CoreScope: Graph Mining Using k-Core Analysis – Patterns, Anomalies and Algorithms

Jun-14-2016

Kijung Shin (kijungs@cs.cmu.edu)

1 General Information

- Version: 1.0
- Date: Jun-14-2016
- Authors: Kijung Shin (kijungs@cs.cmu.edu)

2 Introduction

Core Scope v1.0 includes

- streaming algorithm for degeneracy **(*CoreD*)**
- anomaly detection algorithm based on coreness **(*CoreA*)**
- influential spreader detection method based on the structure of degeneracy-cores **(*CoreS*)**

Detailed information about each module is explained in the following paper:

- Kijung Shin, Tina Eliassi-Rad, and Christos Faloutsos, *"**CoreScope: Graph Mining Using k-Core Analysis - Patterns, Anomalies and Algorithms**"*, IEEE International Conference on Data Mining (ICDM) 2016, Barcelona, Spain

3 Installation

- This package requires that java 1.7 or greater be installed in the system and set in PATH.
- For compilation (optional), type *./compile.sh*
- For packaging (optional), type *./package.sh*
- For demo (optional), type *make*

4 Input File Format

The input file lists all edges in a graph. Each line corresponds to an edge, which consists of source node index and destination node index, which are separated by a tab. In addition, we assume the followings:

- Graph is symmetrized, i.e., if a link (u,v) is included in the file, (v,u) is also included.
- Node indices range from 0 to (the number of nodes – 1)

example_graph.tsv is an example of the input file.

5 ***CoreD***: Streaming Algorithm for Degeneracy

5.1 How to Run

```
./run_coreD.sh input_path model sampling_ratio
```

5.2 Parameters

- input_path: path of the input file. See 4 for the detailed format of the input file.
- model: model to use. This parameter should be one among [basic, triangle, overall]
- sampling_ratio: sampling ratio. This parameter should be in [0, 1].

5.3 Output

- Estimated degeneracy of the input graph is printed on the console

6 ***CoreA***: Anomaly Detection Based on Coreness

6.1 How to Run

```
./run_coreA.sh input_path output_path
```

6.2 Parameters

- input_path: path of the input file. See 4 for the detailed format of the input file.
- output_path: path of the output file. See 6.3 for the detailed format of the output file.

6.3 Output

The output file lists nodes and their anomaly score, coreness, and, degree, in the descending order of anomaly score. Each line consists of *ranking* (with regard to anomaly score), *node index*, *anomaly score*, *coreness*, and *degree*, which are separated by tabs.

output_demo/CoreA_result.tsv is an example of the output file.

7 ***CoreA + DSM***: Dense-Subgraph Detection Using CoreA Results

7.1 How to Run

```
./run_comb.sh input_path output_path weight
```

7.2 Parameters

- input_path: path of the input file. See 4 for the detailed format of the input file.
- output_path: path of the output file. See 7.3 for the detailed format of the output file.
- weight: weight that result of CoreA is multiplied by for being balanced with average degree. This parameter should be greater than or equal 0. This parameter should be set to 0 to run simple DSM without CoreA.

7.3 Output

The output file lists nodes belonging to the densest subgraph found by CoreA + DSM.

output_demo/comb_result.tsv is an example of the output file.

8 ***CoreS***: Identifying Spreaders based on the Structure of Degeneracy-Cores

8.1 How to Run

```
./run_coreS.sh input_path output_path spreader_num
```

8.2 Parameters

- input_path: path of the input file. See 4 for the detailed format of the input file.
- output_path: path of the output file. See 8.3 for the detailed format of the output file.
- spreader_num: number of spreaders to find. This parameter should be an integer at least 1.

8.3 Output

The output file lists the identified spreaders with their rank. Each line consists of *rank* and *node index*, which are separated by tabs.

output_demo/CoreS_result.tsv is an example of the output file.

9 SIR Simulation Using Spreaders Identified by CoreS as Seeds.

9.1 How to Run

```
./run_simulation.sh input_path output_path spreader_num infection_rate repetition_num
```

9.2 Parameters

- input_path: path of the input file. See 4 for the detailed format of the input file.
- output_path: path of the output file. See 9.3 for the detailed format of the output file.
- spreader_num: number of spreaders to find. This parameter should be an integer at least 1.

- Infection_rate: probability that an infected node infects each of its neighbors. This parameter should be in (0,1)

- repetition_num: number of repetitions of simulation for each seed. This parameter should be an integer at least 1.

9.3 Output

The output file lists the average number of infected nodes for each seed, which is a spreader identified by CoreS. Each line consists of *node index* and *the average number of infected nodes* when the node is used as a seed. They are separated by tabs.

output_demo/simulation _result.tsv is an example of the output file.