Model-based boosting in R

Build your own `family`

Andreas Mayr

andreas.mayr@imbe.med.uni-erlangen.de

Institut für Medizininformatik, Biometrie und Epidemiologie (IMBE)

Friedrich-Alexander-Universität Erlangen-Nürnberg

Statistical Computing 2011

Build your own family

```
Family(ngradient, loss = NULL, risk = NULL,
       offset = function(y, w)
            optimize(risk, interval = range(y), y=y, w=w)$minimum, ...)
```

- `Family()`: can be used to create a new `family`
- `ngradient`: a function with arguments `y`, `f` and `w` (weights) implementing the **negative** gradient of the `loss` function
- `loss`: function to be minimized with arguments `y` and `f`
- `risk`: By default, the (weighted) sum of the `loss` function
- `offset`: starting value for the algorithm: $\hat{f}^{[0]}$

Example: quantile regression

Quantile regression

What's our loss?

- Quantile regression, in comparison to standard regression, fits quantiles rather than the expected mean of the conditional distribution function

$$y_i = \hat{f}_{\tau i} + \varepsilon_{\tau i}$$

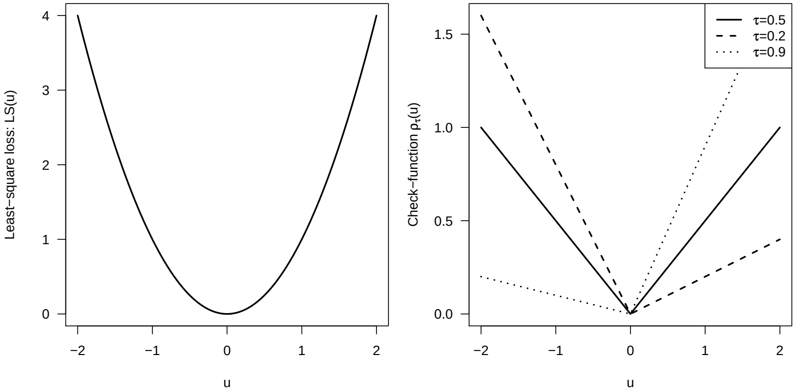
- The appropriate loss function for the τ-quantile is the *check-function* $\rho_\tau(\cdot)$:

$$\rho_\tau(y_i - \hat{f}_{\tau i}) = \begin{cases} (y_i - \hat{f}_{\tau i}) \cdot \tau & (y_i - \hat{f}_{\tau i}) \geq 0 \\ (y_i - \hat{f}_{\tau i}) \cdot (\tau - 1) & (y_i - \hat{f}_{\tau i}) < 0. \end{cases}$$

```
loss = function(y, f) tau *(y - f) * ((y - f) >= 0) +
                      (tau - 1) * (y - f) * ((y - f) < 0)
```

Quantile regression

What's our loss?



Loss function for standard regression (left) and the check function for quantile regression (right) for different values of τ.

Quantile regression

What's the gradient?

- The *check-function* is not differentiable at the point 0. But in practice, as the response is continuous, we can ignore this by defining:

$$-\frac{\partial \rho_\tau(y_i, f_{\tau i})}{\partial f} = \begin{cases} \tau & (y_i - \hat{f}_{\tau i}) \geq 0 \\ \tau - 1 & (y_i - \hat{f}_{\tau i}) < 0. \end{cases}$$

```
ngradient = function(y, f, w = 1)
                tau * ((y - f) >=0) + (tau- 1) * ((y - f) < 0)
```

Quantile regression

Construct our own `family`

```
R> OurQuantReg <- function(tau = 0.5){
+   Family(
+    ngradient = function(y,f,w = 1)
+                    tau * ((y - f) >=0) +  (tau - 1) * ((y - f) < 0),
+    loss = function(y, f) tau *(y - f) * ((y - f) >= 0) +
+               (tau - 1) * (y - f) * ((y - f) < 0) ,
+    offset = function(y, w = 1) median(y),
+    name = "Our new family for quantile regression"
+    )}
R> OurQuantReg()
```

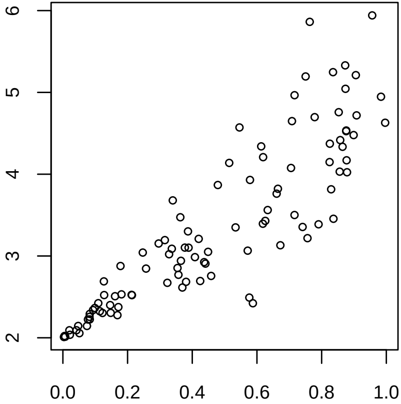
```
         Our new family for quantile regression

Loss function: tau * (y - f) * ((y - f) >= 0) + (tau - 1) * (y - f)
* ((y - f) < 0)
```

Let's try how it works

Our new family for quantile regression

Let's try how it works

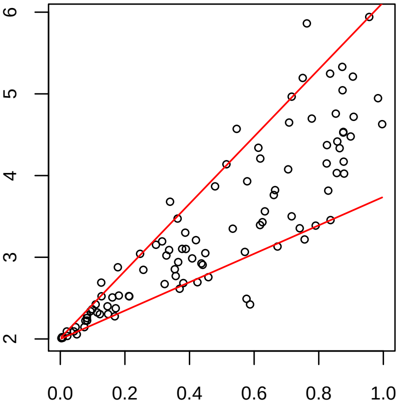
```
R> set.seed(0804)
R> x <- runif(100)
R> y <- 2 + 3*x + x*rnorm(100)
R> plot(x,y)
```



Our new family for quantile regression

Let's try how it works

```
R> ctrl <- boost_control(mstop=500)
R> gb1 <- glmboost(y~x, family=OurQuantReg(tau=0.1), control=ctrl)
R> gb2 <- glmboost(y~x, family=OurQuantReg(tau=0.9), control=ctrl)
R> lines(x, fitted(gb1), col=2)
R> lines(x, fitted(gb2), col=2)
```



Our new `family` for quantile regression

Let's try how it works

Compare to `rq()` of Koenker's `quantreg` package, which can be seen as the gold standard for low-dimensional quantile regression, based on linear programming:

```
R> library(quantreg)
R> rq1 <- rq(y~x, tau=0.1)
R> rq2 <- rq(y~x, tau=0.9)
R> rbind(coef(rq1), coef(rq2))
```

```
     (Intercept)        x
[1,]    1.998975 1.740673
[2,]    2.009817 4.111221
```

```
R> rbind(coef(gb1, off2int=TRUE), coef(gb2, off2int=TRUE))
```

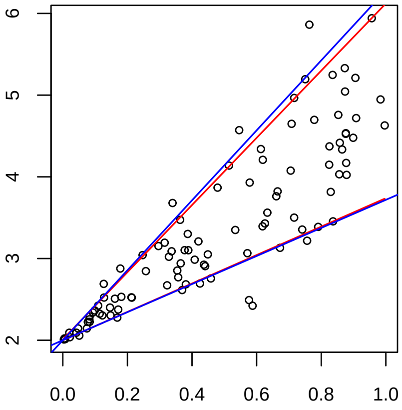
```
     (Intercept)        x
[1,]    1.998600 1.740263
[2,]    2.010295 4.111727
```

Our new `family` for quantile regression

Let's try how it works

Compare to the true coefficients:

```
R> abline(a=2, b= 3 + qnorm(0.1), col=4)
R> abline(a=2, b= 3 + qnorm(0.9), col=4)
```



For more on quantile regression

Family `QuantReg()` is implemented in **mboost** (Fenske *et al.*, 2011).

- Koenker, R. (2005). *Quantile Regression*. New York: Cambridge University Press.
- Fenske, N., Kneib, T. and Hothorn, T. (2011). Identifying risk factors for severe childhood malnutrition by boosting additive quantile regression. *Journal of the American Statistical Association*, to appear.