Package ‘mvMISE’

April 16, 2018

Title A General Framework of Multivariate Mixed-Effects Selection Models

Version 1.0

Date 2017-07-20

Author Jiebiao Wang and Lin S. Chen

Maintainer Jiebiao Wang <randel.wang@gmail.com>

Description Offers a general framework of multivariate mixed-effects models for the joint analysis of multiple correlated outcomes with clustered data structures and potential missingness. The missingness of outcome values may depend on the values themselves (missing not at random and non-ignorable), or may depend on only the covariates (missing at random and ignorable), or both. This package provides functions for two models: 1) the mvMISE_b model that allows correlated outcome-specific random intercepts with a factor-analytic structure, and 2) the mvMISE_e model that allows the correlated outcome-specific error terms with a graphical lasso penalty on the error precision matrix. Both functions are motivated by the multivariate data analysis on data with clustered structures from labelling-based quantitative proteomic studies. These models and functions can also be applied to univariate and multivariate analyses of clustered data with balanced or unbalanced design and no missingness.

License GPL

Depends lme4, MASS

URL https://github.com/randel/mvMISE

BugReports https://github.com/randel/mvMISE/issues

RoxygenNote 6.0.1

R topics documented:

mvMISE_b *A multivariate mixed-effects selection model with correlated outcome-specific random intercepts*

Description

This function fits a multivariate mixed-effects selection model with correlated outcome-specific random intercepts allowing potential ignorable or non-ignorable missing values in the outcome. Here an outcome refers to a response variable, for example, a genomic feature. The proposed model and function jointly analyze multiple outcomes/features.

Usage

```
mvMISE_b(Y, X, id, maxIter = 100, tol = 0.001, verbose = FALSE, cov_miss = NULL,
    miss_y = TRUE, sigma_diff = FALSE)
```

Arguments

`Y` an outcome matrix. Each row is a sample, and each column is an outcome variable, with potential missing values (NAs).

`X` a covariate matrix. Each row is a sample, and each column is a covariate. The covariates can be common among all of the outcomes (e.g., age, gender) or outcome-specific. If a covariate is specific for the k-th outcome, one may set all the values corresponding to the other outcomes to be zero. If X is common across outcomes, the row number of X equals the row number of Y. Otherwise, if X is outcome-specific, the row number of X equals the number of elements in Y, i.e., outcome-specific X is stacked across outcomes within each cluster. See the Examples for demonstration.

`id` a vector of cluster/batch index, matching with the rows of Y, and X if it is not outcome specific.

`maxIter` the maximum number of iterations for the EM algorithm.

`tol` the tolerance level for the relative change in the observed-data log-likelihood function.

`verbose` logical. If TRUE, the iteration history of each step of the EM algorithm will be printed. The default is FALSE.

`cov_miss` the covariate that can be used in the missing-data model. If it is NULL, the missingness is assumed to be independent of the covariates. Check the Details for the missing-data model. If it is specified and the covariate is not outcome specific, its length equals the length of id. If it is outcome specific, the outcome-specific covariate is stacked across outcomes within each cluster.

`miss_y` logical. If TRUE, the missingness depends on the outcome Y (see the Details). The default is TRUE. This outcome-dependent missing data pattern was motivated by and was observed in the mass-spectrometry-based quantitative proteomics data.

`sigma_diff` logical. If TRUE, the sample error variance of the first sample in each cluster/batch is different from that for the rest of samples within the same cluster/batch. This option is designed and used when analyzing batch-processed proteomics data with the first sample in each cluster/batch being the common reference sample. The default is FALSE.

Details

The multivariate mixed-effects selection model consists of two components, the outcome model and the missing-data model. Here the outcome model is a multivariate mixed-effects model, with correlations among multivariate outcomes modeled via correlated outcome-specific random intercepts with a factor-analytic structure

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + (\mathbf{I}_K \otimes \mathbf{1}_{n_i})\,\boldsymbol{\tau} b_i + \mathbf{e}_i,$$

where i denotes a cluster/batch, n_i is the number of samples/observations within each cluster, $\boldsymbol{\tau}$ is a $K \times 1$ vector for the outcome-specific variance components corresponding to the random effect b_i (a standard normal random variable), and K is the number of outcomes. By default, a matrix with each column as an indicator for each outcome is generated and is used as the random-effect design matrix ($\mathbf{I}_K \otimes \mathbf{1}_{n_i}$), and the model will estimate the outcome-specific random intercepts. The factor-analytic structure assumes the outcome-specific random intercepts are identically correlated and this model is often used to capture the highly structured experimental or biological correlations among naturally related outcomes. For example, the correlation among multiple phosphopeptides (i.e. phosphorylated segments) of a same protein. The model assumes that the random effects are derived from a latent variable b_i with a loading vector $\boldsymbol{\tau}$. With this model specification, only K parameters instead of $K(K+1)/2$ are needed in the estimation for the covariance matrix of random-effects, and as such that greatly facilitates the computation.

The missing-data model can be written as

$$\text{Pr}\left(r_{ik} = 1 | \mathbf{y}_{ik}\right) = \exp\left(\phi_0 + \phi_1/n_i \cdot \mathbf{1}^{'}\mathbf{y}_{ik} + \phi_2/n_i \cdot \mathbf{1}^{'}\mathbf{c}_i\right),$$

where r_{ik} is the missing indicator for the k-th outcome in the i-th cluster. If $r_{ik} = 1$, the values of the k-th outcome in the i-th cluster $\mathbf{y}_{ik}$ are missing altogether. The estimation is implemented via an EM algorithm. Parameters in the missing-data models can be specified via the arguments miss_y and cov_miss. If miss_y = TURE, the missingness depends on the outcome values. If cov_miss is specified, the missingness can (additionally) depend on the specified covariate (cov_miss).

The model also works for fully observed data if miss_y = FALSE and cov_miss = NULL. It would also work for a univariate outcome with potential missing values, if the outcome Y is a matrix with one column.

Value

A list containing

`beta` the estimated fixed-effects.

`var` the variance-covariance matrix of the estimated fixed effects. With the fixed effects and their covariance matrix estimates, one can obtain the Wald-statistics for testing fixed-effects beta/sqrt(diag(var)).

`pval` the parametric p-values for testing non-zero fixed-effects. It is obtained as the two-sided p-value based on the Wald statistics of beta/sqrt(diag(var)).

`sigma2` the estimated sample error variance(s). If sigma_diff is TRUE, it returns a vector of two elements, the variances for the first sample and for the rest of samples within each cluster.

`tau` the estimated variance components for the outcome-specific factor-analytic random-effects.

`phi` the estimated parameters for the missing-data mechanism. Check the Details for the missing-data model. A zero estimate implies that the parameter is ignored via the specification of miss_y and/or cov_miss.

`loglikelihood` the observed-data log-likelihood values.

`iter` the number of iterations for the EM algorithm when reaching the convergence.

References

Jiebiao Wang, Pei Wang, Donald Hedeker, and Lin S. Chen. A multivariate mixed-effects selection model framework for labelling-based proteomics data with non-ignorable missingness. (In preparation).

Examples

```
data(sim_dat)

# Covariates X common across outcomes with common coefficients

fit0 = mvMISE_b(Y = sim_dat$Y, X = sim_dat$X, id = sim_dat$id)

## Not run:

# In the example below, we showed how to estimate outcome-specific
# coefficients for a common covariate. The second column of
# sim_dat$X matrix is a common covariate. But it has different
# effects/coefficients on different outcomes.

nY = ncol(sim_dat$Y)
# stack X across outcomes
X_mat = sim_dat$X[rep(1:nrow(sim_dat$X), nY), ]
# Y_ind is the indicator matrix corresponding to different outcomes
Y_ind = kronecker(diag(nY), rep(1, nrow(sim_dat$Y)))
# generate outcome-specific covariates
cidx = 2  # the index for the covariate with outcome-specific coefficient
X_mat = cbind(1, X_mat[, cidx] * Y_ind)

# X_mat is a matrix of 460 (92*5) by 6, the first column is
# intercept and the next 5 columns are covariate for each outcome

fit1 = mvMISE_b(Y = sim_dat$Y, X = X_mat, id = sim_dat$id)


# A covariate only specific to the first outcome

X_mat1 = X_mat[, 1:2]

fit2 = mvMISE_b(Y = sim_dat$Y, X = X_mat1, id = sim_dat$id)


## An example that allows missingness depending on both a covariate
## and the outcome

fit3 = mvMISE_e(Y = sim_dat$Y, X = sim_dat$X, id = sim_dat$id,
    cov_miss = sim_dat$X[, 2])


## End(Not run)
```

mvMISE_e *A multivariate mixed-effects selection model with correlated outcome-specific error terms*

Description

This function fits a multivariate mixed-effects selection model with correlated outcome-specific error terms and potential missing values in the outcome. Here an outcome refers to a response variable, for example, a genomic feature. The proposed model and function jointly analyze multiple outcomes/features. For high-dimensional outcomes, the model can regularize the estimation by shrinking the error precision matrix with a graphical lasso penalty. Given the introduction of the penalty and the choice of tuning parameter often being data-dependant, we recommend using permutation to calculate p-values for testing with the mvMISE_e model. Please see mvMISE_e_perm for calculating the permutation-based p-values.

Usage

```
mvMISE_e(Y, X, id, Zidx = 1, maxIter = 100, tol = 0.001, lambda = NULL, admm = TRUE,
    verbose = FALSE, cov_miss = NULL, miss_y = NULL, sigma_diff = FALSE)
```

Arguments

`Y` an outcome matrix. Each row is a sample, and each column is an outcome variable, with potential missing values (NAs).

`X` a covariate matrix. Each row is a sample, and each column is a covariate. The covariates can be common among all of the outcomes (e.g., age, gender) or outcome-specific. If a covariate is specific for the k-th outcome, one may set all the values corresponding to the other outcomes to be zero. If X is common across outcomes, the row number of X equals the row number of Y. Otherwise if X is outcome-specific, the row number of X equals the number of elements in Y, i.e., outcome-specific X is stacked across outcomes. See the Examples for demonstration.

`id` a vector for cluster/batch index, matching with the rows of Y, and X if it is not outcome specific.

`Zidx` the column indices of matrix X used as the design matrix of random effects. The default is 1, i.e., a random intercept is included if the first column of X is a vector of 1s. If Zidx=c(1,2), then the model would estimate the random intercept and the random effects of the 2nd column in the covariate matrix X. The random-effects in this model are assumed to be independent.

`maxIter` the maximum number of iterations for the EM algorithm.

`tol` the tolerance level for the relative change in the observed-data log-likelihood function.

`lambda` the tuning parameter for the graphical lasso penalty of the error precision matrix. It can be selected by AIC (an output). The default is sqrt(log(ncol(Y))/nrow(Y)).

`admm` logical. If TRUE (the default), we impose a L1 graphical lasso penalty on the error precision (inverse of covariance) matrix, and the alternating direction method of multipliers (ADMM) is used to estimate the error precision and the error covariance matrix. If FALSE, no penalty is used to estimate the unstructured error covariance matrix, and that is only applicable to low-dimensional multivariate outcomes. For an univariate outcome, it should be set as FALSE.

`verbose` logical. If TRUE, the iteration history of each step of the EM algorithm will be printed. The default is FALSE.

`cov_miss` the covariate that can be used in the missing-data model. If it is NULL, the missingness is assumed to be independent of the covariates. Check the Details for the missing-data model. If it is specified and the covariate is not outcome specific, its length equals the length of id. If it is outcome specific, the outcome-specific covariate is stacked across outcomes within each cluster.

`miss_y`	logical. If TRUE, the missingness depends on the outcome Y (see the Details). The default is TRUE if the average missing rate is greater than 5%, otherwise is FALSE. This outcome-dependent missing data pattern was motivated by and was observed in the mass-spectrometry-based quantitative proteomics data.
`sigma_diff`	logical. If TRUE, the sample error variance of the first sample is different from that for the rest of samples within each cluster. This option is designed and used when analyzing batch-processed proteomics data with the first sample in each cluster/batch being the common reference sample. The default is FALSE.

Details

The multivariate mixed-effects selection model consists of two components, the outcome model and the missing-data model. Here the outcome model is a multivariate mixed-effects model. The correlations among multivariate outcomes are modeled via outcome-specific error terms with an unstructured covariance matrix. For the i-th cluster, the outcome matrix $\mathbf{Y}_i$ is a matrix of n_i samples (rows) and K outcomes (columns). Let $\mathbf{y}_i = \mathrm{vec}\left(\mathbf{Y}_i\right)$. The outcome vector $\mathbf{y}_i$ can be modelled as

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \mathbf{b}_i + \mathbf{e}_i,$$

where the random effects ($\mathbf{b}_i$) follow a normal distribution $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$; and the error term $\mathbf{e}_i = \mathrm{vec}\left(\mathbf{E}_i\right) \sim N(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{S}_i)$. The matrix $\mathbf{S}_i$ is an $n_i \times n_i$ diagonal matrix with diagonal elements corresponding to the error variances of the n_i samples within the i-th cluster. The variances for the first and other samples can be different if sigma_diff = TRUE. The matrix $\mathbf{\Sigma}$ captures the error (or unexplained) covariances among the K outcomes. For high-dimensional outcomes, if admm = TRUE (the default), the off-diagonal elements of the inverse of $\mathbf{\Sigma}$ will be shrinked by a graphical lasso penalty and the alternating direction method of multipliers (ADMM) is used to estimate $\mathbf{\Sigma}$. If admm = FALSE, no penalty is used to estimate the unstructured error covariance matrix, and that is only applicable to low-dimensional multivariate outcomes.

The missing-data model can be written as

$$\Pr\left(r_{ik} = 1 | \mathbf{y}_{ik}\right) = \exp\left(\phi_0 + \phi_1/n_i \cdot \mathbf{1}^{'} \mathbf{y}_{ik} + \phi_2/n_i \cdot \mathbf{1}^{'} \mathbf{c}_i\right),$$

where r_{ik} is the missing indicator for the k-th outcome in the i-th cluster. If missing $r_{ik} = 1$, the k-th outcome in the i-th cluster $\mathbf{y}_{ik}$ is missing altogether. The estimation is implemented within an EM algorithm framework. Parameters in the missing-data models can be specified via the arguments miss_y and cov_miss. If miss_y = TURE, the missingness depends on the outcome values. If cov_miss is specified, the missingness can (additionally) depend on the specified covariates (cov_miss).

The model also works for fully observed data if miss_y = FALSE and cov_miss = NULL. It would also work for an univariate outcome with potential missing values, if the outcome Y is a matrix with one column.

Value

A list containing

`beta`	the estimated fixed-effects.
`stat`	the parametric Wald statistics for testing non-zero fixed-effects. It is used in permutation tests.
`Sigma`	the estimated error covariance matrix for the outcomes.

`sigma2` the estimated sample error variance(s). If sigma_diff is TRUE, it returns a vector of two elements, the variances for the first sample and the rest of samples within each cluster.

`D` the estimated covariance matrix for the random-effects.

`phi` the estimated parameters for the missing-data mechanism. Check the Details for the missing-data model. A zero value implies that parameter is ignored via the specification of miss_y and cov_miss.

`loglikelihood` the observed-data log-likelihood values.

`iter` the number of iterations for the EM algorithm when reaching the convergence.

`AIC` The Akaike information criterion (AIC) calculated for selecting the tuning parameter lambda of the graphical lasso penalty.

References

Jiebiao Wang, Pei Wang, Donald Hedeker, and Lin S. Chen. A multivariate mixed-effects selection model framework for labelling-based proteomics data with non-ignorable missingness. (In preparation).

Examples

```
data(sim_dat)

# Covariates X common across outcomes with common coefficients

fit0 = mvMISE_e(Y = sim_dat$Y, X = sim_dat$X, id = sim_dat$id)

## Not run:

# In the example below, we showed how to estimate outcome-specific
# coefficients for a common covariate. The second column of
# sim_dat$X matrix is a common covariate. But it has different
# effects/coefficients on different outcomes.

nY = ncol(sim_dat$Y)
# stack X across outcomes
X_mat = sim_dat$X[rep(1:nrow(sim_dat$X), nY), ]
# Y_ind is the indicator matrix corresponding to different outcomes
Y_ind = kronecker(diag(nY), rep(1, nrow(sim_dat$Y)))
# generate outcome-specific covariates
cidx = 2  # the index for the covariate with outcome-specific coefficient
X_mat = cbind(1, X_mat[, cidx] * Y_ind)

# X_mat is a matrix of 460 (92*5) by 6, the first column is
# intercept and the next 5 columns are covariate for each outcome

fit1 = mvMISE_e(Y = sim_dat$Y, X = X_mat, id = sim_dat$id)


# A covariate only specific to the first outcome

X_mat1 = X_mat[, 1:2]

fit2 = mvMISE_e(Y = sim_dat$Y, X = X_mat1, id = sim_dat$id)
```

```
## An example to allow missingness to depend on both a covariate and
## the outcome

fit3 = mvMISE_e(Y = sim_dat$Y, X = sim_dat$X, id = sim_dat$id,
    cov_miss = sim_dat$X[, 2])


## End(Not run)
```

mvMISE_e_perm *A function to obtain permutation-based p-values for fixed effects estimates in mvMISE_e*

Description

This function calls mvMISE_e multiple times by permuting the row index (observations) of the covariate matrix X. It may take a long time to permute high-dimensional outcomes, but can be run in parallel using multiple nodes.

Usage

```
mvMISE_e_perm(nperm = 100, nnodes = 2, Y, X, id, Zidx = 1, maxIter = 100, tol = 0.001,
    lambda = 0.05, cov_miss = NULL, miss_y = TRUE, sigma_diff = FALSE)
```

Arguments

nperm the number of permutations.

nnodes the number of nodes that will be used in parallel for permutations.

Y an outcome matrix. Each row is a sample, and each column is an outcome variable, with potential missing values (NAs).

X a covariate matrix. Each row is a sample, and each column is a covariate. The covariates can be common among all of the outcomes (e.g., age, gender) or outcome-specific. If a covariate is specific for the k-th outcome, one may set all the values corresponding to the other outcomes to be zero. If X is common across outcomes, the row number of X equals the row number of Y. Otherwise if X is outcome-specific, the row number of X equals the number of elements in Y, i.e., outcome-specific X is stacked across outcomes within each cluster. See the Examples for demonstration.

id a vector for cluster/batch index, matching with the rows of Y, and X if it is not outcome specific.

Zidx the column indices of matrix X used as the design matrix of random effects. The default is 1, i.e., a random intercept is included if the first column of X is a vector of 1s. If Zidx=c(1,2), then the model would estimate the random intercept and the random effects of the 2nd column in the covariate matrix X. The random-effects in this model are assumed to be independent.

maxIter the maximum number of iterations for the EM algorithm.

tol the tolerance level for the relative change in the observed-data log-likelihood function.

`lambda`	the tuning parameter for the graphical lasso penalty of the error precision matrix. It can be selected by AIC (an output).
`cov_miss`	the covariate that can be used in the missing-data model. If it is NULL, the missingness is assumed to be independent of the covariates. Check the Details for the missing-data model. If it is specified and the covariate is not outcome specific, its length equals the length of id. If it is outcome specific, the outcome-specific covariate is stacked across outcomes within each cluster.
`miss_y`	logical. If TRUE, the missingness depends on the outcome Y (see the Details). The default is TRUE. This outcome-dependent missing data pattern was motivated by and was observed in the mass-spectrometry-based quantitative proteomics data.
`sigma_diff`	logical. If TRUE, the sample error variance of the first sample is different from that for the rest of samples within each cluster. This option is designed and used when analyzing batch-processed proteomics data with the first sample in each cluster/batch being the common reference sample. The default is FALSE.

Value

The permutation based p-values for testing if fixed-effects (excluding the intercept) are zeros.

References

Jiebiao Wang, Pei Wang, Donald Hedeker, and Lin S. Chen. A multivariate mixed-effects selection model framework for labelling-based proteomics data with non-ignorable missingness. (In preparation).

Examples

```
## Not run:

data(sim_dat)

pval_perm = mvMISE_e_perm(nperm = 100, Y = sim_dat$Y, X = sim_dat$X, id = sim_dat$id)

## End(Not run)
```

`sim_dat` *A Simulated Example data*

Description

This simulated data list is for demonstration.

Value

A list containing

`Y`	a 92 by 5 outcome/feature matrix, each row is a sample, and each column is an outcome/feature variable, with potential missing values (NAs).

`X` a 92 by 2 covariate matrix, each row is a sample, and each column is a covariate with the first column being 1s for the intercept. In this example, we simulated the covariates to be common for all the outcomes and would estimate the common/averaged effects for all outcomes. If a covariate is specific for the k-th outcome, one may set all the values corresponding to the other outcomes to be zero.

`id` a vector of cluster/batch ID, matching with the rows of Y and X.

Examples

```
data(sim_dat)
```

Index