Package ‘LMN’

October 12, 2022

Type  Package
Title  Inference for Linear Models with Nuisance Parameters
Version  1.1.3
Date  2022-08-11
Description  Efficient Frequentist profiling and Bayesian marginalization of parameters for which the conditional likelihood is that of a multivariate linear regression model. Arbitrary inter-observation error correlations are supported, with optimized calculations provided for independent-heteroskedastic and stationary dependence structures.

URL  https://github.com/mlysy/LMN

BugReports  https://github.com/mlysy/LMN/issues
License  GPL-3
Imports  Rcpp (>= 0.12.4.4), SuperGauss, stats
LinkingTo  Rcpp, RcppEigen
Encoding  UTF-8
RoxygenNote  7.2.1
Suggests  testthat, numDeriv, mniw, knitr, rmarkdown, bookdown, kableExtra
VignetteBuilder  knitr
NeedsCompilation  yes
Author  Martin Lysy [aut, cre], Bryan Yates [aut]
Maintainer  Martin Lysy <mlysy@uwaterloo.ca>
Repository  CRAN
Date/Publication  2022-08-22 16:20:02 UTC

R topics documented:

LMN-package .................................................. 2
list2mniw .................................................. 3
Inference for Linear Models with Nuisance Parameters.

Description

Efficient profile likelihood and marginal posteriors when nuisance parameters are those of linear regression models.

Details

Consider a model \( p(Y \mid B, \Sigma, \theta) \) of the form

\[
Y \sim \text{Matrix-Normal}(X(\theta)B, V(\theta), \Sigma),
\]

where \( Y_{n \times q} \) is the response matrix, \( X(\theta)_{n \times p} \) is a covariate matrix which depends on \( \theta \), \( B_{p \times q} \) is the coefficient matrix, \( V(\theta)_{n \times n} \) and \( \Sigma_{q \times q} \) are the between-row and between-column variance matrices, and (suppressing the dependence on \( \theta \)) the Matrix-Normal distribution is defined by the multivariate normal distribution \( \text{vec}(Y) \sim N(\text{vec}(XB), \Sigma \otimes V) \), where \( \text{vec}(Y) \) is a vector of length \( nq \) stacking the columns of \( Y \), and \( \Sigma \otimes V \) is the Kronecker product.

The model above is referred to as a Linear Model with Nuisance parameters (LMN) \( (B, \Sigma) \), with parameters of interest \( \theta \). That is, the LMN package provides tools to efficiently conduct inference on \( \theta \) first, and subsequently on \( (B, \Sigma) \), by Frequentist profile likelihood or Bayesian marginal inference with a Matrix-Normal Inverse-Wishart (MNIW) conjugate prior on \( (B, \Sigma) \).

Author(s)

Maintainer: Martin Lysy <mlysy@uwaterloo.ca>

Authors:

- Bryan Yates

See Also

Useful links:

- [https://github.com/mlysy/LMN](https://github.com/mlysy/LMN)
list2mniw

Convert list of MNIW parameter lists to vectorized format.

Description
Converts a list of return values of multiple calls to \texttt{lmn\_prior()} or \texttt{lmn\_post()} to a single list of MNIW parameters, which can then serve as vectorized arguments to the functions in \texttt{mniw}.

Usage
\begin{verbatim}
list2mniw(x)
\end{verbatim}

Arguments
\begin{verbatim}
x List of n MNIW parameter lists.
\end{verbatim}

Value
A list with the following elements:
- \texttt{Lambda} The mean matrices as an array of size $p \times p \times n$.
- \texttt{Omega} The between-row precision matrices, as an array of size $p \times p \times n$.
- \texttt{Psi} The between-column scale matrices, as an array of size $q \times q \times n$.
- \texttt{nu} The degrees-of-freedom parameters, as a vector of length $n$.

\hrulefill

\texttt{lmn\_loglik}

Loglikelihood function for LMN models.

Description
Loglikelihood function for LMN models.

Usage
\begin{verbatim}
1mn\_loglik(Beta, Sigma, suff)
\end{verbatim}

Arguments
\begin{verbatim}
Beta A $p \times q$ matrix of regression coefficients (see \texttt{lmn\_suff()}).
Sigma A $q \times q$ matrix of error variances (see \texttt{lmn\_suff()}).
suff An object of class \texttt{lmn\_suff} (see \texttt{lmn\_suff()}).
\end{verbatim}

Value
Scalar; the value of the loglikelihood.
Examples

```r
# generate data
n <- 50
q <- 3
Y <- matrix(rnorm(n*q), n, q) # response matrix
X <- 1 # intercept covariate
V <- 0.5 # scalar variance specification
suff <- lmn_suff(Y, X = X, V = V) # sufficient statistics

# calculate loglikelihood
Beta <- matrix(rnorm(q), 1, q)
Sigma <- diag(rexp(q))
lln_loglik(Beta = Beta, Sigma = Sigma, suff = suff)
```

**lmn_marg**

Marginal log-posterior for the LMN model.

**Description**

Marginal log-posterior for the LMN model.

**Usage**

```r
lmn_marg(suff, prior, post)
```

**Arguments**

- `suff`: An object of class `lmn_suff` (see `lmn_suff()`).
- `prior`: A list with elements `Lambda`, `Omega`, `Psi`, `nu` corresponding to the parameters of the prior MNIW distribution. See `lmn_prior()`.
- `post`: A list with elements `Lambda`, `Omega`, `Psi`, `nu` corresponding to the parameters of the posterior MNIW distribution. See `lmn_post()`.

**Value**

The scalar value of the marginal log-posterior.

**Examples**

```r
# generate data
n <- 50
q <- 2
p <- 3
Y <- matrix(rnorm(n*q), n, q) # response matrix
X <- matrix(rnorm(n*p), n, p) # covariate matrix
V <- .5 * exp(-(1:n)/n) # Toeplitz variance specification
suff <- lmn_suff(Y = Y, X = X, V = V, Vtype = "acf") # sufficient statistics
```
# default noninformative prior \( \pi(B, \Sigma) \sim |\Sigma|^{-(q+1)/2} \)
prior <- lmn_prior(p = suff$p, q = suff$q)
post <- lmn_post(suff, prior = prior) # posterior MNIW parameters
lmn_marg(suff, prior = prior, post = post)

---

**lmn_post**

*Parameters of the posterior conditional distribution of an LMN model.*

**Description**

Calculates the parameters of the LMN model’s Matrix-Normal Inverse-Wishart (MNIW) conjugate posterior distribution (see Details).

**Usage**

```r
lmn_post(suff, prior)
```

**Arguments**

- `suff` An object of class `lmn_suff` (see `lmn_suff()`).
- `prior` A list with elements Lambda, Omega, Psi, nu as returned by `lmn_prior()`.

**Details**

The Matrix-Normal Inverse-Wishart (MNIW) distribution \((B, \Sigma) \sim \text{MNIW}(\Lambda, \Omega, \Psi, \nu)\) on random matrices \(X_{p \times q}\) and symmetric positive-definite \(\Sigma_{q \times q}\) is defined as

\[
\Sigma \sim \text{Inverse-Wishart}(\Psi, \nu) \\
B \mid \Sigma \sim \text{Matrix-Normal}(\Lambda, \Omega^{-1}, \Sigma),
\]

where the Matrix-Normal distribution is defined in `lmn_suff()`.

The posterior MNIW distribution is required to be a proper distribution, but the prior is not. For example, `prior = NULL` corresponds to the noninformative prior

\[
\pi(B, \Sigma) \sim |\Sigma|^{-(q+1)/2}.
\]

**Value**

A list with elements named as in `prior` specifying the parameters of the posterior MNIW distribution. Elements Omega = NA and nu = NA specify that parameters Beta = 0 and Sigma = diag(q), respectively, are known and not to be estimated.
Examples

```r
# generate data
n <- 50
q <- 2
p <- 3
Y <- matrix(rnorm(n*q),n,q) # response matrix
X <- matrix(rnorm(n*p),n,p) # covariate matrix
V <- .5 * exp(-(1:n)/n) # Toeplitz variance specification

suff <- lmn_suff(Y = Y, X = X, V = V, Vtype = "acf") # sufficient statistics
```

### lmn_prior

Conjugate prior specification for LMN models.

#### Description

The conjugate prior for LMN models is the Matrix-Normal Inverse-Wishart (MNIW) distribution. This convenience function converts a partial MNIW prior specification into a full one.

#### Usage

```r
lmn_prior(p, q, Lambda, Omega, Psi, nu)
```

#### Arguments

- **p**: Integer specifying row dimension of Beta. \( p = 0 \) corresponds to no Beta in the model, i.e., \( X = 0 \) in `lmn_suff()`.
- **q**: Integer specifying the dimension of Sigma.
- **Lambda**: Mean parameter for Beta. Either:
  - A \( p \times q \) matrix.
  - A scalar, in which case \( \text{Lambda} = \text{matrix}(\text{Lambda}, p, q) \).
  - Missing, in which case \( \text{Lambda} = \text{matrix}(0, p, q) \).
- **Omega**: Row-wise precision parameter for Beta. Either:
  - A \( p \times p \) matrix.
  - A scalar, in which case \( \text{Omega} = \text{diag}(\text{rep}(\text{Omega}, p)) \).
  - Missing, in which case \( \text{Omega} = \text{matrix}(0, p, p) \).
  - NA, which signifies that Beta is known, in which case the prior is purely Inverse-Wishart on Sigma (see **Details**).
- **Psi**: Scale parameter for Sigma. Either:
  - A \( q \times q \) matrix.
  - A scalar, in which case \( \text{Psi} = \text{diag}(\text{rep}(\text{Psi}, q)) \).
  - Missing, in which case \( \text{Psi} = \text{matrix}(0, q, q) \).
- **nu**: Degrees-of-freedom parameter for Sigma. Either a scalar, missing (defaults to \( \text{nu} = 0 \)), or NA, which signifies that \( \text{Sigma} = \text{diag}(q) \) is known, in which case the prior is purely Matrix-Normal on Beta (see **Details**).
The Matrix-Normal Inverse-Wishart (MNIW) distribution \((B, \Sigma) \sim \text{MNIW}(\Lambda, \Omega, \Psi, \nu)\) on random matrices \(X_{p \times q}\) and symmetric positive-definite \(\Sigma_{q \times q}\) is defined as

\[
\begin{align*}
\Sigma & \sim \text{Inverse-Wishart}(\Psi, \nu) \\
B | \Sigma & \sim \text{Matrix-Normal}(\Lambda, \Omega^{-1}, \Sigma),
\end{align*}
\]

where the Matrix-Normal distribution is defined in \texttt{lmn_suff()}. 

A list with elements Lambda, Omega, Psi, nu with the proper dimensions specified above, except possibly Omega = NA or nu = NA (see Details).

Examples

```r
# problem dimensions
p <- 2
q <- 4

# default noninformative prior \(p(Beta, Sigma) \sim |Sigma|^{-(q+1)/2}\)
lmn_prior(p, q)

# \(p(Sigma) \sim |Sigma|^{-(q+1)/2}\)
# Beta | Sigma \sim \text{Matrix-Normal}(0, I, Sigma)
lmn_prior(p, q, Lambda = 0, Omega = 1)

# Sigma = diag(q)
# Beta \sim \text{Matrix-Normal}(0, I, Sigma = diag(q))
lmn_prior(p, q, Lambda = 0, Omega = 1, nu = NA)
```

\hlm

\text{lmn_prof} \quad \text{Profile loglikelihood for the LMN model.}

\hlm

\text{Description}

Calculate the loglikelihood of the LMN model defined in \texttt{lmn_suff()} at the MLE Beta = Bhat and Sigma = Sigma.hat.

\text{Usage}

\texttt{lmn_prof(suff, noSigma = FALSE)}

\text{Arguments}

- **suff**: An object of class \texttt{lmn_suff} (see \texttt{lmn_suff()}).
- **noSigma**: Logical. If \texttt{TRUE} assumes that Sigma = diag(ncol(Y)) is known and therefore not estimated.
Value

Scalar; the calculated value of the profile loglikelihood.

Examples

```r
# generate data
n <- 50
q <- 2
Y <- matrix(rnorm(n*q),n,q) # response matrix
X <- matrix(1,n,1) # covariate matrix
V <- exp(-(1:n)/n) # diagonal variance specification
suff <- lmn_suff(Y, X = X, V = V, Vtype = "diag") # sufficient statistics

# profile loglikelihood
lmn_prof(suff)

# check that it's the same as loglikelihood at MLE
lmn_loglik(Beta = suff$Bhat, Sigma = suff$S/suff$n, suff = suff)
```

**lmn_suff**

*Calculate the sufficient statistics of an LMN model.*

Description

Calculate the sufficient statistics of an LMN model.

Usage

```r
lmn_suff(Y, X, V, Vtype, npred = 0)
```

Arguments

- **Y**
  - An \( n \times q \) matrix of responses.

- **X**
  - An \( N \times p \) matrix of covariates, where \( N = n + npred \) (see Details). May also be passed as:
    - A scalar, in which case the one-column covariate matrix is \( X = X \times \text{matrix}(1, N, 1) \). \( X = 0 \), in which case the mean of \( Y \) is known to be zero, i.e., no regression coefficients are estimated.

- **V, Vtype**
  - The between-observation variance specification. Currently the following options are supported:
    - \( Vtype = "full" \): \( V \) is an \( N \times N \) symmetric positive-definite matrix.
    - \( Vtype = "diag" \): \( V \) is a vector of length \( N \) such that \( V = \text{diag}(V) \).
    - \( Vtype = "scalar" \): \( V \) is a scalar such that \( V = V \times \text{diag}(N) \).
    - \( Vtype = "acf" \): \( V \) is either a vector of length \( N \) or an object of class `SuperGauss::Toeplitz`, such that \( V = \text{toeplitz}(V) \).
For \( V \) specified as a matrix or scalar, \( V \text{type} \) is deduced automatically and need not be specified.

**npred**

A nonnegative integer. If positive, calculates sufficient statistics to make predictions for new responses. See **Details**.

**Details**

The multi-response normal linear regression model is defined as

\[
Y \sim \text{Matrix-Normal}(XB, V, \Sigma),
\]

where \( Y_{n \times q} \) is the response matrix, \( X_{n \times p} \) is the covariate matrix, \( B_{p \times q} \) is the coefficient matrix, \( V_{n \times n} \) and \( \Sigma_{q \times q} \) are the between-row and between-column variance matrices, and the Matrix-Normal distribution is defined by the multivariate normal distribution \( \text{vec}(Y) \sim \mathcal{N}(\text{vec}(XB), \Sigma \otimes V) \), where \( \text{vec}(Y) \) is a vector of length \( nq \) stacking the columns of \( Y \), and \( \Sigma \otimes V \) is the Kronecker product.

The function `lmn_suff()` returns everything needed to efficiently calculate the likelihood function

\[
L(B, \Sigma \mid Y, X, V) = p(Y \mid X, V, B, \Sigma).
\]

When \( npred > 0 \), define the variables \( Y_{\text{star}} = \text{rbind}(Y, y) \), \( X_{\text{star}} = \text{rbind}(X, x) \), and \( V_{\text{star}} = \text{rbind}(\text{cbind}(V, w), \text{cbind}(t(w), v)) \). Then `lmn_suff()` calculates summary statistics required to estimate the conditional distribution

\[
p(y \mid Y, X_{\text{star}}, V_{\text{star}}, B, \Sigma).
\]

The inputs to `lmn_suff()` in this case are \( Y = Y, X = X_{\text{star}}, \) and \( V = V_{\text{star}} \).

**Value**

An S3 object of type `lmn_suff`, consisting of a list with elements:

- `Bhat` The \( p \times q \) matrix \( \hat{B} = (X'V^{-1}X)^{-1}X'V^{-1}Y \).
- `T` The \( p \times p \) matrix \( T = X'V^{-1}X \).
- `S` The \( q \times q \) matrix \( S = (Y - X\hat{B})'V^{-1}(Y - X\hat{B}) \).
- `ldV` The scalar log-determinant of \( V \).
- `n`, `p`, `q` The problem dimensions, namely \( n = nrow(Y), p = nrow(Beta) \) (or \( p = 0 \) if \( X = 0 \)), and \( q = ncol(Y) \).

In addition, when \( npred > 0 \) and with \( x, w, \) and \( v \) defined in **Details**:

- `Ap` The \( npred \times q \) matrix \( A_p = w'V^{-1}Y \).
- `Xp` The \( npred \times p \) matrix \( X_p = x - wV^{-1}X \).
- `Vp` The scalar \( V_p = v - wV^{-1}w \).
Examples

# Data
n <- 50
q <- 3
Y <- matrix(rnorm(n*q), n, q)

# No intercept, diagonal V input
X <- 0
V <- exp(-(1:n)/n)
lmn_suff(Y, X = X, V = V, Vtype = "diag")

# X = (scaled) Intercept, scalar V input (no need to specify Vtype)
X <- 2
V <- .5
lmn_suff(Y, X = X, V = V)

# X = dense matrix, Toeplitz variance matrix
p <- 2
X <- matrix(rnorm(n*p), n, p)
Tz <- SuperGauss::Toeplitz$new(acf = 0.5*exp(-seq(1:n)/n))
lmn_suff(Y, X = X, V = Tz, Vtype = "acf")
Index

list2mnw, 3
LMN (LMN-package), 2
LMN-package, 2
lmn_loglik, 3
lmn_marg, 4
lmn_post, 5
lmn_post(), 3, 4
lmn_prior, 6
lmn_prior(), 3–5
lmn_prof, 7
lmn_suff, 8
lmn_suff(), 3–7

SuperGauss::Toeplitz, 8