Package ‘mombf’

July 24, 2009

Version 1.0.4

Date 2009-07-17

Title Moment and Inverse Moment Bayes factors

Author David Rossell <rosselldavid@gmail.com>.

Maintainer David Rossell <rosselldavid@gmail.com>

Depends R (>= 2.5.0), mvtnorm

Description This package computes Moment and Inverse Moment Bayes factors for linear models, and approximate Bayes factors for GLM and situations having a statistic which is asymptotically normally distributed and sufficient. Routines to evaluate prior densities, distribution functions, quantiles and modes are included.

License GPL (>= 2)

LazyLoad no

Repository CRAN

Date/Publication 2009-07-24 06:06:29

R topics documented:

`dmom` *Moment prior and inverse moment prior.*

Description

`dmom` and `dimom` return the density for the moment and inverse moment priors, in the univariate and multivariate setting. `pmom` and `pimom` return the distribution function for the univariate moment and inverse moment priors. `qmom` and `qimom` return the quantiles for the univariate moment and inverse moment priors.

Usage

```
dmom(x, V1 = 1, g = 1, n = 1, theta0, baseDensity = "normal", nu = 3)
dimom(x, V1 = 1, g = 1, n = 1, nu = 1, theta0, logscale = FALSE)
pmom(q, V1 = 1, g = 1, n = 1)
pimom(q, V1 = 1, g = 1, n = 1, nu = 1)
qmom(p, V1 = 1, g = 1, n = 1)
qimom(p, V1 = 1, g = 1, n = 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.
`theta0`	Location parameter. Defaults to 0.
`V1`	Scale matrix. Defaults to 1 in univariate setting and the identity matrix in the multivariate setting.
`g`	Prior parameter. See details.
`n`	Prior parameter. See details.
`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 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.
`logscale`	For `logscale==TRUE`, `dimom` returns the natural log of the prior density.

Details

Define the quadratic form q(theta)= (theta-theta0)' * solve(V1) * (theta-theta0) / (n*g). The normal moment prior density is proportional to q(theta)*dmvnorm(theta,theta0,n*g*V1). The T moment prior is proportional to q(theta)*dmvt(theta,theta0,n*g*V1,df=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.

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

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)
g <- 1
thseq <- seq(-3,3,length=1000)
plot(thseq,dmom(thseq,g=g),type='l',ylab='Prior density')
lines(thseq,dimom(thseq,g=g),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., x1: *tricalcium aluminate*, x2: *tricalcium silicate*, x3: *tetracalcium alumino ferrite*, x4: *dicalcium silicate*. An important feature of these data is that the variables x1 and x3 are highly correlated (corr(x1,x3)=-0.824), as well as the variables x2 and x4 (with corr(x2,x4)=-0.975). Thus we should expect any subset of (x1,x2,x3,x4) 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 = 'iMom', nu = 1, dim = 1)
g2mode(g, prior = 'iMom', nu = 1, dim = 1)
priorp2g(priorp, q, nu = 1, prior = 'iMom')
```

Arguments

`prior.mode`	Prior mode for the quadratic form (theta-theta0)' * solve(Sigma) * (theta-theta0)/(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

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
```

---

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!=theta0` with the null model where `theta==theta0`. 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*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!=theta0 with the null model where theta==theta0. 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
```

Index