MA8701 Advanced methods in statistical inference and learning

L4: Statistical inference

Mette Langaas IMF/NTNU

31 January, 2021

Contents

Shrinkage - third act

Outline

- South African heart disease: classification with two classes, understand print-out from glm and glmnet, and inference with CI and p-values
- Prediction vs statistics inference: what are the aims? Sampling distributions
- Bayesian lasso
- Boostrapping
- Sample splitting
- Inference after selection (Forward regression example, Polyhedral result, PoSI)
- Reproducibility crisis and selective inference

Literature L4

Main source:

- [HTW] Hastie, Tibshirani, Wainwrigh: "Statistical Learning with Sparsity: The Lasso and Generalizations". CRC press. Ebook. Chapter 6.0, 6.1, 6.2, 6.5. (Mention some results from 6.3 and 6.4.)

Supplemental sources:

- Short note on multiple hypothesis testing in TMA4267 Linear Statistical Models, Kari K. Halle, Øyvind Bakke and Mette Langaas, March 15, 2017.

- Single/multi-sampling splitting part of Dezeure, Bühlmann, Meier, Meinshausen (2015). "High-Dimensional Inference: Confidence Intervals, p-Values and R-Software hdi". Statistical Science, 2015, Vol. 30, No. 4, 533–558 DOI: 10.1214/15-STS527 (only the single/multiple sample splitting part in 2.1.1 and 2.2 for linear regression, and using the method in practice).

- Taylor and Tibshirani (2015): Statistical learning and selective inference, PNAS, vol 112, no 25, pages 7629-7634. (Soft version of HTW 6.3.2)

South African heart disease

We start by discussing the data analysis included in the class material from L3, but not discussed in class before.

L3 South African heart disease

Group discussion:

- What is done?
- What are the results?
- Where are the confidence intervals and p-values in the ridge and lasso print-out?

Confidence interval

Set-up

- We have a random sample $Y_1, Y_2, \ldots, Y_N$ from
- some distribution F with some (unkonwn) parameter θ.
- Let $y_1, y_2, \ldots, y_N$ be the observed values for the random sample.

Statistics

- We have two statistics $\hat{\theta}_L(Y_1, Y_2, \ldots, Y_N)$ and $\hat{\theta}_U(Y_1, Y_2, \ldots, Y_N)$ so that

$$P(\hat{\theta}_L(Y_1, Y_2, \ldots, Y_N) \leq \theta \leq \hat{\theta}_U(Y_1, Y_2, \ldots, Y_N)) = 1 - \alpha$$

where $\alpha \in [0, 1]$

Confidence interval

The numerical interval

$$[\hat{\theta}_L(y_1, y_2, \ldots, y_N), \hat{\theta}_U(y_1, y_2, \ldots, y_N)]$$

is called a $(1 - \alpha)$ 100% confidence interval.

Single hypothesis test

$$H_0 \colon \beta_j = 0 \quad \text{vs.} \quad H_1 \colon \beta_j \neq 0$$

	Not reject H_0	Reject H_0
H_0 true	Correct	Type I error
H_0 false	Type II error	Correct

- Two types of errors are possible, type I error and type II error.
- A type I error would be to reject H_0 when H_0 is true, that is concluding that there is a linear association between the response and the predictor where there is no such association. This is called a *false positive finding.*
- A type II error would be to fail to reject H_0 when the alternative hypothesis H_1 is true, that is not detecting that there is a linear association between the response and the covariate. This is called a *false negative finding.*

---

p-value

- A p-value $p(X)$ is a test statistic satisfying $0 \leq p(\mathbf{Y}) \leq 1$ for every vector of observations $\mathbf{Y}$.
- Small values give evidence that H_1 is true.
- In single hypothesis testing, if the p-value is less than the chosen significance level (chosen upper limit for the probability of committing a type I error), then we reject the null hypothesis, H_0.
- The chosen significance level is often referred to as α.

A p-value is *valid* if

$$P(p(\mathbf{Y}) \leq \alpha) \leq \alpha$$

for all α, $0 \leq \alpha \leq 1$, whenever H_0 is true, that is, if the p-value is valid, rejection on the basis of the p-value ensures that the probability of type I error does not exceed α.

An *exact* p-value satisfies

$$P(p(\mathbf{Y}) \leq \alpha) = \alpha$$

for all α, $0 \leq \alpha \leq 1$.

- The exact p-value is uniformly distributed when the null hypothesis is true.
- This is a fact that is often misunderstood by users of p-values.
- The incorrect urban myth is that p-values from true null hypotheses are close to one, when the correct fact is that all values in intervals of the same length are equally probable (which is a property of the uniform distribution).

Statistical inference

We have now heard about the South African heart disease data, and we will also look at at a regression problem with prediction of disease progression in diabetes.

Diabetes data

In a medical study the aim was to explain the ethiology of diabetes progression. Data was collected from $n = 442$ diabetes patients, and from each patient the following measurements are available:

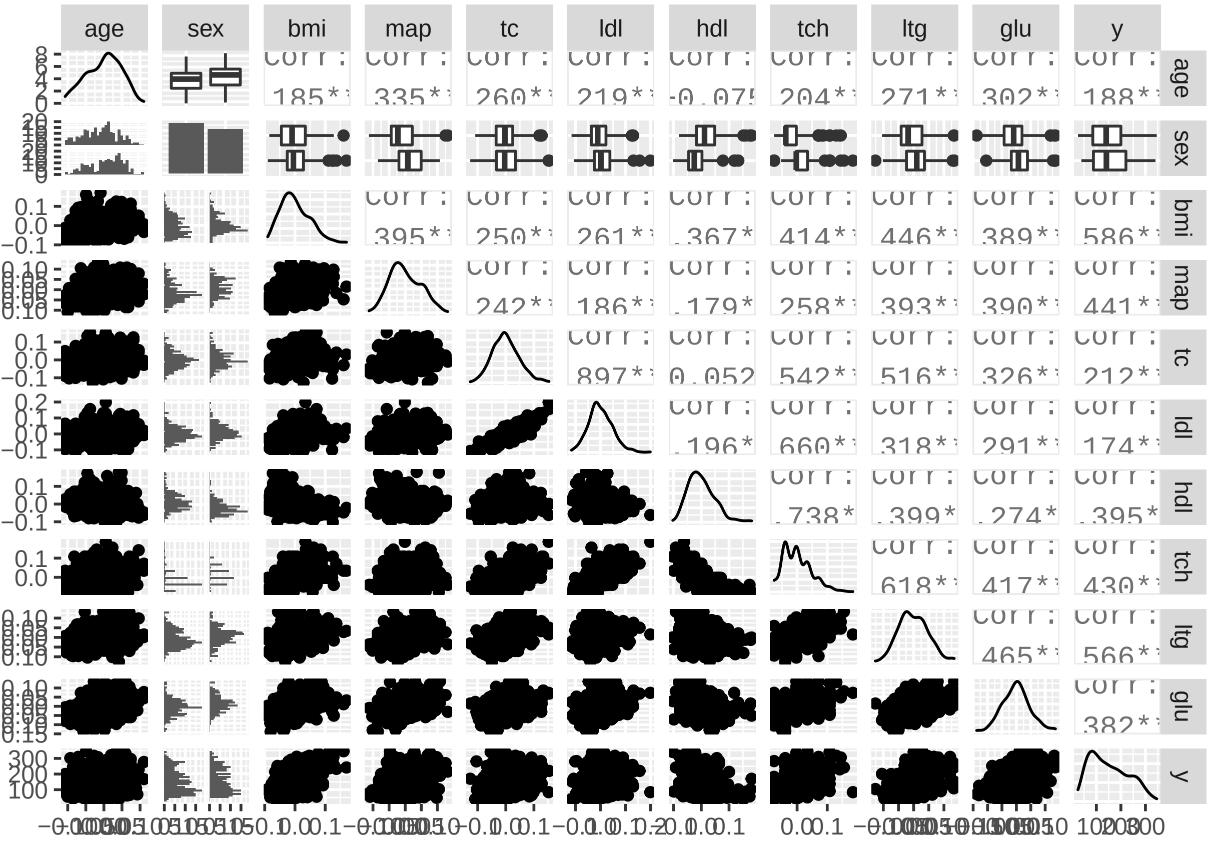
- `age` (in years) at baseline
- `sex` (0=female and 1=male) at baseline
- body mass index (`bmi`) at baseline
- mean arterial blood pressure (`map`) at baseline
- six blood serum measurements: total cholesterol (`tc`), ldl cholesterol (`ldl`), hdl cholesterol (`hdl`), `tch`, `ltg`, glucose `glu`, all at baseline,
- a quantitative measurement of disease progression one year after baseline (`prog`)

All measurements except `sex` are continuous. There are 10 covariates.

The response is the disease progression `prog` - thus a regression problem.

Data can be

- downloaded from https://web.stanford.edu/~hastie/StatLearnSparsity_files/DATA/diabetes.html in three variants: raw, standardized and 442 × 64 matrix with quadratic terms (not used here).
- Or, loaded from the `lars` package, that is automatically loaded in the `monomvn` package (where `blasso` is found).



Prediction vs statistical inference

Prediction

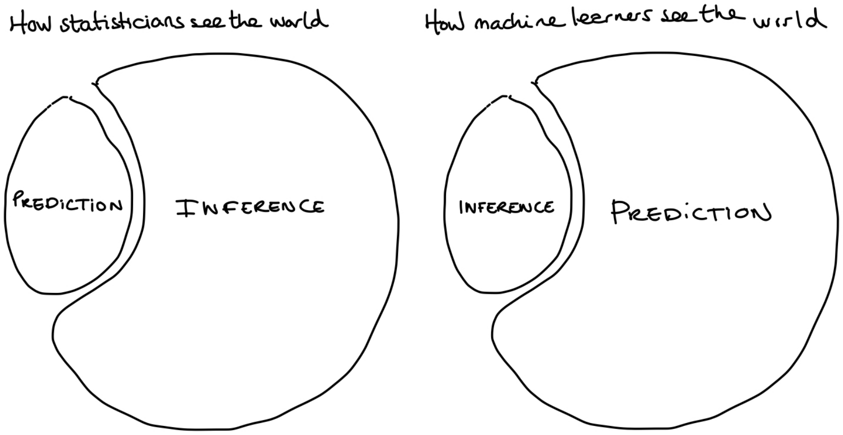
- Predict the value of the progression variable for a person with diabetes.
- Predict the probability of heart disease for a person from the population in the South African heart disease example.

Inference

- Assess the goodness of the prediction (MSE, error rate, ROC-AUC) - with uncertainty.
- Interpret the GLM-model - which covariates is included?
- Confidence interval for the model regression parameters.
- Testing hypotheses about the model regression parameters.

Statistics vs Machine learning

Figures redrawn from [Robert Tibshiran´s Breiman lecture at the NIPS 2015] (https://www.youtube.com/watch?v=RKQJEvc02hc&t=81s). (Conference on Neural Information Processing System)



How statisticians see the world

How machine leerners see the world

Redrawn from NIPS 2015 talk by Robert Tibshirani

Known sampling distributions

For the linear regression and logistic regression we know the sampling distribution of the regression coefficient estimators.

Then it is easy to construct confidence intervals and perform hypothesis tests.

What are the known results?

Sampling distribution for ridge and lasso?

Ridge

From L2: for the normal linear model

$$\hat{\beta}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$$

$$\hat{\beta}(\lambda)_{\text{ridge}} \sim N\{(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{X}\,\beta,$$

$$\sigma^2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\}.$$

$$\text{df}(\lambda) = \text{tr}(\mathbf{H}_\lambda) = \text{tr}(\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T) = \cdots = \sum_{j=1}^{p} \frac{d_j^2}{d_j^2 + \lambda}$$

What can we do with that?

Lasso

Some results using approximations to ridge (for mean and variance, see WNvW p 97), but else *no parametric version of sampling distribution* known.

Conclusion

This is absolutely not straight forward.

The *adaptive nature* of the estimation procedures makes it challenging to perform inference.

- We will *discuss* possible solutions to finding confidence intervals for regression parameters for lasso, and for constructing p-values for testing hypotheses about the regression parameters.
- We will address some *philosophical principles behind inference*
- and mention topics that can be studied further for the interested student!

Warning: there seems not to be consensus, but many interesting approaches and ideas that we may consider.

Bayesian lasso

(HTW 6.1)

In the Bayesian statistics the regression parameters β are random quantities, and in addition to the likelihood also a prior for the regression parameters (and other parameters) are needed.

Multiple linear regression: distribution of response - where we for simplicity assume that we have centred covariates and centred response (so no intercept term)

$$\mathbf{y} \mid \beta, \lambda, \sigma \sim N(\mathbf{X}\beta, \sigma^2\mathbf{I})$$

Prior for regression parameters

$$\beta \mid \lambda, \sigma \sim \prod_{j=1}^{p} \frac{\lambda}{2\sigma} \exp(-\frac{\lambda}{\sigma}|\beta_j|)$$

This prior is called an i.i.d. *Laplacian* (or double exponential) prior.

It can be shown that the negative log of the posterior density for $\beta \mid \mathbf{y}, \lambda, \sigma$ is (up to an additive constant)

$$\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \frac{\lambda}{\sigma}\|\beta\|_1$$

Does this look familiar?

For a fixed value of σ and λ - the β giving the minimum of the negative log posterior is the *lasso* estimate where the regularization parameter is $\sigma\lambda$.

The minimum negative log posterior will then be the same as the maximum log posterior - and the maximum of a distribution is called the *mode* of the distribution.

The lasso estimate is the posterior mode in the Bayesian model.

(Study Figure 6.1 in HTW: describe what we see.)

(Study Figure 6.2 in HTW: compares lasso path with Bayesian lasso with different λs.)

A full Bayesian approach requires priors for λ and σ also.

Markov Chain Monte Carlo MCMC is used efficiently sample realizations form the posterior distribution.

Not only the point estimate

The posterior distribution gives the

- point estimates for the lasso (the mode of the distribution)

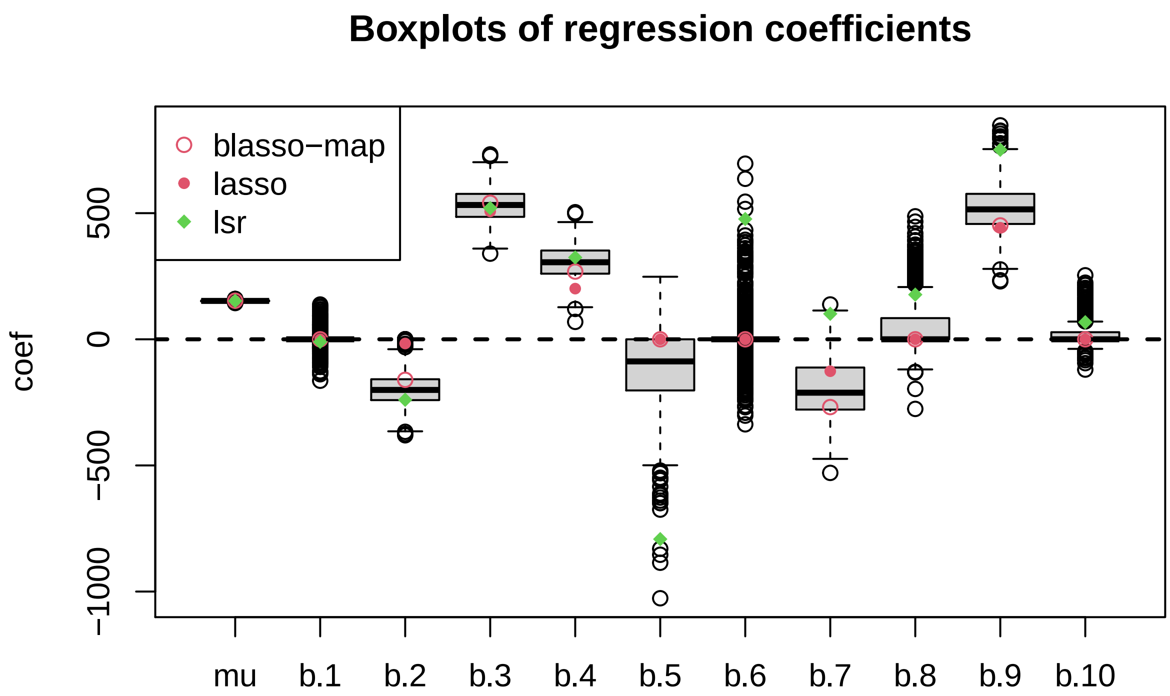
but

- also the *entire joint distribution*.

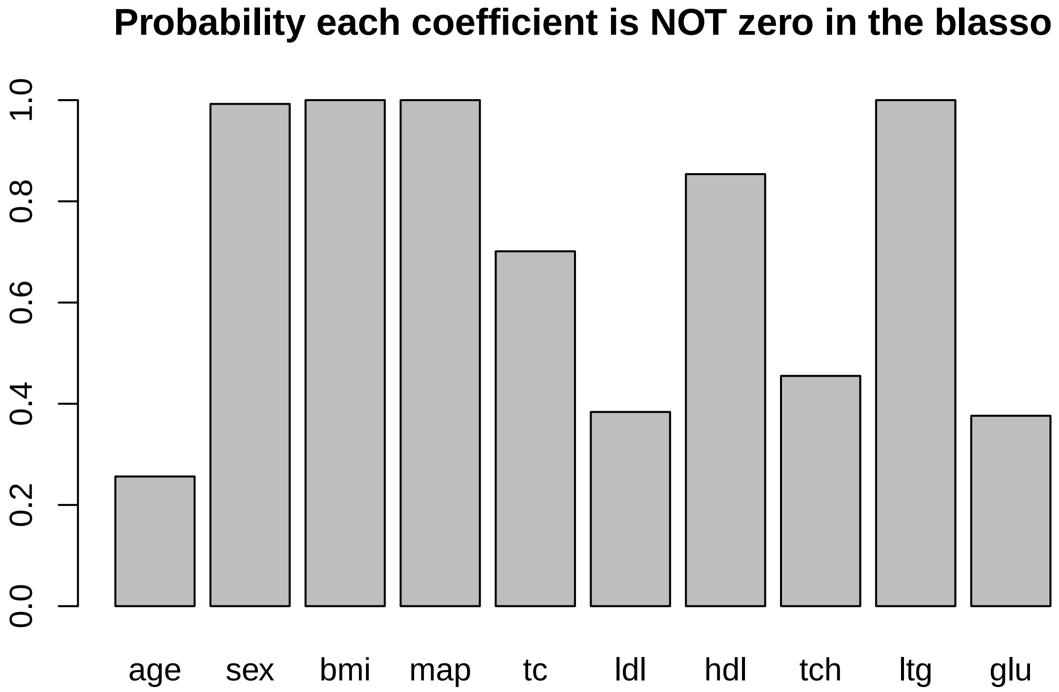
(Study Figure 6.3 in HTW for boxplots and marginal density one regression parameter - based on a *sample* from the posterior distribution.)

Diabetes example with `blasso`

```r
## code below copied from the help(blasso)
## following the lars diabetes example
data(diabetes)
attach(diabetes)

## Ordinary Least Squares regression
reg.ols <- regress(x, y)

## Lasso regression
reg.las <- regress(x, y, method="lasso")

## Bayesian Lasso regression
reg.blas <- blasso(x, y,verb=0)

## summarize the beta (regression coefficients) estimates
plot(reg.blas, burnin=200)
points(drop(reg.las$b), col=2, pch=20)
points(drop(reg.ols$b), col=3, pch=18)
legend("topleft", c("blasso-map", "lasso", "lsr"),
       col=c(2,2,3), pch=c(21,20,18))
```



```
## plot the size of different models visited
#plot(reg.blas, burnin=200, which="m")

## get the summary
s <- summary(reg.blas, burnin=200)

## calculate the probability that each beta coef != zero
barplot(s$bn0,names.arg=colnames(diabetes$x),main="Probability each coefficient is NOT zero in the blas
```



Bootstrap

(HTW 6.2)

Procedure to find lasso estimate $\hat{\beta}(\hat{\lambda}_{CV})$

(Copied word by word from HTW page 142)

Refer to these 6 steps as $\hat{\beta}(\hat{\lambda}_{CV})$-loop

1. Fit a lasso path to (X, y) over a dense grid of values $\Lambda = \{\lambda_l\}_{l=1}^{L}$.
2. Divide the training samples into 10 groups at random.
3. With the kth group left out, fit a lasso path to the remaining 9/10ths, using the same grid Λ.
4. For each $\lambda \in \Lambda$ compute the mean-squared prediction error for the left-out group.
5. Average these errors to obtain a prediction error curve over the grid Λ.
6. Find the value $\hat{\beta}(\hat{\lambda}_{CV})$ that minimizes this curve, and then return the coefficient vector from our original fit in step (1) at that value of λ.

Observe:

- λ-path is the same for each run of the lasso
- the chosen λ is then used on the orginal data

Q: Is it possible to use resampling to estimate the distribution of the lasso $\hat{\beta}$ estimator including the model selection (choosing λ)?

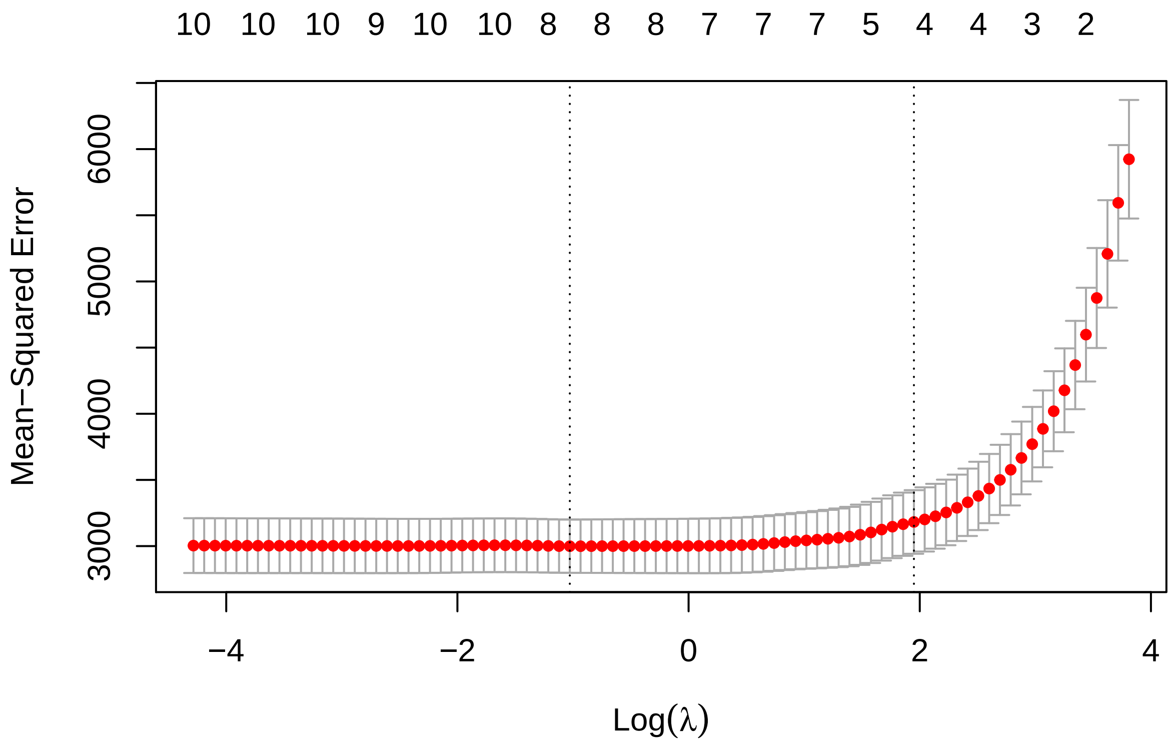
Non-parametric (paired) bootstrap

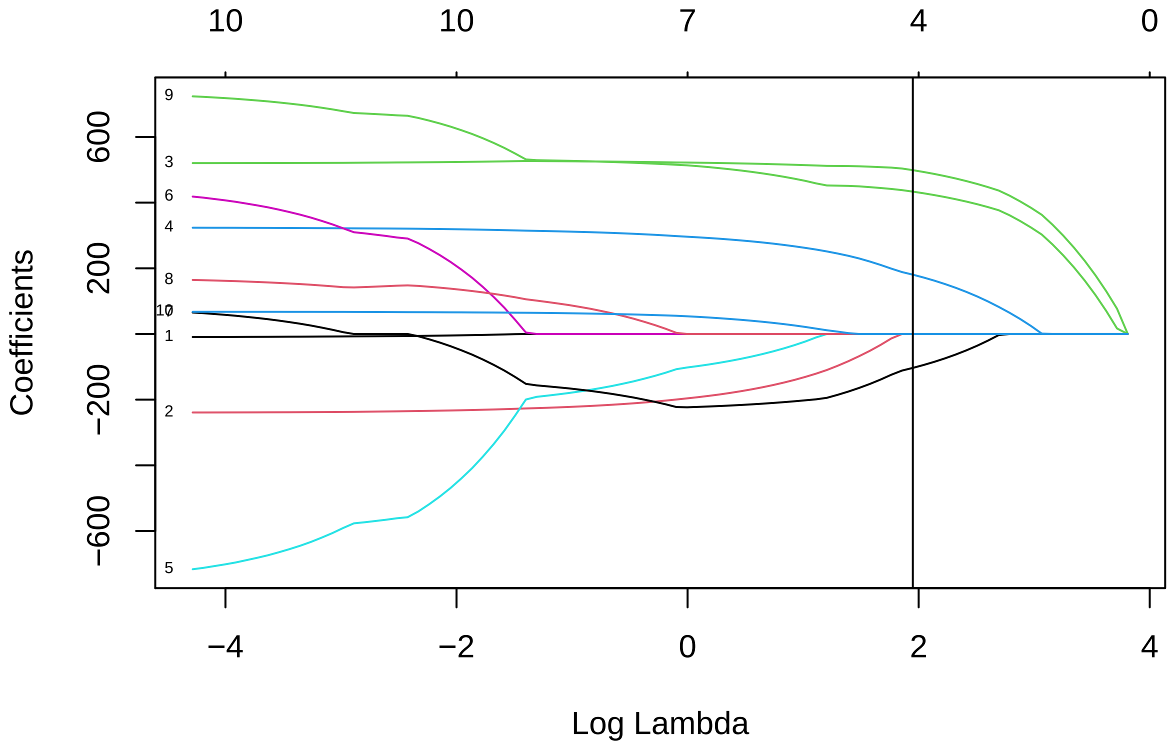
- Let F denote the joint distribution of (X, Y).
- The empirical $\hat{F}$ is $\frac{1}{N}$ for each observation (X, Y) in our training data (X_i, Y_i), $i = 1, \ldots, N$.
- Drawing from $\hat{F}$ is the same as drawing from the N observations in the training data with replacement.

Now, we draw B bootstrap samples from the training data, and for each new bootstrap sample we run through the 6 steps in the $\hat{\beta}(\hat{\lambda}_{CV})$-loop.

- The result is B vectors $\hat{\beta}(\hat{\lambda}_{CV})$.
- We plot the result as
  - boxplots,
  - proportion of times each element of $\hat{\beta}(\hat{\lambda}_{CV})$ is equal 0.

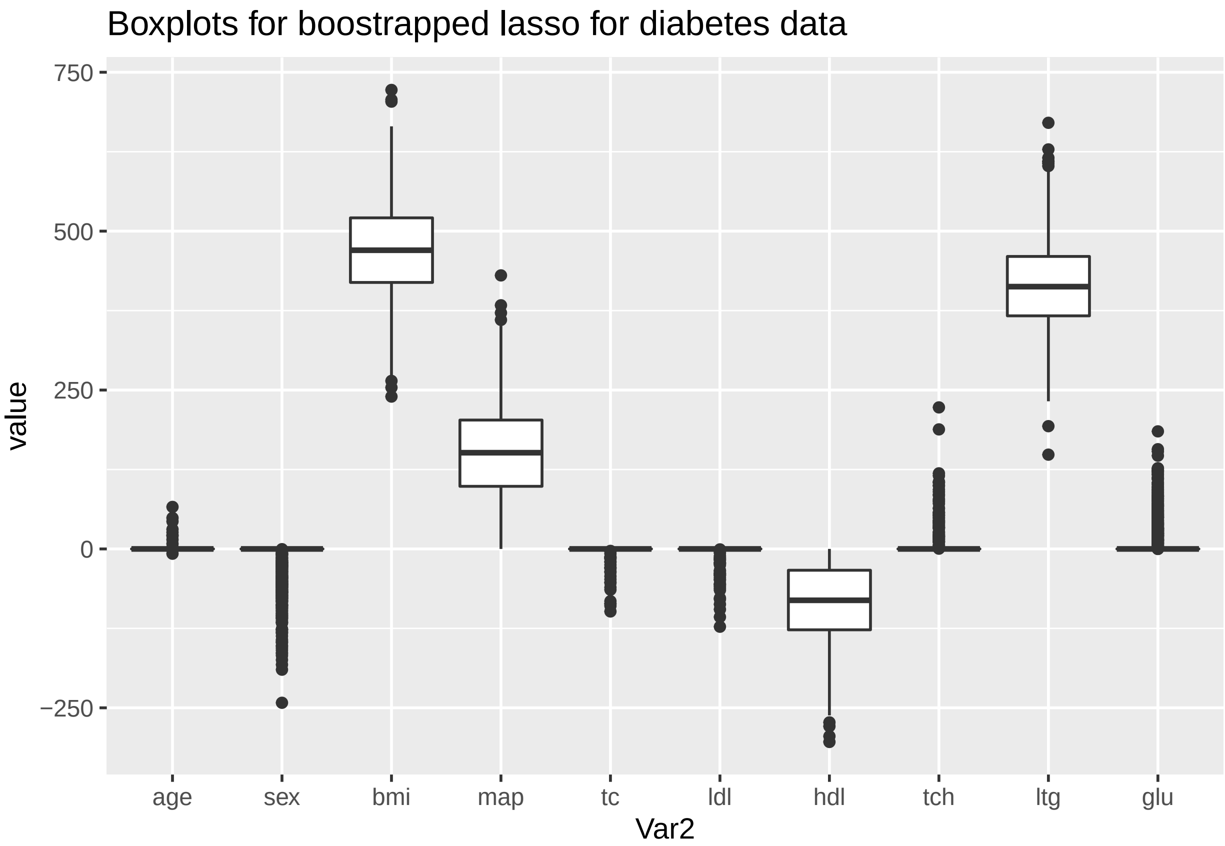
Diabetes example



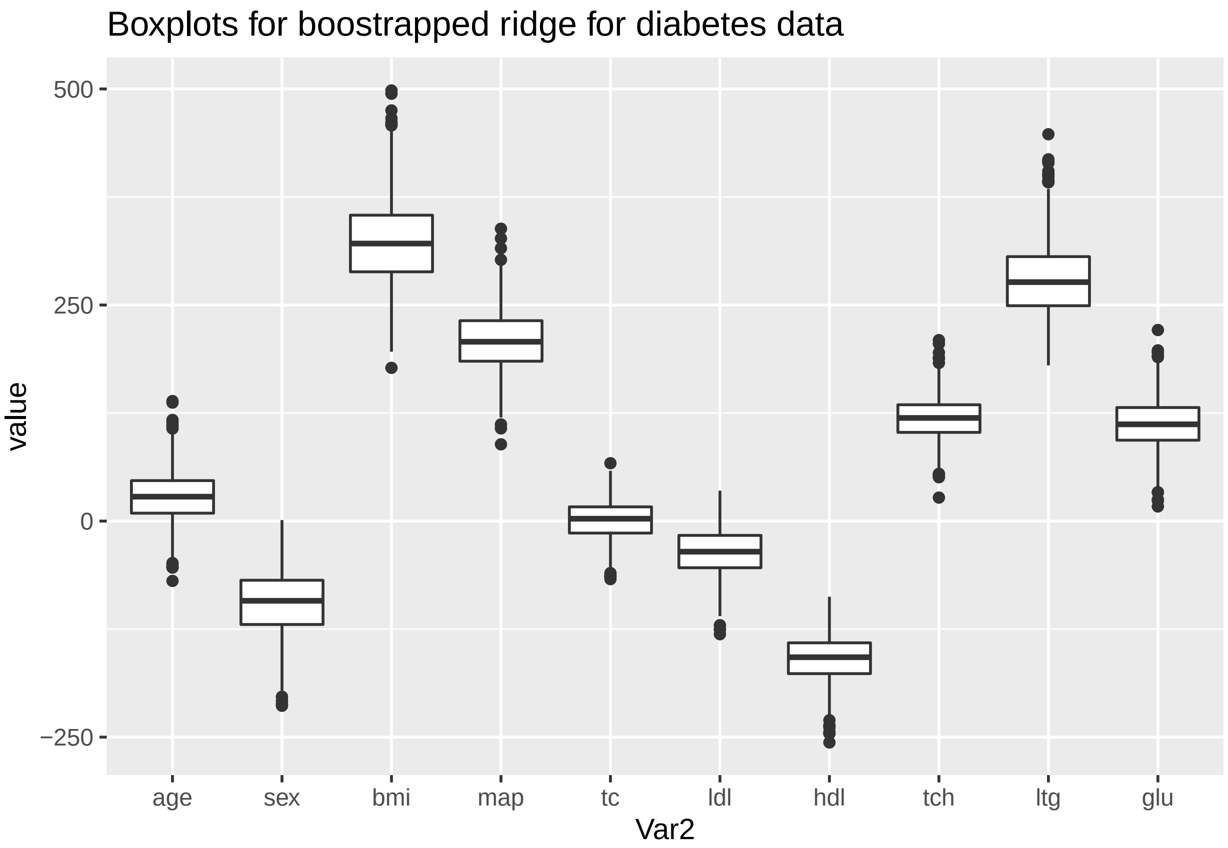


```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                         1
## (Intercept)  152.1335
## age             .
## sex             .
## bmi          498.9191
## map          180.6812
## tc              .
## ldl             .
## hdl         -103.3210
## tch             .
## ltg          433.5648
## glu             .
```

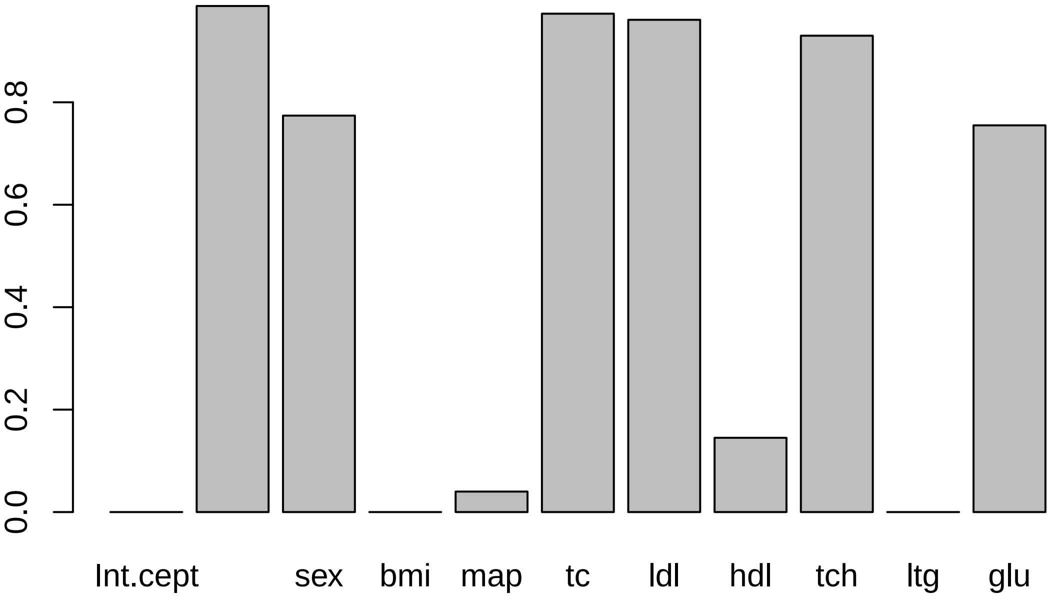
```
lassomat=dget("diabeteslassomat.dd")
ridgemat=dget("diabetesridgemat.dd")

# plotting boxplots

lassomatUI=lassomat[,-1]
lassods=reshape2::melt(lassomatUI,
         variable.name ="variable",value.name="value")
lassopp=ggplot(lassods,aes(x=Var2,y=value))+geom_boxplot()+ggtitle("Boxplots for boostrapped lasso for
lassopp
```



```
ridgematUI=ridgemat[,-1]
ridgeds=reshape2::melt(ridgematUI,variable.name="variable",value.name="value")
ridgepp=ggplot(ridgeds,aes(x=Var2,y=value))+geom_boxplot()+ggtitle("Boxplots for boostrapped ridge for
ridgepp
```



```
lasso0perc=apply(abs(lassomat)<.Machine$double.eps,2,mean)
barplot(lasso0perc)
```



Bootstrapping vs Bayesian lasso

The results from the Bayesian lasso on the proportion of times a coefficient is 0 and the boxplots are very similar to the results from the bootstrapping. The bootstrap seems to be doing the "same" as a Bayesian analysis with the Laplacian prior.

When the model is not so complex and the number of covariates is not too large ($p \sim 100$) the Bayesian lasso might be as fast as the bootstrapping, but for larger problems the bootstrap "scales better".

For GLMs the Bayesian solution is more demanding, but the bootstrap is as easy as for the linear model.

(STudy and compare Figure 6.3 and 6.4 in HTW.)

Medical example

(See Figures from study in slides.)

Bootstrap ridge and lasso percentile CI

What if we calculated percentile bootstrap intervals - could we use that to say anything about the true underlying regression coefficients?

Sadly, there are two main challenges:

- The percentile interval is not a good choice for biased estimators.
- It has been shown that (for fixed p) the asymptotic ($n \leftarrow \infty$) distribution of the lasso has point mass at zero (which leads to that bootstrapping not having optimal properties).

However: much research to check out here - with both different ways perform the bootstrapping and different adjustments for the intervals!

One possibility is the R-package HDCI but the background manuscript is not published yet https://arxiv.org/abs/1706.02150.

Sample splitting

What if we just split the data in two?

Linear regression or logistic regression.

Dataset with p covariates and N observations. Divided into a training set of size aN and a test set of $(1-a)N$, where $a \in [0,1]$.

- Training data used to decide on λ using CV - gives final model where some coefficients is set to 0 and some are shrunken. (The 6 steps.)
- Test data:
  - Fit ordinary LS or GLM model with only the non-zero lasso covariates
  - present CI and p-values from LS

Group discussion: Is this ok? What is gained and what is lost?

From single to multiple hypotheses

In many situations we are not interested in testing only one hypothesis, but instead m hypotheses.

	Not reject H_0	Reject H_0	Total
H_0 true	U	V	m_0
H_0 false	T	S	$m - m_0$
Total	$m - R$	R	m

- Out of the m hypotheses tested, the (unknown) number of true null hypotheses is m_0.
- V: the number of type I errors (false positive findings) and
- T: the number of type II errors (false negative findings).
- U: the number of true null hypotheses that are not rejected and
- S: the number of false null hypotheses that are rejected.
- R: the number of hypoteses rejected for a specific cut-off

Observe: only m and R is observed!

Familywise error rate

The familywise error rate (FWER) is defined as *the probability of one or more false positive findings*

$$\text{FWER} = P(V > 0)$$

The number of false positive findings V is not known in a real life situation, but still we may find a cut-off on the p-value, called α_{loc}, that gives an upper limit to (controls) the FWER.

- Raw p-value, p_j, the lowest nominal level to reject the null hypothesis.

- Adjusted p-value, $\tilde{p}_j$, is the nominal level of the multiple (simultaneous) test procedure at which $H_{0j}, j = 1, \ldots, m$ is just rejected, given the values of all test statistics involved.

The adjusted p-values can be defined as

$$\tilde{p}_j = \inf\{\alpha \mid H_{0j} \text{ is rejected at FWER level } \alpha\}$$

In a multiple testing problem where all adjusted p-value below α are rejected, the overall type I error rate (for example FWER) will be controlled at level α.

The Bonferroni method controls the FWER

Single-step methods controls for multiple testing by estimating one local significance level, α_{loc}, which is used as a cut-off to detect significance for each individual test.

The Bonferroni method is valid for all types of dependence structures between the test statistics.

The local significance level is

$$\alpha_{\text{l}oc} = \frac{\alpha}{m}$$

The adjusted p-value is

$$\tilde{p}_j = \min(1, mp_j)$$

Read more here if needed: Short note on multiple hypothesis testing

High-dimensional inference

(Dezeure, Bühlmann, Meier, Meinshausen, 2.1.1 + 2.2)

- The article has focus on frequentist methods for high-dimensional inference with confidence intervals and p-values in linear and generalized linear models.
- We will focus on linear models.

Set-up:

$$Y = \mathbf{X}\beta^0 + \varepsilon$$

- $\mathbf{X}$ is $n \times p$ design matrix
- Y is an $n \times 1$ response vector
- ε is an $n \times 1$ error vector, independent of $\mathbf{X}$ and i.i.d. entries with $\mathrm{E}(\varepsilon)_i) = 0$.
- The number of parameters *may* be larger than the sample size (then the regression parameter is not identifiable in general).

The *active set* is

$$S_0 = \{j; \beta_j^0 \neq 0, j = 1, \ldots, p\}$$

with cardinality (size) $|S_0|$.

Now: construct CI and p-values for *individual regression parameters* β_j^0, $j = 1, \ldots, p$, and also with multiple testing adjustment.

Remark: We want inference for *all* coefficients - not only the ones that the lasso has selected.

- The lasso has desirable properties for estimating β^0 in high dimensional models, in particular for prediction $\mathbf{X}\beta^0$ or a new response Y_{new}.
- But, the distribution of the estimator is hard to characterize, and
- it has been shown that (for fixed p) the asymptotic ($n \leftarrow \infty$) distribution of the lasso has point mass at zero (which leads to that bootstrapping not having optimal properties).

For the situation $p >> n$ extra assumptions are needed, called the *compatibility condition* on the design matrix, and this guarantees identifiability and socalled oracle optimality results for the lasso. However, this is reported to be *unrealistic in practical situations*.

Single-sample splitting

1) Split the data into two (equal) halves, I_1 and I_2, no observations in common.
2) I_1 is used for model selection (with the lasso), with active variables in $\hat{S}(I_1)$.
3) The selected covariates in $\hat{S}(I_1)$ is used for estimation in I_2. To construct p-values, P_j, for example use LS with t-tests. P-values for variables not selected is set to 1. Remark: then the number of covariates selected in I_1 need be smaller than the sample size for I_2.
4) The raw p-values is corrected for multiple testing (Bonferroni method controlling FWER)

$$P_{\text{corr},j} = \min(P_j \cdot |\hat{S}|, 1)$$

This avoids using the data twice! But, is very sensitive to the split - giving *wildly* different p-values=*p-value lottery*

How can this be amended?

Multiple-sample splitting

- The single-sample splitting routine is run B times giving $P^{[b]}_{\text{corr},j}$ for $b = 1, \ldots, B$ and $j = 1, \ldots, p$.
- Problem: how aggregate the B p-values for each j to give one p-value?
- The different b runs have many observations in common, so the p-values for covariate j are correlated.
- The authors have shown in a previous article that for dependent p-values that one solution (that gives valid p-values) is to take the median and multiply with 2.
- The result is more general, and γ is a general quantile ($\gamma = 0.5$ for the median):

$$Q_k(\gamma) = \min(\text{empirical } \gamma - \text{quantile}\{P^{[b]}_{\text{corr},j}/\gamma, b = 1, \ldots, B\}, 1)$$

- The authors get more advanced and choose to search all γ within the interval $(\gamma_{\min}, 1)$, where a common choice is $\gamma_{\min} = 0.05$, to get the smallest p-value. However there is a price to pay: $(1 - \log(\gamma_{\min}))$

$$P_j = \min((1 - \log(\gamma_{\min}) \cdot \inf_{\gamma \in (\gamma_{\min}, 1)} Q_j(\gamma)), 1)$$

for $j = 1, \ldots, p$.

Some assumptions are necessary to assure FWER control.

Confidence intervals are found

- from the adjusted p-values P_j
- using the duality of p-values and two-sided confidence intervals. That is,
- a $(1 - \alpha)$ 100% CI contains values c where the p-value is below α for testing $H_0 : \beta_j = c$. A closed form solution involving P_j is found.
- Both single testing and multiple corrected testing CIs are found. (Appendix A.2 in article)

```
data(diabetes)
x=cbind(diabetes$x)#,diabetes$x2)
y=diabetes$y

hdires=multi.split(x=x,y=y,B=1000,fraction=0.5,
                   ci.level=0.95, model.selector=lasso.cv,
                   classical.fit=lm.pval, classical.ci=lm.ci,
                   return.nonaggr = FALSE, #if not adj for multiple testing
                   return.selmodels=FALSE, #just to have a look!
                   verbose=FALSE)
dput(hdires,"hdires.dd")
```

```
hdires=dget("hdires.dd")
names(hdires)
```

```
##  [1] "pval"             "pval.corr"        "pvals.nonaggr"    "ci.level"
##  [5] "lci"              "uci"              "gamma.min"        "sel.models"
##  [9] "method"           "call"             "clusterGroupTest"
```

```
hdires$gamma.min
```

```
##  [1] 0.999 0.999 0.050 0.064 0.999 0.999 0.050 0.999 0.052 0.999
```

```r
#summary(hdires$pvals.nonaggr) # if return.nonaggr=TRUE
hdires$pval.corr
```

```
##          age          sex          bmi          map           tc          ldl
## 1.000000e+00 1.000000e+00 5.178832e-10 1.331537e-02 1.000000e+00 1.000000e+00
##          hdl          tch          ltg          glu
## 4.533731e-01 1.000000e+00 6.863052e-08 1.000000e+00
```

```r
cbind(hdires$lci, hdires$uci)
```

```
##           [,1]      [,2]
## age       -Inf       Inf
## sex -435.92911 103.48598
## bmi  363.10330 776.42063
## map   66.58321 472.64439
## tc        -Inf       Inf
## ldl       -Inf       Inf
## hdl -406.96777  14.72096
## tch -764.32633 201.01275
## ltg  301.70828 715.18364
## glu -327.08642 254.88420
```

---

hdi with logistic regression

For modifications to the call to `mult.split` see the Appendix of the master thesis of

- Martina Hall: "Statistical Methods for early Prediction of Cerebral Palsy based on Data from Computer-based Video Analysis".
- Dag J. Kristiansen: "Detecting Neuronal Activity with Lasso Penalized Logistic Regression"
- Haris Fawad: " Modelling Neuronal Activity using Lasso Regularized Logistic Regression"(Modelling Neuronal Activity using Lasso Regularized Logistic Regression) - also git repo neuro-lasso.

hdi - also with other solutions

In the `hdi` package also solutions for *debiasing* the lasso estimator is included. (See HTW 6.4 or the Dezure et al article.)

Summing up

What is the take home message from this "Sample splitting" story?

Inference after selection

(Taylor and Tibshirani, 2015 and HTW 6.3)

The plot

Let us leave the lasso for a while.

1980: small data sets, planned hypothesis to test ready before data collected, no model selection. Only fit model and look at CI and p-values.

After 1980: larger data sets and looking at data to give best model. New challenge: *how to do inference after selection*.

This is an important topic that is not a part of ANY statistical courses at IMF.

The main question is:

- we have used a selection method (forward selection, lasso) to find potential association between covariates and response,
- with focus on interpreting the selected model: how can we assess the strength (read: CI and p-value) of these findings?

The answer includes:

- we have "cherry picked" the strongest associations, and we can thus not just report CI and p-values based on the final model - when all is done on the same data set.

In this story we now focus on *understanding how our model selection influences the inference on the final model.*

The technical solutions are of less importance, and is not presented with enough mathematical detail so that we understand the method in detail.

Remark: the single and multiple sample splitting strategy isvalid.

Forward stepwise regression

Aim: Multiple linear regression - where forward stepwise regression is used to select the model

$$y_i = \beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i$$

- Start with an empty model (only intercept)

While some stopping criterion not reach - perform step

- At step k add the predictor that gives the most decrease in the sums of squares of error (here now - to follow the notation denoted by RSS instead of what we previously called SSE)

$$\text{RSS} = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where $\hat{y}_i$ is the predicted value for observation i in this model.

If we have a model with $k-1$ predictors and we would like to add one more predictor (the k to be added):

$$\text{R}_k = \frac{1}{\sigma^2}(\text{RSS}_{k-1} - \text{RSS}_k) \sim \chi^2_1$$

where here σ^2 is assumed known.

This can be seen as an hypothesis test:

H_0 : The new predictor is not relevant vs. H_1 : The new predictor is relevant with a p-value calculated from the upper tail of the χ^2_1-distribution.

Alternative scenario:

For simplicity - we look at the $k=1$:

The order in which the predictors is to be entered in to the model was decided before the data was collected (or at least before the data was analysed or plottet).

Then: we are all good - and this p-value from the χ^2_1-distribution will be a valid p-value.

What does this mean in practice?

Back to original scenario:

Assume we are at step $k = 1$, and will add the first predictor which is the one with the largest R_1.

Will the distribution of the *maximal* R_1 be the same as the distribution of a given predefined R_1?

Distribution to the maximum given that we have p predictors:

$$P(\max R_1 \geq c) = 1 - P(\max R_1 < c) = 1 - P(\text{all p } R_1 \text{ are} < c)$$

Study Figure 1 in the article for a plot of nominal vs. actual p-value for $p = 10$ and $p = 50$. The figure was made using Monte Carlo sampling.

Moving on to $k > 1$

- We would like to obtain valid ("correct") p-values for all steps, not only for $k = 1$.
- Monte Carlo solution would be elaborate.

The method used in the article is to calculate a p-value for the covariate at step k by conditioning on the fact that already the strongest $k - 1$ predictors in this sequential set-up has already been chosen.

The p-value at step k would be dependent on the number of covariates p.

We now change focus and look at the distribution of the estimated regression coefficient for the covariate added at step k, because that can be used to construct both a CI for the coefficient and a p-value for testing if the coefficient is different from zero.

The polyhedral result

(for details consult HTW 6.3 or articles references to in the Taylor and Tibshirani article)

Distribution for regression coefficient:

- Assume that we are at some step k, and that $k - 1$ covariates are in the model.
- We have found the new covariate to include, and fitted the model with the k covariates.
- Standard theory tells us that the estimator $\hat{\beta}$ for covariate k is unbiased and follows a normal distribution with some variance τ^2.

$$\hat{\beta} \sim N(\beta, \tau^2)$$

But, this is given that we only had these k covariates available at the start. We will instead *condition on* selection event.

It turns out that the selection event can be written in a *polyhedral form* $Ay \leq b$ for some matrix A and some vector b.

At each step of the forward selection we have a competition among all p variables, and the A and b is used to construct the competition.

The correct distribution of the estimator $\hat{\beta}$ for covariate now has a *truncated normal distribution*

$$\hat{\beta} \sim TN^{c,d}(\beta, \tau^2)$$

i.e. the *same* normal distribution, but scaled to lie within the interval (c, d).

The limits (c, d) depends on both the data and the selection evetns that lead to the current model.

The formulae for these limits are somewhat complicated but easily computable.

This truncated normal distribution is used to calculate *selection-adjusted* p-values and confidence interval.

(Study Figure 3. in the article by Taylor and Tibshirani.)

Polythedral lasso result

The same methodology can be used for the lasso, here also the selection of predictors can be described as a polythedral region of the form $Ay \leq b$ - for a fixed value λ.

For the lass the A and b will depend on

- the predictors
- the active set
- λ

but not on y.

The methods are on closed form, but the values c and d may be of complicated form.

The R package `selectiveInference` can be used to find post selection p-values both for forward stepwise selection and for the lasso.

See the package help for details.

Study Figure 5 from the article for an example of Naive and Selection-adjusted intervals for the lasso.

Further improvements

The method yields rather wide confidence intervals for the regression parameters (given that we translate the p-values into CIs).

There exists improvements to the results, in particular a method called *carving* which is explained in the you-tube videos from a course with Snigdha Paragrahi

- Tutorial I
- Tutorial II

PoSI

(HTW 6.5)

- The POSI method also fits a selected submodel and
- adjust the standard CIs by *accounting for all possible models that might have been delivered by the selection procedure.*

- This means the method can be used on published results where the complete selection process is not explained in detail.
- This lack of information of the selection process leads to *very wide CIs*.

Inference is based on the submodel M chosen, and on the projection of $\mathbf{X}\beta$ onto the space spanned by the submodel M:

$$\beta_M = (\mathbf{X}_M^T \mathbf{X}_M)^{-1} \mathbf{X}_M^T \mathbf{X}\beta$$

The method considers a confidence interval for the jth element of β_M of the form

$$\text{CI}_{jM} = \hat{\beta}_{jM} \pm K\hat{\sigma}v_{jM}$$

where $v_{jM}^2 = (\mathbf{X}_M^T \mathbf{X}_M)_{jj}^{-1}$

The constant K is found to satisfy

$$P(\beta_{jM} \in \text{CI}_{jM}) \leq 1 - 2\alpha$$

over all possible selection models.

K is a function of the data matrix $\mathbf{X}$ and the maximum number of nonzero component allowed in β_M. An upper bound on K is known from a result on simultaneous intervals by Scheffe.

HTW page 161: Reports on the diabetes data with submodels of size 5, where the 95% CI value of K is 4.42 (“little less than 2 hours of computing”).

Details may be found in the PoSI 2013 article (reference Berk et al below).

PoSI R-package

*In linear LS regression, calculate for a given design matrix the multiplier K of coefficient standard errors such that the confidence intervals [b - K*SE(b)*, b + K*SE(b)*] have a guaranteed coverage probability for all coefficient estimates b in any submodels after performing arbitrary model selection.*

Results for the Boston housing data is available in the help section.

https://cran.r-project.org/web/packages/PoSI/index.html

Post selection inference and the reproducibility crisis

The *incorrect* use of CIs and p-values in models found from model selection *and* inference on the same data - is though to be one of the main contributors to the *reproducibility crisis in science.*

Selective Inference: The Silent Killer of Replicability by Yoav Benjamini Published on Dec 16, 2020

Conclusion

How will you perform inference

on Data Analysis Project 1?

We discuss:

1) You can afford a test set
2) You have no test set

References

(also given in the text and not repeated here)

- Wessel N. van Wieringen: Lecture notes on ridge regression
- A. Chatterjee and S. N. Lahiri (2011). Bootstrapping Lasso Estimators. Journal of the American Statistical Association. Vol. 106, No. 494 (June 2011), pp. 608-625 (18 pages)
- Single/multi-sampling splitting part of Dezeure, Bühlmann, Meier, Meinshausen (2015). "High-Dimensional Inference: Confidence Intervals, p-Values and R-Software hdi". Statistical Science, 2015, Vol. 30, No. 4, 533–558 DOI: 10.1214/15-STS527 (only the single/multiple sample splitting part in 2.1.1 and 2.2 for linear regression, and using the method in practice).
- Taylor and Tibshirani (2015): Statistical learning and selective inference, PNAS, vol 112, no 25, pages 7629-7634. (Soft version of HTW 6.3.2)
- Berk, Richard; Brown, Lawrence; Buja, Andreas; Zhang, Kai; Zhao, Linda. Valid post-selection inference. Ann. Statist. 41 (2013), no. 2, 802–837. doi:10.1214/12-AOS1077.