Benchmarking API Costs of Network Sampling Strategies

Michele Coscia & Luca Rossi
ITU København

August 27th, 2019
(i) Why Network Sampling?
(i) Why Network Sampling?

(ii) Are there understudied real world obstacles that should make us reconsider how we choose the best sampling strategy?
(i) Why Network Sampling?
The Observation Problem
The Observation Problem
The Observation Problem
~330M monthly users
~330M monthly users
~1.1m per user
~330M monthly users
~1.1m per user
~21.78B seconds
~690 years
~330M monthly users
~1.1m per user
~21.78B seconds
~690 years
1 year of crawling
Network Exploration

- BFS, DFS
- Random Walks
- Snowball
- Forest Fire
Network Exploration

• BFS, DFS  ➔ Full exploration as the objective

• Random Walks
• Snowball
• Forest Fire
Network Exploration

- BFS, DFS
- Full exploration as the objective
- Random Walks
- Snowball
- Forest Fire

Preventing bias from samples
Random Walks
Random Walks

Name 1 of your friends
Random Walks

Name 1 of your friends
Random Walk

Name 1 of your friends

Name 1 of your friends
Random Walk

Name 1 of your friends

Name 1 of your friends
Degree Bias
Degree Bias

• Stationary distr $\pi$
Degree Bias

- Stationary distr $\pi$
- $\pi = \text{degree}$
Degree Bias

- Stationary distr $\pi$
- $\pi = \text{degree}$
- Oversampled hubs!
Metropolis-Hastings

Name 1 of your friends
Metropolis-Hastings

Name 1 of your friends
Metropolis-Hastings

$p \sim k_v / k_u$

Name 1 of your friends
Metropolis-Hastings

$p \sim \frac{k_v}{k_u}$

Name 1 of your friends

Mmm… Name another
Re-Weighted RW
Re-Weighted RW

- Perform vanilla RW
Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest
Re-Weighted RW

- Perform vanilla RW

- Re-weight property of interest

\[ p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} x_{v'}^{-1}} \]
Re-Weighted RW

• Perform vanilla RW

• Re-weight property of interest

\[ p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} x_{v'}^{-1}} \]
Re-Weighted RW

• Perform vanilla RW

• Re-weight property of interest

\[
p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} X_{v'}}
\]

p of nodes with value i  
Set of nodes with value i
Re-Weighted RW

• Perform vanilla RW

• Re-weight property of interest

\[ p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} X_{v'}^{-1}} \]

Set of nodes with value \( i \)
Set of nodes in the sample

Set of nodes with value \( i \)
Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest

\[ p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} x_{v'}} \]

- Set of nodes with value i
- Value for v'
- Set of nodes in the sample
- Set of nodes with value i
Re-Weighted RW

- Perform vanilla RW
- Re-weight property of interest
- Respondent-Driven Sampling

\[ p_i = \frac{\sum_{v \in V_i} i^{-1}}{\sum_{v' \in V} x_{v'}} \]

- \( p_i \): Probability of nodes with value \( i \)
- \( V_i \): Set of nodes with value \( i \)
- \( x_{v'} \): Value for \( v' \)
- \( V \): Set of nodes in the sample
- \( i^{-1} \): Inverse of the value \( i \)
Re-Weighted RW: Example
Re-Weighted RW: Example

• $p$ of a node having $k=2$?
Re-Weighted RW: Example

• p of a node having k=2?

• Observed: 20 over 100 (p = 0.2)
Re-Weighted RW: Example

• p of a node having k=2?
• Observed: 20 over 100 (p = 0.2)
• Other nodes:
  – k=1: 50
  – k=3: 10
  – k=4: 8
  – k=5: 7
  – k=6: 5
Re-Weighted RW: Example

- $p$ of a node having $k=2$?
- Observed: 20 over 100 ($p = 0.2$)
- Other nodes:
  - $k=1$: 50
  - $k=3$: 10
  - $k=4$: 8
  - $k=5$: 7
  - $k=6$: 5

\[
p_2 = \frac{20 \times 1/2}{(50/1) + (20/2) + (10/3) + (8/4) + (7/5) + (5/6)}
\]
Re-Weighted RW: Example

• $p$ of a node having $k=2$?
• Observed: 20 over 100 ($p = 0.2$)
• Other nodes:
  - $k=1$: 50
  - $k=3$: 10
  - $k=4$: 8
  - $k=5$: 7
  - $k=6$: 5

\[
p_2 = \frac{20 \times 1/2}{\frac{50}{1} + \frac{20}{2} + \frac{10}{3} + \frac{8}{4} + \frac{7}{5} + \frac{5}{6}}
\]

\[
p_2 = \frac{10}{67.56} \sim 0.148
\]
Snowball
Snowball

Name k of your friends
Snowball

Name $k$ of your friends
Snowball

Name $k$ of your friends

Name $k$ of your friends
Snowball: Advantages
Snowball: Advantages

• Cheap in the physical world

Name k of your friends

Name k of your friends
Snowball: Advantages

• Cheap in the physical world

• Smaller degree bias
Snowball: Advantages

- Cheap in the physical world
- Smaller degree bias
- Works well with pagination
Forest Fire

Name all your friends
Forest Fire

Name all your friends
Forest Fire

Name all your friends
Forest Fire

Name all your friends

Name all your friends
The Network Sampling Zoo

(a) BFS

(b) DFS

(c) Snowball

(d) Random Walk

(e) MHRW

(f) Forest Fire
(ii) Are there understudied real world obstacles that should make us reconsider how we choose the best sampling strategy?
Social Media APIs
Social Media APIs

Size of the page
Social Media APIs

Size of the page

Length of the interval
Social Media APIs

Length of the interval

Size of the page

(...latency)
Pagination Paradox
Pagination Paradox

- Edges per page: 100
- Seconds between queries: 2
- 50 edges / sec
Pagination Paradox

- Edges per page: 100
- Seconds between queries: 2
- 50 edges / sec

- Edges per page: 10
- Seconds between queries: 1
- 10 edges / sec
Pagination Paradox
Pagination Paradox

10 edges / sec
Pagination Paradox

![Graph showing the distribution of counts for two different rates of edges per second: 10 edges/sec and 50 edges/sec. The x-axis represents the value of k, and the y-axis represents the count. The graph illustrates the paradox where higher rates of edges per second lead to lower counts for certain values of k.]
Pagination Paradox

Way more nodes here

Count

10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^2
10^1
10^0

10 edges / sec

50 edges / sec

k
Pagination Paradox

Way more nodes here

50 edges / sec

10 edges / sec

10^0 10^1 10^2

Count

k

5x slower in theory
2x faster in practice
Benchmark Setup

• Three types of topologies:
  − Barabasi-Albert
  − Small World
  − LFR Benchmark
Benchmark Setup

• Three types of topologies:
  – Barabasi-Albert
  – Small World
  – LFR Benchmark
Benchmark Setup

- Three types of topologies:
  - Barabasi-Albert
  - Small World
  - LFR Benchmark
Benchmark Setup

- Three types of topologies:
  - Barabasi-Albert
  - Small World
  - LFR Benchmark
Benchmark Setup

• Six API systems from real social media:
  – Flickr
  – Lastfm
  – Twitter
  – Youtube
  – Tumblr
  – Google+
Benchmark Setup

• Different objectives:
  - Degree Distribution
  - Assortativity / Disassortativity
  - Centrality
  - Reciprocity
Benchmark Setup
Benchmark Setup

Quality Measure

(NB: not always “lower is better”)

Budget Level

Low Budget = Few edges  High Budget = Many edges
Disassortativity MAE
(lower is better)

Tumblr

LastFM

MAE

Relative Budget

0.14
0.12
0.1
0.08
0.06
0.04
0.02
0

10^3
10^4
10^5
10^6

0.025
0.02
0.015
0.01
0.005
0

10^3
10^4
10^5
10^6

BFS  DFS  SBS  RW  MHRW  RWRW  FF
Assortativity MAE

(lower is better)

**SW**

**B-A**

**LFR**
Budget Levels

Degree Distribution
(lower is better)

Centrality Correlation
(higher is better)
Conclusion
Conclusion

• We have to sample
Conclusion

- We have to sample
- We have good theory...
Conclusion

- We have to sample
- We have good theory...
- ...for the case of infinite time and paging sizes
Conclusion

• We have to sample
• We have good theory...
• ...for the case of infinite time and paging sizes
• Which is not realistic
Conclusion

- We have to sample
- We have good theory...
- ...for the case of infinite time and paging sizes
- Which is not realistic
- Realistic constraints paint a critical picture
Thanks

Benchmarking API Costs of Network Sampling Strategies

Michele Coscia & Luca Rossi
mcos@itu.dk lucr@itu.dk
http://www.michelecoscia.com