Package ‘iLaplace’

August 19, 2016

Type Package

Title Improved Laplace Approximation for Integrals of Unimodal Functions

Version 1.1.0

Date 2015-12-31

Description Improved Laplace approximation for integrals of unimodal functions. The method requires user-supplied R functions for: the integrand, its gradient and its Hessian matrix. The computations are run in parallel.

Depends R (>= 3.0.1), fastGHQuad

Imports doParallel, foreach, iterators, Rcpp, stats

LinkingTo Rcpp, RcppArmadillo

Suggests devtools

License GPL-2

URL https://github.com/erlisR/iLaplace

BugReports https://github.com/erlisR/iLaplace/issues

Author Erlis Ruli [aut, cre], Nicola Sartori [aut], Laura Ventura [aut]

Maintainer Erlis Ruli <erlisr@yahoo.it>

LazyData true

RoxygenNote 5.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-08-19 16:30:33

R topics documented:

`iLaplace-package` *iLaplace: A package for Approximating Multidimensional Integrals of Unimodal Functions*

Description

This package gives (a parallel) implementation of the improved Laplace approximation for multivariate integrals of user-written unimodal functions (see Ruli et al. 2015). The method essentially approximates the target integral by the ratio of the (unnormalised) integrand and an approximation of its normalised version, both evaluated at the modal value. The normalised integrand is obtained through a sequential application of the Laplace approximation for marginal densities. Like the standard Laplace approximation, the improved Laplace approximation is a deterministic method which approximates intractable multidimensional integrals by (essentially) numerical optimisations. However, whith respect to the Laplace approximation, the improved Laplace involves scalar numerical integrations. Nevertheless, the improved Laplace approximation tends to be fast and extremely accurate, especially with skewed fat-tailed integrands.

Details

Currently the packages provides two functions `iLap` and `ilap2d` which perform approximation of d-variate and bivariate integrals, respectively.

Author(s)

Erlis Ruli `<erlisr@yahoo.it>`

References

Ruli E., Sartori N. & Ventura L. (2015) Improved Laplace approximation for marignal likelihoods. `http://arxiv.org/abs/1502.06440`

See Also

`iLap` and `iLap2d` for more details and examples.

`iLap` *Improved Laplace approximation for integrals of unimodal functions*

Description

This function implements the improved Laplace approximation of Ruli et al. (2015) for multivariate integrals of user-written unimodal functions. See "Details" below for more information.

Usage

```
iLap(fullOpt, ff, ff.gr, ff.hess, quad.data, ...)
```

Arguments

`fullOpt`	A list containing the minium (to be accesed via `fullOpt$par`), the value of the function at the minimum (to be accessed via `fullOpt$objective`) and the Hessian matrix at the minimum (to be accessed via `fullOpt$hessian`
`ff`	The minus logarithm of the integrand function (the `h` function; see "Details").
`ff.gr`	The gradient of `ff`, having the exact same arguments as `ff`.
`ff.hess`	The Hessian matrix of`ff`, having the exact same arguments as `ff`.
`quad.data`	Data for the Gaussian-Herimte quadratures; see "Details"
`...`	Additional arguments to be passed to `ff`, `ff.gr` and `ff.hess`

Details

`iLap` approximates integrals of the type

$$I = \int_{x \in \mathcal{R}^d} \exp\{-h(x)\}\, dx$$

where $-h(\cdot)$ is a concave and unimodal function, with x being d dimensional real vector ($d > 1$). The approximation of I is obtained as the ratio between the unormalised kernel $-h(x)$ and an approximate density function $f(x)$, both evaluated at the modal value $x = \hat{x}$. The approximate density function $f(x)$ is obtained by resorting to the Laplace approximation for marginal densities. The minimisations are performed with `nlminb` by suppling the gradient `ff.gr` and Hessian matrix ff.hess of $f(x)$. The normalisation of the univariate components is perforemd via Gaussian-Hermite quadratures as implemented in the function `aghQuad`. The Gaussian-Quadrature data, to be provided via the argument `quad.data`, can be computed with the function `gaussHermiteData` for a desired number of quadrature points. See "Examples" below.

Value

A double, the logarithm of the integral

References

Ruli E., Sartori N. & Ventura L. (2015) Improved Laplace approximation for marignal likelihoods. http://arxiv.org/abs/1502.06440

Liu, Q. and Pierce, D. A. (1994). A Note on Gauss-Hermite Quadrature. *Biometrika* **81**, 624-629.

Examples

```
# The negative integrand function in log
# is the negative log-density of the multivariate
# Student-t density centred at 0 with unit scale matrix
ff <- function(x, df) {
```

```
        d <- length(x)
        S <- diag(1, d, d)
        S.inv <- solve(S)
        Q <- colSums((S.inv %*% x) * x)
        logDet <- determinant(S)$modulus
        logPDF <- (lgamma((df + d)/2) - 0.5 * (d * logb(pi * df) +
        logDet) - lgamma(df/2) - 0.5 * (df + d) * logb(1 + Q/df))
        return(-logPDF)
        }

# the gradient of ff
ff.gr <- function(x, df){
            m <- length(x)
            kr = 1 + crossprod(x,x)/df
            return((m+df)*x/(df*kr))
            }

# the Hessian matrix of ff
ff.hess <- function(x, df) {
m <- length(x)
kr <- as.double(1 + crossprod(x,x)/df)
ll <- -(df+m)*2*tcrossprod(x,x)/(df*kr)^2.0
dd = (df+m)*(kr - 2*x^2/df)/(df*kr^2.0)
diag(ll) = dd;
return(ll)
}

df = 5
dims <- 5:8
normConts <- sapply(dims, function(mydim) {
opt <- nlminb(rep(1,mydim), ff, gradient = ff.gr, hessian = ff.hess, df = df)
opt$hessian <- ff.hess(opt$par, df = df);
quad.data = gaussHermiteData(50)
iLap <- iLap(opt, ff, ff.gr, ff.hess, quad.data = quad.data, df = df);
Lap <- mydim*log(2*pi)/2 - opt$objective - 0.5*determinant(opt$hessian)$mod;
return(c(iLap = iLap, Lap = Lap))
})
# plot the results
## Not run:
plot(dims, normConts[1,], pch="*", cex = 1.6,
 ylim = c(-5, 0)) #improved Laplace
lines(dims, normConts[2,], type = "p", pch = "+") #standard Laplace
abline(h = 0) # the true value

## End(Not run)

## Not run:
## See also the examples provided in the pacakge iLaplaceExamples, which is
## an auxiliary R pacakge for iLaplace. To download it (be sure you have
## the devtools package) run from R
## devtools::install_github(erlisR/iLaplaceExamples)
## or download the source at \url{https://github.com/erlisR/iLaplaceExamples}.
```

```
## End(Not run)
```

iLap2d	*Improved Laplace approximation for bivariate integrals of unimodal functions*

Description

This function is similar to `iLap` except that it handles only bivariate integrals of user-written unimodal functions.

Usage

```
iLap2d(fullOpt, ff, ff.gr, ff.hess, quad.data, ...)
```

Arguments

`fullOpt`	A list containing the minium (to be accesed via `fullOpt$par`), the value of the function at the minimum (to be accessed via `fullOpt$objective`) and the Hessian matrix at the minimum (to be accessed via `fullOpt$hessian`
`ff`	The minus logarithm of the integrand function (the `h` function, see `iLap` for further details).
`ff.gr`	The gradient of `ff`, having the exact same arguments as `ff`
`ff.hess`	The Hessian matrix of`ff`, having the exact same arguments as `ff`
`quad.data`	Data for the Gaussian-Herimte quadratures; see "Details"
`...`	Additional arguments to be passed to `ff`, `ff.gr` and `ff.hess`

Value

a double, the logarithm of the integral

References

Ruli E., Sartori N. & Ventura L. (2015) Improved Laplace approximation for marignal likelihoods. http://arxiv.org/abs/1502.06440

Examples

```
# The negative integrand function in log
# is the negative log-density of the multivariate
# Student-t density centred at 0 with unit scale matrix
ff <- function(x, df) {
       d <- length(x)
       S <- diag(1, d, d)
```

```
        S.inv <- solve(S)
        Q <- colSums((S.inv %*% x) * x)
        logDet <- determinant(S)$modulus
        logPDF <- (lgamma((df + d)/2) - 0.5 * (d * logb(pi * df) +
        logDet) - lgamma(df/2) - 0.5 * (df + d) * logb(1 + Q/df))
        return(-logPDF)
        }

# the gradient of ff
ff.gr <- function(x, df){
            m <- length(x)
            kr = 1 + crossprod(x,x)/df
            return((m+df)*x/(df*kr))
            }

# the Hessian matrix of ff
ff.hess <- function(x, df) {
m <- length(x)
kr <- as.double(1 + crossprod(x,x)/df)
ll <- -(df+m)*2*tcrossprod(x,x)/(df*kr)^2.0
dd = (df+m)*(kr - 2*x^2/df)/(df*kr^2.0)
diag(ll) = dd;
return(ll)
}

dgf = 5
opt <- nlminb(rep(1,2), ff, gradient = ff.gr, hessian = ff.hess, df = dgf)
opt$hessian <- ff.hess(opt$par, df = dgf);
quad.data = gaussHermiteData(50)

# The improved Laplace approximation (the truth equals 0.0)
iLap <- iLap2d(fullOpt = opt, ff = ff, ff.gr = ff.gr,
               ff.hess = ff.hess, quad.data = quad.data,
               df = dgf)
# The standard Laplace approximation (the truth equals 0.0)
Lap <- log(2*pi) - opt$objective - 0.5*determinant(opt$hessian)$mod;
```

---

iLapCW *Improved Laplace approximation without nested optimisation*

Description

This function implements the improved Laplace approximation of Ruli et al. (2015) for multivariate integrals of user-written unimodal functions. See "Details" below for more information.

Usage

```
iLapCW(fullOpt, ff, ff.gr, ff.hess, quad.data, ...)
```

Arguments

fullOpt	A list containing the minium (to be accesed via fullOpt$par), the value of the function at the minimum (to be accessed via fullOpt$objective) and the Hessian matrix at the minimum (to be accessed via fullOpt$hessian
ff	The minus logarithm of the integrand function (the h function; see "Details").
ff.gr	The gradient of ff, having the exact same arguments as ff.
ff.hess	The Hessian matrix offf, having the exact same arguments as ff.
quad.data	Data for the Gaussian-Herimte quadratures; see "Details"
...	Additional arguments to be passed to ff, ff.gr and ff.hess

Details

iLapCW approximates integrals of the type

$$I = \int_{x \in \mathcal{R}^d} \exp\{-h(x)\} \, dx$$

where $-h(\cdot)$ is a concave and unimodal function, with x being d dimensional real vector ($d > 1$). The approximation of I is obtained as in iLap but with nested optimisations replaced by the approximations prposed by Cox & Wermuth (1990).

Value

A double, the logarithm of the integral

References

Ruli E., Sartori N. & Ventura L. (2015) Improved Laplace approximation for marignal likelihoods. http://arxiv.org/abs/1502.06440

Liu, Q. and Pierce, D. A. (1994). A Note on Gauss-Hermite Quadrature. *Biometrika* **81**, 624-629.

Cox, D.R and Wermuth, W. (1990). An approximation to maximum likelihood estimates in reduced models. *Biometrika* **77**, 747-761

Examples

```
# The negative integrand function in log
# is the negative log-density of the multivariate
# Student-t density centred at 0 with unit scale matrix
ff <- function(x, df) {
       d <- length(x)
       S <- diag(1, d, d)
       S.inv <- solve(S)
       Q <- colSums((S.inv %*% x) * x)
       logDet <- determinant(S)$modulus
       logPDF <- (lgamma((df + d)/2) - 0.5 * (d * logb(pi * df) +
       logDet) - lgamma(df/2) - 0.5 * (df + d) * logb(1 + Q/df))
       return(-logPDF)
```

```
       }

# the gradient of ff
ff.gr <- function(x, df){
            m <- length(x)
            kr = 1 + crossprod(x,x)/df
            return((m+df)*x/(df*kr))
            }

# the Hessian matrix of ff
ff.hess <- function(x, df) {
m <- length(x)
kr <- as.double(1 + crossprod(x,x)/df)
ll <- -(df+m)*2*tcrossprod(x,x)/(df*kr)^2.0
dd = (df+m)*(kr - 2*x^2/df)/(df*kr^2.0)
diag(ll) = dd;
return(ll)
}

df = 5
dims <- 5:8
normConts <- sapply(dims, function(mydim) {
opt <- nlminb(rep(1,mydim), ff, gradient = ff.gr, hessian = ff.hess, df = df)
opt$hessian <- ff.hess(opt$par, df = df);
quad.data = gaussHermiteData(50)
iLap <- iLapCW(opt, ff, ff.gr, ff.hess, quad.data = quad.data, df = df);
Lap <- mydim*log(2*pi)/2 - opt$objective - 0.5*determinant(opt$hessian)$mod;
return(c(iLap = iLap, Lap = Lap))
})
# plot the results
## Not run:
plot(dims, normConts[1,], pch="*", cex = 1.6,
 ylim = c(-5, 0)) #improved Laplace
lines(dims, normConts[2,], type = "p", pch = "+") #standard Laplace
abline(h = 0) # the true value

## End(Not run)

## Not run:
## See also the examples provided in the pacakge iLaplaceExamples, which is
## an auxiliary R pacakge for iLaplace. To download it (be sure you have
## the devtools package) run from R
## devtools::install_github(erlisR/iLaplaceExamples)
## or download the source at \url{https://github.com/erlisR/iLaplaceExamples}.


## End(Not run)
```

iLapCW_par — *Improved Laplace approximation without nested optimisation in parallel*

Description

Does the same as `iLapCW` but in parallel.

Usage

```
iLapCW_par(fullOpt, ff, ff.gr, ff.hess, quad.data,
     control = list(n.cores = 1), ...)
```

Arguments

`fullOpt`	A list containing the minium (to be accesed via `fullOpt$par`), the value of the function at the minimum (to be accessed via `fullOpt$objective`) and the Hessian matrix at the minimum (to be accessed via `fullOpt$hessian`
`ff`	The minus logarithm of the integrand function (the `h` function; see "Details").
`ff.gr`	The gradient of `ff`, having the exact same arguments as `ff`.
`ff.hess`	The Hessian matrix of`ff`, having the exact same arguments as `ff`.
`quad.data`	Data for the Gaussian-Herimte quadratures; see "Details".
`control`	A named list of control parameters with elements `n.cores` which sets the number of cores to be used for the parallel computiations. See "Details" for more information.
`...`	Additional arguments to be passed to `ff`, `ff.gr` and `ff.hess`

Details

See `iLapCW`.

Value

A double, the logarithm of the integral

References

Ruli E., Sartori N. & Ventura L. (2015) Improved Laplace approximation for marignal likelihoods. http://arxiv.org/abs/1502.06440

Liu, Q. and Pierce, D. A. (1994). A Note on Gauss-Hermite Quadrature. *Biometrika* **81** 624-629.

Cox, D.R and Wermuth, W. (1990). An approximation to maximum likelihood estimates in reduced models. *Biometrika* **77**, 747-761

Examples

```
# The negative integrand function in log
# is the negative log-density of the multivariate
# Student-t density centred at 0 with unit scale matrix
ff <- function(x, df) {
       d <- length(x)
       S <- diag(1, d, d)
       S.inv <- solve(S)
       Q <- colSums((S.inv %*% x) * x)
       logDet <- determinant(S)$modulus
       logPDF <- (lgamma((df + d)/2) - 0.5 * (d * logb(pi * df) +
       logDet) - lgamma(df/2) - 0.5 * (df + d) * logb(1 + Q/df))
       return(-logPDF)
       }

# the gradient of ff
ff.gr <- function(x, df){
            m <- length(x)
            kr = 1 + crossprod(x,x)/df
            return((m+df)*x/(df*kr))
            }

# the Hessian matrix of ff
ff.hess <- function(x, df) {
m <- length(x)
kr <- as.double(1 + crossprod(x,x)/df)
ll <- -(df+m)*2*tcrossprod(x,x)/(df*kr)^2.0
dd = (df+m)*(kr - 2*x^2/df)/(df*kr^2.0)
diag(ll) = dd;
return(ll)
}

df = 5
dims <- 5:8
normConts <- sapply(dims, function(mydim) {
opt <- nlminb(rep(1,mydim), ff, gradient = ff.gr, hessian = ff.hess, df = df)
opt$hessian <- ff.hess(opt$par, df = df);
quad.data = gaussHermiteData(50)
iLap <- iLap(opt, ff, ff.gr, ff.hess, quad.data = quad.data, df = df);
Lap <- mydim*log(2*pi)/2 - opt$objective - 0.5*determinant(opt$hessian)$mod;
return(c(iLap = iLap, Lap = Lap))
})
# plot the results
## Not run:
plot(dims, normConts[1,], pch="*", cex = 1.6,
 ylim = c(-5, 0)) #improved Laplace
lines(dims, normConts[2,], type = "p", pch = "+") #standard Laplace
abline(h = 0) # the true value

## End(Not run)
```

```
## Not run:
## See also the examples provided in the pacakge iLaplaceExamples, which is
## an auxiliary R pacakge for iLaplace. To download it (be sure you have
## the devtools package) run from R
## devtools::install_github(erlisR/iLaplaceExamples)
## or download the source at \url{https://github.com/erlisR/iLaplaceExamples}.


## End(Not run)
```

---

iLap_par *Improved Laplace approximation in parallel*

Description

Does the same as `iLap` but in parallel

Usage

```
iLap_par(fullOpt, ff, ff.gr, ff.hess, quad.data,
     control = list(n.cores = 1), ...)
```

Arguments

`fullOpt`	A list containing the minium (to be accesed via `fullOpt$par`), the value of the function at the minimum (to be accessed via `fullOpt$objective`) and the Hessian matrix at the minimum (to be accessed via `fullOpt$hessian`
`ff`	The minus logarithm of the integrand function (the `h` function; see "Details").
`ff.gr`	The gradient of `ff`, having the exact same arguments as `ff`.
`ff.hess`	The Hessian matrix of`ff`, having the exact same arguments as `ff`.
`quad.data`	Data for the Gaussian-Herimte quadratures; see "Details"
`control`	A named list of control parameters with elements `n.cores` which sets the number of cores to be used for the parallel computiations. See "Details" for more information.
`...`	Additional arguments to be passed to `ff`, `ff.gr` and `ff.hess`

Details

See `iLap`.

Value

A double, the logarithm of the integral

References

Ruli E., Sartori N. & Ventura L. (2015) Improved Laplace approximation for marignal likelihoods. http://arxiv.org/abs/1502.06440

Liu, Q. and Pierce, D. A. (1994). A Note on Gauss-Hermite Quadrature. *Biometrika* **81**, 624-629.

Cox, D.R and Wermuth, W. (1990). An approximation to maximum likelihood estimates in reduced models. *Biometrika* **77**, 747-761

Examples

```
# The negative integrand function in log
# is the negative log-density of the multivariate
# Student-t density centred at 0 with unit scale matrix
ff <- function(x, df) {
       d <- length(x)
       S <- diag(1, d, d)
       S.inv <- solve(S)
       Q <- colSums((S.inv %*% x) * x)
       logDet <- determinant(S)$modulus
       logPDF <- (lgamma((df + d)/2) - 0.5 * (d * logb(pi * df) +
       logDet) - lgamma(df/2) - 0.5 * (df + d) * logb(1 + Q/df))
       return(-logPDF)
       }

# the gradient of ff
ff.gr <- function(x, df){
            m <- length(x)
            kr = 1 + crossprod(x,x)/df
            return((m+df)*x/(df*kr))
            }

# the Hessian matrix of ff
ff.hess <- function(x, df) {
m <- length(x)
kr <- as.double(1 + crossprod(x,x)/df)
ll <- -(df+m)*2*tcrossprod(x,x)/(df*kr)^2.0
dd = (df+m)*(kr - 2*x^2/df)/(df*kr^2.0)
diag(ll) = dd;
return(ll)
}

df = 5
dims <- 5:8
normConts <- sapply(dims, function(mydim) {
opt <- nlminb(rep(1,mydim), ff, gradient = ff.gr, hessian = ff.hess, df = df)
opt$hessian <- ff.hess(opt$par, df = df);
quad.data = gaussHermiteData(50)
iLap <- iLap_par(opt, ff, ff.gr, ff.hess, quad.data = quad.data, df = df);
Lap <- mydim*log(2*pi)/2 - opt$objective - 0.5*determinant(opt$hessian)$mod;
return(c(iLap = iLap, Lap = Lap))
})
```

```
# plot the results
## Not run:
plot(dims, normConts[1,], pch="*", cex = 1.6,
 ylim = c(-5, 0)) #improved Laplace
lines(dims, normConts[2,], type = "p", pch = "+") #standard Laplace
abline(h = 0) # the true value

## End(Not run)

## Not run:
## See also the examples provided in the pacakge iLaplaceExamples, which is
## an auxiliary R pacakge for iLaplace. To download it (be sure you have
## the devtools package) run from R
## devtools::install_github(erlisR/iLaplaceExamples)
## or download the source at \url{https://github.com/erlisR/iLaplaceExamples}.


## End(Not run)
```

Index