Package 'evclass'

June 21, 2016

Type Package

Title Evidential Distance-Based Classification

Version 1.0.1

Date 2016-06-21

Author Thierry Denoeux

Maintainer Thierry Denoeux <tdenoeux@utc.fr>

Description Different evidential distance-based classifiers, which provide outputs in the form of Dempster-Shafer mass functions. The methods are: the evidential K-nearest neighbor rule and the evidential neural network.

License GPL-3

Depends R (>= 3.1.0)

Imports FNN

LazyData TRUE

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-21 17:42:52

R topics documented:

EkNNfit *Training of the EkNN classifier*

Description

EkNNfit optimizes the parameters of the EkNN classifier.

Usage

```
EkNNfit(x, y, K, param = NULL, alpha = 0.95,
  lambda = 1/max(as.numeric(y)), optimize = TRUE, options = list(maxiter =
  300, eta = 0.1, gain_min = 1e-06, disp = TRUE))
```

Arguments

x	Input matrix of size n x d, where n is the number of objects and d the number of attributes.
y	Vector of class labels (of length n). May be a factor, or a vector of integers.
K	Number of neighbors.
param	Initial parameters (default: NULL).
alpha	Parameter α (default: 0.95)
lambda	Parameter of the cost function. If lambda=1, the cost function measures the error between the plausibilities and the 0-1 target values. If lambda=1/M, where M is the number of classes (default), the piginistic probabilities are considered in the cost function. If lambda=0, the beliefs are used.
optimize	Boolean. If TRUE (default), the parameters are optimized.
options	A list of parameters for the optimization algorithm: maxiter (maximum number of iterations), eta (initial step of gradient variation), gain_min (minimum gain in the optimisation loop), disp (Boolean; if TRUE, intermediate results are displayed during the optimization).

Details

If the argument param is not supplied, the function EkNNinit is called.

Value

A list with five elements:

param The optimized parameters.

cost Final value of the cost function.

err Leave-one-out error rate.

ypred Leave-one-out predicted class labels.

m Leave-one-out predicted mass functions. The first M columns correspond to the mass assigned to each class. The last column corresponds to the mass assigned to the whole set of classes.

Author(s)

Thierry Denoeux.

References

T. Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics, 25(05):804–813, 1995.

L. M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 28(2):263–271,1998.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

EkNNinit, EkNNval

Examples

```
## Iris dataset
data(iris)
x<-iris[,1:4]
y<-iris[,5]
fit<-EkNNfit(x,y,K=5)
```

EkNNinit *Initialization of parameters for the EkNN classifier*

Description

EkNNinit returns initial parameter values for the EkNN classifier.

Usage

```
EkNNinit(x, y, alpha = 0.95)
```

Arguments

x	Input matrix of size n x d, where n is the number of objects and d the number of attributes.
y	Vector of class lables (of length n). May be a factor, or a vector of integers.
alpha	Parameter α.

Details

Each parameter γ_k is set ot the inverse of the square root of the mean Euclidean distances wihin class k. Note that γ_k here is the square root of the γ_k as defined in (Zouhal and Denoeux, 1998). By default, parameter alpha is set to 0.95. This value normally does not have to be changed.

Value

A list with two elements:

gamma Vector of parameters γ_k, of length c, the number of classes.

alpha Parameter α, set to 0.95.

Author(s)

Thierry Denoeux.

References

T. Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics, 25(05):804–813, 1995.

L. M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 28(2):263–271,1998.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

EkNNfit, EkNNval

Examples

```
## Iris dataset
data(iris)
x<-iris[,1:4]
y<-iris[,5]
param<-EkNNinit(x,y)
param
```

EkNNval *Classification of a test set by the EkNN classifier*

Description

EkNNval classifies instances in a test set using the EkNN classifier.

Usage

```
EkNNval(xtrain, ytrain, xtst, K, ytst = NULL, param = NULL)
```

Arguments

xtrain	Matrix of size ntrain x d, containing the values of the d attributes for the training data.
ytrain	Vector of class labels for the training data (of length ntrain). May be a factor, or a vector of integers.
xtst	Matrix of size ntst x d, containing the values of the d attributes for the test data.
K	Number of neighbors.
ytst	Vector of class labels for the test data (optional). May be a factor, or a vector of integers.
param	Parameters, as returned by EkNNfit.

Details

If class labels for the test set are provided, the test error rate is also returned. If parameters are not supplied, they are given default values by EkNNinit.

Value

A list with three elements:

m Predicted mass functions for the test data. The first M columns correspond to the mass assigned to each class. The last column corresponds to the mass assigned to the whole set of classes.

ypred Predicted class labels for the test data.

err Test error rate.

Author(s)

Thierry Denoeux.

References

T. Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics, 25(05):804–813, 1995.

L. M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 28(2):263–271,1998.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

EkNNinit, EkNNfit

Examples

```
## Iris dataset
data(iris)
train<-sample(150,100)
xtrain<-iris[train,1:4]
ytrain<-iris[train,5]
xtst<-iris[-train,1:4]
ytst<-iris[-train,5]
K<-5
fit<-EkNNfit(xtrain,ytrain,K)
test<-EkNNval(xtrain,ytrain,xtst,K,ytst,fit$param)
```

`evclass` *evclass: A package for evidential classification*

Description

The evclass package currently contains functions for two evidential classifiers: the evidential K-nearest neighbor (EK-NN) rule (Denoeux, 1995; Zouhal and Denoeux, 1998) and the evidential neural network (Denoeux, 2000). In contrast with classical statistical classifiers, evidential classifier quantify the uncertainty of the classification using Dempster-Shafer mass functions.

Details

The main functions are: `EkNNinit`, `EkNNfit` and `EkNNval` for the initialization, training and evaluation of the EK-NN classifier, and `proDSinit`, `proDSfit` and `proDSval` for the evidential neural network classifier.

References

T. Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man and Cybernetics, 25(05):804–813, 1995.

T. Denoeux. A neural network classifier based on Dempster-Shafer theory. IEEE Trans. on Systems, Man and Cybernetics A, 30(2):131–150, 2000.

L. M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 28(2):263–271,1998.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

`EkNNinit`, `EkNNfit`, `EkNNval`, `proDSinit`, `proDSfit`, `proDSval`.

`glass` *Glass dataset*

Description

This data set contains the description of 214 fragments of glass originally collected for a study in the context of criminal investigation. Each fragment has a measured reflectivity index and chemical composition (weight percent of Na, Mg, Al, Si, K, Ca, Ba and Fe). As suggested by Ripley (1994), 29 instances were discarded, and the remaining 185 were re-grouped in four classes: window float glass (70), window non-float glass (76), vehicle window glass (17) and other (22). The data set was split randomly in a training set of size 89 and a test set of size 96.

Usage

```
data(glass)
```

Format

A list with two elements:

x The 185 x 9 object-attribute matrix.

y A 185-vector containing the class labels.

References

P. M. Murphy and D. W. Aha. UCI Reposition of machine learning databases. [Machine readable data repository]. University of California, Departement of Information and Computer Science, Irvine, CA.

B.D.Ripley, Flexible nonlinear approaches to classification, in "From Statistics to Neural Networks", V. Cherkassly, J. H. Friedman, and H. Wechsler, Eds., Berlin, Germany: Springer-Verlag, 1994, pp. 105–126.

T. Denoeux. A neural network classifier based on Dempster-Shafer theory. IEEE Trans. on Systems, Man and Cybernetics A, 30(2):131–150, 2000.

Examples

```
data(glass)
table(glass$y)
```

ionosphere *Ionosphere dataset*

Description

This dataset was collected by a radar system and consists of phased array of 16 high-frequency antennas with a total transmitted power of the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not. There are 351 instances and 34 numeric attributes. The first 175 instances are training data, the rest are test data. This version of dataset was used by Zouhal and Denoeux (1998).

Usage

```
data(ionosphere)
```

Format

A list with two elements:

x The 351 x 18 object-attribute matrix.

y A 351-vector containing the class labels.

References

P. M. Murphy and D. W. Aha. UCI Reposition of machine learning databases. [Machine readable data repository]. University of California, Departement of Information and Computer Science, Irvine, CA.

L. M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 28(2):263–271,1998.

Examples

```
data(ionosphere)
table(vehicles$y)
```

proDSfit *Training of the evidential neural network classifier*

Description

`proDSfit` performs parameter optimization for the evidential neural network classifier.

Usage

```
proDSfit(x, y, param, lambda = 1/max(as.numeric(y)), mu = 0,
  optimProto = TRUE, options = list(maxiter = 500, eta = 0.1, gain_min =
  1e-04, disp = 10))
```

Arguments

x	Input matrix of size n x d, where n is the number of objects and d the number of attributes.
y	Vector of class lables (of length n). May be a factor, or a vector of integers.
param	Initial parameters (see link{proDSinit}).
lambda	Parameter of the cost function. If lambda=1, the cost function measures the error between the plausibilities and the 0-1 target values. If lambda=1/M, where M is the number of classes (default), the piginistic probabilities are considered in the cost function. If lambda=0, the beliefs are used.
mu	Regularization hyperparameter (default=0).
optimProto	Boolean. If TRUE, the prototypes are optimized (default). Otherwise, they are fixed.
options	A list of parameters for the optimization algorithm: maxiter (maximum number of iterations), eta (initial step of gradient variation), gain_min (minimum gain in the optimisation loop), disp (integer; if >0, intermediate results are displayed every disp iterations).

Details

The prototypes are initialized by the k-means algorithms. The initial membership values u_{ik} of each prototype p_i to class ω_k are normally defined as the proportion of training samples from class ω_k in the neighborhood of prototype p_i. If arguments crisp and nprotoPerClass are set to TRUE, the prototypes are assigned to one and only one class.

Value

A list with three elements:

param Optimized network parameters.

cost Final value of the cost function.

err Training error rate.

Author(s)

Thierry Denoeux.

References

T. Denoeux. A neural network classifier based on Dempster-Shafer theory. IEEE Trans. on Systems, Man and Cybernetics A, 30(2):131–150, 2000.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

proDSinit, proDSval

Examples

```
## Glass dataset
data(glass)
xapp<-glass$x[1:89,]
yapp<-glass$y[1:89]
xtst<-glass$x[90:185,]
ytst<-glass$y[90:185]
## Initialization
param0<-proDSinit(xapp,yapp,nproto=7)
## Training
fit<-proDSfit(xapp,yapp,param0)
```

---

proDSinit *Initialization of parameters for the evidential neural network classifier*

Description

proDSinit returns initial parameter values for the evidential neural network classifier.

Usage

```
proDSinit(x, y, nproto, nprotoPerClass = FALSE, crisp = FALSE)
```

Arguments

x	Input matrix of size n x d, where n is the number of objects and d the number of attributes.
y	Vector of class lables (of length n). May be a factor, or a vector of integers.
nproto	Number of prototypes.
nprotoPerClass	Boolean. If TRUE, there are nproto prototypes per class. If FALSE (default), the total number of prototypes is equal to nproto.
crisp	Boolean. If TRUE, the prototypes have full membership to only one class. (Available only is nprotoPerClass=TRUE).

Details

The prototypes are initialized by the k-means algorithms. The initial membership values u_{ik} of each prototype p_i to class ω_k are normally defined as the proportion of training samples from class ω_k in the neighborhood of prototype p_i. If arguments crisp and nprotoPerClass are set to TRUE, the prototypes are assigned to one and only one class.

Value

A list with four elements containg the initialized network parameters

alpha Vector of length r, where r is the number of prototypes.

gamma Vector of length r

beta Matrix of size (r,M), where M is the number fo classes.

W Matrix of size (r,d), containing the prototype coordinates.

Author(s)

Thierry Denoeux.

References

T. Denoeux. A neural network classifier based on Dempster-Shafer theory. IEEE Trans. on Systems, Man and Cybernetics A, 30(2):131–150, 2000.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

proDSfit, proDSval

Examples

```
## Glass dataset
data(glass)
xapp<-glass$x[1:89,]
yapp<-glass$y[1:89]
param0<-proDSinit(xapp,yapp,nproto=7)
param0
```

---

proDSval *Classification of a test set by the evidential neural network classifier*

Description

proDSval classifies instances in a test set using the evidential neural network classifier.

Usage

```
proDSval(x, param, y = NULL)
```

Arguments

x	Matrix of size n x d, containing the values of the d attributes for the test data.
param	Neural network parameters, as provided by proDSfit.
y	Optimnal vector of class labels for the test data. May be a factor, or a vector of integers.

Details

If class labels for the test set are provided, the test error rate is also returned.

Value

A list with three elements:

m Predicted mass functions for the test data. The first M columns correspond to the mass assigned to each class. The last column corresponds to the mass assigned to the whole set of classes.

ypred Predicted class labels for the test data.

err Test error rate (if the class label of test data has been provided).

Author(s)

Thierry Denoeux.

References

T. Denoeux. A neural network classifier based on Dempster-Shafer theory. IEEE Trans. on Systems, Man and Cybernetics A, 30(2):131–150, 2000.

Available from https://www.hds.utc.fr/~tdenoeux.

See Also

proDSinit, proDSfit

Examples

```
## Glass dataset
data(glass)
xapp<-glass$x[1:89,]
yapp<-glass$y[1:89]
xtst<-glass$x[90:185,]
ytst<-glass$y[90:185]
## Initialization
param0<-proDSinit(xapp,yapp,nproto=7)
## Training
fit<-proDSfit(xapp,yapp,param0)
## Test
val<-proDSval(xtst,fit$param,ytst)
## Confusion matrix
table(ytst,val$ypred)
```

vehicles *Vehicles dataset*

Description

This dataset was collected from silhouettes by the HIPS (Hierarchical Image Processing System) extension BINATTS Four model vehicles were used for the experiment: bus, Chevrolet van, Saab 9000 and Opel Manta. The data were used to distinguish 3D objects within a 2-D silhouette of the objects. There are 846 instances and 18 numeric attributes. The first 564 objects are training data, the rest are test data. This version of dataset was used by Zouhal and Denoeux (1998).

Usage

```
data(vehicles)
```

Format

A list with two elements:

x The 846 x 18 object-attribute matrix.

y A 846-vector containing the class labels.

References

P. M. Murphy and D. W. Aha. UCI Reposition of machine learning databases. [Machine readable data repository]. University of California, Departement of Information and Computer Science, Irvine, CA.

L. M. Zouhal and T. Denoeux. An evidence-theoretic k-NN rule with parameter optimization. IEEE Transactions on Systems, Man and Cybernetics Part C, 28(2):263–271,1998.

Examples

```
data(vehicles)
table(vehicles$y)
```

Index