

Package ‘picasso’

March 12, 2026

Type Package

Title Sparse Learning with Convex and Concave Penalties

Version 1.5

Date 2026-02-18

Maintainer Tuo Zhao <tourzhao@gatech.edu>

Depends R (>= 2.15.0), MASS, Matrix

Imports methods

Description Fast tools for fitting sparse generalized linear models with convex penalties (lasso) and concave penalties (smoothly clipped absolute deviation and minimax concave penalty). Computation uses multi-stage convex relaxation and pathwise coordinate optimization with warm starts, active-set updates, and screening rules. Core solvers are implemented in C++, and coefficient paths are stored as sparse matrices for memory efficiency.

License GPL-3

Copyright See inst/COPYRIGHTS for bundled third-party copyright and license notices.

Repository CRAN

NeedsCompilation yes

Author Jason Ge [aut],
Xingguo Li [aut],
Haoming Jiang [aut],
Mengdi Wang [aut],
Tong Zhang [aut],
Han Liu [aut],
Tuo Zhao [aut, cre],
Gael Guennebaud [ctb] (Contributor to bundled Eigen headers),
Benoit Jacob [ctb] (Contributor to bundled Eigen headers),
Eigen Library Authors [cph] (Copyright holders of bundled Eigen headers
in src/include/eigen3)

Date/Publication 2026-03-12 14:00:02 UTC

Contents

picasso-package	2
coef.gaussian	3
coef.logit	3
coef.poisson	4
coef.sqrlasso	5
eyedata	6
picasso	7
plot.gaussian	12
plot.logit	12
plot.poisson	13
plot.sqrlasso	14
predict.gaussian	14
predict.logit	15
predict.poisson	16
predict.sqrlasso	17
print.gaussian	18
print.logit	19
print.poisson	20
print.sqrlasso	21
Index	22

picasso-package	<i>picasso: Sparse Learning with Convex and Non-Convex Penalties</i>
-----------------	--

Description

Computationally efficient tools for fitting sparse generalized linear models with convex or non-convex penalties. Supported penalties include lasso, smoothly clipped absolute deviation (SCAD), and minimax concave penalty (MCP). Computation is based on multi-stage convex relaxation and pathwise coordinate optimization with warm starts, active-set updates, and screening rules.

Details

Core optimization routines are implemented in C++ for speed, and coefficient paths are stored as sparse matrices for memory efficiency.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#).

coef.gaussian	<i>Extract Coefficients from an Object with S3 Class "gaussian"</i>
---------------	---

Description

Extract estimated coefficient vectors from selected points on the regularization path.

Usage

```
## S3 method for class 'gaussian'  
coef(object, lambda.idx = c(1:3), beta.idx = c(1:3), ...)
```

Arguments

object	An object with S3 class "gaussian".
lambda.idx	Indices of regularization parameters in the solution path to extract. The default is c(1:3).
beta.idx	Indices of coefficient entries to extract. The default is c(1:3).
...	Arguments to be passed to methods.

Value

A numeric matrix of extracted coefficients. Columns correspond to lambda.idx. Rows contain the intercept ("(Intercept)") and the selected coefficients ("beta[i]" for i in beta.idx).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

coef.logit	<i>Extract Coefficients from an Object with S3 Class "logit"</i>
------------	--

Description

Extract estimated coefficient vectors from selected points on the regularization path.

Usage

```
## S3 method for class 'logit'  
coef(object, lambda.idx = c(1:3), beta.idx = c(1:3), ...)
```

Arguments

object	An object with S3 class "logit".
lambda.idx	Indices of regularization parameters in the solution path to extract. The default is c(1:3).
beta.idx	Indices of coefficient entries to extract. The default is c(1:3).
...	Arguments to be passed to methods.

Value

A numeric matrix of extracted coefficients. Columns correspond to lambda.idx. Rows contain the intercept ("Intercept") and the selected coefficients ("beta[i]" for i in beta.idx).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

 coef.poisson

Extract Coefficients from an Object with S3 Class "poisson"

Description

Extract estimated coefficient vectors from selected points on the regularization path.

Usage

```
## S3 method for class 'poisson'
coef(object, lambda.idx = c(1:3), beta.idx = c(1:3), ...)
```

Arguments

object	An object with S3 class "poisson".
lambda.idx	Indices of regularization parameters in the solution path to extract. The default is c(1:3).
beta.idx	Indices of coefficient entries to extract. The default is c(1:3).
...	Arguments to be passed to methods.

Value

A numeric matrix of extracted coefficients. Columns correspond to lambda.idx. Rows contain the intercept ("Intercept") and the selected coefficients ("beta[i]" for i in beta.idx).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

coef.sqrlasso *Extract Coefficients from an Object with S3 Class "sqrlasso"*

Description

Extract estimated coefficient vectors from selected points on the regularization path.

Usage

```
## S3 method for class 'sqrlasso'  
coef(object, lambda.idx = c(1:3), beta.idx = c(1:3), ...)
```

Arguments

object	An object with S3 class "sqrlasso".
lambda.idx	Indices of regularization parameters in the solution path to extract. The default is c(1:3).
beta.idx	Indices of coefficient entries to extract. The default is c(1:3).
...	Arguments to be passed to methods.

Value

A numeric matrix of extracted coefficients. Columns correspond to lambda.idx. Rows contain the intercept ("Intercept") and the selected coefficients ("beta[i]" for i in beta.idx).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

eyedata	<i>Bardet-Biedl Syndrome Gene Expression Data from Scheetz et al. (2006)</i>
---------	--

Description

Gene expression data (200 genes, 120 samples) from microarray experiments on mammalian eye tissue reported by Scheetz et al. (2006).

Usage

```
data(eyedata)
```

Format

A list with components:

x A 120 by 200 matrix of gene probes.

y A numeric response vector of length 120 representing the expression level of the TRIM32 gene.

Details

This data set contains 120 samples and 200 predictors.

Author(s)

Xingguo Li, Tuo Zhao, Tong Zhang and Han Liu
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

References

1. T. Scheetz, K. Kim, R. Swiderski, A. Philp, T. Braun, K. Knudtson, A. Dorrance, G. DiBona, J. Huang, T. Casavant, V. Sheffield, and 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

[picasso-package](#).

Examples

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

Description

picasso fits regularization paths for sparse Gaussian, logistic (family = "binomial"), Poisson, and sqrt-lasso models using lasso, MCP, or SCAD penalties.

Usage

```
picasso(X, Y, lambda = NULL, nlambda = 100, lambda.min.ratio =
      0.05, family = "gaussian", method = "l1",
      type.gaussian = "naive", gamma = 3, df = NULL,
      dfmax = NULL, standardize = TRUE, intercept = TRUE,
      prec = 1e-07, max.ite = 1000, verbose = FALSE)
```

Arguments

X	Numeric design matrix with n rows (samples) and d columns (features).
Y	Response vector of length n . Use numeric responses for family = "gaussian" and family = "sqrtlasso", a two-level response for family = "binomial", and non-negative integer counts for family = "poisson".
lambda	Optional sequence of decreasing regularization values. If NULL, the sequence is generated automatically from nlambda and lambda.min.ratio.
nlambda	Number of regularization values to generate when lambda = NULL. The default is 100.
lambda.min.ratio	Smallest generated lambda value as a fraction of λ_{\max} when lambda = NULL. The default is 0.05. For non-convex penalties in logistic and Poisson models, values much smaller than 0.05 may lead to ill-conditioning.
family	Model family. Supported values are "gaussian", "binomial", "poisson", and "sqrtlasso". The default is "gaussian".
method	Penalty type. Supported values are "l1" (lasso), "mcp", and "scad". The default is "l1".
type.gaussian	Gaussian solver mode. "naive" maintains residuals and updates them in $O(n)$ per coordinate; "covariance" pre-computes the Gram matrix and updates gradients in $O(d)$ per coordinate, which is faster when $n \gg d$. The default is "naive".
gamma	Concavity parameter for MCP and SCAD penalties. The default is 3.
df	Reserved for backward compatibility with older Gaussian covariance-update implementations. It is currently not used by the active solver backend.
dfmax	Maximum number of nonzero coefficients for early stopping on the lambda path. When the number of nonzero coefficients exceeds dfmax, the solver stops and returns only the solutions computed so far. If NULL (default), no limit is imposed.

In addition to this user-controlled limit, the solver automatically stops when the deviance ratio exceeds 0.999 or the relative deviance change falls below $1e-5$ (checked after at least 5 lambda values).

<code>standardize</code>	If TRUE, standardize columns of X before fitting. The default is TRUE.
<code>intercept</code>	If TRUE, include an intercept term. The default is TRUE.
<code>prec</code>	Convergence tolerance. The default is $1e-7$.
<code>max.ite</code>	Maximum number of iterations. The default is 1000.
<code>verbose</code>	If TRUE, print tracing information during fitting. The default is FALSE.

Details

When `lambda` is not supplied, `picasso` constructs a logarithmically spaced path between λ_{\max} and $\lambda_{\min} = \text{lambda.min.ratio} \cdot \lambda_{\max}$.

The method solves a penalized optimization problem:

$$\min_{\beta_0, \beta} \mathcal{L}(\beta_0, \beta) + \sum_{j=1}^d p_{\lambda, \gamma}(|\beta_j|),$$

where $p_{\lambda, \gamma}$ is lasso, MCP, or SCAD.

Loss functions by family are:

- "gaussian": $\mathcal{L}(\beta_0, \beta) = \frac{1}{2n} \|Y - \beta_0 \mathbf{1} - X\beta\|_2^2$.
- "binomial": $\mathcal{L}(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^n [\log(1 + \exp(\beta_0 + x_i^\top \beta)) - y_i(\beta_0 + x_i^\top \beta)]$.
- "poisson": $\mathcal{L}(\beta_0, \beta) = \frac{1}{n} \sum_{i=1}^n [\exp(\beta_0 + x_i^\top \beta) - y_i(\beta_0 + x_i^\top \beta)]$.
- "sqrtlasso": $\mathcal{L}(\beta_0, \beta) = \frac{1}{\sqrt{n}} \|Y - \beta_0 \mathbf{1} - X\beta\|_2$.

Value

A fitted object of class "gaussian", "logit", "poisson", or "sqrtlasso" depending on family.

Common components include:

beta Sparse matrix of fitted coefficients. Columns correspond to regularization parameters in the solution path.

intercept Intercept values along the solution path.

lambda Regularization sequence used for fitting.

nlambda Number of regularization values.

df Degrees of freedom (number of nonzero coefficients) along the path.

method Penalty type used for fitting.

ite Iteration information returned by the solver.

verbose Input verbose value.

runtime Elapsed fitting time.

family Input family value.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

References

1. J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 2007.
2. C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 2010.
3. J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 2001.
4. R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B*, 2012.
5. J. Ge, X. Li, H. Jiang, M. Wang, T. Zhang, H. Liu, and T. Zhao. PICASSO: A sparse learning library for high-dimensional data analysis in R and Python. *arXiv preprint <https://arxiv.org/abs/2006.15261>*.

See Also

[picasso-package](#).

Examples

```
#####
## Sparse linear regression
## Generate the design matrix and regression coefficient vector
n = 100 # sample number
d = 80 # sample dimension
c = 0.5 # correlation parameter
s = 20 # support size of coefficient
set.seed(2016)
X = scale(matrix(rnorm(n*d),n,d)+c*rnorm(n))/sqrt(n-1)*sqrt(n)
beta = c(runif(s), rep(0, d-s))

## Generate response using Gaussian noise, and fit sparse linear models
noise = rnorm(n)
Y = X%*%beta + noise

## l1 regularization solved with naive update
fitted.l1.naive = picasso(X, Y, nlambda=100, type.gaussian="naive")

## covariance update (faster when n >> d)
fitted.l1.covariance = picasso(X, Y, nlambda=100, type.gaussian="covariance")

## early stopping: stop when more than 10 nonzero coefficients
fitted.l1.dfmax = picasso(X, Y, nlambda=100, dfmax=10)

## mcp regularization
fitted.mcp = picasso(X, Y, nlambda=100, method="mcp")
```

```

## scad regularization
fitted.scad = picasso(X, Y, nlambda=100, method="scad")

## lambdas used
print(fitted.l1.naive$lambda)

## number of nonzero coefficients for each lambda
print(fitted.l1.naive$df)

## coefficients and intercept for the i-th lambda
i = 30
print(fitted.l1.naive$lambda[i])
print(fitted.l1.naive$beta[,i])
print(fitted.l1.naive$intercept[i])

## Visualize the solution path
plot(fitted.l1.naive)
plot(fitted.l1.covariance)
plot(fitted.mcp)
plot(fitted.scad)

#####
## Sparse logistic regression
## Generate the design matrix and regression coefficient vector
n <- 100 # sample number
d <- 80 # sample dimension
c <- 0.5 # parameter controlling the correlation between columns of X
s <- 20 # support size of coefficient
set.seed(2016)
X <- scale(matrix(rnorm(n*d),n,d)+c*rnorm(n))/sqrt(n-1)*sqrt(n)
beta <- c(runif(s), rep(0, d-s))

## Generate response and fit sparse logistic models
p = 1/(1+exp(-X%*%beta))
Y = rbinom(n, rep(1,n), p)

## l1 regularization
fitted.l1 = picasso(X, Y, nlambda=100, family="binomial", method="l1")

## mcp regularization
fitted.mcp = picasso(X, Y, nlambda=100, family="binomial", method="mcp")

## scad regularization
fitted.scad = picasso(X, Y, nlambda=100, family="binomial", method="scad")

## lambdas used
print(fitted.l1$lambda)

## number of nonzero coefficients for each lambda
print(fitted.l1$df)

```

```
## coefficients and intercept for the i-th lambda
i = 30
print(fitted.l1$lambda[i])
print(fitted.l1$beta[,i])
print(fitted.l1$intercept[i])

## Visualize the solution path
plot(fitted.l1)

## Estimate of Bernoulli parameters
param.l1 = fitted.l1$p

#####
## Sparse poisson regression
## Generate the design matrix and regression coefficient vector
n <- 100 # sample number
d <- 80 # sample dimension
c <- 0.5 # parameter controlling the correlation between columns of X
s <- 20 # support size of coefficient
set.seed(2016)
X <- scale(matrix(rnorm(n*d),n,d)+c*rnorm(n))/sqrt(n-1)*sqrt(n)
beta <- c(runif(s), rep(0, d-s))/sqrt(s)

## Generate response and fit sparse poisson models
p = X%%beta+rnorm(n)
Y = rpois(n, exp(p))

## l1 regularization
fitted.l1 = picasso(X, Y, nlambda=100, family="poisson", method="l1")

## mcp regularization
fitted.mcp = picasso(X, Y, nlambda=100, family="poisson", method="mcp")

## scad regularization
fitted.scad = picasso(X, Y, nlambda=100, family="poisson", method="scad")

## lambdas used
print(fitted.l1$lambda)

## number of nonzero coefficients for each lambda
print(fitted.l1$df)

## coefficients and intercept for the i-th lambda
i = 30
print(fitted.l1$lambda[i])
print(fitted.l1$beta[,i])
print(fitted.l1$intercept[i])

## Visualize the solution path
plot(fitted.l1)
```

plot.gaussian	<i>Plot Method for an Object with S3 Class "gaussian"</i>
---------------	---

Description

Plot coefficient paths across regularization parameters.

Usage

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

Arguments

x	An object with S3 class "gaussian".
...	Arguments to be passed to methods.

Value

No return value, called for side effects (drawing the coefficient-path plot).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

plot.logit	<i>Plot Method for an Object with S3 Class "logit"</i>
------------	--

Description

Plot coefficient paths across regularization parameters.

Usage

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

Arguments

x	An object with S3 class "logit".
...	Arguments to be passed to methods.

Value

No return value, called for side effects (drawing the coefficient-path plot).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

plot.poisson

Plot Method for an Object with S3 Class "poisson"

Description

Plot coefficient paths across regularization parameters.

Usage

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

Arguments

x An object with S3 class "poisson".
... Arguments to be passed to methods.

Value

No return value, called for side effects (drawing the coefficient-path plot).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

plot.sqrtlasso *Plot Method for an Object with S3 Class "sqrtlasso"*

Description

Plot coefficient paths across regularization parameters.

Usage

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

Arguments

x An object with S3 class "sqrtlasso".
... Arguments to be passed to methods.

Value

No return value, called for side effects (drawing the coefficient-path plot).

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

predict.gaussian *Prediction Method for an Object with S3 Class "gaussian"*

Description

Predict responses for new data using fitted models.

Usage

```
## S3 method for class 'gaussian'  
predict(object, newdata, lambda.idx = c(1:3), Y.pred.idx = c(1:5), ...)
```

Arguments

object	An object with S3 class "gaussian".
newdata	Optional matrix/data frame used for prediction. If omitted, the training design matrix from fitting is used.
lambda.idx	Indices of regularization parameters along the solution path used for prediction. The default is c(1:3).
Y.pred.idx	Optional row indices to subset returned predictions when explicitly set to a value different from the default c(1:5).
...	Arguments to be passed to methods.

Details

predict.gaussian returns predicted responses for newdata using fitted coefficients from object:

$$\hat{Y} = \hat{\beta}_0 + X_{new}\hat{\beta}.$$

Value

Numeric matrix of predicted responses. Rows correspond to observations in newdata; columns correspond to lambda.idx.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

predict.logit

Prediction Method for an Object with S3 Class "logit"

Description

Predict responses for new data using fitted models.

Usage

```
## S3 method for class 'logit'
predict(object, newdata, lambda.idx = c(1:3), p.pred.idx = c(1:5), ...)
```

Arguments

object	An object with S3 class "logit".
newdata	Optional matrix/data frame used for prediction. If omitted, the training design matrix from fitting is used.
lambda.idx	Indices of regularization parameters along the solution path used for prediction. The default is c(1:3).
p.pred.idx	Optional row indices to subset returned predictions when explicitly set to a value different from the default c(1:5).
...	Arguments to be passed to methods.

Details

predict.logit returns predicted Bernoulli probabilities for newdata using fitted coefficients from object:

$$\hat{p} = \frac{e^{\hat{\beta}_0 + X_{new}\hat{\beta}}}{1 + e^{\hat{\beta}_0 + X_{new}\hat{\beta}}}.$$

Value

Numeric matrix of predicted Bernoulli probabilities. Rows correspond to observations in newdata; columns correspond to lambda.idx.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

predict.poisson *Prediction Method for an Object with S3 Class "poisson"*

Description

Predict responses for new data using fitted models.

Usage

```
## S3 method for class 'poisson'
predict(object, newdata, lambda.idx = c(1:3), p.pred.idx = c(1:5), ...)
```

Arguments

object	An object with S3 class "poisson".
newdata	Optional matrix/data frame used for prediction. If omitted, the training design matrix from fitting is used.
lambda.idx	Indices of regularization parameters along the solution path used for prediction. The default is c(1:3).
p.pred.idx	Optional row indices to subset returned predictions when explicitly set to a value different from the default c(1:5).
...	Arguments to be passed to methods.

Details

predict.poisson returns predicted Poisson means for newdata using fitted coefficients from object:

$$\hat{p} = e^{\hat{\beta}_0 + X_{new}\hat{\beta}}$$

Value

Numeric matrix of predicted Poisson means. Rows correspond to observations in newdata; columns correspond to lambda.idx.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

predict.sqrlasso *Prediction Method for an Object with S3 Class "sqrlasso"*

Description

Predict responses for new data using fitted models.

Usage

```
## S3 method for class 'sqrlasso'
predict(object, newdata, lambda.idx = c(1:3), Y.pred.idx = c(1:5), ...)
```

Arguments

<code>object</code>	An object with S3 class "sqrtlasso".
<code>newdata</code>	Optional matrix/data frame used for prediction. If omitted, the training design matrix from fitting is used.
<code>lambda.idx</code>	Indices of regularization parameters along the solution path used for prediction. The default is <code>c(1:3)</code> .
<code>Y.pred.idx</code>	Optional row indices to subset returned predictions when explicitly set to a value different from the default <code>c(1:5)</code> .
<code>...</code>	Arguments to be passed to methods.

Details

`predict.sqrtlasso` returns predicted responses for `newdata` using fitted coefficients from `object`:

$$\hat{Y} = \hat{\beta}_0 + X_{new}\hat{\beta}.$$

Value

Numeric matrix of predicted responses. Rows correspond to observations in `newdata`; columns correspond to `lambda.idx`.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
 Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

`print.gaussian` *Print Method for an Object with S3 Class "gaussian"*

Description

Print a concise summary for an object with S3 class "gaussian".

Usage

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

Arguments

x An object with S3 class "gaussian".
... Arguments to be passed to methods.

Details

This method prints tuning parameters and fit statistics for a fitted model object.

Value

The input object x, returned invisibly.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

print.logit

Print Method for an Object with S3 Class "logit"

Description

Print a concise summary for an object with S3 class "logit".

Usage

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

Arguments

x An object with S3 class "logit".
... Arguments to be passed to methods.

Details

This method prints tuning parameters and fit statistics for a fitted logistic model object.

Value

The input object x, returned invisibly.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

print.poisson	<i>Print Method for an Object with S3 Class "poisson"</i>
---------------	---

Description

Print a concise summary for an object with S3 class "poisson".

Usage

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

Arguments

x	An object with S3 class "poisson".
...	Arguments to be passed to methods.

Details

This method prints tuning parameters and fit statistics for a fitted Poisson model object.

Value

The input object x, returned invisibly.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

print.sqrtlasso	<i>Print Method for an Object with S3 Class "sqrtlasso"</i>
-----------------	---

Description

Print a concise summary for an object with S3 class "sqrtlasso".

Usage

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

Arguments

x	An object with S3 class "sqrtlasso".
...	Arguments to be passed to methods.

Details

This method prints tuning parameters and fit statistics for a fitted model object.

Value

The input object x, returned invisibly.

Author(s)

Jason Ge, Xingguo Li, Haoming Jiang, Mengdi Wang, Tong Zhang, Han Liu and Tuo Zhao
Maintainer: Tuo Zhao <tourzhao@gatech.edu>

See Also

[picasso](#) and [picasso-package](#).

Index

coef.gaussian, [3](#)
coef.logit, [3](#)
coef.poisson, [4](#)
coef.sqrlasso, [5](#)

eyedata, [6](#)

picasso, [2–5](#), [7](#), [12–21](#)
picasso-package, [2](#)
plot.gaussian, [12](#)
plot.logit, [12](#)
plot.poisson, [13](#)
plot.sqrlasso, [14](#)
predict.gaussian, [14](#)
predict.logit, [15](#)
predict.poisson, [16](#)
predict.sqrlasso, [17](#)
print.gaussian, [18](#)
print.logit, [19](#)
print.poisson, [20](#)
print.sqrlasso, [21](#)