dynamic-community

Dynamic Community Tracking Tool
Derek Greene, Insight Centre @ University College Dublin

Overview

The Dynamic Community Tracking Tool is a console application written in C++ for identifying and tracking communities of nodes in dynamic networks, where these networks are represented as a set of step graphs representing snapshots of the network at successive time periods.
The tool takes as its input a series of two or more sets of step communities, produced on individual time step graphs using a standard static community finding algorithm, such as MOSES or the Louvain algorithm. These step communities can overlapping or non-overlapping. Note that all nodes need be present in consecutive time step graphs, some degree of overlap is sufficient.
The tool builds dynamic community timelines from sequences of individual step communities, which can be used to chart the evolution of these dynamic communities over time.
For further details on the algorithm and its applications, please consult the following paper:

- D.Greene, D.Doyle, and P.Cunningham, "Tracking the evolution of communities in dynamic social networks," in Proc. International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10), 2010.
  [PDF] [Supplementary material]

Compilation

To compile all of the command line tools, in the source root directory run:

```
make all
```

Usage

The main **tracker** tool is run from the command line as follows:

```
./tracker [-t matching_threshold] [-o output_prefix] <step1_communities> <step2_communities>
...
```

Parameter explanation:

- The optional parameter *matching_threshold* is a value [0,1] indicating the threshold required to match communities between time steps. A higher value indicates a more conservative matching threshold. Low values are suitable for data where community memberships are expected to be transient over time, high values are suitable where community memberships are expected to be consistent over time.

  The default threshold value is 0.1.
- The optional parameter *output_prefix* provides a string that is added as a prefix to the output files produced by the tool. The default prefix is “dynamic”.
- The subsequent parameters correspond to a list of paths of input files containing step communities, with one file per step. The first file is assumed to correspond to the first time step, the second file to the second time step, and so on. The format for the input files is given in the next section.
  For example, to apply the tool to a number of step community files, with a matching threshold of 0.3 and output prefix of “res”:
  ./tracker -t 0.3 -o res sample/sample.t*.comm

Input Format

Each plain text input file for the tracker tool contains one or more step communities, with one line corresponding to each community. The entries on each line correspond to the node identifiers (positive numeric values) separated by spaces. Note that node identifier numbers need not be consecutive, or ordered in the file.

Below shows a simple example of an input file containing three overlapping communities:

```
1 2 3 10 4
5 3 6 7 8 9
10 11 12 1 4
```

Timeline Output Format

The output of the tracker tool is a single file, where each line in the file correspond to the timeline of a dynamic community. The entries in each line correspond to the sequence of associated step community observations which form that dynamic community.
Below shows a simple example of an output file containing two dynamic communities over three time steps:

```
M1:1=1,2=2,3=1
M2:2=2,3=1
```

For example, in the case of the second dynamic community (named “M2”), the dynamic community was not observed at t = 1, and consists of the 2nd step community at time t = 2, and the 1st step community at time t = 3. These step community indices correspond to the line numbers in the original input files supplied to the tracker tool.

Producing Communities

To produce a set of dynamic communities in the same format as the input file, the **aggregator** tool is run from the command line as follows:

./aggregator -i <timeline_file> [-o output_prefix] [-p persist_threshold] [-m max_step] [-l min_length] <step1_communities> <step2_communities> ...

Parameter explanation:

- The mandatory parameter *timeline_file* corresponds to the name of the output file from the tracker tool. The default prefix is “dynamic”.
- The optional parameter *output_prefix* provides a string that is added as a prefix to the output community files produced by the tool.
- The optional parameter *persist_threshold* specifies the proportion of time steps required for a node to be deeemed to be a member of a community. By default, a node is only required to appear in a single time step community.
- The optional parameter *min_length* is an integer indicating the minimum length (in terms of number of steps) for a dynamic community to be included in the final results. By default, a dynamic community must be present in at least two time steps.
- The optional parameter *max_step* indicates the maximum step number for which communities should be included. Typically this should correspond to the number of step community files specified. By default process all step communities specified.
  For instance, having generated a dynamic timeline file res.timeline using the **tracker** tool, a final set of potentially overlapping communities can be produced as follows, where we require nodes to appear in a dynamic community across 50% of the time steps:
  ./aggregator -i res.timeline -p 0.5 -o res sample/sample.t*.comm

The resulting output file, res.persist, contains communities in the same format as the original input file (i.e. one community per line, specified in terms of node identifiers):

```
1 2 3 4
5 6 7 8 9
10 11 12
```