

Package ‘CVEK’

June 29, 2018

Title Cross-Validated Kernel Ensemble

Version 0.1-0

Date 2018-06-27

Description Using a library of base kernels, it learns a proper generating function from data by directly minimizing the ensemble model’s cross-validation error, therefore guaranteeing robust test for a wide range of data-generating functions.

Depends R (>= 3.0.1), mvtnorm, MASS, psych, limSolve

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Wenying Deng [aut, cre],
Jeremiah Zhe Liu [ctb]

Maintainer Wenying Deng <wdeng@hsph.harvard.edu>

R topics documented:

baseEstimate

Estimating Projection Matrices

Description

Calculate the estimated projection matrices for every kernels in the kernel library.

Usage

```
baseEstimate(size, magn, Y, X1, X2, Kernlist, mode, lambda)
```

Arguments

size	A numeric number specifying the number of observations.
magn	A numeric number specifying the number of kernels in the kernel library.
Y	Reponses of the dataframe.
X1	The first type of factor in the dataframe (could contains several subfactors).
X2	The second type of factor in the dataframe (could contains several subfactors).
Kernlist	The kernel library containing several kernels given by user.
mode	A character string indicating which tuning parameter criteria is to be used.
lambda	A numeric string specifying the range of noise to be chosen. The lower limit of lambda must be above 0.

Details

For a given mode, this function return a list of projection matrices for every kernels in the kernel library and a size*magn matrix indicating errors.

Value

A_hat	A list of projection matrices for every kernels in the kernel library.
error_mat	A size*magn matrix indicating errors.

Author(s)

Wenying Deng

References

Jeremiah Zhe Liu and Brent Coull. Robust Hypothesis Test for Nonlinear Effect with Gaussian Processes. October 2017.

Examples

```
##baseEstimate(size = 50, magn = 3, Y, X1, X2, Kernlist = NULL,
##mode = "loocv", lambda = exp(seq(-5, 5)))
```

dataGenerate

Generating Original Data

Description

Generate original data based on specific kernels.

Usage

```
dataGenerate(size, label_names, method = "rbf", int_effect = 0, l = 1,
p = 2, eps = 0.01)
```

Arguments

size	A numeric number specifying the number of observations.
label_names	A character string indicating all the interior variables included in each predictor.
method	A character string indicating which kernel is to be computed.
int_effect	A numeric number specifying the size of interaction.
l	A numeric number indicating the hyperparameter (flexibility) of a specific kernel.
p	For polynomial, p is the power; for matern, $v = p + 1 / 2$; for rational, $\alpha = p$.
eps	A numeric number indicating the size of noise.

Details

This function generates with a specific kernel. The argument `int_effect` represents the strength of interaction relative to the main effect since all sampled functions have been standardized to have unit norm.

Value

data	A dataframe to be fitted.
------	---------------------------

Author(s)

Wenying Deng

Examples

```
##data <- dataGenerate(size = 50, label_names =
##list(X1 = c("x1", "x2"), X2 = c("x3", "x4")),
##method = "rbf", int_effect = 0, l = 1, p = 2, eps = 0.01)
```

ensemble

Estimating Ensemble Kernel Matrices

Description

Conduct gaussian process regression based on the estimated ensemble kernel matrix.

Usage

```
ensemble(formula, label_names, Kernlist, data, mode, strategy, beta, lambda)
```

Arguments

formula	A symbolic description of the model to be fitted.
label_names	A character string indicating all the interior variables included in each predictor.
Kernlist	The kernel library containing several kernels given by user.
data	A dataframe to be fitted.
mode	A character string indicating which tuning parameter criteria is to be used.
strategy	A character string indicating which ensemble strategy is to be used.
beta	A numeric value specifying the parameter when strategy = "exp".
lambda	A numeric string specifying the range of noise to be chosen. The lower limit of lambda must be above 0.

Details

There are three ensemble strategies available here:

Empirical Risk Minimization

After obtaining the estimated errors $\{\hat{\epsilon}_d\}_{d=1}^D$, we estimate the ensemble weights $u = \{u_d\}_{d=1}^D$ such that it minimizes the overall error

$$\hat{u} = u \in \Delta \argmin \left\| \sum_{d=1}^D u_d \hat{\epsilon}_d \right\|^2 \quad \text{where } \Delta = \{u | u \geq 0, \|u\|_1 = 1\}$$

Then produce the final ensemble prediction:

$$\hat{h} = \sum_{d=1}^D \hat{u}_d h_d = \sum_{d=1}^D \hat{u}_d A_{d, \hat{\lambda}_d} y = \hat{A} y$$

where $\hat{A} = \sum_{d=1}^D \hat{u}_d A_{d, \hat{\lambda}_d}$ is the ensemble matrix.

Simple Averaging

Motivated by existing literature in omnibus kernel, we propose another way to obtain the ensemble matrix by simply choosing unsupervised weights $u_d = 1/D$ for $d = 1, 2, \dots, D$.

Exponential Weighting

Additionally, another scholar gives a new strategy to calculate weights based on the estimated errors $\{\hat{\epsilon}_d\}_{d=1}^D$.

$$u_d(\beta) = \frac{\exp(-\|\hat{\epsilon}_d\|_2^2 / \beta)}{\sum_{d=1}^D \exp(-\|\hat{\epsilon}_d\|_2^2 / \beta)}$$

Then we can calculate the output of gaussian process regression, the solution is given by

$$\begin{aligned} \hat{\beta} &= [1^T (K + \lambda I)^{-1} 1]^{-1} 1^T (K + \lambda I)^{-1} y \\ \hat{\alpha} &= (K + \lambda I)^{-1} (y - \hat{\beta} 1) \end{aligned}$$

where $\beta = \text{intercept}$.

Value

lam	The selected tuning parameter based on the estimated ensemble kernel matrix.
intercept	Estimated bias of the model.
alpha	Estimated coefficients of the estimated ensemble kernel matrix.
K	Estimated ensemble kernel matrix.
u_hat	A vector of weights of the kernels in the library.

Author(s)

Wenying Deng

References

Jeremiah Zhe Liu and Brent Coull. Robust Hypothesis Test for Nonlinear Effect with Gaussian Processes. October 2017.

Xiang Zhan, Anna Plantinga, Ni Zhao, and Michael C. Wu. A fast small-sample kernel independence test for microbiome community-level association analysis. December 2017.

Arnak S. Dalalyan and Alexandre B. Tsybakov. Aggregation by Exponential Weighting and Sharp Oracle Inequalities. In Learning Theory, Lecture Notes in Computer Science, pages 97–111. Springer, Berlin, Heidelberg, June 2007.

Examples

```
##ensemble(formula = Y ~ X1 + X2,
##label_names = list(X1 = c("x1", "x2"), X2 = c("x3", "x4")),
##Kernlist = Kernlist, data = data, mode = "loocv",
##strategy = "erm", beta = 1, lambda = exp(seq(-5, 5)))
```

genericFormula

*From Vectors to Single Variables***Description**

Transform format of predictors from vectors to single variables.

Usage

```
genericFormula(formula, label_names)
```

Arguments

formula	A symbolic description of the model to be fitted.
label_names	A character string indicating all the interior variables included in each predictor.

Value

generic_formula	A symbolic description of the model written in single variables format.
length_main	A numeric value indicating the length of main effects.

Author(s)

Wenying Deng

Examples

```
##generic_formula0 <- genericFormula(formula = Y ~ X1 + X2,
##label_names = list(X1 = c("x1", "x2"), X2 = c("x3", "x4")))
```

infoMat

Computing Information Matrices

Description

Compute information matrices based on block matrices.

Usage

```
infoMat(P0_mat, mat_del = NULL, mat_sigma2 = NULL, mat_tau = NULL)
```

Arguments

P0_mat	Scale projection matrix under REML.
mat_del	Derivative of the scale covariance matrix of Y with respect to delta.
mat_sigma2	Derivative of the scale covariance matrix of Y with respect to sigma2.
mat_tau	Derivative of the scale covariance matrix of Y with respect to tau

Details

This function gives the information value of the interaction strength.

Value

I0	The computed information value.
----	---------------------------------

Author(s)

Wenying Deng

References

Arnab Maity and Xihong Lin. Powerful tests for detecting a gene effect in the presence of possible gene-gene interactions using garrote kernel machines. December 2011.

Examples

```
##I0 <- infoMat(P0_mat, mat_del = drV0_del,
##mat_sigma2 = drV0_sigma2, mat_tau = drV0_tau)
```

kernelGenerate	<i>Generating A Single Kernel</i>
----------------	-----------------------------------

Description

Generate kernels for the kernel library.

Usage

```
kernelGenerate(method = "rbf", Sigma = NULL, l = 1, p = 2)
```

Arguments

method	A character string indicating which kernel is to be computed.
Sigma	The covariance matrix for neural network kernel.
l	A numeric number indicating the hyperparameter (flexibility) of a specific kernel.
p	For polynomial, p is the power; for matern, $\nu = p + 1 / 2$; for rational, $\alpha = p$.

Details

There are seven kinds of kernel available here. For convenience, we define $r = |x - x'|$.

Gaussian RBF Kernels

$$k_{SE}(r) = \exp\left(-\frac{r^2}{2l^2}\right)$$

Matern Kernels

$$k_{Matern}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{l}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}r}{l}\right)$$

Rational Quadratic Kernels

$$k_{RQ}(r) = \left(1 + \frac{r^2}{2\alpha l^2}\right)^{-\alpha}$$

Polynomial Kernels

$$k(x, x') = (x \cdot x')^p$$

We have intercept kernel when $p = 0$, and linear kernel when $p = 1$.

Neural Network Kernels

$$k_{NN}(x, x') = \frac{2}{\pi} \sin^{-1}\left(\frac{2\tilde{x}^T \Sigma \tilde{x}'}{\sqrt{(1 + 2\tilde{x}^T \Sigma \tilde{x})(1 + 2\tilde{x}'^T \Sigma \tilde{x}')}}\right)$$

Value

Kern	A function indicating the generated kernel.
------	---

Author(s)

Wenying Deng

References

The MIT Press. Gaussian Processes for Machine Learning, 2006.

Examples

```
##kernelGenerate(method = "rbf", Sigma = NULL, l = 1, p = 2)

##Kernlist <- NULL
##Kernlist <- c(Kernlist, kernelGenerate('rbf', l = .6))
##Kernlist <- c(Kernlist, kernelGenerate('rbf', l = 1))
##Kernlist <- c(Kernlist, kernelGenerate('rbf', l = 2))
```

kernelTest

Conducting Score Tests for Interaction

Description

Conduct score tests comparing a fitted model and a more general alternative model.

Usage

```
kernelTest(formula, label_names, Kernlist, data = NULL, mode = "loocv",
  strategy = "erm", beta = 1, test = "boot", lambda = exp(seq(-5, 5)),
  B = 100)
```

Arguments

formula	A symbolic description of the model to be fitted.
label_names	A character string indicating all the interior variables included in each predictor.
Kernlist	The kernel library containing several kernels given by user.
data	A dataframe to be fitted.
mode	A character string indicating which tuning parameter criteria is to be used.
strategy	A character string indicating which ensemble strategy is to be used.
beta	A numeric value specifying the parameter when strategy = "exp".
test	A character string indicating which test is to be used.
lambda	A numeric string specifying the range of noise to be chosen. The lower limit of lambda must be above 0.
B	A numeric value indicating times of resampling when test = "boot".

Details

There are two tests available here:

Asymptotic Test

This is based on the classical variance component test to construct a testing procedure for the hypothesis about Gaussian process function.

Bootstrap Test

When it comes to small sample size, we can use bootstrap test instead, which can give valid tests with moderate sample sizes and requires similar computational effort to a permutation test.

Value

pvalue p-value of the test.

Author(s)

Wenying Deng

References

Xihong Lin. Variance component testing in generalised linear models with random effects. June 1997.

Arnab Maity and Xihong Lin. Powerful tests for detecting a gene effect in the presence of possible gene-gene interactions using garrote kernel machines. December 2011.

Petra Bu z̃kova , Thomas Lumley, and Kenneth Rice. Permutation and parametric bootstrap tests for gene-gene and gene-environment interactions. January 2011.

See Also

method: [kernelGenerate](#)

mode: [tuning](#)

strategy: [ensemble](#)

Examples

```
##kernelTest(formula = Y ~ X1 + X2,
##label_names = list(X1 = c("x1", "x2"), X2 = c("x3", "x4")),
##Kernlist, data = data, mode = "loocv", strategy = "erm",
##beta = 1, test = "boot", lambda = exp(seq(-5, 5)))
```

multiScore

Computing Score Test Statistics.

Description

Compute score test statistics.

Usage

```
multiScore(formula_int, label_names, Kernlist, data, mode, strategy, beta,
          lambda)
```

Arguments

formula_int	A symbolic description of the model with interaction.
label_names	A character string indicating all the interior variables included in each predictor.
Kernlist	The kernel library containing several kernels given by user.
data	A dataframe to be fitted.

mode	A character string indicating which tuning parameter criteria is to be used.
strategy	A character string indicating which ensemble strategy is to be used.
beta	A numeric value specifying the parameter when strategy = "exp".
lambda	A numeric string specifying the range of noise to be chosen. The lower limit of lambda must be above 0.

Details

The test statistic is distributed as a scaled Chi-squared distribution.

Value

test_stat	The computed test statistic.
-----------	------------------------------

Author(s)

Wenying Deng

References

Arnab Maity and Xihong Lin. Powerful tests for detecting a gene effect in the presence of possible gene-gene interactions using garrote kernel machines. December 2011.

Examples

```
##multiScore(formula_int = Y ~ X1 + X2 + X1 * X2,
##label_names = list(X1 = c("x1", "x2"), X2 = c("x3", "x4")),
##Kernlist = Kernlist, data = data, mode = "loocv",
##strategy = "erm", test = "boot",
##lambda = exp(seq(-5, 5)))
```

noiseEstimate	<i>Estimating Noise</i>
---------------	-------------------------

Description

An implementation of Gaussian processes for estimating noise.

Usage

```
noiseEstimate(Y, lambda_hat, beta_hat, alpha_hat, K_hat)
```

Arguments

Y	Reponses of the dataframe.
lambda_hat	The selected tuning parameter based on the estimated ensemble kernel matrix.
beta_hat	Estimated bias of the model.
alpha_hat	Estimated coefficients of the estimated ensemble kernel matrix.
K_hat	Estimated ensemble kernel matrix.

Value

sigma2_hat The estimated noise.

Author(s)

Wenying Deng

References

Jeremiah Zhe Liu and Brent Coull. Robust Hypothesis Test for Nonlinear Effect with Gaussian Processes. October 2017.

Examples

```
##sigma2_hat <- noiseEstimate(y, lam, beta0, alpha0, K_gpr)
```

tuning	<i>Calculating Tuning Parameters</i>
--------	--------------------------------------

Description

Calculate tuning parameters based on given criteria.

Usage

```
tuning(Y, K_mat, mode, lambda)
```

Arguments

Y Reponses of the dataframe.
K_mat Estimated ensemble kernel matrix.
mode A character string indicating which tuning parameter criteria is to be used.
lambda A numeric string specifying the range of noise to be chosen. The lower limit of lambda must be above 0.

Details

There are four tuning parameter selections here:

leave-one-out Cross Validation

$$\lambda_{n-CV} = \lambda \in \Lambda \argmin \left\{ \log y^{*T} \left[I - \text{diag}(A_\lambda) - \frac{1}{n} I \right]^{-1} (I - A_\lambda)^2 \left[I - \text{diag}(A_\lambda) - \frac{1}{n} I \right]^{-1} y^* \right\}$$

Akaike Information Criteria

$$\lambda_{AICc} = \lambda \in \Lambda \argmin \left\{ \log y^{*T} (I - A_\lambda)^2 y^* + \frac{2[\text{tr}(A_\lambda) + 2]}{n - \text{tr}(A_\lambda) - 3} \right\}$$

Generalized Cross Validation

$$\lambda_{GCV_c} = \lambda \in \Lambda \argmin \left\{ \log y^{*T} (I - A_\lambda)^2 y^* - 2 \log \left[1 - \frac{\text{tr}(A_\lambda)}{n} - \frac{2}{n} \right]_+ \right\}$$

Generalized Maximum Profile Marginal Likelihood

$$\lambda_{GMPML} = \lambda \in \Lambda \argmin \left\{ \log y^{*T} (I - A_\lambda) y^* - \frac{1}{n-1} \log |I - A_\lambda| \right\}$$

Value

lambda0 The estimated tuning parameter.

Author(s)

Wenying Deng

References

Philip S. Boonstra, Bhramar Mukherjee, and Jeremy M. G. Taylor. A Small-Sample Choice of the Tuning Parameter in Ridge Regression. July 2015.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer Series in Statistics. Springer- Verlag, New York, 2 edition, 2009.

Hiroto Akaike. Information Theory and an Extension of the Maximum Likelihood Principle. In Selected Papers of Hirotugu Akaike, Springer Series in Statistics, pages 199–213. Springer, New York, NY, 1998.

Clifford M. Hurvich and Chih-Ling Tsai. Regression and time series model selection in small samples. June 1989.

Hurvich Clifford M., Simonoff Jeffrey S., and Tsai Chih-Ling. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. January 2002.

Examples

```
##tuning(Y, K_mat = K, mode = "loocv", lambda = exp(seq(-5, 5)))
```