# Package: KLexp (via r-universe)

September 13, 2024

**Title** Kernel_lasso Expansion

**Version** 0.0.0.9000

**Description** Kernel_lasso package can expands the features of existed data. It used the Gauss function to amplify the dimension of dataset and decrease the feature by lasso.

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1.9001

**LazyLoad** yes

**Depends** glmnet (>= 4.1-2)

**Imports** graphics, stats

**Repository** https://zongrui-dai.r-universe.dev

**RemoteUrl** https://github.com/zongrui-dai/kernel-lasso-feature-expansion

**RemoteRef** HEAD

**RemoteSha** 64648f44bfb53c4b44027b72cfd29304f738b964

## Contents

gauss *Gauss function*

## Description

Gauss function

## Usage

```
gauss(d1, d2, sigma = 0.5)
```

## Arguments

| | |
|---|---|
| d1 | vector1 |
| d2 | vector2 |
| sigma | The hyperparameter of RBF kernel function, which indicates the width. |

## Examples

```
##
data(iris,package = 'datasets')
w<-gauss(iris[,1],iris[,2])
print(w)
```

kernel_lasso_expansion *kernel_lasso_expansion*

## Description

Kernel_lasso is one feature selection method, which combines the feature expansion and lasso regression together. Kernel function will increase the dimensions of the existed data and then reduce the features by lasso. 'glmnet' package should be higher than 4.1-2.

## Arguments

| | |
|---|---|
| x | Your input features, which can have to be data.frame with at least two variables. |
| y | The dependent variable |
| sigma | The hyperparameter of RBF kernel function, which indicates the width. |
| dataframe | Wether the data is dataframe. The default is TURE |
| standard | Using 'max_min_scale' or 'Z_score' method to standardize the data. NULL means no standardization |

## Examples

```
##Regression (MSE)
data(attenu,package = 'datasets')
result<-kernel_lasso_expansion(x=attenu[,-c(3,5)],y=attenu[,5],
standard = 'max_min',sigma=0.01,control = lasso.control(nfolds=3,type.measure = 'mse'))
summary(result)

#Plot the lasso
plot(result$lasso)

#Result
result$original ##The original feature space
result$expansion  ##The feature space after expansion
result$final_feature  ##The name of the final feature
result$final_data  ##The dataframe of final feature
```

---

`lasso.control` *lasso.control*

---

## Description

The same function from glmnet, which controls the training of lasso.

## Usage

```
lasso.control(nfolds = 10, trace.it = 1, type.measure = "auc")
```

## Arguments

| | |
|---|---|
| `nfolds` | n-fold cross-validation. |
| `trace.it` | Whether to plot the training process |
| `type.measure` | Choose the loss funcrion. |

## Value

The lasso control setting

## Examples

```
##10-fold Cross-validation with MSE as loss function
c<-lasso.control(nfolds=10,type.measure='mse')
```

## max_min_scale *max_min_scale*

### Description

max_min_scale is used to calculate the standardization value of data.The formula is (x-min(x))/(max(x)-min(x)). It can compress the data into the (0,1).

### Arguments

| | |
|---|---|
| data | Your input data, which can be numerci or data.frame |
| dataframe | Wether the data is dataframe. The default is False(numeric) |

### Examples

```
##For the numeric data
data(iris,package = 'datasets')
w<-max_min_scale(iris[,1])
print(w)

##For the data.frame data
w1<-max_min_scale(iris[,-5],dataframe=TRUE)
print(w1)
```

## Z_score *Z_score*

### Description

Z-score method is used to calculate the standardization value of data.The formula is (x-mean(x))/var(x). It can compress the data into the (0,1).

### Usage

```
Z_score(data, dataframe = FALSE)
```

### Arguments

| | |
|---|---|
| data | Your input data, which can be numerci or data.frame |
| dataframe | Wether the data is dataframe. The default is False(numeric) |

## Examples

```
##For the numeric data
data(iris,package = 'datasets')
w<-Z_score(iris[,1])
print(w)

##For the data.frame data
w1<-Z_score(iris[,-5],dataframe=TRUE)
print(w1)
```

# Index