Package ‘higrad’

October 13, 2022

Type Package

Title Statistical Inference for Online Learning and Stochastic
Approximation via HiGrad

Version 0.1.0

Description Implements the Hierarchical Incremental GRAdient Descent (HiGrad) algorithm,
a first-order algorithm for finding the minimizer of a function in online learning just like stochastic gradient descent (SGD).
In addition, this method attaches a confidence interval to assess the uncertainty of its predictions.
See Su and Zhu (2018) <arXiv:1802.04876> for details.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports Matrix

NeedsCompilation no

Author Weijie Su [aut],
Yuancheng Zhu [aut, cre]

Maintainer Yuancheng Zhu <yuancheng.zhu@gmail.com>

Repository CRAN

Date/Publication 2018-03-14 15:44:13 UTC

R topics documented:

higrad *Fitting HiGrad*

Description

higrad is used to implement hierarchical incremental gradient descent (HiGrad), an algorithm that conducts statistical inference for online learning.

Usage

```
higrad(x, y, model = "lm", nsteps = nrow(x), nsplits = 2, nthreads = 2,
  step.ratio = 1, n0 = NA, skip = 0, eta = 1/2, alpha = 1/2,
  burnin = round(nsteps/10), start = rnorm(ncol(x), 0, 0.01),
  replace = FALSE, track = FALSE)
```

Arguments

`x` input matrix of features. Each row is an observation vector, and each column is a feature.

`y` response variable. Quantitative for `model = "lm"`. For `model = "logistic"` it should be a factor with two levels.

`model` type of model to fit. Currently only linear regression (`"lm"`) and logistic regression (`"logistic"`) are supported.

`nsteps` total number of steps. This is equivalent to the number of queries made to get a noisy evaluation of the gradient.

`nsplits` number of splits in the HiGrad tree.

`nthreads` numbers of threads each previous thread is split into. Either a number (equal split size throughout) or a vector.

`step.ratio` ratio of the lengths of the threads from the two adjacent levels (the latter one divided by the previous). Either a number (equal ratio throughout) or a vector.

`n0` length of the 0th-level thread.

`skip` number of steps to skip when estimating the coefficients by averaging.

`eta` constant in front of the step size. See Details for the formula of the step size.

`alpha` exponent of the step size. See Details for the formula of the step size.

`burnin` number of steps as the burn-in period. The burn-in period is not accounted for in the total budget `nsteps`.

`start` starting values of the coefficients.

`replace` logical; whether or not to sample the data with replacement.

`track` logical; whether or not to store the entire path for plotting.

Details

HiGrad is designed to conduct statistical inference for online learning, without incurring additional computational cost compared with the vanilla stochastic gradient descent (SGD). The HiGrad procedure begins by performing SGD iterations for a while and then split the single thread into a few, and this procedure hierarchically operates in this fashion along each thread. With predictions provided by multiple threads in place, a t-based confidence interval is constructed by de-correlating predictions using covariance structures given by the Ruppert–Polyak averaging scheme. In order to implement HiGrad, a configuration of the tree structure needs to be specified. The default setting is a binary tree with 2 splits. The step size is set to be `eta*t^(-alpha)`.

Value

An object with S3 class `higrad`.

`coefficients` estimate of the coefficients.

`coefficients.bootstrap`
matrix of estimates of the coefficients along each HiGrad threads.

`model` model type.

`Sigma0` covariance structure Σ of the estimates.

`track` entire path of the estimates along each thread. Can be used for diagnostic and check for convergence.

References

Weijie Su and Yuancheng Zhu. (2018) *Statistical Inference for Online Learning and Stochastic Approximation via Hierarchical Incremental Gradient Descent*. https://arxiv.org/abs/1802.04876.

See Also

See `print.higrad`, `plot.higrad`, `predict.higrad` for other methods for the `higrad` class.

Examples

```
# fitting linear regression on a simulated dataset
n <- 1e3
d <- 10
sigma <- 0.1
theta <- rep(1, d)
x <- matrix(rnorm(n * d), n, d)
y <- as.numeric(x %*% theta + rnorm(n, 0, sigma))
fit <- higrad(x, y, model = "lm")
print(fit)
# predict for 10 new samples
newx <- matrix(rnorm(10 * d), 10, d)
pred <- predict(fit, newx)
pred
```

`plot.higrad` *Plot a* `higrad` *Object*

Description

Produces a coefficient paths for a fitted `higrad` object.

Usage

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

Arguments

`x` a fitted object of class `higrad`.

`...` additional graphical parameters.

`predict.higrad` *Obtain Prediction and Confidence Intervals From a HiGrad Fit*

Description

`predict` can be applied with a `higrad` object to obtain predictions and confidence intervals.

Usage

```
## S3 method for class 'higrad'
predict(object, newx, alpha = 0.05, type = "link",
  prediction.interval = FALSE, ...)
```

Arguments

`object` a fitted object of class `higrad`.

`newx` matrix of new values for `x` at which predictions are to be made.

`alpha` significance level. The confidence level of the interval is thus 1 - `alpha`.

`type` type of prediction required. Type `"link"` gives the linear predictors for `"logistic"`; for `"lm"` models it gives the fitted values. Type `"response"` gives the fitted probabilities for `"logistic"`; for `"lm"` type `"response"` is equivalent to type `"link"`.

`prediction.interval` logical; indicator of whether prediction intervals should be returned instead of confidence intervals.

`...` other prediction options.

Value

A list with components

`pred`	predicted values.
`upper`	upper limit of the confidence/prediction intervals.
`lower`	lower limit of the confidence/prediction intervals.

`print.higrad` *Print a* `higrad` *Object*

Description

Print the coefficients estimates obtained by HiGrad.

Usage

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

Arguments

`x`	a fitted object of class `higrad`.
`...`	additional print arguments.

Index