Package ‘LPMachineLearning’

August 25, 2020

Type Package

Title Integrated Nonparametric Statistical Machine Learning

Version 1.0

Date 2020-07-15

Author Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer Kaijun Wang <kaijunwang.19@gmail.com>

Description Statistical modeling tools for converting a black-box ML algorithm into an interpretable conditional distribution prediction machine, which provides a wide range of facilities, including goodness-of-fit, various types of exploratory graphical diagnostics, generalized feature selection, predictive inference methods, and others. The primary reference is Mukhopadhyay, S. and Wang, K. (2020, Technical Report).

Imports graphics,methods,glmnet,caret,h2o,leaps,HDInterval,parallel

Depends R (>= 3.5.0),stats,orthopolynom

License Apache License (>= 2.0)

R topics documented:

LPMachineLearning-package *Integrated Nonparametric Statistical Machine Learning*

Description

This package provides a unified interface to convert any black-box ML regression algorithms into an exploratory uncertainty prediction machine that is robust, interpretable, and scalable for large datasets. A large variety of modeling and predictive inference tasks can be done using the fitted model.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

autompg *Auto MPG data.*

Description

Modified Auto MPG data set based on the UCI Machine Learning Repository version (Bache and Lichman, 2013). we discarded examples with missing entries, ending up with 392 observations.

Usage

```
data(autompg)
```

Format

A data frame with 392 observations on the following 8 variables.

`x.cylinders` Number of cylinders.

`x.displacement` Engine displacement (cu. inches).

`x.horsepower` Horsepower.

`x.weight` Vehicle weight (lbs).

`x.acceleration` Time to accelerate from O to 60 mph (seconds).

`x.model.year` Model year (modulo 100).

`x.origin` Origin of car (1. American, 2. European, 3. Japanese).

`y` Miles per gallon.

`baseball` *Baseball data.*

Description

Age and weight data for 1015 major league baseball players.

Usage

```
data(baseball)
```

Format

A data frame with 1015 observations on the following 2 variables.

`x` Age.

`y` Weight.

References

Matloff, N. (2017) "Statistical regression and classification: From linear models to machinelearning". Chapman and Hall/CRC.

`bone` *Bone mineral density data.*

Description

The data set contains measurements on the relative change in (spinal) bone mineral density over one year for 485 North American adolescents.

Usage

```
data(bone)
```

Format

A data frame with 485 observations on the following 2 variables.

`x` Age of the subject.

`y` Relative change in (spinal) bone mineral density.

References

Bachrach et al., (1999) "Bone mineral acquisition in healthy Asian, Hispanic, black, and Caucasian youth: a longitudinal study". The journal of clinical endocrinology & metabolism.

boxOffice *Movie Box-office Revenue Data*

Description

Box-office revenues during opening and after the first week.

Usage

```
data(boxOffice)
```

Format

A data frame with 4031 observations on the following 2 variables.

- `x` Log of opening box-office revenues.
- `y` Log of box-office revenues after the first week.

References

Voudouris et al., (2012) "Modelling skewness and kurtosis with the BCPE density in GAMLSS". Journal of Applied Statistics, 39(6), 1279-1293.

bupa *BUPA liver disorders data.*

Description

A modified version of BUPA liver disorders data set, containing measurements of gamma-glutamyl transpeptidase (GGT) and alanine-aminotransferase (ALT) extracted from 345 male individuals' blood sample.

Usage

```
data(bupa)
```

Format

A data frame with 345 observations on the following 2 variables.

- `x` Log of gamma-glutamyl transpeptidase.
- `y` Log of alanine-aminotransferase.

References

McDermott, J. & Forsyth, R.S. (2016) "Diagnosing a disorder in a classification benchmark". Pattern Recognition Letters, 73, 41-43.

`butterfly` *The Butterfly data.*

Description

The stylized simulated example used in our paper.

Usage

```
data(butterfly)
```

Format

A data frame with 700 observations on 2 variables.

`x` Values of covariate X.

`y` Values of Y.

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

`cholesterol` *LDL cholesterol of Quail.*

Description

Completely randomized experiment investigating LDL (low-density lipoprotein) cholesterol in quails.

Usage

```
data(cholesterol)
```

Format

A data frame with 39 observations on the following 2 variables.

`x` Type of diet, each is mixed with a different drug compound.

`y` Measurments of LDL cholesterol levels

References

Hettmansperger, T. P. and J. W. McKean (2010), "Robust nonparametric statistical methods", CRC Press.

DIF *Distributional Impact Function.*

Description

This function deal with the "XYZ" problem where we observe covariates X, response Y and a binary treatment Z. The goal is capturing the heterogeneous impact from the treatment Z on the response Y, as a function of the covariate X.

Usage

```
DIF(X, y, z, m = c(2, 4), X.test, method = "gbm", ...)
```

Arguments

`X`	A n-by-d feature matrix
`y`	A length n vector of response.
`z`	A length n binary vector. Indicating treatment.
`m`	An ordered pair (m_1, m_2). m_1 indicates how many LP-nonparametric basis to construct for each column of X, m_2 indicates how many to construct for y.
`X.test`	A k-by-d matrix providing k sets of covariates for target cases to investigate.
`method`	Method for estimating the conditional LP-Fourier coefficients. Valid options: gbm and rf (both requires `h2o`).
`...`	Extra parameters to pass into `UPM`.

Value

A list of values containing:

`X.test`	The `X.test` values of dimension k-by-d.
`DIF`	A vector of length k containing the DIF values for the `X.test`.
`comp.DIF`	A k-by-m_2 matrix containing the components of DIF values.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

`dutch` *Dutch Boys BMI data*

Description

This dataset is a part of the Fourth Dutch Growth Study, which comprised of observations on age and BMI of 7294 Dutch boys. A slightly modified version that is available in `gamlss.data` package.

Usage

```
data(dutch)
```

Format

A data frame with 7294 observations on the following 2 variables.

`x` Subject age.

`y` Subject BMI.

References

Fredriks et al., (2000) "Body index measurements in 1996-7 compared with 1980". Archives of disease in childhood 82(2), 107-112.

`GSP` *Generalized Shape Predictor.*

Description

Generalized shape predictors are those that influence the whole conditional distribution (beyond just mean) of the response Y. This function finds the most relevant attributes that are predictive for certain shapes (that user is interested in) of the conditional distribution $f_{Y|X=x}(y)$.

Usage

```
GSP(X, y, comp, mx = NULL)
```

Arguments

`X`	A n-by-d feature matrix
`y`	A length n vector of response.
`comp`	A length l vector indicating the target order. `comp=1` will identify the variables that affect the condtional mean of Y, `comp=1:2` will find the variables that are informative for the location and scale, etc.
`mx`	Optional. The number of LP-nonparametric basis m to construct for each feature.

Value

A list of values containing:

coef	A m-by-l-by-d array of coefficients. See example for details.
signif.mat	A binary matrix indicating the significant order features

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

Examples

```
data(autompg)
X<-autompg[,-8]
y<-autompg$y
GSP.mpg<-GSP(X, y, comp=1:2, mx = 4)
#feature coefficients for location component:
GSP.mpg$coef[,1,]
#feature coefficients for scale component:
GSP.mpg$coef[,2,]
#Coefficients for features at first order LP bases:
GSP.mpg$coef[1,,]
```

HCA *Heterogeneity component analysis*

Description

This function performs heterogeneity component analysis of of the response variable Y for identifying which shape compliments are changing with the covarite X.

Usage

```
HCA(X, y, m = c(4, 6), alpha = 0.05, method.ml = "glmnet")
```

Arguments

X	A n-by-d feature matrix
y	A length n vector of response.
m	An ordered pair (m_1, m_2). m_1 indicates how many LP-nonparametric basis to construct for each column of X, m_2 indicates how many to construct for y.
alpha	Threshold for p-values of F-statistics. The plot will only display LP-coefficients whose p-value is smaller than alpha.
method.ml	Method for estimating the conditional LP-Fourier coefficients. In this case, valid input includes: "glmnet", "lm", and "subset"

Value

A list of values containing:

`f.stat`	A vector of length m_2. F-statistics for LP-coefficients.
`dev.rate`	A vector of length m_2. Deviance ratios for LP-coefficients.
`pval`	A vector of length m_2. p-values of the F-statistics.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

Examples

```
##Fig 10(b) of the paper
data(cholesterol)
attach(cholesterol)
m=c(length(unique(x))-1,4)
ldlch.hca<- HCA(x,y,m=m,alpha=NULL,method.ml="lm")

##HCA can also check the heterogeneity of residual series: (Fig 6b of the paper)
data(bone)
attach(bone)
fit.reg<- smooth.spline(x,y)
yhat<-predict(fit.reg,x)$y
y.res<-y-yhat #residuals
bone.hca<-HCA(x,y.res,m=c(2,6),alpha=NULL,method.ml="lm")
```

`LP.basis` *Computes LP basis function from samples.*

Description

This function computes `m` LP basis functions for samples `X`. User can provide an initial pivot density as starting guess.

Usage

```
LP.basis(X, m, pivot = NULL, Fmid = TRUE)
```

Arguments

`X`	Observed values of the random variable. Can also a n-by-d matrix where each column is a realization from a random variable. In that case the function will compute m LP basis functions for each column.
`m`	An integer denoting the number of required LP basis functions.
`pivot`	This accepts either (i) a function object; or (ii) a vector of sub-samples for X. Set to `NULL` to use the marginal ecdf. Note that for multivariate `X`, it is better to leave this option empty as it will attempt to use same pivot on all columns.
`Fmid`	Whether to use mid-rank empirical distribution. Recommended for samples with ties.

Value

A matrix of dimension $n \times mk$.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S. and Parzen, E. (2020) Nonparametric Universal Copula Modeling, Applied Stochastic Models in Business and Industry, special issue on "Data Science", 36(1), 77-94.

Mukhopadhyay, S. (2017) Large-Scale Mode Identification and Data-Driven Sciences. Electronic Journal of Statistics, 11 215-240.

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

Examples

```
#figure 16 of the paper
data(autompg)
m<-4
#weight
x<-sort(autompg[,4])
TX <- LP.basis(x,m)
par(mfrow=c(2,2),mar=c(3,3,3,2))
ux<-ecdf(x)(x)
plot(ux,TX[,1],type="s")
plot(ux,TX[,2],type="s")
plot(ux,TX[,3],type="s")
plot(ux,TX[,4],type="s")
#acceleration
x<-sort(autompg[,5])
TX <- LP.basis(x,m)
ux<-ecdf(x)(x)
plot(ux,TX[,1],type="s")
plot(ux,TX[,2],type="s")
plot(ux,TX[,3],type="s")
plot(ux,TX[,4],type="s")
```

onlineNews *Online news popularity data.*

Description

Popularity study of online articles.

Usage

```
data(onlineNews)
```

Format

A data frame with 39644 observations on the following 60 variables.

`x.timedelta` Days between the article publication and the dataset acquisition.

`x.n_tokens_title` Number of words in the title.

`x.n_tokens_content` Number of words in the content.

`x.n_unique_tokens` Rate of unique words in the content.

`x.n_non_stop_words` Rate of non-stop words in the content.

`x.n_non_stop_unique_tokens` Rate of unique non-stop words in the content.

`x.num_hrefs` Number of links.

`x.num_self_hrefs` Number of links to other articles published by Mashable.

`x.num_imgs` Number of images.

`x.num_videos` Number of videos.

`x.average_token_length` Average length of the words in the content.

`x.num_keywords` Number of keywords in the metadata.

`x.data_channel_is_lifestyle` Is data channel 'Lifestyle'?

`x.data_channel_is_entertainment` Is data channel 'Entertainment'?

`x.data_channel_is_bus` Is data channel 'Business'?

`x.data_channel_is_socmed` Is data channel 'Social Media'?

`x.data_channel_is_tech` Is data channel 'Tech'?

`x.data_channel_is_world` Is data channel 'World'?

`x.kw_min_min` Worst keyword (min. shares).

`x.kw_max_min` Worst keyword (max. shares).

`x.kw_avg_min` Worst keyword (avg. shares).

`x.kw_min_max` Best keyword (min. shares).

`x.kw_max_max` Best keyword (max. shares).

`x.kw_avg_max` Best keyword (avg. shares).

`x.kw_min_avg` Avg. keyword (min. shares).

`x.kw_max_avg` Avg. keyword (max. shares).

`x.kw_avg_avg` Avg. keyword (avg. shares).

`x.self_reference_min_shares` Min. shares of referenced articles in Mashable.

x.self_reference_max_shares Max. shares of referenced articles in Mashable.

x.self_reference_avg_sharess Avg. shares of referenced articles in Mashable.

x.weekday_is_monday Was the article published on a Monday?

x.weekday_is_tuesday Was the article published on a Tuesday?

x.weekday_is_wednesday Was the article published on a Wednesday?

x.weekday_is_thursday Was the article published on a Thursday?

x.weekday_is_friday Was the article published on a Friday?

x.weekday_is_saturday Was the article published on a Saturday?

x.weekday_is_sunday Was the article published on a Sunday?

x.is_weekend Was the article published on the weekend?

x.LDA_00 Closeness to LDA topic 0.

x.LDA_01 Closeness to LDA topic 1.

x.LDA_02 Closeness to LDA topic 2.

x.LDA_03 Closeness to LDA topic 3.

x.LDA_04 Closeness to LDA topic 4.

x.global_subjectivity Text subjectivity.

x.global_sentiment_polarity Text sentiment polarity.

x.global_rate_positive_words Rate of positive words in the content.

x.global_rate_negative_words Rate of negative words in the content.

x.rate_positive_words Rate of positive words among non-neutral tokens.

x.rate_negative_words Rate of negative words among non-neutral tokens.

x.avg_positive_polarity Avg. polarity of positive words.

x.min_positive_polarity Min. polarity of positive words.

x.max_positive_polarity Max. polarity of positive words.

x.avg_negative_polarity Avg. polarity of negative words.

x.min_negative_polarity Min. polarity of negative words.

x.max_negative_polarity Max. polarity of negative words.

x.title_subjectivity Title subjectivity.

x.title_sentiment_polarity Title polarity.

x.abs_title_subjectivity Absolute subjectivity level.

x.abs_title_sentiment_polarity Absolute polarity level.

y Response variable, log of number of shares (base 10).

References

Fernandes, K., P. Vinagre, and P. Cortez (2015) "A proactive intelligent decision supportsystem for predicting the popularity of online news." Portuguese Conference on Artificial Intelligence, pp. 535-546. Springer.

---

rosnerFEV *Rosner's FEV data.*

Description

This data set consists of 654 observations on youths aged 3 to 19 from East Boston recorded duing the middle to late 1970's. Forced expiratory volume (FEV), a measure of lung capacity, is the variable of interest. We slightly modified the original data, this version only includes the covariates used in our paper.

Usage

```
data(rosnerFEV)
```

Format

A data frame with 654 observations on the following 3 variables.

- `x` Age (years).
- `z` A binary variable indicating whether or not the youth smokes. Nonsmoker is 0. Smoker is 1.
- `y` Forced expiratory volume (liters). Roughly the amount of air an individual can exhale in the first second of a forceful breath.

References

Rosner, B. (1995) "Fundamentals of biostatistics". Duxbury Press: New York.

UPM *Uncertainty Prediction Machine*

Description

An integrated statistical learning framework that converts an ML-procedure into an uncertainty distribution prediction machine (UPM). Using this function, one can extract the estimated conditional density, contrast density, conditional quantile, highest density prediction interval, and finally, can simulated samples.

Usage

```
UPM(X, y, X.test, pivot = NULL, m = c(4, 6), method.ml = "glmnet", LP_smooth = "BIC",
     nsample = NULL, quantile.probs=NULL, credMass = 0.6, centering = TRUE,
     parallel = FALSE, ...)
```

Arguments

`X`	A n-by-d feature matrix
`y`	A length n vector of response.
`X.test`	A k-by-d matrix providing k sets of covariates for target cases to investigate.
`pivot`	Pivot density for computing conditional distribution. This accepts either (i) a function object; or (ii) a vector of sub-samples for y. Set to `NULL` to use the marginal ecdf of y as the pivot.
`m`	An ordered pair (m_1, m_2). m_1 indicates how many LP-nonparametric basis to construct for each column of X, m_2 indicates how many to construct for y.
`method.ml`	Method for estimating the conditional LP-Fourier coefficients. Currently supports these options: subset (lm with subset selection), glmnet, svm (requires `caret`), knn (requires `caret`), gbm (requires `h2o`) and rf (requires `h2o`).
`LP_smooth`	Specifies the method to use for LP coefficient smoothing (AIC or BIC). Uses BIC by default.
`nsample`	Number of relevance samples generated for each case. Leave at `NULL` to disable.
`credMass`	A scalar $[0, 1]$ specifying the mass within the desired coverage of the highest-density prediction interval.
`centering`	Set to TRUE to allow modeling the conditional mean function and obtain the residuals y using the method given in `method.ml`.
`quantile.probs`	Numeric vector of length q for target quantile values. Leave at `NULL` to disable quantile regression.
`parallel`	Use parallel computing for obtaining the relevance samples, mainly used for very huge `nsample`, default is FALSE.
`...`	Extra parameters to pass into other functions. Currently supports the arguments for `caret::knnreg()`, `caret::train()`, `h2o::h2o.gbm()`, `h2o::h2o.randomForest()`.

Value

A list of values containing:

`LP.coef`	A k-by-m matrix giving the conditional LP-coefficients for y residuals given each `X.test`.
`cond.mean`	conditional means for y given each `X.test`.
`y.res`	residuals after modeling conditional mean function, equals to `y` when `centering=FALSE`.
`cond.den`	list of conditional density functions given each `X.test`.
`dhat`	list of contrast density functions d_x for each `X.test`.
`samples`	A matrix with k columns, each column is a set of relevance sample points generated for `X.target`.
`hdi.laser`	list of prediction intervals of y given each `X.test`.
`quantiles`	A k-by-q matrix containing the quantiles for each `X.test`.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

See Also

UPM.gof

Examples

```
data(butterfly)
attach(butterfly)
UPM.out<-UPM(x,y,X.test=2,method.ml='knn',nsample=NULL,centering=FALSE)
##LP coefficients:
UPM.out$LP.coef
##conditional density:
y.axe=seq(-4,4,length.out=1000)
plot(y.axe,UPM.out$cond.den[[1]](y.axe),type="l")
```

UPM.gof — *Goodness-of-fit Diagnostics for UPM.*

Description

This function provides diagnosis for the performance of UPM. It provides a graphical diagnostics and test statistic to check whether the models are congruent with the observed data.

Usage

```
UPM.gof(X, y, m = c(4, 6), method, indx, ...)
```

Arguments

X	A n-by-d feature matrix
y	A length n vector of response.
m	An ordered pair. First number indicates how many LP-nonparametric basis to construct for each column of X, second number indicates how many to construct for y.
method	Method for estimating the conditional LP-Fourier coefficients. Currently supports these options: subset (lm with subset selection), glmnet, svm (requires caret), knn (requires caret), gbm (requires h2o) and rf (requires h2o).
indx	Indices for the observations to be used as holdout set.
...	Extra parameters to pass into UPM.

Value

A list of values containing:

q.residuals	Generalized quantile-residuals for the holdout set.
qdiv	qDIV statistic.
pval	Test p-value.

Author(s)

Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer: Kaijun Wang <kaijunwang.19@gmail.com>

References

Mukhopadhyay, S., and Wang, K (2020) "Statistical Machine Learning: An Integrated Approach". Technical Report.

Index