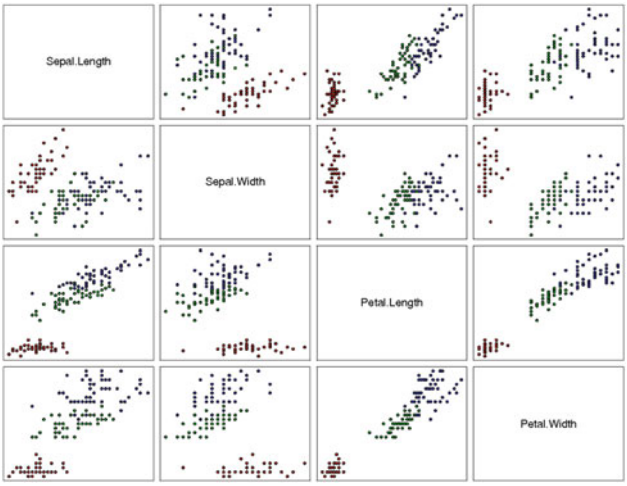
4

Generalized Linear Models



This was the sort of thing that impressed
Rebus: not nature, but ingenuity.
—**Ian Rankin**, ***A Question of Blood***.—

Roadmap

Generalized linear models are extensions of the linear regression model described in the previous chapter. In particular, they avoid the selection of a single transformation of the data that must achieve the possibly conflicting goals of normality and linearity imposed by the linear regression model, which is for instance impossible for binary or count responses. The trick that allows both a feasible processing and an extension of linear regression is first to turn the covariates into a real number by a linear projection and then to transform this value so that it fits the support of the response. We focus here on the Bayesian analysis of probit and logit models for binary data and of log-linear models for contingency tables.

On the methodological side, we present a general MCMC method, the Metropolis–Hastings algorithm, which is used for the simulation of complex distributions where both regular and Gibbs sampling fail. This includes in particular the random walk Metropolis–Hastings algorithm, which acts like a plain vanilla MCMC algorithm.

J.-M. Marin and C.P. Robert, *Bayesian Essentials with R*, Springer Texts in Statistics, DOI 10.1007/978-1-4614-8687-9_4,
© Springer Science+Business Media New York 2014

4.1 A Generalization of the Linear Model

4.1.1 Motivation

In the previous chapter, we modeled the connection between a response variable y and a vector $\mathbf{x}$ of explanatory variables by a linear dependence relation with normal perturbations. There are many instances where both the linearity and the normality assumptions are not appropriate, especially when the support of y is restricted to $\mathbb{R}_+$ or $\mathbb{N}$. For instance, in dichotomous models, y takes its values in $\{0, 1\}$ as it represents the indicator of occurrence of a particular event (death in a medical study, unemployment in a socioeconomic study, migration in a capture–recapture study, etc.); in this case, a linear conditional expectation $\mathbb{E}[y|\mathbf{x}, \boldsymbol{\beta}] = \mathbf{x}^{\mathsf{T}}\boldsymbol{\beta}$ would be fairly cumbersome to handle, both in terms of the constraints on $\boldsymbol{\beta}$ and the corresponding distribution of the error $\varepsilon = y - \mathbb{E}[y|\mathbf{x}, \boldsymbol{\beta}]$.

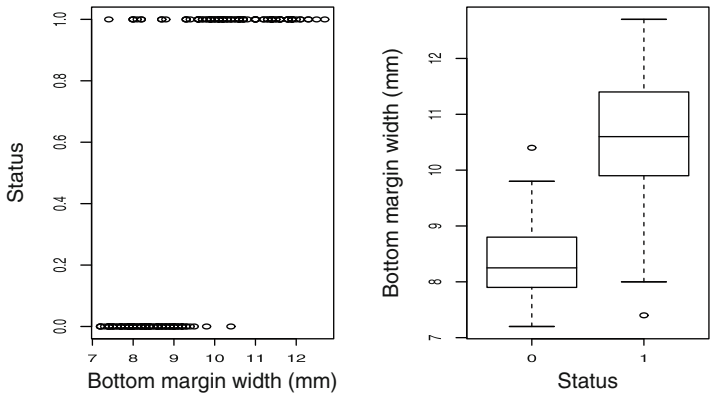


Fig. 4.1. Dataset **bank**: (*left*) Plot of the status indicator versus the bottom margin width; (*right*) boxplots of the bottom margin width for both counterfeit statuses

The **bank** dataset we analyze in the first part of this chapter comes from Flury and Riedwyl (1988) and is made of four measurements on 100 genuine Swiss banknotes and 100 counterfeit ones. The response variable y is thus the status of the banknote, where 0 stands for genuine and 1 stands for counterfeit, while the explanatory factors are the length of the bill x_1, the width of the left edge x_2, the width of the right edge x_3, and the bottom margin width x_4,

all expressed in millimeters. We want a probabilistic model that predicts the type of banknote (i.e., that detects counterfeit banknotes) based on the four measurements above. To motivate the introduction of the generalized linear models, we only consider here the dependence of y on the fourth measure, x_4, which again is the bottom margin width of the banknote. To start, the y_i's being binary, the conditional distribution of y given x_4 cannot be normal. Nonetheless, as shown by Fig. 4.1, the variable x_4 clearly has a strong influence on whether the banknote is or is not counterfeit. To model this dependence in a proper manner, we must devise a realistic (if not real!) connection between y and x_4. The fact that y is binary implies a specific form of dependence: Indeed, both its marginal and conditional distributions necessarily are Bernoulli distributions. This means that, for instance, the conditional distribution of y given x_4 is a Bernoulli $\mathscr{B}(p(x_4))$ distribution; that is, for $x_4 = x_{4i}$, there exists $0 \leq p_i = p(x_{4i}) \leq 1$ such that

$$\mathbb{P}(y_i = 1 | x_4 = x_{4i}) = p_i \,,$$

which turns out to be also the conditional expectation of y_i, $\mathbb{E}[y_i | x_{4i}]$. If we do impose a linear dependence on the p_i's, namely,

$$p(x_{4i}) = \beta_0 + \beta_1 x_{4i} \,,$$

the maximum likelihood estimates of β_0 and β_1 are then equal to -2.02 and 0.268, leading to the estimated prediction equation

$$\hat{p}_i = -2.02 + 0.268 x_{i4} \,. \tag{4.1}$$

This implies that a banknote with bottom margin width equal to 8 is counterfeit with probability

$$-2.02 + 0.268 \times 8 = 0.120 \,.$$

Thus, this banknote has a relatively small probability of having been counterfeited, which coincides with the intuition drawn from Fig. 4.1. However, if we now consider a banknote with bottom margin width equal to 12, (4.1) implies that this banknote is counterfeited with probability

$$-2.02 + 0.268 \times 12 = 1.192 \,,$$

which is certainly embarrassing for a probability estimate! We could try to modify the result by truncating the probability to $(0,\ 1)$ and by deciding that this value of x_4 almost certainly indicates a counterfeit, but still there is a fundamental difficulty with this model. The fact that an ordinary linear

dependence can predict values outside $(0, 1)$ suggests that the connection between this explanatory variable and the probability of a counterfeit cannot be modeled through a linear function but rather can be achieved using functions of x_{4i} that take their values within the interval $(0, 1)$.

4.1.2 Link Functions

As shown by the previous analysis, while linear models are nice to work with, they also have strong limitations. Therefore, we need a broader class of models to cover various dependence structures. The class selected for this chapter is called the family of *generalized linear models* (GLM), which has been formalized in McCullagh and Nelder (1989). This nomenclature stems from the fact that the dependence of y on $\mathbf{x}$ is partly *linear* in the sense that the conditional distribution of y given x is defined in terms of a linear combination $\mathbf{x}^\mathsf{T}\boldsymbol{\beta}$ of the components of $\mathbf{x}$,

$$y|\boldsymbol{\beta} \sim f(y|\mathbf{x}^\mathsf{T}\boldsymbol{\beta}) .$$

As in the previous chapter, we use the notation $\mathbf{y} = (y_1, \ldots, y_n)$ for a sample of n responses and

$$\mathbf{X} = [\mathbf{x}_1 \quad \ldots \quad \mathbf{x}_k] = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1k} \\ x_{21} & x_{22} & \ldots & x_{2k} \\ x_{31} & x_{32} & \ldots & x_{3k} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{nk} \end{bmatrix}$$

for the $n \times k$ matrix of corresponding explanatory variables, possibly with $x_{11} = \ldots = x_{n1} = 1$. We use y and $\mathbf{x}$ as generic notations for single-response and covariate vectors, respectively. Once again, we will omit the dependence on $\mathbf{x}$ or $\mathbf{X}$ to simplify notations.

A *generalized linear model* is specified by two functions:

1. a conditional density f of y given $\mathbf{x}$ that belongs to an exponential family (Sect. 2.2.3) and that is parameterized by an expectation parameter $\mu = \mu(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}]$ and possibly a dispersion parameter $\varphi > 0$ that does not depend on $\mathbf{x}$; and
2. a *link* function g that relates the mean $\mu = \mu(\mathbf{x})$ of f and the covariate vector, $\mathbf{x}$, as $g(\mu) = (\mathbf{x}^\mathsf{T}\boldsymbol{\beta})$, $\boldsymbol{\beta} \in \mathbb{R}^k$.

For identifiability reasons, the link function g is a one-to-one function and we have

$$\mathbb{E}[y|\boldsymbol{\beta}, \varphi] = g^{-1}\left(\mathbf{x}^\mathsf{T}\boldsymbol{\beta}\right) .$$

We can thus write the (conditional) likelihood as

$$\ell(\boldsymbol{\beta}, \varphi|\mathbf{y}) = \prod_{i=1}^{n} f\left(y_i|\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}, \varphi\right)$$

if we choose to reparameterize f with the transform $g(\mu_i)$ of its mean and if we denote by $\mathbf{x}^i$ the covariate vector for the ith observation.[1]

The ordinary linear regression is obviously a special case of GLM where $g(x) = x$, $\varphi = \sigma^2$ and $y|\boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}\left(\mathbf{x}^\mathsf{T}\boldsymbol{\beta}, \sigma^2\right)$. However, outside the linear model, the interpretation of the coefficients β_i is much more delicate because these coefficients do not relate directly to the observables, due to the presence of a link function that cannot be the identity. For instance, in the logistic regression model (defined in the following paragraph), the linear dependence is defined in terms of the *log-odds ratio* $\log\{p_1/(1-p_i)\}$.

The most widely used GLMs are presumably those that analyze binary data, as in **bank**, that is, when $y_i \sim \mathscr{B}(1, p_i)$ (with $\mu_i = p_i = p(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})$). The mean function p thus transforms a real value into a value between 0 and 1, and a possible choice of link function is the *logit transform*,

$$g(p) = \log\{p/(1-p)\}\,,$$

associated with the *logistic regression model*. Because of the limited support of the responses y_i, there is no dispersion parameter in this model and the corresponding likelihood function is

$$\begin{aligned}\ell(\boldsymbol{\beta}|\mathbf{y}) &= \prod_{i=1}^{n} \left(\frac{\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})}{1+\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})}\right)^{y_i} \left(\frac{1}{1+\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})}\right)^{1-y_i} \\ &= \exp\left\{\sum_{i=1}^{n} y_i\, \mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}\right\} \Big/ \prod_{i=1}^{n} \left[1+\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right]\,. \end{aligned} \tag{4.2}$$

It thus fails to factorize conveniently because of the denominator: there is no manageable conjugate prior for this model, called the *logit model*.

There exists a specific form of link function for each exponential family which is called the *canonical link*. This canonical function is chosen as the function $g^\star$ of the expectation parameter that appears in the exponent of the natural exponential family representation of the probability density, namely

$$g^\star(\mu) = \theta \quad \text{if} \quad f(y|\mu,\varphi) = h(y)\exp\varphi\{T(y)\cdot\theta - \Psi(\theta)\}\,.$$

Since the logistic regression model can be written as

$$f(y_i|p_i) = \exp\left\{y_i \log\left(\frac{p_i}{1-p_i}\right) + \log(1-p_i)\right\}\,,$$

the logit link function is the canonical version for the Bernoulli model. Note that, while it is customary to use the canonical link, there is no compelling reason to do so, besides following custom!

[1] This upper indexing allows for the distinction between x_i, the ith component of the covariate vector, and $\mathbf{x}^i$, the ith vector of covariates in the sample.

For binary response variables, many link functions can be substituted for the logit link function. For instance, the *probit* link function, $g(\mu_i) = \Phi^{-1}(\mu_i)$, where Φ is the standard normal cdf, is often used in econometrics. The corresponding likelihood is

$$\ell(\boldsymbol{\beta}|\mathbf{y}) \propto \prod_{i=1}^{n} \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})^{y_i} \left[1 - \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right]^{1-y_i} . \tag{4.3}$$

Although this alternative is also quite arbitrary and any other cdf could be used as a link function (such as the logistic cdf associated with (4.2)), the probit link function enjoys a missing-data (Chap. 6) interpretation that clearly boosted its popularity: This model can indeed be interpreted as a degraded linear regression model in the sense that observing $y_i = 1$ corresponds to the case $z_i \geq 0$, where z_i is a latent (that is, unobserved) variable such that $z_i \sim \mathcal{N}\left(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}, 1\right)$. In other words, $y = \mathbb{I}(z_i \geq 0)$ appears as a dichotomized linear regression response. Of course, this perspective is only an *interpretation* of the probit model in the sense that there may be no hidden z_i's at all in the real world! In addition, the probit and logistic regression models have quite similar behaviors, differing mostly in the tails.

Another type of GLM deals with unbounded integer-valued variables. The *Poisson regression model* starts from the assumption that the y_i's are Poisson $\mathcal{P}(\mu_i)$ and it selects a link function connecting $\mathbb{R}^+$ bijectively with $\mathbb{R}$, such as, for instance, the logarithmic function, $g(\mu_i) = \log(\mu_i)$. This model is thus a *count* model in the sense that the responses are integers, for instance the number of deaths due to lung cancer in a county or the number of speeding tickets issued on a particular stretch of highway, and it is quite common in epidemiology. The corresponding likelihood is

$$\ell(\boldsymbol{\beta}|\mathbf{y}) = \prod_{i=1}^{n} \left(\frac{1}{y_i!}\right) \exp\left\{y_i\, \mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta} - \exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right\} ,$$

where the factorial terms $(1/y_i!)$ are irrelevant for both likelihood and posterior computations. Note that it does not factorize conveniently because of the exponential terms within the exponential.

The three examples above are simply illustrations of the versatility of generalized linear modeling. In this chapter, we discuss only two types of data for which generalized linear modeling is appropriate. We refer the reader to McCullagh and Nelder (1989) and Gelman et al. (2013) for a much more detailed coverage.

4.2 Metropolis–Hastings Algorithms

As partly hinted by the previous examples, posterior inference in GLMs is much harder than in linear models because of less manageable (and non-factorizing) likelihoods, which explains the longevity and versatility of linear

model studies over the past centuries! Working with a GLM typically requires specific numerical or simulation tools. We take the opportunity of this requirement to introduce a universal MCMC method called the *Metropolis–Hastings* algorithm. Its range of applicability is incredibly broad (meaning that it is by no means restricted to GLM applications) and its inclusion in the Bayesian toolbox in the early 1990s has led to considerable extensions of the Bayesian field.[2]

4.2.1 Definition

When compared with the Gibbs sampler, Metropolis–Hastings algorithms are generic (or off-the-shelf) MCMC algorithms in the sense that they can be tuned toward a much wider range of possibilities. Those algorithms are also a natural extension of standard simulation algorithms such as accept–reject (see Chap. 5) or sampling importance resampling methods since they are all based on a *proposal* distribution. However, a major difference is that, for the Metropolis–Hastings algorithms, the proposal distribution is *Markov,* with kernel density $q(x, y)$. If the *target* distribution has density π, the Metropolis–Hastings algorithm is as follows:

Algorithm 4.6 Generic Metropolis–Hastings Sampler

Initialization: Choose an arbitrary starting value $x^{(0)}$.
Iteration t $(t \geq 1)$:

1. Given $x^{(t-1)}$, generate $\tilde{x} \sim q(x^{(t-1)}, x)$.
2. Compute

$$\rho(x^{(t-1)}, \tilde{x}) = \min\left(\frac{\pi(\tilde{x})/q(x^{(t-1)}, \tilde{x})}{\pi(x^{(t-1)})/q(\tilde{x}, x^{(t-1)})}, 1\right).$$

3. With probability $\rho(x^{(t-1)}, \tilde{x})$, accept $\tilde{x}$ and set $x^{(t)} = \tilde{x}$; otherwise reject $\tilde{x}$ and set $x^{(t)} = x^{(t-1)}$.

The distribution q is also called the *instrumental* distribution. As in the accept–reject method (Sect. 5.4), we only need to know either π or q up to a proportionality constant since both constants cancel in the calculation of ρ. Note also the advantage of this approach compared with the Gibbs sampler: it is not necessary to use the conditional distributions of π.

The strong appeal of this algorithm is that it is rather universal in its formulation as well as in its use. Indeed, we only need to simulate from a

[2] This algorithm had been used by particle physicists, including Metropolis, since the late 1940s, but, as is often the case, the connection with statistics was not made until much later!

proposal q that can be chosen quite freely. There is, however, a theoretical constraint, namely that the chain produced by this algorithm must be able to explore the support of $\pi(y)$ in a finite number of steps. As discussed below, there also are many practical difficulties that are such that the algorithm may lose its universal feature and that it may require some specific tuning for each new application.

The theoretical validation of this algorithm is the same as with other MCMC algorithms: The target distribution π is the limiting distribution of the Markov chain produced by Algorithm 4.6. This is due to the choice of the acceptance probability $\rho(x, y)$ since the so-called *detailed balance equation*

$$\pi(x)q(x, y)\rho(x, y) = \pi(y)q(y, x)\rho(y, x)$$

holds and thus implies that π is stationary by integrating out x.

While theoretical guarantees that the algorithm converges are very high, the choice of q remains essential in practice. Poor choices of q may indeed result either in a very high rejection rate, meaning that the Markov chain $(x^{(t)})_t$ hardly moves, or in a myopic exploration of the support of π, that is, in a dependence on the starting value $x^{(0)}$ such that the chain is stuck in a neighborhood region of $x^{(0)}$. A particular choice of proposal q may thus work well for one target density but be extremely poor for another one. While the algorithm is indeed universal, it is impossible to prescribe application-independent strategies for choosing q.

We thus consider below two specific cases of proposals and briefly discuss their pros and cons (see Robert and Casella, 2004, Chap. 7, for a detailed discussion).

4.2.2 The Independence Sampler

The choice of q closest to the accept–reject method (see Algorithm 5.9) is to pick a constant q that is independent of its first argument,

$$q(x, y) = q(y)\,.$$

In that case, ρ simplifies into

$$\rho(x, y) = \min\left(1, \frac{\pi(y)/q(y)}{\pi(x)/q(x)}\right)\,.$$

In the special case in which q is proportional to π, we obtain $\rho(x, y) = 1$ and the algorithm reduces, as expected, to iid sampling from π. The analogy with the accept–reject algorithm is that the maximum of the ratio π/q is replaced with the current value $\pi(x^{(t-1)})/q(x^{(t-1)})$ but the sequence of accepted $x^{(t)}$'s is not iid because of the acceptance step.

The convergence properties of the algorithm depend on the density q. First, q needs to be positive everywhere on the support of π. Second, for good

exploration of this support, it appears that the ratio π/q needs to be bounded (see Robert and Casella, 2004, Theorem 7.8). Otherwise, the chain may take too long to reach some regions with low q/π values. This constraint obviously reduces the appeal of using an independence sampler, even though the fact that it does not require an explicit upper bound on π/q may sometimes be a plus.

This type of MH sampler is thus very model-dependent, and it suffers from the same drawbacks as the importance sampling methodology, namely that tuning the "right" proposal becomes much harder as the dimension increases.

4.2.3 The Random Walk Sampler

Since the independence sampler requires too much global information about the target distribution that is difficult to come by in complex or high-dimensional problems, an alternative is to opt for a local gathering of information, clutching to the hope that the accumulated information will provide, in the end, the global picture. Practically, this means exploring the neighborhood of the current value $x^{(t)}$ in search of other points of interest. The simplest exploration device is based on random walk dynamics.

A *random walk* proposal is based on a symmetric transition kernel $q(x, y) = q_{RW}(y - x)$ with $q_{RW}(x) = q_{RW}(-x)$. Symmetry implies that the acceptance probability $\rho(x, y)$ reduces to the simpler form

$$\rho(x, y) = \min\left(1, \pi(y)/\pi(x)\right) .$$

The appeal of this scheme is obvious when looking at the acceptance probability, since it only depends on the target π and since this version accepts all proposed moves that increase the value of π. There is considerable flexibility in the choice of the distribution q_{RW}, at least in terms of scale (i.e., the size of the neighborhood of the current value) and tails. Note that while from a probabilistic point of view random walks usually have no stationary distribution, the algorithm biases the random walk by moving toward modes of π more often than moving away from them.

The ambivalence of MCMC methods like the Metropolis–Hastings algorithm is that they can be applied to virtually any target. This is a terrific plus in that they can tackle new models, but there is also a genuine danger that they simultaneously fail to converge and fail to signal that they have failed to converge! Indeed, these algorithms can produce seemingly reasonable results, with all outer aspects of stability, while they are missing major modes of the target distribution. For instance, particular attention must be paid to models where the number of parameters exceeds by far the size of the dataset.

4.2.4 Output Analysis and Proposal Design

An important problem with the implementation of an MCMC algorithm is to gauge when convergence has been achieved; that is, to assess at what point the distribution of the chain is sufficiently close to its asymptotic distribution

for all practical purposes or, more practically, when it has covered the whole support of the target distribution with sufficient regularity. The number of iterations T_0 that is required to achieve this goal is called the *burn-in* period. It is usually sensible to discard simulated values within this burn-in period in the Monte Carlo estimation so that the bias caused by the starting value is reduced. However, and this is particularly true in high dimensions, the empirical assessment of MCMC convergence is extremely delicate, to the point that it is rarely possible to be certain that an algorithm has converged.[3] Nevertheless, some partial convergence diagnostic procedures can be found in the literature (see Robert and Casella, 2004, Chap. 12, and Robert and Casella, 2009, Chap. 8). In particular, the latter describes the R package `coda` in Sect. 8.2.4.

A first way to assess whether or not a chain is in its stationary regime is to visually compare trace plots of sequences started at different values, as it may expose difficulties related, for instance, to multimodality. In practice, when chains of length T from two starting values have visited substantially different parts of the state space, the burn-in period for at least one of the chains should be greater than T. Note, however, that the problem of obtaining overdispersed starting values can be difficult when little is known about the target density, especially in large dimensions.

Autocorrelation plots of particular components provide in addition good indications of the chain's mixing behavior. If ρ_k ($k \in \mathbb{N}^*$) denotes the kth-order autocorrelation,

$$\rho_k = \text{cov}\left(x^{(t)}, x^{(t+k)}\right) ,$$

these quantities can be estimated from the observed chain itself,[4] at least for small values of k, and an *effective sample size* factor can be deduced from these estimates,

$$T^{\text{ess}} = T \left(1 + 2 \sum_{k=1}^{T_0} \hat{\rho}_k\right)^{-1/2} ,$$

where $\hat{\rho}_k$ is the empirical autocorrelation function. This quantity represents the sample size of an equivalent iid sample when running T iterations. Conversely, the ratio T/T^{ess} indicates the multiplying factor on the minimum number of iid iterations required to run a simulation. Note, however, that this is only a partial indicator: Chains that remain stuck in one of the modes of the target distribution may well have a high effective ratio.

While we cannot discuss at length the selection of the proposal distribution (see Robert and Casella, 2004, Chap. 7, and Robert and Casella, 2009,

[3] Guaranteed convergence as in accept–reject algorithms is sometimes achievable with MCMC methods using techniques such as *perfect sampling* or *renewal*. But such techniques require a much more advanced study of the target distribution and the transition kernel of the algorithm. These conditions are not met very often in practice (see Robert and Casella 2004, Chap. 13).

[4] In R, this estimation can be conducted using the `acf` function.

Chap. 6), we stress that this is an important choice that has deep consequences for the convergence properties of the simulated Markov chain and thus for the exploration of the target distribution. As for prior distributions, we advise the simultaneous use of different kernels to assess their performances on the run. When considering a random walk proposal, for instance, a quantity that needs to be calibrated against the target distribution is the scale of this random walk. Indeed, if the variance of the proposal is too small with respect to the target distribution, the exploration of the target support will be small and may fail in more severe cases. Similarly, if the variance is too large, this means that the proposal will most often generate values that are outside the support of the target and that the algorithm will reject a large portion of attempted transitions.

It seems reasonable to tune the proposal distribution in terms of its past performances, for instance by increasing the variance if the acceptance rate is high or decreasing it otherwise (or moving the location parameter toward the mean estimated over the past iterations). This must not be implemented outside a burn-in step, though, because a permanent modification of the proposal distribution amounts to taking into account the whole past of the sequence and thus it cancels both its Markovian nature and its convergence guarantees.

Consider, solely for illustration purposes, the standard normal distribution $\mathcal{N}(0,1)$ as a target. If we use Algorithm 4.6 with a normal random walk, i.e.,

$$\tilde{x}|x^{(t-1)} \sim \mathcal{N}\left(x^{(t-1)}, \sigma^2\right) ,$$

the performance of the sampler depends on the value σ. An R function that implements the associated Hastings–Metropolis sampler is coded as

```
hm=function(n,x0,sigma2){
  x=rep(x0,n)
  for (i in 2:n){
    y=rnorm(1,x[i-1],sqrt(sigma2))
    if (runif(1)<=exp(-0.5*(y^2-x[i-1]^2))) x[i]=y
    else x[i]=x[i-1]
    }
  x
  }
```

For instance, picking σ^2 equal to either 10^{-4} or 10^3 provides two extreme cases: As shown in Fig. 4.2, the chain has a high acceptance rate but a low exploration ability and a high autocorrelation in the former case, while its acceptance rate is low but its ability to move around the normal range is high in the latter case (with a quickly decreasing autocorrelation). Both cases use the "wrong scale", though, in that the histograms of the simulation outputs are quite far from the target distribution after 10,000 iterations,

and this indicates that a much larger number of iterations must be used. A comparison with Fig. 4.3, which corresponds to $\sigma = 1$, clearly makes this point but also illustrates the fact that the large variance still induces large autocorrelations.

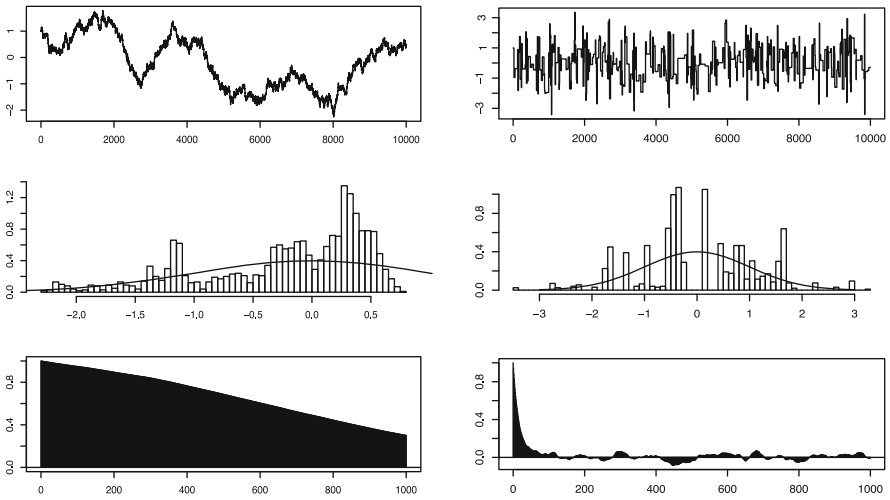


Fig. 4.2. Simulation of a $\mathcal{N}(0,1)$ target with (*left*) a $\mathcal{N}(x, 10^{-4})$ and (*right*) a $\mathcal{N}(x, 10^3)$ random walk proposal. *Top*: Sequence of 10,000 iterations; *middle*: histogram of the last 2,000 iterations compared with the target density; *bottom*: empirical autocorrelations using R function plot.acf

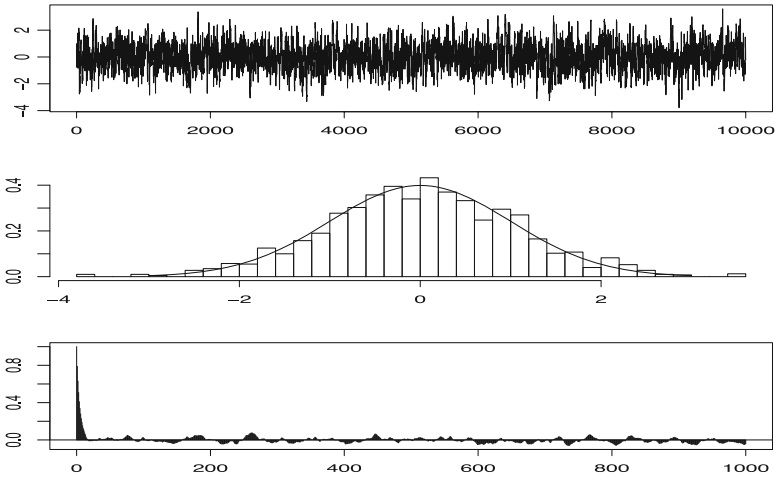


Fig. 4.3. Same legend as Fig. 4.2 for a $\mathcal{N}(x, 1)$ random walk proposal

Several MCMC algorithms can be mixed together within a single algorithm using either a circular or a random design. While this construction is often suboptimal (in that the inefficient algorithms in the mixture are still used on a regular basis), it almost always brings an improvement compared with its individual components. A special case where a mixed scenario is used is the *Metropolis-within-Gibbs* algorithm: When building a Gibbs sampler, it may happen that it is difficult or impossible to simulate from one or several of the conditional distributions. In that case, a single Metropolis step associated with this conditional distribution (as its target) can be used instead.[5]

4.3 The Probit Model

We now engage in a full discussion of the Bayesian processing of the probit model introduced in Sect. 4.1, taking special care to distinguish between the various types of prior modeling.

4.3.1 Flat Prior

If no prior information is available, we can resort (as usual!) to a default flat prior on $\boldsymbol{\beta}$, $\pi(\boldsymbol{\beta}) \propto 1$, and then obtain the posterior distribution

$$\pi(\boldsymbol{\beta}|\mathbf{y}) \propto \prod_{i=1}^{n} \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})^{y_i} \left[1 - \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right]^{1-y_i} ,$$

which is nonstandard and must be simulated using, e.g., MCMC techniques. First, the log-likelihood function is computable, as shown by the following R code[6]:

```
probitll=function(beta,y,X){
# probit likelihood
 if (is.matrix(beta)==F) beta=as.matrix(t(beta))
 n=dim(beta)[1]
 pll=rep(0,n)
 for (i in 1:n){
   lF1=pnorm(X%*%beta[i,],log=T)
   lF2=pnorm(-X%*%beta[i,],log=T)
```

[5] We stress that we do not resort to an MH algorithm for the purpose of simulating exactly from the corresponding conditional since this would require an infinite number of iterations but rather that we use a *single* iteration of the MH algorithm as a substitute for the simulation from the conditional since the resulting MCMC algorithm is still associated with the same stationary distribution.

[6] The use of the is.matrix test ensures that the function can be computed at one point as well as on multiple points and thus allows for calls from plot and other graphical functions.

```
    pll[i]=sum(y*lF1+(1-y)*lF2)
    }
  pll
  }
```

A variety of Metropolis–Hastings algorithms have been proposed for obtaining samples from this posterior distribution. Here we consider a sampler that appears to work well when the number of predictors is reasonably small. This Metropolis–Hastings sampler is a random walk scheme that uses the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ as a starting value and the asymptotic (Fisher) covariance matrix $\hat{\Sigma}$ of the maximum likelihood estimate as the covariance matrix for the proposal[7] density, $\tilde{\boldsymbol{\beta}} \sim \mathcal{N}_k(\boldsymbol{\beta}^{(t-1)}, \tau^2\hat{\Sigma})$.

Algorithm 4.7 Probit Metropolis–Hastings Sampler

Initialization: Compute the MLE $\hat{\boldsymbol{\beta}}$ and the covariance matrix $\hat{\Sigma}$ corresponding to the asymptotic covariance of $\hat{\boldsymbol{\beta}}$, and set $\boldsymbol{\beta}^{(0)} = \hat{\boldsymbol{\beta}}$.
Iteration $t \geq 1$:

1. Generate $\tilde{\boldsymbol{\beta}} \sim \mathcal{N}_k(\boldsymbol{\beta}^{(t-1)}, \tau^2\hat{\Sigma})$.
2. Compute
$$\rho(\boldsymbol{\beta}^{(t-1)}, \tilde{\boldsymbol{\beta}}) = \min\left(1, \pi(\tilde{\boldsymbol{\beta}}|\mathbf{y})/\pi(\boldsymbol{\beta}^{(t-1)}|\mathbf{y})\right).$$
3. With probability $\rho(\boldsymbol{\beta}^{(t-1)}, \tilde{\boldsymbol{\beta}})$, take $\boldsymbol{\beta}^{(t)} = \tilde{\boldsymbol{\beta}}$; otherwise take $\boldsymbol{\beta}^{(t)} = \boldsymbol{\beta}^{(t-1)}$.

The R function `glm` is obviously quite helpful in setting the initialization step of Algorithm 4.7. The step used in the R code to scale the algorithm is based on

```
> mod=summary(glm(y~X,family=binomial(link="probit")))
```

with `mod$coeff[,1]` corresponding to $\hat{\boldsymbol{\beta}}$ and `mod$cov.unscaled` to $\hat{\Sigma}$. The following code is then reproducing the above algorithm in R::

```
hmflatprobit=function(niter,y,X,scale){
 p=dim(X)[2]
 mod=summary(glm(y~-1+X,family=binomial(link="probit")))
 beta=matrix(0,niter,p)
 beta[1,]=as.vector(mod$coeff[,1])
 Sigma2=as.matrix(mod$cov.unscaled)
```

[7] A choice of parameters that depend on the data for the Metropolis–Hastings proposal is completely valid, both from an MCMC point of view (meaning that this is not a self-tuning algorithm) and from a Bayesian point of view (since the parameters of the proposal are not those of the prior).

```
for (i in 2:niter){
  tildebeta=rmnorm(1,beta[i-1,],scale*Sigma2)
  llr=probitll(tildebeta,y,X)-probitll(beta[i-1,],y,X)
  if (runif(1)<=exp(llr)) beta[i,]=tildebeta
  else beta[i,]=beta[i-1,]
  }
 beta
 }
```

It takes advantage of the multivariate normal generator `rmnorm`, part of the package `mnormt` that caters to the multivariate normal distribution.

For **bank**, using a probit modeling with no intercept over the four measurements, we tested three different scales, namely $\tau = 1, 0.1, 10$, by running Algorithm 4.7 over 10,000 iterations. Looking both at the raw sequences and at the autocorrelation graphs, it appears that the best mixing behavior is associated with $\tau = 1$. Figure 4.4 illustrates the output of the simulation run in that case.[8] Using a burn-in range of 1,000 iterations, the averages of the parameters over the last 9,000 iterations are equal to $-1.2193, 0.9540, 0.9795$, and 1.1481, respectively. A plug-in estimate of the predictive probability of a counterfeit banknote is therefore

$$\hat{p}_i = \Phi\left(-1.2193 x_{i1} + 0.9540 x_{i2} + 0.9795 x_{i3} + 1.1481 x_{i4}\right).$$

For instance, according to this equation, a banknote of length 214.9 mm, left-edge width 130.1 mm, right-edge width 129.9 mm, and bottom margin width 9.5 mm is counterfeited with probability

$$\Phi\left(-1.1293 \times 214.9 + \ldots + 1.1481 \times 9.5\right) \approx 0.5917.$$

While the plug-in representation above gives an immediate evaluation of the predictive probability, a better approximation to this probability function is provided by the average over the iterations of the current predictive probabilities, $\Phi\left(\beta_1^{(t)} x_{i1} + \beta_2^{(t)} x_{i2} + \beta_3^{(t)} x_{i3} + \beta_4^{(t)} x_{i4}\right)$. It is easily derived from the output of the `hmflatprobit` function.

4.3.2 Noninformative *G*-Priors

Following the principles discussed in earlier chapters (see, e.g., Chap. 3), a flat prior on $\boldsymbol{\beta}$ is not appropriate for comparison purposes since we cannot validate the corresponding Bayes factors. In a variable selection setup, we thus need to replace the flat prior with, e.g., a hierarchical prior,

[8] We do not include the graphs for the other values of τ, but the curious reader can check that there is indeed a clear difference with the case $\tau = 1$.

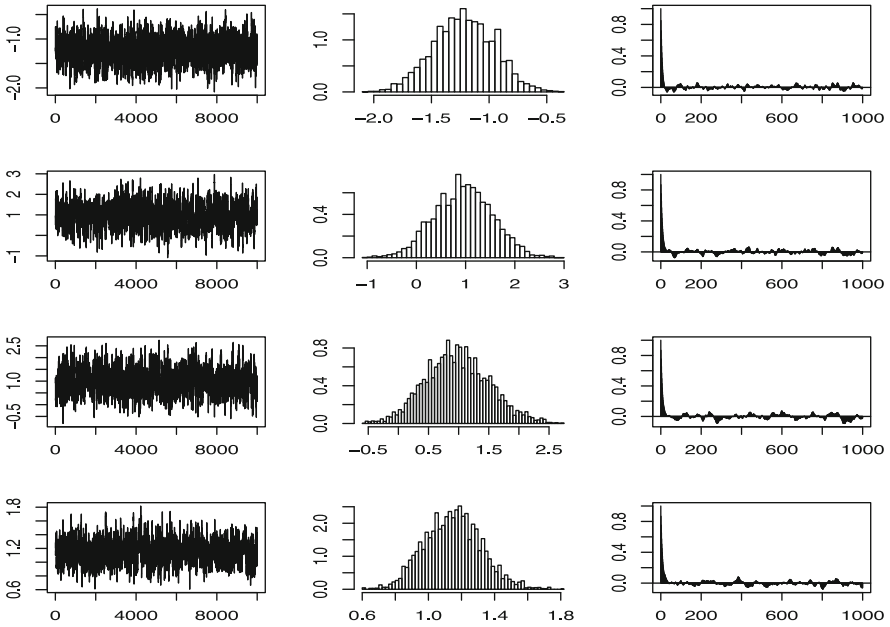


Fig. 4.4. Dataset **bank**: Estimation of the probit coefficients via Algorithm 4.7 and a flat prior. *Left*: β_i's ($i = 1, \ldots, 4$); *center*: histogram over the last 9,000 iterations; *right*: autocorrelation over the last 9,000 iterations

$$\boldsymbol{\beta}|\sigma^2 \sim \mathcal{N}_k\left(\mathbf{0}_k, \sigma^2(\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\right) \quad \text{and} \quad \pi(\sigma^2) \propto \sigma^{-2}\,,$$

inspired by the normal linear regression model.[9] Integrating out σ^2 in this joint prior then leads to

$$\pi(\boldsymbol{\beta}) \propto |\mathbf{X}^{\mathsf{T}}\mathbf{X}|^{1/2}\Gamma(k/2)\left(\boldsymbol{\beta}^{\mathsf{T}}(\mathbf{X}^{\mathsf{T}}\mathbf{X})\boldsymbol{\beta}\right)^{-k/2}\pi^{-k/2}\,,$$

which is clearly improper. Nonetheless, if we consider the *same* hierarchical prior for a submodel associated with a subset of the predictor variables in $\mathbf{X}$, associated with the *same* variance factor σ^2, the marginal distribution of $\mathbf{y}$ then depends on the *same* unknown multiplicative constant as the full model, and this constant cancels in the corresponding Bayes factor. This is exactly the same idea as for Zellner's noninformative G-prior, see Sect. 3.4.3.

The corresponding posterior distribution of $\boldsymbol{\beta}$ is

$$\pi(\boldsymbol{\beta}|\mathbf{y}) \propto |\mathbf{X}^{\mathsf{T}}\mathbf{X}|^{1/2}\Gamma(k/2)\left(\boldsymbol{\beta}^{\mathsf{T}}(\mathbf{X}^{\mathsf{T}}\mathbf{X})\boldsymbol{\beta}\right)^{-k/2}\pi^{-k/2}$$

[9] Note that the matrix $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is *not* the Fisher information matrix outside of the normal model. However, the (genuine) Fisher information matrix usually involves a function of $\boldsymbol{\beta}$ that prevents its use as a prior (inverse) covariance matrix on $\boldsymbol{\beta}$.

$$\times \prod_{i=1}^{n} \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})^{y_i} \left[1 - \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right]^{1-y_i} . \tag{4.4}$$

Note that we need to keep the "constant" terms $|\mathbf{X}^\mathsf{T}\mathbf{X}|^{1/2}$, $\Gamma(k/2)$, and $\pi^{-k/2}$, in this expression because they vary among submodels. To omit these terms would thus result in a bias in the computation of the Bayes factors.

Contrary to the linear regression setting and as for the flat prior in Sect. 4.3.1, neither the posterior distribution of $\boldsymbol{\beta}$ nor the marginal distribution of $\mathbf{y}$ can be derived analytically. We can however use exactly the same Metropolis–Hastings sampler as in Sect. 4.3.1, namely a random walk proposal based on the estimated Fisher information matrix for its scale and the MLE $\hat{\boldsymbol{\beta}}$ as its starting value.

For **bank**, the corresponding approximate Bayes estimate of $\boldsymbol{\beta}$ is given by

$$\mathbb{E}^\pi[\boldsymbol{\beta}|\mathbf{y}] \approx (-1.1552, 0.9200, 0.9121, 1.0820),$$

which slightly differs from the estimate found in Sect. 4.3.1 for the flat prior. This approximation was obtained by running the MH algorithm with scale $\tau^2 = 1$ over 10,000 iterations and averaging over the last 9,000 iterations. Figure 4.5 gives an assessment of the convergence of the MH scheme that does not vary very much compared with the previous figure.

We now address the specific problem of approximating the marginal distribution of $\mathbf{y}$ toward providing approximations to the Bayes factor and thus achieve the Bayesian equivalent of standard software to identify significant variables in the probit model. The marginal distribution of $\mathbf{y}$ is

$$f(\mathbf{y}) \propto |\mathbf{X}^\mathsf{T}\mathbf{X}|^{1/2}\, \pi^{-k/2}\Gamma(k/2) \int \left(\boldsymbol{\beta}^\mathsf{T}(\mathbf{X}^\mathsf{T}\mathbf{X})\boldsymbol{\beta}\right)^{-k/2} \times \prod_{i=1}^{n} \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})^{y_i} \left[1 - \Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right]^{1-y_i} \mathrm{d}\boldsymbol{\beta}\,,$$

which cannot be computed in closed form. We thus propose to use as a generic proxy an importance sampling approximation to this integral based on a normal approximation $\mathcal{N}_k(\hat{\boldsymbol{\beta}}, 2\,\hat{V})$ to $\pi(\boldsymbol{\beta}|\mathbf{y})$, where $\hat{\boldsymbol{\beta}}$ is the MCMC approximation of $\mathbb{E}^\pi[\boldsymbol{\beta}|\mathbf{y}]$ and $\hat{V}$ is the MCMC approximation[10] of $\mathbb{V}(\boldsymbol{\beta}|\mathbf{y})$. The corresponding estimate of the marginal distribution of $\mathbf{y}$ is then, up to a constant,

[10] The factor 2 in the covariance matrix allows some amount of overdispersion, which is always welcomed in importance sampling settings, if only for variance finiteness purposes.

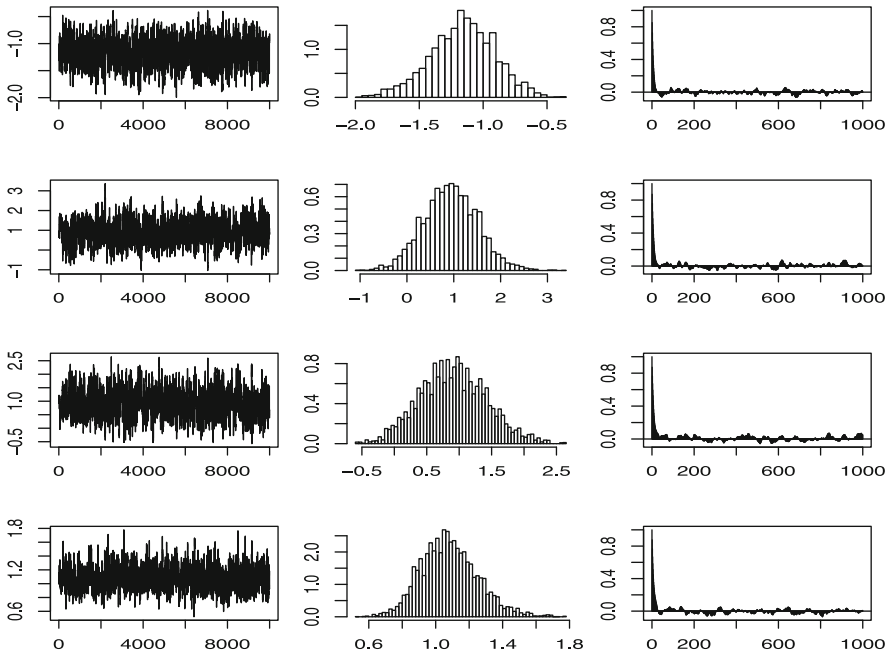


Fig. 4.5. Dataset **bank**: Same legend as Fig. 4.4 using an MH algorithm and a G-prior on $\boldsymbol{\beta}$

$$\frac{|\mathbf{X}^{\mathsf{T}}\mathbf{X}|^{1/2}}{\pi^{k/2}M}\sum_{m=1}^{M}\left({\boldsymbol{\beta}^{(m)}}^{\mathsf{T}}(\mathbf{X}^{\mathsf{T}}\mathbf{X})\boldsymbol{\beta}^{(m)}\right)^{-k/2}\prod_{i=1}^{n}\Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}^{(m)})^{y_i}\left[1-\Phi(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}^{(m)})\right]^{1-y_i}$$

$$\times\,|\hat{V}|^{1/2}(4\pi)^{k/2}\,e^{(\boldsymbol{\beta}^{(m)}-\hat{\boldsymbol{\beta}})^{\mathsf{T}}\hat{V}^{-1}(\boldsymbol{\beta}^{(m)}-\hat{\boldsymbol{\beta}})/4}\,, \tag{4.5}$$

where the $\boldsymbol{\beta}^{(m)}$'s are simulated from the $\mathcal{N}_k(\hat{\boldsymbol{\beta}}, 2\,\hat{V})$ importance distribution.

If we consider a linear restriction on $\boldsymbol{\beta}$ such as $H_0 : R\boldsymbol{\beta} = r$, with $r \in \mathbb{R}^q$ and R a $q \times k$ matrix of rank q, the submodel is associated with the likelihood

$$\ell(\boldsymbol{\beta}^0|\mathbf{y}) \propto \prod_{i=1}^{n}\Phi(\mathbf{x}_0^{i\mathsf{T}}\boldsymbol{\beta}^0)^{y_i}\left[1-\Phi(\mathbf{x}_0^{i\mathsf{T}}\boldsymbol{\beta}^0)\right]^{1-y_i}\,,$$

where $\boldsymbol{\beta}^0$ is $(k-q)$-dimensional and $\mathbf{X}_0$ and $\mathbf{x}_0$ are linear transforms of $\mathbf{X}$ and $\mathbf{x}$ of dimensions $(n, k-q)$ and $(k-q)$, respectively. Under the G-prior

$$\boldsymbol{\beta}^0|\sigma^2 \sim \mathcal{N}_{k-q}\left(\mathbf{0}_{k-q}, \sigma^2(\mathbf{X}_0^{\mathsf{T}}\mathbf{X}_0)^{-1}\right) \quad \text{and} \quad \pi(\sigma^2) \propto \sigma^{-2}\,,$$

the marginal distribution of $\mathbf{y}$ is of the same type as in the unconstrained case, namely,

$$f(\mathbf{y}) \propto |\mathbf{X}_0^{\mathsf{T}}\mathbf{X}_0|^{1/2}\pi^{-(k-q)/2}\Gamma\{(k-q)/2\}\int \left\{(\boldsymbol{\beta}^0)^{\mathsf{T}}(\mathbf{X}_0^{\mathsf{T}}X_0)\boldsymbol{\beta}^0\right\}^{-(k-q)/2}$$

$$\times \prod_{i=1}^{n} \Phi(\mathbf{x}_0^{i\mathsf{T}}\boldsymbol{\beta}^0)^{y_i}\left[1-\Phi(\mathbf{x}_0^{i\mathsf{T}}\boldsymbol{\beta}^0)\right]^{1-y_i} \mathrm{d}\boldsymbol{\beta}^0\,.$$

Once again, if we first run an MCMC sampler for the posterior of $\boldsymbol{\beta}^0$ for this submodel, it provides both parameters of a normal importance distribution and thus allows an approximation of the marginal distribution of $\mathbf{y}$ in the submodel in all ways similar to (4.5).

For **bank**, if we want to test the null hypothesis $H_0 : \beta_1 = \beta_2 = 0$, we obtain the Bayes factor $B_{10}^{\pi} = 8916.0$ via the importance sampling approximation of (4.5). We use the following R commands, which again borrow functions like `dmnorm` and `rmnorm` from the package `mnormt`,

```
# full model
mkprob=apply(noinfprobit,2,mean)
vkprob=var(noinfprobit)
simk=rmnorm(100000,mkprob,2*vkprob)
usk=probitnoinflpost(simk,y,X[,2:5])-
    dmnorm(simk,mkprob,2*vkprob,log=T)
# null model
noinfprobit0=hmnoinfprobit(10000,y,X[,4:5],1)
mk0=apply(noinfprobit0,2,mean)
vk0=var(noinfprobit0)
simk0=rmnorm(100000,mk0,2*vk0)
usk0=probitnoinflpost(simk0,y,X[,4:5])-
     dmnorm(simk0,mk0,2*vk0,log=T)
# Bayes factor
bf0probit=mean(exp(usk))/mean(exp(usk0))
```

Using Jeffreys' scale of evidence, since $\log_{10}(B_{10}^{\pi}) = 3.950$, the posterior distribution is strongly against H_0.

More generally, we can produce a Bayesian regression output, programmed in R, that mimics the standard software output for generalized linear models. Along with the estimates of the β_i's, given by their posterior expectation, we include the posterior variances of the β_i's, also derived from the MCMC sample, and the log Bayes factors $\log_{10}\left(B_{10}^{i}\right)$ corresponding to the null hypotheses $H_0 : \beta_i = 0$. As above, the Bayes factors are computed by importance sampling based on 100,000 simulations. The stars are related to Jeffreys' scale of evidence.

For **bank**, the corresponding outcome is

```
            Estimate  Post. var.  log10(BF)

X1          -1.1552   0.0631       4.5844 (****)
X2           0.9200   0.3299      -0.2875
X3           0.9121   0.2595      -0.0972
X4           1.0820   0.0287      15.6765 (****)

evidence against H0: (****) decisive, (***) strong,
(**) substantial, (*) poor
```

Although these Bayes factors cannot be used simultaneously, an informal conclusion is that the significant variables for the identification of counterfeited banknotes are X_1 and X_4.

4.3.3 About Informative Prior Analyses

In the setting of probit (and other generalized linear) models, it is unrealistic to expect practitioners to come up with precise prior information about the parameters $\boldsymbol{\beta}$. There exists nonetheless an amenable approach to prior information through what is called the *conditional mean family of prior distributions*. The intuition behind this approach is that prior beliefs about the probabilities p_i can be assessed to some extent by the practitioners for *particular values* of the explanatory variables $x_{1i}, \ldots, x_{ki}$. Once this information is taken into account, a corresponding prior can be derived for the parameter vector $\boldsymbol{\beta}$. This technique is certainly one of the easiest methods of incorporating subjective prior information into the processing of the binary regression problem, especially because it appeals to practitioners for whom the $\boldsymbol{\beta}$'s have, at best, a virtual meaning.

Starting with k explanatory variables, we derive the subjective prior information from k different values[11] of the covariate vector, denoted by $\tilde{\mathbf{x}}^1, \ldots, \tilde{\mathbf{x}}^k$. For each of these values, the practitioner is asked to specify two things:

1. a prior guess g_i at the probability of success p_i associated with $\mathbf{x}^i$; and
2. an assessment of her or his certainty about that guess translated as a number K_i of equivalent "prior observations."[12] This question can be expressed as "On how many imaginary observations did you build this guess?"

Both quantities can be turned into a formal prior density on $\boldsymbol{\beta}$ by imposing a beta prior distribution on p_i with parameters $K_i g_i$ and $K_i(1 - g_i)$ since the mean of a $\mathcal{B}e(a, b)$ distribution is $a/(a + b)$. If we make the additional

[11] The theoretical motivation for setting the number of covariate vectors equal to the dimension of $\boldsymbol{\beta}$ will be made clear below.

[12] This technique is called the device of *imaginary observations* and was proposed by the Italian statistician Bruno de Finetti for prior elicitation.

assumption that the k probabilities $p_1, \ldots, p_k$ are a priori independent (which clearly does not hold since they all depend on the same $\boldsymbol{\beta}$!), their joint density is

$$\pi(p_1, \ldots, p_k) \propto \prod_{i=1}^{k} p_i^{K_i g_i - 1} (1 - p_i)^{K_i (1 - g_i) - 1} . \tag{4.6}$$

Now, if we relate the probabilities p_i to the parameter $\boldsymbol{\beta}$, conditional on the covariate vectors $\tilde{\mathbf{x}}^1, \ldots, \tilde{\mathbf{x}}^k$, by $p_i = \Phi(\tilde{\mathbf{x}}^{i\mathsf{T}} \boldsymbol{\beta})$, we conclude that the corresponding distribution on $\boldsymbol{\beta}$ is

$$\pi(\boldsymbol{\beta}) \propto \prod_{i=1}^{k} \Phi(\tilde{\mathbf{x}}^{i\mathsf{T}} \boldsymbol{\beta})^{K_i g_i - 1} \left[1 - \Phi(\tilde{\mathbf{x}}^{i\mathsf{T}} \boldsymbol{\beta})\right]^{K_i (1 - g_i) - 1} \varphi(\tilde{\mathbf{x}}^{i\mathsf{T}} \boldsymbol{\beta}) .$$

This change of variable explains why we needed exactly k different covariate vectors in the prior assessment.

This intuitive approach to prior modeling is also interesting from a computational point of view since the corresponding posterior distribution

$$\begin{aligned} \pi(\boldsymbol{\beta}|\mathbf{y}) \propto & \prod_{i=1}^{n} \Phi(\mathbf{x}^{i\mathsf{T}} \boldsymbol{\beta})^{y_i} \left[1 - \Phi(\mathbf{x}^{i\mathsf{T}} \boldsymbol{\beta})\right]^{1 - y_i} \\ & \times \prod_{j=1}^{k} \Phi(\tilde{\mathbf{x}}^{j\mathsf{T}} \boldsymbol{\beta})^{K_j g_j - 1} \left[1 - \varphi(\tilde{\mathbf{x}}^{j\mathsf{T}} \boldsymbol{\beta})\right]^{K_j (1 - g_j) - 1} \Phi(\tilde{\mathbf{x}}^{j\mathsf{T}} \boldsymbol{\beta}) \end{aligned}$$

is of almost exactly the same type as the posterior distributions in both noninformative modelings above. The main difference stands in the product of the Jacobian terms $\varphi(\tilde{\mathbf{x}}^{j\mathsf{T}} \boldsymbol{\beta})$ $(1 \leq j \leq k)$, but

$$\prod_{j=1}^{k} \varphi(\tilde{\mathbf{x}}^{j\mathsf{T}} \boldsymbol{\beta}) \propto \exp\left\{ -\sum_{j=1}^{k} (\tilde{\mathbf{x}}^{j\mathsf{T}} \boldsymbol{\beta})^2 / 2 \right\} = \exp\left\{ -\boldsymbol{\beta}^{\mathsf{T}} \left[\sum_{j=1}^{k} \tilde{\mathbf{x}}^j \tilde{\mathbf{x}}^{j\mathsf{T}} \right] \boldsymbol{\beta} / 2 \right\}$$

means that, if we forget about the -1's in the exponents, this posterior distribution corresponds to a regular posterior distribution for the probit model when adding to the observations $(y_1, \mathbf{x}^1), \ldots, (y_n, \mathbf{x}^n)$ the pseudo-observations[13] $(g_1, \tilde{\mathbf{x}}^1), \ldots, (g_1, \tilde{\mathbf{x}}^1), \ldots, (g_k, \tilde{\mathbf{x}}^k), \ldots, (g_k, \tilde{\mathbf{x}}^k)$, where each pair $(g_i, \tilde{\mathbf{x}}^i)$ is repeated K_i times and when using the G-prior

$$\boldsymbol{\beta} \sim \mathcal{N}_k \left(\mathbf{0}_k, \left[\sum_{j=1}^{k} \tilde{\mathbf{x}}^j \tilde{\mathbf{x}}^{j\mathsf{T}} \right]^{-1} \right) .$$

Therefore, Algorithm 4.7 need not be adapted to this case.

[13] Note that the fact that the g_j's do not take their values in $\{0, 1\}$ but rather in $(0, 1)$ does not create any difficulty in the implementation of Algorithm 4.7.

4.4 The Logit Model

We now reproduce some of the developments of the previous section in the case of the logit model, as defined in Sect. 4.1.2, not because there exist notable differences with either the processing or the conclusions of the probit model but rather because there is hardly any difference! For instance, Algorithm 4.7 can also be used for this model, while based on the same proposal, by simply modifying the definition of $\pi(\boldsymbol{\beta}|\mathbf{y})$, since the likelihood is now

$$\ell(\boldsymbol{\beta}|\mathbf{y}) = \exp\left\{\sum_{i=1}^{n} y_i \, \mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}\right\} \Big/ \prod_{i=1}^{n} \left[1 + \exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right] . \tag{4.7}$$

The R function that computes the log-likelihood of the logit model is

```
logitll=function(beta,y,X){
 if (is.matrix(beta)==F) beta=as.matrix(t(beta))
 n=dim(beta)[1]
 pll=rep(0,n)
 for (i in 1:n){
   lF1=plogis(X%*%beta[i,],log=T)
   lF2=plogis(-X%*%beta[i,],log=T)
   pll[i]=sum(y*lF1+(1-y)*lF2)
   }
 pll
 }
```

That both models can be processed in a very similar manner means, for instance, that they can be easily compared when one is uncertain about which link function to adopt. The Bayes factor used in the comparison of the probit and logit models is directly derived from the importance sampling experiments described for the probit model. Note also that, while the values of the parameter $\boldsymbol{\beta}$ differ between the two models, a subjective prior modeling as in Sect. 4.3.3 can be conducted simultaneously for both models, the only difference occurring for the change of variables from $(p_1, \ldots, p_k)$ to $\boldsymbol{\beta}$.

If we use a flat prior on $\boldsymbol{\beta}$, the posterior distribution proportional to (4.7) can be inserted directly in Algorithm 4.7 to produce a sample approximately distributed from this posterior (assuming it exists, which means observing a sufficiently large and diverse sample). The corresponding R code is

```
hmflatlogit=function(niter,y,X,scale){
 p=dim(X)[2]
 mod=summary(glm(y~-1+X,family=binomial(link="logit")))
 beta=matrix(0,niter,p)
 beta[1,]=as.vector(mod$coeff[,1])
 Sigma2=as.matrix(mod$cov.unscaled)
 for (i in 2:niter){
   tildebeta=rmvn(1,beta[i-1,],scale*Sigma2)
```

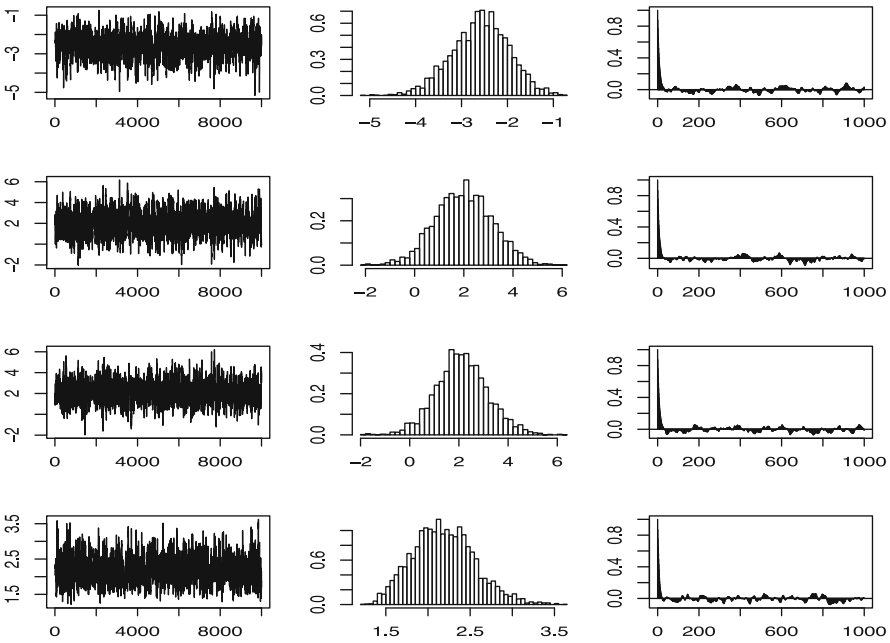


Fig. 4.6. Dataset **bank**: Estimation of the logit coefficients via Algorithm 4.7 under a flat prior. *Left*: β_i's ($i = 1, \ldots, 4$); *center*: histogram over the last 9,000 iterations; *right*: autocorrelation over the last 9,000 iterations

```
  llr=logitll(tildebeta,y,X)-logitll(beta[i-1,],y,X)
     if (runif(1)<=exp(llr)) beta[i,]=tildebeta
  else beta[i,]=beta[i-1,]
  }
beta
}
```

For **bank**, Fig. 4.6 summarizes the results of running Algorithm 4.7 with the scale factor equal to $\tau = 1$: There is no clear difference between these graphs and those of earlier figures, except for a slight increase in the skewness of the histograms of the β_i's. (Obviously, this does not necessarily reflect a different convergence behavior but possibly a different posterior behavior since we are not dealing with the *same* posterior distribution.) The MH approximation—based on the last 9,000 iterations—of the Bayes estimate of $\boldsymbol{\beta}$ is equal to $(-2.5888, 1.9967, 2.1260, 2.1879)$. We can note the numerical difference between these values and those produced by the probit model. The sign and the relative magnitudes of the components are, however, very similar. For comparison purposes, consider the plug-in estimate of the predictive probability of a counterfeit banknote,

$$\hat{p}_i = \frac{\exp\left(-2.5888x_{i1} + 1.9967x_{i2} + 2.1260x_{i3} + 2.1879x_{i4}\right)}{1 + \exp\left(-2.5888x_{i1} + 1.9967x_{i2} + 2.1260x_{i3} + 2.1879x_{i4}\right)}.$$

Using this approximation, a banknote of length 214.9 mm, of left-edge width 130.1 mm, of right-edge width 129.9 mm, and of bottom margin width 9.5 mm is counterfeited with probability

$$\frac{\exp\left(-2.5888 \times 130.1 + \ldots + 2.1879 \times 9.5\right)}{1 + \exp\left(-2.5888 \times 130.1 + \ldots + 2.1879 \times 9.5\right)} \approx 0.5963\,.$$

This estimate of the probability is therefore very close to the estimate derived from the probit modeling, which was equal to 0.5917 (especially if we take into account the uncertainties associated both with the MCMC experiments and with the plug-in shortcut).

For model comparison purposes and the computation of Bayes factors, we can also use the same G-prior as for the probit model and thus multiply (4.7) by $|\mathbf{X}^{\mathsf{T}}\mathbf{X}|^{1/2}\Gamma(k/2)\left(\boldsymbol{\beta}^{\mathsf{T}}(\mathbf{X}^{\mathsf{T}}\mathbf{X})\boldsymbol{\beta}\right)^{-k/2}\pi^{-k/2}$. The MH implementation obviously remains the same.

For **bank**, Fig. 4.7 once more summarizes the output of the MH scheme over 10,000 iterations. Since we observe the same skewness in the histograms as in Fig. 4.6, this feature is most certainly due to the corresponding posterior distribution rather than to a deficiency in the convergence of the algorithm.)

We can repeat the test of the null hypothesis $H_0 : \beta_1 = \beta_2 = 0$ already done for the probit model and then obtain an approximate Bayes factor of $B^{\pi}_{10} = 16972.3$, with the same conclusion as earlier (although with twice as large an absolute value. We can also take advantage of the output software programmed for the probit model to produce the following summary:

```
             Estimate  Post. var.  log10(BF)

X1           -2.3970   0.3286       4.8084 (****)
X2            1.6978   1.2220      -0.2453
X3            2.1197   1.0094      -0.1529
X4            2.0230   0.1132      15.9530 (****)

evidence against H0: (****) decisive, (***) strong,
(**) substantial, (*) poor
```

Therefore, the most important covariates are again X_1 and X_4.

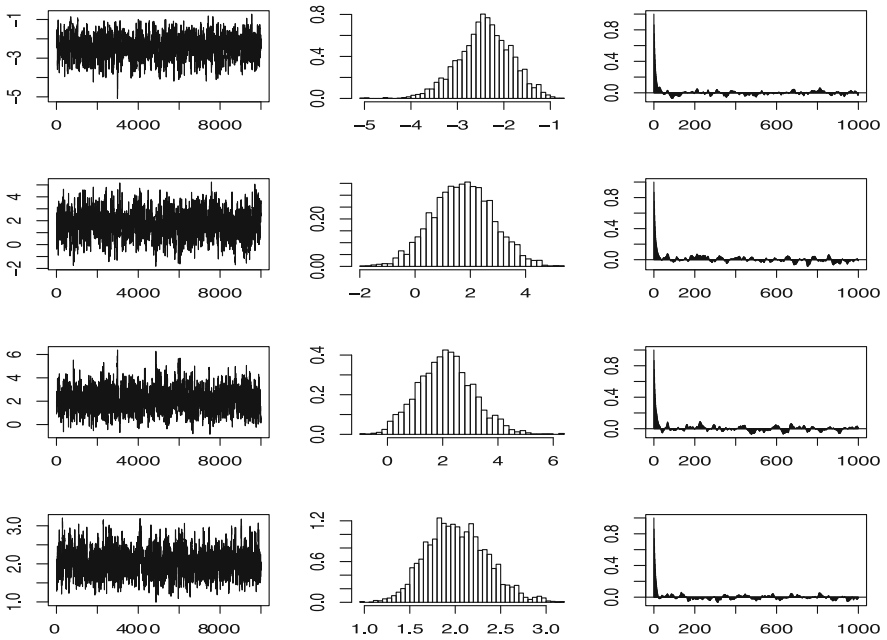


Fig. 4.7. Dataset **bank**: Same legend as Fig. 4.6 using an MH algorithm and a G-prior on $\boldsymbol{\beta}$

4.5 Log-Linear Models

We conclude this chapter with an application of generalized linear modeling to the case of factors, already mentioned in Sect. 3.1. A standard approach to the analysis of associations (or dependencies) between *categorical* variables (that is, variables that take a finite number of values) is to use *log-linear models*. These models are special cases of generalized linear models connected to the Poisson distribution, and their name stems from the fact that they have traditionally been based on the logarithmic link function.

4.5.1 Contingency Tables

In such models, a sufficient statistic is the *contingency table*, which is a multiple-entry table made up of the cross-classified counts for the different categorical variables. There is much literature on contingency tables, including for instance Whittaker (1990) and Agresti (1996), because the corresponding models are quite handy both in the social sciences and in survey processing, where the observables are always reduced to a finite number of values.

The **airquality** dataset was obtained from the New York State Department of Conservation (ozone data) and from the American National Weather Service (meteorological data) and is part of the datasets contained in R (Chambers et al., 1983) and available as

```
> air=data(airquality)
```

This dataset involves two repeated measurements over 111 consecutive days of 1973, namely the mean ozone u (in parts per billion) from 1 pm to 3 pm at Roosevelt Island, the maximum daily temperature v (in degrees F) at La Guardia Airport, and, in addition, the month w (coded from 5 for May to 9 for September). If we discretize the measurements u and v into dichotomous variables (using the empirical median as the cutting point), we obtain the following three-way contingency table of counts per combination of the three (discretize) factors:

```
             month  5  6  7  8  9
ozone    temp
[1,31]   [57,79] 17  4  2  5 18
         (79,97]  0  2  3  3  2
(31,168] [57,79]  6  1  0  3  1
         (79,97]  1  2 21 12  8
```

This contingency table thus has $5 \times 2 \times 2 = 20$ entries deduced from the number of categories of the three factors, among which some are zero because the corresponding combination of the three factors has not been observed in the study.

Each term in the table being an integer, it can then in principle be modeled as a Poisson variable. If we denote the counts by $\mathbf{y} = (y_1, \ldots, y_n)$, where $i = 1, \ldots, n$ is an arbitrary way of indexing the cells of the table, we can thus assume that $y_i \sim \mathscr{P}(\mu_i)$. Obviously, the likelihood

$$\ell(\mu|\mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\mu_i!} \mu_i^{y_i} \exp(-\mu_i)\,,$$

where $\mu = (\mu_1, \ldots, \mu_n)$, shows that the model is *saturated*, namely that no structure can be exhibited because there are as many parameters as there are entries in the table. To exhibit any structure, we need to constrain the μ_i's and do so via a GLM whose covariate matrix $\mathbf{X}$ is directly derived from the contingency table itself. If some entries are structurally equal to zero (as for instance when crossing "number of pregnancies" with "male indicators"), these entries should be removed from the model.

An R function that corresponds to this log-linear model log-likelihood is

```
loglinll=function(beta,y,X){
  if (is.matrix(beta)==FALSE) beta=as.matrix(t(beta))
  n=dim(beta)[1]
  pll=rep(0,n)
  for (i in 1:n){
    lF=exp(X%*%beta[i,])
    pll[i]=sum(dpois(y,lF,log=T))
    }
  pll
  }
```

with again the use of `is.matrix` and `as.matrix` to allow for matricial calls to the `loglinll` function.

When we constrain the mean parameters μ_i of a log-linear model to satisfy

$$\log(\mu_i) = \mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}\,,$$

the covariate vector $\mathbf{x}^i$ is rather peculiar in that it is constituted *only* of indicators. The so-called *incidence matrix* $\mathbf{X}$ with rows equal to the $\mathbf{x}^i$'s is thus such that its elements are all zeros or ones. Given a contingency table, the choice of indicator variables to include in $\mathbf{x}^i$ can vary, depending on what is deemed (or found) to be an important relation between some categorical variables. For instance, suppose that there are three categorical variables, u, v, and w as in airquality, and that u takes I values, v takes J values, and w takes K values. If we only include the indicators for the values of the three categorical variables in $\mathbf{X}$, we have

$$\log(\mu_\tau) = \sum_{b=1}^{I} \beta_b^u \mathbb{I}_b(u_\tau) + \sum_{b=1}^{J} \beta_b^v \mathbb{I}_b(v_\tau) + \sum_{b=1}^{K} \beta_b^w \mathbb{I}_b(w_\tau)\,;$$

that is, $(1 \le i \le I,\, 1 \le j \le J,\, 1 \le k \le K)$,

$$\log(\mu_{l(i,j,k)}) = \beta_i^u + \beta_j^v + \beta_k^w$$

$(1 \le i \le I,\, 1 \le j \le J,\, 1 \le k \le K)$, where $l(i,j,k)$ corresponds to the index of the (i,j,k) entry in the table, namely the case when $u = i$, $v = j$, and $w = k$. Similarly, the saturated log-linear model corresponds to the use of one indicator per entry of the table; that is $1 \le i \le I$, $1 \le j \le J$, $1 \le k \le K$),

$$\log(\mu_{l(i,j,k)}) = \beta_{ijk}^{uvw}\,.$$

For comparative reasons that will very soon become apparent, and by analogy with analysis of variance (ANOVA) conventions, we can also over-parameterize this representation as

$$\log(\mu_{l(i,j,k)}) = \lambda + \lambda_i^u + \lambda_j^v + \lambda_k^w + \lambda_{ij}^{uv} + \lambda_{ik}^{uw} + \lambda_{jk}^{vw} + \lambda_{ijk}^{uvw}\,, \tag{4.8}$$

where λ appears as the overall or reference average effect, λ_i^u appears as the marginal discrepancy (against the reference effect λ) when $u = i$, λ_{ij}^{uv} as the interaction discrepancy (against the added effects $\lambda + \lambda_i^u + \lambda_j^v$) when $(u, v) = (i, j)$, etc.

Using the representation (4.8) is quite convenient because it allows a straightforward parameterization of the nonsaturated models, which then appear as submodels of (4.8) where some groups of parameters are null. For example,

1. if both categorical variables v and w are irrelevant, then
$$\log(\mu_{l(i,j,k)}) = \lambda + \lambda_i^u ;$$
2. if all three categorical variables are mutually independent, then
$$\log(\mu_{l(i,j,k)}) = \lambda + \lambda_i^u + \lambda_j^v + \lambda_k^w ;$$
3. if u and v are associated but are both independent of w, then
$$\log(\mu_{l(i,j,k)}) = \lambda + \lambda_i^u + \lambda_j^v + \lambda_k^w + \lambda_{ij}^{uv} ;$$

(iv) if u and v are conditionally independent given w, then
$$\log(\mu_{l(i,j,k)}) = \lambda + \lambda_i^u + \lambda_j^v + \lambda_k^w + \lambda_{ik}^{uw} + \lambda_{jk}^{vw} ; \quad \text{and}$$

(v) if there is no three-factor interaction, then
$$\log(\mu_{l(i,j,k)}) = \lambda + \lambda_i^u + \lambda_j^v + \lambda_k^w + \lambda_{ij}^{uv} + \lambda_{ik}^{uw} + \lambda_{jk}^{vw} ,$$
which appears as the most complete submodel (or as the global model if the saturated model is not considered at all).

This representation naturally embeds log-linear modeling within a model choice perspective in that it calls for a selection of the most parsimonious submodel that remains compatible with the observations. This is clearly equivalent to a variable-selection problem of a special kind in the sense that *all* indicators related with the same association must remain or vanish *at once*. This specific feature means that there are much fewer submodels to consider than in a regular variable-selection problem.

As stressed above, the representation (4.8) is not identifiable. Although the following is not strictly necessary from a Bayesian point of view (since the Bayesian approach can handle nonidentifiable settings and still estimate properly identifiable quantities), it is customary to impose identifiability constraints on the parameters as in the ANOVA model. A common convention is to set to zero the parameters corresponding to the first category of each variable, which is equivalent to removing the indicator (or *dummy variable*) of the first category for each variable (or group of variables). For instance, for a 2×2 contingency table with two variables u and v, both having two categories, say 1 and 2, the constraint could be

$$\lambda_1^u = \lambda_1^v = \lambda_{11}^{uv} = \lambda_{12}^{uv} = \lambda_{21}^{uv} = 0\,.$$

For notational convenience, we assume below that $\boldsymbol{\beta}$ is the vector of the parameters once the identifiability constraint has been applied and that $\mathbf{X}$ is the indicator matrix with the corresponding columns removed.

4.5.2 Inference Under a Flat Prior

Even when using a noninformative flat prior on $\boldsymbol{\beta}$, $\pi(\boldsymbol{\beta}) \propto 1$, the posterior distribution

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{y}) &\propto \prod_{i=1}^{n} \left\{\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right\}^{y_i} \exp\{-\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\} \\ &= \exp\left\{\sum_{i=1}^{n} y_i\, \mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta} - \sum_{i=1}^{n} \exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right\} \\ &= \exp\left\{\left(\sum_{i=1}^{n} y_i\, \mathbf{x}^{i}\right)^{\mathsf{T}} \boldsymbol{\beta} - \sum_{i=1}^{n} \exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right\}\end{aligned}$$

is nonstandard and must be approximated by an MCMC algorithm. While the shape of this density differs from the posterior densities in the probit and logit cases, we can once more implement Algorithm 4.7 based on the normal Fisher approximation of the likelihood (whose parameters are again derived using the R `glm()` function as in

```
> mod=summary(glm(y~-1+X,family=poisson()))
```

which provides $\hat{\boldsymbol{\beta}}$ as mod$coeff[,1] and $\hat{\Sigma}$ as mod$cov.unscaled).

For **airquality**, we first consider the most general nonsaturated model, as described in Sect. 4.5.1. Taking into account the identifiability constraints, there are therefore

$$1+(2-1)+(2-1)+(5-1)+(2-1)\times(2-1)+(2-1)\times(5-1)+(2-1)\times(5-1)\,,$$

i.e., 16, free parameters in the model (to be compared with the 20 counts in the contingency table). Given the dimension of the simulated parameter, it is impossible to provide a complete picture of the convergence properties of the algorithm, and we represented in Fig. 4.8 the traces and histograms for the marginal posterior distributions of the parameters β_i based on 10,000 iterations using a scale factor equal to $\tau^2 = 0.5$. (This value was obtained by trial and error, producing a smooth trace for all parameters. Larger values of τ required a larger number of iterations since the acceptance rate was lower, as the reader can check using the BCoRe package.) Note that some of the traces represented in Fig. 4.8 show periodic patterns that indicate that more iterations could be necessary. However, the corresponding histograms remain

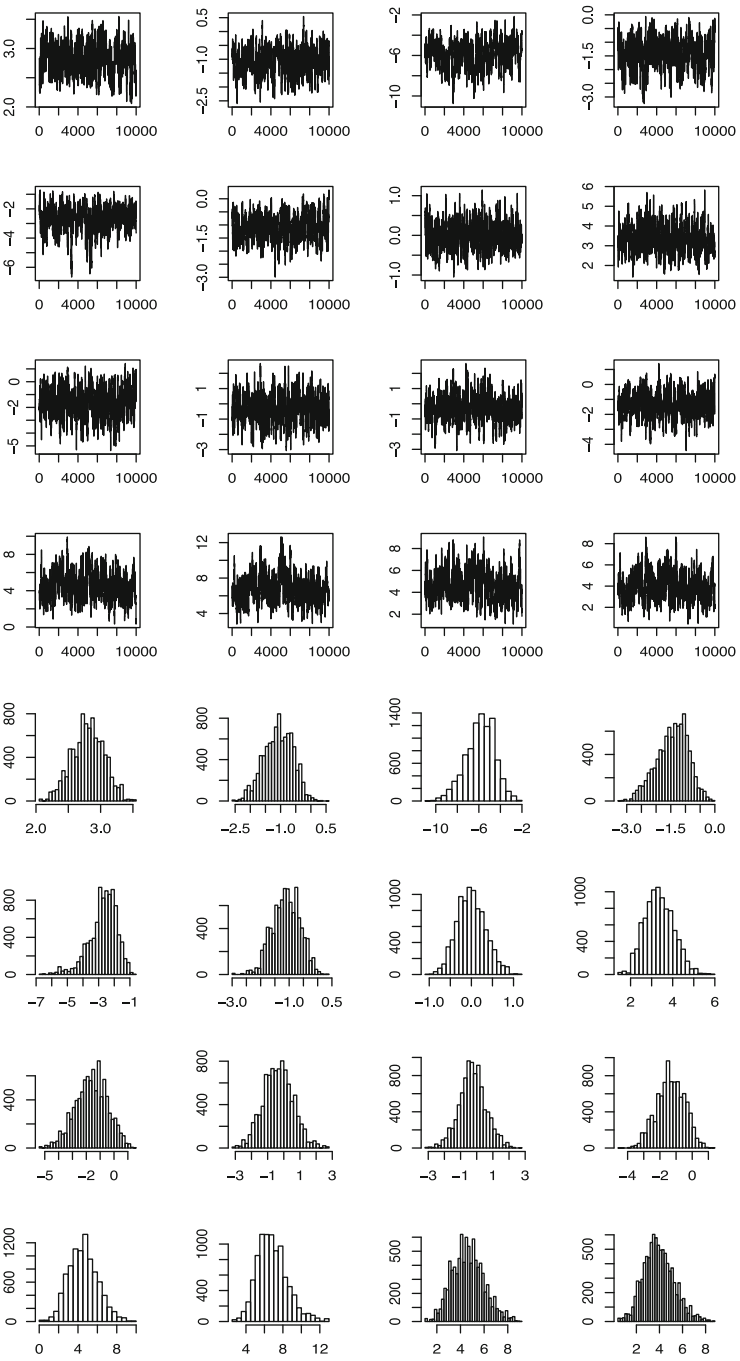


Fig. 4.8. Dataset airquality: Traces (*top*) and histograms (*bottom*) of the simulations from the posterior distributions of the components of $\boldsymbol{\beta}$ using a flat prior and a random walk Metropolis–Hastings algorithm with scale factor $\tau^2 = 0.5$ (same order row-wise as in Table 4.1)

Table 4.1. Dataset airquality: Bayes estimates of the parameter $\boldsymbol{\beta}$ using a random walk MH algorithm with scale factor $\tau^2 = 0.5$

Effect	Post. mean	Post. var.
λ	2.8041	0.0612
λ_2^u	–1.0684	0.2176
λ_2^v	–5.8652	1.7141
λ_2^w	–1.4401	0.2735
λ_3^w	–2.7178	0.7915
λ_4^w	–1.1031	0.2295
λ_5^w	–0.0036	0.1127
λ_{22}^{uv}	3.3559	0.4490
λ_{22}^{uw}	-1.6242	1.2869
λ_{23}^{uw}	–0.3456	0.8432
λ_{24}^{uw}	–0.2473	0.6658
λ_{25}^{uw}	–1.3335	0.7115
λ_{22}^{vw}	4.5493	2.1997
λ_{23}^{vw}	6.8479	2.5881
λ_{24}^{vw}	4.6557	1.7201
λ_{25}^{vw}	3.9558	1.7128

quite stable over iterations. Both the approximated posterior means and the posterior variances for the 16 parameters as deduced from the MCMC run are given in Table 4.1. A few histograms in Fig. 4.8 are centered at 0, signaling a potential lack of significance for the corresponding β_i's.

4.5.3 Model Choice and Significance of the Parameters

If we try to compare different levels of association (or interaction), or if we simply want to test the significance of some parameters β_i, the flat prior is once again inappropriate. The G-prior alternative proposed for the probit and logit models is still available, though, and we can thus replace the posterior distribution of the previous section with

$$\pi(\boldsymbol{\beta}|\mathbf{y}) \propto |\mathbf{X}^{\mathsf{T}}\mathbf{X}|^{1/2}\Gamma(k/2)\left(\boldsymbol{\beta}^{\mathsf{T}}(\mathbf{X}^{\mathsf{T}}\mathbf{X})\boldsymbol{\beta}\right)^{-k/2}\pi^{-k/2} \exp\left\{\left(\sum_{i=1}^{n} y_i\,\mathbf{x}^i\right)^{\mathsf{T}}\boldsymbol{\beta} - \sum_{i=1}^{n}\exp(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta})\right\} \tag{4.9}$$

as an alternative posterior.

Table 4.2. Dataset airquality: Metropolis–Hastings approximations of the posterior means under the G-prior

Effect	Post. mean	Post. var.
λ	2.7202	0.0603
λ_2^u	–1.1237	0.1981
λ_2^v	–4.5393	0.9336
λ_2^w	–1.4245	0.3164
λ_3^w	–2.5970	0.5596
λ_4^w	–1.1373	0.2301
λ_5^w	0.0359	0.1166
λ_{22}^{uv}	2.8902	0.3221
λ_{22}^{uw}	–0.9385	0.8804
λ_{23}^{uw}	0.1942	0.6055
λ_{24}^{uw}	0.0589	0.5345
λ_{25}^{uw}	–1.0534	0.5220
λ_{22}^{vw}	3.2351	1.3664
λ_{23}^{vw}	5.3978	1.3506
λ_{24}^{vw}	3.5831	1.0452
λ_{25}^{vw}	2.8051	1.0061

For **airquality** and the same model as in the previous analysis, namely the maximum nonsaturated model with 16 parameters, Algorithm 4.7 can be used with (4.9) as target and $\tau^2 = 0.5$ as the scale in the random walk. The result of this simulation over 10,000 iterations is presented in Fig. 4.9. The traces of the components of $\boldsymbol{\beta}$ show the same slow mixing as in Fig. 4.8, with similar occurrences of large deviances from the mean value that may indicate the weak identifiability of some of these parameters. Note also that the histograms of the posterior marginal distributions are rather close to those associated with the flat prior, as shown in Fig. 4.8. The MCMC approximations to the posterior means and the posterior variances are given in Table 4.2 for all 16 parameters, based on the last 9,000 iterations. While the first parameters are quite close to those provided by Table 4.1, the estimates of the interaction coefficients vary much more and are associated with much larger variances. This indicates that much less information is available within the contingency table about interactions, as can be expected.

If we now consider the very reason why this alternative to the flat prior was introduced, we are facing the same difficulty as in the probit case for the computation of the marginal density of $\mathbf{y}$. And, once again, the same solution applies: using an importance sampling experiment to approximate the integral works when the importance function is a multivariate normal (or t) distribution with mean (approximately) $\mathbb{E}[\boldsymbol{\beta}|\mathbf{y}]$ and covariance matrix (approximately) $2 \times \mathbb{V}(\boldsymbol{\beta}|\mathbf{y})$ using the Metropolis–Hastings approximations reported in Table 4.2. We can therefore approximate Bayes factors for testing all possible structures of the log-linear model.

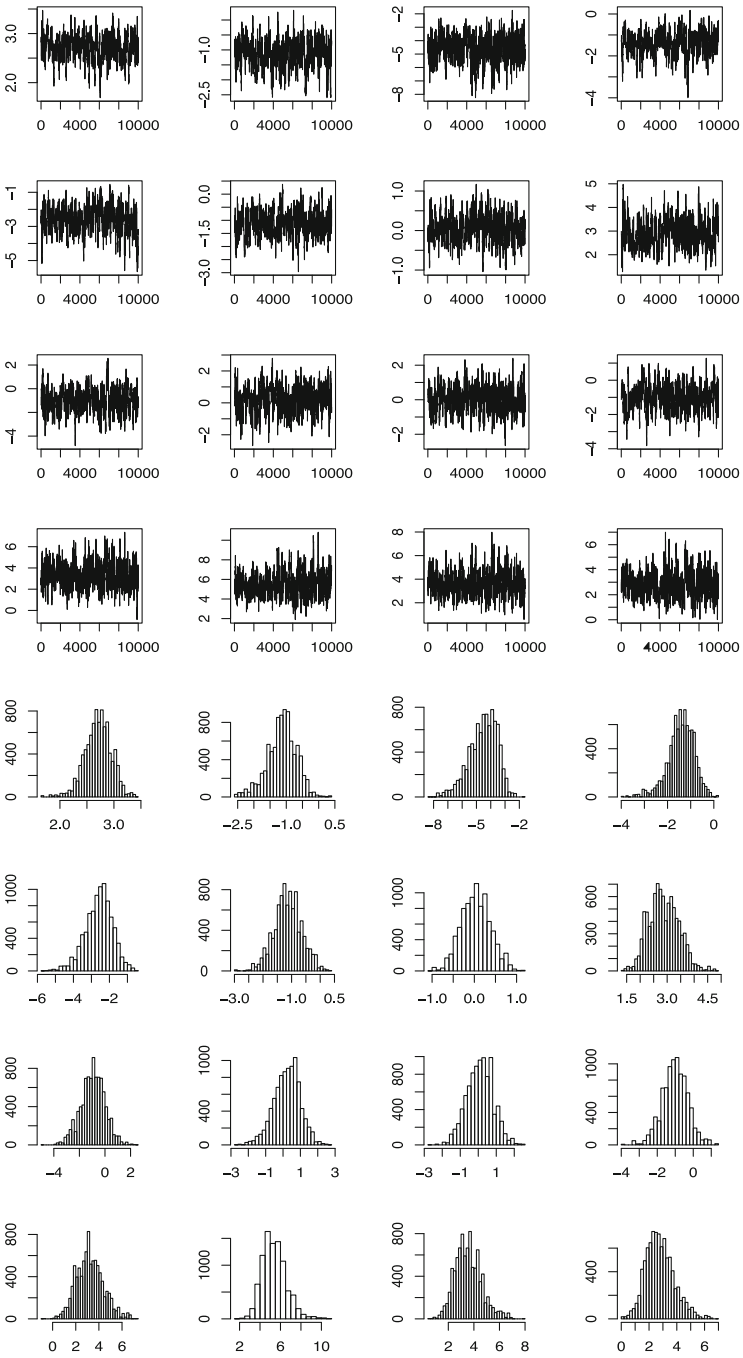


Fig. 4.9. Dataset airquality: Same legend as Fig. 4.8 for the posterior distribution (4.9) as target

For **airquality**, we illustrate this ability by testing the presence of two-by-two interactions between the three variables. We thus compare the largest non-saturated model with each submodel where one interaction is removed. An ANOVA-like output is

```
Effect log10(BF)

u:v     6.0983 (****)
u:w    -0.5732
v:w     6.0802 (****)

evidence against H0: (****) decisive, (***) strong,
(**) substantial, (*) poor
```

which means that the interaction between u and w (that is, ozone and month) is too small to be significant given all the other effects. (Note that it would be excessive to derive from this lack of significance a conclusion of independence between u and w because this interaction is conditional on all other interactions in the complete nonsaturated model.)

The above was obtained by the following R code: first we simulated an importance sample towards approximating the full model integrated likelihood

```
mklog=apply(noinfloglin,2,mean)
vklog=var(noinfloglin)
simk=rmnorm(100000,mklog,2*vklog)
usk=loglinnoinflpost(simk,counts,X)-
    dmnorm(simk,mklog,2*vklog,log=T)
```

then reproduced this computation for the three corresponding submodels, namely

```
noinfloglin1=hmnoinfloglin(10^4,counts,X[,-8],0.5)
mk1=apply(noinfloglin1,2,mean)
vk1=var(noinfloglin1)
simk1=rmnorm(100000,mk1,2*vk1)
usk1=loglinnoinflpost(simk1,counts,X[,-8])-
  dmnorm(simk1,mk1,2*vk1,log=T)
bf1loglin=mean(exp(usk))/mean(exp(usk1))
```

and the same pattern with

```
noinfloglin2=hmnoinfloglin(10^4,counts,cbind(X[,-(9:12)],0.5)
```

and

```
noinfloglin3=hmnoinfloglin(10^4,counts,X[,1:12],0.5)
```

4.6 Exercises

4.1 Show that, for the logistic regression model, the statistic $\sum_{i=1}^n y_i \mathbf{x}^i$ is sufficient when conditioning on the $\mathbf{x}^i$'s $(1 \le i \le n)$, and give the corresponding family of conjugate priors.

4.2 Show that the logarithmic link is the canonical link function in the case of the Poisson regression model.

4.3 Suppose $y_1, \ldots, y_k$ are independent Poisson $\mathscr{P}(\mu_i)$ random variables. Show that, conditional on $n = \sum_{i=1}^k y_i$,

$$\mathbf{y} = (y_1, \ldots, y_k) \sim \mathscr{M}_k(n; \alpha_1, \ldots, \alpha_k)\,,$$

and determine the α_i's.

4.4 For π the density of an inverse normal distribution with parameters $\theta_1 = 3/2$ and $\theta_2 = 2$,

$$\pi(x) \propto x^{-3/2} \exp(-3/2x - 2/x)\mathbb{I}_{x>0},$$

write down and implement an independence MH sampler with a Gamma proposal with parameters $(\alpha, \beta) = (4/3, 1)$ and $(\alpha, \beta) = (0.5\sqrt{4/3}, 0.5)$.

4.5 Consider x_1, x_2, and x_3 iid $\mathscr{C}(\theta, 1)$, and $\pi(\theta) \propto \exp(-\theta^2/100)$. Show that the posterior distribution of θ, $\pi(\theta|x_1, x_2, x_3)$, is proportional to

$$\exp(-\theta^2/100)[(1 + (\theta - x_1)^2)(1 + (\theta - x_2)^2)(1 + (\theta - x_3)^2)]^{-1}$$

and that it is trimodal when $x_1 = 0$, $x_2 = 5$, and $x_3 = 9$. Using a random walk based on the Cauchy distribution $\mathscr{C}(0, \sigma^2)$, estimate the posterior mean of θ using different values of σ^2. In each case, monitor the convergence.

4.6 Estimate the mean of a $\mathscr{G}a(4.3, 6.2)$ random variable using

1. direct sampling from the distribution via the R command
   `> x=rgamma(n,4.3,scale=6.2)`
2. Metropolis–Hastings with a $\mathscr{G}a(4, 7)$ proposal distribution;
3. Metropolis–Hastings with a $\mathscr{G}a(5, 6)$ proposal distribution.

In each case, monitor the convergence of the cumulated average.

4.7 For a standard normal distribution as target, implement a Hastings–Metropolis algorithm with a mixture of five random walks with variances $\sigma = 0.01, 0.1, 1, 10, 100$ and equal weights. Compare its output with the output of Fig. 4.3.

4.8 For the probit model under flat prior, find conditions on the observed pairs $(\mathbf{x}^i, y_i)$ for the posterior distribution above to be proper.

4.9 For the probit model under non-informative prior, find conditions on $\sum_i y_i$ and $\sum_i (1 - y_i)$ for the posterior distribution defined by (4.4) to be proper.

4.10 Include an intercept in the probit analysis of **bank** and run the corresponding version of Algorithm 4.7 to discuss whether or not the posterior variance of the intercept is high.

4.11 Using the latent variable representation of the probit model, introduce $z_i|\boldsymbol{\beta} \sim \mathcal{N}\left(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}, 1\right)$ $(1 \le i \le n)$ such that $y_i = \mathbb{I}_{z_i \le 0}$. Deduce that

$$z_i|y_i, \boldsymbol{\beta} \sim \begin{cases} \mathcal{N}_+\left(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}, 1, 0\right) & \text{if} \quad y_i = 1\,, \\ \mathcal{N}_-\left(\mathbf{x}^{i\mathsf{T}}\boldsymbol{\beta}, 1, 0\right) & \text{if} \quad y_i = 0\,, \end{cases}$$

where $\mathcal{N}_+(\mu, 1, 0)$ and $\mathcal{N}_-(\mu, 1, 0)$ are the normal distributions with mean μ and variance 1 that are left-truncated and right-truncated at 0, respectively. Check that those distributions can be simulated using the R commands

```
> xp=qnorm(runif(1)*pnorm(mu)+pnorm(-mu))+mu
> xm=qnorm(runif(1)*pnorm(-mu))+mu
```

Under the flat prior $\pi(\boldsymbol{\beta}) \propto 1$, show that

$$\boldsymbol{\beta}|\mathbf{y}, \mathbf{z} \sim \mathcal{N}_k\left((\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathsf{T}}\mathbf{z}, (\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}\right)\,,$$

where $\mathbf{z} = (z_1, \ldots, z_n)$, and derive the corresponding Gibbs sampler, sometimes called the *Albert–Chib* sampler. (*Hint*: A good starting point is the maximum likelihood estimate of $\boldsymbol{\beta}$.) Compare the application to **bank** with the output in Fig. 4.4. (*Note*: Account for differences in computing time.)

4.12 For the **bank** dataset and the probit model, compute the Bayes factor associated with the null hypothesis $H_0 : \beta_2 = \beta_3 = 0$.

4.13 In the case of the logit model—i.e., when $p_i = \exp \tilde{\mathbf{x}}^{i\mathsf{T}}\boldsymbol{\beta}/\{1 + \exp \tilde{\mathbf{x}}^{i\mathsf{T}}\boldsymbol{\beta}\}$ $(1 \le i \le k)$—derive the prior distribution on $\boldsymbol{\beta}$ associated with the prior (4.6) on $(p_1, \ldots, p_k)$.

4.14 Examine whether or not the sufficient conditions for propriety of the posterior distribution found in Exercise 4.9 for the probit model are the same for the logit model.

4.15 For the **bank** dataset and the logit model, compute the Bayes factor associated with the null hypothesis $H_0 : \beta_2 = \beta_3 = 0$ and compare its value with the value obtained for the probit model in Exercise 4.12.

4.16 Given a contingency table with four categorical variables, determine the number of submodels to consider.

4.17 In the case of a 2×2 contingency table with fixed total count $n = n_{11} + n_{12} + n_{21} + n_{22}$, we denote by $\theta_{11}, \theta_{12}, \theta_{21}, \theta_{22}$ the corresponding probabilities. If the prior on those probabilities is a Dirichlet $\mathcal{D}_4(1/2, \ldots, 1/2)$, give the corresponding marginal distributions of $\alpha = \theta_{11} + \theta_{12}$ and $\beta = \theta_{11} + \theta_{21}$. Deduce the associated Bayes factor if H_0 is the hypothesis of independence between the factors and if the priors on the margin probabilities α and β are those derived above.