Package ‘predbayescor’ documentation

of

April 10, 2008

Version 1.1-4

Title Classification rule based on Bayesian naive Bayes models with feature selection bias corrected

Author Longhai Li <longhai@math.usask.ca>

Maintainer Longhai Li <longhai@math.usask.ca>

Depends R (>= 2.5.1)

Description This software is used to predict the binary response based on high dimensional features, for example gene expression data. The data are modelled with Bayesian naive Bayes models. When a large number of features are available, one may like to select only a subset of features to use, typically those features strongly correlated with the response in training cases. Such a feature selection procedure is however invalid since the relationship between the response and the features will appear stronger. This package provides a way to avoid this bias and yields well-calibrated prediction for the test cases.

License GPL (>=2)

URL `http://www.r-project.org`, `http://math.usask.ca/~longhai`

R topics documented:

`predict_bayes` *Classification rule based on Bayesian naive Bayes models with feature selection bias corrected*

Description

`predict_bayes` predicts the binary response based on high dimemsional binary features modeled by Bayesian naive Bayes models. It also accepts real values but they will be converted into binary by thresholding at the medians estimated from the data. A smaller number of features can be selected based on the correlations with the response. The bias due to the selection procedure can be corrected. `cv.bayes` is the short-cut function for cross-validation with `predict_bayes`.

Usage

```
predict_bayes(
              test,train,is.binary.features=FALSE,k,
              subset.sel=1:nrow(train),
              theta0=0,no.theta=20,
              alpha.shape=0.5,alpha.rate=5,no.alpha=5,
              correct=TRUE,no.theta.adj=20)

cv.bayes(
          data,is.binary.features=FALSE,no.folds=10,k,
          theta0=0,no.theta=20,
          alpha.shape=0.5,alpha.rate=5,no.alpha=5,
          correct=TRUE,no.theta.adj=20)
```

Arguments

`test` a test data, a matrix, i.e. the data for which we want to predict the responses. The row stands for the cases. The first column is the binary response, which could be NA if they are missing.

`train` a training data, of the same format as `test`

`data` a data used in cross-validation, of the same format as `test`

`no.folds` the number of blocks the data is divided into in cross-validation

`is.binary.features` the indicator whether the features are binary

`k` the number of features retained

`subset.sel` the indice of training cases used to select features

`theta0` the prior of "theta" is uniform over (`theta0`,`1-theta0`)

`no.theta` the parameter in Simpson's rule used to evaluate the integration w.r.t. "theta". The integrant is evaluated at 2*(no.theta)+1 points.

`alpha.shape` the shape parameter of the inverse Gamma, which is the prior distribution of "alpha"

`alpha.rate`	the rate parameter of the inverse Gamma, as above
`no.alpha`	the number of "alpha"'s used in mid-point rule, which is used to approximate the integral with respect to "alpha".
`correct`	the indicator whether the correction method shall be applied
`no.theta.adj`	a parameter of Simpson's rule, which is used to evaluate the integration with respect to "theta" in calculating the adjustment factor

Value

`prediction`	a matrix showing the detailed prediction result: the 1st column being the true responses, the 2nd being the predicted responses, the 3rd being the predictive probabilities of class 1 and the 4th being the indicator whether wrong prediction is made.
`amlp`	the average minus log probabilities
`error.rate`	the ratio of wrong prediction
`mse`	the average square error of the predictive probabilities
`summary.pred`	tabular display of the predictive probabilities and the actual fraction of class 1.
`alpha.prior.adj.post`	a matrix showing the detailed information about the "alpha"'s, the 1st column being the values of "alpha"'s, the 2nd being the adjustment factor, i.e. probability that feature is discarded by the cutoff used in the feature selection, the 3rd being the log of the 2nd column times the numbers of discarded features, the 4th being the posterior probabilities
`features.selected`	The features selected using correlation criterion

References

http://math.usask.ca/~longhai/doc/naivebayes/naivebayes.abstract.html

See Also

`gendata.bayes`

Examples

```
#generate a dataset
d <- gendata.bayes(100,100,500,500,1000,400)

#do prediction with correction applied
pred.d.cor <- predict_bayes(d$test,d$train,TRUE,10,,0,20,0.5,5,20,TRUE,40)

#do prediction without correction applied
pred.d.uncor <- predict_bayes(d$test,d$train,TRUE,10,,0,20,0.5,5,20,FALSE,40)

#do 5-fold cross-validation on the training data with correction applied
cv.dtr.cor <- cv.bayes(d$train,TRUE,5,10,0,20,0.5,5,20,TRUE,40)
```

gendata.bayes *Generate binary data with Bayesian naive Bayes Models*

Description

"gendata.bayes" generates data (both training and test data) with Bayesian naive Bayes model. The prior distribution of "theta" is uniform(0,1). The value of "alpha" is given by argument alpha, which controls the the overall relationship between the response and the predictor variables.

Usage

```
gendata.bayes(n0,n1,m0,m1,p,alpha)
```

Arguments

`n0`	the number of class 0 in training data
`n1`	the number of class 1 in training data
`m0`	the number of class 0 in test data
`m1`	the number of class 1 in test data
`p`	the number of features
`alpha`	a parameter controlling the dependency between the features and the response

Value

`train`	the training data, with the row standing for the cases and the first column being the response
`test`	the test data, of the same format as "train"

See Also

`predict_bayes`

evaluate_by_loss *calculating the total loss of prediction results*

Description

Calculates the average loss of predictions based on threshold with `threshold`. Note that this threshold has 1-1 mapping with the ratio of the loss of assigning 0 to 1 to the loss of assigning 1 to 0: threshold=1-1/(1+ratio).

Usage

```
evaluate_by_loss ( y.true, pred.prob, threshold=0.5)
```

Arguments

`y.true` a vector containing the true response.

`pred.prob` a vector containing the predictive probabilities.

`threshold` When predictive probability is greater than `threshold`, the response is predicted as 1.

Value

`loss` the average loss, with attrib "sd" storing the estimate of the standard error of this loss.

`predbayescor-internal` *Internal Functions*

Description

Internal Functions. Type function name directly to see the definition of this function.

See Also

`predict_bayes`

Index