birdspotter

Release 0.1.4b1

Mar 10, 2021

README:

CHAPTER 1

birdspotter: A tool to measure social attributes of Twitter users

PyPI status PyPI version fury.io Documentation Status

birdspotter is a python package providing a toolkit to measures the *social influence* and *botness* of twitter users. It takes a twitter dump input in json or jsonl format and produces measures for:

- **Social Influence**: The relative amount that one user can cause another user to adopt a behaviour, such as retweeting.
- **Botness**: The amount that a user appears automated.

1.1 References:

```
Rizoiu, M.A., Graham, T., Zhang, R., Zhang, Y., Ackland, R. and Xie, L. #␣
↪DebateNight: The Role and Influence of Socialbots on Twitter During the 1st 2016 US␣
↪Presidential Debate. In Twelfth International AAAI Conference on Web and Social␣
↪Media (ICWSM'18), 2018. https://arxiv.org/abs/1802.09808
```

```
Ram, R., & Rizoiu, M.-A. A social science-grounded approach for quantifying online␣
↪social influence. In Australian Social Network Analysis Conference (ASNAC'19) (p.␣
↪2). Adelaide, Australia, 2019.
```

1.2 Installation

pip3 install birdspotter birdspotter requires a python version >=3.

1.3 Usage

To use birdspotter on your own twitter dump, replace './2016.json' with the path to your twitter dump './path/to/tweet/dump.json'. In this example we use Brendan Brown's archive of @realdonaldtrump tweets in

2016. It can be downloaded here.

```
from birdspotter import BirdSpotter
bs = BirdSpotter('./2016.json')
# This may take a few minutes, go grab a coffee...
labeledUsers = bs.getLabeledUsers(out='./output.csv')
```

After extracting the tweets, `getLabeledDataFrame()` returns a `pandas` dataframe with the influence and botness labels of users and writes a `csv` file if a path is specified i.e. `./output.csv`.

`birdspotter` relies on the Fasttext word embeddings wiki-news-300d-1M.vec, which will automatically be downloaded if not available in the current directory (`./`) or a relative data folder (`./data/`).

1.3.1 Get Cascades Data

After extracting the tweets, the retweet cascades are accessible by using:

```
cascades = bs.getCascadesDataFrame()
```

This dataframe includes the expected structure of the retweet cascade as given by Rizoiu et al. (2018) via the column `expected_parent` in this dataframe.

1.4 Advanced Usage

1.4.1 Adding more influence metrics

`birdspotter` provides DebateNight influence as a standard, when `getLabeledUsers` is run. To generate spatial-decay influence run:

```
bs.getInfluenceScores(time_decay = -0.000068, alpha = 0.15, beta = 1.0)
```

This returns the updated `featureDataframe` with influence scores appended, under the column `influence (<alpha>,<time_decay>,<beta>)`.

1.4.2 Training with your own botness data

`birdspotter` provides functionality for training the botness detector with your own training data. To generate an `csv` to be annotated run:

```
bs.getBotAnnotationTemplate('./annotation_file.csv')
```

Once annotated the botness detector can be trained with:

```
bs.trainClassifierModel('./annotation_file.csv')
```

1.4.3 Defining your own word embeddings

`birdspotter` provides functionality for defining your own word embeddings. For example:

```
customEmbedding # A mapping such as a dict() representing word embeddings
bs = BirdSpotter('./2016.json', embeddings=customEmbedding)
```

Embeddings can be set through several methods, refer to setWord2VecEmbeddings.

Note the default bot training data uses the wiki-news-300d-1M.vec and as such we would need to retrain the bot detector for alternative word embeddings.

1.5 Alternatives to python

1.5.1 Command-line usage

`birdspotter` can be accessed through the command-line to return a `csv`, with the recipe below:

```
birdspotter ./path/to/twitter/dump.json ./path/to/output/directory/
```

1.5.2 R usage

`birdspotter` functionality can be accessed in `R` via the `reticulate` package. `reticulate` still requires a `python` installation on your system and `birdspotter` to be installed. The following produces the same results as the standard usage.

```
install.packages("reticulate")
library(reticulate)
use_python(Sys.which("python3"))
birdspotter <- import("birdspotter")
bs <- birdspotter$BirdSpotter("./2016.json")
bs$getLabeledDataFrame(out = './output.csv')
```

1.6 Acknowledgements

The development of this package was partially supported through a UTS Data Science Institute seed grant.

CHAPTER 2

birdspotter

2.1 birdspotter package

2.1.1 Submodules

2.1.2 birdspotter.BirdSpotter module

birdspotter is a python package providing a toolkit to measures the social influence and botness of twitter users.

class birdspotter.BirdSpotter.**BirdSpotter**(*path*, *tweetLimit=None*, *embeddings='download'*, *quiet=False*)

Bases: object

Birdspotter measures the social influence and botness of twitter users.

This class takes a twitter dump in (json or jsonl format) and extract metrics bot and influence metrics for the users. The class will download word2vec embeddings if they are not specified. It exposes processed data from the tweet dumps.

cascadeDataframe
A dataframe of tweets ordered by cascades and time (the column casIndex denotes which cascade each tweet belongs to)

Type pandas.DataFrame

featureDataframe
A dataframe of users with their respective botness and influence scores.

Type pandas.DataFrame

hashtagDataframe
A dataframe of the text features for hashtags.

Type pandas.DataFrame

extractTweets (*filePath*, *tweetLimit=None*, *embeddings='download'*)
Extracts tweets from a json or jsonl file and generates cascade, feature and hashtag dataframes as class attributes.

Note that we use the file extension to determine how to handle the file.

Parameters

- **filePath** (*str*) – The path to a tweet json or jsonl file containing the tweets for analysis.
- **tweetLimit** (*int, optional*) – A limit on the number of tweets to process if the tweet dump is too large, if None then all tweets are processed, by default None
- **embeddings** (*collections.Mapping or str or None, optional*) – A method for loading word2vec embeddings. A path to a embeddings pickle or txt file, a mapping object, the string 'download', by default 'download'. If None, it does nothing.

Returns A dataframe of user's botness and influence scores (and other features).

Return type DataFrame

getBotAnnotationTemplate (*filename='annotationTemplate.csv'*)
Writes a CSV with the list of users and a blank column "isbot" to be annotated.

A helper function which outputs a CSV to be annotated by a human. The output is a list of users with the blank "isbot" column.

Parameters **filename** (*str*) – The name of the file to write the CSV

Returns A dataframe of the users, with their screen names and a blank "is_bot" column.

Return type Dataframe

getBotness()
Adds the botness of users to the feature dataframe.

It requires the tweets be extracted and the classifier be trained, otherwise exceptions are raised respectively.

Returns The current feature dataframe of users, with associated botness scores appended.

Return type DataFrame

Raises Exception – Tweets haven't been extracted yet. Need to run extractTweets.

getCascadeMembership()

getCascadesDataFrame()
Adds botness column and standard influence to the cascade dataframe and returns the cascadeDataframe

getInfluenceScores (*params={'alpha': None, 'beta': 1.0, 'time_decay': -6.8e-05}*)
Adds a specified influence score to feature dataframe

The specified influence will appear in the returned feature df, under the column 'influence (<alpha>,<time_decay>,<beta>)'.

Parameters

- **time_decay** (*float, optional*) – The time-decay r parameter described in the paper, by default -0.000068
- **alpha** (*float, optional*) – A float between 0 and 1, as described in the paper. If None DebateNight method is used, else spatial-decay method, by default None
- **beta** (*float, optional*) – A social strength hyper-parameter, by default 1.0

Returns The current feature dataframe of users, with associated botness scores.

Return type Dataframe

Raises `Exception` – Tweets haven't been extracted yet. Need to run extractTweets.

getLabeledUsers(*out=None*)
Generates a standard dataframe of users with botness and DebateNight influence scores (and other features), and optionally outputs a csv.

Parameters **out** (*str, optional*) – A output path for a csv of the results, by default None

Returns A dataframe of the botness and influence scores (and other feautes) of each user

Return type DataFrame

Raises `Exception` – Tweets haven't been extracted yet

getLabels()
Adds labels of users to the feature dataframe.

It requires the tweets be extracted and the classifier be trained, otherwise exceptions are raised respectively.

Returns The current feature dataframe of users, with associated label scores appended.

Return type DataFrame

Raises `Exception` – Tweets haven't been extracted yet. Need to run extractTweets.

loadClassifierModel(*fname*)
Loads the XGB booster model, from the saved XGB binary file

Parameters **fname** (*str*) – The path to the XGB binary file

loadPickledBooster(*fname*)
Loads the pickled booster model

Parameters **fname** (*str*) – The path to the pickled xgboost booster

process_tweet(*j*)

set_word_embeddings(*embeddings='download'*, *force_reload=True*)
Sets the word2vec embeddings. The embeddings can be a path to a pickle or txt file, a mapping object or the string 'download' which will automatically download and use the FastText 'wiki-news-300d-1M.vec' if not available in the current path.

Parameters

- **embeddings** (*collections.Mapping or str or None, optional*) – A method for loading word2vec embeddings. A path to a embeddings pickle or txt file, a mapping object, or the string 'download', by default 'download'. If None, it does nothing.
- **force_reload** (*bool, optional*) – If the embeddings are already set, force_reload determines whether to update them, by default True

trainClassifierModel(*labelledData*, *targetColumnName='isbot'*, *saveFileName=None*, *update=False*, *iterations=100*, *hyper_parameter_search=True*)
Trains the bot detection classifier.

Trains the bot detection classifier, using an XGB classifier. Due to the way XGB works, the features used are the intersection, between the features from the tweet dumps and the features from the training set.

Parameters

- **labelledData** (*str or pandas.DataFrame*) – A path to the data with bot labels, as either csv or pickled dataframe, or a dataframe
- **targetColumnName** (*str*) – The name of the column, describing whether a user is a bot or not, by default 'isbot'

- **saveFileName** (*str, optional*) – The path of the file, to save the XGB model binary, which can be loaded with loadClassifierModel, by default None
- **update** (*bool, optional*) – Determines whether data will improve current classifier or restart training, by default False
- **iterations** (*int, optional*) – Determines the number of times the classifier training will iterate through data, by default 100
- **hyper_parameter_search** (*bool, optional*) – Determines if the hyper-parameters of the classifier should be search or if the default parameters will be used. The search may be time-consuming for large training datasets and doesn't work with update flag.

2.1.3 birdspotter.user_influence module

birdspotter.user_influence.**P**(*cascade*, *r=-6.8e-05*, *beta=1.0*)

Computes the P matrix of a cascade

The P matrix describes the stochastic retweet graph.

Parameters

- **cascade** (*DataFrame*) – A dataframe describing a single cascade, with a time column ascending from 0, a magnitude column and index of user ids
- **r** (*float, optional*) – The time-decay r parameter described in the paper, by default -0.000068
- **beta** (*float, optional*) – A social strength hyper-parameter, by default 1.0

Returns A matrix of size (n,n), where n is the number of tweets in the cascade, where P[i][j] is the probability that j is a retweet of tweet i.

Return type array-like

birdspotter.user_influence.**casIn**(*cascade*, *time_decay=-6.8e-05*, *alpha=None*, *beta=1.0*)

Computes influence in one cascade

Parameters

- **cascade** (*str or DataFrame*) – Path to one cascade in a file
- **time_decay** (*float*) – The r parameter described in the paper
- **alpha** (*float, optional*) – A float between 0 and 1, as described in the paper. If None DebateNight method is used, else spatial-decay method, by default None

Returns A dataframe describing the influence of each user in a single cascade.

Return type DataFrame

birdspotter.user_influence.**influence**(*p*, *alpha=None*)

Estimates user influence

This function compute the user influence and store it in matirx m_ij

Parameters

- **p** (*array-like*) – The P matrix describing the stochastic retweet graph
- **alpha** (*float, optional*) – A float between 0 and 1, as described in the paper. If None DebateNight method is used, else spatial-decay method, by default None

Returns A n-array describing the influence of n users/tweets and the (n,n)-array describing the intermediary contribution of influence between tweets

Return type array-like, array-like

2.1.4 birdspotter.utils module

birdspotter.utils.**getSource**(*sourcestring*)

birdspotter.utils.**getTextFeatures**(*key*, *text*)

birdspotter.utils.**getURLs**(*string*)

birdspotter.utils.**grep**(*sourcestring*, *pattern*)

birdspotter.utils.**hourofweekday**(*datestring*)

birdspotter.utils.**parse**(*x*)

2.1.5 Module contents

CHAPTER 3

Index

- genindex

Python Module Index

b

Index