Regularisierungsmethoden

Lasso und Ridge Regression

Jens Hüsers

24 10 2017

Wozu werden Modelle genutzt

Erkennen von Zusammenhängen

- Das Aufzeigen und Erkennen von Zusammenhängen, die in der (komplexen) Realität existieren.
- Zusammenfassen dieser Zusammenhänge mittels Vereinfachungen.
- Diese Vereinfachungen werden dann auch als Modelle bezeichnet.

  Alle Modelle sind falsch, aber einige von Ihenen sind nützlich. (George Box)

Einsatz von Modell in der Forschung:

Beispielhafte **Forschungsfrage** aus der Forschung: Welche Merkmale (Variablen) stehen in Zusammenhang mit erhöhtem systolischen Blutdruck?

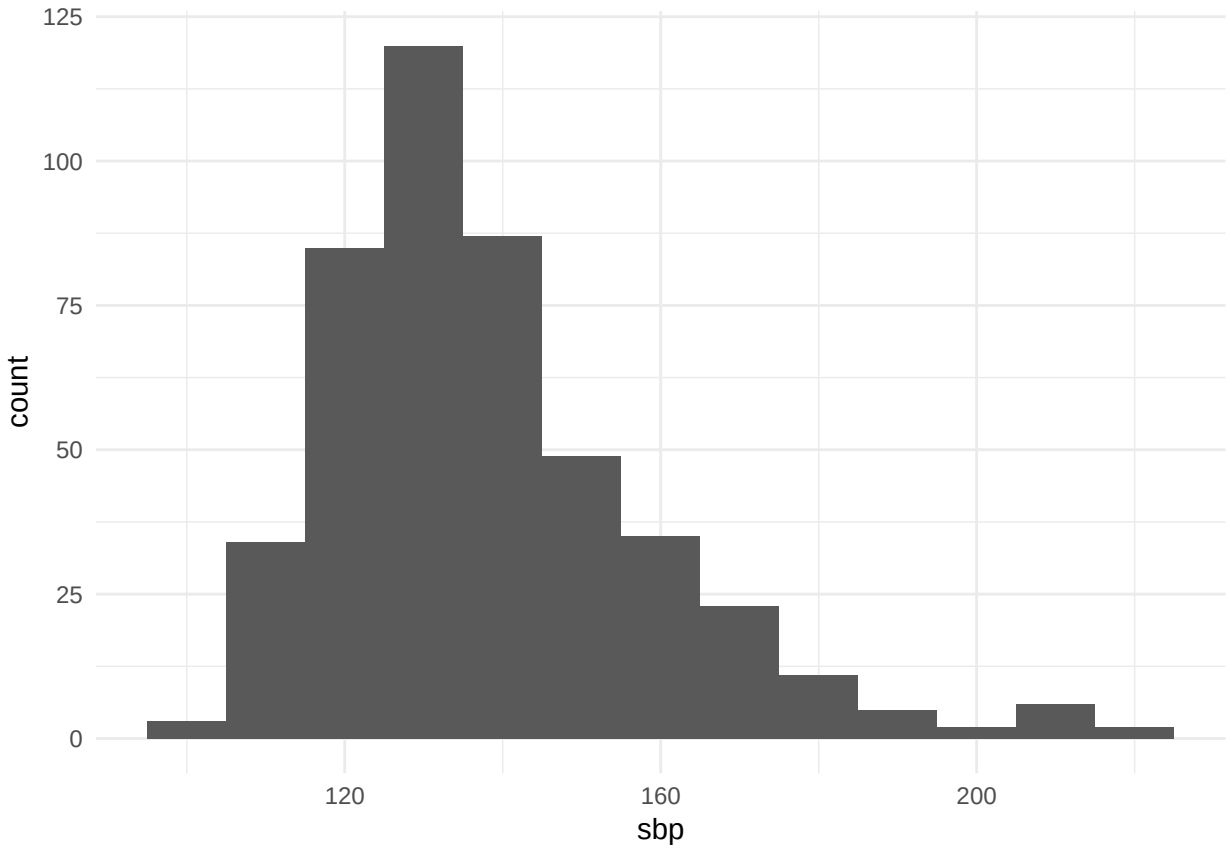
Beispielmodell anhand des `SAheart`- Datensatzes von Rousseauq et al. (1983), welches im `ElemStatLearn` Paket enthalten ist.

Modellierung der Zusammenhänge zwischen dem systolischem Blutdruck und körperlichen Merkmalen wie Alter, Tabakkonsum und Alkoholkonsum.

```
library(ISLR)
library(ElemStatLearn)
library(broom)
library(tidyverse)

# preprocessing
# scaling predictors
heart <- SAheart %>%
  mutate(famhist = if_else(famhist == "Present", 1, 0)) %>%
  mutate_at(.vars = vars(-sbp), scale)

# hist of sbp
ggplot(data = heart, aes(x = sbp)) +
  geom_histogram(binwidth = 10) +
  theme_minimal()
```




```
# modelling
fit <- lm(sbp ~ ., data = heart)

# model summary
summary_fit <- tidy(fit) %>% mutate(sig = emo::ji_p(p.value))

summary_fit %>%
  kable(caption = "Zusammenfassung des Regressionsmodells", digits = 3)
```

Table 1: Zusammenfassung des Regressionsmodells

term	estimate	std.error	statistic	p.value	sig
(Intercept)	138.327	0.868	159.356	0.000	
tobacco	0.391	1.003	0.390	0.697	
ldl	-0.173	0.991	-0.174	0.862	
adiposity	2.479	1.643	1.509	0.132	
famhist	-0.595	0.921	-0.646	0.519	
typea	-0.806	0.895	-0.901	0.368	
obesity	1.480	1.317	1.124	0.261	
alcohol	1.931	0.895	2.157	0.032	
age	5.258	1.281	4.104	0.000	
chd	1.256	0.992	1.265	0.206	

The subset selection methods described in Section 6.1 involve using least squares to fit a linear model that contains a subset of the predictors. As an alternative, we can fit a model containing all p predictors using a technique that *constrains* or *regularizes* the coefficient estimates, or equivalently, that *shrinks* the coefficient estimates towards zero. It may not be immediately

Figure 1: Selektion

Vorhersagen treffen

Treffen von Vorhersagen bezüglich des Systolischen Blutdrucks auf Basis des erstellten Modells.

```
# Root Mean Squared Error
rmse <- augment(fit) %>%
  map_df(as.numeric) %>%
  summarise(rmse = mean(.resid^2) %>% sqrt) %>%
  pull(rmse)
```

Der mittlere Fehler beträgt 18.45.

Welche Probleme löst Regularisierung?

- *Overfitting:* Schlechte Generalisierbarkeit des Modells (Testfehler des Modells hoch).
  - Alternativen: Kreuzvaldieren des Modells

In section 5.2.2, we defined regularization as "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error." There are many regularization strategies. Some put extra

aus(Goodfellow, Bengio, and Courville 2016)

- Variablenselektion und Dedektion von *Multikolliniearität* zwischen den Variablen
  - Alternativen: Schrittweise Regression, Best Subset Regression (ausprobieren aller möglichen Prädiktorkombinationen, 2^p Modelle)

aus (James et al. 2013)

Regulierungsmethoden auch Shrinkage genannt, kontrollieren sowohl Overfitting als auch die Variablenselektion

Methode: Anpassen der Loss- (Objective-, Cost-) Function.

In diesem Fall verwendeten wir die Methode der kleinsten Quadrate (Least Squares, RSS), welche die Summe der Fehler (Residuen, Risidual Sum of Squares) minimiert:

$$\mathrm{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

Bei der Regulierung wird diese Loss Funktion angepasst:

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \mathrm{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2,$$

wobei $\lambda \sum_{j=i}^{p} \beta_j^2$ der Regularisierungsterm ist und λ ein Hyper- resp. Tuningparameter, der separat (via Kreuzvalidierung) bestimmt werden muss.

Dabei gilt: Je größer λ desto größer der Regularisierungsterm und desto stärker werden die Regressionsgewichte in Richtung null verkleinert.

Der Regularisierungsterm kann verschiedene Formen annehmen. In diesem Fall ist der Regularisierungsterm die $\ell 2$ *norm*, definiert als:

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} {\beta_j}^2}$$

Dabei handelt es sich um die euklidische Norm des Vektors, welcher die Regressionsgewichte enthält. Wird dieser Regulierungsterm eingesetzt, spricht man auch von **Ridge-Regression**.

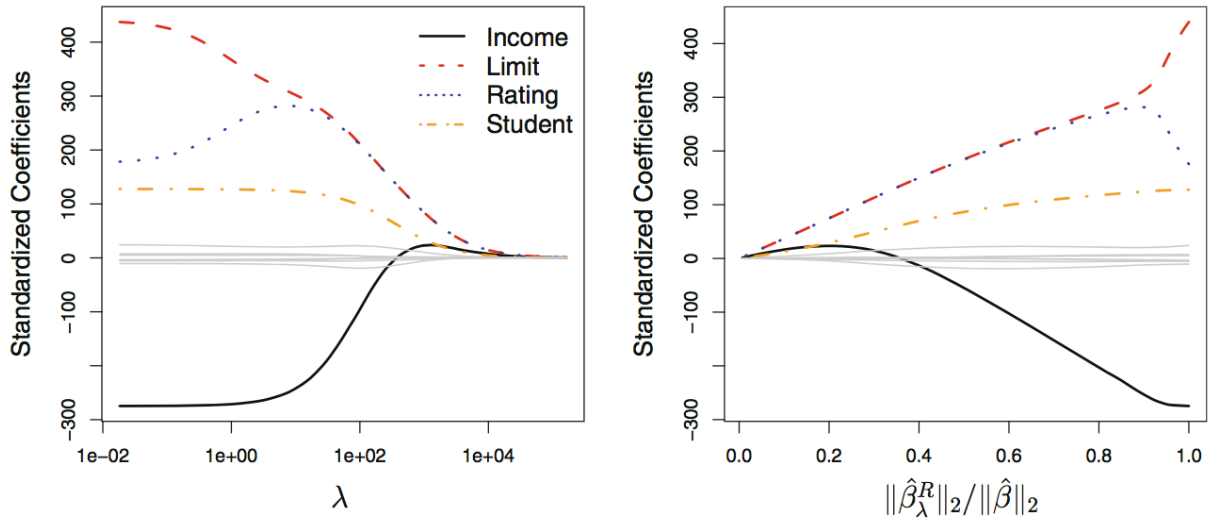


FIGURE 6.4. *The standardized ridge regression coefficients are displayed for the* `Credit` *data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$.

aus (James et al. 2013)

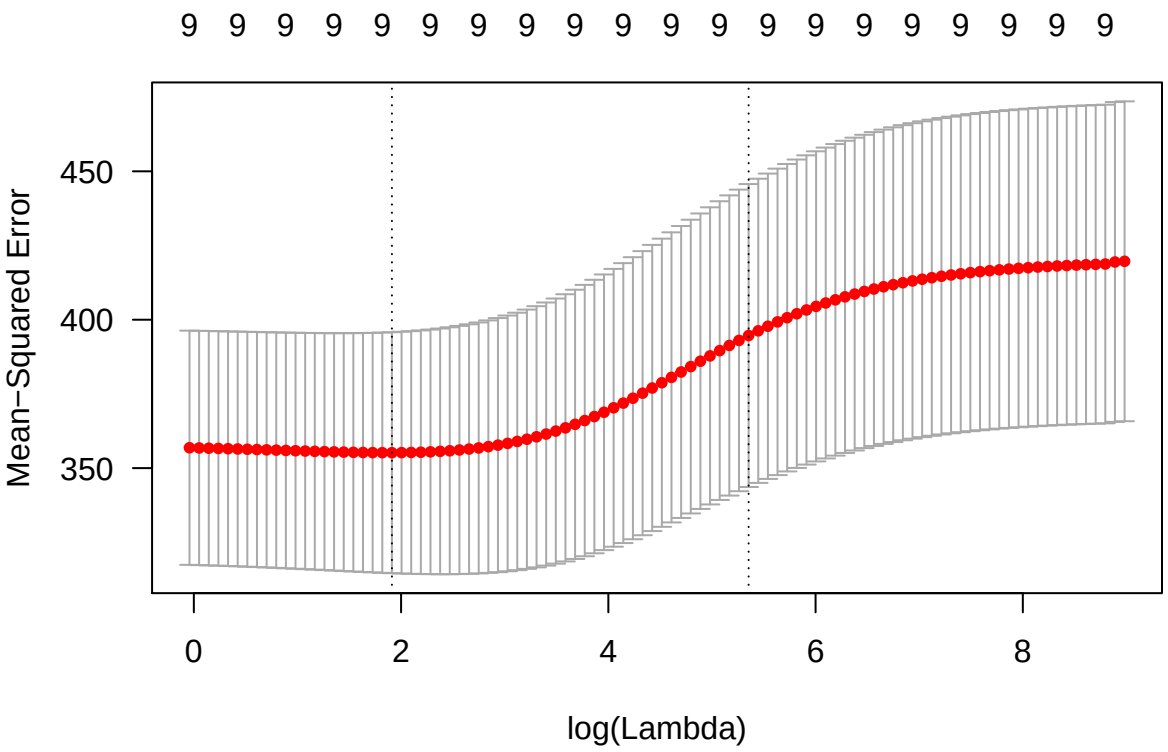
Beispiel: Generalisierbarkeit

Ich setzte die Ridge Regression (L2 Norm) zur Regulierung unseres Regressionsmodell ein. Dazu nutzte ich das `glmnet` Paket.

```
# fit with glmnet
library(glmnet)

# model matrix
# glmnet does not take dataframes or formulas
x <- model.matrix(sbp ~ -1  + ., data = heart)
y <- heart$sbp

# modelling with cross validation
cv_ridge_fit <- cv.glmnet(x, y, alpha = 0, nfolds = 5, type.measure = "mse")
plot(cv_ridge_fit, las = 1)
```




```
# get beta coefs of final model
ridge_betas <- coef(cv_ridge_fit, s = "lambda.min") %>% data.matrix()

# predict
pred_ridge_sbp <- predict(cv_ridge_fit, newx = x, s = "lambda.min")
rmse_ridge <- sqrt(mean((y - pred_ridge_sbp)^2))

# comparison of beta coefs
summary_fit  %>%
  select(term, coef_model = estimate) %>%
  full_join(tibble(term = rownames(ridge_betas), coef_ridge  = ridge_betas[, 1]), by = "term") %>%
  mutate(delta = coef_ridge - coef_model) %>%
  arrange(delta) %>%
  kable(digits = 2)
```

term	coef_model	coef_ridge	delta
age	5.26	3.77	-1.49
alcohol	1.93	1.49	-0.44
obesity	1.48	1.31	-0.17
chd	1.26	1.14	-0.12
adiposity	2.48	2.44	-0.04
(Intercept)	138.33	138.33	0.00
typea	-0.81	-0.71	0.10
ldl	-0.17	0.18	0.35
famhist	-0.59	-0.18	0.41
tobacco	0.39	0.85	0.45

Der Mittlere Fehler des regulierten Modells beträgt 18.51 und der Mittlere Fehler des vollen Modells beträgt 18.45.

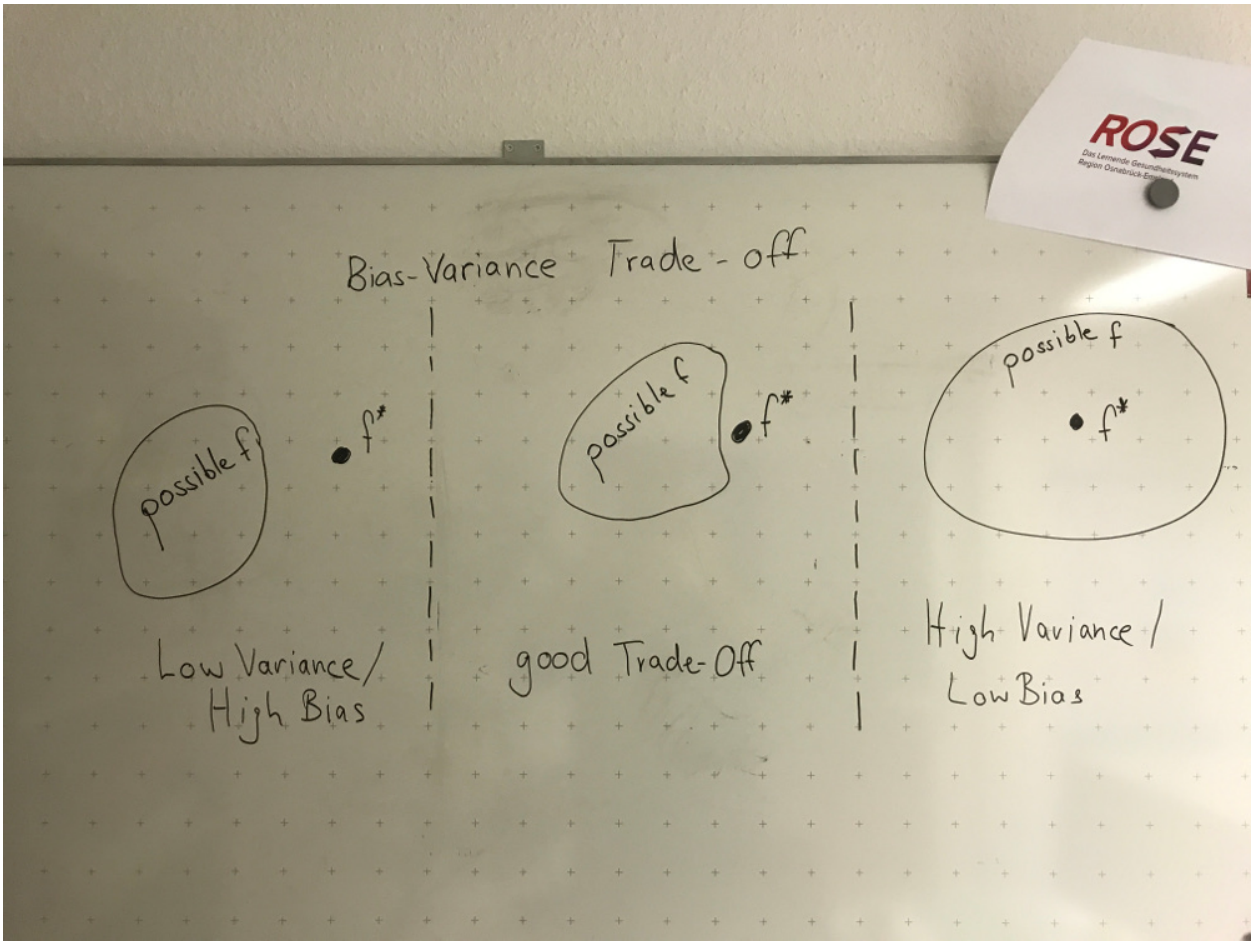


Figure 2: Darstellung des Bias Variance Tradeoff (Visualisierung Übernommen von Hugo Larochelle)

Exkurs Significance Tests for Regularized Models

Why is it inadvisable to get statistical summary information for regression coefficients from glmnet model?

A SIGNIFICANCE TEST FOR THE LASSO,

Intuition der Regularization: Bias-Variance Trade-Off

- Modelle besitzen eine *Varianz*: Wie stark ändert sich das Modell, wenn sich die Trainingsdaten (Stichprobe) ändern?
- Modelle haben einen *Bias*: Wie nah ist das durchschnittliche Model am tatsächlichen Sachverhalt?

Der Fehler eines Modells setzt sich zusammen aus der Summe des quadrierten Bias und der Varianz

---

aus (Friedman, Hastie, and Tibshirani 2001)

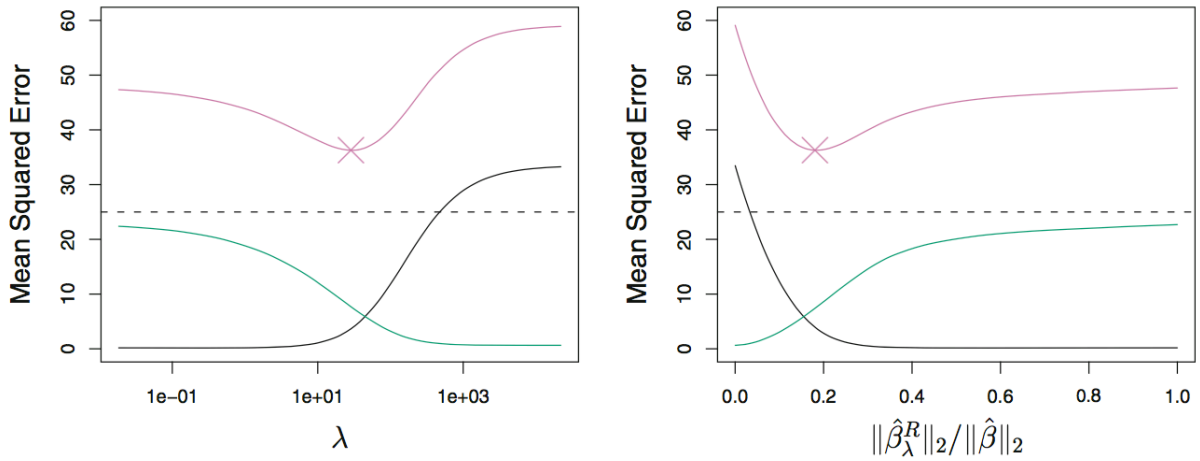


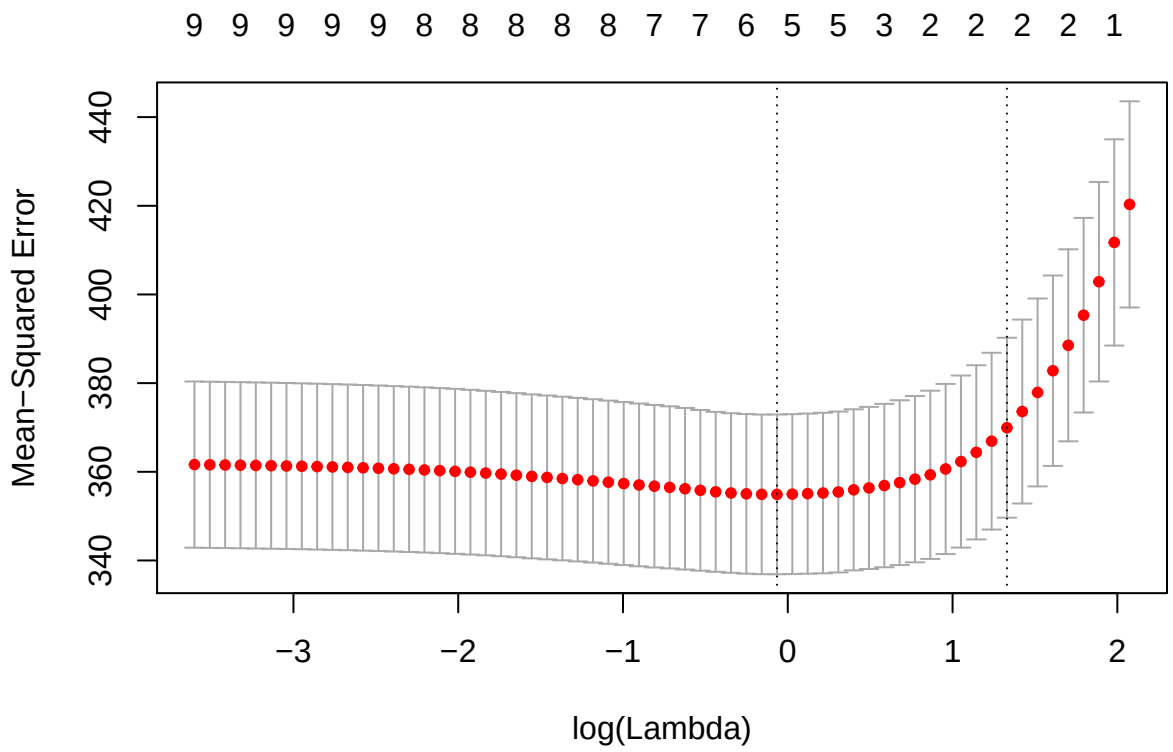
FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$*. The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

Figure 3:

Beispiel: Prädiktorselektion (Lasso Regression)

Bei der Ridge Regression werden die Koeffizienten gen null geschrumpft. Jedoch werden die Koeffizienten nicht genau null. Damit die Koeffizienten genau null werden können, wird die Lasso Regression eingesetzt. Die einzige Änderung ist der Penalty Term. Anstatt der L2 Norm wird die L1 Norm eingesetzt. Die L1 Norm ($\ell 2$ Norm, Summennorm) des Koeffizientenvektors ist definiert als $||\beta||_1 = \sum |\beta_j|$

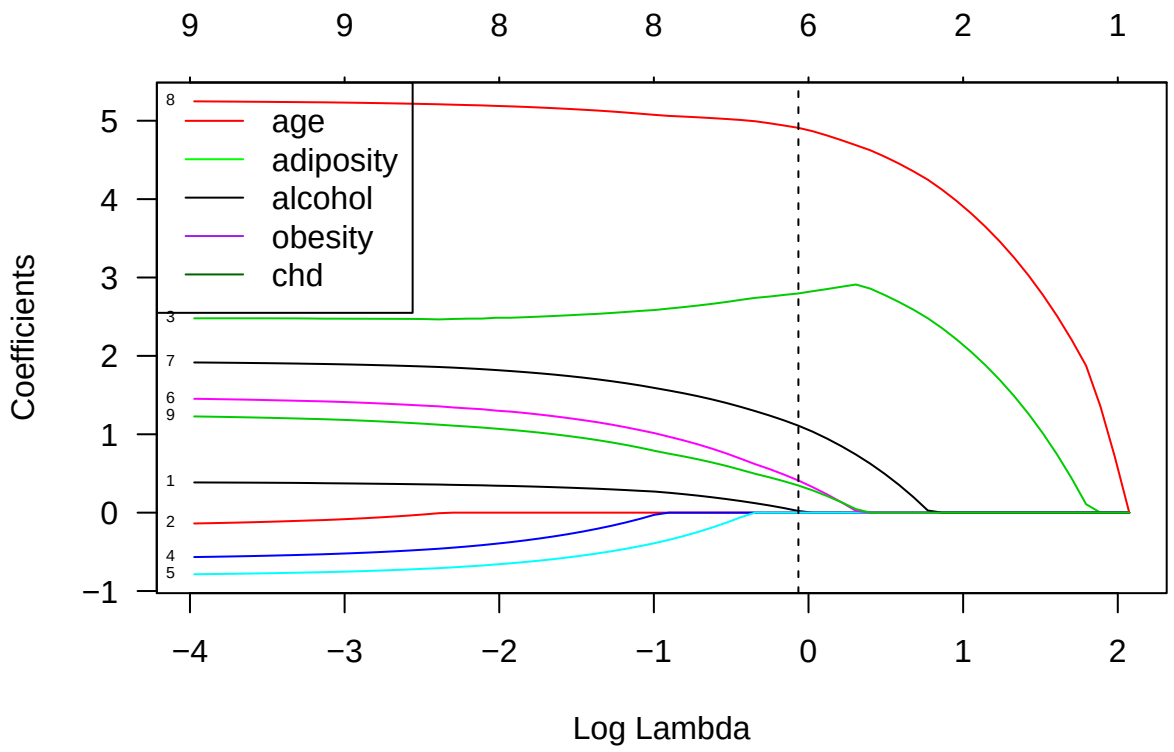
```
# modelling with cross validation
cv_lasso_fit <- cv.glmnet(x = x, y = y, type.measure = "mse", nfolds = 5, alpha = 1)
plot(cv_lasso_fit)
```




```r
# coef(cv_lasso_fit, s = "lambda.min", las = 1)
# predict(cv_lasso_fit, newx = x, s = "lambda.min")

# get beta coefs of final model
lasso_betas <- coef(cv_lasso_fit, s = "lambda.min") %>% data.matrix()

lasso_fit <- glmnet(x = x, y = y, alpha = 1)
plot(lasso_fit, label = TRUE, las = 1, xvar = "lambda")
leg_names <- colnames(x)[c(8, 3, 7, 6, 9)]
legend("topleft", legend = leg_names, col = c("red", "green", "black", "purple", "darkgreen"), lty = 1)
abline(v = log(cv_lasso_fit$lambda.min), lty = 2)
```




```
# predict
pred_lasso_sbp <- predict(cv_lasso_fit, newx = x, s = "lambda.min")
rmse_lasso <- sqrt(mean((y - pred_lasso_sbp)^2))

# comparison of beta coefs
summary_fit  %>%
  select(term, coef_model = estimate) %>%
  full_join(tibble(term = rownames(ridge_betas), coef_ridge  = ridge_betas[, 1]), by = "term") %>%
  full_join(tibble(term = rownames(lasso_betas), lasso_ridge  = lasso_betas[, 1]), by = "term") %>%
  kable(digits = 2)
```

term	coef_model	coef_ridge	lasso_ridge
(Intercept)	138.33	138.33	138.33
tobacco	0.39	0.85	0.02
ldl	-0.17	0.18	0.00
adiposity	2.48	2.44	2.80
famhist	-0.59	-0.18	0.00
typea	-0.81	-0.71	0.00
obesity	1.48	1.31	0.41
alcohol	1.93	1.49	1.11
age	5.26	3.77	4.91
chd	1.26	1.14	0.35

References

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics Springer, Berlin. http://statweb.stanford.edu/~tibs/book/preface.ps.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 103. Springer Texts in Statistics. New York, NY: Springer New York. http://link.springer.com/10.1007/978-1-4614-7138-7.