Package 'bigKRLS'

August 3, 2019

Type Package

Title Optimized Kernel Regularized Least Squares

Version 3.0.5.1

Description Functions for Kernel-Regularized Least Squares optimized for speed and memory usage are provided along with visualization tools. For working papers, sample code, and recent presentations visit <https://sites.google.com/site/petemohanty/software/>. bigKRLS, as well its dependencies, require current versions of R and its compilers (and RStudio if used). For details, see <https://github.com/rdrr1990/bigKRLS/blob/master/INSTALL.md>.

License GPL (>= 2)

Imports bigalgebra, biganalytics, ggplot2, parallel, Rcpp (>= 0.12.4), shiny

LinkingTo bigmemory, BH, Rcpp, RcppArmadillo

Depends R (>= 3.3.0), bigmemory

URL https://github.com/rdrr1990/bigKRLS

BugReports https://github.com/rdrr1990/bigKRLS/issues

RoxygenNote 6.1.1

Suggests covr, knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Maintainer Pete Mohanty <pete.mohanty@gmail.com>

Repository CRAN

Encoding UTF-8

Author Pete Mohanty [aut, cre] (<https://orcid.org/0000-0001-8531-3345>), Robert Shaffer [aut] (<https://orcid.org/0000-0002-2081-2407>)

Date/Publication 2019-08-02 22:10:07 UTC

R topics documented:

`bigKRLS` *bigKRLS*

Description

Runtime and Memory Optimized Kernel Regularized Least Squares Pete Mohanty (Stanford University) and Robert Shaffer (University of Texas at Austin)

Usage

```
bigKRLS(y = NULL, X = NULL, sigma = NULL, derivative = TRUE,
  which.derivatives = NULL, vcov.est = TRUE, Neig = NULL,
  eigtrunc = NULL, lambda = NULL, L = NULL, U = NULL, tol = NULL,
  model_subfolder_name = NULL, overwrite.existing = FALSE,
  Ncores = NULL, acf = FALSE, noisy = NULL, instructions = TRUE)
```

Arguments

`y`	A vector of numeric observations on the dependent variable. Missing values not allowed. May be base R matrix or big.matrix.
`X`	A matrix of numeric observations of the independent variables. Factors, missing values, and constant vectors not allowed. May be base R matrix or big.matrix.
`sigma`	Bandwidth parameter, shorthand for sigma squared. Default: sigma <- ncol(X).
`derivative`	Logical. Estimate derivatives (as opposed to just coefficients)? Recommended for interpretability.
`which.derivatives`	Optional. For which columns of X should marginal effects be estimated? If derivative=TRUE and which.derivative=NULL (default), all marginal effects will be estimated.
`vcov.est`	Logical. Estimate variance covariance matrix? Required to obtain derivatives and standard errors on predictions.
`Neig`	Number of eigenvectors and eigenvalues to calculate. The default is to calculate all N and only use those where eigval >= 0.001 * max(eigval). Smaller values will reduce runtime, but decrease precision.

`eigtrunc` Eigentruncation parameter. If NULL, defaults to 0.001 if N > 3000 and 0 otherwise. eigtrunc = 0.25 keeps only those eigenvectors/values such that the eigenvalue is at least 25% of the max. If eigtrunc == 0, all Neig are used to select lambda and to estimate variances. Larger values will reduce runtime, but decrease precision.

`lambda` Regularization parameter. Default: selected based (in part) on the eigenvalues of the kernel via Golden Search. Must be positive, real number.

`L` Lower bound of Golden Search for lambda.

`U` Upper bound of Golden Search for lambda.

`tol` tolerance parameter for Golden Search for lambda. Default: N / 1000.

`model_subfolder_name`
If not null, will save estimates to this subfolder of your current working directory. Alternatively, use save.bigKRLS() on the outputted object.

`overwrite.existing`
Logical. overwrite contents in folder 'model_subfolder_name'? If FALSE, appends lowest possible number to model_subfolder_name name (e.g., ../myresults3/).

`Ncores` Number of processor cores to use. Default = ncol(X) or N - 2 (whichever is smaller). More than N - 2 NOT recommended. Uses library(parallel) unless Ncores = 1.

`acf` Logical. Experimental; default == FALSE. Calculate Neffective as function of mean absolute auto-correlation in X to correct p-values?

`noisy` Logical. Display detailed version of progress to console (intermediate output, time stamps, etc.) as opposed to minimal display? Default: if(N > 2000) TRUE else FALSE. SSH users should use X11 forwarding to see Rcpp progress display.

`instructions` Logical. Display syntax after estimation with other library(bigKRLS) functions that can be used on output?

Details

Kernel Regularized Least Squares (KRLS) is a kernel-based, complexity-penalized method developed by Hainmueller and Hazlett (2014) to minimize parametric assumptions while maintaining interpretive clarity. Here, we introduce bigKRLS, an updated version of the original KRLS R package with algorithmic and implementation improvements designed to optimize speed and memory usage. These improvements allow users to straightforwardly fit KRLS models to medium and large datasets (N > ~2500).

Major Updates:

1. C++ integration. We re-implement most major computations in the model in C++ via Rcpp and RcppArmadillo. These changes produce up to a 50% runtime decrease compared to the original R implementation even on a single core.

2. Leaner algorithm. Because of the Tikhonov regularization and parameter tuning strategies used in KRLS, this method of estimation is inherently memory-heavy O(N^2), making memory savings important even for small- and medium-sized applications. We develop and implement a new local derivatives algorithm, which reduces peak memory usage by approximately an order of magnitude, and cut the number of computations needed to find regularization parameter in half.

3. Improved memory management. Most data objects in R perform poorly in memory-intensive applications. We use a series of packages in the bigmemory environment to ease this constraint, allowing our implementation to handle larger datasets more smoothly.

4. Parallel Processing. We parallelize most major calculations in the model. Time savings are especially noticeable in the derivative estimation portion of the algorithm.

5. Interactive data visualization. We include an R Shiny app that allows users bigKRLS users to easily share results with collaborators or more general audiences. Simply call shiny.bigKRLS() on the outputted regression object.

6. Improved uncertainty estimates. bigKRLS uses an adjusted degrees-of-freedom estimator that reflect both the regularization process and the number of predictors. For details and other options, see help(summary.bigKRLS).

7. Cross-validation. crossvalidate.bigKRLS performs CV and stores performance results and parameter settings for each fold. See vignette("bigKRLS_basics") or help("crossvalidate.bigKRLS") for syntax.

Requirements. bigKRLS is under active development. bigKRLS, as well its dependencies, require current versions of R and its compilers (and RStudio if used). For details, see https://github.com/rdrr1990/code/blob/master/bigKRLS_installation.md.

For details on syntax, load the library and then open our vignette vignette("bigKRLS_basics"). Because of the quadratic memory requirement, users working on a typical laptop (8-16 gigabytes of RAM) may wish to start at N = 2,500 or 5,000, particularly if the number of *x* variables is large. When you have a sense of how bigKRLS runs on your system, you may wish to only estimate a subset of the marginal effects at N = 10-15,000 by setting bigKRLS(... which.derivatives = c(1, 3, 5)) for the marginal effects of the first, third, and fifth x variable.

Pete Mohanty and Robert Shaffer. 2018. "Messy Data, Robust Inference? Navigating Obstacles to Inference with bigKRLS." Political Analysis. Cambridge University Press. DOI=10.1017/pan.2018.33. pages 1-18. See also: Mohanty, Pete; Shaffer, Robert, 2018, "Replication Data for: Messy Data, Robust Inference? Navigating Obstacles to Inference with bigKRLS", https://doi.org/10.7910/DVN/CYYLOK, Harvard Dataverse, V1.

Hainmueller, Jens and Chad Hazlett. 2014. "Kernel Regularized Least Squares: Reducing Misspecification Bias with a Flexible and Interpretable Machine Learning Approach." Political Analysis. 22:143-68. https://web.stanford.edu/~jhain/Paper/PA2014a.pdf (Accessed May 20th, 2016).

Recent papers, presentations, and other code available at https://github.com/rdrr1990/code/

Value

bigKRLS Object containing slope and uncertainty estimates; summary() and predict() defined for class bigKRLS, as is shiny.bigKRLS().

Author(s)

Maintainer: Pete Mohanty <pete.mohanty@gmail.com> (0000-0002-4985-5160)

Authors:

- Robert Shaffer <shafferr@upenn.edu> (0000-0002-2081-2407)

See Also

Useful links:

- https://github.com/rdrr1990/bigKRLS
- Report bugs at https://github.com/rdrr1990/bigKRLS/issues

Examples

```
# Analyzing chicken weights
# y <- as.matrix(ChickWeight$weight)
# X <- matrix(cbind(ChickWeight$Time, ChickWeight$Diet == 1), ncol = 2)

# out <- bigKRLS(y, X)
# out$R2
# summary(out, labs = c("Time", "Diet"))

# don't use save() unless out$has.big.matrices == FALSE
# save.bigKRLS(out, "exciting_results")
```

`crossvalidate.bigKRLS` *crossvalidate.bigKRLS*

Description

crossvalidate.bigKRLS

Usage

```
crossvalidate.bigKRLS(y, X, seed, Kfolds = NULL, ptesting = NULL,
  estimates_subfolder = NULL, ...)
```

Arguments

`y`	A vector of numeric observations on the dependent variable. Missing values not allowed.
`X`	A matrix of numeric observations of the independent variables. Factors, missing values, and constant vectors not allowed.
`seed`	Randomization seed to be used when partitioning data.
`Kfolds`	Number of folds for cross validation. Requires ptesting == NULL. Note KRLS assumes variation in each column; rare events or rarely observed factor levels may violate this assumption if Kfolds is too large given the data.
`ptesting`	Percentage of data to be used for testing (e.g., ptesting = 20 means 80% training, 20% testing). Requires Kfolds == NULL. Note KRLS assumes variation in each column; rare events or rarely observed factor levels may violate this assumptions if ptesting is too small given the data.
`estimates_subfolder`	If non-null, saves all model estimates in current working directory.

...	Additional arguments to be passed to bigKRLS() or predict(). E.g., crossvalidate.bigKRLS(y, X, derivative = FALSE) will run faster but compute fewer test stats comparing in and out of sample performance (because the marginal effects will not be estimated).

Value

bigKRLS_CV (list) Object of estimates and summary stats; summary() is defined. For train/test, contains a bigKRLS regression object and a predict object. For Kfolds, contains a nested series of training and testing models.

Examples

```
# y <- as.matrix(ChickWeight$weight)
# X <- matrix(cbind(ChickWeight$Time, ChickWeight$Diet == 1), ncol = 2)

# cv.out <- crossvalidate.bigKRLS(y, X, seed = 123, ptesting = 20)
# cv.out$pseudoR2_oos
# cv <- summary(cv.out)

# cv$training.ttests

# kcv.out <- crossvalidate.bigKRLS(y, X, seed = 123, Kfolds = 3)
# kcv <- summary(kcv.out, digits = 3)

# kcv$overview
# kcv$training2.ttests

# save.bigKRLS(kcv.out, "myKfolds")
# load.bigKRLS("/path/to/myKfolds")
```

`load.bigKRLS` *load.bigKRLS*

Description

Reconstructs bigKRLS output object as list.

Usage

```
load.bigKRLS(path, newname = NULL, pos = 1, noisy = TRUE,
  return_object = FALSE)
```

Arguments

`path`	Path to folder where bigKRLS object was saved.
`newname`	If NULL (default), bigKRLS regression and prediction output will appear as "bigKRLS_out", while crossvalidation results will appear as "object".

pos	position. Default == 1 (global environment). NULL means don't assign (return only).
noisy	Logical. display updates?
return_object	Logical. return library(bigKRLS) object? Default == FALSE.

Examples

```
# save.bigKRLS(out, "exciting_results") # don't use save()
# load.bigKRLS("exciting_results") # don't use load()
```

---

predict.bigKRLS *predict.bigKRLS*

Description

Predict function for bigKRLS object. crossvalidate.bigKRLS() provides additional functionality.

Usage

```
## S3 method for class 'bigKRLS'
predict(object, newdata, se.pred = FALSE,
  correct_SE = TRUE, ytest = NULL, ...)
```

Arguments

object	bigKRLS output
newdata	New data. Columns in newdata should be ordered identically to columns in X.
se.pred	Calculate standard errors on predictions?
correct_SE	If se.pred == TRUE, default is to use Neffective (if available) rather than model N in calculating uncertainty of predictions.
ytest	Provide testing data to have it returned with the object. Optional. To automatically generate out-of-sample test statistics, use crossvalidate.bigKRLS() instead.
...	ignore

Value

Returns bigKRLS_predicted list object.

Examples

```
# set.seed(123)
# y <- as.matrix(ChickWeight$weight)
# X <- matrix(cbind(ChickWeight$Time, ChickWeight$Diet == 1), ncol = 2)
# N <- length(y)
```

```
# train <- sample(N, 100, replace = FALSE)
# outOfSample <- c(1:N)[-train]
# test <- sample(outOfSample, 10, replace = FALSE)

# fit <- bigKRLS(y[train], X[train,], instructions = FALSE)
# p <- predict(fit, X[test,])
```

---

save.bigKRLS *save.bigKRLS*

Description

save function, recommended when bigKRLS output contains big matrices (once N > 2,500 the kernel is stored this way). Base R data will be stored in a list in an .RData file, big matrices will be stored in .txt files. Call load.bigKRLS() to retrieve.

Usage

```
save.bigKRLS(object, model_subfolder_name, overwrite.existing = FALSE,
  noisy = TRUE)
```

Arguments

object	bigKRLS output (regression, prediction, and crossvalidation). Use load.bigKRLS(model_subfolder_name not load().
model_subfolder_name	A name of a folder where the file(s) will be written.
overwrite.existing	Logical. If TRUE, write over folders with the same name. Default == FALSE.
noisy	Logical. If TRUE display progress, additional instructions. Default == TRUE.

Examples

```
# y <- as.matrix(ChickWeight$weight)
# X <- matrix(cbind(ChickWeight$Time, ChickWeight$Diet == 1), ncol = 2)

# out <- bigKRLS(y, X, Ncores=1)
# save.bigKRLS(out, "exciting_results")

# don't use save() unless out$has.big.matrices == FALSE
# load.bigKRLS("/path/to/exciting_results")
```

shiny.bigKRLS *shiny.bigKRLS*

Description

shiny.bigKRLS

Usage

```
shiny.bigKRLS(out, export = FALSE, main.label = "bigKRLS estimates",
  plot.label = NULL, xlabs = NULL, font_size = 20, hline = 0,
  shiny.palette = c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3",
  "#FF7F00", "#FFFF33", "#A65628", "#F781BF", "#999999"))
```

Arguments

out	bigKRLS output. Does not require any N * N matrices.
export	Logical. Instead of running Shiny, prepare the key estimates as a separate file.
main.label	Main label (upper left of app)
plot.label	Optional character
xlabs	Optional character vector for the x variables.
font_size	Font size. Default == 20. calls "ggplot2::theme_minimal(base_size = font_size)"
hline	horizontal line. Default == 0 (x axis)
shiny.palette	color scheme for main app. 9 colors.

Examples

```
# N <- 500
# P <- 4
# X <- matrix(rnorm(N*P), ncol=P)

# b <- 1:P
# y <- sin(X[,1]) + X %*% b + rnorm(N)

# out <- bigKRLS(y, X, Ncores=1)
# shiny.bigKRLS(out, "exciting_results", "The Results", c("Frequency", "xA", "xB", "xC"))
```

summary.bigKRLS *summary.bigKRLS*

Description

Summary function for bigKRLS output.

Usage

```
## S3 method for class 'bigKRLS'
summary(object, degrees = "Neffective",
  probs = c(0.05, 0.25, 0.5, 0.75, 0.95), digits = 4, labs = NULL,
  ...)
```

Arguments

object	bigKRLS output.
degrees	"Neffective" (default) or "N". If "Neffective" (default), degrees of freedom for t tests reflects degrees of freedom used to obtain regularization parameter, lambda. Neffective = N - sum(eigenvalues/(eigenvalues + lambda)); see e.g. Hastie et al. (2015, 61-68). 'N' is simply the observed sample size (note this is the default for library(KRLS)). Degrees of freedom for t-tests is either Neffective - P or N - P.
probs	Which quantiles of the marginal effects of each x variable should be displayed?
digits	Number of signficant digits.
labs	Optional vector of x labels.
...	ignore

Value

Returns list with "ttests" (Average Marginal Effect estimates, standard errors, t-values, and p values) and "percentiles" (of the marginal effects).

Examples

```
# y <- as.matrix(ChickWeight$weight)
# X <- matrix(cbind(ChickWeight$Time, ChickWeight$Diet == 1), ncol = 2)

# out <- bigKRLS(y, X, Ncores=1)
# summary(out)

# s = summary(out, digits = 2, labs = c("Time", "ChickWeightDiet"))

# knitr::kable(s[["ttests"]])     # to format with RNotebook or RMarkdown
# knitr::kable(s[["percentiles"]])

# Calcuate p-values using uncorrected standard errors (not recommended)
# summary(out, degrees = "N")
```

summary.bigKRLS_CV *summary.bigKRLS_CV*

Description

Summary function for bigKRLS crossvalidated output.

Usage

```
## S3 method for class 'bigKRLS_CV'
summary(object, ...)
```

Arguments

object	bigKRLS_CV output.
...	Additional parameters to be passed to summary() for the training model(s). For example, summary(cv, digits = 3). See ?bigKRLS.summary for details.

Examples

```
# y <- as.matrix(ChickWeight$weight)
# X <- matrix(cbind(ChickWeight$Time, ChickWeight$Diet == 1), ncol = 2)

# cv.out <- crossvalidate.bigKRLS(y, X, seed = 123, ptesting = 20)
# summary(cv.out)

# cv <- summary(cv.out, labs = c("Alpha", "Beta", "Gamma", "Delta", "Epsilon"))
# cv$training.ttests

# kcv.out <- crossvalidate.bigKRLS(y, X, seed = 123, Kfolds = 3)
# summary(kcv.out)

# kcv <- summary(kcv.out)
# kcv$overview
# kcv$training2.ttests
```

Index