UNIVERSITY OF TWENTE.

Quick detection of nodes with large degrees

Nelly Litvak

University of Twente,
Stochastic Operations Research group

NADINE meeting, 14-06-2013

Finding top-k largest degree nodes

with Konstantin Avrachenkov, Marina Sokol, Don Towsley

Finding top-k largest degree nodes

with Konstantin Avrachenkov, Marina Sokol, Don Towsley

What if we would like to find in a network top-k nodes with largest degrees?

Some applications:

- Routing via large degree nodes
- Proxy for various centrality measures
- Node clustering and classification
- Epidemic processes on networks

Top-k largest degree nodes

If the adjacency list of the network is known...

the top-k list of nodes can be found by the HeapSort
with complexity $O(n + klog(n))$, where n is the total number of nodes.

Even this modest complexity can be quite demanding for large networks.

Random walk approach

Let us now try a random walk on the network.

We actually recommend the random walk with jumps with the following transition probabilities:

$$p_{ij} = \begin{cases} \frac{\alpha/n+1}{d_i+\alpha}, & \text{if } i \text{ has a link to } j, \\ \frac{\alpha/n}{d_i+\alpha}, & \text{if } i \text{ does not have a link to } j, \end{cases} \quad (1)$$

where d_i is the degree of node i and α is a parameter.

Random walk approach

Let us now try a random walk on the network.

We actually recommend the random walk with jumps with the following transition probabilities:

$$p_{ij} = \begin{cases} \frac{\alpha/n+1}{d_i+\alpha}, & \text{if } i \text{ has a link to } j, \\ \frac{\alpha/n}{d_i+\alpha}, & \text{if } i \text{ does not have a link to } j, \end{cases} \qquad (1)$$

where d_i is the degree of node i and α is a parameter.
The introduced random walk is time reversible, its stationary distribution is given by a simple formula

$$\pi_i(\alpha) = \frac{d_i + \alpha}{2|E| + n\alpha} \quad \forall i \in V. \qquad (2)$$

Random walk approach

Example:

If we run a random walk on the web graph of the UK domain (about 18 500 000 nodes), the random walk spends on average only about 5 800 steps to detect the largest degree node.

Three order of magnitude faster than HeapSort!

Random walk approach

We propose the following algorithm for detecting the top k list of largest degree nodes:

1. Set k, α and m.
2. Execute a random walk step according to (1). If it is the first step, start from the uniform distribution.
3. Check if the current node has a larger degree than one of the nodes in the current top k candidate list. If it is the case, insert the new node in the top-k candidate list and remove the worst node out of the list.
4. If the number of random walk steps is less than m, return to Step 2 of the algorithm. Stop, otherwise.

How to choose α

W_t – state of the random walk at time $t = 0, 1, \ldots$

$$P_\pi[W_t = i|\text{jump}] = \frac{1}{n}, \quad P_\pi[W_t = i|\text{no jump}] = \frac{d_i}{2|E|} = \pi_i(0)$$

α is too small: the random walk can gets 'lost' in the network.

α is too large: jumps are too frequent, no useful information

How to choose α

W_t – state of the random walk at time $t = 0, 1, \ldots$

$$P_\pi[W_t = i|\text{jump}] = \frac{1}{n}, \quad P_\pi[W_t = i|\text{no jump}] = \frac{d_i}{2|E|} = \pi_i(0)$$

α is too small: the random walk can gets 'lost' in the network.

α is too large: jumps are too frequent, no useful information

Maximize the long-run fraction of independent samples from $\pi(0)$

$$\frac{1 - P_\pi[\text{ jump}]}{(P_\pi[\text{ jump}])^{-1}} = P_\pi[\text{ jump}](1 - P_\pi[\text{ jump}]) \to \max.$$

How to choose α

W_t – state of the random walk at time $t = 0, 1, \ldots$

$$P_\pi[W_t = i|\text{jump}] = \frac{1}{n}, \quad P_\pi[W_t = i|\text{no jump}] = \frac{d_i}{2|E|} = \pi_i(0)$$

α is too small: the random walk can gets 'lost' in the network.

α is too large: jumps are too frequent, no useful information

Maximize the long-run fraction of independent samples from $\pi(0)$

$$\frac{1 - P_\pi[\text{ jump}]}{(P_\pi[\text{ jump}])^{-1}} = P_\pi[\text{ jump}](1 - P_\pi[\text{ jump}]) \to \max .$$

$$P_\pi[\text{jump}] = \frac{n\alpha}{2|E| + n\alpha} = \frac{1}{2} \qquad \alpha = 2|E|/n = \text{average degree.}$$

Stopping rules

- Objective: on average at least $\bar{b}$ of the top k nodes are identified correctly.
- Let us compute the expected number of top k elements observed in the candidate list up to trial m.

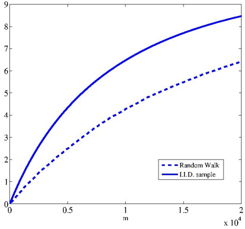
$$H_j = \begin{cases} 1, & \text{node } j \text{ has been observed at least once,} \\ 0, & \text{node } j \text{ has not been observed.} \end{cases}$$

Assuming we sample in i.i.d. fashion from the distribution (2), we can write

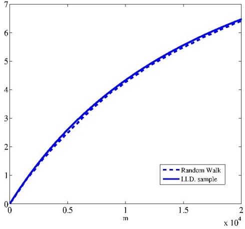
$$E[\sum_{j=1}^{k} H_j] = \sum_{j=1}^{k} E[H_j] = \sum_{j=1}^{k} P[X_j \geqslant 1] =$$

$$\sum_{j=1}^{k} (1 - P[X_j = 0]) = \sum_{j=1}^{k} (1 - (1 - \pi_j)^m). \qquad (3)$$

Stopping rules (cont.)



(a) $\alpha = 0.001$



(b) $\alpha = 28.6$

Figure: Average number of correctly detected elements in top-10 for UK.

Stopping rules (cont.)

Here we can use the Poisson approximation

$$E[\sum_{j=1}^{k} H_j] \approx \sum_{j=1}^{k} (1 - e^{-m\pi_j}).$$

and propose stopping rule. Denote

$$b_m = \sum_{i=1}^{k} (1 - e^{-X_{j_i}}).$$

Stopping rule: Stop at $m = m_0$, where

$$m_0 = \arg\min\{m : b_m \geqslant \bar{b}\}.$$

Example

- UK domain, about 18 500 000 nodes
- The random walk spends on average only about 5 800 steps to detect the largest degree node
- With $\bar{b} = 7$ we obtain on average 9.22 correct elements out of top-10 list for an average of 65 802 random walk steps for the UK network.

Example

- UK domain, about 18 500 000 nodes
- The random walk spends on average only about 5 800 steps to detect the largest degree node
- With $\bar{b} = 7$ we obtain on average 9.22 correct elements out of top-10 list for an average of 65 802 random walk steps for the UK network.

Directed networks: Twitter

with Konstantin Avrachenkov and Liudmila Ostroumova

Directed networks: Twitter

with Konstantin Avrachenkov and Liudmila Ostroumova

- Huge network (more than 500M users)

Directed networks: Twitter

with Konstantin Avrachenkov and Liudmila Ostroumova

- Huge network (more than 500M users)
- Network accessed only through Twitter API

Directed networks: Twitter

with Konstantin Avrachenkov and Liudmila Ostroumova

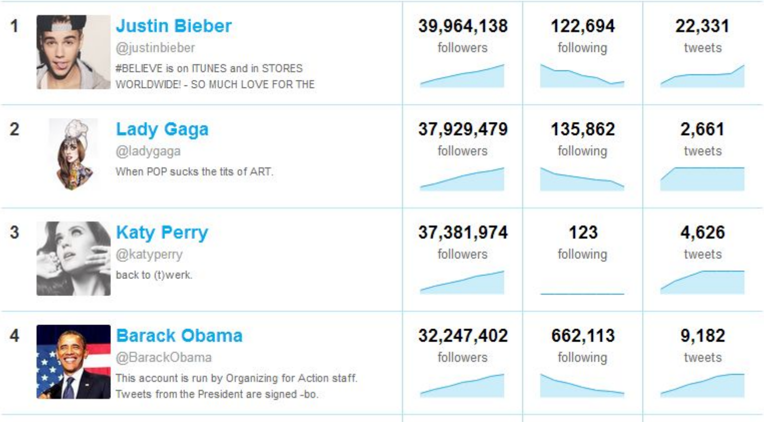
- Huge network (more than 500M users)
- Network accessed only through Twitter API
- The rate of requests is limited
- One request:
  - ID's of at most 5000 followers of a node, or
  - the number of followers of a node

Random walk?

Random walk quickly arrives to a large node and cannot randomly sample from its followers/followees because it is much more than 5000

Random walk?

Random walk quickly arrives to a large node and cannot randomly sample from its followers/followees because it is much more than 5000



1	Justin Bieber @justinbieber #BELIEVE is on ITUNES and in STORES WORLDWIDE! - SO MUCH LOVE FOR THE	39,964,138 followers	122,694 following	22,331 tweets
2	Lady Gaga @ladygaga When POP sucks the tits of ART.	37,929,479 followers	135,862 following	2,661 tweets
3	Katy Perry @katyperry back to (t)werk.	37,381,974 followers	123 following	4,626 tweets
4	Barack Obama @BarackObama This account is run by Organizing for Action staff. Tweets from the President are signed -bo.	32,247,402 followers	662,113 following	9,182 tweets

Algorithm for finding top-k most followed on Twitter

1. Choose n_1 nodes at random
2. Retrieve the id's of at most 5000 users followed by each of the n_1 nodes
3. Let S_j be the number of followers of node j discovered among the n_1 nodes
4. Check the number of followers for n_2 users with the largest values of S_j
5. Return the identified top-k most followed users

In total, there are $n = n_1 + n_2$ requests to API

Performance prediction

- Heuristic: Let $1, 2, \ldots, k$ be the top-k nodes
- Approximate the probability that the node j is discovered by

$$P(S_j > \max\{S_{n_2}, 1\})$$

Then the fraction of correctly identified nodes is

$$\frac{1}{k}\sum_{j=1}^{k} P(S_j > \max\{S_{n_2}, 1\})$$

and S_j have approximately $Poisson(n_1 d_j / N)$ distribution, where N is the number of users

Extreme value theory

Theorem (Extreme value theory)

$D_1, D_2, \dots, D_n$ are i.i.d. with $1 - F(x) = P(D > x) = Cx^{-\alpha+1}$. Then

$$\lim_{n\to\infty} P\left(\frac{\max\{D_1, D_2, \dots, D_n\} - b_n}{a_n} \leqslant x\right) = \exp(-(1+\delta x)^{-1/\delta}),$$

with $\delta = 1/(\alpha - 1)$, $a_n = \delta C^\delta n^\delta$, $b_n = C^\delta n^\delta$.
(Therefore, the maximum is 'of the order' $n^{1/(\alpha-1)}$)

Prediction based on identified top-m, $m < k$

- We do not know $d_1, d_2, \ldots, d_n$ but we can predict their value using the quantile estimation from the Extreme Value Theory (Dekkers et al, 1989):

$$\hat{d}_j = d_m \left(\frac{m}{j-1}\right)^{\hat{\gamma}}, \qquad j > 1, j << N,$$

where

$$\hat{\gamma} = \frac{1}{m-1} \sum_{i=1}^{m-1} \log(d_i) - \log(d_m).$$

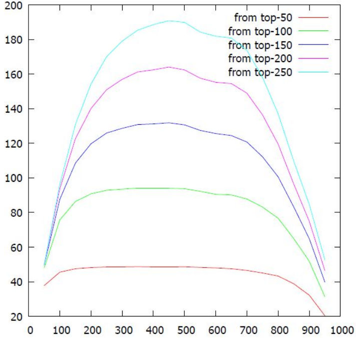
- If m is small enough then we can be almost sure that we discovered top-m correctly.

Caveats in the prediction based on top-m, $m < k$

- We do not know the top-m degrees either. However, we can find them with high precision.

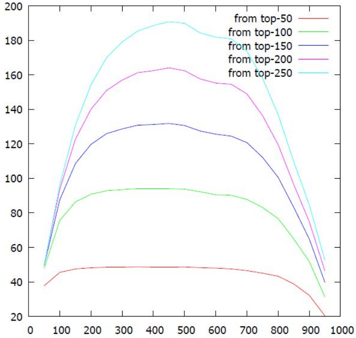
Caveats in the prediction based on top-m, $m < k$

- We do not know the top-m degrees either. However, we can find them with high precision.



Caveats in the prediction based on top-m, $m < k$

- We do not know the top-m degrees either. However, we can find them with high precision.



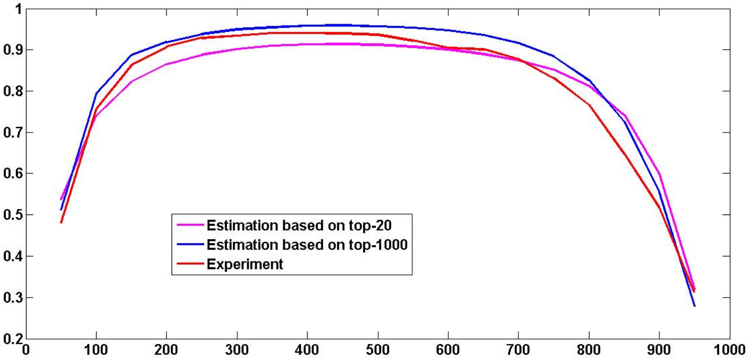
- The consistency of the estimator $\hat{d}_j$ is proved for $j < m$ but we use it for $j > m$. Can we prove the consistency, and if not: can we encounter some pathological behaviour?

Results

$n = 1000$, $n = n_1 + n_2$, $N = 500M$, $k = 100$

Results

$n = 1000$, $n = n_1 + n_2$, $N = 500M$, $k = 100$



Fraction of correctly identified top-100 nodes as a function of n_1

Predictions of trends in retweet graph

with Marijn ten Thij, TNO

Predictions of trends in retweet graph

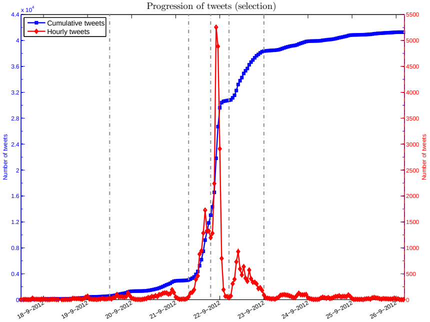
with Marijn ten Thij, TNO

- Data: Project X Haren, 21-09-2012

Predictions of trends in retweet graph

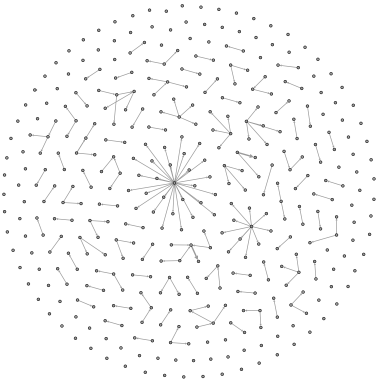
with Marijn ten Thij, TNO

- Data: Project X Haren, 21-09-2012
- Retweet graph: a link between two users if one of them retweeted the other



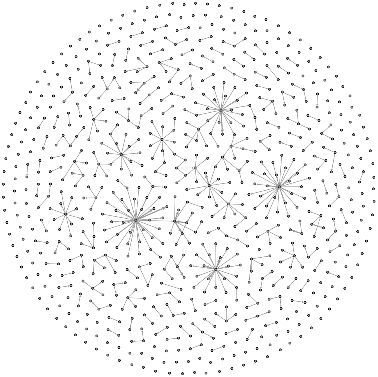
Graph structure

19-9-2012 12:00



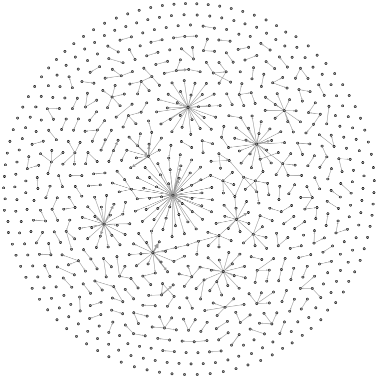
Graph structure

19-9-2012 23:00



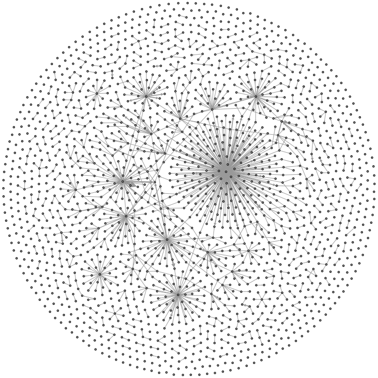
Graph structure

20-9-2012 00:00



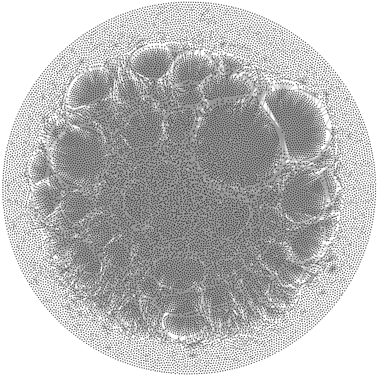
Graph structure

21-9-2012 07:00



Graph structure

22-9-2012 05:00



Ongoing work

- Connection between graph structures and important trends

Ongoing work

- Connection between graph structures and important trends
- Mathematical modelling

Ongoing work

- Connection between graph structures and important trends
- Mathematical modelling
- Possible future topic: trend prediction

Thank you!