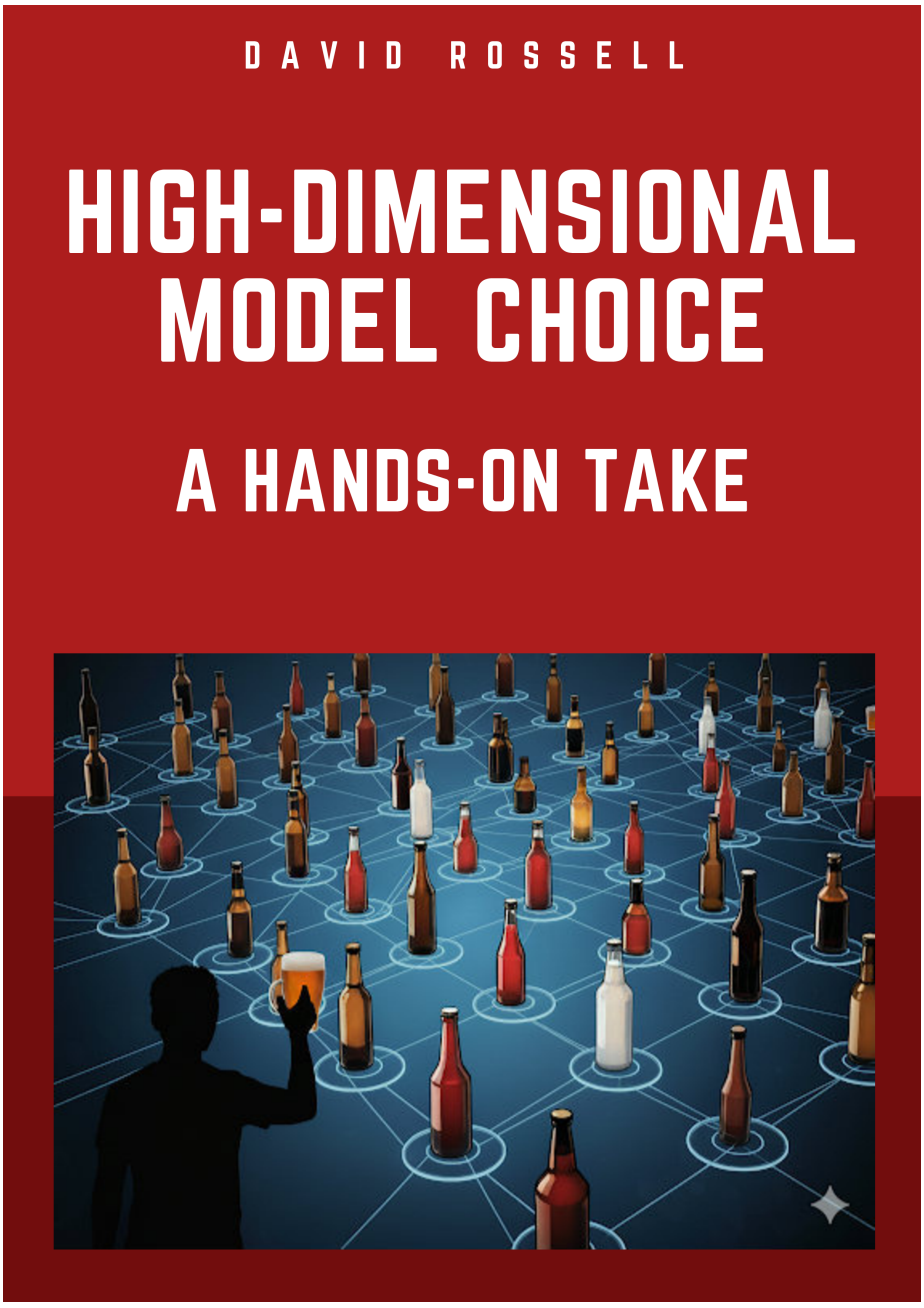

High-dimensional model choice. A hands-on take

David Rossell

2025-09-08

Contents

Preface

This book shows how to use the modelSelection package for sparse inference, mainly Bayesian model selection (BMS) and averaging (BMA), for a number of popular models listed below. It also implements L0 criteria like the AIC, BIC, EBIC, or other general information criteria. The package's C++ implementation is not optimal, but it's designed to be minimally scalable in sparse high-dimensional settings (large p). A lot of work went into coding and maintaining the package, if you use it please cite at least one of the papers indicated below.

For a quick start guide, see Section 1. The main models handled by the package are:

- Generalized linear models: linear, logistic and Poisson regression. BMS, BMA and L0 criteria (Johnson and Rossell, 2012; Rossell and Telesca, 2017; Rossell et al., 2021).
- Linear regression with non-normal residuals (Rossell and Rubio, 2018), including asymmetric Normal, Laplace and asymmetric Laplace residuals.
- Accelerated Failure Time models for right-censored survival data (Rossell and Rubio, 2021).
- Bayesian inference for Gaussian graphical models
- Bayesian for Gaussian mixture models (Fúquene et al., 2019).

On the Bayesian side, `modelSelection` is the main package implementing **non-local priors** (NLPs) but other popular priors are also implemented, e.g. Zellner's and Normal shrinkage priors in regression, or Gaussian spike-and-slab priors in graphical models. NLPs are briefly reviewed in this book, see Johnson and Rossell (2010) and Johnson and Rossell (2012) for their model selection properties, Rossell and Telesca (2017) for parameter estimation, and Rossell et al. (2021) for computational approximations to marginal likelihoods.

Chapter 1

Quick start

The main functions for regression-type models are `modelSelection` and `bestBIC` (along with companions like `bestEBIC`, `bestAIC` and `bestIC`). Details are in subsequent sections. Below we illustrate how to obtain:

- Information criteria for all models (including those from MCMC exploration)
- Posterior model probabilities
- Marginal posterior inclusion probabilities
- BMA point estimates and posterior intervals

See `modelSelectionGGM` for Gaussian graphical models and `bfnormmix` for Gaussian mixture models.

1.1 Linear regression

We simulate linear regression data

$$y_i = \sum_{j=1}^{p} x_{ij}\theta_j + \epsilon_i,$$

for $p = 3$ covariates and $i = 1, \ldots, 100$ individuals. We set regression coefficients $\theta_1 = 1$, $\theta_2 = 1$, $\theta_3 = 0$ and $\epsilon_i \sim N(0, 1)$, and the random number seed for reproducibility. It is good practice to store the outcome and covariates into a data frame, as we do below.

```
library(modelSelection)
set.seed(1234)
x <- matrix(rnorm(100*3), nrow=100, ncol=3)
theta <- matrix(c(1,1,0), ncol=1)
```

```
y <- x %*% theta + rnorm(100)
df <- data.frame(y, x)
```

1.1.1 L0 criteria

`bestBIC` obtains the BIC for all models. As usual in R we can use formulas like `y ~ X1 + X2 + X3` (provided the variables are stored in a data frame), or simply `y ~ .` to consider all the variables in the data (other than `y`) as covariates. An intercept is added automatically, giving a total of 4 variables and $2^4 = 16$ possible models (the intercept can be removed by adding -1 to the formula, as usual). `bestBIC` enumerates these 16 models and finds the model with best (lowest) BIC (when there are many covariates, MCMC is used to explore the model space). For our simulated data the selected model matches the data-generating truth, which only features the first two covariates.

```
fit.bic <- bestBIC(y ~ ., data=df)
```

```
## Enumerating models...
##  Computing posterior probabilities
## 0%6%12%18%25%31%37%43%50%56%62%68%75%81%87%93% Done
```

```
print(fit.bic)
```

```
## icfit object
##
## Model with best BIC : X1 X2
##
## Use summary(), coef() and predict() to get inference for the top model
## Use coef(object$msfit) and predict(object$msfit) to get BMA estimates and predictio
```

We list the BIC for the top 5 models (index 1 corresponds to the intercept, 2 to `x[,1]` and so on). We can also use standard functions like `summary` and `coef` to view the MLE for the best model, and `predict` to obtain predictions for new data.

```
fit.bic$models[1:5,]
```

```
## # A tibble: 5 x 2
##   modelid   bic
##   <chr>   <dbl>
## 1 2,3      302.
## 2 2,3,4    307.
## 3 1,2,3    307.
## 4 1,2,3,4  311.
## 5 3        381.
```

```
summary(fit.bic)
```

```
##
## Call:
## glm(formula = f, family = family2glm(ms$family), data = data)
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## X1   1.1505     0.1022   11.26   <2e-16 ***
## X2   1.1509     0.1006   11.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.06776)
##
##     Null deviance: 371.43  on 100  degrees of freedom
## Residual deviance: 104.64  on  98  degrees of freedom
## AIC: 294.32
##
## Number of Fisher Scoring iterations: 2
```

```
coef(fit.bic)
```

```
##       X1       X2
## 1.150549 1.150920
```

1.1.2 Bayesian model selection

A limitation of L0 criteria is that they ignore the uncertainty in the selected model, i.e. we're not completely sure that it's the correct one. We use BMS to assess that uncertainty, in a Bayesian sense. BMS requires setting a prior distribution on the models and on the parameters for each model. For now, we run `modelSelection` with default priors (Beta-Binomial prior on the models, pMOM prior with default prior precision on the coefficients).

```
priorCoef <- momprior()
priorDelta <- modelbbprior()
fit.bms <- modelSelection(y ~ ., data=df,
                     priorCoef=priorCoef,
                     priorDelta=priorDelta)
```

```
## Enumerating models...
##  Computing posterior probabilities
## 0%6%12%18%25%31%37%43%50%56%62%68%75%81%87%93% Done
```

`postProb` shows posterior model probabilities (sorted decreasingly), and `coef` gives BMA point estimates, 0.95 posterior intervals, and marginal posterior inclusion probabilities $P(\theta_j \neq 0 \mid y)$. Below, `phi` refers to the error variance σ^2 (which is included with probability 1). In our example, BMS selects the right covariates and assigns high posterior probability to that solution, as one would

ideally wish.

```
coef(fit.bms)
```

```
##                 estimate        2.5%      97.5%      margpp
## (Intercept) 0.007082034 -0.02658464 0.04089499 0.007366249
## X1          1.133309621  0.93331088 1.33480178 1.000000000
## X2          1.134404673  0.93919629 1.33501531 1.000000000
## X3          0.000366013  0.00000000 0.00000000 0.008254065
## phi         1.103715115  0.84213596 1.44604848 1.000000000
```

```
postProb(fit.bms)[1:5,]
```

```
##    modelid family           pp
## 7      2,3 normal 9.845428e-01
## 8    2,3,4 normal 8.090989e-03
## 15   1,2,3 normal 7.203173e-03
## 16 1,2,3,4 normal 1.630761e-04
## 3        3 normal 3.424188e-17
```

Finally, we can use `predict` to obtain point predictions and 0.95 posterior predictive intervals.

```
ypred <- predict(fit.bms)
head(ypred)
```

```
##         mean       2.5%       97.5%
## 1 -0.8928148 -1.1160111 -0.66885457
## 2 -0.2161415 -0.3514485 -0.08236455
## 3  1.3134407  1.0653356  1.56205993
## 4 -3.2261301 -3.6793885 -2.77249364
## 5 -0.4427614 -0.6498843 -0.23853199
## 6  0.7716784  0.6332783  0.90914325
```

1.2 Logistic regression

For binary outcomes, we simply specify `family='binomial'` (and for Poisson we specify `family='poisson'`). We first create a binary version of our simulated outcome.

```
dfbin <- transform(df, ybin = (y > 0)) |>
  dplyr::select(!y) #drop variable y
```

We next use `bestBIC` and `modelSelection` as before. The selected model is still the correct one, but the posterior probability for (wrongly) including `x[,3]` is higher than in the linear regression data. This is intuitively expected, binary outcomes carry less information than Gaussian ones, so there is more uncertainty on the chosen model.

```
fit2.bic <- bestBIC(ybin ~ ., family='binomial', data=dfbin)
```

```
## Enumerating models...
##  Computing posterior probabilities
## 0%6%12%18%25%31%37%43%50%56%62%68%75%81%87%93% Done
```

```
print(fit2.bic)
```

```
## icfit object
##
## Model with best BIC : X1 X2
##
## Use summary(), coef() and predict() to get inference for the top model
## Use coef(object$msfit) and predict(object$msfit) to get BMA estimates and predictions
```

```
fit2.bms <- modelSelection(ybin ~ ., data=dfbin,
                     priorCoef=priorCoef,
                     priorDelta=priorDelta,
                     family='binomial')
```

```
## Enumerating models...
##  Computing posterior probabilities
## 0%6%12%18%25%31%37%43%50%56%62%68%75%81%87%93% Done
```

```
coef(fit2.bms)
```

```
## Warning in hasPostSampling(object): Exact posterior sampling not implemented,
## using Normal approx instead
```

```
##                estimate       2.5%     97.5%       margpp
## (Intercept) 0.16323081 0.03275118 0.2669387 2.730669e-10
## X1          1.38893900 0.78354882 2.0102542 1.000000e+00
## X2          1.05744788 0.53310589 1.6067301 9.999136e-01
## X3          0.07043283 0.00000000 0.7391405 1.695233e-01
```

1.3 Non-Linear effects via Generalized Additive Models (GAMs)

Non-linear effects can be modeled via cubic splines using the `smooth` argument (the default is 9 knots, producing 5 columns in design matrix for each non-linear covariate). When using the smooth argument we cannot use the `~ .` notation for including all covarites, rather we must list those for which we wish to include a non-linear effect (see the example below). The effect of each covariate is decomposed as a linear part plus a deviation from linearity (which is forced to be orthogonal to the linear term). This is useful to identify covariates for which a linear effect is sufficient, and covariates for which there are non-linearities. `modelSelection` considers 3 possibilities for each covariate: excluding it entirely,

including only the linear effect, and including both linear and non-linear terms. For further details on this decomposition, see (Rossell and Rubio, 2021).

The linear effect coefficients are displayed using the original variable names, and the non-linear coefficients with an `.s` appended. Here we have 5 columns coding for the non-linear effect, labelled as `.s1` though `.s5`. In our example, there is strong evidence for (correctly) including the linear effect of `X1` and `X2`, excluding their non-linear effects, and exclusing `X3` entirely.

```
fit.gam <- modelSelection(y ~ ., data=df,
                      smooth = ~ X1 + X2 + X3,
                      priorCoef=priorCoef,
                      priorDelta=priorDelta, verbose=FALSE)
```

```
coef(fit.gam)
```

```
## Warning in hasPostSampling(object): Exact posterior sampling not implemented,
## using Normal approx instead
```

```
##                   estimate        2.5%      97.5%       margpp
## (Intercept)  8.127021e-03 -0.01028342 0.02684467 7.499102e-03
## X1           1.140319e+00  1.02940753 1.25420735 1.000000e+00
## X2           1.139502e+00  1.03040279 1.24949508 1.000000e+00
## X3           1.579339e-04  0.00000000 0.00000000 8.545216e-03
## X1.s1       -1.088515e-04  0.00000000 0.00000000 4.026757e-04
## X1.s2        1.357836e-04  0.00000000 0.00000000 4.026757e-04
## X1.s3       -2.810730e-04  0.00000000 0.00000000 4.026757e-04
## X1.s4        2.847856e-04  0.00000000 0.00000000 4.026757e-04
## X1.s5        2.822071e-05  0.00000000 0.00000000 4.026757e-04
## X2.s1        4.295849e-03  0.00000000 0.00000000 4.296470e-03
## X2.s2        5.891295e-03  0.00000000 0.00000000 4.296470e-03
## X2.s3        2.200417e-03  0.00000000 0.00000000 4.296470e-03
## X2.s4        3.947671e-03  0.00000000 0.00000000 4.296470e-03
## X2.s5        1.376460e-03  0.00000000 0.00000000 4.296470e-03
## X3.s1       -1.142249e-04  0.00000000 0.00000000 1.855263e-05
## X3.s2        8.956766e-05  0.00000000 0.00000000 1.855263e-05
## X3.s3        7.400172e-05  0.00000000 0.00000000 1.855263e-05
## X3.s4        5.668310e-05  0.00000000 0.00000000 1.855263e-05
## X3.s5        1.327733e-04  0.00000000 0.00000000 1.855263e-05
## phi          1.093750e+00  0.82918219 1.43831078 1.000000e+00
```

Chapter 2

Background on Bayesian model selection and averaging

We use the term Bayesian model selection (BMS) for the prototypical setting where one considers multiple hypotheses or models and wishes to obtain posterior model probabilities for each model. For example, in regression each model may be associated to what covariates are included in the regression equation (i.e. have non-zero coefficients), in graphical models each model may be associated to what edges are present, and in mixtures each model may correspond to a number of mixture components (clusters).

Section 2.1 illustrates the basic BMS notions by testing whether a Gaussian mean is zero. Section 2.2 discusses the general BMS framework. Sections 2.1-2.2 are intended for readers who are unfamiliar with BMS. Sections 2.3 and 2.4 discuss slightly more nuanced details on how to set prior distributions, some reasonable defaults and how to induce sparsity in high dimensional problems using either sparse model priors or non-local parameter priors. Finally, Section 2.5 outlines some key ideas behind computational aspects of BMS.

2.1 A simplest example

Example 2.1. Let $y_i \sim N(\mu, \sigma^2)$ independently for $i = 1, \dots, n$ and consider the two models (or hypotheses)

$$\mu = 0$$
$$\mu \neq 0.$$

As explained in Section 2.2, in this book we denote models by γ. In this case we use $\gamma = 0$ to denote the null model $\mu = 0$ and $\gamma = 1$ to denote the alternative model $\mu \neq 0$.
The goal is to obtain the posterior probability of the alternative $P(\mu \neq 0 \mid \mathbf{y})$, or equivalently $P(\gamma = 1 \mid \mathbf{y})$, where $\mathbf{y} = (y_1, \dots, y_n)$ are the observed data. We may also want to obtain a BMA estimate $E(\mu \mid \mathbf{y}) =$

$$E(\mu \mid \gamma = 0, \mathbf{y})P(\gamma = 0 \mid \mathbf{y}) + E(\mu \mid \gamma = 1, \mathbf{y})P(\gamma = 1 \mid \mathbf{y}) = E(\mu \mid \gamma = 1, \mathbf{y})P(\gamma = 1 \mid \mathbf{y}),$$

since $E(\mu \mid \gamma = 0, \mathbf{y}) = 0$, and a 0.95 posterior interval for μ, that is an interval $[u_1, u_2]$ such that $P(\mu \in [u_1, u_2] \mid \mathbf{y}) = 0.95$.
To perform a Bayesian analysis, one must specify a prior distribution on everything that is unknown. In our context:

1. We don't know which model is the correct one. We hence need to specify prior model probabilities, e.g. $P(\gamma = 0)$ and $P(\gamma = 1)$ in this example.
2. Even if we knew the model, we don't know the value of its parameters. We hence need to specify a prior on the parameters of each model. In this example, we must set prior densities $p(\sigma^2 \mid \gamma = 0)$ and $p(\mu, \sigma^2 \mid \gamma = 1)$.

In simple settings with two models it is customary to set a uniform model prior. That is, equal prior model probabilities $P(\gamma = 0) = P(\gamma = 1) = 1/2$, unless one has strong reasons for doing otherwise (e.g. data from a related past experiment). In Section 2.3 we discuss how to set the model prior in more advanced settings.
For the prior on parameters, both models feature the error variance. A popular choice is setting an inverse gamma prior $\sigma^2 \sim IG(a/2, l/2)$ under both models, for some given $a, l > 0$. Typically posterior probabilities are robust to the choice of (a, l), provided they're small values (by default, `modelSelection` sets $a = l = 0.01$). To complete the parameter prior under the alternative model we must set a prior on μ given σ^2. A popular choice is $p(\mu \mid \sigma^2, \gamma = 1) = N(\mu; 0, g\sigma^2)$, for some given $g > 0$. Posterior probabilities are sensitive to g, but a common default (unit information prior) is $g = 1$ and results are typically fairly robust as long as g is not too different from this default. In Section 2.4.3 we extend the example to consider a wide range of g values. See also Section 2.4 for some discussion on parameter priors in more advanced examples.

Let us work out Example 2.1 in R. We simulate a dataset where $\mu = 0$, set a uniform model prior and a Gaussian prior on μ with `normalidprior` setting the default $g = 1$ (corresponding to argument `taustd`). Importantly, we set the argument `center=FALSE` because otherwise `modelSelection` centers `y` by subtracting its sample mean and, while this is conventional in regression where there is little interest in the intercept, in this example it would be inappropriate (the centered `y` has mean 0 by definition!). We obtain a high posterior probability $P(\gamma = 0 \mid \mathbf{y})$, hence there isn't Bayesian evidence that $\mu \neq 0$. We remark that for a sample size of $n = 100$ and a single parameter being tested (μ), one might expect to get more conclusive evidence in favor of the null. This issue can be addressed by setting a non-local prior on μ, please see Section 2.4. We use `coef` to obtain the BMA estimates and 0.95 intervals for μ (row `Intercept` in the output below) and σ^2 (row `phi`). For comparison, a t-test also doesn't lead to rejecting the null model.

```
set.seed(1234)
n <- 100
y <- rnorm(n)
df <- data.frame(y=y)
```

```
priorDelta <- modelunifprior()
priorCoef <- normalidprior(taustd=1)
ms <- modelSelection(y ~ 1, data=df, priorCoef=priorCoef, priorDelta=priorDelta, cente
postProb(ms)
```

```
##   modelid family        pp
## 1         normal 0.7510781
## 2       1 normal 0.2489219
```

```
coef(ms)
```

```
##                estimate       2.5%    97.5%    margpp
## (Intercept) -0.03949096 -0.2848408 0.000000 0.2489219
## phi          1.03658377  0.7882084 1.365121 1.0000000
```

```
t.test(y)
```

```
##
##  One Sample t-test
##
## data:  y
## t = -1.5607, df = 99, p-value = 0.1218
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.35605755  0.04253406
## sample estimates:
##  mean of x
## -0.1567617
```

We next plot how the posterior probability and the t-test P-value change when the sample mean ranges from $0, 0.05, 0.1, \ldots, 0.5$. Figure 2.1 shows the results. As the sample mean of $\mathbf{y}$ grows, we obtain overwhelming evidence for $\mu \neq 0$. Note that the Bayesian framework is more conservative, in that one obtains a P-value < 0.05 for smaller before one obtains $p(M_2 \mid \mathbf{y}) > 0.95$. This conservative character of Bayes factors is well-known, and it is one of the reasons why BMS induces sparsity in high-dimensional problems. One may obtain even more conservative results by setting certain model and parameter priors, as discussed in the next sections.

```
y0 <- y - mean(y)
mu <- seq(0, .5, by=.05)
pp.yplus <- pval.yplus <- double(length(mu))
for (i in 1:length(mu)) {
  dfplus <- transform(df, yplus= y0 + mu[i])
  ms <- modelSelection(yplus ~ 1, data=dfplus, priorCoef=priorCoef, priorDelta=priorDelta, center
  pp.yplus[i] <- coef(ms)['(Intercept)', 'margpp']
  pval.yplus[i] <- t.test(dfplus$yplus)$p.value
}
```

```
plot(mu, pp.yplus, ylab='Posterior probability', xlab='Sample mean', ylim=c(0,1), type='l')
lines(mu, pval.yplus, col='blue')
abline(h= c(0.05, 0.95), lty=2, col=c('blue','black'))
legend('topleft', c("Posterior probability", "P-value"), lty=1, col=c("black","blue"))
```

2.2 General framework

Consider a fully general setting where one considers a set of models Γ. We denote individual models by $\gamma \in \Gamma$, and the parameters of that model by $_\gamma$. This is without loss of generality, if one has K arbitrary models then one could simply set $\gamma \in \{1, \ldots, K\}$, and the parameters could be an infinite-dimensional object such as a density function.

In regression settings it is convenient to relate γ to the non-zero parameters, as done in Example 2.1, where we had $\gamma = \mathrm{I}(\mu \neq 0)$ and I() is the indicator function.

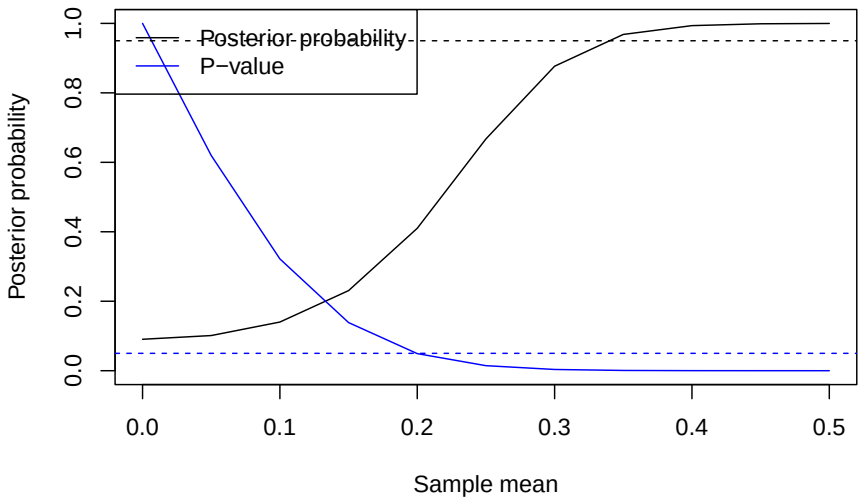

Figure 2.1: Normal mean example (n=100). Posterior probability $P(\mu \neq 0 \mid \mathbf{y})$ and t-test P-value as a function of the sample mean. The dotted lines indicate standard 0.95 and 0.05 thresholds for posterior probabilities and P-values respectively.

Example 2.2. Consider a linear regression

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

where $\epsilon_i \sim N(0, \sigma^2)$ independently across $i = 1, \ldots, n$. Suppose that we wish to consider the 2^p models arising from excluding/including each of the p covariates. To this end, let $\gamma_j = \mathrm{I}(\beta_j \neq 0)$ be an inclusion indicator for variable $j = 1, \ldots, p$. Then we can denote an arbitrary model by $= (\gamma_1, \ldots, \gamma_p)$, the model space is $\Gamma = \{0, 1\}^p$, and $_\gamma = (\ _\gamma, \sigma^2)$, where $_\gamma = \{\beta_j : \gamma_j = 1\}$ are the non-zero regression parameters under . In such settings we use bold face notation to stress that it is a vector.

Given a prior model probability $p(\gamma)$ and a prior on parameters $p(\ _\gamma \mid \gamma)$ for every γ, Bayes formula gives posterior model probabilities

$$p(\gamma \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid \gamma) p(\gamma)}{\sum_{\gamma'} p(\mathbf{y} \mid \gamma') p(\gamma')} \tag{2.1}$$

where $p(\gamma)$ is the prior probability of model γ and

$$p(\mathbf{y} \mid \gamma) = \int p(\mathbf{y} \mid \ _\gamma, \gamma) p(\ _\gamma \mid \gamma) d\ _\gamma \tag{2.2}$$

is the so-called **integrated (or marginal) likelihood**. In (2.2), $p(\mathbf{y} \mid \ _\gamma, \gamma)$ is the likelihood function for model γ. For simplicity, Equation (2.2) assumes the standard setting where $_\gamma$ follows a continuous distribution, but it can be directly extended to cases where $_\gamma$ is discrete (then the integral becomes a sum) or a mixture of discrete and continuous distribution (then it's an integral with respect to a suitable dominating measure).

Intuitively,(2.2) says that if model γ has a large prior probability $p(\gamma)$ and a large average value of its likelihood function (with respect to the specified prior), then it has high posterior probability $p(\gamma \mid \mathbf{y})$.

A related quantity are the so-called **Bayes factors** between models any pair of models γ and γ',

$$B_{\gamma\gamma'} = \frac{p(\mathbf{y} \mid \gamma)}{p(\mathbf{y} \mid \gamma')}. \tag{2.3}$$

Posterior model probabilities in (2.1) are one-to-one functions of Bayes factors and prior model probability ratios, namely

$$p(\gamma \mid \mathbf{y}) = \left(1 + \sum_{\gamma' \neq \gamma} B_{\gamma'\gamma} \frac{p(\gamma')}{p(\gamma)}\right)^{-1}$$

$$\frac{p(\gamma \mid \mathbf{y})}{p(\gamma' \mid \mathbf{y})} = B_{\gamma\gamma'} \frac{p(\gamma)}{p(\gamma')}.$$

2.3 Prior on models

In simple problems like Example 2.1 where one considers only a few models, it is customary to assign equal prior probabilities

$$p(\ ) = 1/|\Gamma|. \tag{2.4}$$

We refer to (2.4) as a **uniform model prior**. This prior is not recommended for problems with a moderate to large number of parameters. To see why, consider Example 2.2. If one sets (2.4), then it is easy to see that the implied prior distribution on the model size $\sum_{j=1}^{p} \gamma_j \sim \text{Bin}(p, 1/2)$. This prior concentrates heavily on mid-size models including roughly $p/2$ covariates, and in particular it assigns very low prior probability to models including a few covariates. That is, the prior does not induce sparsity.

We discuss three other model priors that are popular in the regression context where $\ = (\gamma_1, ..., \gamma_p)$. We denote by $|\ |_0 = \sum_{j=1}^{p} \gamma_j$ the model size, i.e. the number of non-zero regression parameters in $\ $.

1. Binomial prior, possibly combined with empirical Bayes (Rognon-Vael and Rossell, 2025).
2. Beta-Binomial prior (Scott and Berger, 2010).
3. Complexity prior (Castillo et al., 2015).

We found the Beta-Binomial prior to be a very good default in practice, attaining a good balance between false positive control and preserving power to detect non-zero coefficients. This is the default prior in `modelSelection` and, unless you have good reasons for doing otherwise, we suggest that you use this. For readers who are more familiar with the frequentist literature, the Beta-Binomial prior inspired the popular Extended BIC (EBIC) criterion (Chen and Chen, 2008). Roughly speaking, the model with highest posterior probability under the Beta-Binomial prior is asymptotically equivalent to the model with best EBIC.

We next discuss these prior in some detail. First-time readers may wish to skip these sections.

2.3.1 Binomial prior

Let $\gamma_j \sim \text{Bern}(n, \pi_j)$ independently for $j = 1, ..., p$. By default `modelSelection` sets $\pi_1 = ... = \pi_p = \pi$, and then the prior on the model size is

$$p(\ ) = \prod_{j=1}^{p} \pi^{|\ |_0} (1-\pi)^{p-|\ |_0} \tag{2.5}$$

$$|\ |_0 \sim \text{Bin}(n, \pi). \tag{2.6}$$

Setting small π encourages sparse solutions, but the question is what value of π should be chosen. A common default is to set $\pi = c/p$ for some constant $c > 0$, so that prior expected model size $E(|\ |_0) = c$ regardless of c. Unless one has a rough idea on how many variables may have an effect, it's unclear what c should be chosen.

A possible strategy, implemented in function `modelSelection_eBayes`, is to set π using empirical Bayes. Briefly, one sets

$$\widehat{\pi} = \arg\max_{\pi} p(\mathbf{y} \mid \pi) p(\pi)$$

where $p(\pi)$ is a minimally-informative prior on π (basically, preventing extreme values like $\pi = 0$ and $\pi = 1$), and $p(\mathbf{y} \mid \pi) = \sum_{\gamma} p(y \mid\ )p(\ \mid \pi)$ the marginal likelihood. The idea is to learn how much sparsity is appropriate to impose to the data at hand, as an alternative to discriminately assuming strongly sparse priors. For example, see Giannone et al. (2021) for a discussion that sparse priors may often be inappropriate in the Social Sciences. We refer the reader to Section 6 for a more detailed discussion on empirical Bayes.

2.3.2 Beta-Binomial prior

Scott and Berger (2010) argued for setting a uniform prior $\pi \sim U(0,1)$ and $\gamma_j \sim \text{Bern}(\pi)$ independently across $j = 1, \dots, p$. These define $p(\pi)$ and $p(\ \mid \pi)$, which imply the following marginal prior

$$p(\ ) \propto \frac{1}{\binom{p}{|\ |_0}} \tag{2.7}$$

It is a well-known result that then the model size follows a Beta-Binomial distribution, that is $|\ |_0 \sim \text{Beta-Binomial}(p, 1, 1)$ (this holds basically by definition of the Beta-Binomial distribution). We hence refer to (2.7) as **Beta-Binomial** prior. In fact, the Beta-Binomial$(p, 1, 1)$ is simply a discrete uniform distribution in $0, 1, \dots, p$.

A perhaps simpler way to think about the Beta-Binomial prior is that one sets a uniform prior on the model size $|\ |_0$ (in stark contrast with the Binomial imposed by (2.6)), and that all models of a given size have the same probability.

2.3.3 Complexity prior

Castillo et al. (2015) showed that one may obtain optimal minimax parameter estimation rates in linear regression by setting a very sparse model prior, which they referred to as **Complexity prior**. As a side remark, for their results to hold, one should also set a prior on parameters than has Laplace tails or thicker, which in particular rules out using Gaussian priors. For model selection purposes (as opposed to estimation), using Gaussian priors on parameters leads to good rates (Rossell, 2022), and they're much more convenient computationally,

Table 2.1: Model prior probabilities in a regression example with p=2 covariates

gamma1	gamma2	Uniform	Beta-Binomial	Complexity(1)
0	0	0.25	0.3333333	0.6652410
1	0	0.25	0.1666667	0.1223642
0	1	0.25	0.1666667	0.1223642
1	1	0.25	0.3333333	0.0900306

in particular in regression where they give closed-form marginal likelihoods in (2.2)). Hence `modelSelection` focuses on using Gaussian priors on parameters.

The main idea of a Complexity prior is that the implied prior on the model size $P(|\ |_0 = l)$ decreases essentially exponentially with l. Specifically, here we define $\ \sim \text{Complexity}(c)$ for some given $c > 0$ (and by default, we take $c = 1$), whenever

$$p(\ ) \propto \frac{1}{p^{cl}\binom{p}{|\ |_0}} \Longrightarrow P(|\ |_0 = l) \propto \frac{1}{p^{cl}}. \tag{2.8}$$

2.3.4 A simple example

Consider a regression example with $p = 2$ covariates, leading to the four models shown in Table 2.1. The uniform model prior assigns 1/4 probability to each, implying a prior probabilities 1/4, 1/2 and 1/4 to model sizes 0, 1 and 2 respectively. The Beta-Binomial prior assigns 1/3 to each model size. Since there are 2 models of size $|\ |_0 = 1$, each receives probability $(1/3)(1/2) = 1/6$. The Complexity prior results in a much sparser model prior, which works great when the data-generating truth is truly sparse or the sample size n is large enough, otherwise it may suffer from lower power of detecting truly non-zero parameters.

Let us illustrate these issues in a simple simulation.

```
p <- 4; n <- 50
x <- matrix(rnorm(n * p), nrow=n)
beta <- matrix(c(rep(0, 2), rep(0.5, p-2)), ncol=1)
y <- x %*% beta + rnorm(n)
df <- data.frame(y, x)
ms.unif <- modelSelection(y ~ -1 + ., data=df, priorDelta = modelunifprior(), verbose=
ms.bbin <- modelSelection(y ~ -1 + ., data=df, priorDelta = modelbbprior(), verbose=FA
ms.comp <- modelSelection(y ~ -1 + ., data=df, priorDelta = modelcomplexprior(), verbo
```

```
coef(ms.unif)
```

```
##              estimate        2.5%      97.5%     margpp
## intercept 0.010088004 -0.07671732 0.06700013 1.00000000
## X1        0.052849466  0.00000000 0.43701525 0.17309500
```

```
## X2        0.003335324  0.00000000 0.00000000 0.01986338
## X3        0.458777850  0.00000000 0.75584664 0.91440091
## X4        0.717333220  0.44869371 0.99985446 0.99952478
## phi       0.890176124  0.59739527 1.33357635 1.00000000
```

```
coef(ms.bbin)
```

```
##               estimate        2.5%      97.5%     margpp
## intercept 0.007733875 -0.07805773 0.06894077 1.00000000
## X1        0.076415774  0.00000000 0.46090766 0.23952664
## X2        0.007316171  0.00000000 0.16517675 0.04047445
## X3        0.451085060  0.00000000 0.76375200 0.88959840
## X4        0.725632631  0.45322866 1.00879847 0.99904588
## phi       0.892743238  0.60243438 1.34532111 1.00000000
```

```
coef(ms.comp)
```

```
##                estimate        2.5%      97.5%      margpp
## intercept -0.0175989596 -0.08065022 0.05930749 1.000000000
## X1         0.0159372331  0.00000000 0.30412930 0.050765743
## X2         0.0008506681  0.00000000 0.00000000 0.006235675
## X3         0.3067855113  0.00000000 0.71561984 0.612913816
## X4         0.7173532130  0.43773061 1.01398309 0.991811587
## phi        0.9598673704  0.62117928 1.48441620 1.000000000
```

2.4 Prior on coefficients

2.4.1 Local priors

2.4.2 Non-local priors

2.4.3 Sensitivity to prior variance

We return to Example 2.1, and assess the robustness of the results for the original data as one varies the value of the prior dispersion g. This is interesting because much literature has been devoted to the so-called Jeffreys-Lindley-Bartlett paradox (Lindley, 1957). Briefly, as $g \to \infty$ the posterior probability $P(\mu = 0 \mid \mathbf{y})$ converges to 1, which a number of authors viewed as problematic. We contend that this is not so: if one views g as a tuning parameter, then $g = \infty$ is an extreme value and it's therefore unsurprising that one obtains extreme results. For example, an infinite LASSO penalty also leads to $\hat{\mu} = 0$, yet this doesn't stop anyone from using LASSO. The question is whether one can set tuning parameters to values that lead to good behavior, and there's abundant theoretical and empirical evidence that $g = 1$ does so.

Here we consider the range $g \in [0.1, 10]$, i.e. some of these prior variances are very different from the default $g = 1$. We do the exercise with the dataset `y` simulated in Example 2.1, where truly $\mu = 0$, and also with another dataset `y1`

where truly $\mu = 1$.

```
df1 <- data.frame(y1= rnorm(n, mean=0.5))
gseq <- seq(0.001, 10^5, length=200)
pp0.gseq <- pp1.gseq <- double(length(gseq))
for (i in 1:length(gseq)) {
  priorCoef <- normalidprior(taustd=gseq[i])
  ms <- modelSelection(y ~ 1, data=df, priorCoef=priorCoef, priorDelta=priorDelta, cen
  pp0.gseq[i] <- coef(ms)['(Intercept)', 'margpp']
  ms1 <- modelSelection(y1 ~ 1, data=df1, priorCoef=priorCoef, priorDelta=priorDelta,
  pp1.gseq[i] <- coef(ms1)['(Intercept)', 'margpp']
}
```

```
plot(gseq, pp0.gseq, xlab='g', ylab='Posterior probability', ylim=c(0,1), type='l')
plot(gseq, pp1.gseq, xlab='g', ylab='Posterior probability', ylim=c(0,1), type='l')
```

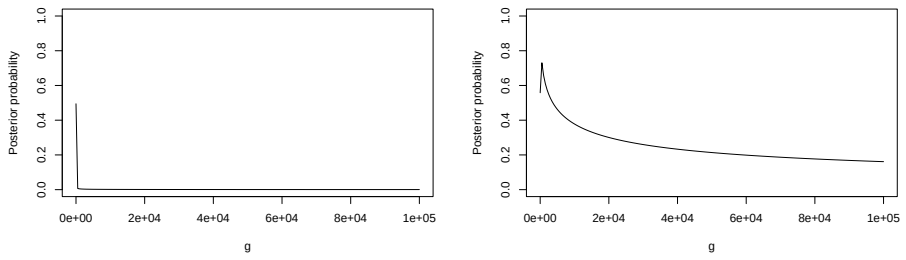

Figure 2.2: Posterior probability $P(\mu \neq 0 \mid \mathbf{y})$ vs. prior dispersion g in the Gaussian mean example (n=100). Left: truly $\mu = 0$. Right: truly $\mu = 0.5$.

Figure 2.2 shows the results. Although $P(\mu \neq 0 \mid \mathbf{y})$ decreases as g grows, when truly $\mu = 0$ the changes are rather small and when truly $\mu = 0.5$ the changes cannot be appreciated. Overall, the conclusions are unaffected by g. Note that as $g \to 0$ we have $P(\mu \neq 0 \mid \mathbf{y})$ approaching 0.5, this is reasonable because for $g = 0$ the $N(0, g\sigma^2)$ prior under the alternative ($\gamma = 1$) states that $\mu = 0$, i.e. the null and alternative hypotheses are equivalent and both receive the same posterior probability. This effect can only be appreciated when truly $\mu = 0$, when truly $\mu = 0.5$ we would need to consider much smaller g. Just for fun, below we consider an absurdly small $g = 0.001$ and an absurdly large $g = 10^6$, and even then we obtain some evidence for $P(\mu \neq 0 \mid \mathbf{y})$.

```
ms1 <- modelSelection(y1 ~ 1, data=df1, priorCoef=normalidprior(taustd=0.001), priorDel
coef(ms1)
```

```
##               estimate        2.5%      97.5%    margpp
## (Intercept) 0.01192704 -0.03740929 0.07927645 0.5580635
## phi         0.98260934  0.65829790 1.46853024 1.0000000
```

```
ms1 <- modelSelection(y1 ~ 1, data=df1, priorCoef=normalidprior(taustd=10^6), priorDelta=priorDel
coef(ms1)
```

```
##                estimate      2.5%     97.5%     margpp
## (Intercept) 0.02612707 0.0000000 0.4720005 0.05728419
## phi         0.97362830 0.6372641 1.4619225 1.00000000
```

2.5 Computation

2.5.1 Approximating marginal likelihoods

The marginal likelihood in (2.2) has a closed-form expression in some instances, mainly regression with Gaussian errors (e.g., linear regression, non-linear additive regression) with conjugate parameter priors. Recall that the marginal likelihood for model is

$$p(\mathbf{y} \mid \gamma) = \int p(\mathbf{y} \mid \ _{\gamma}, \gamma) p(\ _{\gamma} \mid \gamma) d\ _{\gamma}.$$

Outside these special cases, a numerical approximation is required. The `modelSelection` function implements some such approximations, and their use can be specified with the argument `method`. If `method` is not specified, `modelSelection` selects a sensible default.

A popular strategy in the context of model selection is to use Laplace approximations: they are fairly computationally efficient, and also highly accurate as n grows (Kass et al., 1990). To use Laplace approximations, set `method=='Laplace'`.

Laplace approximations require finding the posterior mode (or alternatively, the MLE) $\hat{\ }_{\gamma}$ and the hessian of the log-posterior density at $\hat{\ }_{\gamma}$. Both these quantities can be found quickly for models that feature a few parameters, but the calculations get cumbersome when:

- One considers many models, i.e. one must repeat the optimization exercise many times
- Some models that feature many parameters have high posterior probability, and hence they're visited often by an MCMC model search algorithm
- The sample size n is large, so evaluating gradients (or hessians) to obtain $\hat{\ }_{\gamma}$ gets costly.

An alternative is to use approximate Laplace approximations (ALA) (Rossell et al., 2021), which are available by using `method=='ALA'`. Briefly, ALA approximates $\hat{\ }_{\gamma}$ by taking a single Newton-Raphson step from an initial estimate $\hat{\ }^{(0)}_{\gamma}$. By default $\hat{\ }^{(0)}_{\gamma} = 0$ is taken, see Rossell et al. (2021) for a study of the theoretical properties of this choice. Alternatively, in `modelSelection` one may provide other $\hat{\ }^{(0)}_{\gamma}$ with the argument `initpar`.

2.5.2 Model search

`modelSelection` uses an MCMC model search, based on classical Gibbs sampling. `modelSelectionGGM` also implements newer birth-death-swap (Yang et al., 2016) and locally informed thresholded (LIT) algorithms (Zhou et al., 2022). The latter are theoretically appealing in that they have been shown to be scalable to high dimensions under relatively stringent sparsity constraints. In practice simple Gibbs sampling works remarkably well, in our experience it usually attains a similar numerical accuracy when it's run for the same clock time as birth-death-swap and LIT.

Chapter 3

Background on L0 criteria

To be added.

3.1 Basics

3.2 MCMC for model search

Chapter 4

Generalized linear models

To be added.

Chapter 5

Generalized additive models

To be added.

Chapter 6

Empirical Bayes for transfer learning

To be added.

Chapter 7

Survival data

To be added.

Chapter 8

Gaussian graphical models

To be added.

Chapter 9

Gaussian mixture models

To be added.

Bibliography

Castillo, I., Schmidt-Hieber, J., and van der Vaart, A. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986–2018.

Chen, J. and Chen, Z. (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771.

Fúquene, J., Steel, M., and Rossell, D. (2019). On choosing mixture components via non-local priors. *Journal of the Royal Statistical Society B*, 81(5):809–837.

Giannone, D., Lenza, M., and Primiceri, G. E. (2021). Economic predictions with big data: The illusion of sparsity. *Econometrica*, 89(5):2409–2437.

Johnson, V. and Rossell, D. (2010). On the use of non-local prior densities for default Bayesian hypothesis tests. *Journal of the Royal Statistical Society B*, 72:143–170.

Johnson, V. and Rossell, D. (2012). Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 24(498):649–660.

Kass, R., Tierney, L., and Kadane, J. (1990). The validity of posterior expansions based on Laplace's method. *Bayesian and likelihood methods in statistics and econometrics*, 7:473–488.

Lindley, D. (1957). A statistical paradox. *Biometrika*, 44:187–192.

Rognon-Vael, P. and Rossell, D. (2025). Empirical Bayes for data integration. *arXiv*, 2508.08336:1–51.

Rossell, D. (2022). Concentration of posterior model probabilities and normalized L0 criteria. *Bayesian Analysis*, 17(2):565–591.

Rossell, D., Abril, O., and Bhattacharya, A. (2021). Approximate Laplace approximations for scalable model selection. *Journal of the Royal Statistical Society B*, 83(4):853–879.

Rossell, D. and Rubio, F. (2018). Tractable bayesian variable selection: beyond normality. *Journal of the American Statistical Association*, 113(524):1742–1758.

Rossell, D. and Rubio, F. (2021). Additive Bayesian variable selection under censoring and misspecification. *Statistical Science*, 38(1):13–29.

Rossell, D. and Telesca, D. (2017). Non-local priors for high-dimensional estimation. *Journal of the American Statistical Association*, 112:254–265.

Scott, J. and Berger, J. (2010). Bayes and empirical Bayes multiplicity adjustment in the variable selection problem. *The Annals of Statistics*, 38(5):2587–2619.

Yang, Y., Wainwright, M., and Jordan, M. (2016). On the computational complexity of high-dimensional Bayesian variable selection. *The Annals of Statistics*, 44(6):2497–2532.

Zhou, Q., Yang, J., Vats, D., Roberts, G. O., and Rosenthal, J. S. (2022). Dimension-free mixing for high-dimensional Bayesian variable selection. *Journal of the Royal Statistical Society B*, 84(5):1751–1784.