Package 'AdaptiveSparsity'

August 21, 2018

Type Package

Title Adaptive Sparsity Models

Version 1.6

Date 2018-08-20

Author Kristen Zygmunt, Eleanor Wong, Tom Fletcher

Maintainer Kris Campbell <kris@sci.utah.edu>

Description Implements Figueiredo EM algorithm for adaptive sparsity (Jeffreys prior) (see Figueiredo, M.A.T.; , ``Adaptive sparseness for supervised learning,'' Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.9, pp. 1150-1159, Sept. 2003) and Wong algorithm for adaptively sparse gaussian geometric models (see Wong, Eleanor, Suyash Awate, and P. Thomas Fletcher. ``Adaptive Sparsity in Gaussian Graphical Models.'' In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 311-319. 2013.)

License LGPL (>= 3.0)

Depends R (>= 3.0.2)

Imports MASS, Matrix, Rcpp

LinkingTo Rcpp (>= 0.12.13), RcppArmadillo (>= 0.2.0)

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-08-20 22:20:03 UTC

R topics documented:

aslm-package *Adaptive Sparsity Models Model*

Description

implements the adaptive sparse linear model using Figueiredo's EM algorithm for adaptive sparsity (Jeffreys prior) and the adaptively sparse gaussian graphical model using Wong's parameter-free algorithm.

Author(s)

Kristen Zygmunt, Eleanor Wong, Tom Fletcher

Maintainer: Kristen Zygmunt <krismz@sci.utah.edu>

References

Figueiredo, M.A.T.; , "Adaptive sparseness for supervised learning", Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.9, pp. 1150- 1159, Sept. 2003

Wong, Eleanor, Suyash Awate, and P. Thomas Fletcher. "Adaptive Sparsity in Gaussian Graphical Models."In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 311-319. 2013.

See Also

aslm, asggm

asggm *Adaptively Sparse Gaussian Graphical Model*

Description

implements a parameter-free adaptively sparse Gaussian graphical model.

Usage

```
## S3 method for class 'formula'
asggm(formula, data=list(), ...)
## Default S3 method:
asggm(x, iterations = 100000000, init = NULL, epsilon = 0.001, ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. See lm Details for further information.
data	an optional data frame, list or environment containing the variables in the model.
x	design matrix
iterations	number of iterations of the algorithm to run.
init	optional initialization, for instance, the cholesky of x. If NULL, it defaults to the cholesky of x.
epsilon	amount to add for numerical stability.
...	further arguments

Details

An effective approach to structure learning and parameter estimation for Gaussian graphical models is to impose a sparsity prior, such as a Laplace prior, on the entries of the precision matrix. We introduce a parameter-free method for estimating a precision matrix with sparsity that adapts to the data automatically, achieved by formulating a hierarchical Bayesian model of the precision matrix with a non-informative Jeffreys' hyperprior. We also naturally enforce the symmetry and positive-definiteness constraints on the precision matrix by parameterizing it with the Cholesky decomposition.

Value

asggm returns an object of class "asggm".

An object of class "asggm" is a list containing at least the following components:

Author(s)

Kristen Zygmunt, Eleanor Wong, Tom Fletcher

References

Wong, Eleanor, Suyash Awate, and P. Thomas Fletcher. "Adaptive Sparsity in Gaussian Graphical Models."In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 311-319. 2013.

Examples

```
A = diag(3)
asggm(A)
```

asggm-internal *asggm internal functions*

Description

These are the fitting and initialization functions used by asggm. These should generally *not* be used directly.

Usage

```
rCSL(x, iterations = 500, init = NULL, epsilon = 1e-05, ansL = NULL)
genL(kNodes, spP)
genData(L, nSamples)
```

Arguments

x	design matrix
iterations	number of iterations of the algorithm to run.
init	optional initialization, for instance, the cholesky of x. If NULL, it defaults to the cholesky of x.
epsilon	amount to add for numerical stability.
ansL	
kNodes	
spP	
L	L created by genL
nSamples	number of samples.

Details

rCSL calls the C++ code to compute the Wong EM algorithm. genL and genData are used together to create example data.

Value

rCSL returns a list with the following components:

References

Wong, Eleanor, Suyash Awate, and P. Thomas Fletcher. "Adaptive Sparsity in Gaussian Graphical Models."In Proceedings of the 30th International Conference on Machine Learning (ICML-13), pp. 311-319. 2013.

See Also

asggm, which should be used directly instead of these methods

aslm *Adaptive Sparse Linear Model*

Description

implements the adaptive sparse linear model using Figueiredo's EM algorithm for adaptive sparsity (Jeffreys prior)

Usage

```
## S3 method for class 'formula'
aslm(formula, data=list(), na.action=na.omit, ...)
## Default S3 method:
aslm(x, y, ...)
getSparseModel(object)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. See lm Details for further information.
data	an optional data frame, list or environment containing the variables in the model.
na.action	action to use when data contains NAs. Options include na.omit, na.exclude, na.fail
x	design matrix
y	vector of observations
...	further arguments
object	an object of class "aslm".

Value

aslm returns an object of class c("aslm", "lm").

An object of class "aslm" is a list containing at least the following components:

coefficients	a named vector of coefficients
residuals	the residuals, that is response minus fitted values.
fitted.values	the fitted mean values
rank	the numeric rnak of the fitted linear model
df	the residual degrees of freedom
call	the matched call
terms	the terms object used
sigma	

getSparseModel returns an object of class "lm" that is a model consisting of only the sparse nonzero variables from the original model.

Author(s)

Kristen Zygmunt, Eleanor Wong, Tom Fletcher

References

Figueiredo, M.A.T.; , "Adaptive sparseness for supervised learning", Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.9, pp. 1150- 1159, Sept. 2003

See Also

summary.aslm, logLik.aslm, print.aslm

Examples

```
s = aslm(Infant.Mortality~.,data=swiss)
m = getSparseModel(s)

summary(s)
coef(m)
```

---

aslm-internal *aslm internal functions*

Description

These are the fitting and initialization functions used by aslm. These should generally *not* be used.

Usage

```
figEM(x, y, init = NULL, stopDiff = 1e-08, epsilon = 1e-06, a = 1)
fit.ols.lm(x, y)
init.ones(x, y)
init.rnorm(x, y)
init.runif(x, y)
```

Arguments

x	design matrix of dimension n * p .
y	vector of observations of length n, or a matrix with n rows.
init	optional initialization, a list with components containing an initial estimate for beta and sigma
stopDiff	convergence criteria. Algorithm stops once difference in beta and sigma from one iteration to the next is less than stopDiff.
epsilon	amount to add to beta for numerical stability,
a	scaling of sigmaSqr to provide numerical stability for solving steps.

Details

figEM computes the Figueiredo EM algorithm for adaptive sparsity using Jeffreys prior.

fit.ols.lm computes an initial beta and sigma based on finding the lm.fit of the full design matrix.

init.ones computes an initial beta that is all ones and computes the associated sigmas.

init.rnorm computes an initial beta that is normally distributed with a mean of 0 and a standard deviation of 50

init.runif computes an initial beta that is uniformly distributed from 0 to 1

Currently, figEM uses fit.ols.lm to initialize beta and sigma if no init list is provided.

Value

figEM returns a list with the following components:

`coefficients`	`p` vector (also known as `beta`).
`vcov`	variance-covariance matrix.
`sigma`	norm of the model error.
`df`	degrees of freedom of residuals.

fit.ols.lm and init.ones are used to initialize beta and sigma if init is not provided to figEM. Each of these functions returns a list with the following components:

`beta`	initial `p` vector.
`sigma`	initial norm of the model error based on this initial beta.

References

Figueiredo, M.A.T.; , "Adaptive sparseness for supervised learning", Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.25, no.9, pp. 1150- 1159, Sept. 2003

See Also

`aslm`, which should be used directly instead of these methods

`aslm-methods` *Methods handled by lm*

Description

These methods are implemented by the `lm` parent class:

- `logLik` – Extract log-likelihood
- `predict` – Predict values based on linear model
- `nobs` – Extract the number of observations from a fit

See Also

`predict.lm`, `logLik.lm`, `nobs`

summary.aslm *Handling aslm objects*

Description

summary and print methods for class "aslm"

Usage

```
## S3 method for class 'aslm'
summary(object, ...)
## S3 method for class 'summary.aslm'
print(x, ...)
## S3 method for class 'aslm'
print(x, ...)
```

Arguments

object	An object of class "aslm", usually a result of a call to aslm
x	An object of class "summary.aslm" or "aslm"
...	Further arguments

Details

summary and print methods to help display and work with aslm objects.

Value

print prints a brief overview

Author(s)

Kristen Zygmunt, Eleanor Wong, Tom Fletcher

See Also

aslm

Index