Package 'knncat'

February 20, 2015

Version 1.2.2

Date 2015-02-05

Title Nearest-neighbor Classification with Categorical Variables

Author Sam Buttrey

Maintainer Sam Buttrey <buttrey@nps.edu>

Description Scale categorical variables in such a way as
to make NN classification as accurate as possible. The code also
handles continuous variables and prior probabilities, and does
intelligent variable selection and estimation of both error rates
and the right number of NN's.

License GPL-2

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-02-06 06:22:24

R topics documented:

`knncat` *Build a knncat classifier*

Description

Build a knncat classifier, which is used for nearest-neighbor classification with categorical variables; continuous are permitted too.

Usage

```
knncat (train, test, k = c(1, 3, 5, 7, 9), xvals = 10, xval.ceil = -1,
    knots = 10, prior.ind = 4, prior, permute = 10, permute.tail = 1,
    improvement = .01, ridge = .003, once.out.always.out = FALSE,
    classcol = 1, verbose = 0)
```

Arguments

train	data frame of training data, with the correct classification in the classcol column
test	data frame of test data (can be omitted). This should have the correct classification in the classcol column, too.
k	vector of choices for number of nn's. Default c(1, 3, 5, 7, 9).
xvals	number of cross-validations to use to find the best model size and number of nn's. Default 10.
xval.ceil	Maximum number of variables to add. -1 = Use the smallest number from any xval; 0 = use the smallest number from the first xval; >= 0, use that.
knots	vector of number of knots for numeric variables. Reused if necessary. Default: 10 for each.
prior.ind	Integer telling how to compute priors. 1 = estimated from training set; 2 = all equal; 3 = supplied in "prior"; 4 = ignored. Default: 4.
prior	Numeric vector, one entry per unique element in the training set's classcol column, giving prior probabilities. Ignored unless prior.ind = 3; then they're normalized to sum to 1 and each entry must be strictly > 0.
permute	Number of permutations for variable selection. Default: 10.
permute.tail	A variable fails the permutation test if permute.tail or more permutations do better than the original. Default: 1.
improvement	Minimum improvement for variable selection. Ignored unless present and permute missing, or permute = 0; then default = .01.
ridge	Amount by which to "ridge" the W matrix for numerical stability. Default: .003.
once.out.always.out	if TRUE, a variable that fails a permutation test or doesn't improve by enough is excluded from further consideration during that cross-validation run. Default FALSE.
classcol	Column with classification in it. Default: 1.
verbose	Controls level of diagnostic output. Higher numbers produce more output, sometimes 'way too much. 0 produces no output; 1 gives progress report for xvals. Default: 1.

Details

A knncat classifier converts categorical labels into real numbers (phi) so as to produce a good k-nearest neighbor classifier. Continuous variables are handled by means of knots, in a manner similar to the linear spline representation. Variable selection is done by a permutation test, or by setting an "improvement" cutoff; error rate estimation is done by cross-validation. After the cross-validations are done, we choose the best value of k from among those proposed and the "best" number of variables, then make one more pass through all the data to estimate the phis.

Value

A list of S3 class knncat, containing the following entries:

cdata	A vector with one entry for each of the columns of train, except the classification column, with value 1 if that column was used in the final classifier, and 0 otherwise.
phi	A list with the phi's. Each element of the list has, as its name, the name of a column of train; the values of the element are the phi's, and the names of that element are the levels of the variable. For numeric variables, these names are "knot.1", "knot.2" etc.
k	The vector of k's to be tried, as passed in.
best.k	The best k selected.
misclass.mat	A matrix, number of classes * number of classes, whose columns give the correct classifications and rows, the estimates.
prior.ind	Method used to compute the prior, as passed in.
prior	A numeric vector, one per class, giving the prior probabilties, as computed by the program according to prior.ind.
status	Return value from the program. 0 = no error.
misclass.type	Type of misclass.mat. "train" means misclass.rate came from the training set; "test," from the test set.
train	Name of training set at build time.
vars	Vector of names of columns actually used in model.
knots.vec	Vector of numbers of knots, as passed in.
build	Named vector holding five of the arguments used at build time: permute, improvement, ridge, once.out.always.out, and xvals
missing	Vector of values with which to replace missing values. These are the most common values for categorical variables, and the means for continuous ones.
knot.values	List of knot locations, one element for each continuous variable.

Author(s)

Samuel E. Buttrey, <buttrey@nps.edu>

References

Buttrey, S.E., Nearest-neighbor classification with categorical variables, Comp. Stat. Data Analysis 28 (1998), 157-169.

Examples

```
## Not run:
data ("synth.tr", package="MASS")
data ("synth.te", package="MASS")
syncat <- knncat (synth.tr, classcol=3)
syncat
```

```
Train set misclass rate: 12.8

synpred <- predict (syncat, synth.tr, synth.te, train.classcol=3,
                    newdata.classcol=3)
table (synpred, synth.te$yc)

synpred 0   1
      0 460  91
      1  40 409
#
# Or do the whole thing in one pass:
#

knncat (synth.tr, synth.te, classcol=3)
Test set misclass rate: 13.1

## End(Not run)
```

`plot.knncat` *Plot a knncat classifier*

Description

Plot a knncat classifier

Usage

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

Arguments

`x`	Knncat object, from knncat
`...`	Other arguments, currently ignored

Details

This plot shows all the estimated numnbers associated with each level of a variable (or knot, for a continuous variable) in a knncat classifier.

Value

None.

Author(s)

Samuel E. Buttrey, <buttrey@nps.edu>

predict.knncat *Prediict on a knncat classifier*

Description

Produce predictions for a knncat classifier

Usage

```
## S3 method for class 'knncat'
predict(object, train, newdata,
train.classcol=1, newdata.classcol=1, return.classes=TRUE,
more=FALSE, verbose = 0, ...)
```

Arguments

object	Knncat object, from knncat
train	Training set used to build classifier
newdata	New data on which to make predictions
train.classcol	Column number for classification in training set. Default: 1
newdata.classcol	Column number for classification in newdata set. Default: 1. If <= 0, new data has no classifications.
return.classes	Logical; if TRUE, return a vector of classifications of the newdata set. Default: TRUE
more	Logical; if TRUE, also print error rate. Default: FALSE
verbose	Level of verbosity for debugging. Default: 0
...	Other arguments, currently ignored

Details

This prints the misclassification rate from the knncat classifier, together with an indication as to whether it was based on a training or test set.

Value

None.

Author(s)

Samuel E. Buttrey, <buttrey@nps.edu>

`print.knncat` *Print a knncat classifier*

Description

Print the misclassification rate for a knncat classifier

Usage

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

Arguments

`x`	Knncat object, from knncat
`...`	Other arguments, currently ignored

Details

This prints the misclassification rate from the knncat classifier, together with an indication as to whether it was based on a training or test set.

Value

None.

Author(s)

Samuel E. Buttrey, <buttrey@nps.edu>

Index