Package ‘SMLE’

June 24, 2020

Title Joint Feature Screening via Sparse MLE
Version 0.4.1
Author Qianxiang Zang, Chen Xu, Kelly Burkett
Maintainer Qianxiang Zang <qzang023@uottawa.ca>
Imports foreach, mvnfast, doParallel

License GPL-2
Depends R (>= 4.0.0), glmnet(>= 4.0)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
NeedsCompilation no
Repository CRAN
Date/Publication 2020-06-24 14:30:08 UTC

R topics documented:

smle-package ................................................................. 2
Gen_Data ................................................................. 3
plot.selection ............................................................ 5
plot.smle ................................................................. 6
predict.smle .............................................................. 7
print.sdata ............................................................... 8
print.selection .......................................................... 9
print.smle ............................................................... 9
SMLE ................................................................. 10
smle_select ............................................................. 13
Description

Feature screening is a powerful tool in processing ultra-high dimensional data. It attempts to screen out most irrelevant features before an elaborative analysis. This package provides an efficient implementation of SMLE-screening for linear, logistic, and Poisson models, where joint effects among features are naturally incorporated in the screening process. The package also provides a function for conducting feature selection based on a user-specified selection criterion after screening.

Details

Package: smle
Type: Package
Version: 0.2
Date: 2020-01-29
License: GPL-2

Input a $n \times 1$ response vector $Y$ and a $n \times p$ predictor (feature) matrix $X$. The package outputs a set of $k < n$ features that seem to be most relevant for joint regression. Moreover, the package provides a data simulator that generates a synthetic datasets from high-dimensional GLMs, which accommodate both numerical and categorical features with commonly used correlation structures.

Important functions:
Gen_Data
SMLE
smle_select
smle_predict

Author(s)

Qianxiang Zang, Chen Xu, Kelly Burkett
Maintainer: Qianxiang Zang <qzang023@uottawa.ca>

References


**Examples**

```r
gen_data(123.456)
#Generate correlated data
Data<-Gen_Data(correlation="MA",family = "gaussian")
print(Data)

# joint feature screening via SMLE
fit<-SMLE(Data$Y,Data$X,k=10,family = "gaussian")
print(fit)
plot(fit)

#Are there any features missed after screening?
setdiff(Data$index, fit$Retained_Feature_IDs)

# Elaborative selection after screening
E<-smle_select(fit,gamma_ebic = 0.5,vote = FALSE)

#Are there any features missed after selection?
setdiff( Data$index ,E$Retained_Feature_IDs)
print(E)
plot(E)
```

---

**Gen_Data**  
*Data simulator for high-dimensional*

**Description**

This function generates synthetic datasets from GLMs with a user-specified correlation structure. It permits both numerical and categorical features, whose quantity can be larger than the sample size.

**Usage**

```r
Gen_Data(
  n = 200,
  p = 5000,
  sigma = 1,
  num_ctgidx = NULL,
  pos_ctgidx = NULL,
  num_truecoef = NULL,
  pos_truecoef = NULL,
  level_ctgidx = NULL,
  effect_truecoef = NULL,
  correlation = c("ID", "AR", "MA", "CS"),
  rho = 0.5,
  family = c("gaussian", "binomial", "poisson")
)
```
### Arguments

- **n**: Sample size, number of rows of for the feature matrix to be generated.
- **p**: Number of columns for the feature matrix to be generated.
- **sigma**: Parameter for noise level.
- **num_ctgidx**: The number of features that are categorical. Set to FALSE for only numerical features. Default is FALSE.
- **pos_ctgidx**: Vector of indices denoting which columns are categorical.
- **num_truecoef**: The number of features (columns) that affect response. Default is 5.
- **pos_truecoef**: Vector of indices denoting which features (columns) affect the response variable.
- **level_ctgidx**: A vector to indicate the levels of categorical features in `pos_ctgidx`. Default is 2.
- **effect_truecoef**: Effects for the relevant features in `pos_truecoef`.
- **correlation**: Correlation structure among features. correlation = 'ID' for independent, correlation = 'MA' for moving average, correlation = 'CS' for compound symmetry, correlation = 'AR' for auto regressive. Default is "ID". For more information see details.
- **rho**: Parameter controlling the correlation strength. See details.
- **family**: Models to generate the response from the synthetic features: 'gaussian' for normally distributed data, 'poisson' for non-negative counts, 'binomial' for binary (0-1).

### Details

Simulated data \((y_i, x_i)\) for \(i = 1, ..., n\) are generated as follows: First, we generate a \(p \times 1\) model coefficient vector beta with all entries being zero, except on the positions specified in `pos_truecoef`, on which `effect_truecoef` is used. When `pos_truecoef` is not specified, we randomly choose `num_truecoef` positions from the coefficient vector. When `effect_truecoef` is not specified, we randomly set the strength of the true model coefficients as follow:

\[(0.5 + U) \cdot Z\]

where \(U\) is a uniform distribution from 0 to 1, and \(Z\) is a binomial distribution \(P(Z = 1) = 1/2, P(Z = -1) = 1/2\).

Next, we generate a \(n \times p\) feature matrix \(X\) based on the choice in `correlation` specified as follows:

**Independent (ID)**: all features are independently generated from \(N(0, 1)\).

**Moving average (MA)**: candidate features \(x_1, ..., x_p\) are joint normal, marginally \(N(0, 1)\), with

\[\text{cov}(x_j, x_{j-1}) = \rho, \quad \text{cov}(x_j, x_{j-2}) = \frac{\rho^2}{2} \quad \text{and} \quad \text{cov}(x_j, x_h) = 0 \quad \text{for} \quad |j - h| \geq 3.\]

**Compound symmetry (CS)**: candidate features \(x_1, ..., x_p\) are joint normal, marginally \(N(0, 1)\), with

\[\text{cov}(x_j, x_h) = \rho \quad \text{if} \quad j, h \quad \text{are both in the set of important features and} \quad \text{cov}(x_j, x_h) = \frac{\rho^2}{2} \quad \text{when only} \quad \text{one of} \quad j \quad \text{or} \quad h \quad \text{are in the set of important features.}\]

**Auto-regressive (AR)**: candidate features \(x_1, ..., x_p\) are joint normal marginally \(N(0, 1)\), with

\[\text{cov}(x_j, x_h) = \rho^{|j-h|} \quad \text{for all} \quad j \quad \text{and} \quad h.\]
Then, generate the response variable $Y$ according to its response type. For Gaussian model, $Y = x^T \cdot \beta + \epsilon$ where $\epsilon \sim N(0,1)$. For the binary model let $\pi = P(Y = 1|x)$. Sample $y$ from Bernoulli($\pi$) where $\logit(\pi) = x^T \cdot \beta$. Finally, for the Poisson model, $Y$ is generated from Poisson distribution with the link $\pi = exp(x^T \cdot \beta)$. For more details (see reference below)

**Value**

Returns a "sdata" object with

- $Y$: Response variable vector of length $n$
- $X$: Feature matrix or Dataframe (Matrix if `num_ctgidx` = `FALSE` and dataframe otherwise)
- index: Vector of columns indices of $X$ for the features that affect the response variables (relevant features).
- $\beta$: Vector of effects for the relevant features.

**References**


**Examples**

```r
#Simulating data with binomial response and independent structure.
Data<-Gen_Data(family="binomial",correlation="ID")
cor(Data$X[,1:5])
print(Data)
```

---

**plot.selection**

*Plots to visualize the selection*

**Description**

This function constructs a sparsity vs. selection criterion curve for a selection object. When EBIC is used with voting, it also constructs a histogram showing the voting result.

**Usage**

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

**Arguments**

- `x`: Fitted 'selection' object from `smle_select`.
- `...`: Additional arguments to the plot function.
Value

No return value, called for side effects.

Examples

```r
Data <- Gen_Data(correlation = "MA", family = "gaussian")
fit <- SMLE(Data$Y, Data$X, k = 20, family = "gaussian")
E <- smle_select(fit)
# Then E is a object of "selection"
plot(E)
```

---

## plot.smle

Plots to visualize the SMLE screening step

### Description

This function returns two plot windows. By default, the first contains 4 plots to assess: 1) log-likelihood, 2) Euclidean distance between the current and the previous coefficient estimates, 3) the number of tries in tuning parameter "u" in IHT algorithm (see "Ucheck" in SMLE), and 4) the number of features changed in the current active set. By default, the second plot shows the solution path (estimated coefficient by iteration step) for selected features.

### Usage

```r
## S3 method for class 'smle'
plot(
  x, 
  Display = c("top_row", "all"),
  num_path = NULL,
  which_path = NULL,
  out_plot = 5,
  ...
)
```

### Arguments

- **x**: Fitted "smle" object from SMLE.
- **Display**: For the solution path plot, show path for the most significant coefficients("top_row") or for all coefficients("all").
- **num_path**: Number of top coefficients to be shown in solution path plot if type = "top_row". Default in solution path plot is 5.
- **which_path**: A vector to control which features are shown in addition to the paths for the most significant coefficients if type = "top_row".
- **out_plot**: A number from 1 to 5 indicating which plot is to be shown in the separate window; the default for solution path plot is "5". See Description for plot labels 1-4.
- **...**: Additional arguments to the plot function.
**Value**

No return value, called for side effects.

**Examples**

```r
Data<-Gen_Data(correlation="MA",family = "gaussian")
fit<-SMLE(Data$Y,Data$X,k=20,family = "gaussian")
plot(fit)
```

---

**predict.smle** Prediction based on SMLE screening and selection

**Description**

Similar to the usual predict methods, this function returns predicted mean values of the response based on the features retained in `smle` object or selected by `selection` object.

**Usage**

```r
## S3 method for class 'smle'
predict(object, newdata = NULL, type = c("link", "response"), ...)
## S3 method for class 'selection'
predict(object, newdata = NULL, type = c("link", "response"), ...)
```

**Arguments**

- `object`:
  A fitted object of class `smle`, as the output from SMLE; or `selection` as the output from smle_select.

- `newdata`:
  Matrix of new values for x at which predictions are to be made, without the intercept term. If omitted, the fitted linear features are used.

- `type`:
  Type of prediction required. "response" gives fitted values for "gaussian"; fitted probabilities for 'binomial', fitted mean for 'poisson'. "link" returns prediction on the scale of the linear predictors. (Same to "response" in "gaussian" models)

- `...`:
  Further arguments pass to predict.glm().

**Value**

Returns a vector of the predicted mean values of the response based on `newdata` and the features retained in `object`. The predicted values depend on the model specified in `type`.
Examples

set.seed(123.456)
Data_sim<-Gen_Data(n= 200, p =1000, correlation="AR",family = "gaussian")
fit<-SMLE(Data_sim$Y,Data_sim$X, family = "gaussian")
predict(fit, type ="link")

E<-smle_select(fit, tune="ebic")
predict(E, type ="link")

print.sdata

Print function for simulated data

Description

This functions prints a summary of a data set generated by Gen_data.
In particular, it prints the indices of relevant features, true model coefficients, and the correlation structure.

Usage

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

Arguments

x "sdata" object from Gen_Data function.
...
This argument is not used and listed for method consistency.

Value

No return value, called for side effects.

Examples

Data<-Gen_Data(family ="binomial",correlation = "ID")
cor(Data$X[,1:10])
print(Data)
**print.selection**

Print a selection object from smle_select

---

**Description**

This function prints a summary of a 'selection' object. In particular, it gives the selected features along with their re-fitted model coefficients. For reference, it also shows the values of the selection criterion used in selection for all candidate models.

**Usage**

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

**Arguments**

- `x` Fitted 'selection' object.
- `...` This argument is not used and listed for method consistency.

**Value**

No return value, called for side effects.

**Examples**

```r
Data<-Gen_Data(correlation="MA",family = "gaussian")
fit<-SMLE(Data$Y,Data$X,k=20,family = "gaussian")
E<-smle_select(fit)
print(E)
```

---

**print.smle**

Print a SMLE object from SMLE

---

**Description**

This functions prints a summary of a SMLE object. In particular, it shows the features retained after SMLE-screening and the related convergence information.

**Usage**

```r
## S3 method for class 'smle'
print(x, ...)
```
Arguments

x Fitted `smle` object.

... This argument is not used and listed for method consistency.

Value

No return value, called for side effects.

Examples

Data<-Gen_Data(correlation="MA",family = "gaussian")
fit<-SMLE(Data$Y,Data$X,k=20,family = "gaussian")
print(fit)

Description

Input a $n \times 1$ response $Y$ and a $n \times p$ feature matrix $X$; the function uses SMLE to retain only a set of $k < n$ features that seem to be most relevant for a GLM. It thus serves as a pre-processing step for an elaborative analysis. In SMLE, the joint effects between features are naturally accounted; this makes the screening more reliable. The function uses the efficient iterative hard thresholding (IHT) algorithm with step parameter adaptively tuned for fast convergence. Users can choose to further conduct an elaborative selection after SMLE-screening. See `smle_select` for more details.

Usage

SMLE(
  Y,
  X,
  k = NULL,
  family = c("gaussian", "binomial", "poisson"),
  categorical = NULL,
  keyset = NULL,
  intercept = TRUE,
  group = TRUE,
  codingtype = NULL,
  maxit = 50,
  tol = 10^(-2),
  selection = F,
  standardize = TRUE,
  fast = FALSE,
  U_rate = 0.5,
  penalize_mod = TRUE
)
Arguments

Y
The response vector of dimension $n \times 1$. Quantitative for family = 'gaussian',
non-negative counts for family = 'poisson', binary (0-1) for family = 'binomial'.
Input Y should be 'numeric'.

X
The $n \times p$ feature matrix with each column denoting a feature (covariate) and
each row denoting an observation vector. The input should be the object of
"matrix" for numerical data, and "data.frame" for categorical data (or a mixture
of numerical and categorical data). The algorithm will treat covariates having
class "factor" as categorical data and extend the data frame dimension by the
dummy columns needed for coding the categorical features.

k
Total number of features (including 'keyset') to be retained after screening. De-
fault is $\frac{1}{2} \log(n)n^{1/3}$.

family
Model assumption between Y and X; the default model is Gaussian linear.

categorical
Logical flag whether the input feature matrix includes categorical features. If
categorical = TRUE, a model intercept will be used in the screening process.
Default is NULL.

keyset
A vector to indicate a set of key features that do not participate in feature screen-
ing and are forced to remain in the model. Default is null.

intercept
A vector to indicate whether to an intercept be used in the model. An intercept
will not participate in screening.

group
Logical flag for whether to treat the dummy covariates of a categorical feature
as a group. (Only for categorical data, see details). Default is TRUE.

codingtype
Coding types for categorical features; default is "DV". Codingtype = "all"
Convert each level to a 0-1 vector. Codingtype = "DV" conducts deviation cod-
ing for each level in comparison with the grand mean. Codingtype = "standard"
conducts standard dummy coding for each level in comparison with the refer-
ence level (first level).

maxit
Maximum number of iteration steps. Default is 500. Set maxit = NULL to loosen
this protective stopping criterion.

tol
A tolerance level to stop the iteration, when the squared sum of differences
between two successive coefficient updates is below it. Default is $10^{-2}$. Set
tol = NULL to loosen this stopping criterion.

selection
A logical flag to indicate whether an elaborate selection is to be conducted by
smle_select after screening (Using default arguments). Default is FALSE.

standardize
Logical flag for feature standardization, prior to performing (iterative) feature
screening. The resulting coefficients are always returned on the original scale.
Default is standardize=TRUE. If features are in the same units already, you
might not wish to standardize.

fast
Set to TRUE to enable early stop for SMLE-screening. It may help to boost the
screening efficiency with a little sacrifice of accuracy. Default is FALSE, see
details.

U_rate
Decreasing rate in tuning step parameter $u^{-1}$ in IHT algorithm. See details.
penalize_mod  A logical flag to indicate whether adjustment is used in ranking groups of features. This augment is applicable only when categorical= TRUE with group=T; the default is true: a factor of $\sqrt{J}$ is divided from the $L_2$ effect of a group with J members.

Details

With the input Y and X, SMLE conducts joint feature screening by running iterative hard thresholding algorithm (IHT), where the initial value is set to be the Lasso estimate with the sparsity closest to the sample size minus one.

In SMLE, the step parameter $u^{-1}$ in IHT is adaptively tuned in the same way as described in Xu and Chen (2014). Specifically, each step, we set the initial u as the max row sum of X and recursively decrease the value of $u^{-1}$ by U_rate to guarantee the likelihood increment.

SMLE terminates IHT iterations when either tol or maxit is satisfied. When fast=TRUE, the algorithm also stops when the non-zero members of the coefficient estimates remain the same for 10 successive iterations.

In SMLE, categorical features are coded by dummy covariates with the method specified in codingtype. Users can use group to specify whether to treat those dummy covariates as a single group feature or as individual features. When group=TRUE with penalize_mod=TRUE, the effect for a group of J dummy covariates is computed by

$$\beta_i = \frac{1}{\sqrt{J}} \cdot \sqrt{(\beta_1)^2 + \ldots + (\beta_J)^2}$$

which will be treated as a single feature in IHT iterations.

Since feature screening is usually a preprocessing step, users may wish to further conduct an elaborate feature selection after screening. This can be done by setting selection=TRUE in SMLE or applying any existing selection method on the output of SMLE.

Value

Returns a ‘smle’ object with

- I: A list of iteration information.
- Y: Same as input Y.
- CM: Design matrix of class matrix for numeric features (or data.frame with categorical features).
- DM: A matrix with dummy variable features added. (only if there are categorical features).
- IM: Iteration path matrix with columns recording IHT coefficient updates.
- nlevel: Number of levels for all categorical features.
- CI: Indices of categorical features in CM.
- Beta0: Initial value of regression coefficient for IHT.
- DFI: Indices of categorical features in IM.
- codingtype: Same as input.
- ID_Retained: A vector indicating the features retained after SMLE screening. The output includes both features retained by SMLE and the features specified in keyset.
Coef_Retained  The vector of coefficients for the retained features.
Path_Retained  Iteration path matrix with columns recording the coefficient updates over the IHT procedure.
Num_Retained  Number of retained features after screening.
Intercept  The value, if Intercept = TRUE.
steps  Number of iterations.
LH  A list of log-likelihood updates over the IHT iterations
Uchecks  Number of times in searching a proper $u^{-1}$ at each step over the IHT iterations.

References


Examples

```r
#Example
set.seed(123.456)
Data<-Gen_Data(n=100, p=5000, family = "gaussian", correlation="ID")
Data
fit<-SMLE(Data$Y, Data$X, k=9, family = "gaussian")
fit
## The important features we missed:
setdiff(Data$index,fit$ID_Retained)
## Check if the important features are retained.
Data$index %in% fit$ID_Retained
plot(fit)
```

Description

Given a response and a set of $K$ features, this function first runs SMLE (fast=TRUE) to generate a series of sub-models with sparsity $k$ varying from $k_{\text{min}}$ to $k_{\text{max}}$. It then selects the best model from the series based on a selection criterion. When criterion EBIC is used, users can choose to repeat the selection with different values of the tuning parameter, $\gamma$, and conduct importance voting for each feature.
Usage

smle_select(x, ...)

## S3 method for class 'smle'
smle_select(x, ...)

## S3 method for class 'sdata'
smle_select(
  x,
  k_min = 1,
  k_max = 10,
  sub_model = NULL,
  gamma_ebic = 0.5,
  vote = FALSE,
  tune = "ebic",
  codingtype = NULL,
  gamma_seq = c(seq(0, 1, 0.2)),
  vote_threshold = NULL,
  para = FALSE,
  num_cores = NULL,
  ...
)

## Default S3 method:
smle_select(x, X = NULL, family = "gaussian", ...)

Arguments

x Object of class 'smle' or 'sdata'. Users can also input a response vector and a feature matrix. See examples

... Further arguments passed to or from other methods.

k_min The lower bound of candidate model sparsity. Default is 1.

k_max The upper bound of candidate model sparsity. Default is as same as the number of columns in input.

sub_model A index vector indicating which features (columns of the feature matrix) are to be selected. Not applicable if a 'smle' object is the input.

gamma_ebic The EBIC parameter in [0, 1]. Default is 0.5.

vote The logical flag for whether to perform the voting procedure. Only available when tune = 'ebic'.

tune Selection criterion. Default is ebic.

codingtype Coding types for categorical features; details see SMLE.

gamma_seq The sequence of values for gamma_ebic when vote = TRUE.

vote_threshold A relative voting threshold in percentage. A feature is considered to be important when it receives votes passing the threshold.
**smle_select**

para Logical flag to use parallel computing to do voting selection. Default is FALSE. See Details.

num_cores The number of cores to use. The default will be all cores detected.

X Input features matrix. When feature matrix input by users.

family Model assumption; see SMLE. Default is Gaussian linear. When input is 'smle' or 'sdata', the same model will be used in the selection.

**Details**

This function accepts three types of input for GLMdata; 1. 'smle' object, as the output from SMLE; 2. 'sdata' object, as the output from Gen_Data; 3. Other response and feature matrix input by users.

Note that this function is mainly design to conduct an elaborative selection after feature screening. We do not recommend using it directly for ultra-high-dimensional data without screening.

**Value**

Returns a 'selection' object with

- **ID_Selected** A list of selected features.
- **Coef_Selected** Fitted model coefficients based on the selected features.
- **Criterion_value** Values of selection criterion for the candidate models with various sparsity.
- **ID_Voted** A list of Voting selection results; item returned only when vote==T.

**References**


**Examples**

```r
# This a simple example for Gaussian assumption.
Data<-Gen_Data(correlation="MA",family = "gaussian")
fit<-SMLE(Data$Y,Data$X,k=20,family = "gaussian")
E<-smle_select(fit)
plot(E)
```
Index

Gen_Data, 3
plot.selection, 5
plot.smle, 6
predict.selection(predict.smle), 7
predict.smle, 7
print.sdata, 8
print.selection, 9
print.smle, 9

SMLE, 10
smle-package, 2
smle_select, 13