Package ‘smac’

November 1, 2014

Type Package

Title Sparse Multi-category Angle-Based Large-Margin Classifiers

Version 1.0

Date 2014-10-29

Author Chong Zhang, Guo Xian Yau, Yufeng Liu

Maintainer Chong Zhang <chong.zhang@uwaterloo.ca>

Description This package provides a solution path for L1-penalized angle-based classification. Three loss functions are implemented in smac, including the deviance loss in logistic regression, the exponential loss in boosting, and the proximal support vector machine loss.

License GPL-2

LazyLoad yes

R topics documented:

`cvsmac` *A cross validation function for smac.*

Description

This function is used to perform cross validation based on a training data set to select the best tuning parameter.

Usage

```
cvsmac(x,y,kfold=5,lambda=NULL,nlambda=100,lambda.min=NULL,seed=0,weight=NULL,...)
```

Arguments

`x` The usage of this argument is the same as that in the "smac" function.

`y` The usage of this argument is the same as that in the "smac" function.

`kfold` The number of subsamples for cross validation. This number should be less than the sample size of the class with the fewest obervations. A warning will be given if kfold is too large. Default is 5.

`lambda` The usage of this argument is the same as that in the "smac" function.

`nlambda` The usage of this argument is the same as that in the "smac" function.

`lambda.min` The usage of this argument is the same as that in the "smac" function.

`seed` The seed for generating the random split of the training data set. Default is 0.

`weight` The usage of this argument is the same as that in the "smac" function.

`...` Other arguments used by function "smac" that are not specified.

Value

`lambda` The sequence of tuning parameters used in cross validation. Notice that the lambdas will be in a decreasing order.

`beta0` The estimated intercepts with respect to each tuning parameter lambda.

`beta` The estimated parameters of the predictors with respect to each tuning parameter lambda.

`error` The total number of misclassifications with respect to each tuning parameter lambda. If weight is specified, each miscalssification is multiplied by its corresponding weight.

`best.lambda` The sequence of lambda values that have the smallest cross validation error.

`best.beta0` The intercepts that correspond to best.lambda.

`best.beta` The parameters that correspond to best.lambda.

`model` The fitted model with respect to each tuning parameter, using the entire training data set.

`min.error` The minimum error in cross validation.

Author(s)

Chong Zhang, Guo Xian Yau and Yufeng Liu

References

C. Zhang and Y. Liu (2014). Multicategory Angle-based Large-margin Classification. *Biometrika*, 101(3), 625-640.

See Also

`smac`,`predict.cvsmac`

Examples

```
data(ex1.data)
cvsmac(ex1.data$ex1.x,ex1.data$ex1.y,loss="p",nlambda=30)
```

ex1.data *Toy example data set used in smac*

Description

This is a simulated toy example data set to illustrate the functionality of smac.

Usage

```
data(ex1.data)
```

Format

Two matrices consisting of predictors, and their corresponding label vectors.

`ex1.x` The training matrix consisting of 2 predictors and 100 observations.

`ex1.y` The label for the training data.

`ex1.new.x` The testing matrix consisting of 2 predictors and 100 observations.

`ex1.new.y` The label for the testing data.

References

C. Zhang and Y. Liu (2014). Multicategory Angle-based Large-margin Classification. *Biometrika*, 101(3), 625-640.

Examples

```
data(ex1.data)
dim(ex1.data$ex1.x)
ex1.data$ex1.y
dim(ex1.data$ex1.new.x)
ex1.data$ex1.new.y
```

predict.cvsmac *A function that provides class label prediction and class conditional probability estimation for objects returned by the "cvsmac" function.*

Description

This function provides prediction on a test data set using the obtained classifier from a call of the cvsmac function.

Usage

```
## S3 method for class 'cvsmac'
predict(object,new.x = NULL,...)
```

Arguments

`object` An object returned by a call from the function "cvsmac".

`new.x` The predictor matrix used for prediction. If not specified, the training matrix x will be used.

`...` Not used.

Value

`new.x` The new testing matrix used for prediction.

`best.lambda` The sequence of lambda values that have the smallest cross validation error in the training data set.

`best.beta0` The beta0 values that correspond to the best.lambda sequence.

`best.beta` The beta values that correspond to the best.lambda sequence.

`best.pred.y` The predicted labels for new.x.

`best.pred.prob` A list of data.frame containing predicted class conditional probabilities. Each data.frame corresponds to a value in the best.lambda object. Each row of the data.frames corresponds to an observation in new.x, and each column represents a class, with the column name as the class label.

`call` The function call that returns this result.

Author(s)

Chong Zhang, Guo Xian Yau and Yufeng Liu

References

C. Zhang and Y. Liu (2014). Multicategory Angle-based Large-margin Classification. *Biometrika*, 101(3), 625-640.

See Also

`cvsmac`, `predict.smac`

Examples

```
data(ex1.data)
a=cvsmac(ex1.data$ex1.x,ex1.data$ex1.y,loss="p",nlambda=30)
predict(a,ex1.data$ex1.new.x)
```

`predict.smac` *A function that provides class label prediction and class conditional probability estimation for objects returned by the "smac" function.*

Description

This function provides prediction on a test data set using the obtained classification model from a call of the smac function.

Usage

```
## S3 method for class 'smac'
predict(object,new.x=NULL,lambda=NULL,...)
```

Arguments

`object` An object returned by the "smac" function.

`new.x` The new predictor matrix. The number and order of predictors in new.x should be the same as those of x, which is used in the smac function. If not specified, the program uses the training x matrix as the prediction object.

`lambda` The lambda values on which the user wants to predict. If not specified, the program will use the lambda values returned by the smac function.

`...` Not used.

Value

`new.x` The predictor matrix used for prediction.

`lambda` The lambda values used for prediction.

`fitted.beta0` The predicted intercept for each lambda value.

`fitted.beta` A list of predicted matrices of parameters for each lambda value. The matrices in the list correspond to the lambda values in the returned lambda object in orders. Each row of a matrix corresponds to a predictor, and the name of the predictor is recorded as the row name. Note that a predictor does not affect the class label if and only if all elements in that corresponding row are 0.

`pred.y` A list of predicted labels for each observation in new.x. A prediction is made for each lambda value in the returned lambda object.

`pred.prob` A list of data.frames containing predicted class conditional probabilities. Each data.frame object corresponds to a value in the lambda object. Each row of the data.frames corresponds to an observation in new.x, and each column represents a class, with the column name as the class label.

Author(s)

Chong Zhang, Guo Xian Yau and Yufeng Liu

References

C. Zhang and Y. Liu (2014). Multicategory Angle-based Large-margin Classification. *Biometrika*, 101(3), 625-640.

See Also

`smac`

Examples

```
data(ex1.data)
a=smac(ex1.data$ex1.x,ex1.data$ex1.y,loss="p",nlambda=30)
predict(a,ex1.data$ex1.new.x)
```

`smac` *Classification function that provides solution path to Multicategory Angle-based large-margin Classifiers (MAC) with L1 penalty*

Description

A classifier that works under the structure of MAC (Zhang and Liu, 2014) with linear learning and the L1 penalty for variable selection.

Usage

```
smac(x,y,loss=c("logistic", "psvm", "boost"),weight=NULL,nlambda = 100,
lambda.min=ifelse(nobs < np, 0.05, 1e-03),lambda = NULL,
standardize = TRUE, epsilon = 1e-05)
```

Arguments

`x` The x matrix/data.frame for the training dataset. Columns represent the covariates, and rows represent the instances. There should be no NA/NaN values in x.

`y` The labels for the training dataset.

`loss` The binary large margin classifer loss function to be used. By default the program uses the logistic loss. Exponential loss in boosting and squared loss in proximal support vector machines are also available. "l" or "logi" for logistic loss, "b" or "boost" for boosting loss, and "p" or "psvm" for squared loss in proximal support vector machines.

`weight` The weight vector for each observation. By default the program uses equal weights for all observations.

`nlambda` The number of lambda values in a solution path, if the user does not specify which lambdas to use. Default is 100.

`lambda.min` In a classification problem where the user does not provide a list of lambda values, the program will automatically find the smallest lambda value that makes all the estimated parameters 0 as a starting lambda. Then the program will create a solution path for a list of lambda values based on the starting lambda (this starting lambda is in fact the largest lambda in the solution path). This option specifies how small the last lambda is compared to the starting lambda in terms of ratios. By default if the number of observations is larger than the number of parameters, the smallest lambda in the solution path is set to be 1/1,000 of the starting lambda, and is set to be 1/20 otherwise. The program then chooses nlambda's of lambda values between the starting lambda and the last lambda, based on an even split of log(lambda) values.

`lambda` The user specified lambda values. If used, the options nlambda and lambda.min will be ignored.

`standardize` Whether the input x should be standardized or not. Default is TRUE (standardize).

`epsilon` Convergence threshold in coordinate descent circling algorithm. The smaller epsilon is, the more accurate the final model is, and the more time it takes for calculation. Default is 1e-5.

Value

`All` All arguments that are used are recorded.

`k` Number of classes in the classification problems.

`x.name` The column names of x.

`y.name` The class names of y.

`lambda` The lambda vector of all lambdas in the solution path.

`beta0` A list of the intercepts of the classification function. Each vector in the list corresponds to the lambda in the solution path in order.

`beta` A list of matrices containing the estimated parameters of the classification function. Each matrix in the list corresponds to the lambda value in the solution path in order. For one single matrix, the rows correspond to a specific predictor, whose name is recorded as the row name. Note that a predictor does not have a significant effect on the label if and only if all elements in its corresponding row are 0.

`loss` The loss function used.

`way` A numeric value specifying if the user provides the lambda values in the solution path (2), or not (1). This return is mainly used in the prediction function.

`call` The call of smac.

Author(s)

Chong Zhang, Guo Xian Yau and Yufeng Liu

References

C. Zhang and Y. Liu (2014). Multicategory Angle-based Large-margin Classification. *Biometrika*, 101(3), 625-640.

See Also

`predict.smac`

Examples

```
data(ex1.data)
smac(ex1.data$ex1.x,ex1.data$ex1.y,loss="p",nlambda=30)
```

Index