Alternatives to the MCMC Method; an Example with Real Data

by Leo Knüsel, University of Munich, Germany

Working Paper, April 2003

Abstract

In this paper we present an example with real data as a supplement to Knüsel (2003). We show that within a Bayesian model the posterior distribution that corresponds to a noninformative prior distribution can be found approximately using classical simulation methods if the number of independent variables is not too large. We work with the multivariate Normal, Laplace, Cauchy and Uniform distribution as prior distributions, and we use an iterative procedure that takes into account the approximate covariance matrix of the posterior distribution.

1. Summary

The example is taken from Fahrmeir-Tutz (2001), page 3: Credit-scoring. For $n = 1000$ bank customers the creditability y ($y = 0$: credit-worthy, $y = 1$: not credit worthy) and $k = 8$ explanatory variables $x_1, \ldots, x_k$ are given. We apply a probit model in a Bayesian framework, and we want to apply the simple classical acceptance-rejection algorithm (Algorithm A of Knüsel, 2003) to generate random data from the posterior distribution (section 2).

First we have to find the maximum likelihood estimator of the parameter vector $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)$ so that Algorithm A can work efficiently (section 3). Then we have to find a noninformative prior distribution; we call a prior distribution *noninfomative* if the posterior distribution does not change anymore when enlarging the dispersion of the prior. We face the problem that too few acceptable data are found from the posterior distribution if the prior distribution is too far away from the posterior distribution.

In section 4 we work with a $(k+1)$ -dimensional prior distribution with independent normal components $N(\mu_j, \sigma_\beta^2)$ where $\mu_j = \hat{\beta}_j$ is the maximum likelihood estimate of β_j . But we are not able to increase σ_β to a value such that the corresponding prior distribution could be considered as approximately noninformative as the number of acceptable data becomes too small for larger values of σ_β .

In section 5 we work with the multivariate normal distribution as prior distribution. In the first iteration step we compute the covariance matrix $\mathbf{C}_{\text{app}}$ found from the posterior data in section 4; this covariance matrix can be considered as an approximation to the unknown covariance matrix of the posterior distribution connected with the uninformative prior distribution. As prior distribution we now use the multivariate normal distribution $N_9(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} = \hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate of $\boldsymbol{\beta}$ and where $\boldsymbol{\Sigma} = r^2 \mathbf{C}_{\text{app}}$ with a factor $r > 1$ chosen as large as possible in order to come close to the noninformative prior distribution; we started with $r = 1.5$ and in a second series of simulations we chose to $r = 2$. Then we generate the corresponding posterior data. In the second iteration step we compute an updated covariance matrix $\mathbf{C}_{\text{app}}$ found from the posterior data in step 1. As prior distribution we now use the multivariate normal distribution $N_9(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = r^2 \mathbf{C}_{\text{app}}$ with the updated covariance matrix $\mathbf{C}_{\text{app}}$. We repeat this procedure until the correlation matrix and the eigenvalues of $\mathbf{C}_{\text{app}}$ become stable. But if we replace $r = 1.5$ by $r = 2$ the eigenvalues increase significantly and so we cannot consider our prior distribution as noninformative. We are again not able to increase *r* to value such that the corresponding prior distribution could be considered as approximately noninformative as the number of acceptable data becomes too small for larger values of *r*.

In section 6 we work with a $(k+1)$ -dimensional multivariate uniform prior distribution and perform the same iterative procedure as in section 4 with the multivariate normal distribution. We also tried out the multivariate Laplace (two-sided exponential distribution) and Cauchy distribution, but the best results were found with the multivariate uniform distribution.

In section 7 we compare the results found with the multivariate uniform distribution as prior distribution with the results found by the MCMC method with a diffuse prior. I thank Dr. Stefan Lang from the Department of Statistics, University of Munich, for performing the MCMC computations. Our results come rather close to the results of the MCMC method.

2 Data and model

The example is taken from Fahrmeir-Tutz (2001), page 3: Credit-scoring. The corresponding data set can be found under *www.stat.uni-muenchen.de/~lang/compstat20022003/compstat0203.html*. The data set gives for $n = 1000$ bank customers the creditability y and $k = 8$ explanatory variables $x_1, \ldots, x_k$:

y = creditability ($y = 0$: credit-worthy; $y = 1$: not credit worthy)
x_1 = duration of credit in months
x_2 = payment of previous credits (1: good; -1: bad)
x_3 = intended use of credit (1: private; -1: professional)
x_4 = amount of credit (in 1000 DM)
x_5 = gender (1: male; -1: female)
x_6 = marital status (1: married; -1: unmarried)
x_7 = covariable 1 (running account)
x_8 = covariable 2 (running account)

The creditability y of a customer is to be explained by the $k = 8$ explanatory variables $x_1, \ldots, x_k$. We want to apply a probit model in a Bayesian framework. Let $Y_1, \ldots, Y_n$ be independent random variables with

$$Y_i \,\big|\, \boldsymbol{\beta} \sim Bi\left(1, p_i\right), i = 1, \ldots, n;$$

$$p_i = E\left(Y_i \,|\, \boldsymbol{\beta}\right) = \Phi\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}\right).$$

Then

$$\Pr\left\{Y_i = 1\right\} = p_i \quad \text{and} \quad \Pr\left\{Y_i = 0\right\} = 1 - p_i$$

or

$$p(y_i \,|\, \boldsymbol{\beta}) = \Pr\left\{Y_i = y_i\right\} = p_i^{y_i}\left(1 - p_i\right)^{1-y_i}, y_i \in \{0,1\}$$

and

$$p\left(\mathbf{y} \,|\, \boldsymbol{\beta}\right) = \prod_{i=1}^{n} p\left(y_i \,|\, \boldsymbol{\beta}\right) = \prod_{i=1}^{n} p_i^{y_i}\left(1 - p_i\right)^{1-y_i}, y_i \in \{0,1\}.$$

Within the framework of our model the values $x_{ij}, i = 1, \ldots, n, \ j = 1, \ldots, k$ are given (k deterministic predictor variables), and in a first step we assume that $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)$ is a random vector with the following prior distribution:

$\beta_0, \beta_1, \ldots, \beta_k$ are independent random variables, where $\beta_j \sim N\left(\mu_j, \sigma_\beta^2\right), \ j = 1, \ldots, k$.

The parameters $\mu_0, \ldots, \mu_k, \sigma_\beta^2$ are assumed to be known; the density of the prior distribution of $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)$ is then given by

$$p\left(\boldsymbol{\beta}\right) = \left(\sigma_\beta \sqrt{2\pi}\right)^{-k} \exp\left(-\frac{1}{2\sigma_\beta^2} \sum_{j=0}^{k} \left(\beta_j - \mu_j\right)^2\right).$$

The posterior distribution of $\boldsymbol{\beta}$ for given observations $\mathbf{y} = \left(y_1, \ldots, y_n\right)$ has the density

$$p\left(\boldsymbol{\beta} \,|\, \mathbf{y}\right) \propto p\left(\boldsymbol{\beta}\right) \times p\left(\mathbf{y} \,|\, \boldsymbol{\beta}\right) = p\left(\boldsymbol{\beta}\right) \times \prod_{i=1}^{n} p_i^{y_i}\left(1 - p_i\right)^{1-y_i} .$$

As the random variables Y_i are discrete we obviously have $p\left(\mathbf{y} \,|\, \boldsymbol{\beta}\right) \leq 1$. Random data from $p\left(\boldsymbol{\beta}\right)$ $\left(= g\left(\boldsymbol{\beta}\right) = \tilde{g}\left(\boldsymbol{\beta}\right)\right)$, proposal distribution) are available, and random data from the posterior distribution $p(\boldsymbol{\beta} \,|\, \mathbf{y})$ $\left(= f\left(\boldsymbol{\beta}\right)\right)$, target distribution) have to be generated. The normalising constant of the posterior distribution is unknown:

$$p\left(\boldsymbol{\beta} \,|\, \mathbf{y}\right) = f\left(\boldsymbol{\beta}\right) = c_f \ p\left(\boldsymbol{\beta}\right) p\left(\mathbf{y} \,|\, \boldsymbol{\beta}\right) = c_f \tilde{f}\left(\boldsymbol{\beta}\right) \text{ with unknown } c_f .$$

But we have

$$\frac{\tilde f(\beta)}{\tilde g(\beta)} = \frac{\tilde f(\beta)}{g(\beta)} = \frac{p(\beta)\,p(\mathbf{y}|\beta)}{p(\beta)} = p(\mathbf{y}|\beta) \le 1,$$

and so Algorithm A works (in theory) with

$$\tilde f(\beta) = p(\beta)\,p(\mathbf{y}|\beta);\ \tilde g(\beta) = g(\beta) = p(\beta);\ c = 1.$$

In Table 2 we find the raw data of our example. We consider $n = 1000$ bank customers with the variables $y, \tilde x_1, \ldots, \tilde x_8$. Column 1 gives the observed values $y_1, \ldots, y_n$ and column 2 to 9 give the values $\tilde x_{ij}$ of the $k = 8$ explanatory variables $\tilde x_1, \ldots, \tilde x_8$. In order to avoid numerical problems we will work with normalised variables $x_1, \ldots, x_8$ where

(1) $$x_j = \frac{\tilde x_j - m_j}{2r_j} - \frac{1}{4},$$

where $m_j = \min_i \tilde x_{ij}$, $M_j = \max_i \tilde x_{ij}$, and $r_j = M_j - m_j$. The normalised data x_{ij} will all lie in the interval $[-\frac{1}{4}, \frac{1}{4}]$. The normalising constants are to be found in Table 1. In Table 3 we compute for a given vector $\beta = (\beta_0, \beta_1, \ldots, \beta_k)$ the quantity $\log p(\mathbf{y}|\beta)$. In column 10 and 11 we compute $z_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$ and $p_i = \Phi(z_i)$. Column 12 gives the probabilities $p(y_i|\beta) = p_i^{y_i}(1 - p_i)^{1-y_i}$, and column 13 gives the logarithms of these probabilities. Due to our normalisation the values p_i will most probably neither underflow to zero nor overflow to 1 so that the probabilities $p(y_i|\beta)$ will not become zero and the logarithm can be taken without error.

Table 1: Normalising constants

j	m_j	M_j	r_j
1	4	72	68
2	-1	1	2
3	-1	1	2
4	0.25	18.424	18.174
5	-1	1	2
6	-1	1	2
7	-1	1	2
8	-1	1	2

The sum of column 13 is

$$\log p(\mathbf{y}|\beta) = \sum_{i=1}^{n} \log p(y_i|\beta) = -700.1879$$

and $p(\mathbf{y}|\beta) = \exp(-700.19) = 0.82 \times 10^{-304}$. Such small probabilities can cause underflow problems and therefore we will work with logarithms and not with probabilities wherever possible.

Table 2: Raw data of our example with $n = 1000$ bank customers

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
i	y_i	$\tilde{x}_{i1}$	$\tilde{x}_{i2}$	$\tilde{x}_{i3}$	$\tilde{x}_{i4}$	$\tilde{x}_{i5}$	$\tilde{x}_{i6}$	$\tilde{x}_{i7}$	$\tilde{x}_{i8}$
1	0	24	1	1	2.957	-1	1	1	0
2	0	12	1	1	1.424	1	-1	-1	-1
3	0	18	1	1	2.051	-1	1	0	1
4	0	12	1	1	4.675	1	-1	0	1
5	0	24	1	-1	2.978	-1	1	0	1
6	0	24	1	1	2.603	1	-1	0	1
7	0	36	1	1	4.686	-1	1	0	1
8	0	24	1	1	2.835	-1	1	0	1
9	0	21	1	-1	1.602	1	1	1	0
10	0	10	1	1	2.848	-1	1	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
991	1	36	1	1	1.977	-1	1	1	0
992	1	18	1	-1	0.976	1	-1	1	0
993	1	48	1	-1	3.914	-1	-1	0	1
994	1	30	1	1	11.998	-1	-1	1	0
995	1	15	-1	-1	6.850	-1	1	-1	-1
996	1	48	1	-1	15.672	-1	1	-1	-1
997	1	15	1	1	2.327	1	-1	-1	-1
998	1	24	1	1	1.024	1	1	1	0
999	1	30	1	-1	1.908	-1	1	-1	-1
1000	1	60	1	-1	14.027	-1	1	-1	-1
sum	300								

Table 3: Computation of $\log p(\mathbf{y}|\boldsymbol{\beta})$ for $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_8)$ where

β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
-0.3079	-1.9600	-1.6480	-1.6173	0.5248	0.3849	-0.7045	-1.8751	-0.5106

(0)	(1)	(2)		(9)	(10)	(11)	(12)	(13)
i	y_i	x_{i1}	⋯	x_{i8}	z_i	p_i	$p(y_i \vert \boldsymbol{\beta})$	$\log(p(y_i \vert \boldsymbol{\beta}))$
1	0	-0.1029	⋯	0.00	-1.7557	0.0396	0.9604	-0.0404
2	0	-0.1912	⋯	-0.25	0.0050	0.5020	0.4980	-0.6971
3	0	-0.1471	⋯	0.25	-1.3411	0.0899	0.9101	-0.0942
4	0	-0.1912	⋯	0.25	-0.6721	0.2508	0.7492	-0.2887
5	0	-0.1029	⋯	0.25	-0.6056	0.2724	0.7276	-0.3180
6	0	-0.1029	⋯	0.25	-0.8750	0.1908	0.8092	-0.2117
7	0	-0.0147	⋯	0.25	-1.5625	0.0591	0.9409	-0.0609
8	0	-0.1029	⋯	0.25	-1.4163	0.0783	0.9217	-0.0816
9	0	-0.1250	⋯	0.00	-0.7309	0.2324	0.7676	-0.2645
10	0	-0.2059	⋯	0.25	-1.2143	0.1123	0.8877	-0.1191
⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋮
991	1	-0.0147	⋯	0.00	-1.9427	0.0260	0.0260	-3.6488
992	1	-0.1471	⋯	0.00	-0.3445	0.3652	0.3652	-1.0072
993	1	0.0735	⋯	0.25	-0.5857	0.2790	0.2790	-1.2764
994	1	-0.0588	⋯	0.00	-1.3593	0.0870	0.0870	-2.4416
995	1	-0.1691	⋯	-0.25	1.1280	0.8703	0.8703	-0.1389
996	1	0.0735	⋯	-0.25	-0.0442	0.4824	0.4824	-0.7290
997	1	-0.1691	⋯	-0.25	-0.0252	0.4900	0.4900	-0.7135
998	1	-0.1029	⋯	0.00	-1.5911	0.0558	0.0558	-2.8862
999	1	-0.0588	⋯	-0.25	0.0165	0.5066	0.5066	-0.6800
1000	1	0.1618	⋯	-0.25	-0.2408	0.4048	0.4048	-0.9043
sum								-700.1879

3. Maximum Likelihood Estimate

Now we try to apply Algorithm A (see Knüsel, 2003) to generate random data from the posterior distribution by means of random data from the prior distribution $p(\boldsymbol{\beta})$ with $\mu_0 = \cdots = \mu_8 = 0$ and $\sigma_\beta^2 = 1$. We again use Minitab with the seed 77 to generate $M = 100000$ data vectors $\boldsymbol{\beta}_i = (\beta_{i0}, \beta_{i1,} \ldots, \beta_{i8})$, $i = 1, \ldots, M$. These data are to be found in column 0 to 8 of Table 4a. Column 9 gives the logarithms $\log p(\mathbf{y}|\boldsymbol{\beta}_i)$ computed according to Table 3; the first line in Table 4a contains the result of Table 3; for $\boldsymbol{\beta}_1 = (-0.3079, \ldots, -0.5106)$ we have $\log p\left(\mathbf{y}|\boldsymbol{\beta}_1\right) = -700.19$. In column 10 we find the sorted values of column 9. The smallest and largest value in column 10 are $l_{\min} = -8808.60$ and $l_{\max} = -536.90$ so that the smallest and largest probability $p(\mathbf{y}|\boldsymbol{\beta}_i)$ are given by

$$p_{\min} = \exp(l_{\min}) = 0.296 \cdot 10^{-3825};$$

$$p_{\max} = \exp(l_{\max}) = 0.671 \cdot 10^{-233}.$$

If we tried to use Algorithm A with $c = 1$, the acceptance probabilities $p(\mathbf{y} | \boldsymbol{\beta}_i)$ would be so small that most probably not a single vector $\boldsymbol{\beta}_i$ could be accepted as a random vector of the posterior distribution. But from the sorted data in column 10 of Table 4a we see that for the given data points $(y_i, x_{i1}, \ldots, x_{i8}), i = 1, \ldots, n$, the conditional probability $p(\mathbf{y}|\boldsymbol{\beta})$ as a function of the parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_8)$ seems to have a maximum that will be close to $p_{\max} = \exp(l_{\max})$. The maximum will be reached for the maximum likelihood estimate $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_8)$. We obviously have $p(\mathbf{y} | \boldsymbol{\beta}) \le p(\mathbf{y} | \hat{\boldsymbol{\beta}})$ for all possible values of $\boldsymbol{\beta}$. So we can apply Algorithm A with $c = p(\mathbf{y} | \hat{\boldsymbol{\beta}})$ instead of $c = 1$, and thus we want to find the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$. In Table 4b we find the 10 data vectors $\boldsymbol{\beta}_i = (\beta_{i0}, \beta_{i1}, \ldots, \beta_{i8})$ of Table 4a with the largest values of $p(\mathbf{y}|\boldsymbol{\beta}_i)$. We see that $p(\mathbf{y}|\boldsymbol{\beta})$ becomes maximum for $\hat{\boldsymbol{\beta}} \approx (0.0, 0.7, -1.1, -0.3, 1.2, 0.3, 0.0, 1.0, 1.7)$, which is the mean of the 10 vectors in Table 4b rounded to 1 decimal place. As the maximum $l_{\max} = -536.90$ and the corresponding maximum likelihood estimate are not very accurate (look at the sorted values in column 10 of Table 4a) we want to improve the maximum likelihood estimate before we apply Algorithm A.

In our first simulation (Table 4a and Table 4b) the parameters of the prior distribution $p(\boldsymbol{\beta})$ were $\mu_0 = \cdots = \mu_8 = 0$ and $\sigma_\beta^2 = 1$. In the second simulation we choose $\sigma_\beta = \frac{1}{2}$ and $(\mu_0, \ldots, \mu_8) = \left(0.0, 0.7, \ldots, -1.7\right)$ which is our provisional estimate of $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_8)$ found in the first simulation. With this prior distribution we find the data in Table 5a and 5b. The maximum value of $\log p(\mathbf{y}|\boldsymbol{\beta}_i)$ has increased from $l_{\max} = -536.90$ in Table 4a to $l_{\max} = -514.24$ in Table 5a. So we can expect that the new estimate for $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_8)$ given by $(0.14, 1.40, \ldots, -2.24)$ will be better than the old one. This procedure is repeated until the estimate becomes stable. Our final estimate is found in Table 6a and Table 6b. The maximum of the log-likelihood function $\log p(\mathbf{y}|\boldsymbol{\beta}_i)$ is now $l_{\max} = -508.998$ and our maximum-likelihood estimate for $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_8)$ is given by the values in Table 6c; in the view of the author these estimates are correct with an absolute error $< \pm 0.01$. So we now know that for our data $(y_i, x_{i1}, \ldots, x_{i8}), i = 1, \ldots, n$ the upper limit of the log-likelihood $\log p(\mathbf{y}|\boldsymbol{\beta})$ is given by $l_{\max} = -508.998$, and now we can apply Algorithm A with the constant $c = \exp(l_{\max})$.

Table 4a: Simulation with $\beta_j \sim N(0,1),\ j=0,\ldots,8$ and $M=100\,000$

	(0)		(8)	(9)	(10)	(11)	(12)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	(9)$-l_{\max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$
1	-0.3079	$\cdots$	-0.5106	-700.19	-8808.60	-163.29	0.0000
2	-0.8409	$\cdots$	-0.8646	-840.11	-8533.03	-303.20	0.0000
3	2.5018	$\cdots$	-0.0407	-4689.85	-8426.73	-4152.94	0.0000
4	1.4639	$\cdots$	-0.4728	-2020.32	-7943.92	-1483.42	0.0000
5	-0.4848	$\cdots$	1.4662	-699.34	-7720.46	-162.44	0.0000
6	0.3611	$\cdots$	0.7853	-1125.67	-7709.91	-588.77	0.0000
7	-0.5657	$\cdots$	-0.9004	-653.03	-7384.15	-116.12	0.0000
8	0.5270	$\cdots$	-1.7131	-1009.61	-7269.52	-472.71	0.0000
9	1.0423	$\cdots$	0.4248	-1532.74	-7214.35	-995.84	0.0000
10	-0.9534	$\cdots$	0.6656	-781.71	-7114.27	-244.81	0.0000
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	1.2408	$\cdots$	-1.7120	-1861.72	-544.63	-1324.82	0.0000
99 992	-0.0757	$\cdots$	0.3897	-881.96	-544.60	-345.06	0.0000
99 993	-0.7856	$\cdots$	0.2108	-726.45	-544.28	-189.55	0.0000
99 994	-0.2130	$\cdots$	0.0309	-598.61	-543.57	-61.71	0.0000
99 995	-0.6739	$\cdots$	-0.8967	-789.97	-542.70	-253.07	0.0000
99 996	-0.5381	$\cdots$	1.9608	-761.53	-540.09	-224.63	0.0000
99 997	-0.1721	$\cdots$	-2.4754	-863.38	-539.28	-326.48	0.0000
99 998	0.9925	$\cdots$	0.2018	-2423.74	-538.96	-1886.84	0.0000
99 999	-1.6129	$\cdots$	-0.2222	-780.53	-538.76	-243.63	0.0000
100 000	1.2603	$\cdots$	-1.2523	-2482.00	-536.90	-1945.09	0.0000
sum							1.4238

$l_{\max} = -536.90$ (Maximum of column 9); $p_{\max} = \exp(l_{\max})$

Table 4b: The 10 data vectors $\boldsymbol{\beta}_i = (\beta_{i0},\ldots,\beta_{i8})$ of Table 4a with the largest values of $p(\mathbf{y}|\boldsymbol{\beta}_i)$

i	β_{i0}	β_{i1}	β_{i2}	β_{i3}	β_{i4}	β_{i5}	β_{i6}	β_{i7}	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$
6558	0.2640	0.1137	-1.7095	-0.8097	1.3562	0.7755	0.4383	0.9235	-1.5331	-544.63	0.0004
30324	-0.1328	1.4296	-0.6186	0.3987	1.4232	0.1062	-0.1876	0.9782	-1.4693	-544.60	0.0005
85500	0.0156	1.3439	-2.3623	0.0080	0.5519	-0.3711	-0.1158	1.3830	-1.9515	-544.28	0.0006
32528	0.0614	0.4902	-0.9626	-0.2633	1.6502	0.4529	0.2293	0.3928	-1.6524	-543.57	0.0013
41695	-0.3635	0.9383	-0.0743	-0.5906	0.9310	0.6163	0.0852	1.0256	-2.2649	-542.70	0.0030
66759	-0.2886	0.5887	-0.0343	-0.0445	0.3141	0.2990	-0.1270	1.5290	-2.0055	-540.09	0.0412
46451	0.0637	0.0547	-0.8878	-0.6488	2.0261	0.0793	-0.1535	0.6189	-1.4089	-539.28	0.0931
70833	0.0491	1.7412	-1.8137	-0.1716	0.0031	0.2713	-0.1071	0.5096	-1.0515	-538.96	0.1273
49598	-0.0687	-0.3137	-0.4785	-0.8969	2.2658	0.7387	0.5530	1.4297	-2.0962	-538.76	0.1559
19202	0.0985	0.6737	-1.7471	-0.2160	1.2136	-0.1218	-0.9260	0.7276	-1.2722	-536.90	1.0000
mean	-0.0301	0.7060	-1.0689	-0.3235	1.1735	0.2846	-0.0311	0.9518	-1.6705		

Table 5a: Simulation with $\sigma_\beta = \frac{1}{2}$, $\beta_j \sim N(\mu_j, \sigma_\beta^2)$, $j = 0,\ldots,8$ and $M = 100000$

j	0	1	2	3	4	5	6	7	8
μ_j	0.0	0.7	-1.1	-0.3	1.2	0.3	0.0	1.0	-1.7

	(0)		(8)	(9)	(10)	(11)	(12)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\vert\boldsymbol{\beta}_i)$	(9) sorted	(9)$-l_{\max}$	$\frac{p(\mathbf{y}\vert\boldsymbol{\beta}_i)}{p_{\max}}$
1	-0.1539	$\cdots$	-1.9553	-590.65	-1983.57	-76.41	0.0000
2	-0.4205	$\cdots$	-2.1323	-669.83	-1914.25	-155.59	0.0000
3	1.2509	$\cdots$	-1.7203	-1177.23	-1895.07	-662.99	0.0000
4	0.7319	$\cdots$	-1.9364	-696.55	-1843.49	-182.31	0.0000
5	-0.2424	$\cdots$	-0.9669	-576.89	-1784.88	-62.65	0.0000
6	0.1806	$\cdots$	-1.3074	-572.51	-1760.78	-58.27	0.0000
7	-0.2828	$\cdots$	-2.1502	-551.22	-1715.56	-36.98	0.0000
8	0.2635	$\cdots$	-2.5565	-561.93	-1696.18	-47.69	0.0000
9	0.5211	$\cdots$	-1.4876	-614.77	-1690.49	-100.53	0.0000
10	-0.4767	$\cdots$	-1.3672	-572.71	-1655.74	-58.47	0.0000
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	0.6204	$\cdots$	-2.5560	-692.20	-516.26	-177.96	0.0000
99 992	-0.0378	$\cdots$	-1.5051	-554.06	-516.13	-39.82	0.0000
99 993	-0.3928	$\cdots$	-1.5946	-620.54	-515.85	-106.30	0.0000
99 994	-0.1065	$\cdots$	-1.6845	-529.92	-515.61	-15.68	0.0000
99 995	-0.3370	$\cdots$	-2.1483	-655.14	-515.61	-140.90	0.0000
99 996	-0.2690	$\cdots$	-0.7196	-574.19	-515.47	-59.96	0.0000
99 997	-0.0861	$\cdots$	-2.9377	-583.45	-515.38	-69.21	0.0000
99 998	0.4962	$\cdots$	-1.5991	-764.41	-514.94	-250.17	0.0000
99 999	-0.8064	$\cdots$	-1.8111	-629.71	-514.85	-115.47	0.0000
100 000	0.6301	$\cdots$	-2.3262	-781.41	-514.24	-267.17	0.0000
sum							5.7054

$l_{\max} = -514.24$ (Maximum of column 9); $p_{\max} = \exp(l_{\max})$

Table 5b: The 10 data vectors $\boldsymbol{\beta}_i = (\beta_{i0},\ldots,\beta_{i8})$ of Table 5a with the largest values of $p(\mathbf{y}\vert\boldsymbol{\beta}_i)$

i	β_{i0}	β_{i1}	β_{i2}	β_{i3}	β_{i4}	β_{i5}	β_{i6}	β_{i7}	β_{i8}	$\ln p(\mathbf{y}\vert\boldsymbol{\beta}_i)$	$\frac{p(\mathbf{y}\vert\boldsymbol{\beta}_i)}{p_{\max}}$
5837	0.0579	1.6484	-1.3853	-0.4044	0.5597	0.4204	-0.0575	1.6405	-1.9325	-516.26	0.1329
80594	0.2312	1.7712	-1.2685	-0.3603	1.4566	0.5683	-0.0330	1.3211	-2.2235	-516.13	0.1513
48997	0.1770	1.0681	-1.4054	-0.6197	1.3122	0.1093	-0.2624	2.1728	-2.2593	-515.85	0.2001
24211	0.1709	1.4467	-1.2827	-0.4801	1.1979	0.1214	-0.6598	1.6117	-1.8036	-515.61	0.2540
79123	0.1695	1.0113	-1.3463	-0.7838	1.1988	0.0478	-0.6884	1.7812	-2.3318	-515.61	0.2542
16215	0.0612	1.2741	-0.9954	-0.3605	1.3543	0.0205	-0.2050	2.2081	-2.6671	-515.47	0.2931
97091	0.1236	0.9531	-0.7810	-0.4192	2.0337	0.3055	-0.3580	2.1528	-2.3815	-515.38	0.3181
34170	0.2024	1.6183	-1.0916	-0.4315	1.4545	0.7944	-0.3328	1.9082	-2.3583	-514.94	0.4942
59328	0.2241	1.6693	-1.3258	-0.4010	1.3221	0.2162	-0.2452	1.3798	-2.0192	-514.85	0.5412
53942	0.0286	1.4939	-0.8164	-0.7252	0.9306	0.4360	-0.0962	1.8855	-2.4385	-514.24	1.0000
mean	0.1446	1.3955	-1.1698	-0.4986	1.2820	0.3040	-0.2938	1.8062	-2.2415		

Table 6a: Simulation with $\sigma_\beta = 0.01$, $\beta_j \sim N(\mu_j, \sigma_\beta^2)$, $j = 0,\ldots,8$ and $M = 100000$

j	0	1	2	3	4	5	6	7	8
μ_j	0.232	2.785	-1.166	-0.558	0.685	0.264	-0.441	2.025	-2.505

	(0)		(8)	(9)	(10)	(11)	(12)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	$(9) - l_{\max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$
1	0.2289	$\cdots$	-2.5101	-509.024	-509.480	-0.026	0.9739
2	0.2236	$\cdots$	-2.5136	-509.059	-509.472	-0.061	0.9405
3	0.2570	$\cdots$	-2.5054	-509.215	-509.455	-0.217	0.8047
4	0.2466	$\cdots$	-2.5097	-509.063	-509.454	-0.065	0.9371
5	0.2272	$\cdots$	-2.4903	-509.015	-509.451	-0.017	0.9835
6	0.2356	$\cdots$	-2.4971	-509.010	-509.449	-0.012	0.9882
7	0.2263	$\cdots$	-2.5140	-509.008	-509.446	-0.010	0.9903
8	0.2373	$\cdots$	-2.5221	-509.019	-509.443	-0.021	0.9794
9	0.2424	$\cdots$	-2.5008	-509.034	-509.438	-0.036	0.9647
10	0.2225	$\cdots$	-2.4983	-509.012	-509.431	-0.014	0.9860
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	0.2444	$\cdots$	-2.5221	-509.074	-508.999	-0.076	0.9268
99 992	0.2312	$\cdots$	-2.5011	-509.008	-508.998	-0.010	0.9901
99 993	0.2241	$\cdots$	-2.5029	-509.038	-508.998	-0.040	0.9612
99 994	0.2299	$\cdots$	-2.5047	-509.005	-508.998	-0.007	0.9928
99 995	0.2253	$\cdots$	-2.5140	-509.050	-508.998	-0.052	0.9494
99 996	0.2266	$\cdots$	-2.4854	-509.013	-508.998	-0.015	0.9855
99 997	0.2303	$\cdots$	-2.5298	-509.030	-508.998	-0.032	0.9687
99 998	0.2419	$\cdots$	-2.5030	-509.074	-508.998	-0.075	0.9273
99 999	0.2159	$\cdots$	-2.5072	-509.037	-508.998	-0.039	0.9615
100 000	0.2446	$\cdots$	-2.5175	-509.087	-508.998	-0.089	0.9148
sum							96669.9

$l_{\max} = -508.998$ (Maximum of column 9); $p_{\max} = \exp(l_{\max})$

Table 6b: The 10 data vectors $\boldsymbol{\beta}_i = (\beta_{i0},\ldots,\beta_{i8})$ of Table 6a with the largest values of $p(\mathbf{y}|\boldsymbol{\beta}_i)$

i	β_{i0}	β_{i1}	β_{i2}	β_{i3}	β_{i4}	β_{i5}	β_{i6}	β_{i7}	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$
39188	0.2335	2.8007	-1.1664	-0.5527	0.6771	0.2635	-0.4479	2.0306	-2.5038	-508.999	0.9995
58201	0.2319	2.7929	-1.1647	-0.5529	0.6829	0.2701	-0.4353	2.0306	-2.5075	-508.998	0.9996
93483	0.2327	2.7915	-1.1700	-0.5552	0.6730	0.2621	-0.4405	2.0301	-2.5033	-508.998	0.9996
64494	0.2291	2.7896	-1.1657	-0.5555	0.6705	0.2634	-0.4365	2.0305	-2.5048	-508.998	0.9996
4006	0.2326	2.7942	-1.1724	-0.5584	0.6729	0.2589	-0.4480	2.0322	-2.5072	-508.998	0.9997
98968	0.2306	2.7930	-1.1728	-0.5554	0.6664	0.2662	-0.4373	2.0290	-2.5050	-508.998	0.9998
41525	0.2311	2.7963	-1.1636	-0.5589	0.6763	0.2587	-0.4450	2.0299	-2.5036	-508.998	0.9998
22361	0.2316	2.7901	-1.1651	-0.5532	0.6849	0.2679	-0.4381	2.0306	-2.5043	-508.998	0.9998
81937	0.2324	2.7874	-1.1702	-0.5535	0.6839	0.2640	-0.4446	2.0307	-2.5052	-508.998	0.9999
79658	0.2319	2.7874	-1.1660	-0.5551	0.6830	0.2621	-0.4410	2.0286	-2.5045	-508.998	1.0000
mean	0.2317	2.7923	-1.1677	-0.5551	0.6771	0.2637	-0.4414	2.0303	-2.5049		

Table 6c: Maximum likelihood estimate $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0,\ldots,\hat{\beta}_8)$ from Table 6b

j	0	1	2	3	4	5	6	7	8
$\hat{\beta}_j$	0.232	2.792	-1.168	-0.555	0.677	0.264	-0.441	2.030	-2.505

4. Application of Algorithm A with normal prior distribution

We now know that

$$\frac{\tilde{f}(\boldsymbol{\beta})}{\tilde{g}(\boldsymbol{\beta})} = \frac{\tilde{f}(\boldsymbol{\beta})}{g(\boldsymbol{\beta})} = \frac{p(\boldsymbol{\beta})\, p(\mathbf{y}|\boldsymbol{\beta})}{p(\boldsymbol{\beta})} = p(\mathbf{y}|\boldsymbol{\beta}) \le c = p_{\max} = \exp(l_{\max}),$$

where $l_{\max} = -508.998$, and we also know that $p(\mathbf{y}|\boldsymbol{\beta})$ becomes maximal in the neighbourhood of the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ (see Table 6c). So we choose in a first try the following proposal distribution $p(\boldsymbol{\beta})$: $\beta_0,\ldots,\beta_8$ are independent random variables with $\beta_j \sim N(\mu_j, \sigma_\beta^2)$, $j = 0,\ldots,8$, where $(\mu_0,\ldots,\mu_8) = (\hat{\beta}_0,\ldots,\hat{\beta}_8)$ and $\sigma_\beta = 0.2$. The results are to be found in Table 7a, which has the same form as Table 4a; the starting value of the random number generator is again 77. In column 12 we find the acceptance probabilities

$$\frac{1}{c}\, p(\mathbf{y}|\boldsymbol{\beta}_i) = \frac{p(\mathbf{y}|\boldsymbol{\beta}_i)}{p_{\max}} = \exp\left(\ln p(\mathbf{y}|\boldsymbol{\beta}_i) - l_{\max}\right);$$

so these probabilities are computed simply by exponentiation of the values in column 11. In column 13 we give M random data $u_1,\ldots,u_M$ from $U(0,1)$ and if $u_i < p(\mathbf{y}|\boldsymbol{\beta}_i)/c$ then the corresponding data vector $\boldsymbol{\beta}_i = (\beta_{i0}, \beta_{i1},\ldots,\beta_{i8})$ is accepted as a random realisation of the posterior distribution; this is indicated in column 14. So the total number of accepted data vectors is $M_{\text{acc}} = 3024$ (sum of column 14).

In a second try we choose a proposal distribution $p(\boldsymbol{\beta})$ that is characterised by the parameters $\sigma_\beta = 0.25$ and $(\mu_0,\ldots,\mu_8) = (\hat{\beta}_0,\ldots,\hat{\beta}_8)$. The results are to be found in Table 7b. The total number of accepted vectors has now decreased from 3024 to only 1195. The results of a third try with $\sigma_\beta = 0.3$ and $(\mu_0,\ldots,\mu_8) = (\hat{\beta}_0,\ldots,\hat{\beta}_8)$ are to be found in Table 7c. The number of accepted data vectors is now reduced to 481. For $\sigma_\beta = 0.5$ we would find only 25 acceptable data vectors out of the $M = 100\,000$ vectors generated from the prior distribution.

Bayesian statisticians are often interested in a *noninformative prior distribution* to avoid that a subjective choice of the parameters of the prior distribution influences the posterior distribution. In our case the so-called diffuse prior corresponding to Lebesque measure can be considered as a noninformative prior distribution. Within our framework we cannot generate diffuse random data, but we can approximate the diffuse prior by increasing the standard deviation σ_β of our prior distribution. If we could find a value σ_0 such that the posterior distribution does not change anymore if the standard deviation σ_β of the prior distribution becomes larger than σ_0, then such a prior could be considered as noninformative for the data at hand. Now the question arises whether the posterior distribution in our three simulations (see Table 7a,7b,7c) still depends on the parameter σ_β of the prior distribution. To answer this question we consider the covariance matrix of the generated data vectors from the posterior distribution and compute its eigenvalues. Note that the sum of the eigenvalues corresponds to the sum of the posterior variances of $\beta_0,\ldots,\beta_8$. If the covariance matrix does not change anymore then also the eigenvalues must become stable. Table 7d gives the eigenvalues for the three simulations in Table 7a, 7b, and 7c. We can see that our posterior distributions heavily depend on the prior distributions, as the eigenvalues clearly increase for

Table 7a: Simulation with $\sigma_\beta = 0.2$, $\beta_j \sim N(\hat{\beta}_j, \sigma_\beta^2)$, $j = 0,\ldots,8$ and $M = 100000$; $l_{\max} = -508.998$

	(0)		(8)	(9)	(10)	(11)	(12)	(13)	(14)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	(9)$-l_{\max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$	u_i	accept?
1	0.1704	$\cdots$	-2.6071	-517.697	-714.845	-8.699	0.0002	0.8823	0
2	0.0638	$\cdots$	-2.6779	-532.952	-700.872	-23.954	0.0000	0.9751	0
3	0.7324	$\cdots$	-2.5131	-599.024	-700.541	-90.026	0.0000	0.3074	0
4	0.5248	$\cdots$	-2.5996	-535.178	-698.325	-26.180	0.0000	0.4620	0
5	0.1350	$\cdots$	-2.2118	-516.496	-687.393	-7.498	0.0006	0.4663	0
6	0.3042	$\cdots$	-2.3479	-513.687	-685.686	-4.689	0.0092	0.4623	0
7	0.1189	$\cdots$	-2.6851	-511.781	-683.060	-2.783	0.0619	0.0598	1
8	0.3374	$\cdots$	-2.8476	-516.287	-681.569	-7.289	0.0007	0.7843	0
9	0.4405	$\cdots$	-2.4200	-524.081	-680.547	-15.083	0.0000	0.9679	0
10	0.0413	$\cdots$	-2.3719	-514.469	-679.669	-5.471	0.0042	0.6104	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	0.4802	$\cdots$	-2.8474	-539.040	-509.267	-30.042	0.0000	0.5323	0
99 992	0.2169	$\cdots$	-2.4271	-513.015	-509.264	-4.017	0.0180	0.1487	0
99 993	0.0749	$\cdots$	-2.4628	-524.420	-509.261	-15.422	0.0000	0.2785	0
99 994	0.1894	$\cdots$	-2.4988	-512.546	-509.258	-3.548	0.0288	0.8768	0
99 995	0.0972	$\cdots$	-2.6843	-527.850	-509.248	-18.852	0.0000	0.0173	0
99 996	0.1244	$\cdots$	-2.1128	-515.131	-509.225	-6.133	0.0022	0.7312	0
99 997	0.1976	$\cdots$	-3.0001	-519.377	-509.221	-10.379	0.0000	0.1777	0
99 998	0.4305	$\cdots$	-2.4646	-539.985	-509.213	-30.987	0.0000	0.3573	0
99 999	-0.0906	$\cdots$	-2.5494	-524.212	-509.167	-15.214	0.0000	0.1206	0
100 000	0.4841	$\cdots$	-2.7555	-545.342	-509.140	-36.344	0.0000	0.5435	0
sum							2984.9		3024

Table 7b: Simulation with $\sigma_\beta = 0.25$, $\beta_j \sim N(\hat{\beta}_j, \sigma_\beta^2)$, $j = 0,\ldots,8$ and $M = 100000$; $l_{\max} = -508.998$

	(0)		(8)	(9)	(10)	(11)	(12)	(13)	(14)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	(9)$-l_{\max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$	u_i	accept?
1	0.1550	$\cdots$	-2.6326	-522.516	-835.710	-13.518	0.0000	0.8823	0
2	0.0218	$\cdots$	-2.7211	-545.976	-814.208	-36.978	0.0000	0.9751	0
3	0.8574	$\cdots$	-2.5152	-651.474	-813.338	-142.476	0.0000	0.3074	0
4	0.5980	$\cdots$	-2.6232	-550.114	-808.670	-41.116	0.0000	0.4620	0
5	0.1108	$\cdots$	-2.1385	-520.671	-791.586	-11.673	0.0000	0.4663	0
6	0.3223	$\cdots$	-2.3087	-516.363	-785.131	-7.365	0.0006	0.4623	0
7	0.0906	$\cdots$	-2.7301	-513.335	-781.869	-4.337	0.0131	0.0598	0
8	0.3638	$\cdots$	-2.9333	-520.394	-779.137	-11.396	0.0000	0.7843	0
9	0.4926	$\cdots$	-2.3988	-532.652	-774.553	-23.654	0.0000	0.9679	0
10	-0.0063	$\cdots$	-2.3386	-517.542	-771.760	-8.544	0.0002	0.6104	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	0.5422	$\cdots$	-2.9330	-556.053	-509.417	-47.055	0.0000	0.5323	0
99 992	0.2131	$\cdots$	-2.4076	-515.297	-509.414	-6.299	0.0018	0.1487	0
99 993	0.0356	$\cdots$	-2.4523	-532.912	-509.408	-23.914	0.0000	0.2785	0
99 994	0.1788	$\cdots$	-2.4973	-514.519	-509.405	-5.521	0.0040	0.8768	0
99 995	0.0635	$\cdots$	-2.7292	-538.192	-509.389	-29.194	0.0000	0.0173	0
99 996	0.0975	$\cdots$	-2.0148	-518.558	-509.353	-9.560	0.0001	0.7312	0
99 997	0.1890	$\cdots$	-3.1239	-525.170	-509.348	-16.172	0.0000	0.1777	0
99 998	0.4801	$\cdots$	-2.4545	-557.963	-509.334	-48.965	0.0000	0.3573	0
99 999	-0.1712	$\cdots$	-2.5606	-532.656	-509.263	-23.658	0.0000	0.1206	0
100 000	0.5471	$\cdots$	-2.8181	-566.237	-509.220	-57.239	0.0000	0.5435	0
sum							1185.4		1195

Table 7c: Simulation with $\sigma_\beta = 0.3$, $\beta_j \sim N(\hat{\beta}_j, \sigma_\beta^2)$, $j = 0,\ldots,8$ and $M = 100000$; $l_{\max} = -508.998$

	(0)		(8)	(9)	(10)	(11)	(12)	(13)	(14)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	(9)$- l_{\max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{\max}}$	u_i	accept?
1	0.1396	$\cdots$	-2.6582	-528.355	-986.248	-19.357	0.0000	0.8823	0
2	-0.0203	$\cdots$	-2.7644	-561.596	-955.790	-52.598	0.0000	0.9751	0
3	0.9825	$\cdots$	-2.5172	-716.680	-954.038	-207.682	0.0000	0.3074	0
4	0.6712	$\cdots$	-2.6468	-568.507	-945.591	-59.509	0.0000	0.4620	0
5	0.0866	$\cdots$	-2.0651	-525.741	-921.070	-16.743	0.0000	0.4663	0
6	0.3403	$\cdots$	-2.2694	-519.662	-912.248	-10.664	0.0000	0.4623	0
7	0.0623	$\cdots$	-2.7751	-515.225	-906.243	-6.227	0.0020	0.0598	0
8	0.3901	$\cdots$	-3.0189	-525.423	-895.672	-16.425	0.0000	0.7843	0
9	0.5447	$\cdots$	-2.3776	-543.187	-889.741	-34.189	0.0000	0.9679	0
10	-0.0540	$\cdots$	-2.3053	-521.289	-888.947	-12.291	0.0000	0.6104	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	-3.0186	$\cdots$	-3.0186	-576.907	-509.600	-67.909	0.0000	0.5323	0
99 992	-2.3881	$\cdots$	-2.3881	-518.101	-509.598	-9.103	0.0001	0.1487	0
99 993	-2.4417	$\cdots$	-2.4417	-543.165	-509.587	-34.167	0.0000	0.2785	0
99 994	-2.4957	$\cdots$	-2.4957	-516.911	-509.585	-7.913	0.0004	0.8768	0
99 995	-2.7740	$\cdots$	-2.7740	-550.657	-509.562	-41.659	0.0000	0.0173	0
99 996	-1.9168	$\cdots$	-1.9168	-522.727	-509.510	-13.729	0.0000	0.7312	0
99 997	-3.2476	$\cdots$	-3.2476	-532.217	-509.503	-23.219	0.0000	0.1777	0
99 998	-2.4444	$\cdots$	-2.4444	-580.296	-509.481	-71.298	0.0000	0.3573	0
99 999	-2.5717	$\cdots$	-2.5717	-542.892	-509.380	-33.894	0.0000	0.1206	0
100 000	-2.8807	$\cdots$	-2.8807	-592.061	-509.318	-83.063	0.0000	0.5435	0
sum							493.3		481

Table 7d: Eigenvalues of covariance matrices of posterior data in Table 7a, 7b, 7c

σ_β	M_{acc}	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	sum
0.20	3024	0.039	0.033	0.030	0.029	0.026	0.019	0.016	0.013	0.002	0.207
0.25	1195	0.061	0.048	0.041	0.040	0.034	0.024	0.019	0.016	0.002	0.285
0.30	481	0.086	0.055	0.054	0.050	0.042	0.028	0.022	0.017	0.002	0.356

increasing standard deviations of the prior distribution. One could try to increase σ_β and M in order to get closer to the noninformative prior, but in the next section we will apply a more efficient method.

5. Multivariate normal distribution as prior distribution

If we look at the eigenvalues of the posterior data in Table 7c (see Table 7d) we see that $\lambda_{max}/\lambda_{min} = 0.086/0.002 \approx 40$. This shows that it is no good idea to use a prior distribution for $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_8)$ where all components β_j have independent normal distributions $N(\mu_j, \sigma_\beta^2)$ with the same standard deviation σ_β. The proposal (prior) distribution is quite different from the target (posterior) distribution and so too many proposed data vectors are lost. In this section we will apply an iterative procedure where the prior distribution takes into account an approximate covariance structure of the posterior distribution. Our procedure works as follows:

1. Determine a first approximation $\mathbf{C}_{app}$ of the covariance matrix $\mathbf{C} = (9 \times 9)$ of the posterior distribution. In our example the covariance matrix of the 481 data vectors in Table 7c will be used as such an approximation $\mathbf{C}_{app}$.
2. Generate $M = 100\,000$ random vectors from the multivariate normal distribution $N_9(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = \hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate for $\boldsymbol{\beta}$ (see section 3) and where $\boldsymbol{\Sigma} = r^2 \mathbf{C}_{app}$ with a factor $r > 1$ chosen as large as possible to come close to the noninformative prior.
3. As the acceptance probability for a data vector $\boldsymbol{\beta}_i = (\beta_{i0}, \ldots, \beta_{i8})$ is given by $p(\mathbf{y} \mid \boldsymbol{\beta}_i)/c = p(\mathbf{y} \mid \boldsymbol{\beta}_i)/p_{max}$, we can proceed as in section 4 to compute the acceptable data vectors from the posterior distribution.
4. Compute the covariance matrix of the accepted data vectors from the posterior distribution and denote it by $\mathbf{C}_{app}$. Continue with step 2 until stabilisation of $\mathbf{C}_{app}$.

How can we generate $M = 100\,000$ random vectors from a multivariate normal distribution $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$?

1. Determine the decomposition $\boldsymbol{\Sigma} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\mathsf{T}$ where $\mathbf{R}$ is orthogonal and $\boldsymbol{\Lambda}$ is diagonal. Compute $\mathbf{A} = \mathbf{R}\boldsymbol{\Lambda}^{1/2}\mathbf{R}^\mathsf{T}$. The matrix $\mathbf{A}$ is symmetric and $\mathbf{A}^2 = \boldsymbol{\Sigma}$, i.e. $\mathbf{A}$ is the symmetric root of $\boldsymbol{\Sigma}$. If $\mathbf{x} = (p \times 1)$ is a random vector with independent standard normal components then $\mathbf{y} = \mathbf{A}\mathbf{x}$ has a multivariate normal distribution $N_p(\mathbf{0}, \boldsymbol{\Sigma})$ as $\mathbf{A}\mathbf{A}^\mathsf{T} = \mathbf{A}^2 = \boldsymbol{\Sigma}$.
2. Generate $M = 100\,000$ random vectors from $N_p(\mathbf{0}, \mathbf{I})$, and arrange these data in a matrix $\mathbf{X} = (M \times p)$. Let $\mathbf{Y} = \mathbf{X}\mathbf{A}$. Now the rows of $\mathbf{Y}$ are independent random vectors from $N_p(\mathbf{0}, \boldsymbol{\Sigma})$.
3. Add μ_j to column *j* of the matrix **Y**; we denote the resulting matrix by $\mathbf{Z} = (M \times p)$; its rows are independent random vectors from $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Table 8a gives the results of the first iteration. We start with the covariance matrix $\mathbf{C}_{app}$ computed from the 481 accepted data vectors in Table 7c. Column (1) to (8) in Table 8a give the $M = 100\,000$ data vectors from $N_9(\hat{\boldsymbol{\beta}}, r^2\mathbf{C}_{app})$ with $r = 1.5$. Column (9) to (14) are computed exactly the same way as in section 4 (see Table 7a). We find $M_{acc} = 2954$ acceptable data vectors from the posterior distribution. We compute the new covariance matrix $\mathbf{C}_{app}$ from these data vectors, list its eigenvalues in Table 8b (first line) and proceed with the next iteration. In Table 8b we can see that the sum of the eigenvalues increases and tends to a limit, but we have to ask ourselves whether the factor $r = 1.5$ is large enough so that we can hope that the prior $N_9(\hat{\boldsymbol{\beta}}, r^2\mathbf{C}_{app})$ in the last iteration is close enough to the noninformative prior.

Table 8a: $M = 100000$ data vectors $\boldsymbol{\beta}_i = (\beta_{0j}, \ldots, \beta_{8j})$ from $N_k(\hat{\boldsymbol{\beta}}, r^2 \mathbf{C}_{app})$ with $r = 1.5$.

	(0)		(8)	(9)	(10)	(11)	(12)	(13)	(14)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	(9)$- l_{max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{max}}$	u_i	accept?
1	0.2704	$\cdots$	-2.5275	-516.545	-540.304	-7.547	0.0005	0.8823	0
2	0.3078	$\cdots$	-2.7263	-516.266	-538.496	-7.268	0.0007	0.9751	0
3	0.3447	$\cdots$	-2.4607	-520.161	-535.578	-11.163	0.0000	0.3074	0
4	0.3389	$\cdots$	-2.7011	-513.695	-535.149	-4.697	0.0091	0.4620	0
5	0.2644	$\cdots$	-2.1079	-511.975	-535.130	-2.977	0.0509	0.4663	0
6	0.2137	$\cdots$	-2.2397	-512.576	-534.660	-3.578	0.0279	0.4623	0
7	0.1144	$\cdots$	-2.7074	-511.605	-534.513	-2.607	0.0738	0.0598	1
8	0.2220	$\cdots$	-3.0228	-514.270	-533.903	-5.272	0.0051	0.7843	0
9	0.3069	$\cdots$	-2.4954	-514.950	-533.826	-5.952	0.0026	0.9679	0
10	0.0300	$\cdots$	-2.3465	-514.143	-533.557	-5.145	0.0058	0.6104	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
99 991	0.2436	$\cdots$	-3.0347	-521.267	-509.357	-12.269	0.0000	0.5323	0
99 992	0.1631	$\cdots$	-2.3553	-513.576	-509.355	-4.578	0.0103	0.1487	0
99 993	0.2727	$\cdots$	-2.4463	-512.655	-509.340	-3.657	0.0258	0.2785	0
99 994	0.2484	$\cdots$	-2.5816	-512.855	-509.334	-3.857	0.0211	0.8768	0
99 995	0.3255	$\cdots$	-2.6882	-516.716	-509.332	-7.718	0.0004	0.0173	0
99 996	0.2036	$\cdots$	-1.9601	-514.283	-509.324	-5.285	0.0051	0.7312	0
99 997	0.1797	$\cdots$	-3.2001	-525.273	-509.293	-16.275	0.0000	0.1777	0
99 998	0.1778	$\cdots$	-2.4521	-516.243	-509.286	-7.245	0.0007	0.3573	0
99 999	-0.0199	$\cdots$	-2.5560	-513.349	-509.199	-4.351	0.0129	0.1206	0
100 000	0.2118	$\cdots$	-2.8880	-516.771	-509.198	-7.773	0.0004	0.5435	0
sum							2865.2		2954

Table 8b: Eigenvalues of the covariance matrix of the posterior data of 11 iterations with $r = 1.5$ and 4 iterations with $r = 2$

	M	r	M_{acc}	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	sum
1	100 000	1.5	2954	0.153	0.074	0.059	0.056	0.045	0.024	0.019	0.013	0.001	0.444
2	100 000	1.5	3112	0.234	0.086	0.062	0.060	0.046	0.022	0.017	0.012	0.001	0.541
3	100 000	1.5	2962	0.306	0.095	0.063	0.060	0.047	0.021	0.017	0.011	0.001	0.623
4	100 000	1.5	2785	0.354	0.100	0.063	0.061	0.048	0.021	0.016	0.011	0.001	0.674
5	100 000	1.5	2707	0.371	0.102	0.064	0.061	0.048	0.021	0.016	0.011	0.001	0.695
6	100 000	1.5	2643	0.381	0.103	0.063	0.061	0.048	0.021	0.016	0.011	0.001	0.705
7	100 000	1.5	2638	0.385	0.104	0.064	0.061	0.049	0.021	0.016	0.011	0.001	0.711
8	100 000	1.5	2608	0.390	0.104	0.063	0.061	0.049	0.021	0.016	0.011	0.001	0.716
9	100 000	1.5	2627	0.393	0.105	0.064	0.061	0.049	0.021	0.016	0.011	0.001	0.721
10	100 000	1.5	2574	0.397	0.105	0.064	0.061	0.049	0.021	0.016	0.011	0.001	0.724
11	100 000	1.5	2602	0.399	0.105	0.064	0.061	0.049	0.021	0.016	0.011	0.001	0.727
12	500 000	2	2578	0.470	0.135	0.080	0.074	0.061	0.026	0.021	0.013	0.001	0.881
13	500 000	2	1269	0.520	0.142	0.083	0.078	0.061	0.028	0.022	0.014	0.001	0.949
14	500 000	2	1019	0.525	0.143	0.083	0.077	0.061	0.028	0.022	0.014	0.001	0.955
15	500 000	2	1027	0.539	0.145	0.082	0.078	0.061	0.029	0.022	0.014	0.001	0.970

Table 8c: Correlation matrix of $\beta_0, \beta_1, \ldots, \beta_k$ in the posterior distribution of simulation number 15

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
β_0	1.00								
β_1	0.05	1.00							
β_2	-0.46	0.08	1.00						
β_3	-0.10	0.00	-0.05	1.00					
β_4	0.57	-0.60	-0.02	0.02	1.00				
β_5	0.16	0.00	-0.00	-0.05	0.18	1.00			
β_6	0.04	-0.06	-0.01	0.01	0.15	0.69	1.00		
β_7	0.06	0.06	0.07	-0.14	0.02	0.02	0.05	1.00	
β_8	0.07	-0.10	-0.17	0.05	0.02	0.02	-0.07	-0.51	1.00

To answer this question we choose $r = 2$ and now we generate $M = 500\,000$ data vectors from the prior distribution in order to find sufficiently many acceptable data vectors. The results of 4 iterations starting with the covariance matrix $\mathbf{C}_{\text{app}}$ of iteration 11 is given in Table 8b (last 4 lines). We see that the sum of the eigenvalues is still clearly increasing and so we cannot hope to have found the noninfomative prior yet.

What can we do to come closer to the noninformative prior? One could choose a still larger value of *r* and enhance the number *M* of generated data vectors from the prior distribution correspondingly. Another possibility is to replace the multivariate normal prior distribution by some other multivariate distribution. We tried with the Cauchy, Laplace, uniform and triangular distribution. And the best results (largest eigenvalues of the posterior distribution with a given value of *M*) were found with the uniform distribution. These results are given in the next section.

6. Multivariate uniform distribution as prior distribution

Here we proceed as in the preceeding section but with the multivariate normal distribution being replaced by the multvariate uniform distribution. If *u* denotes a random variable with a uniform distribution in $[a,b]$ then $E(u) = (a+b)/2$ and $Var(u) = (b-a)^2 / 12$. So if $a = -b = \sqrt{3}$ *u* has mean zero and variance 1, and we say that *u* has a standard uniform distribution. Let $u_1, \ldots, u_p$ be independent standard uniform variables and let $\mathbf{\Sigma} = (p \times p)$ be any covariance matrix. $\mathbf{\Sigma}$ can be written as $\mathbf{\Sigma} = \mathbf{R\Lambda R}^\mathsf{T}$, where $\mathbf{R}$ is orthogonal and $\mathbf{\Lambda}$ is diagonal. Let $\mathbf{u} = (u_1, \ldots, u_p)^\mathsf{T}$ and $\mathbf{v} = \mathbf{R\Lambda}^{1/2}\mathbf{u}$. As $\mathbf{\Lambda}^{1/2}\mathbf{u} = (\sqrt{\lambda_1}\, u_1, \ldots, \sqrt{\lambda_p}\, u_p)^\mathsf{T}$ we obtain $\mathbf{v}$ by rescaling and rotating $\mathbf{u}$, and so the distribution of $\mathbf{v}$ is the uniform distribution in a rotated p-dimensional rectangle. The covariance matrix of $\mathbf{v}$ is given by $\mathbf{R\Lambda}^{1/2}(\mathbf{R\Lambda}^{1/2})^\mathsf{T} = \mathbf{R\Lambda R}^\mathsf{T} = \mathbf{\Sigma}$. If $\mu_1, \ldots, \mu_p$ denote any real numbers and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)^\mathsf{T}$, then the vector $\mathbf{v} + \boldsymbol{\mu}$ has a multivariate uniform distribution $U_p(\boldsymbol{\mu}, \mathbf{\Sigma})$ with $E(\mathbf{v}) = \boldsymbol{\mu}$ and $Cov(\mathbf{v}) = \mathbf{\Sigma}$.

Now we proceed as follows:

Step 1: Determine a first approximation $\mathbf{C}_{\text{app}}$ of the covariance matrix $\mathbf{C} = (p \times p) = (9 \times 9)$ of the posterior distribution. One can start with $\mathbf{C}_{\text{app}} = \mathbf{I}_p$ (identity matrix), or with some other covariance matrix that could be an approximation to $\mathbf{C}$.

Step 2: Generate $M = 500\,000$ random vectors from the multivariate uniform distribution $U_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu} = \hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate for $\boldsymbol{\beta}$ (see section 3) and where $\boldsymbol{\Sigma} = r^2 \mathbf{C}_{\text{app}}$ with a factor $r > 1$ chosen as large as possible to come close to the noninformative prior.

Step 3: As the acceptance probability for a data vector $\boldsymbol{\beta}_i = (\beta_{i0}, \ldots, \beta_{i8})$ is given by $p(\mathbf{y} \mid \boldsymbol{\beta}_i)/c = p(\mathbf{y} \mid \boldsymbol{\beta}_i)/p_{\max}$, we can proceed as in section 4 to compute the acceptable data vectors from the posterior distribution.

Step 4: Compute the covariance matrix of the accepted data vectors from the posterior distribution and denote it by $\mathbf{C}_{\text{app}}$. Continue with step 2 until stabilisation of $\mathbf{C}_{\text{app}}$.

How can we generate $M = 500\,000$ random vectors from a multivariate uniform distribution $U_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$?

(i) Determine the decomposition $\boldsymbol{\Sigma} = \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^{\mathsf{T}}$ where $\mathbf{R}$ is orthogonal and $\boldsymbol{\Lambda}$ is diagonal. Compute $\mathbf{A} = \mathbf{R} \boldsymbol{\Lambda}^{1/2}$. Note that we do not choose the symmetric root of $\boldsymbol{\Sigma}$ here as in section 4 with the normal distribution; if $\mathbf{x} = (p \times 1)$ is a random vector with independent standard normal components then $\mathbf{y} = \mathbf{R}\mathbf{x}$ has again independent standard normal components, but this is not true for the uniform distribution. Now, if $\mathbf{x} = (p \times 1)$ is a random vector with independent standard uniform components then $\mathbf{y} = \mathbf{A}\mathbf{x}$ has a multivariate uniform distribution $N_p(\mathbf{0}, \boldsymbol{\Sigma})$ as $\mathbf{A}\mathbf{A}^{\mathsf{T}} = \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^{\mathsf{T}} = \boldsymbol{\Sigma}$.

(ii) Generate $M = 500\,000$ random vectors from $U_p(\mathbf{0}, \mathbf{I})$, and arrange these data in a matrix $\mathbf{X} = (M \times p)$. Let $\mathbf{Y} = \mathbf{X}\mathbf{A}$. Now the rows of $\mathbf{Y}$ are independent random vectors from $U_p(\mathbf{0}, \boldsymbol{\Sigma})$.

(iii) Add μ_j to column *j* of the matrix **Y**; we denote the resulting matrix by $\mathbf{Z} = (M \times p)$; its rows are independent random vectors from $U_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Table 9a gives the results of the first simulation. As an approximation $\mathbf{C}_{\text{app}}$ of the covariance matrix $\mathbf{C} = (p \times p) = (9 \times 9)$ of the posterior distribution we compute the covariance matrix of the 1027 data vectors of the last simulation in Table 8b. Then we compute $\boldsymbol{\Sigma} = r^2 \mathbf{C}_{\text{app}}$ with $r^2 = 3$, determine the decomposition $\boldsymbol{\Sigma} = \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^{\mathsf{T}}$ and compute the matrix $\mathbf{A} = \mathbf{R} \boldsymbol{\Lambda}^{1/2}$. The starting value of our random number generator is again set to 77, and the result of step 2 are the data vectors in column (0) to (8) of Table 9a. These vectors can be considered as independent random vectors from $U_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The rest of Table 9a is computed exactly the same way as the corresponding columns in Table 7a. We find $M_{\text{acc}} = 690$ acceptable data vectors from the posterior distribution (out of the $M = 500\,000$ data vectors from the multivariate uniform distribution). We denote the covariance matrix of these 690 vectors as $\mathbf{C}_{\text{app}}$ (new approximation), and compute its eigenvalues; they are found in the first row of Table 9b. If we compare these eigenvalues with those of the old approximation (last line in Table 8b), we can see that the eigenvalues of the new approximation are somewhat larger, and so the multivariate uniform distribution comes closer to the noninformative prior distribution than the multivariate normal distribution of simulation 15 in Table 8b.

For the second simulation we choose the new covariance matrix $\mathbf{C}_{\text{app}}$ as an approximation to the covariance matrix of the posterior distribution and we perform again the above steps 2 to 4, but this time we generate $M = 1\,000\,000$ data vectors from our prior distribution and we find $M_{\text{acc}} = 513$ acceptable data vectors from the posterior distribution (out of the $M = 1\,000\,000$). We compute the covariance matrix of these $M_{\text{acc}} = 513$ vectors, compute its eigenvalues (second row of Table 9b), and use this new covariance matrix as $\mathbf{C}_{\text{app}}$ for our third simulation.

In our third and last simulation we find $M_{\text{acc}} = 475$ acceptable data vectors from the posterior distribution (out of the $M = 1\,000\,000$). The third row of Table 9b gives the corresponding eigenvalues; not much has changed as compared with simulation 2. The correlation matrix of the $M_{\text{acc}} = 475$ accepted vectors in simulation 3 is given in Table 9c. If we compare this correlation matrix with the corresponding matrix in the last simulation of section 5 (see last row of Table 8b and Table 8c), we can see that eigenvalues with the uniform distribution are somewhat larger than with the normal distribution, but the correlation matrix is approximately the same.

Now we have to ask ourselves whether our prior distribution can be considered as approximately noninformative. This would be the case if any multivariate uniform prior distribution with a wider support would give essentially the same results. So we try to use a multivariate uniform distribution with $\mathbf{C}_{\text{app}}$ as in simulation 3 but with $r^2 = 6$ instead of $r^2 = 3$. As the ratio between the new and old prior density in the central part is given by $(1/2)^{9/2} = 0.0442$ ($k+1=9$ dimensions) we can expect only $475 \times 0.0442 = 21.0$ acceptable data vectors from the posterior distribution. We performed the simulation along the same lines as in simulation 3 and found just 15 acceptable data points out of one million generated from the prior distribution. So we must admit that it can be difficult to find a noninformative prior distribution if the number k of independent variables is too large ($k > 10$, about).

Table 9a: $M = 100000$ data vectors $\boldsymbol{\beta}_i = (\beta_{0j},\ldots,\beta_{8j})$ from a multivariate uniform distribution ($r^2 = 3$)

	(0)		(8)	(9)	(10)	(11)	(12)	(13)	(14)
i	β_{i0}	$\cdots$	β_{i8}	$\ln p(\mathbf{y}\|\boldsymbol{\beta}_i)$	(9) sorted	(9) $- l_{max}$	$\frac{p(\mathbf{y}\|\boldsymbol{\beta}_i)}{p_{max}}$	u_i	accept?
1	0.0915	$\cdots$	-1.9866	-514.358	-535.149	-5.360	0.0047	0.9239	0
2	0.4287	$\cdots$	-1.9431	-515.090	-534.414	-6.092	0.0023	0.2976	0
3	0.2479	$\cdots$	-2.8310	-514.265	-534.240	-5.267	0.0052	0.7315	0
4	0.4447	$\cdots$	-2.3770	-520.507	-534.207	-11.509	0.0000	0.1413	0
5	0.0903	$\cdots$	-2.9840	-518.933	-534.039	-9.935	0.0000	0.5389	0
6	0.4257	$\cdots$	-2.4369	-520.181	-533.675	-11.183	0.0000	0.5142	0
7	-0.0505	$\cdots$	-2.1193	-521.143	-533.634	-12.145	0.0000	0.5777	0
8	0.2757	$\cdots$	-2.0805	-515.818	-533.526	-6.820	0.0011	0.7893	0
9	0.5560	$\cdots$	-2.0514	-521.455	-533.433	-12.457	0.0000	0.0254	0
10	0.5853	$\cdots$	-2.6138	-519.884	-533.306	-10.886	0.0000	0.0470	0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
499991	0.4677	$\cdots$	-3.2353	-518.987	-509.834	-9.989	0.0000	0.9090	0
499992	-0.0470	$\cdots$	-2.9289	-521.851	-509.811	-12.853	0.0000	0.2265	0
499993	0.3454	$\cdots$	-2.8488	-512.699	-509.771	-3.701	0.0247	0.4353	0
499994	0.0923	$\cdots$	-2.5542	-522.325	-509.752	-13.327	0.0000	0.6102	0
499995	0.6657	$\cdots$	-2.5089	-520.250	-509.745	-11.252	0.0000	0.4774	0
499996	0.1032	$\cdots$	-1.8918	-523.651	-509.716	-14.653	0.0000	0.8084	0
499997	0.3099	$\cdots$	-2.3252	-514.756	-509.706	-5.758	0.0032	0.1900	0
499998	0.0945	$\cdots$	-2.9679	-515.036	-509.698	-6.038	0.0024	0.0098	0
499999	0.3410	$\cdots$	-2.2448	-521.220	-509.587	-12.222	0.0000	0.8073	0
500000	0.3639	$\cdots$	-2.6123	-513.330	-509.459	-4.332	0.0131	0.4647	0
sum							694.8		690

Table 9b: Eigenvalues of the covariance matrix of the posterior data in 3 simulations with $r^2 = 3$

	M	M_{acc}	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	sum
1	500 000	690	0.633	0.183	0.108	0.104	0.073	0.032	0.029	0.019	0.002	1.183
2	1 000 000	513	0.672	0.192	0.119	0.102	0.076	0.032	0.029	0.019	0.002	1.242
3	1 000 000	475	0.654	0.182	0.128	0.103	0.078	0.035	0.030	0.021	0.002	1.232

Table 9c: Correlation matrix of $\beta_0, \beta_1, \ldots, \beta_k$ in the posterior distribution of the last simulation

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
β_0	1.00								
β_1	0.07	1.00							
β_2	-0.46	0.05	1.00						
β_3	0.02	-0.04	-0.08	1.00					
β_4	0.59	-0.57	-0.01	0.14	1.00				
β_5	0.23	0.08	-0.16	-0.02	0.07	1.00			
β_6	0.03	-0.02	-0.10	-0.00	0.03	0.66	1.00		
β_7	0.00	0.05	0.10	-0.03	0.02	0.00	0.02	1.00	
β_8	0.13	-0.03	-0.20	0.01	0.01	0.06	0.02	-0.56	1.00

7. Results with MCMC method

Here we want to compare our results with the results that are found when the data from the posterior distribution are generated with the MCMC method using a diffuse prior distribution. I thank Dr. Stefan Lang from the Department of Statistics of the University of Munich who performed the computations with the program BayesX. The program was developed by Dr. Lang and two co-authors (see Brezger-Kneib-Lang, 2003).

Three simulations were performed all with a diffuse prior distribution:

	Simulation 1	Simulation 2	Simulation 2
Number of iterations	12 000	15 000	18 000
Burn-in period	1 000	2 000	4 000
Thinning parameter	5	10	20
Number of generated data vectors	2 200	1 300	700

The log-files of the three runs are given in the Appendix. The MCMC simulations were performed with the original variables $\tilde{x}_1,\ldots,\tilde{x}_8$ and not with the normalised variables $x_1,\ldots,x_8$ that we used up to now. So we want to transform the data vectors $\tilde{\boldsymbol{\beta}}_i = (\tilde{\beta}_{i0},\ldots,\tilde{\beta}_{i8})$ for the original variables to data vectors $\boldsymbol{\beta}_i = (\beta_{i0},\ldots,\beta_{i8})$ for the normalised variables. According to (1) we have

$$x_j = \frac{\tilde{x}_j - m_j}{2r_j} - \tfrac{1}{4},$$

where the normalising constants m_j and r_j are given in Table 1, and so

$$\tilde{x}_j = m_j + 2r_j\tilde{x}_j + \tfrac{1}{2}r_j\,.$$

From the equality

$$\tilde{\beta}_0 + \tilde{\beta}_1\tilde{x}_1 + \cdots + \tilde{\beta}_8\tilde{x}_8 \;=\; \beta_0 + \beta_1 x_1 + \cdots + \beta_8 x_8$$

we derive

$$\beta_0 = \tilde{\beta}_0 + \tilde{\beta}_1\left(m_1 + \tfrac{1}{2}r_1\right) + \tilde{\beta}_2\left(m_2 + \tfrac{1}{2}r_2\right) + \cdots + \tilde{\beta}_8\left(m_8 + \tfrac{1}{2}r_8\right) = \tilde{\beta}_0\left(4+34\right) + \tilde{\beta}_4\left(0.25 + 9.087\right)$$

$$\beta_j = 2r_j\tilde{\beta}_j\,,\;\; j = 1,\ldots,8\,.$$

After these transformations we compute for the $M_{acc} = 2200$ data vectors from the posterior distribution in simulation 1 the correlation matrix and the eigenvalues of the covariance matrix. The same is done for simulation 2 and 3. The correlation matrices are given in Table 10a to 10c and the eigenvalues in Table 11. We can see that the correlation matrices and the eigenvalues are essentially the same in all three simulations. So the optional MCMC parameters are not chosen too small. If we compare Table 10a to 10c and Table 11 with the corresponding tables in section 6 (Table 9b and 9c) we again see that the correlation matrices are essentially the same; but the eigenvalues with the MCMC method are somewhat larger than the eigenvalues in Table 9b. So one can find with the classical method described in section 6 (multivariate uniform distribution as prior distribution) essentially the same results as with the MCMC method. But we must admit that the classical method will fail if the number of dimensions becomes too large ($k > 10$, about) as then it will be difficult to find a noninformative prior distribution that still gives sufficiently many data from the posterior distribution (see section 6).

Table 10a: Correlation matrix of $\beta_0, \beta_1, \ldots, \beta_k$ in the posterior distribution simulation 1

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
β_0	1.00								
β_1	0.06	1.00							
β_2	-0.47	0.01	1.00						
β_3	-0.04	-0.09	-0.07	1.00					
β_4	0.57	-0.59	0.04	0.09	1.00				
β_5	0.14	0.05	-0.01	-0.07	0.11	1.00			
β_6	-0.02	-0.02	-0.01	0.02	0.05	0.68	1.00		
β_7	0.03	0.01	0.04	-0.03	0.05	0.01	0.00	1.00	
β_8	0.07	-0.01	-0.10	0.03	-0.02	0.02	0.02	-0.54	1.00

Table 10b: Correlation matrix of $\beta_0, \beta_1, \ldots, \beta_k$ in the posterior distribution of simulation 2

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
β_0	1.00								
β_1	0.09	1.00							
β_2	-0.49	-0.03	1.00						
β_3	-0.03	-0.04	-0.09	1.00					
β_4	0.57	-0.57	0.04	0.06	1.00				
β_5	0.15	0.09	0.01	-0.06	0.11	1.00			
β_6	0.00	0.03	-0.01	0.04	0.06	0.71	1.00		
β_7	0.09	0.01	0.03	0.02	0.09	0.00	0.00	1.00	
β_8	-0.01	0.01	-0.06	-0.00	-0.07	-0.01	0.00	-0.55	1.00

Table 10c: Correlation matrix of $\beta_0, \beta_1, \ldots, \beta_k$ in the posterior distribution of simulation 3

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8
β_0	1.00								
β_1	0.02	1.00							
β_2	-0.43	0.10	1.00						
β_3	-0.08	-0.05	-0.10	1.00					
β_4	0.58	-0.63	-0.02	0.01	1.00				
β_5	0.14	0.00	0.01	-0.12	0.12	1.00			
β_6	-0.06	0.01	0.04	-0.02	0.03	0.71	1.00		
β_7	0.06	-0.02	0.06	-0.09	0.07	0.04	0.03	1.00	
β_8	0.07	0.07	-0.07	0.05	-0.05	0.03	0.00	-0.55	1.00

Table 11: Eigenvalues of the covariance matrix of the posterior data in the 3 MCMC-simulations

	M_{acc}	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	sum
1	2200	0.724	0.191	0.114	0.108	0.094	0.038	0.030	0.020	0.002	1.322
2	1300	0.700	0.204	0.116	0.110	0.099	0.038	0.031	0.019	0.002	1.319
3	700	0.777	0.185	0.118	0.110	0.087	0.036	0.030	0.019	0.002	1.365

Appendix: Log-file of the three simulations with BayesX

Simulation 1

```
> dataset d
> d.infile using c:\texte\compstat\daten\kredit.raw
NOTE: 9 variables with 1000 observations read from file
c:\texte\compstat\daten\kredit.raw

> bayesreg b
> b.outfile = c:\tmp\kr_12000_1000_5
> b.regress boni = laufz + moral + zweck + hoehe + geschl + famst + ko1 + ko2 , itera-
tions=12000 burnin=1000 step=5  family=binomialprobit using d

BAYESREG OBJECT b: regression procedure

GENERAL OPTIONS:

  Number of iterations:  12000
  Burn-in period:        1000
  Thinning parameter:    5

RESPONSE DISTRIBUTION:
  Family: binomial
  Number of observations: 1000
  Number of observations with positive weights: 1000
  Response function: standard normal (probit link)

OPTIONS FOR ESTIMATION:
  OPTIONS FOR FIXED EFFECTS:

  Priors:

  diffuse priors

MCMC SIMULATION STARTED

  ITERATION: 1

  APPROXIMATE RUN TIME: 23 seconds

  ITERATION: 1000
  ITERATION: 2000
  ITERATION: 3000
  ITERATION: 4000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.158158
  Variance:           3.30948e+14
  Minimum:            0.641668
  Maximum:            0.529792

  ITERATION: 5000
  ITERATION: 6000
  ITERATION: 7000
  ITERATION: 8000

  FixedEffects1

  Acceptance rate:    100 %
```

```
  Relative Changes in

  Mean:               0.00432139
  Variance:           0.0511951
  Minimum:            0.0431187
  Maximum:            0.116425

  ITERATION: 9000
  ITERATION: 10000
  ITERATION: 11000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.00160812
  Variance:           0.0140995
  Minimum:            0.0143712
  Maximum:            0.000479932

  ITERATION: 12000

SIMULATION TERMINATED

SIMULATION RUN TIME: 24 seconds


ESTIMATION RESULTS:

  Estimation results for the intercept:

         mean           Std. Dev.      2.5% quant.    median         97.5% quant.
  const  -0.728514      0.114421       -0.941884      -0.729197      -0.50626

  Results for the intercept are also stored in file
  c:\tmp\kr_12000_1000_5_intercept.res


  FixedEffects1


  Acceptance rate:    100 %


  Variable  mean           Std. Dev.      2.5% quant.    median         97.5% quant.
  laufz     0.020626       0.00460928     0.0117804      0.0206232      0.0298025
  moral     -0.293213      0.0791512      -0.450335      -0.292876      -0.139568
  zweck     -0.139871      0.0487599      -0.235281      -0.139976      -0.0426651
  hoehe     0.0189294      0.0196181      -0.019854      0.0184192      0.0565769
  geschl    0.0668102      0.0641114      -0.057821      0.0668595      0.192727
  famst     -0.109951      0.0654435      -0.237741      -0.109278      0.0143968
  ko1       0.511998       0.0623867      0.388653       0.513778       0.634742
  ko2       -0.629133      0.066475       -0.764348      -0.62726       -0.501886
```

Simulation 2

```
> b.outfile = c:\tmp\kr_15000_2000_10
> b.regress boni = laufz + moral + zweck + hoehe + geschl + famst + ko1 + ko2 , itera-
tions=15000 burnin=2000 step=10  family=binomialprobit using d

BAYESREG OBJECT b: regression procedure

GENERAL OPTIONS:

  Number of iterations:  15000
  Burn-in period:        2000
  Thinning parameter:    10

RESPONSE DISTRIBUTION:

  Family: binomial
  Number of observations: 1000
  Number of observations with positive weights: 1000
  Response function: standard normal (probit link)

OPTIONS FOR ESTIMATION:

  OPTIONS FOR FIXED EFFECTS:

  Priors:

  diffuse priors

MCMC SIMULATION STARTED

  ITERATION: 1

  APPROXIMATE RUN TIME: 28 seconds

  ITERATION: 1000
  ITERATION: 2000
  ITERATION: 3000
  ITERATION: 4000
  ITERATION: 5000
  ITERATION: 6000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.165164
  Variance:           7.07928e+14
  Minimum:            0.624997
  Maximum:            0.580698

  ITERATION: 7000
  ITERATION: 8000
  ITERATION: 9000
  ITERATION: 10000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.00753161
  Variance:           0.0332624
  Minimum:            0.0302839
  Maximum:            0.143588
```

```
  ITERATION: 11000
  ITERATION: 12000
  ITERATION: 13000
  ITERATION: 14000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.00359838
  Variance:           0.0431959
  Minimum:            0.0806361
  Maximum:            0.029805

  ITERATION: 15000

SIMULATION TERMINATED

SIMULATION RUN TIME: 30 seconds

ESTIMATION RESULTS:

  Estimation results for the intercept:

         mean           Std. Dev.      2.5% quant.    median         97.5% quant.
  const  -0.724311      0.115032       -0.96177       -0.721248      -0.505426

  Results for the intercept are also stored in file
  c:\tmp\kr_15000_2000_10_intercept.res

  FixedEffects1

  Acceptance rate:    100 %

  Variable  mean           Std. Dev.      2.5% quant.    median         97.5% quant.
  laufz     0.0205542      0.00460502     0.011275       0.0205432      0.029393
  moral     -0.290763      0.0796846      -0.440505      -0.292199      -0.132041
  zweck     -0.139515      0.0481798      -0.231566      -0.140301      -0.044125
  hoehe     0.0181128      0.0192823      -0.0217374     0.0183151      0.0545265
  geschl    0.0631148      0.0675965      -0.0685707     0.0609438      0.197365
  famst     -0.113338      0.0656589      -0.245869      -0.113483      0.013942
  ko1       0.511449       0.0645808      0.386591       0.50865        0.63758
  ko2       -0.630771      0.0669017      -0.765591      -0.629044      -0.496258
```

Simulation 3

```
> b.outfile = c:\tmp\kr_18000_4000_20
> b.regress boni = laufz + moral + zweck + hoehe + geschl + famst + ko1 + ko2 , itera-
tions=18000 burnin=4000 step=20  family=binomialprobit using d

BAYESREG OBJECT b: regression procedure

GENERAL OPTIONS:

  Number of iterations:  18000
  Burn-in period:        4000
  Thinning parameter:    20

RESPONSE DISTRIBUTION:

  Family: binomial
  Number of observations: 1000
  Number of observations with positive weights: 1000
  Response function: standard normal (probit link)

OPTIONS FOR ESTIMATION:

  OPTIONS FOR FIXED EFFECTS:

  Priors:

  diffuse priors

MCMC SIMULATION STARTED

  ITERATION: 1

  APPROXIMATE RUN TIME: 35 seconds

  ITERATION: 1000
  ITERATION: 2000
  ITERATION: 3000
  ITERATION: 4000
  ITERATION: 5000
  ITERATION: 6000
  ITERATION: 7000
  ITERATION: 8000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.229289
  Variance:           4.80195e+14
  Minimum:            0.703814
  Maximum:            0.597886

  ITERATION: 9000
  ITERATION: 10000
  ITERATION: 11000
  ITERATION: 12000
  ITERATION: 13000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.00355154
  Variance:           0.0402934
  Minimum:            0.0150448
  Maximum:            0.0684176
```

```
  ITERATION: 14000
  ITERATION: 15000
  ITERATION: 16000
  ITERATION: 17000

  FixedEffects1

  Acceptance rate:    100 %

  Relative Changes in

  Mean:               0.00482424
  Variance:           0.0303285
  Minimum:            0.0367421
  Maximum:            0.0945194

  ITERATION: 18000

SIMULATION TERMINATED

SIMULATION RUN TIME: 36 seconds

ESTIMATION RESULTS:

  Estimation results for the intercept:

        mean           Std. Dev.      2.5% quant.    median         97.5% quant.
  const  -0.722283      0.114144       -0.946497      -0.721339      -0.513788

  Results for the intercept are also stored in file
  c:\tmp\kr_18000_4000_20_intercept.res

  FixedEffects1

  Acceptance rate:    100 %

  Variable  mean           Std. Dev.      2.5% quant.    median         97.5% quant.
  laufz     0.0206172      0.00491122     0.0111257      0.0204497      0.030657
  moral     -0.294855      0.0740638      -0.447033      -0.292939      -0.150401
  zweck     -0.138648      0.0466079      -0.227665      -0.138944      -0.0471488
  hoehe     0.0188599      0.0194518      -0.0186513     0.018544       0.0588403
  geschl    0.0657355      0.0668609      -0.0742874     0.0665778      0.187904
  famst     -0.111984      0.0662743      -0.249423      -0.11182       0.0165249
  ko1       0.510375       0.0647189      0.382203       0.510969       0.641179
  ko2       -0.628938      0.0672198      -0.768028      -0.628772      -0.497938

  Results for fixed effects are also stored in file
  c:\tmp\kr_18000_4000_20_FixedEffects1.res

> logclose
```

References

Besag, J. (2000). Markov Chain Monte Carlo for Statistical Inference. Working Paper No. 9 of Center for Statistics and the Social Sciences, University of Washington, USA.

Brezger, A., Kneib, Th., Lang, S. (2003). BayesX Manual. Can be obtained together with the program BayesX under the address *www.stat.uni-muenchen.de/~lang*

Fahrmeir, L. and Tutz, G. (2001). Multivariate Statistical Modelling Based on Generalized Linear Models. Second Edition. Springer, New York.

Fahrmeir, L. and Lang, S. (2001). Bayesian Inference for Generalized Additive Mixed Models Based on Markov Random Field Priors. Journal of the Royal Statistical Society, Series C (Applied Statistics), 50, 201-220.

Knüsel, L. (2003). Alternatives to the MCMC Method; an Example with Real Data. Submitted for publication in *Computational Statistics and Data Analysis*.

Ripley, B. (1987). Stochastic Simulation. Wiley, New York.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). Annals of Statistics, 22, 1701-1762.

Address:
Professor Leo Knüsel
Department of Statistics
University of Munich
80539 Munich, Germany
knuesel@stat.uni-muenchen.de