

Think before You Discard: Accurate Triangle Counting in Graph Streams with Deletions (Software User Guide)

Kijung Shin (kijungs@cs.cmu.edu)

1 General Information

- Version: 1.0
- Date: March 12, 2018
- Author: Kijung Shin (kijungs@cs.cmu.edu)

2 Introduction

THINKD (**Think** before you **Discard**) is a streaming algorithm for triangle counting in a fully dynamic graph stream with edge additions and deletions. THINKD estimates the counts of global triangles and local triangles by making a single pass over the stream. THINKD has the following advantages:

- *Accurate*: THINKD is up to $4.3\times$ more accurate than its best competitors within the same memory budget
- *Fast*: THINKD is up to $2.2\times$ faster than its best competitors for the same accuracy requirements
- *Theoretically Sound*: THINKD always maintains unbiased estimates.

Detailed information about THINKD is explained in the following paper:

- Kijung Shin, Jisu Kim, Bryan Hooi, and Christos Faloutsos, “Think before You Discard: Accurate Triangle Counting in Graph Streams with Deletions”, ECML/PKDD 2018

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 the additions and deletions in an **undirected** and **unweighted** graph in the order that they arrive. Each line corresponds to an edge addition or deletion. Each line consists of a source node id, a destination node id, and an indicator (1 for addition and -1 for deletion), which are integers separated by a tab. Additionally, we assume that there are **no parallel edges**. That is, if an edge has been added and has not been deleted yet, the same edge cannot be added. See *example_graph.txt* for an example input file.

5 Output File Format

Two output files are created for each trial:

- *global(trial#).txt*: this file has the estimated number of global triangles.
- *local(trial#).txt*: this file lists the estimated number of local triangles of each node. Each line consists of the node id and the number of its local triangle count, separated by a tab.

The directory named *output_fast* contains example output files.

6 Running ThinkD_{FAST} (Batch Mode)

6.1 How to Run

`./run_fast.sh input_path output_path sampling_ratio number_of_trials`

6.2 Parameters

- *input_path*: path of the input file. See Section 4 for the detailed format of the input file
- *output_path*: path of the directory for output files. See Section 5 for the detailed format of the output files
- *sampling_ratio*: probability that each added edge is sampled
- *number_of_trials*: number of trials

7 APIs for ThinkD_{FAST} (Incremental Mode)

7.1 Package: *thinkd*

7.2 Class: *ThinkDFast*

7.3 Methods:

- public *ThinkDFast* (double *sampling_ratio*, int *random_seed*):
 - create a *ThinkDFast* object
 - *sampling_ratio*: probability that each added edge is sampled
 - *random_seed*: a non-negative integer
- public void *processAddition* (int *src*, int *dst*)
 - insert an edge
 - *src*: id of the source node
 - *dst*: id of the destination node
- public void *processDeletion* (int *src*, int *dst*)
 - delete an edge
 - *src*: id of the source node
 - *dst*: id of the destination node
- public double *getGlobalTriangle*()
 - return the estimated number of global triangles
- public *it.unimi.dsi.fastutil.ints.Int2DoubleMap* *getLocalTriangle*()
 - return the estimated numbers of local triangles
 - return: a map whose keys are node ids and values the estimated number of local triangle counts of the corresponding nodes

7.4 Example Code:

See *ExampleFast.java* for an example code using *ThinkDFast*.

8 Running ThinkD_{ACC} (Batch Mode)

8.1 How to Run

`./run_acc.sh input_path output_path memory_budget number_of_trials`

8.2 Parameters

- *input_path*: path of the input file. See Section 4 for the detailed format of the input file
- *output_path*: path of the directory for output files. See Section 5 for the detailed format of the output files
- *memory_budget*: maximum number of sampled edges (an integer greater than or equal to 2)
- *number_of_trials*: number of trials

9 APIs for ThinkD_{ACC} (Incremental Mode)

9.1 Package: *thinkd*

9.2 Class: *ThinkDAcc*

9.3 Methods:

- public *ThinkDAcc* (double *sampling_ratio*, int *random_seed*):
 - create a *ThinkDAcc* object
 - *memory_budget*: maximum number of sampled edges (an integer greater than or equal to 2)
 - *random_seed*: a non-negative integer
- public void *processAddition* (int *src*, int *dst*)
 - insert an edge
 - *src*: id of the source node
 - *dst*: id of the destination node
- public void *processDeletion* (int *src*, int *dst*)
 - delete an edge
 - *src*: id of the source node
 - *dst*: id of the destination node
- public double *getGlobalTriangle*()
 - return the estimated number of global triangles
- public *it.unimi.dsi.fastutil.ints.Int2DoubleMap* *getLocalTriangle*()
 - return the estimated numbers of local triangles
 - return: a map whose keys are node ids and values the estimated number of local triangle counts of the corresponding nodes

9.4 Example Code:

See *ExampleAcc.java* for an example code using *ThinkDAcc*.