

# Package ‘mombf’

January 4, 2012

**Version** 1.3.4

**Date** 2012-01-04

**Title** Moment and Inverse Moment Bayes factors

**Author** David Rossell, John D. Cook, Donatello Telesca, P. Roebuck

**Maintainer** David Rossell <rosselldavid@gmail.com>

**Depends** R (>= 2.11.0), methods, mvtnorm, ncvreg, actuar, mgcv

**Suggests** multicore

**Description** This package performs model selection based on non-local priors, including MOM, eMOM and iMOM priors. Routines are provided to compute Bayes factors, marginal densities and to perform variable selection in regression setups. Routines to evaluate prior densities, distribution functions, quantiles and modes are included.

**License** GPL (>= 2)

**URL** <http://mombf.r-forge.r-project.org/>

**LazyLoad** yes

**Collate** AllClasses.R AllGenerics.R msPriorSpec.R imombf.R mode2g.R  
pimom.R zellnerbf.knownsig.R imomknown.R modelSelection.R  
pmom.R zellnerbf.lm.R dimom.R imomunknown.R priorp2g.R  
zellnerbf.R margpimom.R mombf.lm.R qimom.R emom.R  
zellnerbf.unknownsig.R dmom.R mombf.R qmom.R g2mode.R  
margpmom.R momknown.R zbfknown.R imombf.lm.R momunknown.R  
zbfunknown.R pmomLM.R pmomPM.R emomLM.R postMode.R pplProbit.R  
greedyGLM.R ppmodel.R

**Repository** CRAN

**Date/Publication** 2012-01-04 13:57:42

## R topics documented:

bbPrior	2
dmom	3
hald	5
mode2g	5
modelSelection	7
mombf	10
momknown	12
msPriorSpec-class	14
nlpmarginals	15
pmomLM	17
<b>Index</b>	<b>22</b>

---

bbPrior	<i>Priors on model space for variable selection problems</i>
---------	--

---

### Description

unifPrior implements a uniform prior (equal a priori probability for all models). binomPrior implements a Binomial prior. bbPrior implements a Beta-Binomial prior.

### Usage

```
unifPrior(sel, logscale = TRUE)
binomPrior(sel, prob = 0.5, logscale = TRUE)
bbPrior(sel, alpha = 1, beta = 1, logscale = TRUE)
```

### Arguments

sel	Logical vector indicating which variables are included in the model
logscale	Set to TRUE to return the log-prior probability.
prob	Success probability for the Binomial prior
alpha	First parameter of the Beta-Binomial prior, which is equivalent to specifying a Beta(alpha,beta) prior on prob.
beta	First parameter of the Beta-Binomial prior, which is equivalent to specifying a Beta(alpha,beta) prior on prob.

### Value

Prior probability of the specified model

### Author(s)

David Rossell

## Examples

```
library(mombf)
sel <- c(TRUE,TRUE,FALSE,FALSE)
unifPrior(sel,logscale=FALSE)
binomPrior(sel,prob=.5,logscale=FALSE)
bbPrior(sel,alpha=1,beta=1,logscale=FALSE)
```

---

 dmom

*Moment prior and inverse moment prior.*


---

## Description

dmom, dimom and demom return the density for the moment, inverse moment and exponential moment priors. pmom, pimom and pemom return the distribution function for the univariate moment, inverse moment and exponential moment priors (respectively). qmom and qimom return the quantiles for the univariate moment and inverse moment priors.

## Usage

```
dmom(x, tau, a.tau, b.tau, phi=1, r=1, V1, baseDensity='normal', nu=3, logscale=FALSE, penalty='product')
dimom(x, tau=1, phi=1, V1, logscale=FALSE, penalty='product')
demom(x, tau, a.tau, b.tau, phi=1, logscale=FALSE)
```

```
pmom(q, V1 = 1, tau = 1)
pimom(q, V1 = 1, tau = 1, nu = 1)
pemom(q, tau, a.tau, b.tau)
```

```
qmom(p, V1 = 1, tau = 1)
qimom(p, V1 = 1, tau = 1, nu = 1)
```

## Arguments

x	In the univariate setting, x is a vector with the values at which to evaluate the density. In the multivariate setting it is a matrix with an observation in each row.
q	Vector of quantiles.
p	Vector of probabilities.
V1	Scale matrix. Defaults to 1 in univariate setting and the identity matrix in the multivariate setting.
tau	Prior dispersion parameter is tau*phi. See details.
a.tau	If tau is left missing, an Inverse Gamma(a.tau/2,b.tau/2) is placed on tau. In this case dmom and demom return the density marginalized with respect to tau.
b.tau	See a.tau.
phi	Prior dispersion parameter is tau*phi. See details.
r	Prior power parameter for MOM prior is 2*r

baseDensity	For baseDensity=='normal' a normal MOM prior is used, for baseDensity=='t' a T MOM prior with nu degrees of freedom is used.
nu	Prior parameter indicating the degrees of freedom for the quadratic T MOM and iMOM prior densities. The tails of the inverse moment prior are proportional to the tails of a multivariate T with nu degrees of freedom.
penalty	penalty=='product' indicates that product MOM/iMOM should be used. penalty=='quadratic' indicates quadratic iMOM. See Details.
logscale	For logscale==TRUE, dimom returns the natural log of the prior density.

### Details

For type=='quadratic' the density is as follows. Define the quadratic form  $q(\theta) = (\theta - \theta_0)' * \text{solve}(V1) * (\theta - \theta_0) / (\tau * \phi)$ . The normal moment prior density is proportional to  $q(\theta) * \text{dmvnorm}(\theta, \theta_0, \tau * \phi * V1)$ . The T moment prior is proportional to  $q(\theta) * \text{dmvt}(\theta, \theta_0, \tau * \phi * V1, \nu)$ . The inverse moment prior density is proportional to  $q(\theta)^{-(\nu+d)/2} * \exp(-1/q(\theta))$ .

pmom, pimom and qimom use closed-form expressions, while qmom uses `nlminb` to find quantiles numerically. Only the univariate version is implemented. In this case the product MOM is equivalent to the quadratic MOM. The same happens for the iMOM.

Only the product eMOM prior is implemented.

### Value

dmom returns the value of the moment prior density. dimom returns the value of the inverse moment prior density.

### Author(s)

David Rossell

### References

Johnson V.E., Rossell D. Non-Local Prior Densities for Default Bayesian Hypothesis Tests. *Journal of the Royal Statistical Society B*, 2010, 72, 143-170.

Johnson V.E., Rossell D. Bayesian model selection in high-dimensional settings. Technical report. 2011

See <http://rosselldavid.googlepages.com> for technical reports.

### See Also

[g2mode](#) to find the prior mode corresponding to a given g. [mode2g](#) to find the g value corresponding to a given prior mode.

### Examples

```
#evaluate and plot the moment and inverse moment priors
library(mombf)
tau <- 1
thseq <- seq(-3,3,length=1000)
```

```
plot(thseq,dmom(thseq,tau=tau),type='l',ylab='Prior density')
lines(thseq,dimom(thseq,tau=tau),lty=2,col=2)
```

hald

*Hald Data***Description**

Montgomery and Peck (1982) illustrated variable selection techniques on the Hald cement data and gave several references to other analysis. The response variable  $y$  is the *heat evolved* in a cement mix. The four explanatory variables are ingredients of the mix, i.e.,  $x_1$ : *tricalcium aluminate*,  $x_2$ : *tricalcium silicate*,  $x_3$ : *tetracalcium alumino ferrite*,  $x_4$ : *dicalcium silicate*. An important feature of these data is that the variables  $x_1$  and  $x_3$  are highly correlated ( $\text{corr}(x_1,x_3)=-0.824$ ), as well as the variables  $x_2$  and  $x_4$  (with  $\text{corr}(x_2,x_4)=-0.975$ ). Thus we should expect any subset of  $(x_1,x_2,x_3,x_4)$  that includes one variable from highly correlated pair to do as any subset that also includes the other member.

**Usage**

```
data(hald)
```

**Format**

`hald` is a matrix with 13 observations (rows) and 5 variables (columns), the first column is the dependent variable. `y.hald` and `x.hald` are also availables.

**Source**

Montgomery, D.C., Peck, E.A. (1982) *Introduction to linear regression analysis*, John Wiley, New York.

mode2g

*Moment and inverse moment prior elicitation.***Description**

`mode2g` finds the  $g$  value corresponding to a given prior mode. `g2mode` finds the prior mode corresponding to a given  $g$  value. `priorp2g` finds the  $g$  value giving `priorp` prior probability to the interval  $(-q,q)$ .

All routines operate in the standardized effect sizes scale.

**Usage**

```
mode2g(prior.mode, prior=c("iMom", "normalMom", "tMom"), nu=1, dim=1)
g2mode(g, prior=c("iMom", "normalMom", "tMom"), nu=1, dim=1)
priorp2g(priorp, q, nu=1, prior=c("iMom", "normalMom", "tMom"))
```

**Arguments**

prior.mode	Prior mode for the quadratic form $(\theta - \theta_0)' * \text{solve}(\Sigma) * (\theta - \theta_0) / (n * g * \sigma^2)$ , where $\sigma$ is the dispersion parameter and $\Sigma$ is given by the design matrix.
prior	prior=='normalMom' does computations for the normal moment prior, prior=='tMom' for the T moment prior, prior=='iMom' does computations for the inverse moment prior. Currently prior=='tMom' is not implemented in priorp2g.
nu	Prior degrees of freedom for the T moment prior or the iMom prior (ignored if prior=='normalMom').
dim	Dimensionality of the parameter, i.e. dim==1 for univariate, dim==2 for bivariate and so on.
g	Prior parameter. See dimom for details.
priorp	priorp2g returns g giving priorp prior probability to the interval $(-q, q)$ .
q	priorp2g returns g giving priorp prior probability to the interval $(-q, q)$ .

**Details**

See dmom and dimom for details on the meaning of the prior parameters.

**Value**

mode2g returns the value of the prior parameter g matching the given mode.

g2mode returns the prior mode for a given prior parameter g.

priorp2g returns g giving priorp prior probability to the interval  $(-q, q)$ .

**Author(s)**

David Rossell <rosselldavid@gmail.com>

**References**

See <http://rosselldavid.googlepages.com> for technical reports.

**See Also**

[dmom](#), [dimom](#), [mombf](#), [imombf](#)

**Examples**

```
#find g value giving a prior mode for  $(\theta / (\sigma * n * \Sigma))^2$  at  $0.2^2$ 
data(hald)
lm1 <- lm(hald[, 1] ~ hald[, 2] + hald[, 3] + hald[, 4] + hald[, 5])
prior.mode <- .2
gmom <- mode2g(prior.mode^2, prior='normalMom')
gtmom <- mode2g(prior.mode^2, prior='tMom', nu=3)
gimom <- mode2g(prior.mode^2, prior='iMom')
gmom
```

```

gimom

#find g value giving 0.05 probability to interval (-.2,.2)
priorp <- .05; q <- .2
gmom <- priorp2g(priorp=priorp, q=q, prior='normalMom')
gimom <- priorp2g(priorp=priorp, q=q, prior='iMom')
gmom
gimom

```

---

modelSelection

*Bayesian variable selection for linear models via non-local priors.*


---

## Description

modelSelection implements Bayesian model selection for several non-local prior formulations such as the (product) MOM and iMOM. The implemented algorithm uses a Gibbs scheme, so that a posterior sample reflecting the uncertainty in model selection is obtained.

## Usage

```

modelSelection(y, x, center = TRUE, scale = TRUE, niter = 10^4, thinning
= 1, burnin = round(niter/10), priorCoef, priorDelta, priorVar, phi,
deltaini = rep(FALSE, ncol(x)), initSearch = "greedy", method, B = 10^5,
verbose = TRUE)

```

## Arguments

y	Vector with observed responses
x	Design matrix with all potential predictors
center	If center==TRUE, y and x are centered to have zero mean, therefore eliminating the need to include an intercept term in x
scale	If scale==TRUE, y and columns in x are scaled to have standard deviation 1
niter	Number of Gibbs sampling iterations
thinning	MCMC thinning factor, i.e. only one out of each thinning iterations are reported. Defaults to thinning=1, i.e. no thinning
burnin	Number of burn-in MCMC iterations. Defaults to .1*niter. Set to 0 for no burn-in
priorCoef	Prior distribution for the coefficients. Must be object of class msPriorSpec with slot priorType set to 'coefficients'. Possible values for slot priorDistr are 'pMOM', 'piMOM' and 'peMOM'
priorDelta	Prior on model indicator space. Must be object of class msPriorSpec with slot priorType set to 'modelIndicator'. Possible values for slot priorDistr are 'uniform' and 'binomial'
priorVar	Prior on residual variance. Must be object of class msPriorSpec with slot priorType set to 'nuisancePars'. Slot priorDistr must be equal to 'invgamma'

phi	Residual variance. Typically this is unknown and therefore left missing. If specified argument priorVar is ignored
deltaini	Logical vector of length $ncol(x)$ indicating which coefficients should be initialized to be non-zero. Defaults to all variables being excluded from the model
initSearch	Algorithm to refine deltaini. <code>initSearch=='greedy'</code> uses a greedy Gibbs sampling search. <code>initSearch=='SCAD'</code> sets deltaini to the non-zero elements in a SCAD fit with cross-validated regularization parameter. <code>initSearch=='none'</code> leaves deltaini unmodified
method	Method to compute marginal densities. <code>method=='Laplace'</code> for Laplace approx, <code>method=='MC'</code> for Importance Sampling, <code>method=='Hybrid'</code> for Hybrid Laplace-IS (the latter method is only used for piMOM prior with unknown residual variance phi). See Details.
B	Number of samples to use in Importance Sampling scheme. Ignored if <code>method=='Laplace'</code>
verbose	Set <code>verbose==TRUE</code> to print iteration progress

### Details

Let delta be the vector indicating inclusion/exclusion of each column of  $x$  in the model. The algorithm sequentially samples from the posterior of each element in delta conditional on all the remaining elements in delta and the data. To perform this step it is necessary to evaluate the marginal density for any given delta, i.e. to integrate the likelihood of the data given  $(\beta, \delta)$  wrt the prior on  $\beta$  given  $(\delta, \phi)$  times the prior on the residual variance  $\phi$ .

As these marginal densities are not available in closed form we implement Laplace-like approximations and exact evaluation via Importance Sampling. For the iMOM prior we also implement a Hybrid Laplace-IS which uses a Laplace approximation to evaluate the integral wrt  $\beta$  and integrates wrt  $\phi$  numerically.

Regular Laplace approximations are used to compute marginals when the residual variance  $\phi$  is known. In the more typical case where the variance is unknown, direct Laplace approximations can be poor due to the posterior for  $\phi$  being highly non-normal. For MOM priors we integrate  $\phi$  analytically and use a Laplace approximation. For iMOM priors we re-parameterize to  $\log(\phi)$  and then use a regular Laplace approximation.

It should be noted that Laplace approximations tend to under-estimate the marginal densities when the MLE for some parameter is very close to 0. That is, it tends to be conservative in the sense of excluding more variables from the model than the exact calculations.

For more details on the methods used to compute marginal densities see Johnson & Rossell (2011).

### Value

A list with elements

postSample	matrix with posterior samples for the model indicator. <code>postSample[i, j]==1</code> indicates that variable $j$ was included in the model in the MCMC iteration $i$
postOther	<code>postOther</code> returns posterior samples for parameters other than the model indicator, i.e. basically hyper-parameters. If hyper-parameters were fixed in the model specification, <code>postOther</code> will be empty.

margpp	Marginal posterior probability for inclusion of each covariate. This is computed by averaging marginal post prob for inclusion in each Gibbs iteration, which is much more accurate than simply taking colMeans(postSample)
.	
postMode	Model with highest posterior probability amongst all those visited
postModeProb	Unnormalized posterior prob of posterior mode (log scale)
postProb	Unnormalized posterior prob of each visited model (log scale)
coef	Estimated coefficients (via posterior mode) for highest posterior probability model

**Author(s)**

David Rossell

**References**

Johnson V.E., Rossell D. Non-Local Prior Densities for Default Bayesian Hypothesis Tests. Journal of the Royal Statistical Society B, 2010, 72, 143-170. Johnson V.E., Rossell D. Bayesian model selection in high-dimensional settings. Technical report. 2011 See <http://rosselldavid.googlepages.com> for technical reports.

**See Also**

For more details on the prior specification see [msPriorSpec-class](#) To compute marginal densities for a given model see [pmomMarginalK](#), [pmomMarginalU](#), [pimomMarginalK](#), [pimomMarginalU](#).

**Examples**

```
#Simulate data
x <- matrix(rnorm(100*3),nrow=100,ncol=3)
theta <- matrix(c(1,1,0),ncol=1)
y <- x %*% theta + rnorm(100)

#Specify prior parameters
priorCoef <- new("msPriorSpec",priorType='coefficients',priorDistr='piMOM',priorPars=c(tau=1))
priorDelta <- new("msPriorSpec",priorType='modelIndicator',priorDistr='uniform',priorPars=double(0))
priorVar <- new("msPriorSpec",priorType='nuisancePars',priorDistr='invgamma',priorPars=c(alpha=.01,lambda=.01))

#Alternative prior for model space: 0.5 prior prob for including any covariate
priorDelta <-
new("msPriorSpec",priorType='modelIndicator',priorDistr='binomial',priorPars=c(p=.5))

#Alternative prior for model space: Beta hyper-prior for prob of inclusion
priorDelta <-
new("msPriorSpec",priorType='modelIndicator',priorDistr='binomial',priorPars=c(alpha.p=1,beta.p=1))

#Model selection
fit1 <- modelSelection(y=y, x=x, center=FALSE, scale=FALSE, niter=10^2,
priorCoef=priorCoef, priorDelta=priorDelta, priorVar=priorVar, phi=1,
```

```
method='Laplace')
fit1$postMode
fit1$margpp
```

---

mombf

*Moment and inverse moment Bayes factors for linear models.*


---

## Description

mombf computes moment Bayes factors to test whether a subset of regression coefficients are equal to some user-specified value. imombf computes inverse moment Bayes factors. zellnerbf computes Bayes factors based on the Zellner-Siow prior (used to build the moment prior).

## Usage

```
mombf(lm1, coef, g, prior.mode, baseDensity='normal', nu=3, theta0,
logbf=FALSE, B=10^5)
imombf(lm1, coef, g, prior.mode, nu = 1, theta0, method='adapt',
nquant=100, B = 10^5)
```

## Arguments

lm1	Linear model fit, as returned by lm1.
coef	Vector with indexes of coefficients to be tested. e.g. coef==c(2,3) and theta0==c(0,0) tests coef(lm1)[2]=coef(lm1)[3]=0.
g	Vector with prior parameter values. See dmom and dimom for details.
prior.mode	If specified, g is determined by calling g2mode.
baseDensity	Density upon which the Mom prior is based. baseDensity=='normal' results in the normal Mom prior, baseDensity=='t' in the t Mom prior with nu degrees of freedom.
nu	For mombf, nu specifies the degrees of freedom of the t Mom prior. It is ignored unless baseDensity=='t'. nu defaults to 3. For imombf, nu specifies the degrees of freedom for the inverse moment prior (see dimom for details). Defaults to nu=1, which Cauchy-like tails.
theta0	Null value for the regression coefficients. Defaults to 0.
logbf	If logbf==TRUE the natural logarithm of the Bayes factor is returned.
method	Numerical integration method to compute the bivariate integral (only used by imombf). For method=='adapt', the inner integral is evaluated (via integrate) at a series of nquant quantiles of the residual variance posterior distribution, and then averaged as described in Johnson (1992). Set method=='MC' to use Monte Carlo integration.
nquant	Number of quantiles at which to evaluate the integral for known sigma. Only used if method=='adapt'.
B	Number of Monte Carlo samples to estimate the T Mom and the inverse moment Bayes factor. Only used in mombf if baseDensity=='t'. Only used in imombf if method=='MC'.

## Details

These functions actually call `momunknown` and `imomunknown`, but they have a simpler interface. See `dmom` and `dimom` for details on the moment and inverse moment priors. The Zellner-Siow g-prior is given by `dmvnorm(theta,theta0,n*g*V1)`.

## Value

`mombf` returns the moment Bayes factor to compare the model where  $\theta \neq \theta_0$  with the null model where  $\theta = \theta_0$ . Large values favor the alternative model; small values favor the null. `imombf` returns inverse moment Bayes factors. `zellnerbf` returns Bayes factors based on the Zellner-Siow g-prior.

## Author(s)

David Rossell

## References

See <http://rosselldavid.googlepages.com> for technical reports. For details on the quantile integration, see Johnson, V.E. A Technique for Estimating Marginal Posterior Densities in Hierarchical Models Using Mixtures of Conditional Densities. *Journal of the American Statistical Association*, Vol. 87, No. 419. (Sep., 1992), pp. 852-860.

## See Also

[momunknown](#), [imomunknown](#) and [zbfunknown](#) for another interface to compute Bayes factors. [momknown](#), [imomknown](#) and [zbfknown](#) to compute Bayes factors assuming that the dispersion parameter is known, and for approximate Bayes factors for GLMs. [mode2g](#) for prior elicitation.

## Examples

```
##compute Bayes factor for Hald's data
data(hald)
lm1 <- lm(hald[,1] ~ hald[,2] + hald[,3] + hald[,4] + hald[,5])

# Set g so that prior mode for standardized effect size is at 0.2
prior.mode <- .2^2
V <- summary(lm1)$cov.unscaled
gmom <- mode2g(prior.mode,prior='normalMom')
gimom <- mode2g(prior.mode,prior='iMom')

# Set g so that interval (-0.2,0.2) has 5% prior probability
# (in standardized effect size scale)
priorp <- .05; q <- .2
gmom <- c(gmom,priorp2g(priorp=priorp,q=q,prior='normalMom'))
gimom <- c(gmom,priorp2g(priorp=priorp,q=q,prior='iMom'))

mombf(lm1,coef=2,g=gmom) #moment BF
imombf(lm1,coef=2,g=gimom,B=10^5) #inverse moment BF
zellnerbf(lm1,coef=2,g=1) #BF based on Zellner's g-prior
```

momknown

*Bayes factors for moment, inverse moment and Zellner-Siow g-prior.***Description**

momknown and momunknown compute moment Bayes factors for linear models when  $\sigma^2$  is known and unknown, respectively. The functions can also be used to compute approximate Bayes factors for generalized linear models and other settings. imomknown, imomunknown compute inverse moment Bayes factors. zbfknown, zbfunknown compute Bayes factors based on the Zellner-Siow g-prior.

**Usage**

```
momknown(theta1hat, V1, n, g = 1, theta0, sigma, logbf = FALSE)
momunknown(theta1hat, V1, n, nuisance.theta, g = 1, theta0, ssr, logbf =
FALSE)
imomknown(theta1hat, V1, n, nuisance.theta, g = 1, nu = 1, theta0,
sigma, method='adapt', B=10^5)
imomunknown(theta1hat, V1, n, nuisance.theta, g = 1, nu = 1, theta0,
ssr, method='adapt', nquant = 100, B = 10^5)
zbfknown(theta1hat, V1, n, g = 1, theta0, sigma, logbf = FALSE)
zbfunknown(theta1hat, V1, n, nuisance.theta, g = 1, theta0, ssr, logbf =
FALSE)
```

**Arguments**

theta1hat	Vector with regression coefficients estimates.
V1	Matrix proportional to the covariance of theta1hat. For linear models, the covariance is $\sigma^2 \cdot V1$ .
n	Sample size.
nuisance.theta	Number of nuisance regression coefficients, i.e. coefficients that we do not wish to test for.
ssr	Sum of squared residuals from a linear model call.
g	Prior parameter. See dmom and dimom for details.
theta0	Null value for the regression coefficients. Defaults to 0.
sigma	Dispersion parameter is $\sigma^2$ .
logbf	If logbf==TRUE the natural logarithm of the Bayes factor is returned.
nu	Prior parameter for the inverse moment prior. See dimom for details. Defaults to nu=1, which Cauchy-like tails.
method	Numerical integration method (only used by imomknown and imomunknown). Set method=='adapt' in imomknown to integrate using adaptive quadrature of functions as implemented in the function integrate. In imomunknown the integral is evaluated as in imomknown at a series of nquant quantiles of the posterior for sigma, and then averaged as described in Johnson (1992). Set method=='MC' to use Monte Carlo integration.

nquant	Number of quantiles at which to evaluate the integral for known sigma.
B	Number of Monte Carlo samples to estimate the inverse moment Bayes factor. Ignored if method!='MC'.

### Details

See `dmom` and `dimom` for details on the moment and inverse moment priors. The Zellner-Siow g-prior is given by `dmvnorm(theta,theta0,n*g*V1)`.

### Value

`momknown` and `momunknown` return the moment Bayes factor to compare the model where  $\theta \neq \theta_0$  with the null model where  $\theta = \theta_0$ . Large values favor the alternative model; small values favor the null. `imomknown` and `imomunknown` return inverse moment Bayes factors. `zbfknown` and `zbfunknown` return Bayes factors based on the Zellner-Siow g-prior.

### Author(s)

David Rossell

### References

See <http://rosselldavid.googlepages.com> for technical reports.

For details on the quantile integration, see Johnson, V.E. A Technique for Estimating Marginal Posterior Densities in Hierarchical Models Using Mixtures of Conditional Densities. *Journal of the American Statistical Association*, Vol. 87, No. 419. (Sep., 1992), pp. 852-860.

### See Also

[mombf](#) and [imombf](#) for a simpler interface to compute Bayes factors in linear regression. [mode2g](#) for prior elicitation.

### Examples

```
#simulate data from probit regression
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,.5*x[,1],1)
p <- pnorm(x[,1]*theta[1]+x[,2]+theta[2])
y <- rbinom(n,1,p)

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

#compute Bayes factors to test whether x[,1] can be dropped from the model
g <- .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  
bfimom.1

---

msPriorSpec-class      *Class "msPriorSpec"*

---

### Description

Stores the prior distributions to be used for Bayesian variable selection in normal regression models. This class can be used to specify the prior on non-zero regression coefficients, the model indicator or the nuisance parameters.

### Objects from the Class

Objects can be created by calls of the form `new("msPriorSpec", ...)`.

### Slots

**priorType:** Object of class "character". "coefficients" indicates that the prior is for the non-zero regression coefficients. "modelIndicator" that it is for the model indicator, and "nuisancePars" that it is for the nuisance parameters. Several prior distributions are available for each choice of priorType, and these can be specified in the slot priorDist.

**priorDistr:** Object of class "character". If priorType=="coefficients", priorDistr can be equal to "pMOM", "piMOM", or "peMOM" (product moment, product inverse moment or product exponential moment prior, respectively). If priorType=="modelIndicator", priorDistr can be equal to "uniform" or "binomial" to specify a uniform prior (all models equally likely a priori) or a binomial prior. For a binomial prior, the prior inclusion probability for any single variable must be specified in slot priorPars['p']. For a beta-binomial prior, the Beta hyper-prior parameters must be in priorPars['alpha.p'] and priorPars['beta.p']. If priorType=="nuisancePars", priorDistr must be equal to "invgamma". This corresponds to an inverse gamma distribution for the residual variance, with parameters specified in the slot priorPars.

**priorPars:** Object of class "vector", where each element must be named. For priorDistr=='pMOM', there must be an element "r" (MOM power is 2r). For any priorDistr there must be either an element "tau" indicating the prior dispersion or elements "a.tau" and "b.tau" specifying an inverse gamma hyper-prior for "tau". Optionally, there may be an element "tau.adj" indicating the prior dispersion for the adjustment variables (i.e. not undergoing variable selection). If not defined, "tau.adj" is set to 0.001 by default. For priorDistr=='binomial', there must be either an element "p" specifying the prior inclusion probability for any single covariate, or a vector with elements "alpha.p" and "beta.p" specifying a Beta(alpha.p,beta.p) hyper-prior on p. For priorDistr=='invgamma' there must be elements "alpha" and "lambda". The prior for the residual variance is an inverse gamma with parameters  $.5 \cdot \alpha$  and  $.5 \cdot \lambda$ .

### Methods

No methods defined with class "msPriorSpec" in the signature.

**Note**

When new instances of the class are created a series of check are performed to ensure that a valid prior specification is produced.

**Author(s)**

David Rossell.

**References**

Prior Densities for Default Bayesian Hypothesis Tests. Journal of the Royal Statistical Society B, 2010, 72, 143-170

**See Also**

See also [modelSelection](#) for an example of defining an instance of the class and perform Bayesian model selection.

**Examples**

```
showClass("msPriorSpec")
```

---

nlpmarginals

*Marginal density of the observed data for linear models under a given model, using non-local prior distributions for the regression coefficients.*

---

**Description**

The marginal density of the observed data, i.e. the likelihood integrated with respect to a non-local prior distribution on the regression coefficients of the variables included in the model and (optionally) an inverse gamma prior on the residual variance. `pimomMarginalK` and `pimomMarginalU` use a product iMOM prior on the regression coefficients for the known and unknown variance cases (respectively). `pmomMarginalK` and `pmomMarginalU` use a product MOM prior.

**Usage**

```
pimomMarginalK(sel, y, x, phi, tau=1, method='Laplace', B=10^5, logscale=TRUE, XtX, ytX)
pimomMarginalU(sel, y, x, alpha=0.001, lambda=0.001, tau=1,
method='Laplace', B=10^5, logscale=TRUE, XtX, ytX)
pmomMarginalK(sel, y, x, phi, tau, r=1, method='Laplace', B=10^5,
logscale=TRUE, XtX, ytX)
pmomMarginalU(sel, y, x, alpha=0.001, lambda=0.001, tau=1,
r=1, method='Laplace', B=10^5, logscale=TRUE, XtX, ytX)
```

**Arguments**

sel	Vector with indexes of columns in x to be included in the model
y	Vector with observed responses
x	Design matrix with covariates. Only the columns specified in sel are included in the model, the rest are disregarded
phi	Residual variance, assumed to be known by <code>pimomMarginalK</code> and <code>pmomMarginalK</code>
alpha	Prior for phi is inverse gamma $\alpha/2, \lambda/2$
lambda	Prior for phi is inverse gamma $\alpha/2, \lambda/2$
tau	Prior dispersion parameter for MOM and iMOM priors (see details)
r	Prior power parameter for MOM prior is $2*r$
method	Method to approximate the integral. 'Laplace' for Laplace approx. 'MC' for Monte Carlo. 'Plug-in' for plug-in estimate. 'Hybrid' for hybrid Laplace-MC (only available for <code>pimomMarginalU</code> ). See help for <code>modelSelection</code> for details.
B	Number of Monte Carlo samples to use (ignored unless <code>method=='MC'</code> )
logscale	If <code>logscale==TRUE</code> the log marginal density is returned.
XtX	Optionally, specify the matrix $X'X$ . Useful when the function must be called a large number of times.
ytX	Optionally, specify the vector $y'X$ . Useful when the function must be called a large number of times.

**Details**

The marginal density of the data is equal to the integral of  $N(y;x[,sel]*\theta,\phi*I) * \pi(\theta|\phi,\tau) * IG(\phi;\alpha/2,\lambda/2)$  with respect to  $\theta$ , where  $\pi(\theta|\phi,\tau)$  is a non-local prior and IG denotes the density of an inverse gamma.

`pmomMarginalK` and `pimomMarginalK` assume that the residual variance is known and therefore the inverse-gamma term in the integrand can be omitted.

The product MOM and iMOM densities can be evaluated using the functions `dmom` and `dimom`.

**Value**

Marginal density of the observed data under the specified prior.

**Author(s)**

David Rossell

**References**

Johnson V.E., Rossell D. Non-Local Prior Densities for Default Bayesian Hypothesis Tests. *Journal of the Royal Statistical Society B*, 2010, 72, 143-170. See <http://rosselldavid.googlepages.com> for technical reports.

**See Also**

[modelSelection](#) to perform model selection based on product non-local priors. [momunknown](#), [imomunknown](#), [momknown](#), [imomknown](#) to compute Bayes factors for additive MOM and iMOM priors. [mode2g](#) for prior elicitation.

**Examples**

```
x <- matrix(rnorm(100*2),ncol=2)
y <- x %%% matrix(c(.5,1),ncol=1) + rnorm(nrow(x))
pmomMarginalK(sel=1, y=y, x=x, phi=1, tau=1, method='Laplace')
pmomMarginalK(sel=1:2, y=y, x=x, phi=1, tau=1, method='Laplace')
```

pmomLM

*Bayesian variable selection and model averaging for linear and probit models via non-local priors.*

**Description**

Variable selection for linear and probit models, providing a sample from the joint posterior of the model and regression coefficients. pmomLM and pmomPM implement product Normal MOM and heavy-tailed product MOM as prior distribution for linear and probit model coefficients (respectively). emomLM and emomPM set an eMOM prior.

pp1PM finds the value of the prior dispersion parameter tau minimizing posterior expected predictive loss (Gelfand and Ghosh, 1998) for the Probit model, i.e. can be used to automatically set up tau.

ppmodel returns the proportion of visits to each model.

**Usage**

```
pmomLM(y, x, xadj, center=FALSE, scale=FALSE, niter=10^4, thinning=1, burnin=round(niter/10), priorCoe
pmomPM(y, x, xadj, niter=10^4, thinning=1, burnin=round(niter/10),
priorCoef, priorDelta, initSearch='greedy', verbose=TRUE)
```

```
emomLM(y, x, xadj, center=FALSE, scale=FALSE, niter=10^4, thinning=1, burnin=round(niter/10), priorCoe
emomPM(y, x, xadj, niter=10^4, thinning=1, burnin =round(niter/10),
priorCoef, priorDelta, initSearch='greedy', verbose=TRUE)
```

```
pp1PM(tauseq=exp(seq(log(.01),log(2),length=20)), kPen=1, y, x, xadj, niter=10^4,
thinning=1, burnin=round(niter/10), priorCoef, priorDelta, priorVar,
initSearch='greedy', mc.cores=1)
```

```
ppmodel(nlpfit)
```

**Arguments**

<code>y</code>	Vector with observed responses. For <code>pmomLM</code> this must be a numeric vector. For <code>pmomPM</code> it can either be a logical vector, a factor with 2 levels or a numeric vector taking only two distinct values.
<code>x</code>	Design matrix with all potential predictors which are to undergo variable selection.
<code>xadj</code>	Design matrix for adjustment covariates, i.e. variables which are included in the model with probability 1. For instance, <code>xadj</code> can be used to force the inclusion of an intercept in the model.
<code>center</code>	If <code>center==TRUE</code> , <code>y</code> and <code>x</code> are centered to have zero mean, therefore eliminating the need to include an intercept term in <code>x</code>
<code>scale</code>	If <code>scale==TRUE</code> , <code>y</code> and columns in <code>x</code> are scaled to have standard deviation 1
<code>niter</code>	Number of MCMC sampling iterations
<code>thinning</code>	MCMC thinning factor, i.e. only one out of each thinning iterations are reported. Defaults to <code>thinning=1</code> , i.e. no thinning
<code>burnin</code>	Number of burn-in MCMC iterations. Defaults to <code>.1*niter</code> . Set to 0 for no burn-in
<code>priorCoef</code>	Prior distribution for the coefficients. Must be object of class <code>msPriorSpec</code> with slot <code>priorType</code> set to 'coefficients'. Possible values for slot <code>priorDistr</code> are 'pMOM', 'piMOM' and 'peMOM'
<code>priorDelta</code>	Prior on model indicator space. Must be object of class <code>msPriorSpec</code> with slot <code>priorType</code> set to 'modelIndicator'. Possible values for slot <code>priorDistr</code> are 'uniform' and 'binomial'. For 'binomial', you can either set the prior probability 'p' or specify a Beta-binomial prior by specifying the parameters 'alpha.p','beta.p'.
<code>priorVar</code>	Prior on residual variance. Must be object of class <code>msPriorSpec</code> with slot <code>priorType</code> set to 'nuisancePars'. Slot <code>priorDistr</code> must be equal to 'invgamma'
<code>initSearch</code>	Algorithm to refine <code>deltaini</code> . <code>initSearch=='greedy'</code> uses a greedy Gibbs sampling search. <code>initSearch=='SCAD'</code> sets <code>deltaini</code> to the non-zero elements in a SCAD fit with cross-validated regularization parameter. <code>initSearch=='none'</code> initializes to the null model with no variables in <code>x</code> included.
<code>verbose</code>	Set <code>verbose==TRUE</code> to print iteration progress
<code>tauseq</code>	Grid of tau values for which the posterior predictive loss should be evaluated.
<code>kPen</code>	Penalty term specifying the relative importance of deviations from the observed data vs deviation from the posterior predictive. <code>kPen</code> can be set either to a numeric value or to 'msize' to set penalty equal to the average model size. Loss is $Dev(y_p, \hat{y}) + kPen * Dev(\hat{y}, y_{obs})$ , where <code>y<sub>p</sub></code> : draw from post predictive, <code>y<sub>obs</sub></code> : observed data and <code>yhat</code> is $E(y y_{obs})$ .
<code>mc.cores</code>	Allows for parallel computing. <code>mc.cores</code> is the number of processors to use. Setting <code>mc.cores&gt;1</code> requires the <code>multicore</code> package.
<code>nlpfit</code>	Non-local prior model fit, as returned by <code>pmomLM</code> , <code>pmomPM</code> , <code>emomLM</code> or <code>emomPM</code> .

**Details**

The implemented MCMC scheme makes proposals from the joint posterior of  $(\delta[i], \theta[i])$  given all other parameters and the data, where  $\delta[i]$  is the indicator for inclusion/exclusion of covariate  $i$  and  $\theta[i]$  is the coefficient value. In contrast with some model fitting options implemented in `modelSelection`, here the scheme is exact. However, sampling the coefficients can adversely affect the mixing when covariates are very highly correlated. In practice, the mixing seems to be reasonably good for correlations up to 0.9.

`pmomPM` uses the scheme of Albert & Chib (1993) for probit models.

**Value**

`pmomLM` and `pmomPM` returns a list with elements

<code>postModel</code>	matrix with posterior samples for the model indicator. <code>postModel[i, j]==1</code> indicates that variable $j$ was included in the model in the MCMC iteration $i$
<code>postCoef1</code>	matrix with posterior samples for coefficients associated to $x$
<code>postCoef2</code>	matrix with posterior samples for coefficients associated to $x_{adj}$
<code>postPhi</code>	vector with posterior samples for residual variance
<code>postOther</code>	<code>postOther</code> returns posterior samples for other parameters, i.e. basically hyper-parameters. Currently the prior precision parameter $\tau$
<code>margpp</code>	Marginal posterior probability for inclusion of each covariate. This is computed by averaging marginal post prob for inclusion in each MCMC iteration, which is much more accurate than simply taking <code>colMeans(postModel)</code>

.

`pp1PM` returns a list with elements

<code>optfit</code>	Probit model fit using <code>tauopt</code> . It is the result of a call to <code>pmomPM</code> .
<code>PPL</code>	<code>data.frame</code> indicating for each value in <code>tauseq</code> the posterior predictive loss ( $PPL=G+P$ ), the goodness-of-fit ( $G$ ) and penalty terms ( $P$ )

, the average number of covariates in the model (`msize`) including `xadj` and the smoothed `sPPL` obtained via a `gam` fit.

<code>tautopt</code>	Value of $\tau$ minimizing the PPL
----------------------	------------------------------------

**Author(s)**

David Rossell, Donatello Telesca

**References**

Johnson V.E., Rossell D. Non-Local Prior Densities for Default Bayesian Hypothesis Tests. *Journal of the Royal Statistical Society B*, 2010, 72, 143-170.

Johnson V.E., Rossell D. Bayesian model selection in high-dimensional settings. Technical report. 2011 See <http://rosselldavid.googlepages.com> for technical reports.

Albert, J. and Chib, S. (1993) Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88, p669-679

Gelfand, A. and Ghosh, S. (1998) Model choice: A minimum posterior predictive loss approach. *Biometrika*, 85, p1-11.

### See Also

For more details on the prior specification see [msPriorSpec-class](#) To compute marginal densities for a given model see [pmomMarginalK](#), [pmomMarginalU](#), [pimomMarginalK](#), [pimomMarginalU](#).

### Examples

```
#Simulate data
x <- matrix(rnorm(100*3),nrow=100,ncol=3)
xadj <- rep(1,nrow(x))
theta <- matrix(c(1,1,0),ncol=1)
y <- 10*xadj + x %%% theta + rnorm(100)

#Beta-binomial prior on model space
priorDelta <-
new("msPriorSpec",priorType='modelIndicator',priorDistr='binomial',priorPars=c(alpha.p=1,beta.p=1))

#Non-informative prior on residual variance
priorVar <-
new("msPriorSpec",priorType='nuisancePars',priorDistr='invgamma',priorPars=c(alpha=.01,lambda=.01))

#Product MOM prior with tau=0.3 on x coefficients
#Non-informative prior on xadj coefficients
priorCoef <- new("msPriorSpec",priorType='coefficients',priorDistr='pMOM',priorPars=c(tau=.3,tau.adj=10^6,r=1))

mom0 <- pmomLM(y=y,x=x,xadj=xadj,center=FALSE,scale=FALSE,niter=1000,priorCoef=priorCoef,priorDelta=priorDelta,
round(colMeans(mom0$postModel),2)
round(colMeans(mom0$postCoef1),2)
round(colMeans(mom0$postCoef2),2)

#Alternative prior: hyper-prior on tau
priorCoef <- new("msPriorSpec",priorType='coefficients',priorDistr='pMOM',priorPars=c(a.tau=1,b.tau=.135,tau.adj=10^6))
mom1 <-
pmomLM(y=y,x=x,xadj=xadj,center=FALSE,scale=FALSE,niter=1000,priorCoef=priorCoef,priorDelta=priorDelta,priorVar=priorVar)
mean(mom1$postOther) #posterior mean for tau

#Probit model
n <- 500; rho <- .25; niter <- 1000
theta <- c(.4,.6,0); theta.adj <- 0
V <- diag(length(theta)); V[upper.tri(V)] <- V[lower.tri(V)] <- rho
x <- rmvnorm(n,rep(0,length(theta)),V); xadj <- matrix(1,nrow=nrow(x),ncol=1)
lpred <- as.vector(x %%% matrix(theta,ncol=1) + xadj %%% matrix(theta.adj,ncol=1))
p <- pnorm(lpred)
y <- runif(n)<p

mom2 <- pmomPM(y=y,x=x,xadj=xadj,niter=1000,priorCoef=priorCoef,priorDelta=priorDelta,initSearch='greedy')
colMeans(mom2$postCoef1)
```

```
coef(glm(y ~ x + xadj -1, family=binomial(link='probit')))
```

# Index

- \*Topic **classes**
  - msPriorSpec-class, 14
- \*Topic **datasets**
  - hald, 5
- \*Topic **distribution**
  - bbPrior, 2
  - dmom, 3
  - mode2g, 5
- \*Topic **htest**
  - dmom, 3
  - mode2g, 5
  - modelSelection, 7
  - mombf, 10
  - momknown, 12
  - nlpmarginals, 15
  - pmomLM, 17
- \*Topic **models**
  - modelSelection, 7
  - mombf, 10
  - momknown, 12
  - nlpmarginals, 15
  - pmomLM, 17
- bbPrior, 2
- binomPrior (bbPrior), 2
- demom (dmom), 3
- demom, data.frame-method (dmom), 3
- demom, matrix-method (dmom), 3
- demom, vector-method (dmom), 3
- demom-methods (dmom), 3
- dimom, 6
- dimom (dmom), 3
- dmom, 3, 6
- emomLM (pmomLM), 17
- emomPM (pmomLM), 17
- g2mode, 4
- g2mode (mode2g), 5
- hald, 5
- imombf, 6, 13
- imombf (mombf), 10
- imomknown, 11, 17
- imomknown (momknown), 12
- imomunknown, 11, 17
- imomunknown (momknown), 12
- mode2g, 4, 5, 11, 13, 17
- modelSelection, 7, 15, 17
- mombf, 6, 10, 13
- momknown, 11, 12, 17
- momunknown, 11, 17
- momunknown (momknown), 12
- msPriorSpec-class, 9, 20
- msPriorSpec-class, 14
- nlpmarginals, 15
- pemom (dmom), 3
- pimom (dmom), 3
- pimomMarginalK, 9, 20
- pimomMarginalK (nlpmarginals), 15
- pimomMarginalU, 9, 20
- pimomMarginalU (nlpmarginals), 15
- pmom (dmom), 3
- pmomLM, 17
- pmomMarginalK, 9, 20
- pmomMarginalK (nlpmarginals), 15
- pmomMarginalU, 9, 20
- pmomMarginalU (nlpmarginals), 15
- pmomPM (pmomLM), 17
- pp1PM (pmomLM), 17
- ppmodel (pmomLM), 17
- priorp2g (mode2g), 5
- qimom (dmom), 3
- qmom (dmom), 3
- unifPrior (bbPrior), 2

x.hald (hald), 5

y.hald (hald), 5

zbfknown, 11

zbfknown (momknown), 12

zbfunknown, 11

zbfunknown (momknown), 12

zellnerbf (mombf), 10