KUHERD Documentation

Release 1.0

Joseph St.Amand

May 06, 2017

Contents:

1 KUHERD package **1**

1.1 Subpackages 1

1.1.1 KUHERD.Experiments package 1

1.1.1.1 Submodules 1

1.1.1.2 KUHERD.Experiments.MultiDT module 1

1.1.1.3 KUHERD.Experiments.MultiLR module 1

1.1.1.4 KUHERD.Experiments.MultiNB module 1

1.1.1.5 KUHERD.Experiments.MultiRF module 1

1.1.1.6 KUHERD.Experiments.MultiSVM module 2

1.1.1.7 Module contents 2

1.2 Submodules 2

1.3 KUHERD.FeatureSelector module 2

1.4 KUHERD.HerdVectorizer module 2

1.5 KUHERD.LabelSets module 5

1.6 KUHERD.LabelTransformations module 5

1.7 KUHERD.Models module 5

1.8 KUHERD.MultiFeatureSelector module 7

1.9 Module contents 7

2 Indices and tables **9**

Python Module Index **11**

Index **13**

CHAPTER 1

KUHERD package

1.1 Subpackages

1.1.1 KUHERD.Experiments package

1.1.1.1 Submodules

1.1.1.2 KUHERD.Experiments.MultiDT module

`KUHERD.Experiments.MultiDT.`**`MultiDT`**`()`

Program for running an experiment using Decision Tree classifier.

1.1.1.3 KUHERD.Experiments.MultiLR module

`KUHERD.Experiments.MultiLR.`**`MultiLR`**`()`

Program for running an experiment using Loigistic Regression classifier.

1.1.1.4 KUHERD.Experiments.MultiNB module

`KUHERD.Experiments.MultiNB.`**`MultiNB`**`()`

Program for running an experiment using Naive Bayes classifier.

1.1.1.5 KUHERD.Experiments.MultiRF module

`KUHERD.Experiments.MultiRF.`**`MultiRF`**`()`

Program for running an experiment using Random Forest classifier.

1.1.1.6 KUHERD.Experiments.MultiSVM module

`KUHERD.Experiments.MultiSVM.`**`MultiSVM`**`()`

Program for running an experiment using Support Vector Machine classifier.

1.1.1.7 Module contents

1.2 Submodules

1.3 KUHERD.FeatureSelector module

class `KUHERD.FeatureSelector.`**`FeatureSelector`**`(`*scoring_function*, *kbest*`)`

Bases: `object`

`fit``(`*X*, *Y*, *label_set*`)`

Fits the data by training the feature selection model compomnent.

Parameters

- **X** (*numpy matrix*) – The data matrix.
- **y** (*integer list*) – The labels for the data.
- **label_set** (*str*) – Either 'purpose' or 'field'.

Returns None

`transform``(`*X*`)`

Transforms the data, retaining only features learned in the "fit" process.

Parameters

- **X** (*numpy matrix*) – The data matrix.
- **y** (*integer list*) – The labels for the data.

Returns Transformed data matrix.

Return type (numpy matrix)

1.4 KUHERD.HerdVectorizer module

class `KUHERD.HerdVectorizer.`**`HerdVectorizer`**`(`*config*`)`

Bases: `object`

Main class responsible for vectorization of the text data. This class is extremely configurable, with many options for each preprocessing behavior, bigrams, stemmers, stopwords, and feature selection. Use of this class is done in the following manner: - Set configuration options for preproc_config, bigram_config, stemmer, stopwords, and feature selection. - Train the vectorizer on a set of documents and their corresponding labels. - After training is complete, the documents may be given to the transform function to convert to TFIDF form.

`create_bigram_index_map``(`*tokenized_docs*`)`

Creates a mapping from each bigram to a column index.

Parameters **tokenized_docs** (*list*) – A list of documents, each document is represented as a single long string

Returns Dictionary where keys are tokens, values are the index into a feature matrix.

Return type (dictionary)

`create_token_index_map`(*tokenized_docs*)

Given the tokenized documents, finds all unique tokens and forms an index map

Parameters **`tokenized_docs`** (*`list`*) – A list of documents, each document is represented as a single long string

Returns Dictionary where keys are tokens, values are the index into a feature matrix.

Return type (dictionary)

static `filter_docs`(*tokenized_docs*, *tok_index_map*)

Filters tokenized documents, removing all tokens which are not recognized by the specified token index map.

Parameters

- **`tokenized_docs`** (*`list`*) – A list of documents, each document is represented as a single long string
- **`tok_index_map`** (*`dictionary`*) – A mapping from tokens to their index in the feature matrix.

Returns A list of documents, where each document is a list of (filtered) tokens.

Return type (list)

`form_bigram_count_matrix`(*tokenized_docs*)

Calculates a bigram count matrix from a list of tokenized documents.

Parameters **`tokenized_docs`** (*`list`*) – A list of documents, each document is represented as a single long string

Returns A sparse count matrix in COO format.

Return type (sparse numpy matrix)

`form_count_matrix`(*tokenized_docs*)

Forms the count matrix from the tokenized documents

Parameters **`tokenized_docs`** (*`list`*) – A list of lists representing the tokenized documents. Each document is a list of tokens.

Returns A sparse count matrix in COO format.

Return type (sparse numpy matrix)

`getConfig`()

Retrieve the complete configuration needed to build a vectorizer.

Returns the complete configuration needed to build a vectorizer. Note that the config returned may only be used to train a new vectorizer. The config does NOT give model persistance.

`get_bigram_config`()

Retrieve the bigram configuration.

`get_preproc_config`()

Retrieve the preprocessor configuration.

static `lancaster_stemmer`(*docs*)

Lancaster stemming algorithm

static `porter_stemmer`(*docs*)

Porter stemming algorithm

set_bigram_config (*name*, *value*)
Set the bigram configuration.

set_bigrams (*bigrams*, *bigram_window_size*, *bigram_filter_size*, *bigram_nbest*)
Set the bigram configuration.

set_feature_selector (*scoring_func*, *kbest*, *multi_type*)
Set the feature selection configuration values.

set_preproc_config (*name*, *value*)
Set the preprocessor configuration value.

set_stemmer (*the_stemmer*)
Set the stemmer configuration values.

static snowball_stemmer (*docs*)
Snowball stemming algorithm

tokenize_docs (*docs*)
Breaks each document down into a list of words(tokens).

Converts a list of documents(each document is given as a single string) and converts them to their tokenized form in the following manner(some steps may be skipped if configured as such in the configuration settings)

- break document into tokens
- remove punctuation
- stem tokens

Parameters **docs** (`list`) – A list of documents, each document is represented as a single long string

Returns The tokenized documents as a list of lists, each item of the outer list is a document, which is represented as a list of words.

Return type (list)

train (*docs*, *y*, *label_set*)
Takes a list of documents, and the corresponding labels and trains the preprocessor(including feature selection).

Parameters

- **docs** (`list`) – A list of documents, where each document is represented as a string.
- **y** (`list`) – A list of integers representing the label for each document.
- **label_set** (`str`) – Specifies the label set so that the input 'y' may be interpreted. Valiud entries are either 'purpose' or 'field'.

@param docs The list of documents @param y A vector of labels which correspond to each document

transform_data (*docs*)
Tranforms documents into a sparse matrix.

Must be called after the preprocessor has been trained on some data. Process is as follows: -tokenize documents -search for bigrams -transform to TFIDF representation -select features

Parameters **docs** (`list`) – A list of documents, each document is represented as a string.

Returns A sparse CSR formatted matrix, each row corresponds to a document, ordering of documents is preserved.

Return type (sparse numpy matrix)

`KUHERD.HerdVectorizer.`**`main`**`()`

1.5 KUHERD.LabelSets module

1.6 KUHERD.LabelTransformations module

`KUHERD.LabelTransformations.`**`label2mat`**`(`*x*, *label_type*`)`

Converts a vector of integers to a matrix of of zero-one valued columns.

Parameters

- **label_vec** (*list*) – A vector containing integer values mapping to members of the label_type.
- **label_type** (*str*) – Specifies label_type of label_vec, either 'purpose' and 'field'.

Returns A matrix of zero-one valued columns.

Return type (numpy matrix)

`KUHERD.LabelTransformations.`**`mat2vec`**`(`*M*`)`

Converts a zero-one valued label matrix to an integer valued label vector.

Parameters **M** (*numpy mat*) – A zero-one valued label matrix.

`KUHERD.LabelTransformations.`**`vec2mat`**`(`*x*, *label_type*`)`

Converts a vector of integers to a matrix of of zero-one valued columns.

Parameters

- **x** (*list*) – A vector containing integer values mapping to members of the label_type.
- **label_type** (*str*) – Specifies label_type of label_vec, either 'purpose' and 'field'.

Returns A matrix of zero-one valued columns.

Return type (numpy matrix)

`KUHERD.LabelTransformations.`**`vec2string`**`(`*label_vec*, *label_type*`)`

Converts vector containing integers to a string representation using the label set dictionaries.

Parameters

- **label_vec** (*list*) – A vector containing integer values mapping to members of the label_type.
- **label_type** (*str*) – Specifies label_type of label_vec, either 'purpose' and 'field'.

Returns A list of strings that are members oif the label_type.

Return type (list)

1.7 KUHERD.Models module

class `KUHERD.Models.`**`ClassificationModel`**`(`*config*`)`

Bases: `object`

fit (*X*, *Y*)

Trains the model.

Fitting or "training" must be done before the model is able to make predictions.

Parameters

- **X** (*numpy matrix*) – Training samples.
- **Y** (*numpy matrix*) – Training labels.

Returns No return value.

Return type None

get_config ()

Returns the configuration used to build this model.

Returns dictionary containing target label set, internal model configuration, and model name.

Return type dict

predict (*X*)

Make predictions.

Parameters **X** (*numpy matrix*) – Training samples.

Returns predicted label values.

Return type numpy matrix

class KUHERD.Models.**PurposeFieldModel** (*config*)

Bases: object

fit (*abstracts*, *Y_purpose*, *Y_field*)

Trains the model.

Input arguments must all be the same length.

Parameters

- **abstracts** (*list*) – A list of documents, each document is represented as a list of words.
- **Y_purpose** (*list*) – A list of labels of the 'purpose' variety.
- **Y_field** (*list*) – A list of labels of the 'field' variety.

get_config ()

Returns the configuration used to build this model.

Returns dictionary containing the following keys, 'purpose_vectorizer', 'field_vectorizer', 'purpose_model', 'field_model'. Each entry is the configuration required to build the model.

Return type dict

predict (*abstracts*)

Make predictions on the input data.

The list of documents input is vectorized and input to the prediction model, which generates label predictions. This process is done separately for generating both purpose and field label predictions.

Parameters **abstracts** (*list*) – A list of documents, each document is represented as a list of words.

Returns dictionary containing two lists of predictions, dictionary keys are 'purpose' and 'field'.

Return type dictionary

1.8 KUHERD.MultiFeatureSelector module

class `KUHERD.MultiFeatureSelector.`**`MultiFeatureSelector`**(*scoring_function*, *kbest*, *multi_integrator*)

Bases: `object`

`fit`(*X*, *Y*, *label_set*)

Trains the feature selection process

Parameters

- **`X`** (*`numpy matrix`*) – Training samples.
- **`Y`** (*`numpy matrix`*) – Training Labels.
- **`label_set`** (*`str`*) – Denotes if label set is of the 'purpose' or 'field' type.

`transform`(*X*)

Tranforms the data by selecting the features learned in the training or "fit" process.

Parameters **`X`** (*`numpy matrix`*) – Data samples to run feature selection on.

Returns Data with only the selected features.

Return type (numpy matrix)

1.9 Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

k

KUHERD, 7
KUHERD.Experiments, 2
KUHERD.Experiments.MultiDT, 1
KUHERD.Experiments.MultiLR, 1
KUHERD.Experiments.MultiNB, 1
KUHERD.Experiments.MultiRF, 1
KUHERD.Experiments.MultiSVM, 2
KUHERD.FeatureSelector, 2
KUHERD.HerdVectorizer, 2
KUHERD.LabelSets, 5
KUHERD.LabelTransformations, 5
KUHERD.Models, 5
KUHERD.MultiFeatureSelector, 7

Index

C

ClassificationModel (class in KUHERD.Models), 5
create_bigram_index_map() (KUHERD.HerdVectorizer.HerdVectorizer method), 2
create_token_index_map() (KUHERD.HerdVectorizer.HerdVectorizer method), 3

F

FeatureSelector (class in KUHERD.FeatureSelector), 2
filter_docs() (KUHERD.HerdVectorizer.HerdVectorizer static method), 3
fit() (KUHERD.FeatureSelector.FeatureSelector method), 2
fit() (KUHERD.Models.ClassificationModel method), 5
fit() (KUHERD.Models.PurposeFieldModel method), 6
fit() (KUHERD.MultiFeatureSelector.MultiFeatureSelector method), 7
form_bigram_count_matrix() (KUHERD.HerdVectorizer.HerdVectorizer method), 3
form_count_matrix() (KUHERD.HerdVectorizer.HerdVectorizer method), 3

G

get_bigram_config() (KUHERD.HerdVectorizer.HerdVectorizer method), 3
get_config() (KUHERD.Models.ClassificationModel method), 6
get_config() (KUHERD.Models.PurposeFieldModel method), 6
get_preproc_config() (KUHERD.HerdVectorizer.HerdVectorizer method), 3
getConfig() (KUHERD.HerdVectorizer.HerdVectorizer method), 3

H

HerdVectorizer (class in KUHERD.HerdVectorizer), 2

K

KUHERD (module), 7
KUHERD.Experiments (module), 2
KUHERD.Experiments.MultiDT (module), 1
KUHERD.Experiments.MultiLR (module), 1
KUHERD.Experiments.MultiNB (module), 1
KUHERD.Experiments.MultiRF (module), 1
KUHERD.Experiments.MultiSVM (module), 2
KUHERD.FeatureSelector (module), 2
KUHERD.HerdVectorizer (module), 2
KUHERD.LabelSets (module), 5
KUHERD.LabelTransformations (module), 5
KUHERD.Models (module), 5
KUHERD.MultiFeatureSelector (module), 7

L

label2mat() (in module KUHERD.LabelTransformations), 5
lancaster_stemmer() (KUHERD.HerdVectorizer.HerdVectorizer static method), 3

M

main() (in module KUHERD.HerdVectorizer), 5
mat2vec() (in module KUHERD.LabelTransformations), 5
MultiDT() (in module KUHERD.Experiments.MultiDT), 1
MultiFeatureSelector (class in KUHERD.MultiFeatureSelector), 7
MultiLR() (in module KUHERD.Experiments.MultiLR), 1
MultiNB() (in module KUHERD.Experiments.MultiNB), 1
MultiRF() (in module KUHERD.Experiments.MultiRF), 1
MultiSVM() (in module KUHERD.Experiments.MultiSVM), 2

P

porter_stemmer() (KUHERD.HerdVectorizer.HerdVectorizer

static method), 3
predict() (KUHERD.Models.ClassificationModel method), 6
predict() (KUHERD.Models.PurposeFieldModel method), 6
PurposeFieldModel (class in KUHERD.Models), 6

S

set_bigram_config() (KUHERD.HerdVectorizer.HerdVectorizer method), 3
set_bigrams() (KUHERD.HerdVectorizer.HerdVectorizer method), 4
set_feature_selector() (KUHERD.HerdVectorizer.HerdVectorizer method), 4
set_preproc_config() (KUHERD.HerdVectorizer.HerdVectorizer method), 4
set_stemmer() (KUHERD.HerdVectorizer.HerdVectorizer method), 4
snowball_stemmer() (KUHERD.HerdVectorizer.HerdVectorizer static method), 4

T

tokenize_docs() (KUHERD.HerdVectorizer.HerdVectorizer method), 4
train() (KUHERD.HerdVectorizer.HerdVectorizer method), 4
transform() (KUHERD.FeatureSelector.FeatureSelector method), 2
transform() (KUHERD.MultiFeatureSelector.MultiFeatureSelector method), 7
transform_data() (KUHERD.HerdVectorizer.HerdVectorizer method), 4

V

vec2mat() (in module KUHERD.LabelTransformations), 5
vec2string() (in module KUHERD.LabelTransformations), 5