Instructions to run BCT for Influence Maximization

Information:

Version 1.0: Implementation of BCT Algorithms for Cost-aware Targeted Influence Maximization (IM) under Independent Cascade(IC) or Linear Threshold(LT) models. For more details about BCT and the cost-aware targeted IM, please read our paper:

H. T. Nguyen, M. T. Thai, and T. N. Dinh, Cost-aware Targeted Viral Marketing in Billion-scale Networks, in Proc. of the IEEE INFOCOM International Conference on Computer Communication, 2016.

Contact Authors: Hung T Nguyen (hungnt@vcu.edu)

For Terms of use, please check the LISENCE file for details.

How to use:

This package offers a set of functions to use in order to find a seed set of cost at most k (given as input):

0. (Optional) Computing edge weights (probabilities) as described in the experiments:

```
./format <input file> <output file> 1
```

<input file>: the path to text file in edge list format with no weights: the first line contains the number of nodes n and number of edges m, each of the next m lines describes an edge following the format: <src> <dest>. Node index starts from 1.

<output file>: the same format as input file except that for each edge, the computed weights are added (following each edge).

The last parameter (1) means the input graph is considered as directed.

1. Conversion from a text format to binary file

```
./el2bin <input file> <output file>
```

<input file>: the path to text file in edge list format: the first line contains the number of nodes n and number of edges m, each of the next m lines describes an edge following the format: <src> <dest> <weight>. Node index starts from 1.

<output file>: the path to binary output file

2. Run BCT to find the seed sets

```
./bct [Options]
```

Options:

-i <binary graph file>

specify the path to the binary graph file (default: network.bin). This file is generated by el2bin.

-b <node benefit file>

specify the path to the file listing the node benefit (default: network.ben). Each line in the file contains a single number indicating the benefit of influencing the node with the identity being the same as line number.

-c <node cost file>
specify the path to the file listing the node cost (default: network.cost). The format is similar to network.ben.

-o <seed output file>
specify the path to the output file containing selected seeds (default: network.seeds)

-k <budget>
maximum cost of the selected seed set (default: 1)

-epsilon <epsilon value used>
epsilon value in (epsilon,delta)-approximation (see our paper for more details, default: 0.1)

-delta <delta value used>
delta value in (epsilon,delta)-approximation (default: 1/n)

-m <model>
diffusion model (LT or IC, default: LT)

Output format:
The outputs are printed on standard output stream in the following order

```
Seed Nodes: <list of selected seed nodes>

Total Benefit: <Spread of Influence of the select seed set>
Time: <running time in seconds>
Memory: <peak memory used>
```

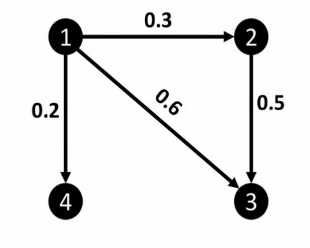
3. (Optional) Verify influence spread of a seed set - returns a (epsilon, 1/n)-estimate of the influence:

```
./verifyBen <binary graph file> <seed file> <epsilon> <model: LT or IC> <benefit file>
```

Examples on a toy network: find seed set having 2 seed nodes on the graph network.txt:

The sample network network.txt, in this case, contains only 4 nodes and 4 edges and is formated as follows:

```
4 4
1 2 0.3
2 3 0.5
1 3 0.6
1 4 0.2
```



The benefit file has the content as follows (every node has benefit of 1):

```
1
1
```

```
1
1
```

The cost file has the content as follows (every nodes has cost of 1):

```
1
1
1
1
```

1. Convert to binary file:

```
./el2bin network.txt network.bin
```

2. Run BCT with k=2, epsilon=0.1,delta=0.01:

```
./bct -i network.bin -b network.ben -c network.cost -k 2 -epsilon 0.1 -delta 0.01 -m LT
```

The output after running BCT:

```
Seed Nodes: 1 2

Total Benefit: 3.01
Time: 0s
Memory: 20.7422 MB
```

Here, the selected seed set is {1;2} which has cost of 2 with the estimated influence of 3.01. BCT took 0 second to run and consumed 20.7422 MB of memory.

3. Verify influence spread with epsilon=0.01, assume that the seed nodes are put in seeds.txt:

```
./verifyBen network.bin network.seeds 0.01 LT network.ben
```

The output after running BCT:

```
3.01
```