Package ‘Tsphere’

February 19, 2015

Type Package

Title Transposable Sphering for Large-Scale Inference with Correlated Data.

Version 1.0

Date 2011-11-12

Author Genevera I. Allen

Maintainer Genevera I. Allen <gallen@rice.edu>

URL http://www.stat.rice.edu/~gallen/

Description Adjusts for correlations among the rows and columns via the Transposable Sphering Algorithm when conducting large-scale inference on the rows of a data matrix.

License GPL-2

Depends glasso, rms

Repository CRAN

Date/Publication 2012-10-29 08:57:46

NeedsCompilation no

R topics documented:

ACE .......................................................... 2
covTranspose11 ............................................. 3
CVcov .......................................................... 4
meanTranspose .............................................. 5
MNloglike ...................................................... 6
TransSphere ............................................... 7

Index 10
Description

Predicts values for scattered missing elements according to the matrix-variate normal distribution. Internal use.

Usage

```r
ACE(x, sig, delt, sigi, delti, M, thr = 1e-04, maxit = 1000)
```

Arguments

- `x`: Data matrix with scattered missing values. Missing values should be denoted with "NA".
- `sig`: Row covariance matrix.
- `delt`: Column covariance matrix.
- `sigi`: Row precision matrix.
- `delti`: Column precision matrix.
- `M`: Mean matrix.
- `thr`: Convergence threshold.
- `maxit`: Maximum number of iterations.

Details

For internal use.

Value

- `x`: Matrix of predicted values.
- `iter`: Number of iterations until convergence.

Author(s)

Genevera I. Allen

References


See Also

`CVcov`
covTranspse11

Covariance Estimation.

Description

Inverse row and column covariance estimation for the L1 penalized matrix-variate normal model.

Usage

```r
covTranspse11(xc, rhor, rhoc, row = TRUE, sigi.init = NULL, delti.init = NULL, thr = 1e-04, maxit = 1000, trace = TRUE, thr.glasso = 1e-04, maxit.glasso = 1000, pen.diag = TRUE)
```

Arguments

- `xc`: Centered data matrix.
- `rhor`: Row regularization parameter.
- `rhoc`: Column regularization parameter.
- `row`: Logical. TRUE = Start with row covariance.
- `sigi.init`: Initialization for the row precision matrix. (Optional).
- `delti.init`: Initialization for the column precision matrix. (Optional).
- `thr`: Convergence threshold.
- `maxit`: Maximum number of iterations.
- `trace`: Prints matrix-variate log-likelihood for each iteration.
- `thr.glasso`: Convergence threshold for the graphical lasso.
- `maxit.glasso`: Maximum number of iterations for the graphical lasso.
- `pen.diag`: Logical. Indicates whether the diagonal should be penalized.

Details

Estimates row and column precision matrix via L1 penalized Transposable Regularized Covariance Models.

Value

- `Sigmahat`: Estimated row covariance.
- `Deltahat`: Estimated column covariance.
- `Sigmahihat`: Estimated sparse row precision matrix.
- `Deltahihat`: Estimated sparse column precision matrix.
- `loglike`: Trace of the penalized log-likelihood at each iteration.

Author(s)

Genevera I. Allen
References


See Also

TransSphere

Cross-Validation.

Description

Cross-Validation to estimate regularization parameters for sparse inverse covariance estimation.

Usage

CVcov(x, maxlamL minlamL stepsL pmiss = 0N01L do = RL trace = true)

Arguments

x Data matrix.
maxlam Maximum regularization parameter.
minlam Minimum regularization parameter.
steps Number of regularization parameters to test.
pmiss Percentage missing in each fold.
do Number of folds. Note that for medium or large size data matrices, often one fold is sufficient.
trace Logical. Output the penalized log-likelihood and MSE for each step and fold.

Details

For internal use.

Value

cvmat Matrix of cross-validation mean squared errors.
optlam Optimal value of the regularization parameter as estimated by cross-validation.
lams Values of the regularization parameters tested.

Author(s)

Genevera I. Allen
meanTranspose

References

See Also
covTranspose11, TransSphere

Description
Estimates row and column means.

Usage
meanTranspose(x, tol = 1e-06)

Arguments
x Data matrix.
tol Tolerance for iterative algorithm when data has missing values.

Details
Estimates the row and column means.

Value
x Original data matrix.
xcen Centered data matrix.
mu Column mean.
u nu Row mean.
M Mean matrix.

Author(s)
Genevera I. Allen

References
See Also
covTranspose11, TransSphere

Examples

x = matrix(rnorm(100*50),100,50)

#row and column centered data matrix
xc = meanTranspose(x)$xcen

MNloglike

Penalized Matrix-variate normal log-likelihood.

Description

Penalized Matrix-variate normal log-likelihood.

Usage

MNloglike(x, M, Sig, Delt, rhor, rhoc, qr = 2, qc = 2, Sigi = NULL, Delti = NULL)

Arguments

x Data matrix.
M Mean matrix.
Sig Row covariance matrix.
Delt Column covariance matrix.
rhor Row regularization parameter.
rhoc Column regularization parameter.
qr Row regularization norm. Either 1 or 2.
qc Column regularization norm. Either 1 or 2.
Sigi Row precision matrix. (optional).
Delti Column precision matrix. (optional).

Details

For internal use.

Value

value Log-likelihood.
Description

Applies the Transposable Sphering Algorithm to adjust for correlations among the rows and columns when conducting large-scale inference on the rows of a data matrix.

Usage

```r
TransSphere(dat, y, fdr, minlam, maxlam = NULL)
```

Arguments

- **dat**: Data matrix. Inference will be conducted on the rows and the matrix should be oriented in this manner. For example in gene expression data, the data matrix should be oriented as genes by samples.
- **y**: A vector of group labels. Labels should be denoted as a numeric 1 or 2.
- **fdr**: Desired False Discovery Rate to be controlled. Default is 0.1.
- **minlam**: Minimum regularization parameter to test via cross-validation for sparse inverse covariance estimation. Default is 0.15. Note that small values of this parameter may result in numerical instabilities. It is recommended to keep this parameter at the default.
- **maxlam**: Maximum regularization parameter to test via cross-validation for sparse inverse covariance estimation. Default is 0.25.

Details

The Transposable Sphering Algorithm adjusts for correlations among the rows and columns of a data matrix before conducting large-scale inference. Currently, this method is only written for two-sample problems. The data matrix is row and column centered and two-sample T-statistics are computed for each row. The Transposable Sphering method is applied to the top 500 rows corresponding to the largest absolute T-statistics. The matrix is decomposed into a signal matrix, corresponding to the two classes of interest, and a noise matrix. This noise matrix is sphered so that both the rows...
and columns are approximately independent. Specifically, sparse inverse covariances of the rows and columns are estimated via Transposable Regularized Covariance Models and used to whiten the noise matrix. Cross-validation is used to estimate the regularization parameters controlling the amount of sparsity. The estimated signal matrix and sphered noise matrix are then added to form the sphered data matrix that is used to conduct large-scale inference. Test statistics are adjusted using central-matching, and the Benjamini-Hochberg step-up procedure is used to control the False Discovery Rate.

**Value**

- **sig.rows**: The indices of the statistically significant rows after controlling the False Discovery Rate at the value \( fdr \).
- **t.stats**: Sphered two-sample T-statistics.
- **p.vals**: Sphered (unadjusted) p-values.
- **x.sphered**: The sphered data matrix. Note that only the top 500 rows are used in the algorithm so this data matrix is has row dimension at most 500.

**Author(s)**

Genevera I. Allen

**References**


**Examples**

```R
# batch-effect simulation
n = 250
p = 50
y = c(rep(1, 25), rep(2, 25))
mu1true = c(rep(.5, 25), rep(-.5, 25), rep(0, n-50))
mu2true = c(rep(-.5, 25), rep(.5, 25), rep(0, n-50))
Smat = cbind(matrix(mu1true, n, p/2), matrix(mu2true, n, p/2))
mus = c(1, 0, .5)
Bmatsig = matrix(1, n, 1)
Bmat = Bmatsig + matrix(rnorm(n*p)*.75, n, p)
xxt = matrix(rnorm(2*n^2), n, 2*n)
Sig = xxt %*% t(xxt)/(2*n); eSig = eigen(Sig);
xx = matrix(rnorm(n*p), n, p)
x.b = Smat + eSig$eigenvectors %*% diag(sqrt(eSig$values)) %*% eSig$eigenvectors xx + Bmat

# Transposable Sphering Algorithm
ans = TransSphere(x.b, y, fdr=.1, .15, .25)
```
#significant rows
ans$sig.rows

#true positive rate
sum(ans$sig.rows<=50)/50

#false positive rate
sum(ans$sig.rows>50)/200
Index

ACE, 2

covTranspose11, 3, 5–7
CVcov, 2, 4

meanTranspose, 5
MNloglike, 6

TransSphere, 4–6, 7