# Package ‘PO.EN’

June 18, 2020

**Type** Package

**Title** An Elastic-Net Regularized Presence-Only Model

**Version** 1.0

**Date** 2020-06-02

**Author** Zikun Yang, Chen Wang, Iuliana Ionita-Laza

**Maintainer** Zikun Yang <yangzikun1125@gmail.com>

**Description** Presence-only model with Elastic Net penalty is a regularized generalized linear model training on the presence-absence response. This package provides functions for tuning and fitting the presence-only model. The presence-only model can be used to predict regulatory effects of genetic variants at sequence-level resolution by integrating a large number of epigenetic features and massively parallel reporter assays (MPRAs).

**Depends** R (>= 3.1.0),
PUlasso,
Rcpp,
glmnet,
RcppArmadillo,
pROC

**License** GPL (>= 2)

**Encoding** UTF-8

**Imports** Rcpp (>= 1.0.4),
RcppArmadillo

**LinkingTo** Rcpp,
RcppArmadillo

**RoxygenNote** 7.1.0.9000

## R topics documented:

## PO.EN-package *An Elastic-Net Regularized presence-only Model*

### Description

This package fits a presence-only model with elastic-net penalty using coordinate descent. This package also provides a feature of tuning the prevalence parameter through a two-dimensional cross-validation. The package can be used in genetics study mainly for predicting regulatory effects of genetic variants given a large number of epigenetic features.

### Details

Accept typical presence-only response vector `y`, a vector consisted of presence and background observations, and design matrix `x`. Three main functions:

| | |
|---|---|
| `cv.PO.EN` | The cross-validation tuning function |
| `PO.EN` | The main model-fitting function |
| `PO.EN.predict` | The predicting function |

### Author(s)

Zikun Yang, Chen Wang, Iuliana Ionita-Laza

Maintainer: Zikun Yang <yangzikun1125@gmail.com>

### References

Zikun Yang, Chen Wang, Iuliana Ionita-Laza. A robust presence-only model to predict regulatory effects of genetic variants at single nucleotide resolution by integrating epigenetic information and massively parallel reporter assays. 2020

### Examples

```
data(example.data) # example training dataset, including training dataset and testing dataset
train_data<-example.data$train.data
y_train=train_data$response;x_train=train_data[,-1] # response and design matrix of training data
test_data<-example.data$test.data
y_test=test_data$response;x_test=test_data[,-1] # response and design matrix of testing data
PO.EN.cv<-cv.PO.EN(x_train,y_train,input.pi=seq(0.01,0.4,length.out=10))
PO.EN.beta<-PO.EN(x_train,y_train,lambda=PO.EN.cv$lambda.min,
          true.prob=PO.EN.cv$pi,beta_start=rep(0,ncol(x_train)+1))
predictions<-PO.EN.predict(x_test,PO.EN.beta)
roc(y_test~predictions)
```

## `cv.PO.EN` *Cross-validation function of PO-EN model*

### Description

Does k-fold cross-validation for PO-EN, produces a pair values of lambda and the prevalence parameter for an optimal fitting.

### Usage

```
cv.PO.EN(X, Y, alpha=0.5, o.iter=5, i.iter=20,
epsilon=1e-4,nfolds=10,type.measure='deviance',
depth=100,input.pi=0.5,a=sqrt(0.5))
```

### Arguments

`X` Input design matrix. Should not include the intercept vector.

`Y` Response variable. Should be a binary vector.

`alpha` The elastic net mixing parameter, with $0 \leq$`alpha`$\leq 1$.

`o.iter` Number of outer loop iteration.

`i.iter` Number of inner loop iteration.

`epsilon` The threshold for stopping the coordinate descent algorithm.

`nfolds` The number of folds for applying cross validation. The default setting is 10. The number of presence observations must be a multiple of `nfolds`.

`type.measure` The loss function to use for tuning lambda. The default is `type.measure='deviance'`. Other choices include AUROC (`type.measure='auc'`) and F measure (`type.measure='F'`).

`depth` The ratio between the largest lambda and the smallest lambda of the candidate sequence of lambda.

`input.pi` The user-supplied prevalence sequence.

`a` The parameter of F measure for tuning the true prevalence, the default value is $\sqrt{0.5}$.

`seed` A single value used for random number generation of the functions.

### Details

The cross-validation function runs a n-folds cross-validation for selecting an optimal pair of lambda and the prevalence parameter. The default setting is 10-folds cross validation. The candidate sequence of lambda is automatically generated by the function based on a warm start. The values of `input.pi` should be supplied by users.

### Value

`lambda.min` value of lambda that returns the minimum (or maximum, depending on `type.measure`) of mean cross-validated error.

`lambda.1se` largest value of lambda such that error is within 1 standard error of the minimum.

`pi` value of the prevalence parameter that returns maximum F measure.

### Examples

```
data(example.data) # example datasets, including training dataset and testing dataset
train_data<-example.data$train.data
y_train=train_data$response;x_train=train_data[,-1] # response and design matrix of training data
PO.EN.cv<-cv.PO.EN(x_train,y_train,input.pi=seq(0.01,0.4,length.out=10))
PO.EN.beta<-PO.EN(x_train,y_train,lambda=PO.EN.cv$lambda.min,
          true.prob=PO.EN.cv$pi,beta_start=rep(0,ncol(x_train)+1))
```

## example.data *Example datasets*

### Description

This data list, example.data, includes three datasets generated based on Saturation mutagenesis results (M. Kircher, et al.,2019) and the DeepSEA features (Zhou & Troyanskaya, 2015). The training and testing datasets in the data list include binary response vectors, which are truncations of the P values of tissue K562 from the Saturation mutagenesis results, and reduced versions of the DeepSEA features for a faster computational demonstration. The full.data dataset includes the original P values, chromosome and allelic information, and the complete DeepSEA features.

### Usage

```
example.data
```

### Format

The example.data$train.data and example.data$test.data are dataframes with 220 and 1574 observations and 146 variables.

**response** A binary response vector

**features** Standardized 145 DeepSEA features

The example.data$full.data is a dataframe with 1794 observations and 924 variables, i.e., including all 919 DeepSEA features.

**chr** The chromosome of SNPs

**pos** The position of SNPs

**ref.alt** The reference and alternative alleles of SNPs

**p.value** The P value of SNPs

**features** The original 919 DeepSEA features

PO.EN *A robust presence-only model with Elastic Net penalty*

### Description

Fit a logistic regression with presence-only response via penalized maximum likelihood. The regularization path is computed for the elastic-net penalty at a pair values of lambda and the prevalence parameter.

### Usage

```
PO.EN(x,y,o.iter=5, i.iter=5, lambda=.01,alpha=.5,
true.prob=0.5,beta_start,epsilon=1e-4, gram.input=F,XtX.input=0,
ytx.input=0,XtX_reduce.input)
```

### Arguments

| | |
|---|---|
| x | Input design matrix. Should not include the intercept vector. |
| y | Response variable. Should be a binary vector, such that 0 represents background observations and 1 represents presence observations. |
| o.iter | Number of outer loop iteration. |
| i.iter | Number of inner loop iteration. |
| lambda | A user supplied Elastic Net penalty parameter. |
| alpha | The elastic net mixing parameter, where 0 ≤alpha≤ 1. |
| true.prob | The prevalence parameter, should be provided by users. Can be tuned in the cross-validation function. |
| epsilon | The threshold for stopping the coordinate descent algorithm. |
| gram.input | The function allows users to feed the gram matrix for fasting computation. The default setting is False, and the function compute the gram matrix for computation. |

### Details

The function fits a presence-only model with an elastic net penalty.

### Value

beta The fitting vector of the coefficients, the intercept included.

### Examples

```
data(example.data) # example datasets, including training dataset and testing dataset
train_data<-example.data$train.data
y_train=train_data$response;x_train=train_data[,-1]  # response and design matrix of training data
```

```
test_data<-example.data$test.data
y_test=test_data$response;x_test=test_data[,-1] # response and design matrix of testing data
PO.EN.cv<-cv.PO.EN(x_train,y_train,input.pi=seq(0.01,0.4,length.out=10))
PO.EN.beta<-PO.EN(x_train,y_train,lambda=PO.EN.cv$lambda.min,
          true.prob=PO.EN.cv$pi,beta_start=rep(0,ncol(x_train)+1))
predictions<-PO.EN.predict(x_test,PO.EN.beta)
roc(y_test~predictions)
```

## PO.EN.predict *PO-EN predicting function*

### Description

A prediction function using the linear predictor of PO-EN fitting results.

### Usage

```
PO.EN.predict(X, beta)
```

### Arguments

| | |
|---|---|
| X | Input design matrix. Should not include the intercept vector. |
| beta | A coefficients vector from the PO-EN fitting function. |

### Examples

```
PO.EN.cv<-cv.PO.EN(x_train,y_train,input.pi=seq(0.01,0.4,length.out=10))
PO.EN.beta<-PO.EN(x_train,y_train,lambda=PO.EN.cv$lambda.min,
          true.prob=PO.EN.cv$pi,beta_start=rep(0,ncol(x_train)+1))
predictions<-PO.EN.predict(x_test,PO.EN.beta)
roc(y_test~predictions)
```

# Index