Bayesianische Variablenselektion

Volker Schmid

10. Juli 2017

Ridge und Lasso

Indikatorvariablen

Spike and Slab

Ridge und Lasso

Bayesianisches (Generalisiertes) Lineares Modell

Gegeben seien n Beobachtungen einer Zielvariable y und von p Kovariablen $x_1, \ldots, x_p$.

$$y_i | \mu_i, \phi_i \sim f(\mu_i, \phi_i) \text{ i.i.d.}$$
$$h(\mu_i) = \sum_{j=1}^{p} \beta_j x_{ij}$$

Bayesianisches lineares Regressionsmodell

$$y_i|\beta, \sigma^2 \sim \mathrm{N}(\mu_i, \sigma^2)$$
$$\mu_i = \sum_{j=1}^{p} \beta_j x_{ij}$$

Welche Priori-Information haben wir über die β? Erstmal keine...

$$p(\beta_j) \propto \text{const.}$$

Können wir sehen als uneigentliche Normalverteilung (hier konjugierte Verteilung) mit Varianz unendlich. Damit ist

$$\beta \sim N_p(\hat{\beta}, \Sigma)$$

mit $\hat{\beta} = (X'X)^{-1}X'y$ dem KQ-Schätzer!

Ridge-Regression I

Nun: $p >> n$.

Idee der Ridge-Regression: Viele der β Parameter sollen gleich oder nahe Null sein. Bestrafe daher Parameter, die zu stark von der Null abweichen. Damit penalisierter log-Likelihood-Ansatz:

$$l_{pen}(\beta) = l(\beta) - \frac{\lambda}{2} \sum_{j=1}^{p} \beta_j^2$$

Bayesianisch gedacht: Wir haben die Vorinformation, dass die Parameter nahe Null sind. Kombiniert mit konjugiertem Priori-Ansatz kommen wir auf:

$$\beta_j \sim N(0, \tau^{-1}) \quad \forall j$$

Ridge-Regression II

Die Log-Priori-Dichte ist

$$\log(p(\beta)) = -\frac{\tau}{2}\sum_{j=1}^{p}\beta^2 + C$$

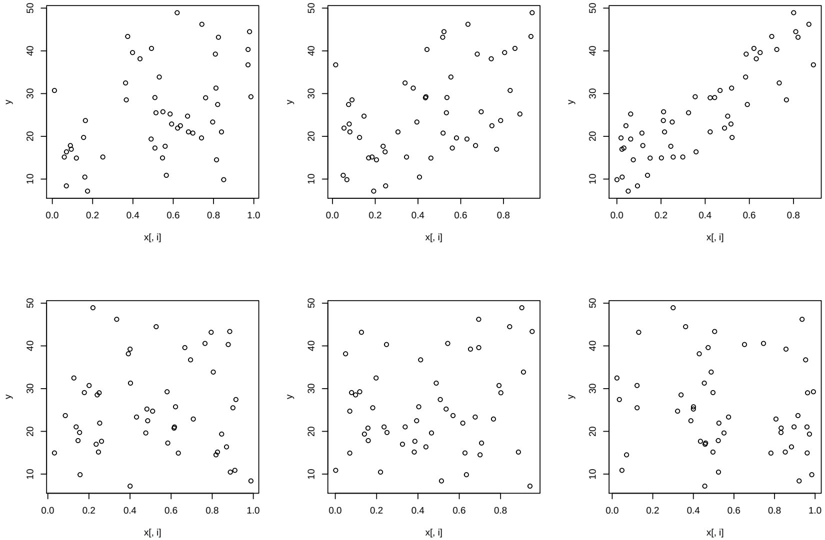
Damit sind penalisierte log-Likelihood und log-Posteriori bis auf Konstanten identisch. Ein Maximum-A-Posteriori-Ansatz liefert also selbes Ergebnis wie ein penalisierter log-Likelihood-Ansatz (Tikhonov-Regularisierung).

Beispiel Bayesianische Ridge-Regression I

Wir konstruieren einen Beispieldatensatz

```
n <- 50
p <- 100
true.sigma2 <- 0.001
x <- matrix(runif(n*p), nrow=n)
true.beta <- c(10,20,30, rep(0,p-3))
mu <- as.vector(x%*%true.beta)
y <- rnorm(n,mu,sqrt(true.sigma2))
par(mfrow=c(2,3))
for (i in c(1:4,10,90))
  plot(x[,i],y)
```

Beispiel Bayesianische Ridge-Regression II



Posteriori

$$
\begin{aligned}
p(\beta, \tau, \sigma^2 | y) \propto \sigma^{-n} \exp &\left(-\frac{1}{2\sigma^2} \sum_i (y_i - \sum_j \beta_j x_{ij})^2 \right) \\
&\cdot \tau^{p/2} \exp \left(-\frac{\tau}{2} \sum_j \beta_j^2 \right) \\
&\cdot \tau^{a-1} \exp(-\tau b) \\
&\cdot \sigma^2 - a_0 - 1 \exp(-b_0/\sigma^2)
\end{aligned}
$$

Damit gilt:

- $\beta | \tau, \sigma^2 \sim N(\hat{\beta}, \Sigma)$ mit $\hat{\beta} = (X'X + \tau I)^{-1} X'y$ und $\Sigma = (X'X + \tau I)^{-1})$.
- $\tau | \beta \sim Ga(a + p/2, b + \sum \beta_j^2/2)$
- $\sigma^2 | \beta, y \sim IG(a_0 + n/2, b + \sum(\epsilon_i^2))$ mit $\epsilon_i = y_i - \sum_j \beta_j x_{ij}$

MMCM I

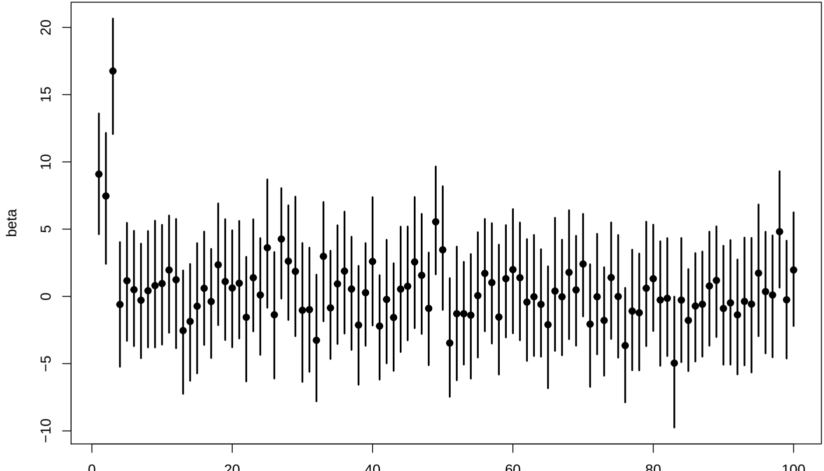
```r
beta<-rep(0,p)
XX <- t(x)%*%x
Xy <- t(x)%*%y
tau <- 1
sigma2 <- 1
a0 <- 1
b0 <- 0.001
a <- 1
b <- 0.1

beta.save<-array(NA,c(p,500))
tau.save<-rep(NA,500)
sigma2.save<-rep(NA,500)
```

MMCM II

```
for (i in 1:1000)
{
  Sigma <- solve(XX+tau*diag(p))
  mu <- Sigma%*%Xy
  beta <- mnormt::rmnorm(1,mu,Sigma)

  tau <- rgamma(1, a+p/2, b+sum(beta^2)/2)

  sigma2 <- 1/rgamma(1, a0+n/2, b0+sum((y-x%*%beta)^2))

  if (i>500)
  {
    beta.save[,i-500]=beta
    tau.save[i-500]=tau
    sigma2.save[i-500]=sigma2
  }
}
```

Plot

```
beta.qu<-apply(beta.save,1,quantile,probs=c(.05,.5,.95))
plot(beta.qu[2,],pch=19,ylim=range(beta.qu),ylab="beta")
for (i in 1:p)
lines(rep(i,2),beta.qu[c(1,3),i],lwd=2)
```



Relevante Kovariablen

```
sum(beta.qu[2,]==0)
```

```
## [1] 0
```

```
print(which(beta.qu[1,]>0))
```

```
## [1]  1  2  3 49 98
```

```
print(which(beta.qu[3,]<0))
```

```
## [1] 83
```

- Bei Ridge werden die Parameter Richtung Null gedrückt
- Aber: Parameter werden nicht genau gleich Null!

Lasso

Alternative: Lasso (L_1-Regularisierung)

$$pen(\beta) = \sum_j |\beta_j|$$

Bayesianisch analog zu Ridge:

$$p(\beta_j) \propto \exp\left(-\frac{\tau}{2} \sum |\beta_j|\right)$$

- Das entspricht einer Laplace-Verteilung mit Erwartungswert 0
- Aber: (erstmal) kein Gibbs-Sampler mehr möglich

Bayesianischer Lasso I

Nach *Park, Trevor and Casella, George. The Bayesian Lasso. Journal of American Statistical Association. 103(482):681-686. 2008* gilt äquivalent:

$$
\begin{aligned}
\beta_j|\sigma^2, \tau_j^2 &\sim N(0, \sigma^2\tau_j^2) \\
\tau_j|\sigma^2 &\sim Exp(\lambda^2/2) \\
\lambda^2 &\sim Ga(a, b)
\end{aligned}
$$

Damit läßt sich wiederum ein Gibbs-Sampler konstruieren.

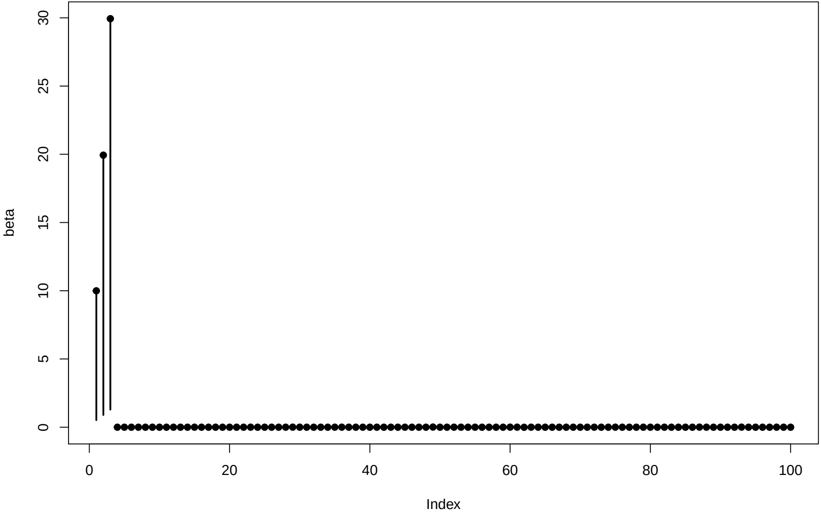
```
beta.L<-gibbsBLasso(x, y, max.steps = 10000)
```

Bayesianischer Lasso II

```
## Iteration: 1000
Iteration: 2000
Iteration: 3000
Iteration: 4000
Iteration: 5000
Iteration: 6000
Iteration: 7000
Iteration: 8000
Iteration: 9000
Iteration: 10000
```

```r
plot(beta.L[2,],pch=19,ylim=range(beta.L),ylab="beta")
for (i in 1:p)
lines(rep(i,2),beta.L[c(1,3),i],lwd=2)
```

Bayesianischer Lasso III




```
sum(beta.L[2,]==0)
```

Bayesianischer Lasso IV

```
## [1] 0
```

```r
print(which(beta.L[1,]>0))
```

```
##  [1]  1  2  3 15 16 17 23 26 27 28 29 30 35 36 37 49 52
## [24] 79 80 87 93 98
```

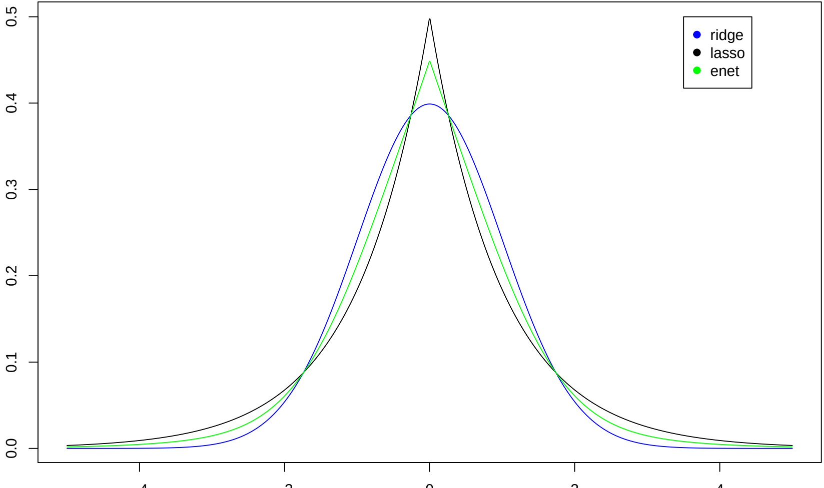
```r
print(which(beta.L[3,]<0))
```

```
## integer(0)
```

Elastic Net

Ridge und Lasso lassen sich kombinieren:

$$p(\beta_j) \propto \exp\left(-\frac{\tau}{2}\sum|\beta_j| - \frac{\nu}{2}\sum\beta_j^2\right)$$



Indikatorvariablen

Indikatorvariablen

Setze

$$\beta_i = I_i \tilde{\beta}_i$$

wobei I_i eine (0/1-)Indikatorvariable ist.

Ist $I_i = 0$, wird β_i auf 0 gesetzt, $\tilde{\beta}_i$ wird aus der Priori gezogen.

Ising-Feld

- Ansatz lässt sich auch auf Kovariablen mit bekannter/angenommener Korrelation anwenden
- Z.B. Gene auf DNA, Bilder
- Auf die Indikatorvariablen wird das ein Ising-Feld angenommen
- mit $J_i = 2I_i - 1$, also $J_i \in \{-1, 1\}$

$$p(I) \propto \exp\left(-\tau \sum_{i \sim j} J_i J_j\right)$$

- Sampling daraus allerdings schwierig
- Alternative: Probit-Modell

$$I_i = \begin{cases} 1 \text{ für } \phi > 0 \\ 0 \text{ für } \phi \leq 0 \end{cases}$$

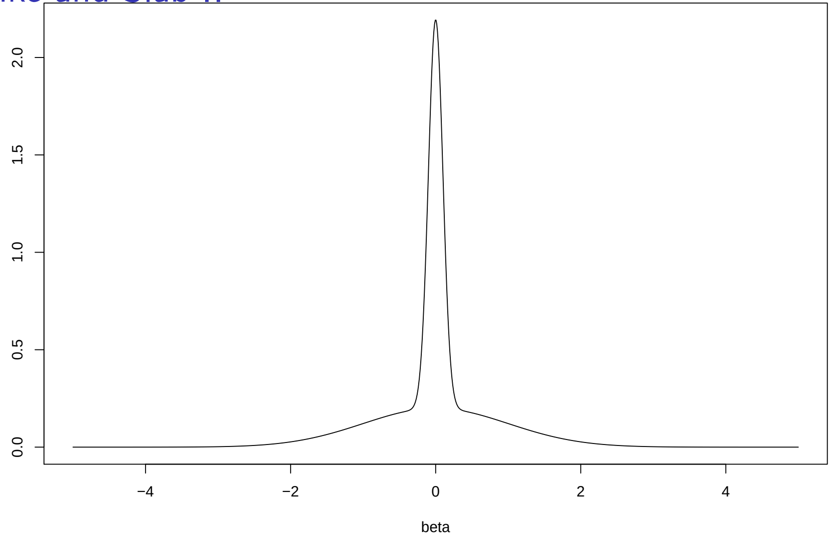
mit

Spike and Slab

Spike and Slab I

- Idee: Damit $p(\beta = 0|y) > 0$, muss $p(\beta = 0) > 0$ sein
- Kombiniere flache Priori (Slab) mit Punktmasse auf Null (Spike)
- Computational bessere Darstellung als Mischung von zwei Normalverteilungen mit sehr großer uns sehr kleiner Varianz

Spike and Slab II



- Ähnlichkeiten zu Elastic Net, wenn Lasso über Normalverteilung modelliert wird.

Spike and Slab III

- z.B. Implementation in spikeSlabGAM-Paker

$$
\begin{aligned}
\beta|\gamma, \tau^2 &\sim N(0, \tau^2\gamma) \\
\gamma|w &\sim wI_1(\gamma) + (1-w)I_{\nu_0}(\gamma) \\
\tau^2 &\sim IG(a_\tau, b_\tau) \\
w &\sim Beta(a_w, b_w)
\end{aligned}
$$

- ν_0 sehr klein, entspricht *spike*

WGRR und gen

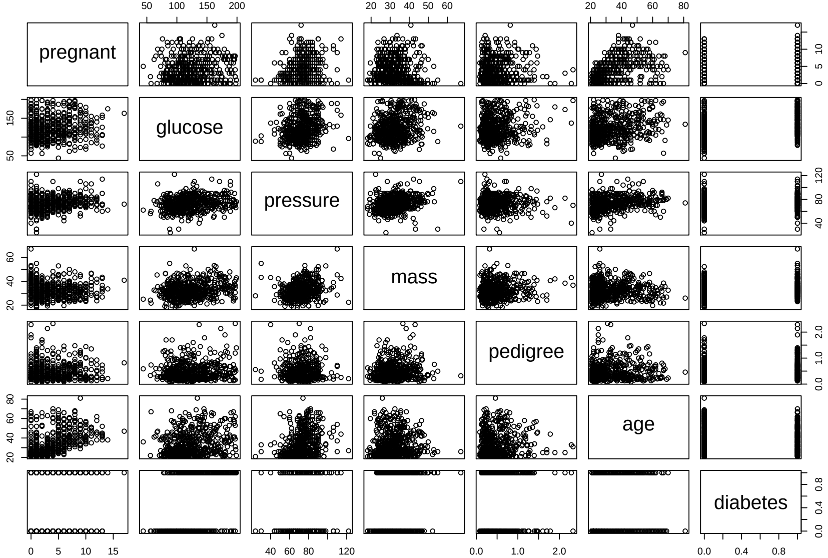
- Ishwaran und Rao (2014) zeigen, dass die Spike and Slab prior ein Spezialfall der gewichteten generalisierten Ridge-Regression sind (weighted generalized Ridge regression, WGRR)
- Beim WGRR können die β Parameter auf Null gesetzt werden, wir erhalten $p(\beta = 0|y)$
- Durch Bayesian Model Averaging gehen aber z.B. bei $E(y^*|y)$ weiterhin viele/alle Regressionsparameter ein
- Ishwaran und Rao (2014) schlagen weiterhin das generalizes elatic net (gen) vor, das mehr Paramete auf Null setzt

spikeslabGAM paket I

Beim spikeSlabGAM-Paket wird die Variablenselektion auf die glatten Effekte angewandt:

```
## ## ----- This is spikeSlabGAM 1.1-11 ----- ##
## Please note that a recent update to gridExtra has made i
##   to change the interface for <plot.spikeSlabGAM> starti
## Instead of arguments 'rows', 'cols', 'widths', 'heights
##   it now accepts only 'nrow' and 'ncol'.
## Arguments 'widths' & 'heights' can still be defined and
##   to <gridExtra::marrangeGrob>.
## Sorry for the inconvenience.
```

spikeslabGAM paket II



spikeslabGAM paket III

```
mcmc <- list(nChains=4, chainLength=1000, burnin=500, thin=
m0 <- spikeSlabGAM(diabetes ~ pregnant + glucose + pressure
        family="binomial", data=pimaDiabTrain, mcmc=mcmc)
```

```
##
## Model has  56  coefficients in  13 model terms.
## Blockwise sampling: alpha: 3  block(s); xi: 5 block(s).
```

```
## Using 2 parallel processes.
## Use 'options(mc.cores = <YourNumberHere>)' to override r
```

spikeslabGAM paket IV

```
##
## starting chain(s):
## bbbb0--------------------------------------100%
##
## Mean acceptance rates:
## alpha   ksi
##  0.92  0.65
```

```r
print(summary(m0), printModels=FALSE)
```

```
## Spike-and-Slab STAR for Binomial data
##
## Model:
## diabetes ~ (lin(pregnant) + sm(pregnant)) + (lin(glucose
##     (lin(pressure) + sm(pressure)) + (lin(mass) + sm(mas
##     (lin(pedigree) + sm(pedigree)) + (lin(age) + sm(age)
## 524 observations; 56 coefficients in 13 model terms.
```

spikeslabGAM paket V

```
##
## Prior:
##  a[tau]  b[tau]    v[0]    a[w]    b[w]
## 5.0e+00 2.5e+01 2.5e-04 1.0e+00 1.0e+00
##
## MCMC:
## Saved 4000 samples from 4 chain(s), each ran 5000 iterat
##   burn-in of 500 ; Thinning: 5
## P-IWLS acceptance rates: 0.92 for alpha; 0.65 for xi.
##
## Null deviance:            679
## Mean posterior deviance: 467
##
## Marginal posterior inclusion probabilities and term impo
##               P(gamma = 1)    pi dim
## u                       NA    NA   1
## lin(pregnant)        0.263 0.027   1   *
```

spikeslabGAM paket VI

```
## sm(pregnant)           0.022 0.000   7
## lin(glucose)           1.000 0.533   1 ***
## sm(glucose)            0.015 0.000   9
## lin(pressure)          0.016 0.000   1
## sm(pressure)           0.015 0.000   9
## lin(mass)              1.000 0.186   1 ***
## sm(mass)               0.810 0.044   8  **
## lin(pedigree)          0.249 0.007   1
## sm(pedigree)           0.630 0.008   8  **
## lin(age)               0.754 0.092   1  **
## sm(age)                0.782 0.103   8  **
## *:P(gamma = 1)>.25 **:P(gamma = 1)>.5 ***:P(gamma = 1)>.
```