Package 'hdlm'

February 20, 2015

Type Package

Title Fitting High Dimensional Linear Models

Version 1.2

Date 2013-10-24

Author Taylor B. Arnold

Maintainer Taylor B. Arnold <taylor.b.arnold@gmail.com>

Description Mimics the lm function found in the package stats to fit high dimensional regression models with point estimates, standard errors, and p-values. Methods for printing and summarizing the results are given.

License GPL (>= 2.0)

Depends R (>= 3.0), glmnet, foreach, MASS, iterators

LazyLoad yes

Repository CRAN

Date/Publication 2013-10-25 14:42:49

NeedsCompilation yes

R topics documented:

hdglm *Fitting High Dimensional Generalized Linear Models*

Description

hdglm is used to fit high dimensional generalized linear models when the model matrix is rank deficent. The default usage is similar to the glm function in stats; for instance running the code: 'summary(hdglm(y ~ x, family='binomial'))' will produce a regression table. A myriad of options are also avaliable, as described below. For technical and theoretical details of the underlyingmethods see the Details section below as well.

Usage

```
hdglm(formula, data, subset, family =c("gaussian","binomial","poisson"),
    bootstrap = 10, siglevel = 0.05,
    alpha = 0.5, M = NULL, N = NULL, model = TRUE, x = FALSE,
    y = FALSE, scale=TRUE, pval.method=c('median', 'fdr', 'holm', 'QA'),
    ..., FUNCVFIT = NULL, FUNLM = NULL, bayes=FALSE, bayesIters=NULL,
    bayesTune=NULL, refit=FALSE)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
family	Which linking function should be used. Current options are avaliable for gaussian, binomial and poisson data.
bootstrap	number of bootstrap trails to conduct. Default is 10.
siglevel	significance level to use for confidence bounds. Default is 0.05.
alpha	elastic net mixing parameter sent to glmnet, can be any value in (0,1]. When alpha = 1, this is the lasso penalty and when alpha = 0 (not supported) this is the ridge penalty. See glmnet help pages for more details.
M	maximum model size sent to the second stage low-dimensional regression. When more than M variables are choosen in the first stage, the model is trimmed by succesively taking larger sized coefficents until only M remain. If NULL, M is taken to be 90 in the second stage. If M = 0, the model is fit with all of the data once, and the estimated parameters are returned as is.

N	Numer of observations to include in the first stage regression. Default is (# samples / 2), so that the data is split evenly amongst the two stages, which will be set when N=NULL.
model, x, y	logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.
scale	Logical; should the variables in the data matrix be scaled.
pval.method	one of 'median', 'fdr', 'holm', or 'QA'. Signifies the method used to combine p-values when bootstrap is greater than 1. For details and relative strengths of the three methods, see the package vignette.
...	additional arguments to be passed to the low level regression fitting functions (see below).
FUNCVFIT	Used to pass alternative model selection function. Must accept data matrix as its first element and response vector as second element. Return should be a vector of length p (the number of regressors), which indicates which variables are included in the final model. Zero terms are considered to be out of the model; typically all non-zero terms are treated as in the model, though if the model size is too large (see 'M' above), it will be trimmed relative to the absolute size of each non-zero term. Therefore, it is advised to return the model vector in a relative scale rather than an absolute one. The default, used when NULL, is the elastic net function from package glmnet, with the appropriate choice of glm family and with the mixing parameter alpha from above. See package vignette for additional details and examples.
FUNLM	Used to pass alternative second stage, low-dimensional function. Must accept as its first argument a formula object. The return class must have a summary method and the summary method in turn must have a coef method. The coef.summary should return a matrix where the first column are the coefficients and the second column are standard errors. Intercepts should be handled according to the passed formula. As an example, stats::lm works by default; stats::lm. Default is appropriate variant on glm.
bayes	logical. Should Bayesian method be used in place of the two stage method. Only implemented for
bayesIters	number of iterations to conduct in the Gibbs sampler when bayes=TRUE. A total of (bayesIters * 0.1) burn-in steps are included as well. Default is 1000, and can be set by setting bayesIters = NULL.
bayesTune	when family='binomial', a numerical vector of length 1 which serves as a tuning parameter for the Bayes estimator. Defines independent Bernoulli(bayesTune) priors on whether a variable is included in the support of the beta vector. When family='gaussian', should be a numerical vector tuning parameter for the Bayes estimator. Defines a Beta(bayesTune[1], bayesTune[2]) prior on the proportion of variables included in the true support.
refit	Either a logical or number in (0,1]. When not equal to false, the final model will be refit from the entire dataset using FUNLM. When a numeric, the model is selected by only including variables with p-values less than refit. When set to TRUE, any variable coorisponding to a non-zero p-value is included. Cannot be non-FALSE when bayes=TRUE and family='binomial'.

Details

Models for hdglm are specified symbolically. A typical model has the form response ~ terms where response is the (numeric) response vector and terms is a series of terms which specifies a linear predictor for response. A terms specification of the form first + second indicates all the terms in first together with all the terms in second with duplicates removed. A specification of the form first:second indicates the set of terms obtained by taking the interactions of all terms in first with all terms in second. The specification first*second indicates the *cross* of first and second. This is the same as first + second + first:second.

If the formula includes an offset, this is evaluated and subtracted from the response.

See model.matrix for some further details. The terms in the formula will be re-ordered so that main effects come first, followed by the interactions, all second-order, all third-order and so on: to avoid this pass a terms object as the formula (see aov and demo(glm.vr) for an example).

A formula has an implied intercept term. To remove this use either y ~ x - 1 or y ~ 0 + x. See formula for more details of allowed formulae. Note that the intercept term will not be penalized along with other terms. If you want a penalized intercept, add it to directly to the matrix x.

Value

hdglm generally returns an object of class "hdlm", unless refit is not set to false. In the latter case the output is dependent on the choice of funtion FUNLM.

The function summary is used to obtain and print a summary of the results. The generic accessor functions coefficients, effects, fitted.values and residuals extract various useful features of the value returned by hdlm.

Note

This package focuses on methods which produce sparse estimates. Users who do not require sparse estimates are directed to other methods such as ridge regression, and the Bayesian lasso.

Author(s)

Created by Taylor B. Arnold for point estimation and confidence intervals in high-dimensional regression.

The Bayesian option for package hdlm is as implemented with Gibbs sampling with C code from Chris hans, avaliable as packged with package 'blasso' from: www.stat.osu.edu/~hans/software/blasso/

The design of the function was inspired by the S/R function lm and glm described in Chambers (1992).

References

Bickel, P.J., Y. Ritov, and A.B. Tsybakov (2009) "Simultaneous analysis of Lasso and Dantzig selector". *The Annals of Statistics* 37.4, pp. 1705–1732.

Buhlmann, P. and S. Van De Geer (2011) *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer-Verlag New York Inc.

Chambers, J. M. (1992) *Linear models.* Chapter 4 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) *Annals of Statistics*; see also http://www-stat.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf.

Fan, J., Y. Feng, and Y. Wu (2009) "Network exploration via the adaptive LASSO and SCAD penalties". *Annals of Applied Statistics* 3.2, pp. 521–541.

Hans, C. (2009). Brief Technical Report to Accompany the R Package blasso Bayesian Lasso Regression. URL http://www.stat.osu.edu/~hans/software/blasso/.

Hastie, Tibshirani and Friedman (2002) Elements of Statistical Learning, Springer, NY.

Wasserman, L., and Roeder, K. (2009), "High Dimensional Variable Selection," The Annals of Statistics, 37, 2178–2201.

Examples

```
set.seed(42)
x <- matrix(rnorm(10*100),ncol=10)
mu <- exp(x[,1] + x[,2]*0.5) / (1 + exp(x[,1] + x[,2]*0.5))

y <- rbinom(100,1,prob=mu)

out <- hdglm(y ~ x, family='binomial')
summary(out)
```

`hdlm` *Fitting High Dimensional Linear Models*

Description

`hdlm` is used to fit high dimensional linear models when the model matrix is rank deficent. The default usage is the same as the lm function in stats; for instance running the code: 'summary(hdlm(y ~ x))' will produce a regression table. A myriad of options are also avaliable, as described below. For technical and theoretical details of the underlying methods see the Details section below as well.

Usage

```
hdlm(formula, data, subset, bootstrap = 10, siglevel = 0.05,
    alpha = 0.5, M = NULL, N = NULL, model = TRUE, x = FALSE,
    y = FALSE, scale=TRUE, pval.method=c('median', 'fdr', 'holm', 'QA'),
    ..., FUNCVFIT = NULL, FUNLM = NULL, bayes=FALSE, bayesIters=NULL,
    bayesTune = c(1,1), refit=FALSE)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. The details of model specification are given under 'Details'.
data	an optional data frame, list or environment (or object coercible by as.data.frame to a data frame) containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which lm is called.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
bootstrap	number of bootstrap trails to conduct. Default is 10.
siglevel	significance level to use for confidence bounds. Default is 0.05.
alpha	elastic net mixing parameter sent to glmnet, can be any value in (0,1]. When alpha = 1, this is the lasso penalty and when alpha = 0 (not supported) this is the ridge penalty. See glmnet help pages for more details.
M	maximum model size sent to the second stage low-dimensional regression. When more than M variables are choosen in the first stage, the model is trimmed by succesively taking larger sized coefficents until only M remain. If NULL, M is taken to be 90 in the second stage. If M = 0, the model is fit with all of the data once, and the estimated parameters are returned as is.
N	Numer of observations to include in the first stage regression. Default is (# samples / 2), so that the data is split evenly amongst the two stages, which will be set when N=NULL.
model, x, y	logicals. If TRUE the corresponding components of the fit (the model frame, the model matrix, the response, the QR decomposition) are returned.
scale	Logical; should the variables in the data matrix be scaled.
pval.method	one of 'median', 'fdr', 'holm', or 'QA'. Signifies the method used to combine p-values when bootstrap is greater than 1. For details and relative strengths of the three methods, see the package vignette.
...	additional arguments to be passed to the low level regression fitting functions (see below).
FUNCVFIT	Used to pass alternative model selection function. Must accept data matrix as its first element and response vector as second element. Return should be a vector of length p (the number of regressors), which indicates which variables are included in the final model. Zero terms are considered to be out of the model; typically all non-zero terms are treated as in the model, though if the model size is too large (see 'M' above), it will be trimmed relative to the absolute size of each non-zero term. Therefore, it is advised to return the model vector in a relative scale rather than an absolute one. The default, used when NULL, is the elastic net function from package glmnet, with the mixing parameter alpha from above. See package vignette for additional details and examples.
FUNLM	Used to pass alternative second stage, low-dimensional function. Must accept as its first argument a formula object. The return class must have a summary method and the summary method in turn must have a coef method. The

coef.summary should return a matrix where the first column are the coefficients and the second column are standard errors. Intercepts should be handled according to the passed formula. As an example, stats::lm works by default; stats::lm is additionally the default when FUNLM is set to NULL. See package vignette for additional details and examples.

`bayes` logical. Should Bayesian method be used in place of the two stage method.

`bayesIters` number of iterations to conduct in the Gibbs sampler when bayes=TRUE. A total of (bayesIters * 0.1) burn-in steps are included as well. Default is 1000, and can be set by setting bayesIters = NULL.

`bayesTune` numerical vector tuning parameter for the Bayes estimator. Defines a Beta(bayesTune[1], bayesTune[2]) prior on the proportion of variables included in the true support.

`refit` Either a logical or number in (0,1]. When not equal to false, the final model will be refit from the entire dataset using FUNLM. When a numeric, the model is selected by only including variables with p-values less than refit. When set to TRUE, any variable corrisponding to a non-zero p-value is included.

Details

Models for `hdlm` are specified symbolically. A typical model has the form `response ~ terms` where `response` is the (numeric) response vector and `terms` is a series of terms which specifies a linear predictor for `response`. A terms specification of the form `first + second` indicates all the terms in `first` together with all the terms in `second` with duplicates removed. A specification of the form `first:second` indicates the set of terms obtained by taking the interactions of all terms in `first` with all terms in `second`. The specification `first*second` indicates the *cross* of `first` and `second`. This is the same as `first + second + first:second`.

If the formula includes an `offset`, this is evaluated and subtracted from the response.

See `model.matrix` for some further details. The terms in the formula will be re-ordered so that main effects come first, followed by the interactions, all second-order, all third-order and so on: to avoid this pass a `terms` object as the formula (see `aov` and `demo(glm.vr)` for an example).

A formula has an implied intercept term. To remove this use either `y ~ x - 1` or `y ~ 0 + x`. See `formula` for more details of allowed formulae. Note that the intercept term will not be penalized along with other terms. If you want a penalized intercept, add it to directly to the matrix x.

Value

`hdlm` generally returns an object of `class` `"hdlm"`, unless refit is not set to false. In the latter case the output is dependent on the choice of funtion FUNLM.

The function `summary` is used to obtain and print a summary of the results. The generic accessor functions `coefficients`, `effects`, `fitted.values` and `residuals` extract various useful features of the value returned by `hdlm`.

Note

This package focuses on methods which produce sparse estimates. Users who do not require sparse estimates are directed to other methods such as ridge regression.

Author(s)

Created by Taylor B. Arnold for point estimation and confidence intervals in high-dimensional regression.

The Bayesian option for package hdlm is as implemented with Gibbs sampling with C code from Chris hans, avaliable as packged with package 'blasso' from: www.stat.osu.edu/~hans/software/blasso/

The design of the function was inspired by the S/R function `lm` described in Chambers (1992).

References

Bickel, P.J., Y. Ritov, and A.B. Tsybakov (2009) "Simultaneous analysis of Lasso and Dantzig selector". *The Annals of Statistics* 37.4, pp. 1705–1732.

Buhlmann, P. and S. Van De Geer (2011) *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer-Verlag New York Inc.

Chambers, J. M. (1992) *Linear models.* Chapter 4 of *Statistical Models in S* eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole.

Efron, Hastie, Johnstone and Tibshirani (2003) "Least Angle Regression" (with discussion) *Annals of Statistics*; see also http://www-stat.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf.

Fan, J., Y. Feng, and Y. Wu (2009) "Network exploration via the adaptive LASSO and SCAD penalties". *Annals of Applied Statistics* 3.2, pp. 521–541.

Hastie, Tibshirani and Friedman (2002) Elements of Statistical Learning, Springer, NY.

Hans, C. (2009). Brief Technical Report to Accompany the R Package blasso Bayesian Lasso Regression. URL http://www.stat.osu.edu/~hans/software/blasso/.

Wasserman, L., and Roeder, K. (2009), "High Dimensional Variable Selection," The Annals of Statistics, 37, 2178–2201.

Examples

```
set.seed(1)
x <- matrix(rnorm(100*40),ncol=100)
y <- x[,1] + x[,2] * 0.5 + rnorm(40, sd=0.1)
out <- hdlm(y ~ x)
summary(out)
```

`summary.hdlm` *summarize coefficients from a "glmnet" object*

Description

'summary' method for class '"lm"'.

Usage

```
## S3 method for class 'hdlm'
summary(object, ..., level=NULL)
```

Arguments

object	an object of class '"hdlm"', usually, a result of a call to 'hdlm'.
level	Determines which coefficients to print on regression table. Level = 1 gives only those with non-zero coefficents if pval.method is equal to 'mean' and only those with p-value < 1 otherwise. Level = 2 gives anything with non-zero coefficient or non-one p-value, and Level = 3 (or any other choice) gives all coefficients.
...	further arguments passed to or from other methods.

Details

When fitting a large model, it can be cumbersome to look at results for all of the variables; when there exists high correlation between variables, level = 2 is often preferable. The different behavior for pval.method equal to 'mean' is due to the fact that the mean method (or Bayes method, which sets pval.method to 'mean') gives many p-values which are close to, but not exactly equal to, one.

Index