Package 'camel'

February 19, 2015

Type Package

Title Calibrated Machine Learning

Version 0.2.0

Date 2013-09-09

Author Xingguo Li, Tuo Zhao, and Han Liu

Maintainer Xingguo Li <xingguo.leo@gmail.com>

Depends R (>= 2.15.0), lattice, igraph, MASS, Matrix

Description The package ``camel'' provides the implementation of a family of high-dimensional calibrated machine learning tools, including (1) LAD, SQRT Lasso and Calibrated Dantzig Selector for estimating sparse linear models; (2) Calibrated Multivariate Regression for estimating sparse multivariate linear models; (3) Tiger, Calibrated Clime for estimating sparse Gaussian graphical models. We adopt the combination of the dual smoothing and monotone fast iterative soft-thresholding algorithm (MFISTA). The computation is memory-optimized using the sparse matrix output, and accelerated by the path following and active set tricks.

License GPL-2

Repository CRAN

NeedsCompilation yes

Date/Publication 2013-09-09 15:49:04

R topics documented:

camel-package 2
camel-internal 3
camel.cmr 4
camel.plot 6
camel.slim 7
camel.tiger 10
camel.tiger.generator 12
camel.tiger.roc 15
camel.tiger.select 16
eyedata 18
plot.roc 19

plot.select 20
plot.sim 20
plot.slim 21
plot.tiger 22
print.cmr 22
print.roc 23
print.select 24
print.sim 24
print.slim 25
print.tiger 26

Index **27**

`camel-package` *camel: Calibrated Machine Learning*

Description

The package "camel" provides the implementation of a family of high-dimensional calibrated machine learning tools, including (1) LAD, SQRT Lasso and Calibrated Dantzig Selector for estimating sparse linear models; (2) Calibrated Multivariate Regression for estimating sparse multivariate linear models; (3) Tiger, Calibrated Clime for estimating sparse Gaussian graphical models. We adopt the combination of the dual smoothing and monotone fast iterative soft-thresholding algorithm (MFISTA). The computation is memory-optimized using the sparse matrix output, and accelerated by the path following and active set tricks.

Details

Package:	camel
Type:	Package
Version:	0.2.0
Date:	2013-09-09
License:	GPL-2

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

References

1. A. Belloni, V. Chernozhukov, and L. Wang. Pivotal recovery of sparse signals via conic programming. *Biometrika*, 2012.
2. L. Wang. L1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 2013.

3. E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 2007.
4. T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 2011.
5. H. Liu and L. Wang. TIGER: A tuning-insensitive approach for optimally estimating large undirected graphs. *Technical Report*, 2012.
6. L. Han, L. Wang, and T. Zhao. Multivariate Regression with Calibration. *http://arxiv.org/abs/1305.2238*, 2013.
7. T. Zhao and H. Liu, Sparse Precision Matrix Estimation with Calibration. *Advances in Neural Information Processing systems*, 2013.
8. A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 2009.
9. Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 2005.

See Also

camel.tiger, camel.slim and camel.cmr .

`camel-internal` *Internal camel functions*

Description

Internal camel functions

Usage

```
tiger.likelihood(Sigma, Omega)
tiger.tracel2(Sigma, Omega)
camel.tiger.cv(obj, loss=c("likelihood", "tracel2"), fold=5)
part.cv(n, fold)
camel.tiger.clime.mfista(Sigma, d, maxdf, mu, lambda, shrink, prec, max.ite)
camel.tiger.slasso.mfista(data, n, d, maxdf, mu, lambda, shrink, prec, max.ite)
camel.slim.lad.mfista(Y, X, lambda, nlambda, n, d, maxdf, mu, max.ite, prec,
                      intercept, verbose)
camel.slim.sqrt.mfista(Y, X, lambda, nlambda, n, d, maxdf, mu, max.ite, prec,
                       intercept, verbose)
camel.slim.dantzig.mfista(Y, X, lambda, nlambda, n, d, maxdf, mu, max.ite, prec,
                          intercept, verbose)
camel.cmr.mfista(Y, X, lambda, nlambda, n, d, m, mu, max.ite, prec)
```

Arguments

`Sigma`	Covariance matrix.
`Omega`	Inverse covariance matrix.

obj	An object with S3 class returned from "tiger".
loss	Type of loss function for cross validation.
fold	The number of fold for cross validatio.
n	The number of observations (sample size).
d	Dimension of data.
m	Columns of parameters in multivariate regression.
maxdf	Maximal degree of freedom.
lambda	Grid of non-negative values for the regularization parameter lambda.
nlambda	The number of the regularization parameter lambda.
shrink	Shrinkage of regularization parameter based on precision of estimation.
mu	The smooth surrogate parameter.
prec	Stopping criterion.
max.ite	Maximal value of iterations.
data	n by d data matrix.
Y	Dependent variables in linear regression.
X	Design matrix in linear regression.
intercept	Whether the intercept is included in the model.
verbose	Tracing information printing is disabled if verbose = FALSE.

Details

These are not intended for use by users.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger, camel.slim, camel.cmr and camel-package.

---

camel.cmr *Calibrated Multivariate Regression*

Description

The function "camel.cmr" implements calibrated multivariate regression using jointly sparse regularization.

Usage

```
camel.cmr(X, Y, lambda = NULL, nlambda = NULL, prec = 1e-3,
          max.ite = 1e3, mu = 0.01, verbose = TRUE)
```

Arguments

Y	The n by m dimensional response matrix.
X	The n by d design matrix.
lambda	A sequence of decresing positive value to control the regularization. Typical usage is to leave the input `lambda = NULL` and have the program compute its own `lambda` sequence based on `nlambda`, d and m. Users can also specify a sequence to override this.
nlambda	The number of values used in `lambda`. Default value is 10.
prec	Stopping criterion. The default value is 1e-3.
max.ite	The iteration limit. The default value is 1e3.
mu	The smoothing parameter. The default value is 0.01.
verbose	Tracing information is disabled if `verbose = FALSE`. The default value is `TRUE`.

Details

Calibrated multivariate regression adjusts the regularization with respect to the noise level of each task. Thus it achieves improved statistical performance and the tuning insensitiveness.

Value

An object with S3 class `"camel.cmr"` is returned:

beta	A list of matrice of regression estimates where each entry corresponds to a regularization parameter.
intercept	The value of intercepts corresponding to regularization parameters.
Y	The value of `Y` used in the program.
X	The value of `X` used in the program.
lambda	The sequence of regularization parameters `lambda` used in the program.
nlambda	The number of values used in `lambda`.
sparsity	The sparsity levels of the solution path.
ite	A list of vectors where ite[[1]] is the number of external iteration and ite[[2]] is the number of internal iteration with the i-th entry corresponding to the i-th regularization parameter.
verbose	The `verbose` from the input.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

References

1. L. Han, L. Wang, and T. Zhao. Multivariate Regression with Calibration. *http://arxiv.org/abs/1305.2238*, 2013.

See Also

camel-package.

Examples

```
## Generate the design matrix and regression coefficient vector
n = 200
d = 400
m = 13
Sigma = matrix(0.5,d,d)
diag(Sigma) = 1
X = mvrnorm(n,rep(0,d),Sigma)
B = matrix(0,d,m)
B[1,] = 3
B[2,] = 2
B[4,] = 1.5
W = matrix(rnorm(n*m,0,1),n,m)
sig = sqrt(2)
D = sig*diag(2^(c(0:-12)/4))
Z = W%*%D
Y = X%*%B + Z
out = camel.cmr(X, Y)
```

---

camel.plot *Graph visualization*

Description

Implements the graph visualization using adjacency matrix. It can automatic organize 2D embedding layout.

Usage

```
camel.plot(G, epsflag = FALSE, graph.name = "default", cur.num = 1,
           location)
```

Arguments

G	The adjacency matrix corresponding to the graph.
epsflag	If epsflag = TRUE, save the plot as an eps file in the target directory. The default value is FALSE.
graph.name	The name of the output eps files. The default value is "default".

cur.num	The number of plots saved as eps files. Only applicable when epsflag = TRUE. The default value is 1.
location	Target directory. The default value is the current working directory.

Details

The user can change cur.num to plot several figures and select the best one. The implementation is based on the popular package "igraph".

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel and camel-package

Examples

```
## visualize the hub graph
L = camel.tiger.generator(graph = "hub")
camel.plot(L$theta)

## visualize the band graph
L = camel.tiger.generator(graph = "band",g=5)
camel.plot(L$theta)

## visualize the cluster graph
L = camel.tiger.generator(graph = "cluster")
camel.plot(L$theta)

#show working directory
getwd()
#plot 5 graphs and save the plots as eps files in the working directory
camel.plot(L$theta, epsflag = TRUE, cur.num = 5)
```

---

camel.slim *Calibrated Linear Regression*

Description

The function "camel.slime" implements LAD/L1 Lasso, SQRT/L2 Lasso, and carlibrated Dantizg selector using L1 regularization.

Usage

```
camel.slim(X, Y, lambda = NULL, nlambda = NULL, lambda.min.ratio = NULL,
           method="lq", q = 2, prec = 1e-4, max.ite = 1e4, mu = 0.01,
           intercept = TRUE, verbose = TRUE)
```

Arguments

Y	The n dimensional response vector.
X	The n by d design matrix.
lambda	A sequence of decresing positive value to control the regularization. Typical usage is to leave the input lambda = NULL and have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Users can also specify a sequence to override this. Default value is from $lambda.max$ to lambda.min.ratio*lambda.max. For Lq regression, the default value of $lambda.max$ is $\pi\sqrt{\log(d)/n}$. For Dantzig selector, the default value of $lambda.max$ is the minimum regularization parameter, which yields an all-zero estiamtes.
nlambda	The number of values used in lambda. Default value is 5.
lambda.min.ratio	The smallest value for lambda, as a fraction of the uppperbound (MAX) of the regularization parameter. The program can automatically generate lambda as a sequence of length = nlambda starting from MAX to lambda.min.ratio*MAX in log scale. The default value is 0.25 for Lq Lasso and 0.5 for Dantzig selector.
method	Dantzig selector is applied if method = "dantzig" and L_q Lasso is applied if method = "lq". The default value is "lq".
q	The loss function used in Lq Lasso. It is only applicable when method = "lq" and must be either 1 or 2. The default value is 2.
prec	Stopping criterion. The default value is 1e-4.
max.ite	The iteration limit. The default value is 1e4.
mu	The smoothing parameter. The default value is 0.01.
intercept	Whether the intercept is included in the model. The defulat value is TRUE.
verbose	Tracing information is disabled if verbose = FALSE. The default value is TRUE.

Details

Calibrated Linear Regression adjust the regularization with respect to the noise level. Thus it achieves both improved finite sample performance and tuning insensitiveness.

Value

An object with S3 class "camel.slim" is returned:

beta	A matrix of regression estimates whose columns correspond to regularization parameters.
intercept	The value of intercepts corresponding to regularization parameters.
Y	The value of Y used in the program.
X	The value of X used in the program.
lambda	The sequence of regularization parameters lambda used in the program.
nlambda	The number of values used in lambda.
method	The method from the input.

sparsity	The sparsity levels of the solution path.
ite	A list of vectors where ite[[1]] is the number of external iteration and ite[[2]] is the number of internal iteration with the i-th entry corresponding to the i-th regularization parameter.
verbose	The verbose from the input.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

References

1. A. Belloni, V. Chernozhukov and L. Wang. Pivotal recovery of sparse signals via conic programming. *Biometrika*, 2012.
2. L. Wang. L1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 2013.
3. E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 2007.

See Also

camel-package.

Examples

```
## Generate the design matrix and regression coefficient vector
n = 200
d = 400
X = matrix(rnorm(n*d), n, d)
beta = c(3,2,0,1.5,rep(0,d-4))

## Generate response using Gaussian noise, and fit a sparse linear model using SQRT Lasso
eps.sqrt = rnorm(n)
Y.sqrt = X%*%beta + eps.sqrt
out.sqrt = camel.slim(X = X, Y = Y.sqrt, lambda = seq(0.8,0.2,length.out=5))

## Generate response using Cauchy noise, and fit a sparse linear model using LAD Lasso
eps.lad = rt(n = n, df = 1)
Y.lad = X%*%beta + eps.lad
out.lad = camel.slim(X = X, Y = Y.lad, q = 1, lambda = seq(0.5,0.2,length.out=5))

## Visualize the solution path
plot(out.sqrt)
plot(out.lad)
```

camel.tiger *Tuning Insensitive Graph Estimation and Regression*

Description

The function "camel.cmr" implements TIGER and Calibrated CLIME using L1 norm regularization

Usage

```
camel.tiger(data, lambda = NULL, nlambda = NULL, lambda.min.ratio = NULL,
      method = "slasso", sym = "or", shrink=NULL, prec = 1e-4, mu = 0.01,
      max.ite = 1e4, standardize = FALSE, correlation = FALSE,
      perturb = TRUE, verbose = TRUE)
```

Arguments

data	There are 2 options for "clime": (1) data is an n by d data matrix (2) a d by d sample covariance matrix. The program automatically identifies the input matrix by checking the symmetry. (n is the sample size and d is the dimension). For "slasso", covariance input is not supported.
lambda	A sequence of decresing positive numbers to control the regularization. Typical usage is to leave the input lambda = NULL and have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Users can also specify a sequence to override this. Default value is from $lambda.max$ to lambda.min.ratio*lambda.max. For TIGER, the default value of $lambda.max$ is $\pi\sqrt{\log(d)/n}$. For CLIME , the default value of $lambda.max$ is the minimum regularization parameter, which yields an all-zero off-diagonal estiamtes.
nlambda	The number of values used in lambda. Default value is 10.
lambda.min.ratio	The smallest value for lambda, as a fraction of the uppperbound (MAX) of the regularization parameter. The program can automatically generate lambda as a sequence of length = nlambda starting from MAX to lambda.min.ratio*MAX in log scale. The default value is 0.25 for TIGER and 0.5 for CLIME.
method	TIGER is applied if method = "slasso", CLIME is applied if method="clime". Default value is "slasso".
sym	Symmetrization of output graphs. If sym = "and", the edge between node i and node j is selected ONLY when both node i and node j are selected as neighbors for each other. If sym = "or", the edge is selected when either node i or node j is selected as the neighbor for each other. The default value is "or".
shrink	Shrinkage of regularization parameter based on precision of estimation. The default value is 1.5 if method = "clime" and the default value is 0 if method="slasso" or method = "aclime".
prec	Stopping criterion. The default value is 1e-4.
mu	The smoothing parameter. The default value is 0.01.

max.ite	The iteration limit. The default value is 1e4.
standardize	All varaibles are standardized to have mean zero and standard deviation one if standardize = TRUE. The default value is FALSE.
correlation	Correlation matrix is used as the input of Sigma for method = "clime" if correlation = TRUE. The default value is FALSE.
perturb	The diagonal of Sigma is added by a positive value to guarantee that Sigma is positive definite if perturb = TRUE. User can specify a numeric value for perturbe. The default value is TRUE.
verbose	Tracing information is disabled if verbose = FALSE. The default value is TRUE.

Details

TIGER and Calibrated CLIME adjust the regularization with respect to each column of the sparse precision matrix. Thus it achieves both improved finite sample performance and tuning insensitiveness.

Value

An object with S3 class "tiger" is returned:

data	The n by d data matrix or d by d sample covariance matrix from the input.
cov.input	An indicator of the sample covariance.
lambda	The sequence of regularization parameters lambda used in the program.
nlambda	The number of values used in lambda.
icov	A list of d by d precision matrices corresponding to regularization parameters.
sym	The sym from the input.
method	The method from the input.
path	A list of d by d adjacency matrices of estimated graphs as a graph path corresponding to lambda.
sparsity	The sparsity levels of the graph path.
ite	If method = "clime", it is a list of two matrices where ite[[1]] is the number of external iterations and ite[[2]] is the number of internal iterations with the entry of (i,j) as the number of iteration of i-th column and j-th lambda. If method="slasso", it is a matrix of iteration with the entry of (i,j) as the number of iteration of i-th column and j-th lambda.
df	It is a d by nlambda matrix. Each row contains the number of nonzero coefficients along the lasso solution path.
standardize	The standardize from the input.
correlation	The correlation from the input.
perturb	The perturb from the input.
verbose	The verbose from the input.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

References

1. T. Cai, W. Liu and X. Luo. A constrained L1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 2011.
2. H. Liu and L. Wang. TIGER: A tuning-insensitive approach for optimally estimating large undirected graphs. *Technical Report*, 2012.

See Also

camel-package, camel.tiger.generator, camel.tiger.select, camel.plot, camel.tiger.roc, plot.tiger, plot.select, plot.roc, plot.sim, print.tiger, print.select, print.roc and print.sim.

Examples

```
## generating data
n = 100
d = 100
D = camel.tiger.generator(n=n,d=d,graph="hub",g=10)
plot(D)

## sparse precision matrix estimation with method "clime"
out1 = camel.tiger(D$data, method = "clime")
plot(out1)
camel.plot(out1$path[[7]])

## sparse precision matrix estimation with method "slasso"
out2 = camel.tiger(D$data, method = "slasso")
plot(out2)
camel.plot(out2$path[[4]])
```

`camel.tiger.generator` *Data generator for undirected graph estimation.*

Description

Implements the data generation from multivariate normal distributions with different graph structures, including `"random"`, `"hub"`, `"cluster"`, `"band"`, and `"scale-free"`.

Usage

```
camel.tiger.generator(n = 200, d = 50, graph = "random", v = NULL, u = NULL,
            g = NULL, prob = NULL, seed = NULL, vis = FALSE, verbose = TRUE)
```

Arguments

n	The number of observations (sample size). The default value is 200.
d	The number of variables (dimension). The default value is 50.
graph	The graph structure with 5 options: "random", "hub", "cluster", "band", and "scale-free".
v	The off-diagonal elements of the precision matrix, controlling the magnitude of partial correlations with u. The default value is 0.3.
u	A positive number being added to the diagonal elements of the precision matrix, to control the magnitude of partial correlations. The default value is 0.1.
g	For "cluster" or "hub" graph, g is the number of hubs or clusters in the graph. The default value is about d/20 if d >= 40 and 2 if d < 40. For "band" graph, g is the bandwidth and the default value is 1. NOT applicable to "random" graph.
prob	For "random" graph, it is the probability that a pair of nodes has an edge. The default value is 3/d. For "cluster" graph, it is the probability that a pair of nodes has an edge in each cluster. The default value is 6*g/d if d/g <= 30 and 0.3 if d/g > 30. NOT applicable to "hub", "band", and scale-free graphs.
seed	Set seed for data generation. The default value is 1.
vis	Visualize the adjacency matrix of the true graph structure, the graph pattern, the covariance matrix and the empirical covariance matrix. The default value is FALSE.
verbose	If verbose = FALSE, tracing information printing is disabled. The default value is TRUE.

Details

Given the adjacency matrix theta, the graph patterns are generated as below:

(I) "random": Each pair of off-diagonal elements are randomly set theta[i,j]=theta[j,i]=1 for i!=j with probability prob, and 0 other wise. It results in about d*(d-1)*prob/2 edges in the graph.

(II)"hub":The row/columns are evenly partitioned into g disjoint groups. Each group is associated with a "center" row i in that group. Each pair of off-diagonal elements are set theta[i,j]=theta[j,i]=1 for i!=j if j also belongs to the same group as i and 0 otherwise. It results in d - g edges in the graph.

(III)"cluster":The row/columns are evenly partitioned into g disjoint groups. Each pair of off-diagonal elements are set theta[i,j]=theta[j,i]=1 for i!=j with the probability probif both i and j belong to the same group, and 0 other wise. It results in about g*(d/g)*(d/g-1)*prob/2 edges in the graph.

(IV)"band": The off-diagonal elements are set to be theta[i,j]=1 if 1<=|i-j|<=g and 0 other wise. It results in (2d-1-g)*g/2 edges in the graph.

(V) "scale-free": The graph is generated using B-A algorithm. The initial graph has two connected nodes and each new node is connected to only one node in the existing graph with the

probability proportional to the degree of the each node in the existing graph. It results in d edges in the graph.

The adjacency matrix theta has all diagonal elements equal to 0. To obtain a positive definite covariance matrix, the smallest eigenvalue of theta*v (denoted by e) is computed. Then we set the covariance matrix equal to cov2cor(solve(theta*v+(|e|+0.1+u)*I)) to generate multivariate normal data.

Value

An object with S3 class "sim" is returned:

data	The n by d matrix for the generated data
sigma	The covariance matrix for the generated data
omega	The precision matrix for the generated data
sigmahat	The empirical covariance matrix for the generated data
theta	The adjacency matrix of true graph structure (in sparse matrix representation) for the generated data

Author(s)

Xingguo Li, Tuo Zhao and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel and camel-package

Examples

```
## band graph with bandwidth 3
L = camel.tiger.generator(graph = "band", g = 3)
plot(L)

## random sparse graph
L = camel.tiger.generator(vis = TRUE)

## random dense graph
L = camel.tiger.generator(prob = 0.5, vis = TRUE)

## hub graph with 6 hubs
L = camel.tiger.generator(graph = "hub", g = 6, vis = TRUE)

## cluster graph with 8 clusters
L = camel.tiger.generator(graph = "cluster", g = 8, vis = TRUE)

## scale-free graphs
L = camel.tiger.generator(graph="scale-free", vis = TRUE)
```

`camel.tiger.roc` *Draw ROC Curve for a graph path*

Description

Draws ROC curve for a graph path according to the true graph structure.

Usage

```
camel.tiger.roc(path, theta, verbose = TRUE)
```

Arguments

`path`	A graph path.
`theta`	The true graph structure.
`verbose`	If `verbose = FALSE`, tracing information printing is disabled. The default value is `TRUE`.

Details

To avoid the horizontal oscillation, false positive rates is automatically sorted in the ascent oder and true positive rates also follow the same order.

Value

An object with S3 class "roc" is returned:

`F1`	The F1 scores along the graph path.
`tp`	The true positive rates along the graph path
`fp`	The false positive rates along the graph paths
`AUC`	Area under the ROC curve

Note

For a lasso regression, the number of nonzero coefficients is at most `n-1`. If `d>>n`, even when regularization parameter is very small, the estimated graph may still be sparse. In this case, the AUC may not be a good choice to evaluate the performance.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

`camel.tiger` and `camel-package`

Examples

```
#generate data
L = camel.tiger.generator(d = 50, graph = "random", prob = 0.1)
out1 = camel.tiger(L$data, lambda=10^(seq(log10(.4), log10(0.03), length.out=20)))

#draw ROC curve
Z1 = camel.tiger.roc(out1$path,L$theta)

#Maximum F1 score
max(Z1$F1)
```

`camel.tiger.select` *Model selection for high-dimensional undirected graph estimation*

Description

Implements the regularization parameter selection for high dimensional undirected graph estimation. The optional approaches are stability approach to regularization selection (stars) and cross validation selection (cv).

Usage

```
camel.tiger.select(est, criterion = "stars", stars.subsample.ratio = NULL,
                   stars.thresh = 0.1,rep.num = 20, fold = 5,
                   loss="likelihood", verbose = TRUE)
```

Arguments

`est`	An object with S3 class `"camel.tiger"`
`criterion`	Model selection criterion. `"stars"` and `"cv"` are available for both graph estimation methods. The default value is `"stars"`.
`stars.subsample.ratio`	The subsampling ratio. The default value is `10*sqrt(n)/n` when `n>144` and `0.8` when `n<=144`, where `n` is the sample size. Only applicable when `criterion = "stars"`.
`stars.thresh`	The variability threshold in stars. The default value is `0.1`. Only applicable when `criterion = "stars"`.
`rep.num`	The number of subsamplings. The default value is `20`.
`fold`	The number of folds used in cross validation. The default value is `5`. Only applicable when `criterion = "cv"`.
`loss`	Loss to be used in cross validation. Two losses are available: `"likelihood"` and `"tracel2"`. Default `"likelihood"`. Only applicable when `criterion = "cv"`.
`verbose`	If `verbose = FALSE`, tracing information printing is disabled. The default value is `TRUE`.

Details

Stability approach to regularization selection (stars) is a natural way to select optimal regularization parameter for all three estimation methods. It selects the optimal graph by variability of subsamplings and tends to over-select edges in Gaussian graphical models. Besides selecting the regularization parameters, stars can also provide an additional estimated graph by merging the corresponding subsampled graphs using the frequency counts. The K-fold cross validation is also provided for selecting the parameter `lambda`, and two loss functions are adopted as follow

$$likelihood : Tr(\Sigma\Omega) - \log|\Omega|$$

$$tracel2 : Tr(diag(\Sigma\Omega - I)^2).$$

Value

An object with S3 class "select" is returned:

`refit`	The optimal graph selected from the graph path
`opt.icov`	The optimal precision matrix selected.
`merge`	The graph path estimated by merging the subsampling paths. Only applicable when the input `criterion = "stars"`.
`variability`	The variability along the subsampling paths. Only applicable when the input `criterion = "stars"`.
`opt.index`	The index of the selected regularization parameter.
`opt.lambda`	The selected regularization/thresholding parameter.
`opt.sparsity`	The sparsity level of `"refit"`.

and anything else inluded in the input `est`

Note

The model selection is NOT available when the data input is the sample covaraince matrix.

Author(s)

Xingguo Li, Tuo Zhao and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

References

1. H. Liu and L. Wang. TIGER: A tuning-insensitive approach for optimally estimating large undirected graphs. *Technical Report*, 2012.
2. T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 2011.

See Also

`camel.tiger` and `camel-package`.

Examples

```
#generate data
L = camel.tiger.generator(d = 20, graph="hub")
out1 = camel.tiger(L$data)

#model selection using stars
out1.select2 = camel.tiger.select(out1, criterion = "stars", stars.thresh = 0.05)
plot(out1.select2)

#model selection using cross validation
out1.select3 = camel.tiger.select(out1, criterion = "cv")
plot(out1.select3)
```

`eyedata` *Gene expression data for Bardet-Biedl syndrome from Scheetz et al. (2006)*

Description

Gene expression data (20 genes for 120 samples) from the microarray experiments of mammalian-eye tissue samples of Scheetz et al. (2006).

Usage

```
data(eyedata)
```

Format

The format is a list containing conatins a matrix and a vector. 1. x - an 120 by 200 matrix, which represents the data of 120 rats with 200 gene probes. 2. y - a 120-dimensional vector of, which represents the expression level of TRIM32 gene.

Details

This data set contains 120 samples with 200 predictors

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

References

1. T. Scheetz, k. Kim, R. Swiderski, A. Philp, T. Braun, K. Knudtson, A. Dorrance, G. DiBona, J. Huang, T. Casavant, V. Sheffield, E. Stone .Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences of the United States of America*, 2006.

See Also

camel-package.

Examples

```
data(eyedata)
image(x)
```

`plot.roc` *Plot function for S3 class "roc"*

Description

Plot the ROC curve for an object with S3 class `"roc"`

Usage

```
## S3 method for class 'roc'
plot(x, ...)
```

Arguments

`x`	An object with S3 class `"roc"`
`...`	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger.roc

plot.select *Plot function for S3 class "select"*

Description

Plot the optimal graph by model selection.

Usage

```
## S3 method for class 'select'
plot(x, ...)
```

Arguments

x	An object with S3 class "select"
...	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger and camel.tiger.select

plot.sim *Plot function for S3 class "sim"*

Description

Visualize the covariance matrix, the empirical covariance matrix, the adjacency matrix and the graph pattern of the true graph structure.

Usage

```
## S3 method for class 'sim'
plot(x, ...)
```

Arguments

x	An object with S3 class "sim"
...	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

`camel.tiger` and `camel.tiger.generator`

`plot.slim` *Plot function for S3 class "slim"*

Description

Visualize the solution path of regression estimate corresponding to regularization paramters.

Usage

```
## S3 method for class 'slim'
plot(x, ...)
```

Arguments

`x`	An object with S3 class `"slim"`
`...`	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

`camel.slim`

plot.tiger *Plot function for S3 class "camel.tiger"*

Description

Plot sparsity level information and 3 typical sparse graphs from the graph path.

Usage

```
## S3 method for class 'tiger'
plot(x, align = FALSE, ...)
```

Arguments

x	An object with S3 class "tiger"
align	If align = FALSE, 3 plotted graphs are aligned
...	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger and camel-package

print.cmr *Print a camel.cmr Object*

Description

Print a summary of the information about the cmr object.

Usage

```
## S3 method for class 'cmr'
print(x, ...)
```

Arguments

x	The cmr object.
...	Additional print options.

Details

This call simply outlines the options used for computing a cmr object.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.cmr and camel-package.

`print.roc` *Print function for S3 class "roc"*

Description

Print the information about true positive rates, false positive rates, the area under curve and maximum F1 score

Usage

```
## S3 method for class 'roc'
print(x, ...)
```

Arguments

x	An object with S3 class "roc"
...	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger and camel.tiger.roc

print.select *Print function for S3 class "select"*

Description

Print the information about the model usage, graph dimension, model selection criterion, sparsity level of the optimal graph

Usage

```
## S3 method for class 'select'
print(x, ...)
```

Arguments

x	An object with S3 class "select"
...	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger.select and camel.tiger

print.sim *Print function for S3 class "sim"*

Description

Print the information about the sample size, the dimension, the pattern and sparsity of the true graph structure.

Usage

```
## S3 method for class 'sim'
print(x, ...)
```

Arguments

x	An object with S3 class "sim"
...	System reserved (No specific usage)

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.tiger and camel.tiger.generator

---

print.slim *Print a camel.slim Object*

Description

Print a summary of the information about the slim object.

Usage

```
## S3 method for class 'slim'
print(x, ...)
```

Arguments

x	The slim object.
...	Additional print options.

Details

This call simply outlines the options used for computing a slim object.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel.slim and camel-package.

print.tiger *Print a camel.tiger Object*

Description

Print a summary of the information about the tiger object.

Usage

```
## S3 method for class 'tiger'
print(x, ...)
```

Arguments

x	The tiger object.
...	Additional print options.

Details

This call simply outlines the options used for computing a tiger object.

Author(s)

Xingguo Li, Tuo Zhao, and Han Liu
Maintainer: Xingguo Li <xingguo.leo@gmail.com>

See Also

camel and camel-package.

Index

camel, *7*, *14*, *26*
camel *(*camel-package*)*, 2
camel-internal, 3
camel-package, 2
camel.cmr, *3, 4*, 4, *23*
camel.cmr.mfista *(*camel-internal*)*, 3
camel.plot, 6, *12*
camel.slim, *3, 4*, 7, *21*, *25*
camel.slim.dantzig.mfista *(*camel-internal*)*, 3
camel.slim.lad.mfista *(*camel-internal*)*, 3
camel.slim.sqrt.mfista *(*camel-internal*)*, 3
camel.tiger, *3, 4*, 10, *15*, *17*, *20–25*
camel.tiger.clime.mfista *(*camel-internal*)*, 3
camel.tiger.cv *(*camel-internal*)*, 3
camel.tiger.generator, *12*, 12, *21*, *25*
camel.tiger.roc, *12*, 15, *19*, *23*
camel.tiger.select, *12*, 16, *20*, *24*
camel.tiger.slasso.mfista *(*camel-internal*)*, 3

eyedata, 18

part.cv *(*camel-internal*)*, 3
plot.roc, *12*, 19
plot.select, *12*, 20
plot.sim, *12*, 20
plot.slim, 21
plot.tiger, *12*, 22
print.cmr, 22
print.roc, *12*, 23
print.select, *12*, 24
print.sim, *12*, 24
print.slim, 25
print.tiger, *12*, 26

tiger.likelihood *(*camel-internal*)*, 3
tiger.tracel2 *(*camel-internal*)*, 3