Networks Structure and Dynamics
8. Efficient measurements using link queries

Maximilien Danisch, Marwan Ghanem, Lionel Tabourier

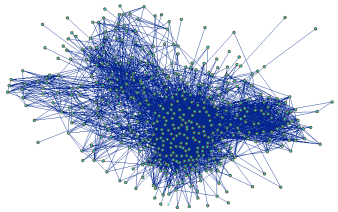
LIP6 – CNRS and Sorbonne Université

first_name.last_name@lip6.fr

December 4th 2018

Outline

Context



Some characteristics related to the size of the networks:

- Partial knowledge of the topology
- Rules out a complete exploration of the network

$\Longrightarrow$ leads to reconsider traditional approaches

Partial view vs. general properties

Goal : Measuring the networks

Assuming that:

- All the elements (nodes) are known
- We don't know the interactions (links) between them
- One can test the existence of a link between two nodes
  $\Rightarrow$ the experimenter makes queries

How to define good strategies for ordering the link queries in order to have quickly a representative sample of the network?

Proposed method:

1. Partial random exploration
2. Compute relevant statistical properties
3. Predict the existing links from these properties

Goal : Measuring the networks

Assuming that:

- All the elements (nodes) are known
- We don't know the interactions (links) between them
- One can test the existence of a link between two nodes
  $\Rightarrow$ the experimenter makes queries

How to define good strategies for ordering the link queries in order to have quickly a representative sample of the network?

Proposed method:

1. Partial random exploration
2. Compute relevant statistical properties
3. Predict the existing links from these properties

Common properties – recall

Most of complex networks share similar properties:

density	low
connexity	giant component
distances	low
degrees	heterogeneous
clustering	high
community	with

Usual properties – reminder

- Graph $G = (V, E)$, $n = |V|$ and $m = |E|$
- Neighbors and degree of v : $N(v)$ and $d(v)$
- Density : $\delta = \frac{2m}{n(n-1)}$
- clustering coefficient: $\mathtt{cc}(G) = \frac{\sum_v \frac{\Delta(v)}{\vee(v)}}{n}$
- transitivity ratio: $\mathtt{tr}(G) = \frac{3\Delta(G)}{\vee(G)}$

Principle (local density)

If a node u is connected to two other distinct nodes v_1 and v_2 then high probability that v_1 and v_2 are connected.

Principle (degree distribution)

The probability that a link between two nodes exists is proportional to the degree of the nodes.

Usual properties – reminder

- Graph $G = (V, E)$, $n = |V|$ and $m = |E|$
- Neighbors and degree of v : $N(v)$ and $d(v)$
- Density : $\delta = \frac{2m}{n(n-1)}$
- clustering coefficient: $\mathtt{cc}(G) = \frac{\sum_v \frac{\Delta(v)}{\vee(v)}}{n}$
- transitivity ratio: $\mathtt{tr}(G) = \frac{3\Delta(G)}{\vee(G)}$

Principle (local density)

If a node u is connected to two other distinct nodes v_1 and v_2 then high probability that v_1 and v_2 are connected.

Principle (degree distribution)

The probability that a link between two nodes exists is proportional to the degree of the nodes.

Outline

Different random approaches

Strategy (RANDOM$_k$)

Test random pairs of nodes (u, v)

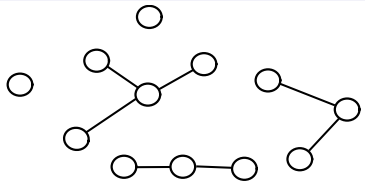
Using *local density* we can improve the previous strategy...

Different random approaches

Using *local density* we can improve the previous strategy...

Strategy (V-RANDOM$_k$)

1: *Choose randomly* u *and* $v \in V$
2: *Test the existence of edge* (u, v)
3: ***if*** (u, v) *exists* ***then***
4: *Test untested pairs* (v, w) *for* w *in* $N(u)$
5: *Test untested pairs* (u, w) *for* w *in* $N(v)$
6: ***end if***

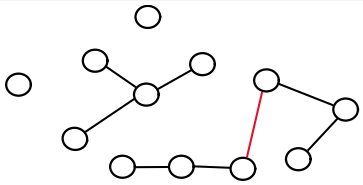


Different random approaches

Using *local density* we can improve the previous strategy...

Strategy (V-RANDOM$_k$)

1: *Choose randomly* u *and* $v \in V$
2: *Test the existence of edge* (u, v)
3: ***if*** (u, v) *exists* ***then***
4: *Test untested pairs* (v, w) *for* w *in* $N(u)$
5: *Test untested pairs* (u, w) *for* w *in* $N(v)$
6: ***end if***

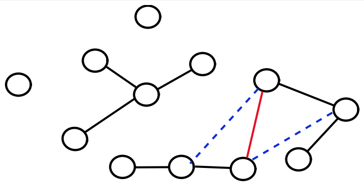


Different random approaches

Using *local density* we can improve the previous strategy...

Strategy (V-RANDOM$_k$)

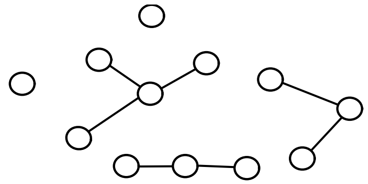
Choose randomly u and v $\in V$
Test the existence of edge (u, v)
if (u, v) *exists* ***then***
  Test untested pairs (v, w) *for w in* $N(u)$
  Test untested pairs (u, w) *for w in* $N(v)$
end if



Complete strategy

Strategy (COMPLETE$_k$)

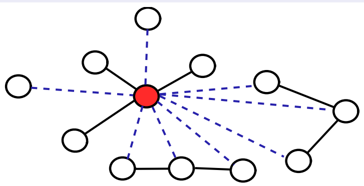
Apply RANDOM$_k$ *(or* V-RANDOM$_k$*)*
Let $X = \{v \in V \text{ s.t. } d(v) > \alpha\}$
while *X is nonempty* ***do***
  Let u in X with $d(u)$ *maximal*
  Remove u from X
  Test all untested pairs (u, v) *for any* $v \in V$
  if (u, v) *exists and is the first link of v discovered and* $d(v) > \alpha\}$ ***then***
    Add v to X
  end if
end while



Complete strategy

Strategy (COMPLETE$_k$)

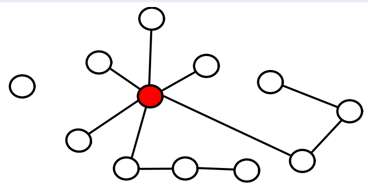
Apply RANDOM$_k$ *(or* V-RANDOM$_k$*)*
Let $X = \{v \in V \text{ s.t. } d(v) > \alpha\}$
while *X is nonempty* ***do***
  Let u in X with $d(u)$ *maximal*
  Remove u from X
  Test all untested pairs (u, v) *for any* $v \in V$
  if (u, v) *exists and is the first link of v discovered and* $d(v) > \alpha\}$ ***then***
    Add v to X
  end if
end while



Complete strategy

Strategy (COMPLETE$_k$)

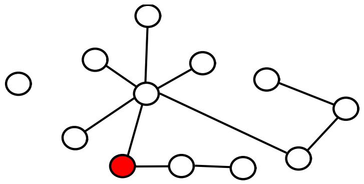
Apply RANDOM$_k$ *(or* V-RANDOM$_k$*)*
Let $X = \{v \in V \text{ s.t. } d(v) > \alpha\}$
while *X is nonempty* ***do***
  Let u in X with $d(u)$ *maximal*
  Remove u from X
  Test all untested pairs (u, v) *for any* $v \in V$
  if (u, v) *exists and is the first link of v discovered and* $d(v) > \alpha\}$ ***then***
    Add v to X
  end if
end while



Complete strategy

Strategy (COMPLETE$_k$)

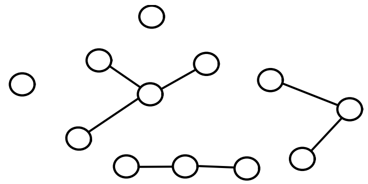
Apply RANDOM$_k$ (or V-RANDOM$_k$)
Let $X = \{v \in V \;\; s.t. \; d(v) > \alpha\}$
while X is nonempty **do**
  Let u in X with $d(u)$ maximal
  Remove u from X
  Test all untested pairs (u, v) for any $v \in V$
  if (u, v) exists and is the first link of v discovered and $d(v) > \alpha\}$ **then**
    Add v to X
  end if
end while



Test-Between-Found strategy

Strategy (TBF$_k$)

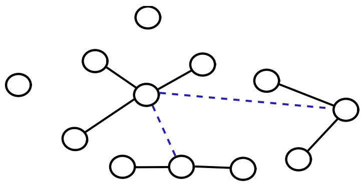
Apply RANDOM$_k$ (or V-RANDOM$_k$)
for $(u, v) \in V \times V$ in decreasing order of $d(u) + d(v)$ **do**
  Test (u, v) if it was untested
end for



Test-Between-Found strategy

Strategy (TBF$_k$)

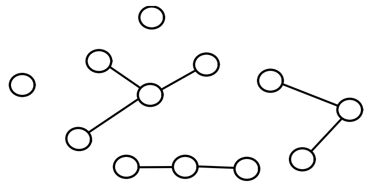
Apply RANDOM$_k$ (or V-RANDOM$_k$)
for $(u, v) \in V \times V$ in decreasing order of $d(u) + d(v)$ **do**
  Test (u, v) if it was untested
end for



Test-Between-Found strategy

Strategy (TBF$_k$)

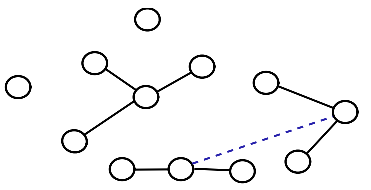
Apply RANDOM$_k$ (or V-RANDOM$_k$)
for $(u, v) \in V \times V$ in decreasing order of $d(u) + d(v)$ **do**
  Test (u, v) if it was untested
end for



Test-Between-Found strategy

Strategy (TBF_k)

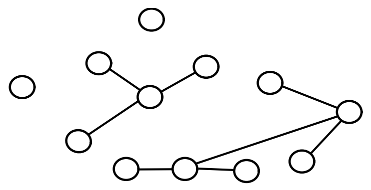
1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: ***for*** $(u, v) \in V \times V$ *in decreasing order of* $d(u) + d(v)$ ***do***
3: *Test* (u, v) *if it was untested*
4: ***end for***



Test-Between-Found strategy

Strategy (TBF_k)

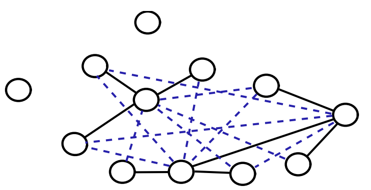
1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: ***for*** $(u, v) \in V \times V$ *in decreasing order of* $d(u) + d(v)$ ***do***
3: *Test* (u, v) *if it was untested*
4: ***end for***



Test-Between-Found strategy

Strategy (TBF_k)

1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: ***for*** $(u, v) \in V \times V$ *in decreasing order of* $d(u) + d(v)$ ***do***
3: *Test* (u, v) *if it was untested*
4: ***end for***



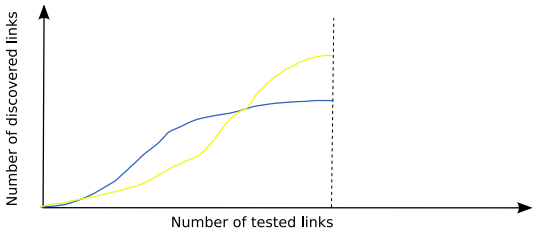
Mixing the approaches

Strategy ($TBFC_k$)

1: *Apply* TBF_k *(or* V-TBF$_k$*)*
2: *Apply* COMPLETE$_0$

Outline

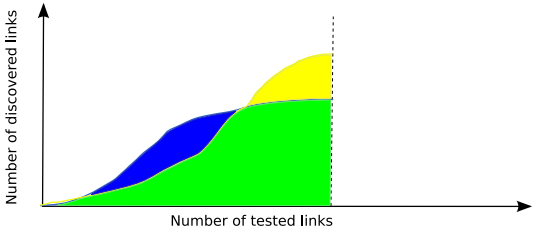
Absolute and relative efficiency



How to compare those strategies?

- Efficiency: $\mathcal{E}_q(S) = \sum_{i=1}^{q} m'_S(i)$
- Normalised efficiency: $\overline{\mathcal{E}}_q(S) = \frac{\mathcal{E}_q(S) - \mathcal{E}_q(\min)}{\mathcal{E}_q(\max) - \mathcal{E}_q(\min)}$
- Relative efficiency: $\mathcal{R}_q(S) = \frac{\overline{\mathcal{E}}_q(S)}{\overline{\mathcal{E}}_q(\text{ran})}$

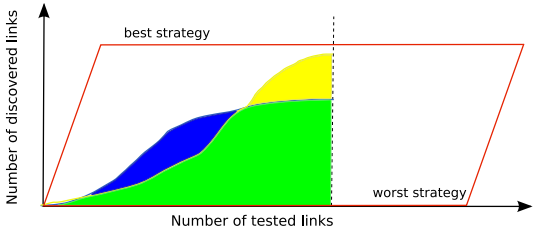
Absolute and relative efficiency



How to compare those strategies?

- Efficiency: $\mathcal{E}_q(S) = \sum_{i=1}^{q} m'_S(i)$
- Normalised efficiency: $\overline{\mathcal{E}}_q(S) = \frac{\mathcal{E}_q(S) - \mathcal{E}_q(\min)}{\mathcal{E}_q(\max) - \mathcal{E}_q(\min)}$
- Relative efficiency: $\mathcal{R}_q(S) = \frac{\overline{\mathcal{E}}_q(S)}{\overline{\mathcal{E}}_q(\text{ran})}$

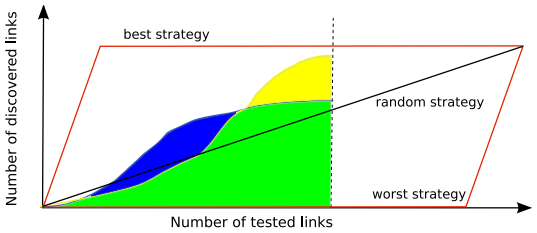
Absolute and relative efficiency



How to compare those strategies?

- Efficiency: $\mathcal{E}_q(S) = \sum_{i=1}^{q} m'_S(i)$
- Normalised efficiency: $\overline{\mathcal{E}}_q(S) = \frac{\mathcal{E}_q(S) - \mathcal{E}_q(\min)}{\mathcal{E}_q(\max) - \mathcal{E}_q(\min)}$
- Relative efficiency: $\mathcal{R}_q(S) = \frac{\overline{\mathcal{E}}_q(S)}{\overline{\mathcal{E}}_q(\text{ran})}$

Absolute and relative efficiency



How to compare those strategies?

- Efficiency: $\mathcal{E}_q(S) = \sum_{i=1}^{q} m'_S(i)$
- Normalised efficiency: $\overline{\mathcal{E}}_q(S) = \frac{\mathcal{E}_q(S) - \mathcal{E}_q(\min)}{\mathcal{E}_q(\max) - \mathcal{E}_q(\min)}$
- Relative efficiency: $\mathcal{R}_q(S) = \frac{\overline{\mathcal{E}}_q(S)}{\overline{\mathcal{E}}_q(\text{ran})}$

Based on Flickr website

Data used for the tests :

- Largest group (31 523 users) of Flickr (August 2006)
- Each user has a list of contacts
- Each user can post a comment on a photo

Allow to generate different graphs depending on:

- whether we use the contact list or the list of comments
- whether we ask for a symmetric interaction or not

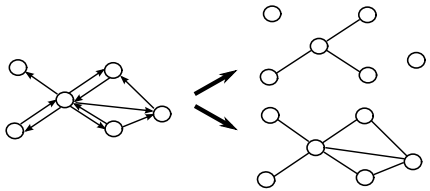
Based on Flickr website

Data used for the tests :

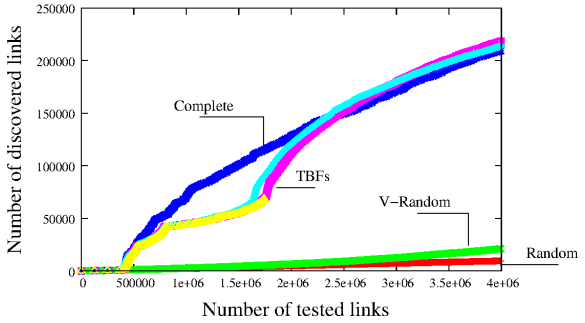
- Largest group (31 523 users) of Flickr (August 2006)
- Each user has a list of contacts
- Each user can post a comment on a photo

Allow to generate different graphs depending on:

- whether we use the contact list or the list of comments
- whether we ask for a symmetric interaction or not



First results



Evolution of the number of discovered links according to the number of tested links for RANDOM$_k$, V-RANDOM$_k$, COMPLETE$_k$, TBFC$_k$ and V-TBFC$_k$ (for $k = 1\,000$ and $q = 4.10^6$ tests)

Strategies efficiency

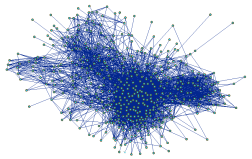
	m'	% tested	% found	$\mathcal{E}$	$\mathcal{R}$
RANDOM	9 609	1.04	1.03	0.006	0.99
V-RANDOM	21 030	1.04	2.25	0.010	1.64
C_{1000}	209 485	1.04	22.4	0.142	24.2
TBF_{1000}	68 874	0.46	7.36	0.048	15.6
$TBFC_{1000}$	218 448	1.04	23.4	0.131	22.3
V-$TBFC_{1000}$	214 175	1.04	22.9	0.134	22.7

Efficiency of each strategy after 4.10^6 links queries:

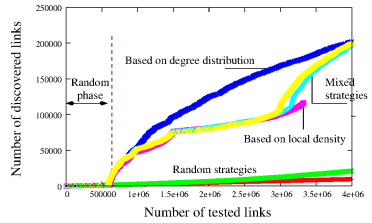
- number m' of discovered links
- percentage of tested pairs of nodes
- percentage of existing links found
- efficiency coefficients $\mathcal{E}$ and $\mathcal{R}$

Summary (1)

Extract a sample **efficiently**

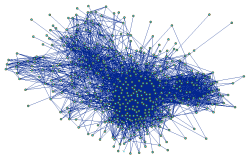


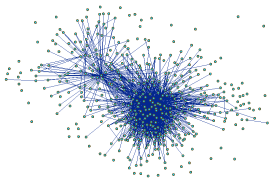
1. Random phase
2. Statistical properties
3. Prediction of existing links



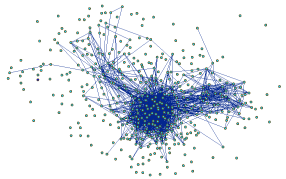
Summary (2)

Extract a **representative** sample **efficiently**





Strategy based on degree distribution



Strategy based on local density

Qualitative assessment

	m'	δ	avg deg	max deg	cc	tr
Reference	21298	0.002	35.5	1708	0.083	0.124
RANDOM	6307	0.000	2.1	38	0.001	0.001
V-RANDOM	6248	0.001	3.1	123	0.133	0.120
C_{1500}	9840	0.001	13.0	1708	0.061	0.422
TBF_{1500}	2289	0.024	54.5	663	0.175	0.208
$TBFC_{1500}$	7717	0.003	20.0	1708	0.085	0.371
V-$TBFC_{1500}$	8789	0.002	17.7	1708	0.072	0.388

Main statistical properties for each extracted samples:

- number m' of links finally discovered
- density (δ)
- average degree, maximal degree
- clustering coefficient (cc), transitivity ratio (tr)

Outline

Going further

- Incorporate qualitative aspects when assessing efficiency
- Elaborate more complex strategies
- Explore different contexts (Internet measurements, web pages explorations) which induce different primitives
- **Proving** good properties of the strategies
- When most of the structure is known, discovering missing links is described as a **classification problem**

Going further: Ordering the link queries

Strategy (COMPLETE$_k$)

Apply RANDOM$_k$ *(or* V-RANDOM$_k$*)*
Let $X = V$
while X *is nonempty* ***do***
    Let u *in* X *with* $d(u)$ *maximal*
    . . .
end while

with $f(u) = d(u)$

Strategy (TBF$_k$)

Apply RANDOM$_k$ *(or* $\vee$-RANDOM$_k$*)*
for $(u, v) \in V \times V$ *in decreasing order of* $d(u) + d(v)$ ***do***
    Test (u, v) *if it was untested*
end for

with $f(u, v) = d(u) + d(v) \ldots$ or $d(u) \times d(v)$ or $\ldots$

Going further: Ordering the link queries

Strategy (COMPLETE$_k$)

Apply RANDOM$_k$ *(or* V-RANDOM$_k$*)*
Let $X = V$
while X *is nonempty* ***do***
    Let u *in* X *with* $f(u)$ *maximal*
    . . .
end while

with $f(u) = d(u)$

Strategy (TBF$_k$)

Apply RANDOM$_k$ *(or* $\vee$-RANDOM$_k$*)*
for $(u, v) \in V \times V$ *in decreasing order of* $f(u, v)$ ***do***
    Test (u, v) *if it was untested*
end for

with $f(u, v) = d(u) + d(v) \ldots$ or $d(u) \times d(v)$ or $\ldots$

Going further: Ordering the link queries

How to take into account the **negative answers**?

Idea: define a notion of **relative degree** for each node

- $f(u) = \frac{d_1(u)}{d_0(u)+d_1(u)}$
- $f(u) = d_1(u) + \frac{d_0(u)+d_1(u)}{n-1}$
- $f(u) = d_1(u) + \frac{1}{1-\frac{d_0(u)+d_1(u)}{n-1}}$

with d_1 and d_0: links tested which do and do not exist

Going further: Ordering the link queries

How to take into account the **negative answers**?

Idea: define a notion of **relative degree** for each node

- $f(u) = \frac{d_1(u)}{d_0(u)+d_1(u)}$
- $f(u) = d_1(u) + \frac{d_0(u)+d_1(u)}{n-1}$
- $f(u) = d_1(u) + \frac{1}{1-\frac{d_0(u)+d_1(u)}{n-1}}$

with d_1 and d_0: links tested which do and do not exist

Going further: Ordering the link queries

How to take into account the **negative answers**?

Idea: define a notion of **relative degree** for each node

- $f(u) = \frac{d_1(u)}{d_0(u)+d_1(u)}$
- $f(u) = d_1(u) + \frac{d_0(u)+d_1(u)}{n-1}$
- $f(u) = d_1(u) + \frac{1}{1-\frac{d_0(u)+d_1(u)}{n-1}}$

with d_1 and d_0: links tested which do and do not exist

Going further: Ordering the link queries

How to take into account the **negative answers**?

Idea: define a notion of **relative degree** for each node

- $f(u) = \frac{d_1(u)}{d_0(u)+d_1(u)}$
- $f(u) = d_1(u) + \frac{d_0(u)+d_1(u)}{n-1}$
- $f(u) = d_1(u) + \frac{1}{1-\frac{d_0(u)+d_1(u)}{n-1}}$

with d_1 and d_0: links tested which do and do not exist