

glmnet

Alina

25 11 2020

```
setwd("~/fallstudien_2_projekt_1/datasets")
data <- read.csv("dataset_ul.csv", header = TRUE, sep = ",")
library(glmnet)
```

Data

read the table and select the covariates for the model

```
data <- subset(data, select = c(scenario,
                                provider,
                                velocity_mps,
                                acceleration_mpss,
                                rsrp_dbm,
                                rsrq_db,
                                rssnr_db,
                                cqi,
                                ta,
                                payload_mb,
                                f_mhz,
                                throughput_mbits))
```

now eliminate the rows with NA's

```
data <- data[complete.cases(data),]
```

separate the full dataset in train and test data, hereby 75% of the data get to be the training set and the rest will be the test set

```
#set.seed(101)
sample <- sample.int(n = nrow(data), size = floor(.75*nrow(data)), replace = F)
train <- data[sample, ]
test <- data[-sample, ]
```

in our case there are two variables that do not contain integers but factors, therefore we have to encode them, so that the model can handle them here one-hot encoding is used

```
X <- makeX(train, test = test)

train <- X[["x"]]
test <- X[["xtest"]]
```

Modelfitting

fit the glmnet model with cross validation for the penalty parameter lambda the parameter alpha for the elastic net model has to be set by user

```
#fit <- glmnet(subset(train, select = -throughput_mbits),  
#             subset(train, select = throughput_mbits))  
  
fit.cv <- cv.glmnet(subset(train, select = -throughput_mbits),  
                   subset(train, select = throughput_mbits),  
                   type.measure = "mae", nfolds = 20, alpha = 1)
```

Prediction

from the fitted cv.glmnet model we now generate the predictions with the covariates from the test set, we hereby use the lambda parameter that generates the lowest error in the cv process

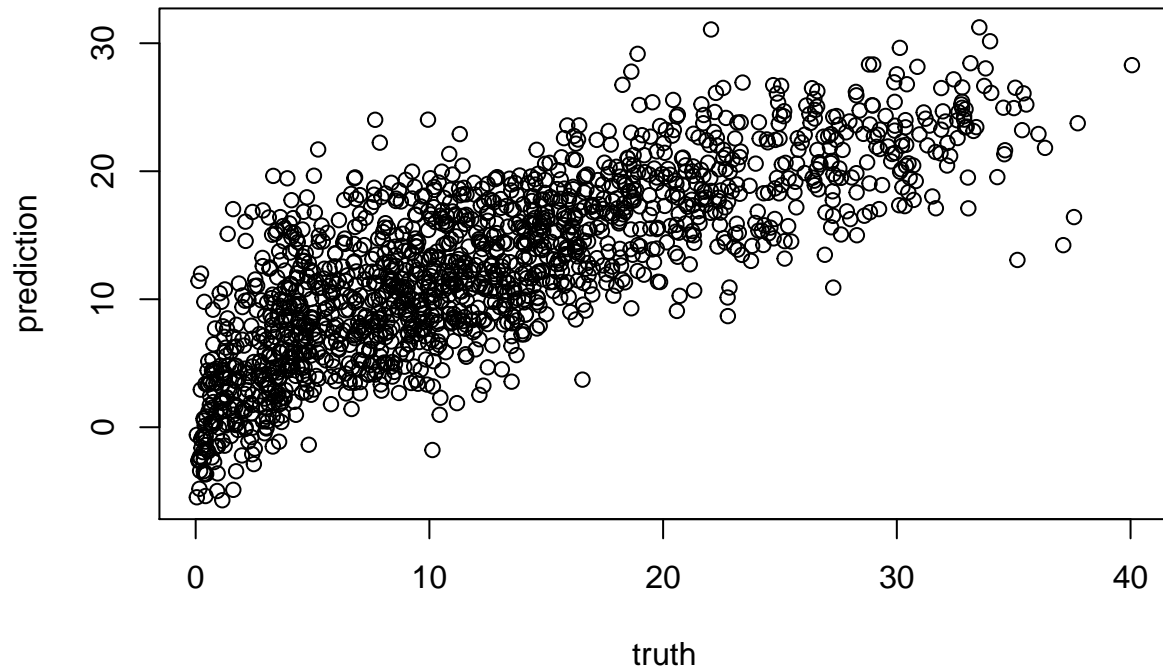
```
#pred <- predict.glmnet(object = fit, newx = subset(test, select = -throughput_mbits),  
#                      s = 0, type = "response")  
  
pred.cv <- predict(object = fit.cv, newx = subset(test, select = -throughput_mbits),  
                  s = "lambda.min", type = "response")
```

Results

plot the predictions from the cv glmnet model against the truth values from our test set

```
#plot(fit.cv)  
#plot(subset(test, select = throughput_mbits), pred, main = "GLMNET",  
#     xlab = "truth", ylab = "prediction")  
  
plot(subset(test, select = throughput_mbits), pred.cv, main = "CV.GLMNET",  
     xlab = "truth", ylab = "prediction")
```

CV.GLMNET



we want to know the model rating, therefore we calculate the R-squared, MSE and MAE

```
#calculate R-Squared R2
yq <- mean(subset(test, select = throughput_mbits))
R2 <- sum((pred.cv - yq)2) / sum((subset(test, select = throughput_mbits) - yq)2)
R2
```

```
## [1] 0.6350165
```

```
#calculate MSE
n <- 1 / length(pred.cv)
mse <- n * sum((subset(test, select = throughput_mbits) - pred.cv)2)
mse
```

```
## [1] 29.79635
```

```
#calculate MAE
mae <- n * sum(abs(subset(test, select = throughput_mbits) - pred.cv))
mae
```

```
## [1] 4.246514
```

Feature Importance

compare the absolute coefficients of the model, the larger the value the more information does the corresponding covariate bring