

Classification with Slice Sampling

Andreas Hofheinz

27 11 2020

```
library(bisemSliceSampler)
library(latex2exp)
library(patchwork)
library(ggplot2)

data(PimaIndiansDiabetes , package = "mlbench")
diabetis_sub <- subset(PimaIndiansDiabetes, select = -c(triceps, insulin))
diabetis_sub$diabetes <- as.numeric(diabetis_sub$diabetes) - 1

n = nrow(diabetis_sub)
trainIndex = sample(1:n, size = round(0.7*n), replace = FALSE)
train = diabetis_sub[trainIndex, ]
test = diabetis_sub[-trainIndex, ]

# Scaling speeds up convergence
X <- scale(as.matrix(subset(train, select = -diabetes)))
y <- train$diabetes
```

Model:

- Verteilungsannahme: $y_i \mid \mathbf{x}_i \stackrel{\text{unabh.}}{\sim} \text{Bin}(1, \pi_i)$
- Linearer Prädiktor: $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$
- Responsefunktion: $\mathbb{E}(y_i \mid x_i) = \pi_i = h(\eta_i) = \frac{\exp(\eta_i)}{1 + \exp(\eta_i)}$
- Features: $\mathbf{x}_i = (x_{\text{pregnant},i}, x_{\text{glucose},i}, x_{\text{pressure},i}, x_{\text{mass},i}, x_{\text{pedigree},i}, x_{\text{age},i})$

Likelihood:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^n (h(\mathbf{x}_i^\top \boldsymbol{\beta}))^{y_i} (1 - h(\mathbf{x}_i^\top \boldsymbol{\beta}))^{1-y_i}$$

Prior

In this example we specify a noninformative prior: $\boldsymbol{\beta} \sim N(\mathbf{0}, 1000 * I)$ such that $p(\boldsymbol{\beta}) \propto \exp\left(-\frac{1}{2*1000} \boldsymbol{\beta}^\top \boldsymbol{\beta}\right)$

Posterior

$$p(\boldsymbol{\beta} \mid \mathbf{y}) \propto \left[\prod_{i=1}^n (h(\mathbf{x}_i^\top \boldsymbol{\beta}))^{y_i} (1 - h(\mathbf{x}_i^\top \boldsymbol{\beta}))^{1-y_i} \right] * \exp\left(-\frac{1}{2000} \boldsymbol{\beta}^\top \boldsymbol{\beta}\right)$$

Log-Posterior

$$\log(p(\boldsymbol{\beta} \mid \mathbf{y})) \propto \left(\sum_{i=1}^n y_i \right) \log h(\mathbf{x}_i^T \boldsymbol{\beta}) + \left(n - \sum_{i=1}^n y_i \right) \log(1 - h(\mathbf{x}_i^T \boldsymbol{\beta})) - \frac{\boldsymbol{\beta}^\top \boldsymbol{\beta}}{2000}$$

```
h <- function(x) exp(x) / (1 + exp(x))

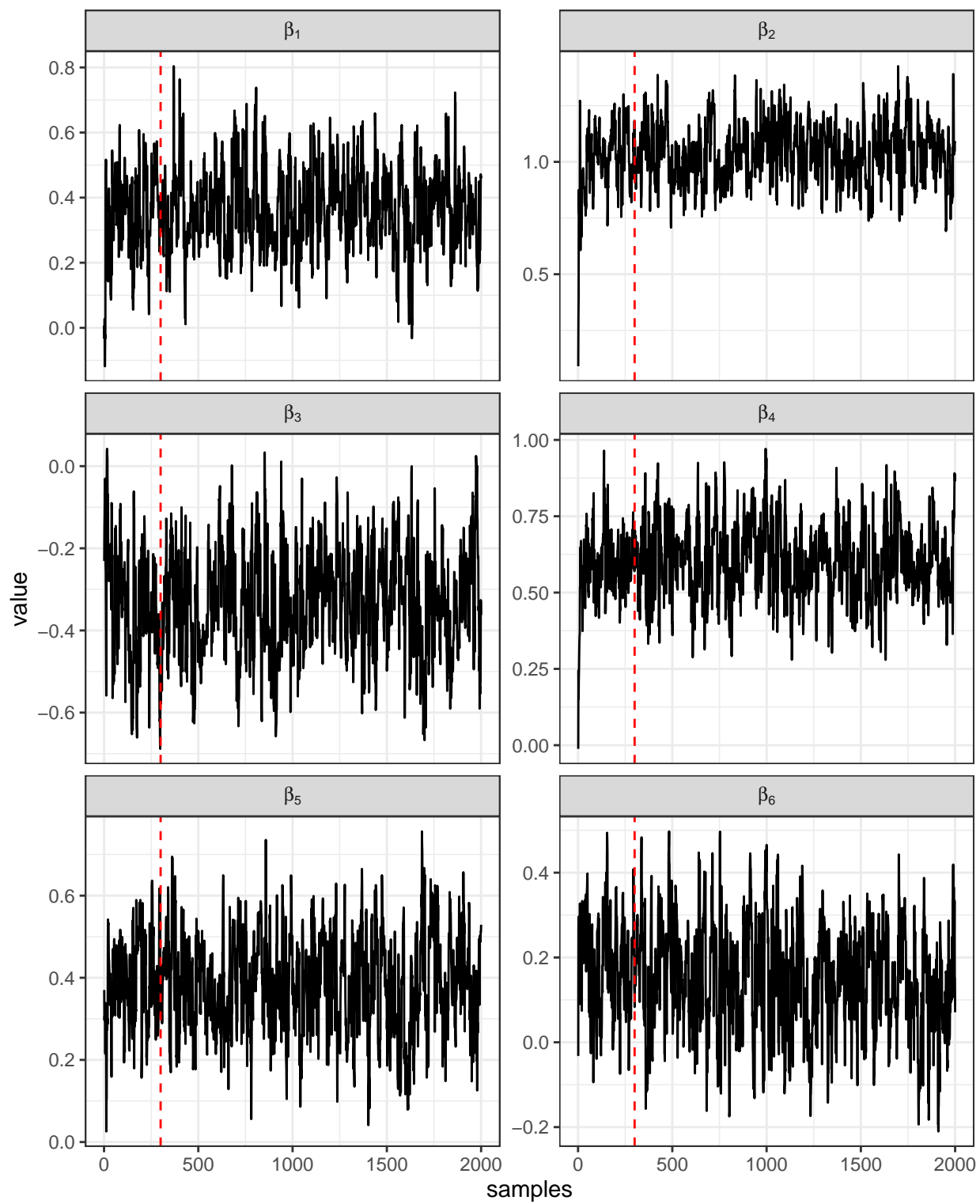
logpost <- function(beta) {
  pi <- h(X %*% beta)
  y %*% log(pi) + (1 - y) %*% log(1 - pi) - (1 / 2000) * crossprod(beta)
}
```

Sampling from Log-Posterior

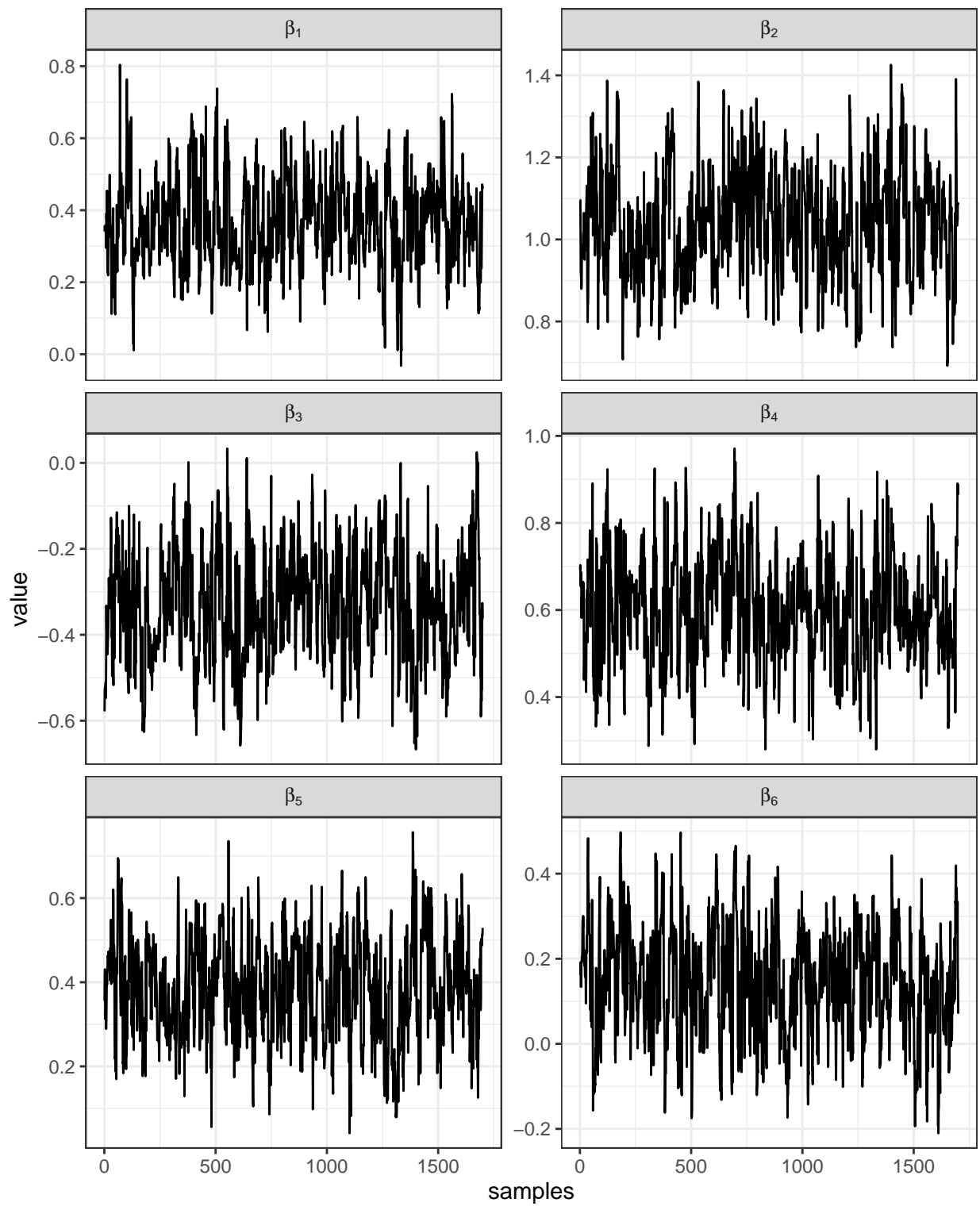
```
w <- rep(1, times = ncol(X))
n_samples <- 2000
beta_init <- rep(0, ncol(X))

beta_samples <- slice_sampler(
  logpost, x_init = beta_init, w = w, n_samples = n_samples
)
```

Convergence Diagnostic



Convergence Diagnostic - 300 burn-in samples removed



Coefficients and Credibility Intervals

	Mean	2.5%	97.5%
pregnant	0.367	0.129	0.623
glucose	1.036	0.798	1.300
pressure	-0.333	-0.581	-0.087
mass	0.602	0.364	0.848
pedigree	0.383	0.163	0.613
age	0.143	-0.104	0.396

Comparison with frequentist calculation

Expected to be similar since non-informative prior was chosen.

Waiting for profiling to be done...

	Mean	2.5%	97.5%
pregnant	0.363	0.122	0.611
glucose	1.018	0.771	1.281
pressure	-0.322	-0.569	-0.090
mass	0.590	0.362	0.834
pedigree	0.372	0.150	0.603
age	0.150	-0.095	0.397

Posterior predictive

Idea for presentation: - Maybe explain for one sample first - Add a bit more formulas

$$p(\tilde{y} | y) = \int p(\tilde{y} | \beta) p(\beta | y) d\beta$$

```
X_test <- scale(as.matrix(subset(test, select = -diabetes)))
```

```
# For each new obs we have 1700 eta (2000 - burn_in)
eta <- X_test %*% t(beta_samples_no_burnin)
pi <- h(eta)
```

```
# For each new obs i: for all (1700) pi_i : draw y ~ Bin(1, pi_i)
post_pred_y <- matrix(NA, nrow = nrow(X_test), ncol = ncol(pi))
for (i in 1:nrow(X_test)) {
  post_pred_y[i, ] <- rbinom(ncol(pi), 1, prob = pi[i, ])
}
```

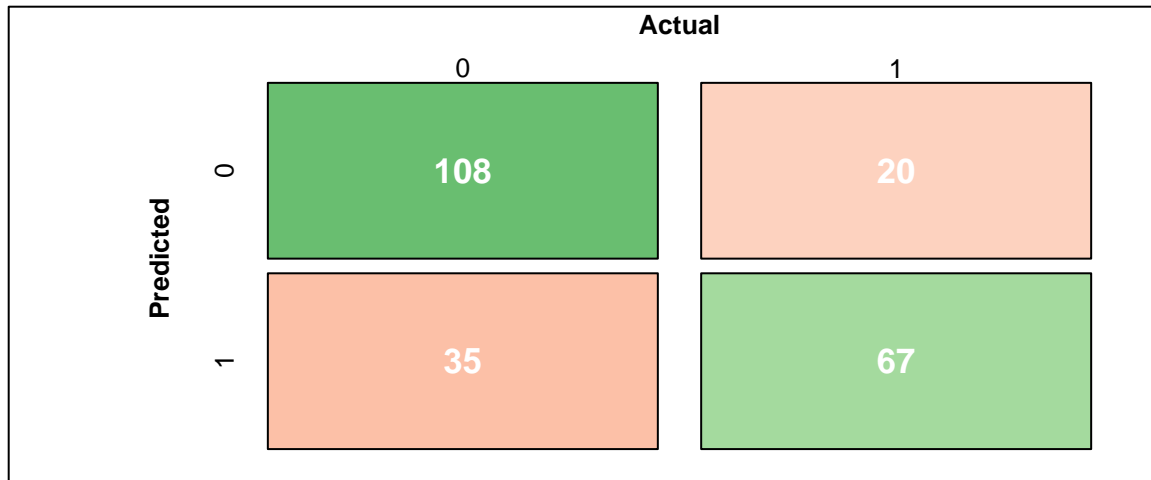
```
dim(post_pred_y)
```

```
## [1] 230 1700
```

Point estimator for \tilde{y}_i :

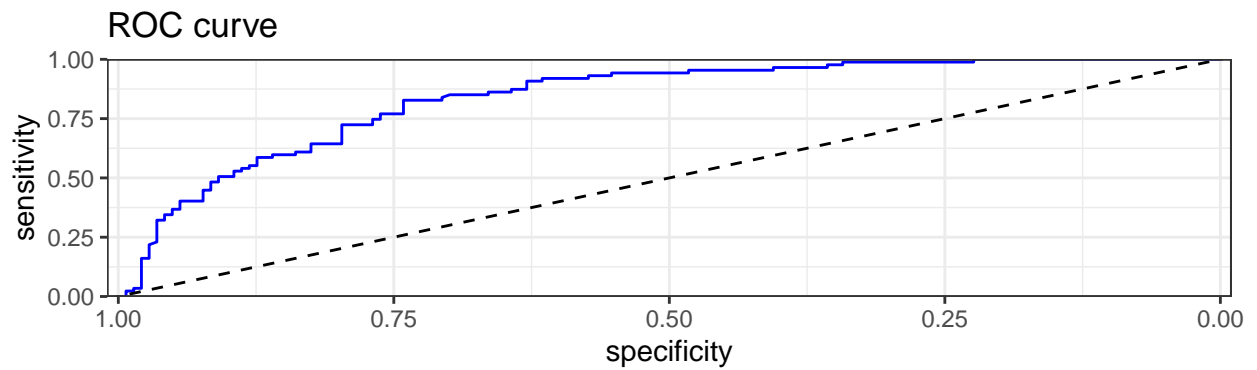
$$\hat{\tilde{y}}_i = \frac{1}{1700} * \sum_{k=1}^{1700} \tilde{y}_{i,k}$$

CONFUSION MATRIX

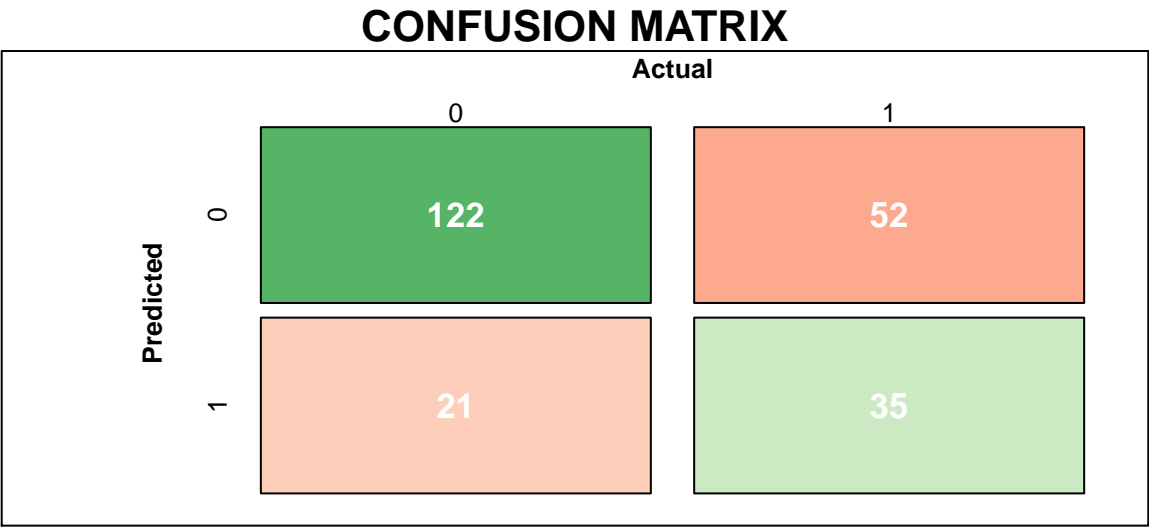


DETAILS

Sensitivity 0.755	Specificity 0.77	Precision 0.844	Recall 0.755	F1 0.797
Accuracy 0.761		Kappa 0.508		



Comparison with GLM



DETAILS				
Sensitivity	Specificity	Precision	Recall	F1
0.853	0.402	0.701	0.853	0.77
Accuracy		Kappa		
0.683		0.275		

