Package 'bnclassify'

June 22, 2015

Title Learning Bayesian Network Classifiers from Data

Description Algorithms for learning Bayesian network classifiers from data.

Version 0.2.0

URL http://github.com/bmihaljevic/bnclassify

BugReports http://github.com/bmihaljevic/bnclassify/issues

Depends R (>= 3.2.0)

Imports
assertthat (>= 0.1),entropy(>= 1.2.0),crossval(>= 1.0.2),graph(>= 1.42.0),matrixStats(>= 0.14.0),pryr(>= 0.1.1),RBGL(
8)

Suggests gRain(>= 1.2-3),gRbase(>= 1.7-
0.1),mlr(>= 2.2),testthat(>= 0.8.1),knitr(>= 1.10.5),microbenchmark(>= 1.4-
2),ParamHelpers(>= 1.5),Rgraphviz(>= 2.8.1),rmarkdown(>= 0.7)

License GPL (>= 2)

Author Mihaljevic Bojan [aut, cre],Bielza Concha [aut],Larranaga Pedro [aut]

Maintainer Mihaljevic Bojan <bmihaljevic@fi.upm.es>

VignetteBuilder knitr

R topics documented:

`as_mlr` *Bridge to* `mlr`

Description

Only `as_mlr` should be called by the user.

Usage

```
makeRLearner.bnc()

as_mlr(x, dag, id = "1")

predictLearner.bnc(.learner, .model, .newdata, ...)
```

Arguments

`x`	A bnc_bn object.
`dag`	A logical. Whether to repeat structure learning on each fold or just parameter learning.
`id`	A character.
`.learner,.task,.subset,.weights,.model,.newdata`	Internal.

`bnc` *Learn a structure and parameters.*

Description

A convenience function to learn the structure and parameters in a single call.

Usage

```
bnc(dag_learner, class, dataset, smooth, dag_args = NULL)
```

Arguments

`dag_learner`	A character. Name of the structure learning function.
`class`	A character. Name of the class variable.
`dataset`	The data frame from which to learn network structure and parameters.
`smooth`	A nonnegative numeric. The smoothing value for Bayesian parameter estimation.
`dag_args`	A list. Optional additional arguments to `dag_learner`.

bnclassify *Algorithms for learning Bayesian network classifiers from data.*

Description

Algorithms for learning Bayesian network classifiers from data.

References

Bielza C and Larra\~naga P (2014), Discrete Bayesian network classifiers: A survey. *ACM Computing Surveys*, **47**(1), Article 5.

Friedman N, Geiger D and Goldszmidt M (1997). Bayesian network classifiers. *Machine Learning*, **29**, pp. 131–163.

car *Car Evaluation Data Set.*

Description

Car Evaluation Data Set.

Format

A `data.frame` with 7 columns and 1728 rows.

Source

http://sourceforge.net/projects/weka/files/datasets/UCIandStatLib/uci-20070111.tar.gz

chowliu *Chow-Liu ODE.*

Description

Chow-Liu ODE.

Usage

```
chowliu(class, dataset, score = "loglik", blacklist = NULL, root = NULL)
```

Arguments

`class`	character
`dataset`	data frame
`score`	character
`blacklist`	character matrix
`root`	character

cv *Stratified cross validation estimate of predictive accuracy.*

Description

Stratified cross validation estimate of predictive accuracy.

Usage

```
cv(x, dataset, k, dag, smooth = NULL)
```

Arguments

x	List of bnc_bn. The classifiers to evaluate.
dataset	The data frame on which to evaluate the classifiers.
k	An integer. The number of folds.
dag	A logical. Whether to repeat structure learning on each fold or just parameter learning.
smooth	A nonnegative numeric. The smoothing value for Bayesian parameter estimation.

Value

A numeric vector. The predictive accuracy of each classifier in x.

family *Gets the parents of a node in the graph*

Description

Gets the parents of a node in the graph

Usage

```
family(x, g)
```

learn_params *Learn the parameters of a Bayesian network structure.*

Description

lpawnb weights the features' CPTs according to the AWNB method.

Usage

```
awnb(class, dataset, trees = 10, bootstrap_size = 0.5)

lp(x, dataset, smooth)

lpawnb(x, dataset, smooth, trees, bootstrap_size)
```

Arguments

dataset	The data frame from which to estimate network parameters.
trees	An integer. The number of bootstrap samples to generate.
bootstrap_size	A numeric. The size of the bootstrap subsample, relative to the size of dataset (given in [0,1]).
x	a bnc_dag object. The Bayesian network structure.
smooth	A nonnegative numeric. The smoothing value for Bayesian parameter estimation.

References

Mark Hall (2004). A decision tree-based attribute weighting filter for naive Bayes. *Knowledge-based Systems*, **20**(2), 120-126.

nb *Returns a naive Bayes network structure.*

Description

Returns a naive Bayes network structure.

Usage

```
nb(class, dataset = NULL, features = NULL)
```

Arguments

class	A character. Name of the class variable.
dataset	The data frame from which to learn the classifier.
features	A character vector. The names of the features. This argument is ignored if dataset is provided.

predict.bnc_bn — *Predicts class label or class posterior probability distributions.*

Description

Predicts class label or class posterior probability distributions.

Usage

```
## S3 method for class 'bnc_bn'
predict(object, newdata, prob = FALSE, ...)
```

Arguments

object	A bnc_bn object.
newdata	A data frame containing observations whose class has to be predicted.
prob	A logical. Whether class posterior probability should be returned.

Details

Ties are resolved randomly. Inference is much slower if newdata contains NAs.

Value

If prob=FALSE, then returns a length-N factor with the same levels as the class variable in x, where N is the number of rows in newdata. Each element of the factor is the most likely class for the corresponding row in newdata. If prob=TRUE, returns a N by C numeric matrix, where C is the number of classes; each row corresponds to the class posterior of the instance.

tan_chowliu — *Learns a tree augmented naive Bayes classifier (TAN).*

Description

Learns a one-dependence Bayesian classifier using Chow-Liu's algorithm. The structure is learned so that either likelihood, the BIC or AIC scores are maximized; the first option corresponds to the well-known tree augmented naive Bayes (Friedman et al., 1997).

Usage

```
tan_cl(class, dataset, score = "loglik", blacklist = NULL, root = NULL)
```

Arguments

class	A character. Name of the class variable.
dataset	The data frame from which to learn the classifier.
score	A character. The score to be maximized. 'loglik', 'bic', and 'aic' return the maximum likelihood, maximum BIC and maximum AIC tree/forest, respectively.
blacklist	A character matrix. Edges that may be blacklisted from the resulting structure.
root	A character. The feature to be used as root of the augmenting tree. Only one feature can be supplied, even in case of an augmenting forest. This argument is optional.

Details

Edges with negative BIC or AIC weights are blacklisted from the final structure. The structure thus might be a forest rather than a tree.

References

Friedman N, Geiger D and Goldszmidt M (1997). Bayesian network classifiers. *Machine Learning*, **29**, pp. 131–163.

---

to_grain *To grain*

Description

To grain

Usage

```
to_grain(x)
```

---

voting *Congress Voting Data Set.*

Description

Congress Voting Data Set.

Format

A data.frame with 17 columns and 435 rows.

Source

http://sourceforge.net/projects/weka/files/datasets/UCIandStatLib/uci-20070111.tar.gz

wrapper *Learns Bayesian network classifiers in a wrapper fashion.*

Description

bsej is the backward *sequential elimination and joining* algorithm whereas fssj is the *forward sequential selection and joining* algorithms for learning a semi-naive Bayes classifier (Pazzani, 1996). tanhc Learns a tree augmented naive Bayes with a greedy hill-climbing search. tanhc is the superparent variant of tanhc.

Usage

```
fssj(class, dataset, k, epsilon = 0.01, smooth = 0.01)

bsej(class, dataset, k, epsilon = 0.01, smooth = 0.01)

tanhc(class, dataset, k, epsilon = 0.01, smooth = 0.01)

tanhc_sp(class, dataset, k, epsilon = 0.01, smooth = 0.01)
```

Arguments

class	A character. Name of the class variable.
dataset	The data frame from which to learn the classifier.
epsilon	A numeric. Minimum absolute improvement required to keep searching.
smooth	A nonnegative numeric. The smoothing value for Bayesian parameter estimation.

References

Pazzani M (1996). Constructive induction of Cartesian product attributes. In *Proceedings of the Information, Statistics and Induction in Science Conference (ISIS-1996)*, pp. 66-77

Koegh E and Pazzani M (2002).Learning the structure of augmented Bayesian classifiers. In *International Journal on Artificial Intelligence Tools*, **11**(4), pp. 587-601.

Index