

# mombf package vignette

David Rossell

Department of Biostatistics and Bioinformatics

Institute for Research in Biomedicine, Barcelona, Spain

david.rossell@irbbarcelona.org

This manual shows how to use the `mombf` library to compute Moment and inverse Moment Bayes factors (Mom BF and iMom BF, respectively). The intuitive appeal of Mom and iMom BF is that they represent prior beliefs under the alternative hypothesis which are fundamentally different from those under the null hypothesis. Mathematically, when the null hypothesis is true they present better convergence rates than BF resulting from most standard procedures. When the alternative hypothesis is true, they present the same convergence rates as most standard procedures.

The routines compute exact BF for linear regression models, and approximate BF for generalized linear models. Approximate BF can also be obtained in other situations where the regression coefficients are asymptotically normally distributed and sufficient. The library also contains routines to evaluate the prior density and to elicit the prior parameters by specifying the mode *a priori* of the standardized regression coefficients.

In Section 1 we briefly review the definition of the Mom and iMom priors, and we present routines to evaluate them. In Section 2 we analyze Hald's data with linear models and compute Bayes factors to assess whether some predictors can be dropped from the model. Section 3 shows the analysis of some simulated logistic regression data.

## 1 Mom and iMom priors

Let  $\boldsymbol{\theta}' = (\boldsymbol{\theta}'_1, \boldsymbol{\theta}'_2)$  be the vector of regression coefficients,  $\sigma^2$  be a dispersion parameter (*i.e.* the residual variance in a linear regression setup) and suppose that the goal is to test  $H_0 : \boldsymbol{\theta}_1 = \boldsymbol{\theta}_0$  versus  $H_1 = \boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_0$ . Define the quadratic distance  $Q(\boldsymbol{\theta}_1) = (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0)^T V_1^{-1} (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_0) / (ng\sigma^2)$ , where  $\boldsymbol{\theta}_1$  is a  $p_1 \times 1$  dimensional real vector,  $V_1$  is a  $p_1 \times p_1$  positive definite matrix and  $g > 0$  is a scalar. We set  $V_1$  to be proportional to the asymptotic covariance

matrix of the maximum likelihood estimate  $\hat{\boldsymbol{\theta}}_1$ . For instance, in a linear regression setup with design matrix  $X$  we set  $V_1 = (X'X)^{-1}$ .

We define an improper prior density on  $\theta_2$  proportional to 1, and in the situation where  $\sigma^2$  is unknown we specify an independent improper prior on  $\sigma^2$  proportional to  $1/\sigma$ .

## 1.1 Mom prior

Let  $\pi_Z(\boldsymbol{\theta}_1)$  be a prior density for  $\boldsymbol{\theta}_1$  for which  $E_{\pi_Z}[Q(\boldsymbol{\theta}_1)^k]$  is finite. We define the multivariate Mom prior as

$$\pi_M(\boldsymbol{\theta}_1) = \frac{Q(\boldsymbol{\theta}_1)^k}{E_{\pi_Z}[Q(\boldsymbol{\theta}_1)^k]} \pi_Z(\boldsymbol{\theta}_1). \quad (1)$$

The package currently implements normal MOM priors (where  $\pi_Z$  is the g-prior of Zellner and Siow (1980), *i.e.*  $\pi_Z(\boldsymbol{\theta}_1) = N(\boldsymbol{\theta}_0, ng\sigma^2 V_1)$ ) and T MOM priors (where  $\pi_Z$  is a multivariate T with  $\nu \geq 3$  degrees of freedom). Both for normal and T MOM priors only the case  $k = 1$  is currently implemented. For the normal MOM prior the normalization constant is  $E_{\pi_Z}(Q(\boldsymbol{\theta})^k) = \prod_{i=0}^{k-1} (p_1 + 2i)$ , *i.e.* the  $k^{th}$  raw moment of a chi-square distribution with  $p_1$  degrees of freedom. For  $k = 1$  this simplifies to  $E_{\pi_Z}(Q(\boldsymbol{\theta})^k) = 1$ . For the T MOM prior and  $k = 1$  the normalization constant is  $E_{\pi_Z}(Q(\boldsymbol{\theta})^k) = d_{\frac{\nu}{\nu-2}}$ .

## 1.2 iMom prior

The iMom prior on  $\boldsymbol{\theta}_1$  is

$$\pi_I(\boldsymbol{\theta}_1) = c_I Q(\boldsymbol{\theta}_1)^{-\frac{\nu+p_1}{2}} \exp [Q(\boldsymbol{\theta}_1)^{-k}], \quad (2)$$

where

$$c_I = \left| \frac{V_1^{-1}}{ng\sigma^2} \right|^{1/2} \frac{k}{\Gamma(\nu/2k)} \frac{\Gamma(p_1/2)}{\pi^{p_1/2}}. \quad (3)$$

As  $Q(\boldsymbol{\theta}_1)$  increases, the influence of the exponential term in (2) disappears and the tails of  $\pi_I$  are of the same order as those of a multivariate  $T$  with  $\nu$  degrees of freedom. Several authors have found appealing to set  $\nu = 1$  (Bayarri and Garcia-Donato, 2007), which is the default value in our routines. Currently the library only implements the case  $k = 1$ .

## 1.3 Evaluating the Mom and iMom priors

The functions `dmom` and `dimom` evaluate the Mom and iMom priors, respectively. Setting the argument `baseDensity='normal'` in `dmom` (the default)

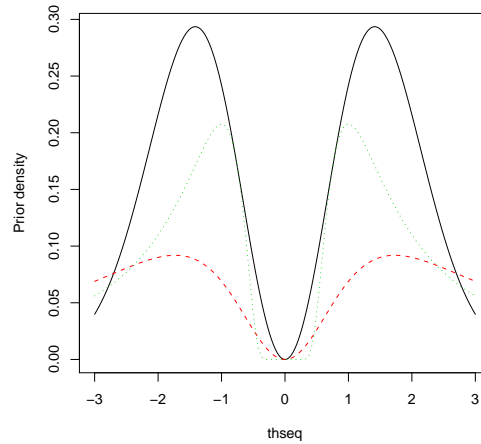


Figure 1: Moment and inverse Moment priors for  $g = 1$

returns the normal MOM density, `baseDensity='t'` returns the t MOM density. The functions `pmom` and `pimom` evaluate the distribution functions, and `qmom` and `qimom` return quantiles. Currently `pmom` and `qmom` are only implemented for the normal MOM. Let's set the prior parameter  $g = 1$  and plot the Mom and iMom priors in a univariate setting for  $\theta_1 \in (-3, 3)$ . By default  $\theta_0$  is set to 0,  $n = 1$  and  $V_1 = 1$ .

```
> library(mombf)
> g <- 1
> thseq <- seq(-3, 3, length = 1000)
> plot(thseq, dmom(thseq, g = g), type = "l", ylab = "Prior density")
> lines(thseq, dmom(thseq, g = g, baseDensity = "t", nu = 3), lty = 2,
+       col = 2)
> lines(thseq, dimom(thseq, g = g), lty = 3, col = 3)
```

The iMOM prior assigns the lowest density for  $\theta_1$  in a neighborhood of 0, whereas the normal MOM prior assigns the largest density. We can also plot the corresponding distribution functions.

```
> library(mombf)
> plot(thseq, pmom(thseq, g = g), type = "l", ylab = "Prior cdf")
> lines(thseq, pimom(thseq, g = g), lty = 3, col = 3)
```

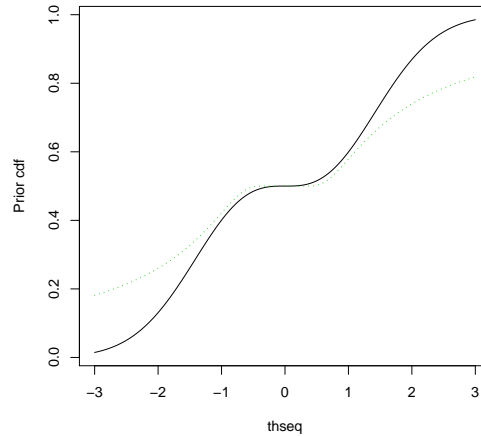


Figure 2: Moment and inverse Moment cdf for  $g = 1$

## 2 Bayes factors for linear regression models

### 2.1 Linear model fit and prior elicitation

The Hald data contains 13 observations, a continuous response variable and 4 predictors. We start by loading the data and fitting a linear regression model.

```
> data(hald)
> dim(hald)

[1] 13  5

> lm1 <- lm(hald[, 1] ~ hald[, 2] + hald[, 3] + hald[, 4] + hald[,
+      5])
> summary(lm1)
```

Call:

```
lm(formula = hald[, 1] ~ hald[, 2] + hald[, 3] + hald[, 4] +
    hald[, 5])
```

Residuals:

	Min	1Q	Median	3Q	Max
	-3.1750	-1.6709	0.2508	1.3783	3.9254

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	62.4054	70.0710	0.891	0.3991
hald[, 2]	1.5511	0.7448	2.083	0.0708 .
hald[, 3]	0.5102	0.7238	0.705	0.5009
hald[, 4]	0.1019	0.7547	0.135	0.8959
hald[, 5]	-0.1441	0.7091	-0.203	0.8441

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.446 on 8 degrees of freedom

Multiple R-squared: 0.9824, Adjusted R-squared: 0.9736

F-statistic: 111.5 on 4 and 8 DF, p-value: 4.756e-07

The goal is to obtain Bayes factors to assess whether any one predictor can be dropped from the model. First, we specify the prior parameter  $g$  based on considerations about the standardized regression coefficient  $(\theta_1/(\sigma n V_1))^2$ .  $\theta_1/\sigma$  is known as the signal-to-noise ratio, or as the standardized effect size. To find the  $g$  value that gives a prior mode at  $\pm 2$ , we use the function `mode2g`. For instance, for the regression coefficient associated to `hald[,2]` we would do as follows.

```
> prior.mode <- 0.2^2
> V <- summary(lm1)$cov.unscaled
> diag(V)

(Intercept)    hald[, 2]    hald[, 3]    hald[, 4]    hald[, 5]
820.65457471    0.09271040    0.08756026    0.09520141    0.08403119

> gmom <- mode2g(prior.mode, prior = "normalMom")
> gtmom <- mode2g(prior.mode, prior = "tMom", nu = 3)
> gimom <- mode2g(prior.mode, prior = "iMom")
> gmom

[1] 0.02

> gtmom

[1] 0.01333333

> gimom
```

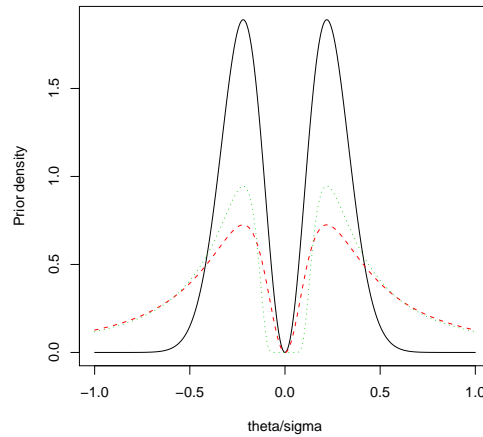


Figure 3: Hald data. Mom and iMom priors for a regression coefficient. The prior mode for  $\theta_1/\sigma$  is set at  $\pm 0.2$

[1] 0.04

We can check the obtained  $g$  values by plotting the prior density.

```
> thseq <- seq(-1, 1, length = 1000)
> plot(thseq, dmom(thseq, V1 = V[2, 2], g = gmom, n = nrow(hald)),
+      type = "l", xlab = "theta/sigma", ylab = "Prior density")
> lines(thseq, dmom(thseq, V1 = V[2, 2], g = gtmom, n = nrow(hald),
+      baseDensity = "t", nu = 3), lty = 2, col = 2)
> lines(thseq, dimom(thseq, V1 = V[2, 2], g = gimom, n = nrow(hald)),
+      lty = 3, col = 3)
```

Another way to specify  $g$  is by finding the value that assigns a desired prior probability to a certain interval. This can be achieved with the function `priorp2g`. For instance, to find the  $g$  value that gives 5% probability to the interval  $(-0.2, 0.2)$  we use the following code.

```
> a <- 0.2
> priorp <- 0.05
> gmom2 <- priorp2g(priorp = priorp, q = a, prior = "normalMom")
> gimom2 <- priorp2g(priorp = priorp, q = -a, prior = "iMom")
> gmom2
```

```
[1] 0.113686

> gimom2

[1] 0.07682918
```

## 2.2 Bayes factor computation

Bayes factors can be easily computed using the functions `mombf` and `imombf`. The normal Mom BF can be computed in explicit form, the T MOM BF require computing a one dimensional integral and the iMom BF a two dimensional integral (regardless of the dimensionality of  $\theta_1$ ). The numerical integration can be achieved either via adaptive quadratures (as implemented in the routines `integrate`) by setting `method='adapt'`, or via Monte Carlo simulation by setting `method='MC'`. When  $\sigma^2$  is unknown, `method=='adapt'` combines `integrate` with the quantile method of Johnson (1992). The parameter `nquant` determines the number of quantiles of the posterior distribution of  $\sigma^2$  at which to evaluate the integral. The default `nquant=100` usually gives a fairly good approximation. For Monte Carlo integration, the argument `B` specifies the number of Monte Carlo samples.

In our example, for computational speed we use `B=100000`, even though in real examples a higher value can be used to ensure proper accuracy. For comparison, we also compute the Bayes factors that would be obtained under Zellner's g-prior with the default value  $g = 1$ , which can be achieved with the function `zellnerbf`. For reproducibility, we set the random number generator seed to the date this document was produced.

```
> set.seed(4 * 2 * 2008)
> mombf(lm1, coef = 2, g = gmom)

      [,1]
[1,] 1.690808

> mombf(lm1, coef = 2, g = gtmom, baseDensity = "t")

[1] 0.007494312

> imombf(lm1, coef = 2, g = gimom, method = "adapt")

      [,1]
[1,] 1.714063

> imombf(lm1, coef = 2, g = gimom, method = "MC", B = 10^5)
```

```

      [,1]
[1,] 1.711426
> zellnerbf(lm1, coef = 2, g = 1)
      [,1]
[1,] 1.582311

```

We assess the Monte Carlo error by re-computing the iMom BF with a different set of Monte Carlo samples. We find the error to be acceptable.

```

> imombf(lm1, coef = 2, g = gimom, method = "MC", B = 10^5)
      [,1]
[1,] 1.711051

```

We now assess the sensitivity to the prior mode specification. For illustration purposes, we exclude the T MOM and iMom BF as these take longer to compute. The estimated standardized regression coefficient is

```

> sr <- sqrt(sum(lm1$residuals^2)/(nrow(hald) - 5))
> thest <- coef(lm1)[2]/sr
> thest
hald[, 2]
0.6341364

```

We define a sequence of prior modes, find the corresponding  $g$  values and compute Bayes factors. Note that `mombf`, `imombf` and `zellnerbf` accept  $g$  to be a vector instead of a single value. For large  $g$  vectors setting the option `method='MC'` in `imombf` can save considerable computing time, as the Monte Carlo samples need only be generated once for all  $g$  values.

```

> prior.mode <- seq(0.01, 1, length = 100)^2
> gmom <- mode2g(prior.mode, prior = "normalMom")
> bf1 <- mombf(lm1, coef = 2, g = gmom)
> bf2 <- zellnerbf(lm1, coef = 2, g = gmom)
> plot(prior.mode, bf1, type = "l", ylab = "BF")
> lines(prior.mode, bf2, lty = 2, col = 2)
> abline(v = thest, lty = 2)

```

The highest possible BF are observed when the prior mode is slightly smaller than the estimated 0.634. As the mode converges to zero both priors converge to a point mass at zero, and hence the BF converges to 1. As the mode goes to infinity the BF goes to 0, as predicted by Lindley's paradox (Lindley, 1957). Although the Mom and Zellner BF show some sensitivity to the prior specification, any prior mode between 0 and 1 results in evidence in favor of including the variable in the model.



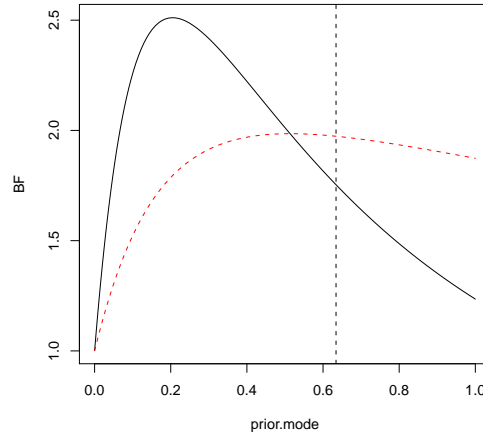


Figure 4: Hald data. BF obtained for Mom and Zellner's g-prior for several prior mode specifications.

### 3 Bayes factors for generalized linear regression models

As an illustration, we simulate data with 50 observations from a probit regression model. We simulate two correlated predictors with coefficients equal to  $\log(2)$  and 0 (*i.e.* the second variable is not actually in the model). The predictors are stored in the matrix `x`, the success probabilities in the vector `p` and the observed responses in the vector `y`. As in Section 2.2, for reproducibility purposes we set the random number generator seed to the date this document was produced.

```
> set.seed(4 * 2 * 2008)
> n <- 50
> theta <- c(log(2), 0)
> x <- matrix(NA, nrow = n, ncol = 2)
> x[, 1] <- rnorm(n, 0, 1)
> x[, 2] <- rnorm(n, 0.5 * x[, 1], 1)
> p <- pnorm(x %*% matrix(theta, ncol = 1))
> y <- rbinom(n, 1, p)
```

Before computing Bayes factors, we fit a probit regression model with the function `glm`. The maximum likelihood estimates are stored in `thetahat` and the asymptotic covariance matrix in `V`.

```
> glm1 <- glm(y ~ x[, 1] + x[, 2], family = binomial(link = "probit"))
> thetahat <- coef(glm1)
> V <- summary(glm1)$cov.scaled
```

To compute Bayes factors we use the functions `momknown` and `imomknown`. These functions take as primary arguments a vector of regression coefficients and their covariance matrix, and hence they can be used in any setting where one has a statistic that is asymptotically sufficient and normally distributed. The resulting Bayes factors are approximate. The functions also allow for the presence of a dispersion parameter `sigma`, *i.e.* the covariance of the regression coefficients is `sigma*V`, but they assume that `sigma` is known. The probit regression model that we simulated has no over-dispersion and hence it corresponds to `sigma=1`. We first compare the full model with the model resulting from excluding the second covariate, setting  $g = 0.5$  for illustration (note that `thetahat[1]` contains the intercept).

```
> g <- 0.5
> bfmom.1 <- momknown(thetahat[2], V[2, 2], n = n, g = g, sigma = 1)
> bfimom.1 <- imomknown(thetahat[2], V[2, 2], n = n, nuisance.theta = 2,
+   g = g, sigma = 1)
> bfmom.1
```

```
      [,1]
[1,] 4.262401
```

```
> bfimom.1
```

```
      [,1]
[1,] 3.336888
```

Both priors result in evidence for including the first covariate. We now check whether the second covariate can be dropped.

```
> bfmom.2 <- momknown(thetahat[3], V[3, 3], n = n, g = g, sigma = 1)
> bfimom.2 <- imomknown(thetahat[3], V[3, 3], n = n, nuisance.theta = 2,
+   g = g, sigma = 1)
> bfmom.2
```

```
      [,1]
[1,] 0.02784354
```

```
> bfimom.2
```

```

      [,1]
[1,] 0.00825012

```

Both Mom and iMom BF provide strong evidence in favor of the simpler model, *i.e.* excluding  $x[,2]$ . To compare the full model with the model that has no covariates (*i.e.* only the constant term remains) we use the same routines, passing a vector as the first argument and a matrix as the second argument.

```

> bfmom.0 <- momknown(thetahat[2:3], V[2:3, 2:3], n = n, g = g,
+   sigma = 1)
> bfimom.0 <- imomknown(thetahat[2:3], V[2:3, 2:3], n = n, nuisance.theta = 2,
+   g = g, sigma = 1)
> bfmom.0

```

```

      [,1]
[1,] 0.5272556

```

```

> bfimom.0

```

```

      [,1]
[1,] 0.953978

```

Based on the resulting BF being close to 1, it is not clear whether the full model is preferable to the model with no covariates.

The BF can be used to easily compute posterior probabilities for each of the four considered models: no covariates, only  $x[,1]$ , only  $x[,2]$  and both  $x[,1]$  and  $x[,2]$ . We assume equal probabilities *a priori*.

```

> prior.prob <- rep(1/4, 4)
> bf <- c(bfmom.0, bfmom.1, bfmom.2, 1)
> pos.prob <- prior.prob * bf/sum(prior.prob * bf)
> pos.prob

```

```

[1] 0.090632677 0.732686026 0.004786169 0.171895128

```

The model with the highest posterior probability is the one including only  $x[,1]$ , *i.e.* the correct model, and the model with the lowest posterior probability is that including only  $x[,2]$ .

## References

- M.J. Bayarri and G. Garcia-Donato. Extending conventional priors for testing general hypotheses in linear models. *Biometrika*, 94:135–152, 2007.
- V.E. Johnson. A technique for estimating marginal posterior densities in hierarchical models using mixtures of conditional densities. *Journal of American Statistical Association*, 87:852–860, 1992.
- D.V. Lindley. A statistical paradox. *Biometrika*, 44:187–192, 1957.
- A. Zellner and A. Siow. *Posterior odds ratios for selected regression hypotheses*, volume 1. Valencia: University Press, 1980.