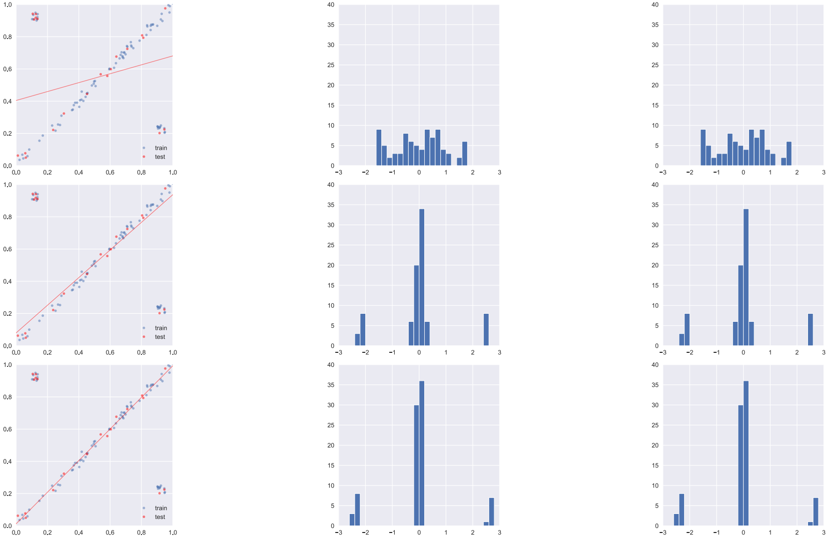
- even less sensitive to outliers than Huber

Error histogram

square

Huber

log Huber



fit

training error

test error

Quantile regression

Quantile regression

- ERM or RERM with tilted penalty p_τ^{tlt} is called *quantile regression*
- intuition:
  - $\tau > 1/2$ makes it worse to under-estimate, so predictions are 'high'
  - $\tau < 1/2$ makes it worse to over-estimate, so predictions are 'low'

Connection to quantiles

- assume the predictor has an offset (say, θ_1) that is *not* regularized
  - $g_\theta(x) = \theta_1 + \tilde{g}_\theta(x)$, where $\tilde{g}_\theta$ does not depend on θ_1 (*e.g.*, linear predictor with $x_1 = 1$)
  - regularizer $r(\theta)$ does not depend on θ_1 (*e.g.*, ridge regression with $r(\theta) = \theta_2^2 + \cdots + \theta_p^2$)
- then on the training set, with RERM predictor
  - the $(1 - \tau)$-quantile of residuals is zero
  - *i.e.*, the fraction of data for which we over-estimate ($r > 0$) is τ
- hence the name quantile regression
- if predictor generalizes, we'd expect the fraction of test data for which we over-estimate is around τ
- can create predictors for multiple τs, which gives multiple quantile estimates for a given x

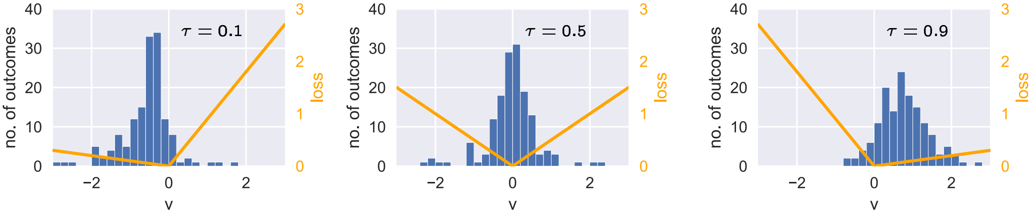
Why the $(1-\tau)$-quantile of residuals is zero

- let's fix $\theta_2, \dots, \theta_p$
- θ_1 must minimize the function $\mathcal{L}(\theta) + \lambda r(\theta)$
- $r(\theta)$ doesn't depend on θ_1, so θ_1 must minimize

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} p_\tau^{\mathrm{tlt}}(\theta_1 + \tilde{g}_\theta(x^i) - y^i))$$

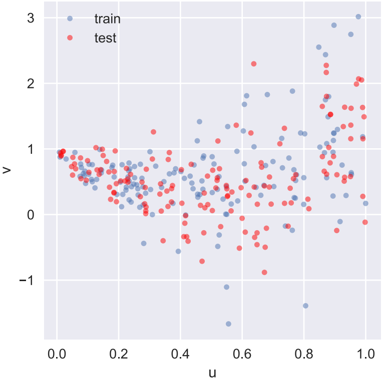
- $\tilde{g}_\theta(x)$ does not depend on θ_1, so θ_1 is the τ-quantile of $y^i - \tilde{g}_\theta(x^i)$, $i = 1, \dots, n$
- so fraction of i for which $y^i - \tilde{g}_\theta(x^i) \leq \theta_1$ is around τ
- and so, fraction of i for which $r^i = \hat{y}^i - y^i = \theta_1 + \tilde{g}_\theta(x^i) - y^i \geq 0$ is around τ
- *i.e.*, fraction of data points for which we over-estimate is around τ

Example



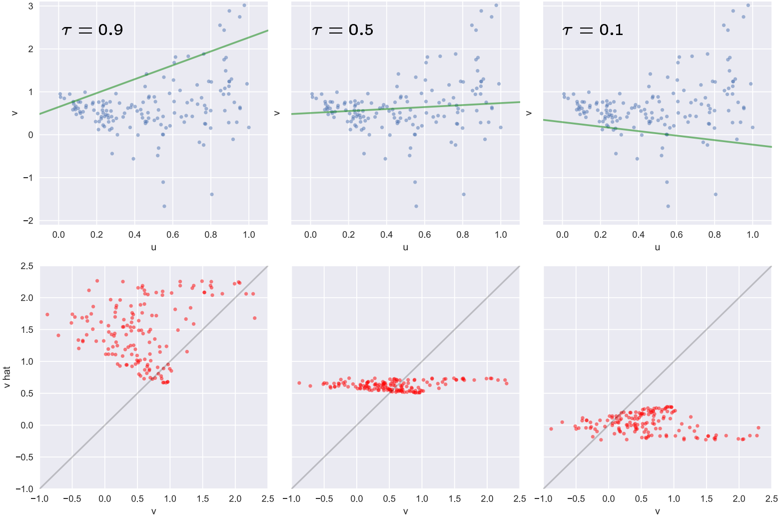
- plots show histogram of residuals training data, for $\tau = 0.1,\ 0.5,\ 0.9$

Example: Quantile straight line regression



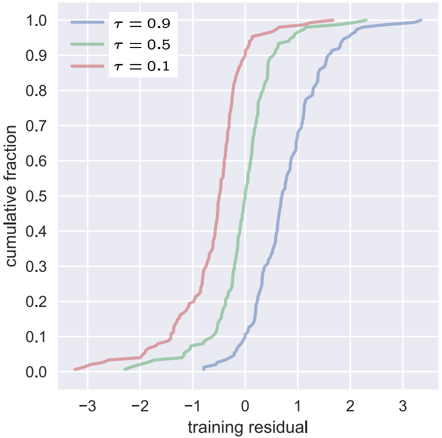
- we'll fit straight line (affine) prediction model using loss $l(\hat{y}, y) = p_\tau^{\mathrm{tlt}}(\hat{y} - y)$, $\tau = 0.1,\ 0.5,\ 0.9$

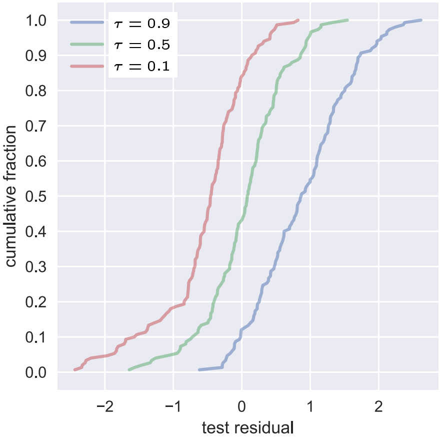
Example: Quantile straight line regression



- three quite different predictors

Example: Quantile straight line regression





Summary

Summary

- loss function is often expressed as a penalty function of the residual $r = \hat{y} - y$
- the loss function expresses how we object to different values of residual
- different choices of loss function lead to different ERM predictors
- specific applications include
  - robust fitting: fitting data with some outliers
  - quantile regression: fitting data with a specified fraction of over-estimation