influ Documentation

Release 0.2.0

Grzegorz Chilczuk

Jan 07, 2019

Contents:

InFlu is an application for social network analysis for finding key nodes in the social influence process as the main feature. It gives you possibility to choose nodes based on the typical SNA metrics and using two models of social influence: Linear Threshold and Independent Cascade. The are three methods that allow you to use influence models to find the key nodes: brute-force, greedy and improved greedy known as CELF++ proposed in this paper

CHAPTER 1

First steps

Only Python 3.6 or higher are supported.

If you have satisfied dependencies the installation should be as simple as

```
pip install influ
```

1.1 Dependencies

All dependencies will be installed automatically. However one of most important dependencies is cool python library called igraph which core is written in C. Sometimes it may cause some problem, igraph documentation should help.

Another problematic dependency is `pycairo`, here you can find precise documentation.

1.1.1 Debian / Ubuntu and derivatives

Installing those dependencies should help with both igraph and pycairo:

```
sudo apt install build-essential python-dev libxml2 libxml2-dev zlib1g-dev
sudo apt install libcairo2-dev pkg-config
```

1.1.2 Windows

If you are using Windows you have to download unofficial installer of igraph here and install it by executing:

```
pip install <python_igraph-[igraph-version]-[python-version]-[windows version]>.whl
```

Despite that's unofficial it's recommended by maintainers of igraph library.

The `pycairo` library on Windows need `Microsoft Visual C++ 14.0` to be installed.

1.1.3 macOS

There where no attempts to install `influ` on macOS. Any information about it will be appreciated.

1.2 Konect Reader

In order to test your concepts quickly there is a `KonectReader` which simplifies downloading and extracting datasets and loading them into Graph object.

```
from influ import reader

kr = reader.KonectReader()
print(kr.list) # list available datasets
graph = kr.load('manufacturing_emails')  # load dataset
```

Currently there is only few datasets available but you can provide your own config file with other datasets specified. Currently only datasets from Konect are supported.

1.2.1 Your own config file

```
# Content of my_custom_config.yaml
example_dataset:  # name that will be used to access dataset
  name: Example Dataset 1
  url: http://konect.uni-koblenz.de/networks/dataset_examle  # url where dataset is␣
↪described [optional]
  download: http://konect.uni-koblenz.de/downloads/tsv/dataset_examle.tar.bz2  # url␣
↪where dataset can be downloaded directly
  file: out.dataset_example_example  # name of file with
  directed: False  # does graph should be considered as directed?
  edge_attributes:  # list of names attributes
    - distance      # if this list will be empty or there will be more attributes
    - another_attr  # it will be named `attrX` where X is index counted from 0
  vertex_attributes:                # list of vertex attributes with files where they␣
↪are stored
    - name: alias                   # name of attribute
      file: ent.vertex_alias_name   # file with attribute
```

Loading your custom config extends (does not override) those previously loaded.

```
from influ import reader

kr = reader.KonectReader('./my_custom_config.yaml')  # loading at creation time
kr.add_config('./my_custom_config.yaml')       # adding config after creation
```

1.3 Working example

```
from influ import reader, finder

kr = reader.KonectReader()
graph = kr.load('manufacturing_emails')

```

(continues on next page)

(continued from previous page)

```
sfinder = finder.SeedFinder(graph)
sfinder.configure(number=5, unit='number')
result = sfinder.greedy(model=finder.Model.IndependentCascade, depth=1)
sfinder.plot_influence(result, model=finder.Model.IndependentCascade, depth=1)
```

CHAPTER 2

Reader

```
from influ import reader
```

2.1 read_graph

Read graph from file if it is saved in "events" format. Otherwise igraph read method is used. *Parameters*:

- **filepath** [required] - path to file with graph
- **file_format** [[optional, default: None] - optional (but recommended) format name; eg: `events` for InFlu native file format `ncol` for NCOL format
- **kwargs** [[optional] - additional keyword arguments specific to source file type: `directed`, `sep` etc.

2.2 konect_to_csv

Converts konect dataset to format readable by influ *Parameters*:

- **source_path** [required] - path to the "out" file downloaded from konect
- **dest_path** [required] - path to destination where to save a file
- **edge_attrs** [required] - optional edge attributes that are known to exists in out file if this parameter is missing or there are more attributes then unnamed attributes will be named as attr{index}

2.3 KonectReader

Interface to download configured datasets from Konect

```
from influ import reader

kr = reader.KonectReader()
print(kr.list) # list available datasets
graph = kr.load(kr.list[0])
```

2.3.1 KonectReader . list

List of names of available datasets

2.3.2 KonectReader . add_config

Add datasets specified by user in configuration file. Detailed description of configuration file can be found in “First steps - Your own config file” section. *Parameters*:

- **path** [required] - path to config file

2.3.3 KonectReader . load

Loads dataset from Konect, and return loaded graph *Parameters*:

- **dataset_name** [required] - name of the dataset to download; it have to be one of names from specified in `list` attribute

CHAPTER 3

Finder

```
from influ import finder
```

3.1 SeedFinder

This is basic object for finding key nodes in your network. Works for both directed and undirected graphs. Graphs needs to have special structure and because of that only graphs loaded with `reader` module are recommended.

Initialization parameters

- **graph** [required] – graph that will be analysed
- **number** [optional, default: 5] - value of number or percentage of seeds to choose
- **unit** [optional, default: 'number'] - either `percent` or `number`;
- **random_seed** [optional, default: None] – value used as seed for random function to ensure repetitive results;

3.1.1 SeedFinder . configure

Parameters:

- **number** [optional, default: None] – value of number or percentage of seeds to choose; have to be configured together with `unit` parameter
- **unit** [optional, default: None] – either `percent` or `number`; have to be configured together wit `number` parameter
- **random_seed** [optional, default: None] – value used as seed for random function to ensure repetitive results. It's used at the beginning of every model evaluation. If `random_seed` is equal to `None` (default) then no random seed will be used

3.1.2 SeedFinder . by_indegree

Return list of n first vertices indices sorted by their indegree. Takes no parameters.

3.1.3 SeedFinder . by_outdegree

Return list of n first vertices indices sorted by their outdegree. Takes no parameters.

3.1.4 SeedFinder . by_degree

Return list of n first vertices indices sorted by their degree. Takes no parameters.

3.1.5 SeedFinder . by_betweenness

Return list of n first vertices indices sorted by their betweenness. Takes no parameters.

3.1.6 SeedFinder . by_clustering_coefficient

Return list of n first vertices indices sorted by their clustering coefficient (transitivity). IMPORTANT: in directed graph only mutual edges will be considered Takes no parameters.

3.1.7 SeedFinder . greedy

Search for vertices indices that are the best seeds using greedy approach.

Parameters:

- **model** [optional, default: Model.LinearThreshold] - model of social influence. Currently only Linear Treshold (LT) and Independent Cascade (IC) are available
- **threshold** [optional, default: None] - defines value of threshold in influence model. In Linear Threshold model it defines threshold of sum of influence that have to applied to node to activate it. In Independent Cascade model it's probability that activated node activates another node.
- **depth** [optional, default: None] - how many iterations will be in spreading simulations :return: list of ids of nodes considered as the best seeds

3.1.8 SeedFinder . brute_force

Search for vertices indices that are the best seeds using brute force approach.

Parameters:

- **model** [optional, default: Model.LinearThreshold] - model of social influence. Currently only Linear Treshold (LT) and Independent Cascade (IC) are available
- **threshold** [optional, default: None] - defines value of threshold in influence model. In Linear Threshold model it defines threshold of sum of influence that have to applied to node to activate it. In Independent Cascade model it's probability that activated node activates another node.
- **depth** [optional, default: None] - how many iterations will be in spreading simulations :return: list of ids of nodes considered as the best seeds

3.1.9 SeedFinder . CELFpp

Search for vertices indices that are the best seeds using CELF++ approach.

Parameters:

- **model** [optional, default: Model.LinearThreshold] - model of social influence. Currently only Linear Treshold (LT) and Independent Cascade (IC) are available
- **threshold** [optional, default: None] - defines value of threshold in influence model. In Linear Threshold model it defines threshold of sum of influence that have to applied to node to activate it. In Independent Cascade model it's probability that activated node activates another node.
- **depth** [optional, default: None] - how many iterations will be in spreading simulations :return: list of ids of nodes considered as the best seeds

3.1.10 SeedFinder . plot_influence

Run influence simulation for given set of seed and plot result graph.

Parameters:

- **seeds** [required] - list of seed ids for influence spreading simulation
- **model** [optional, default: Model.LinearThreshold] - model of social influence. Currently only Linear Treshold (LT) and Independent Cascade (IC) are available
- **threshold** [optional, default: None] - defines value of threshold in influence model. In Linear Threshold model it defines threshold of sum of influence that have to applied to node to activate it. In Independent Cascade model it's probability that activated node activates another node.
- **depth** [optional, default: None] - how many iterations will be in spreading simulations :return: list of ids of nodes considered as the best seeds

3.2 Model

Social influence model enum

3.2.1 Model . LinearThreshold

Enum value. Represents Linear Threshold Model

3.2.2 Model . IndependentCascade

Enum value. Represents Independent Cascade Model