

Package ‘Immigrate’

January 21, 2019

Title Iterative Max-Min Entropy Margin-Maximization with Interaction Terms for Feature Selection

Version 0.0.2

Author Ruzhang Zhao, Pengyu Hong, Jun S. Liu

Description Based on large margin principle, this package performs feature selection methods: ``IM4E"(Iterative Margin-Maximization under Max-Min Entropy Algorithm); ``imIM4E"(imbalance Iterative Margin-Maximization under Max-Min Entropy Algorithm); ``Immigrate"(Iterative Max-Min Entropy Margin-Maximization with Interaction Terms Algorithm); ``BIM"(Boosted version of IMMIGRATE algorithm); ``Simba"(Iterative Search Margin Based Algorithm); ``LFE"(Local Feature Extraction Algorithm). This package also performs prediction for the above feature selection methods. See Zhao et al. (2018) <arXiv:1810.02658> for more details.

Maintainer Ruzhang Zhao<zrz6787@gmail.com>

Depends R (>= 3.0.0)

License GPL (>= 2)

URL <https://CRAN.R-project.org/package=Immigrate>,
<https://github.com/RuzhangZhao/Immigrate/>,
<https://arxiv.org/abs/1810.02658>

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0.9000

LinkingTo Rcpp

Imports Rcpp, pROC, stats

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-21 16:50:03 UTC

R topics documented:

BIM	2
IM4E	3
imIM4E	4
Immigrate	5
LFE	6
one.IM4E	6
one.imIM4E	7
one.Immigrate	8
park	8
pred.values	9
predict.BIM	10
predict.IM4E	11
predict.Immigrate	12
predict.LFE	13
Simba	13

Index	15
--------------	-----------

BIM

BIM

Description

This function performs BIM algorithm.

Usage

```
BIM(xx, yy, nBoost = 3, max_iter = 5, removesmall = FALSE,
     sigstart = 0.02, sigend = 4)
```

Arguments

xx	model matrix of explanatory variables
yy	label vector
nBoost	number of classifiers in BIM, default to be 3
max_iter	maximum number of iteration for IMMIRGATE classifier, default to be 5
removesmall	whether remove features with small weights, default to be FALSE
sigstart	start of sigma used in algorithm, default to be 0.02
sigend	end of sigma used in algorithm, default to be 4

Value

matrix	list of matrix weights
weights	coefficient vector for classifiers
sample_wt	sample weights

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
re<-BIM(xx,yy)
```

IM4E

IM4E

Description

This function performs IM4E(Iterative Margin-Maximization under Max-Min Entropy) algorithm.

Usage

```
IM4E(xx, yy, epsilon = 0.01, sig = 1, lambda = 1, max_iter = 10,
      removesmall = FALSE)
```

Arguments

xx	model matrix of explanatory variables
yy	label vector
epsilon	criterion for stopping iteration
sig	sigma used in algorithm, default to be 1
lambda	lambda used in algorithm, default to be 1
max_iter	maximum number of iteration
removesmall	whether remove features with small weights, default to be FALSE

Value

w	new weight after one loop
iter_num	number of iteration for convergence
final_c	final cost

References

Bei Y, Hong P. Maximizing margin quality and quantity[C]//Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on. IEEE, 2015: 1-6.

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
re<-IM4E(xx,yy)
print(re)
```

 imIM4E

imIM4E

Description

This function performs imIM4E(imbalance Iterative Margin-Maximization under Max-Min entropy) algorithm.

Usage

```
imIM4E(xx, yy, epsilon = 0.01, sig = 1, rho = 1, lambda = 1,
       max_iter = 10, removesmall = FALSE)
```

Arguments

xx	model matrix of explanatory variables
yy	label vector
epsilon	criterion for stopping iteration
sig	sigma used in algorithm, default to be 1
rho	rho used in algorithm, default to be 1
lambda	lambda used in algorithm, default to be 1
max_iter	maximum number of iteration
removesmall	whether remove features with small weights, default to be FALSE

Value

w	new weight after one loop
iter_num	number of iteration for convergence
final_c	final cost

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
re<-imIM4E(xx,yy)
print(re)
```

Immigrate	<i>Immigrate</i>
-----------	------------------

Description

This function performs Immigrate(Iterative Max-Min Entropy Margin-Maximization with Interaction Terms) algorithm.

Usage

```
Immigrate(xx, yy, w0, epsilon = 0.01, sig = 1, max_iter = 10,  
removesmall = FALSE, randomw0 = FALSE)
```

Arguments

xx	model matrix of explanatory variables
yy	label vector
w0	initial weights
epsilon	criterion for stopping iteration
sig	sigma used in algorithm, default to be 1
max_iter	maximum number of iteration
removesmall	whether to remove features with small weights, default to be FALSE
randomw0	whether to use randomly initial weights, default to be FALSE

Value

w	new weight after one loop
iter_num	number of iteration for convergence
final_c	final cost

References

Zhao R, Hong P, Liu J S. IMMIGRATE: A Margin-based Feature Selection Method with Interaction Terms[J]. arXiv preprint arXiv:1810.02658, 2018.

See Also

<https://arxiv.org/abs/1810.02658> for more details.

Examples

```
data(park)  
xx<-park$xx  
yy<-park$yy  
re<-Immigrate(xx,yy)  
print(re)
```

LFE

LFE

Description

This function performs LFE(Local Feature Extraction) algorithm.

Usage

```
LFE(xx, yy, T = 5)
```

Arguments

xx	model matrix of explanatory variables
yy	label vector
T	number of instance used to update weights, default to be 5

Value

w	new weight matrix after LFE algorithm
---	---------------------------------------

References

Sun Y, Wu D. A relief based feature extraction algorithm[C]//Proceedings of the 2008 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2008: 188-195.

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
re<-LFE(xx,yy)
print(re)
```

one.IM4E

one.IM4E

Description

This function performs Iterative Margin-Maximization under Max-Min Entropy algorithm for one loop.

Usage

```
one.IM4E(train_xx, train_yy, w, sig = 1, lambda = 1)
```

Arguments

train_xx	model matrix of explanatory variables
train_yy	label vector
w	initial weight
sig	sigma used in algorithm, default to be 1
lambda	lambda used in algorithm, default to be 1

Value

w	weights
C	cost

 one.imIM4E

one.imIM4E

Description

This function performs imIM4E(imbalance Iterative Margin-Maximization under Max-Min entropy) algorithm for one loop.

Usage

```
one.imIM4E(train_xx, train_yy, w, sig = 1, rho = 1, lambda = 1)
```

Arguments

train_xx	model matrix of explanatory variables
train_yy	label vector
w	initial weight
sig	sigma used in algorithm, default to be 1
rho	rho used in algorithm, default to be 1
lambda	lambda used in algorithm, default to be 1

Value

w	new weight after one loop
C	cost after one loop

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
w0 <- rep(1,ncol(xx))/ncol(xx)
re<-one.imIM4E(xx,yy,w0)
print(re$w)
```

one.Immigrate	<i>one.Immigrate</i>
---------------	----------------------

Description

This function performs Immigrate(Iterative Max-Min Entropy Margin-Maximization with Interaction Terms) algorithm for one loop.

Usage

```
one.Immigrate(train_xx, train_yy, W, sig = 1)
```

Arguments

train_xx	model matrix of explanatory variables
train_yy	label vector
W	initial weight matrix
sig	sigma used in algorithm, default to be 1

Value

W	new weight matrix after one loop
C	cost after one loop

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
W0 <- diag(rep(1,ncol(xx)),ncol(xx))/sqrt(ncol(xx))
re<-one.Immigrate(xx,yy,W0)
print(re$w)
```

park	<i>Parkinsons Dataset</i>
------	---------------------------

Description

Parkinsons Dataset

Usage

```
data(park)
```


Format

An object of class

Source

[parkinsons](#)

References

Frank, A. and A. Asuncion. UCI Machine Learning Repository. 2010.

Examples

```
data(park)
xx <- park$xx
yy <- park$yy
```

`pred.values`

pred.values

Description

This function performs some statistical value prediction

Usage

```
pred.values(y_train, y_test, pred_train, pred_test)
```

Arguments

<code>y_train</code>	label vector for training data
<code>y_test</code>	label vector for test data
<code>pred_train</code>	predicted probabilities for training data
<code>pred_test</code>	predicted probabilities for test data

Value

<code>AUC_train</code>	AUC for training data
<code>AUC_test</code>	AUC for test data
<code>accuracy_test</code>	accuracy for test data
<code>precision_test</code>	precision for test data
<code>recall_test</code>	recall for test data
<code>F1_test</code>	F1 score for test data
<code>thre</code>	threshold to separate two labels, obtained from training data

Examples

```

y_train<-c(0,1,0,1,0,1)
y_test<-c(0,1,0,1)
pred_train<-c(0.77,0.89,0.32,0.96,0.10,0.67)
pred_test<-c(0.68,0.75,0.50,0.81)
re<-pred.values(y_train,y_test,pred_train,pred_test)
print(re)

```

predict.BIM

predict.BIM

Description

This function performs the predion for BIM algorithm.

Usage

```

## S3 method for class 'BIM'
predict(object, xx, yy, newx, type = "both", ...)

```

Arguments

object	result of BIM algorithm
xx	model matrix of explanatory variables
yy	label vector
newx	test data
type	the form of final output
...	further arguments passed to or from other methods

Value

response	preded probabilities for xx
class	preded class for xx

Examples

```

data(park)
xx<-park$xx
yy<-park$yy
index<-c(1:floor(nrow(xx)*0.3))
train_xx<-xx[-index,]
test_xx<-xx[index,]
train_yy<-yy[-index]
test_yy<-yy[index]
re<-BIM(train_xx,train_yy)
res<-predict(re,train_xx,train_yy,test_xx,type="class")
print(res)

```

predict.IM4E	<i>predict.IM4E</i>
--------------	---------------------

Description

This function performs the prediction for IM4E (Iterative Margin-Maximization under Max-Min Entropy) algorithm.

Usage

```
## S3 method for class 'IM4E'
predict(object, xx, yy, newx, sig = 1, type = "both",
  ...)
```

Arguments

object	weight or result of IM4E algorithm
xx	model matrix of explanatory variables
yy	label vector
newx	model matrix to be predicted
sig	sigma used in algorithm, default to be 1
type	the form of final output, default to be "both"
...	further arguments passed to or from other methods

Value

response	predicted probabilities for newx
class	predicted class for newx

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
index<-c(1:floor(nrow(xx)*0.3))
train_xx<-xx[-index,]
test_xx<-xx[index,]
train_yy<-yy[-index]
test_yy<-yy[index]
re<-IM4E(train_xx,train_yy)
res<-predict(re,train_xx,train_yy,test_xx,type="class")
print(res)
```

predict.Immigrate *predict.Immigrate*

Description

This function performs the prediction for Immigrate(Iterative Max-Min Entropy Margin-Maximization with Interaction Terms) algorithm.

Usage

```
## S3 method for class 'Immigrate'
predict(object, xx, yy, newx, sig = 1,
        type = "both", ...)
```

Arguments

object	result of Immigrate algorithm
xx	model matrix of explanatory variables
yy	label vector
newx	model matrix to be predicted
sig	sigma used in algorithm, default to be 1
type	the form of final output, default to be "both"
...	further arguments passed to or from other methods

Value

response	predicted probabilities for newx
class	predicted class for newx

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
index<-c(1:floor(nrow(xx)*0.3))
train_xx<-xx[-index,]
test_xx<-xx[index,]
train_yy<-yy[-index]
test_yy<-yy[index]
re<-Immigrate(train_xx,train_yy)
res<-predict(re,train_xx,train_yy,test_xx,type="class")
print(res)
```

predict.LFE	<i>predict.LFE</i>
-------------	--------------------

Description

This function performs prediction for LFE(Local Feature Extraction) algorithm.

Usage

```
## S3 method for class 'LFE'
predict(object, xx, yy, newx, ...)
```

Arguments

object	weight
xx	model matrix of explanatory variables
yy	label vector
newx	model matrix to be predicted
...	further arguments passed to or from other methods

Value

predicted labels

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
w<-LFE(xx,yy)
pred<-predict(w,xx,yy,xx)
print(pred)
```

Simba	<i>Simba</i>
-------	--------------

Description

This function performs Simba(Iterative Search Margin Based Algorithm).

Usage

```
Simba(xx, yy, T = 5)
```

Arguments

xx	model matrix of explanatory variables
yy	label vector
T	number of instance used to update weights, default to be 5

Value

w	new weight after Simba algorithm
---	----------------------------------

References

Gilad-Bachrach R, Navot A, Tishby N. Margin based feature selection-theory and algorithms[C]//Proceedings of the twenty-first international conference on Machine learning. ACM, 2004: 43.

Examples

```
data(park)
xx<-park$xx
yy<-park$yy
re<-Simba(xx,yy)
print(re)
```

Index

- *Topic **BIM**
 - BIM, 2
 - predict.BIM, 10
 - *Topic **IM4E**
 - IM4E, 3
 - one.IM4E, 6
 - predict.IM4E, 11
 - *Topic **Immigrate**
 - Immigrate, 5
 - one.Immigrate, 8
 - predict.Immigrate, 12
 - *Topic **LFE**
 - LFE, 6
 - predict.LFE, 13
 - *Topic **Simba**
 - Simba, 13
 - *Topic **datasets**
 - park, 8
 - *Topic **imIM4E**
 - imIM4E, 4
 - one.imIM4E, 7
 - *Topic **one**
 - one.Immigrate, 8
 - *Topic **predict,**
 - pred.values, 9
 - *Topic **predict**
 - predict.BIM, 10
 - predict.IM4E, 11
 - predict.Immigrate, 12
 - predict.LFE, 13
 - *Topic **values**
 - pred.values, 9
- BIM, 2
- IM4E, 3
imIM4E, 4
Immigrate, 5
- LFE, 6
- one.IM4E, 6
one.imIM4E, 7
one.Immigrate, 8
- park, 8
pred.values, 9
predict.BIM, 10
predict.IM4E, 11
predict.Immigrate, 12
predict.LFE, 13
- Simba, 13