Package ‘SAM’

February 19, 2015

Type Package
Title Sparse Additive Modelling
Version 1.0.5
Date 2014-02-11
Author Tuo Zhao, Xingguo Li, Han Liu, and Kathryn Roeder
Maintainer Tuo Zhao <tourzhao@gmail.com>
Depends R (>= 2.14), splines
Description The package SAM targets at high dimensional predictive modeling (regression and classification) for complex data analysis. SAM is short for sparse additive modeling, and adopts the computationally efficient basis spline technique. We solve the optimization problems by various computational algorithms including the block coordinate descent algorithm, fast iterative soft-thresholding algorithm, and newton method. The computation is further accelerated by warm-start and active-set tricks.
License GPL-2
Repository CRAN
Date/Publication 2014-02-12 09:34:38
NeedsCompilation yes

R topics documented:

SAM-package ......................................................... 2
plot.samEL .......................................................... 3
plot.samHL .......................................................... 4
plot.samLL .......................................................... 5
plot.samQL .......................................................... 6
predict.samEL ....................................................... 7
predict.samHL ....................................................... 8
predict.samLL ....................................................... 9
predict.samQL ....................................................... 10
Description

The package SAM targets at high dimensional predictive modeling (regression and classification) for complex data analysis. SAM is short for sparse additive modeling, and adopts the computationally efficient basis spline technique. We solve the optimization problems by various computational algorithms including the block coordinate descent algorithm, fast iterative soft-thresholding algorithm, and newton method. The computation is further accelerated by warm-start and active-set tricks.

Details

Package: SAM
Type: Package
Version: 1.0.5
Date: 2014-02-11
License: GPL-2

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, and Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References

plot.samEL

See Also

samQL, samHL, samLL, samEL

---

plot.samEL

Plot function for S3 class "samEL"

Description

This function plots the regularization path (regularization parameter versus functional norm)

Usage

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

Arguments

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

Details

The horizontal axis is for the regularization parameters in log scale. The vertical axis is for the functional norm of each component.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

samEL
Description

This function plots the regularization path (regularization parameter versus functional norm)

Usage

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

Arguments

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

Details

The horizontal axis is for the regularization parameters in log scale. The vertical axis is for the functional norm of each component.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

`samHL`
**Description**

This function plots the regularization path (regularization parameter versus functional norm).

**Usage**

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

**Arguments**

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

**Details**

The horizontal axis is for the regularization parameters in log scale. The vertical axis is for the functional norm of each component.

**Author(s)**

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

**References**


**See Also**

samLL
plot.samQL

Plot function for S3 class "samQL"

Description

This function plots the regularization path (regularization parameter versus functional norm).

Usage

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

Arguments

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

Details

The horizontal axis is for the regularization parameters in log scale. The vertical axis is for the functional norm of each component.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

samQL
predict.samEL

Description

Predict the labels for testing data.

Usage

```r
## S3 method for class 'samEL'
predict(object, newdata, ...)
```

Arguments

- `object`: An object with S3 class "samEL".
- `newdata`: The testing dataset represented in a \( n \) by \( d \) matrix, where \( n \) is testing sample size and \( d \) is dimension.
- `...`: System reserved. (No specific usage)

Details

The testing dataset is rescaled to the samEL range, and expanded by the samEL spline basis functions as the training data.

Value

- `expectations`: Estimated expected counts also represented in a \( n \) by the length of \( \lambda \) matrix, where \( n \) is testing sample size.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

- `samEL`
predict.samHL

Prediction function for S3 class "samHL"

Description

Predict the labels for testing data.

Usage

```r
## S3 method for class 'samHL'
predict(object, newdata, thol = 0, ...)
```

Arguments

- `object`: An object with S3 class "samHL".
- `newdata`: The testing dataset represented in a \( n \times d \) matrix, where \( n \) is testing sample size and \( d \) is dimension.
- `thol`: The decision value threshold for prediction. The default value is 0.
- `...`: System reserved. (No specific usage)

Details

The testing dataset is rescaled to the samHL range, and expanded by the samHL spline basis functions as the training data.

Value

- `values`: Predicted decision values also represented in a \( n \) by the length of \( \lambda \) matrix, where \( n \) is testing sample size.
- `labels`: Predicted labels also represented in a \( n \) by the length of \( \lambda \) matrix, where \( n \) is testing sample size.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder

Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

- `samHL`
**predict.samLL**

**Prediction function for S3 class "samLL"**

---

**Description**

Predict the labels for testing data.

**Usage**

```r
## S3 method for class 'samLL'
predict(object, newdata, thol = 0.5, ...)
```

**Arguments**

- `object`: An object with S3 class "samLL".
- `newdata`: The testing dataset represented in a \( n \) by \( d \) matrix, where \( n \) is testing sample size and \( d \) is dimension.
- `thol`: The decision probability threshold for prediction. The default value is 0.5.
- `...`: System reserved. (No specific usage)

**Details**

The testing dataset is rescale to the same range, and expanded by the same spline basis functions as the training data.

**Value**

- `probs`: Estimated Posterior Probability for Prediction also represented in a \( n \) by the length of \( \lambda \) matrix, where \( n \) is testing sample size.
- `labels`: Predicted labels also represented in a \( n \) by the length of \( \lambda \) matrix, where \( n \) is testing sample size.

**Author(s)**

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder

Maintainers: Tuo Zhao<tourzhao@gmail.com>

**References**


**See Also**

`samLL`
Description

Predict the responses for testing data.

Usage

## S3 method for class 'samQL'
predict(object, newdata, ...)

Arguments

- **object**: An object with S3 class "samQL".
- **newdata**: The testing dataset represented in a \(n\) by \(d\) matrix, where \(n\) is testing sample size and \(d\) is dimension.
- **...**: System reserved. (No specific usage)

Details

The testing dataset is rescale to the samQLe range, and expanded by the samQLe spline basis functions as the training data.

Value

- **values**: Predicted values also represented in a \(n\) by the length of \(\lambda\) matrix, where \(n\) is testing sample size.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

samQL
print.samEL

print.samEL  

Description

Summarize the information of the object with S3 class samEL.

Usage

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

Arguments

x  An object with S3 class "samEL"

...  System reserved (No specific usage)

Details

The output includes length and d.f. of the regularization path.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

samEL
print.samHL

Printing function for S3 class samHL

Description
Summarize the information of the object with S3 class samHL.

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

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

Details
The output includes length and d.f. of the regularization path.

Author(s)
Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References

See Also
samHL
print.samLL

Description

Summarize the information of the object with S3 class samLL.

Usage

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

Arguments

x          An object with S3 class "samLL"
...

Details

The output includes length and d.f. of the regularization path.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

samLL
**print.samQL**

---

**printing function for S3 class samQL**

---

**Description**

Summarize the information of the object with S3 class `samQL`.

**Usage**

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

**Arguments**

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

**Details**

The output includes length and d.f. of the regularization path.

**Author(s)**

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder

Maintainers: Tuo Zhao<tourzhao@gmail.com>

**References**


**See Also**

`samQL`
Training function of Sparse Additive Possion Regression

Description

The log-linear model is learned using training data.

Usage

```r
samEL(X, y, p=3, lambda = NULL, nlambda = NULL,
       lambda.min.ratio = 0.25, thol=1e-5, max.ite = 1e5)
```

Arguments

- `X` The n by d design matrix of the training set, where n is sample size and d is dimension.
- `y` The n-dimensional response vector of the training set, where n is sample size. Responses must be non-negative integers.
- `p` The number of basis spline functions. The default value is 3.
- `lambda` A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Do not supply a single value for lambda. Supply instead a decreasing sequence of lambda values. samEL relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.
- `nlambda` The number of lambda values. The default value is 20.
- `lambda.min.ratio` Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default is 0.1.
- `thol` Stopping precision. The default value is 1e-5.
- `max.ite` The number of maximum iterations. The default value is 1e5.

Details

We adopt various computational algorithms including the block coordinate descent, fast iterative soft-thresholding algorithm, and newton method. The computation is further accelerated by "warm-start" and "active-set" tricks.

Value

- `p` The number of basis spline functions used in training.
- `X.min` A vector with each entry corresponding to the minimum of each input variable. (Used for rescaling in testing)
A vector with each entry corresponding to the range of each input variable.
(Used for rescaling in testing)

A sequence of regularization parameter used in training.

The solution path matrix \((d*p+1 \text{ by length of } \lambda)\) with each column corresponding to a regularization parameter. Since we use the basis expansion with the intercept, the length of each column is \(d*p+1\).

The degree of freedom of the solution path (The number of non-zero component function)

The \(p-1\) by \(d\) matrix. Each column contains the knots applied to the corresponding variable.

The \(2\) by \(d\) matrix. Each column contains the boundary points applied to the corresponding variable.

The functional norm matrix \((d \text{ by length of } \lambda)\) with each column corresponds to a regularization parameter. Since we have \(d\) input variables, the length of each column is \(d\).

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

`SAM`, `plot.samEL`, `print.samEL`, `predict.samEL`

Examples

```r
## generating training data
n = 200
d = 100
X = 0.5*matrix(runif(n*d),n,d) + matrix(rep(0.5*runif(n),d),n,d)
u = exp(-2*sin(X[,1])+X[,2]^2-1/3 + X[,3]-1/2 + exp(-X[,4])+exp(-1)-1+1)
y = rep(0,n)
for(i in 1:n) y[i] = rpois(1,u[i])

## Training
out.trn = samEL(X,y)
out.trn

## plotting solution path
```
## generating testing data

```r
nt = 1000
Xt = 0.5*matrix(runif(nt*d),nt,d) + matrix(rep(0.5*runif(nt),d),nt,d)
Ut = exp(-2*sin(Xt[,1])) + Xt[,2]^2-1/3 + Xt[,3]-1/2 + exp(-Xt[,4])*exp(-1)-1+1
Yt = rep(0,nt)
for(i in 1:nt) Yt[i] = rpois(1,Ut[i])
```

## predicting response

```r
out.tst = predict(out.trn,Xt)
```

---

**samHL**  
*Training function of Sparse Additive Machine*

### Description

The classifier is learned using training data.

### Usage

```r
samHL(X, y, p=3, lambda = NULL, nlambda = NULL, lambda.min.ratio = 0.4, thol=1e-5, mu = 5e-2, max.ite = 1e5)
```

### Arguments

- **X**: The n by d design matrix of the training set, where n is sample size and d is dimension.
- **y**: The n-dimensional label vector of the training set, where n is sample size. Labels must be coded in 1 and 0.
- **p**: The number of basis spline functions. The default value is 3.
- **lambda**: A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Do not supply a single value for lambda. Supply instead a decreasing sequence of lambda values. samHL relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.
- **nlambda**: The number of lambda values. The default value is 20.
- **lambda.min.ratio**: Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default is 0.4.
- **thol**: Stopping precision. The default value is 1e-5.
- **mu**: Smoothing parameter used in approximate the Hinge Loss. The default value is 0.05.
- **max.ite**: The number of maximum iterations. The default value is 1e5.
Details

We adopt various computational algorithms including the block coordinate descent, fast iterative soft-thresholding algorithm, and newton method. The computation is further accelerated by "warm-start" and "active-set" tricks.

Value

- $p$: The number of baisis spline functions used in training.
- $X_{min}$: A vector with each entry corresponding to the minimum of each input variable. (Used for rescaling in testing)
- $X_{ran}$: A vector with each entry corresponding to the range of each input variable. (Used for rescaling in testing)
- $\lambda$: A sequence of regularization parameter used in training.
- $w$: The solution path matrix ($d*p+1$ by length of $\lambda$) with each column corresponding to a regularization parameter. Since we use the basis expansion with the intercept, the length of each column is $d*p+1$.
- $df$: The degree of freedom of the solution path (The number of non-zero component function)
- $knots$: The $p-1$ by $d$ matrix. Each column contains the knots applied to the corresponding variable.
- $Boundary.knots$: The 2 by $d$ matrix. Each column contains the boundary points applied to the corresponding variable.
- $func\_norm$: The functional norm matrix ($d$ by length of $\lambda$) with each column corresponds to a regularization parameter. Since we have $d$ input variables, the length of each column is $d$.

Author(s)

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References


See Also

SAM,plot.samHL,print.samHL,predict.samHL
Examples

```r
## generating training data
n = 200
d = 100
X = 0.5*matrix(runif(n*d), n, d) + matrix(rep(0.5*runif(n), d), n, d)
y = sign(((X[,1]-0.5)^2 + (X[,2]-0.5)^2)-0.06)

## flipping about 5 percent of y
y = y*sign(runif(n)-0.05)

## Training
out.trn = samHL(X, y)

## plotting solution path
plot(out.trn)

## generating testing data
nt = 1000
Xt = 0.5*matrix(runif(nt*d), nt, d) + matrix(rep(0.5*runif(nt), d), nt, d)
yt = sign(((Xt[,1]-0.5)^2 + (Xt[,2]-0.5)^2)-0.06)

## flipping about 5 percent of y
yt = yt*sign(runif(nt)-0.05)

## predicting response
out.tst = predict(out.trn, Xt)
```

### Description

The logistic model is learned using training data.

### Usage

```r
samHL(X, y, p=3, lambda = NULL, nlambda = NULL,
lambda.min.ratio = 0.1, thol=1e-5, max.ite = 1e5)
```

### Arguments

- **X**: The n by d design matrix of the training set, where n is sample size and d is dimension.
- **y**: The n-dimensional label vector of the training set, where n is sample size. Labels must be coded in 1 and 0.
The number of basis spline functions. The default value is 3.

A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. WARNING: use with care. Do not supply a single value for lambda. Supply instead a decreasing sequence of lambda values. samLL relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.

The number of lambda values. The default value is 20.

Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default is 0.1.

Stopping precision. The default value is 1e-5.

The number of maximum iterations. The default value is 1e5.

We adopt various computational algorithms including the block coordinate descent, fast iterative soft-thresholding algorithm, and newton method. The computation is further accelerated by "warm-start" and "active-set" tricks.

The number of basis spline functions used in training.

A vector with each entry corresponding to the minimum of each input variable. (Used for rescaling in testing)

A vector with each entry corresponding to the range of each input variable. (Used for rescaling in testing)

A sequence of regularization parameter used in training.

The solution path matrix (d*p+1 by length of lambda) with each column corresponding to a regularization parameter. Since we use the basis expansion with the intercept, the length of each column is d*p+1.

The degree of freedom of the solution path (The number of non-zero component function)

The p-1 by d matrix. Each column contains the knots applied to the corresponding variable.

The 2 by d matrix. Each column contains the boundary points applied to the corresponding variable.

The functional norm matrix (d by length of lambda) with each column corresponds to a regularization parameter. Since we have d input variables, the length of each column is d.

Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>
References


See Also

SAM, plot.samLL, print.samLL, predict.samLL

Examples

```r
## generating training data
n = 200
d = 100
X = 0.5*matrix(runif(n*d), n, d) + matrix(rep(0.5*runif(n), d), n, d)
y = sign(((X[,1]-0.5)^2 + (X[,2]-0.5)^2)-0.06)

## flipping about 5 percent of y
y = y*sign(runif(n)-0.05)
y = sign(y==1)

## Training
out.trn = samLL(X, y)
out.trn

## plotting solution path
plot(out.trn)

## generating testing data
nt = 1000
Xt = 0.5*matrix(runif(nt*d), nt, d) + matrix(rep(0.5*runif(nt), d), nt, d)

yt = sign(((Xt[,1]-0.5)^2 + (Xt[,2]-0.5)^2)-0.06)

## flipping about 5 percent of y
yt = yt*sign(runif(nt)-0.05)
yt = sign(yt==1)

## predicting response
out.tst = predict(out.trn, Xt)
```

samQL

Training function of Sparse Additive Models

Description

The regression model is learned using training data.
Usage

```r
samQL(X, y, p = 3, lambda = NULL, nlambda = NULL,
   lambda.min.ratio = 5e-3, thol = 1e-05, max.ite = 1e5)
```

Arguments

- **X**: The n by d design matrix of the training set, where n is sample size and d is dimension.
- **y**: The n-dimensional response vector of the training set, where n is sample size.
- **p**: The number of basis spline functions. The default value is 3.
- **lambda**: A user supplied lambda sequence. Typical usage is to have the program compute its own lambda sequence based on nlambda and lambda.min.ratio. Supplying a value of lambda overrides this. **WARNING**: use with care. Do not supply a single value for lambda. Supply instead a decreasing sequence of lambda values. samQL relies on its warms starts for speed, and its often faster to fit a whole path than compute a single fit.
- **nlambda**: The number of lambda values. The default value is 30.
- **lambda.min.ratio**: Smallest value for lambda, as a fraction of lambda.max, the (data derived) entry value (i.e. the smallest value for which all coefficients are zero). The default is 5e-3.
- **thol**: Stopping precision. The default value is 1e-5.
- **max.ite**: The number of maximum iterations. The default value is 1e5.

Details

We adopt various computational algorithms including the block coordinate descent, fast iterative soft-thresholding algorithm, and newton method. The computation is further accelerated by "warm-start" and "active-set" tricks.

Value

- **p**: The number of basis spline functions used in training.
- **X.min**: A vector with each entry corresponding to the minimum of each input variable. (Used for rescaling in testing)
- **X.ran**: A vector with each entry corresponding to the range of each input variable. (Used for rescaling in testing)
- **lambda**: A sequence of regularization parameter used in training.
- **w**: The solution path matrix (d*p by length of lambda) with each column corresponding to a regularization parameter. Since we use the basis expansion, the length of each column is d*p.
- **intercept**: The solution path of the intercept.
- **df**: The degree of freedom of the solution path (The number of non-zero component function)
knots The p-1 by d matrix. Each column contains the knots applied to the corresponding variable.

Boundary.knots The 2 by d matrix. Each column contains the boundary points applied to the corresponding variable.

func_norm The functional norm matrix (d by length of lambda) with each column corresponds to a regularization parameter. Since we have d input variables, the length of each column is d.

sse Sums of square errors of the solution path.

Author(s)
Tuo Zhao, Xingguo Li, Han Liu, Kathryn Roeder
Maintainers: Tuo Zhao<tourzhao@gmail.com>

References

See Also
SAM, plot.samQL, print.samQL, predict.samQL

Examples

```r
## generating training data
n = 100
d = 500
X = 0.5*matrix(runif(n*d),n,d) + matrix(rep(0.5*runif(n),d),n,d)

## generating response
y = -2*sin(X[,1]) + X[,2]^2-1/3 + X[,3]-1/2 + exp(-X[,4])+exp(-1)-1

## Training
out.trn = samQL(X,y)
out.trn

## plotting solution path
plot(out.trn)

## generating testing data
nt = 1000
Xt = 0.5*matrix(runif(nt*d),nt,d) + matrix(rep(0.5*runif(nt),d),nt,d)
yt = -2*sin(Xt[,1]) + Xt[,2]^2-1/3 + Xt[,3]-1/2 + exp(-Xt[,4])+exp(-1)-1
```
## predicting response
out.tst = predict(out.trn,Xt)
Index

plot.samEL, 3, 16
plot.samHL, 4, 18
plot.samLL, 5, 21
plot.samQL, 6, 23
predict.samEL, 7, 16
predict.samHL, 8, 18
predict.samLL, 9, 21
predict.samQL, 10, 23
print.samEL, 11, 16
print.samHL, 12, 18
print.samLL, 13, 21
print.samQL, 14, 23

SAM, 16, 18, 21, 23
SAM (SAM-package), 2
SAM-package, 2
samEL, 3, 7, 11, 15
samHL, 3, 4, 8, 12, 17
samLL, 3, 5, 9, 13, 19
samQL, 3, 6, 10, 14, 21