Package ‘mlapi’
December 17, 2017

Type Package
Title Abstract Classes for Building 'scikit-learn' Like API
Version 0.1.0
Author Dmitriy Selivanov
Maintainer Dmitriy Selivanov <selivanov.dmitriy@gmail.com>
Description Provides 'R6' abstract classes for building machine learning models with 'scikit-learn' like API. <http://scikit-learn.org/> is a popular module for 'Python' programming language which design became de facto a standard in industry for machine learning tasks.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Depends methods
Imports R6 (>= 2.2.1), Matrix (>= 1.1)
Suggests knitr
RoxygenNote 6.0.1
VignetteBuilder knitr
NeedsCompilation no
Repository CRAN
Date/Publication 2017-12-17 17:22:37 UTC

R topics documented:

fit ................................................................. 2
fit_transform .................................................. 2
mlapiDecomposition ......................................... 3
mlapiEstimation ............................................. 5
mlapiEstimationOnline .................................... 7
mlapiTransformation ........................................ 7
mlapiTransformationOnline ................................ 8
predict ......................................................... 9
transform ...................................................... 9
fit

*Fits model to data*

**Description**

Generic function to fit models (inherited from `mlapiEstimation`)

**Usage**

```r
fit(x, model, y = NULL, ...) 
```

## S3 method for class 'Matrix'

```r
fit(x, model, y = NULL, ...) 
```

## S3 method for class 'matrix'

```r
fit(x, model, y = NULL, ...) 
```

**Arguments**

- **x**: A matrix like object, should *inherit from* `Matrix` or `matrix`
- **model**: instance of class `estimator` which should implement method with signature `$fit(x, y, ...)$
- **y**: `NULL` by default. Optional response variable for supervised learning models. Should inherit from `vector` or `Matrix` or `matrix`. See documentation for corresponding models.
- **...**: additional data/model dependent arguments to downstream functions.

**Value**

```r
invisible(object$self()) 
```

---

**fit_transform**

*Fit model to the data, then transforms data*

**Description**

Generic function to fit transformers (inherits from `mlapiTransformation`)
Usage

```r
fit_transform(x, model, y = NULL, ...)
```

### S3 method for class 'Matrix'
```
fit_transform(x, model, y = NULL, ...)
```

### S3 method for class 'matrix'
```
fit_transform(x, model, y = NULL, ...)
```

Arguments

- `x` A matrix like object, should inherit from `Matrix` or `matrix`
- `model` instance of class estimator which should implement method with signature $\text{fit}(x, \ldots)$
- `y` NULL by default. Optional response variable for supervised models. Should inherit from `vector Matrix` or `matrix`. See documentation for corresponding models.
- `...` additional data/model dependent arguments to downstream functions.

Value

Transformed version of the `x`

---

**mlapiDecomposition**

*Base abstract class for all decompositions*

Description

Base class for all **decompositions** which are methods which can decompose matrix into 2 low-dimensional matrices $x = f(A, B)$. (Think of this Latent Dirichlet Allocation, Non-negative Matrix Factorization, etc). It inherits from `mlapiTransformation` and additionally requires to implement components member.

Base class for all **decompositions** which are methods which can decompose matrix into 2 low-dimensional matrices $x = f(A, B)$ *incrementally*. It inherits from `mlapiDecomposition` and additionally requires to implement `partial_fit` method which can learn components incrementally.

Usage

```r
mlapiDecomposition

mlapiDecompositionOnline
```

Format

R6Class object.
Fields

components features embeddings. So if matrix is decomposed in a form $x = f(A, B)$ where $X = n \times m$, $A = n \times k$, $B = k \times m$ then $B = \text{components}$.

Methods

$\text{fit\_transform}(x, y = \text{NULL}, \ldots)$
$\text{transform}(x, \ldots)$ Performs transformation of the new data (after model was trained)

$\text{fit\_transform}(x, y = \text{NULL}, \ldots)$
$\text{partial\_fit}(x, y = \text{NULL}, \ldots)$
$\text{transform}(x, \ldots)$ Performs transformation of the new data (after model was trained)

Arguments

$x$ A matrix like object, should inherit from Matrix or matrix. Allowed classes should be defined in child classes.

$y$ NULL. Optional taget variable. Usually this should be NULL. There few cases when it could be used.

... additional parameters with default values

$x$ A matrix like object, should inherit from Matrix or matrix. Allowed classes should be defined in child classes.

$y$ NULL. Optional taget variable. Usually this should be NULL. There few cases when it could be used.

... additional parameters with default values

Examples

TruncatedSVD = R6::R6Class(
  classname = "TruncatedSVD",
  inherit = mlapi::mlapiDecomposition,
  public = list(
    initialize = function(rank = 10) {
      private$rank = rank
      super$set_internal_matrix_formats(dense = "matrix", sparse = NULL)
    },
    fit_transform = function(x, ...) {
      x = super$check_convert_input(x)
      private$n\_features = ncol(x)
      svd_fit = svd(x, nu = private$rank, nv = private$rank, ...)
      sing_values = svd_fit$d[seq_len(private$rank)]
      result = svd_fit$v %*% diag(x = sqrt(sing_values))
      private$components_ = t(svd_fit$v %*% diag(x = sqrt(sing_values)))
      rm(svd_fit)
      rownames(result) = rownames(x)
mlapiEstimation

Base abstract class for all classification/regression models

Description

Base class for all estimators. Defines minimal set of members and methods (with signatures) which have to be implemented in child classes.

Usage

mlapiEstimation

Format

R6Class object.
Methods

$\text{fit}(x, y, \ldots)$  
$\text{predict}(x, \ldots)$  Makes predictions on new data (after model was trained)

Arguments

$x$  A matrix like object, should inherit from \texttt{Matrix} or \texttt{matrix}. Allowed classes should be defined in child classes.

$y$  target - usually \texttt{vector}, but also can be a matrix like object. Allowed classes should be defined in child classes.

...  additional parameters with default values

Examples

SimpleLinearModel = R6::R6Class(  
classname = "mlapiSimpleLinearModel",  
inherit = mlapi::mlapiEstimation,  
public = list(    
    initialize = function(tol = 1e-7) {      
        private$tol = tol      
        super$set_internal_matrix_formats(dense = "matrix", sparse = NULL)    
    },    
    fit = function(x, y, \ldots) {      
        x = super$check_convert_input(x)      
        stopifnot(is.vector(y))      
        stopifnot(is.numeric(y))      
        stopifnot(nrow(x) == length(y))      
    },      
    predict = function(x) {      
        stopifnot(ncol(x) == private$n\_features)      
        x %*% matrix(private$coefficients, ncol = 1)    
    },    
    private = list(        
        tol = NULL,        
        coefficients = NULL,        
        n_features = NULL    
    ))  
)  
set.seed(1)  
model = SimpleLinearModel$new()  
x = matrix(sample(100 * 10, replace = TRUE), ncol = 10)  
y = sample(c(0, 1), 100, replace = TRUE)  
model$fit(as.data.frame(x), y)  
res1 = model$predict(x)  
# check pipe-compatible S3 interface  
res2 = predict(x, model)  
identical(res1, res2)
**mlapiEstimationOnline**

Base abstract class for all classification/regression models which can be trained incrementally (online)

**Description**

Base class for all online estimators. This class inherits from `mlapiEstimation` and additionally requires to implement `$\text{partial\_fit}(x, y, \ldots)$` method. Idea is that user can pass $x, y$ in chunks and model will be updated/refined incrementally.

**Usage**

`mlapiEstimationOnline`

**Format**

`R6Class` object.

**Methods**

- `$\text{fit}(x, y, \ldots)$`
- `$\text{partial\_fit}(x, y, \ldots)$`
- `$\text{predict}(x, \ldots)$` Makes predictions on new data (after model was trained)

**Arguments**

- $x$ A matrix like object, should inherit from `Matrix` or `matrix`. Allowed classes should be defined in child classes.
- $y$ target - usually vector, but also can be a matrix like object. Allowed classes should be defined in child classes.
- $\ldots$ additional parameters with default values

---

**mlapiTransformation**

Base abstract class for all transformations

**Description**

Base class for all online transformations.

**Usage**

`mlapiTransformation`
mlapiTransformationOnline

Format

R6Class object.

Methods

$fit_transform(x, y = \text{NULL}, \ldots)
$transform(x, \ldots) Performs transformation of the new data (after model was trained)

Arguments

x A matrix like object, should inherit from Matrix or matrix. Allowed classes should be defined in child classes.
y NULL. Optional taget variable. Usually this should be NULL. There few cases when it could be used.
... additional parameters with default values

mlapiTransformationOnline

Base abstract class for all transformations which can be trained incrementally (online)

Description

Base class for all online transformations. This class inherits from mlapiTransformation and additionally requires to implement $partial_fit(x, y, \ldots) method. Idea is that user can pass x, y in chunks and model will be updated/refined incrementally.

Usage

mlapiTransformationOnline

Format

R6Class object.

Methods

$fit_transform(x, y = \text{NULL}, \ldots)
$transform(x, \ldots) Performs transformation of the new data (after model was trained)

Arguments

x A matrix like object, should inherit from Matrix or matrix. Allowed classes should be defined in child classes.
y NULL. Optional taget variable. Usually this should be NULL. There few cases when it could be used.
... additional parameters with default values
**predict**

*Makes predictions on new data using pre-trained model*

**Description**

Makes predictions on new data using pre-trained model (inherits from mlapiEstimation)

**Usage**

```r
## S3 method for class 'matrix'
predict(object, model, ...)
```

```r
## S3 method for class 'Matrix'
predict(object, model, ...)
```

**Arguments**

- `object` = x in other methods. A matrix like object, should **inherit from** Matrix or matrix
- `model` object which **inherits** class mlapiEstimation which implements method model$predict(x, ...)
- `...` additional data/model dependent arguments to downstream functions

**transform**

*Transforms new data using pre-trained model*

**Description**

Generic function to transform data with pre-trained model (inherits from mlapiTransformation)

**Usage**

```r
## S3 method for class 'Matrix'
transform(_data, model, ...)
```

```r
## S3 method for class 'matrix'
transform(_data, model, ...)
```

**Arguments**

- `_data` = x in other methods. A matrix like object, should **inherit from** Matrix or matrix
- `model` object of class mlapiTransformation which implements method $transform(x, ...)
- `...` additional data/model dependent arguments to downstream functions.
Index

*Topic datasets
  - mlapiDecomposition, 3
  - mlapiEstimation, 5
  - mlapiEstimationOnline, 7
  - mlapiTransformation, 7
  - mlapiTransformationOnline, 8

fit, 2
fit_transform, 2

mlapiDecomposition, 3, 3
mlapiDecompositionOnline (mlapiDecomposition), 3
mlapiEstimation, 2, 5, 7, 9
mlapiEstimationOnline, 7
mlapiTransformation, 2, 3, 7, 8, 9
mlapiTransformationOnline, 8

predict, 9

transform, 9