Networks Structure and Dynamics
9. Efficient measurements using link queries

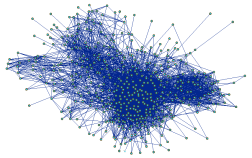
Lionel Tabourier, Fabien Tarissan

LIP6 – CNRS and Université Pierre et Marie Curie

first_name.last_name@lip6.fr

November 9th 2016

The context



Large complex networks appear in :

- Computer science : Internet, Peer-to-peer, Web
- Biology : Gene regulatory networks, Protein-protein interaction networks
- Social science : friendship relations, co-authors networks
- And a lot more : economy, linguistic, ...

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 miss the interactions (links) between them
- One can test the existence of a link between two nodes

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

Efficient measurements using link queries
Introduction

Goal : Measuring the networks

Assuming that :

- All the elements (nodes) are known
- We miss the interactions (links) between them
- One can test the existence of a link between two nodes

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

Proposed method :

1. Short random exploration
2. Computation of relevant statistical properties
3. Predicting the existing links

Common properties – recall

Most of complex networks share similar properties :

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

Common properties – recall

Notations :

- Graph $G = (V, E)$, $n = |V|$ and $m = |E|$
- Neighbors and degree of v : $N(v)$ and $d(v)$
- Density : $\delta = \frac{2m}{n(n-1)}$

Some properties :

- clustering coefficient : $\mathtt{cc}(G) = \frac{\sum_v \frac{\Delta(v)}{\vee(v)}}{n}$
- transitivity ratio : $\mathtt{tr}(G) = \frac{3\Delta(G)}{\vee(G)}$

Common properties – recall

Notations :

- Graph $G = (V, E)$, $n = |V|$ and $m = |E|$
- Neighbors and degree of v : $N(v)$ and $d(v)$
- Density : $\delta = \frac{2m}{n(n-1)}$

Some properties :

- 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 there is a high probability that v_1 and v_2 are also connected.

Principle (degree distribution)

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

Different random approaches

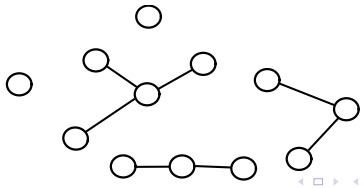
Strategy (RANDOM$_k$)

Test random pairs of nodes (u, v)

Based on the *local density* we can improve the previous strategy :

Strategy (V-RANDOM$_k$)

1: *Choose randomly u and v in N*
2: *Test the existence of the 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

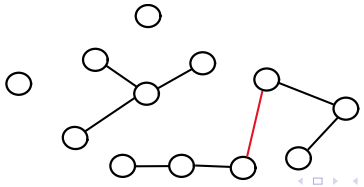
Strategy (RANDOM$_k$)

Test random pairs of nodes (u, v)

Based on the *local density* we can improve the previous strategy :

Strategy (V-RANDOM$_k$)

1: *Choose randomly u and v in N*
2: *Test the existence of the 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

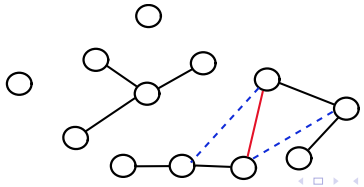
Strategy (RANDOM$_k$)

Test random pairs of nodes (u, v)

Based on the *local density* we can improve the previous strategy :

Strategy (V-RANDOM$_k$)

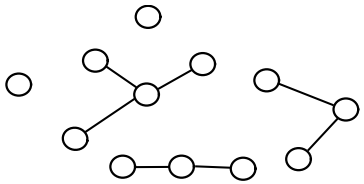
1: *Choose randomly u and v in N*
2: *Test the existence of the edge* (u, v)
3: **if** (u, v) *exists* **then**
4: *Test untested pairs* (v, w) *for w in* $N'(u)$
5: *Test untested pairs* (u, w) *for w in* $N'(v)$
6: **end if**



Complete strategy

Strategy (COMPLETE$_k$)

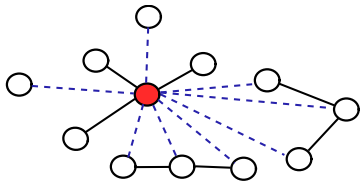
1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: *Let* $X = V'$
3: **while** *X is nonempty* **do**
4: *Let u in X with* $d'(u)$ *maximal*
5: *Remove u from X*
6: *Test all untested pairs* (u, v)*, for any* $v \in V$
7: **if** (u, v) *exists and is the first link of v discovered* **then**
8: *Add v to X*
9: **end if**
10: **end while**



Complete strategy

Strategy (COMPLETE$_k$)

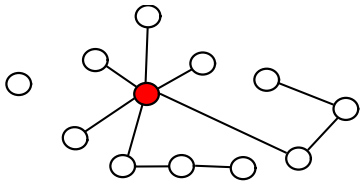
1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: *Let* $X = V'$
3: **while** *X is nonempty* **do**
4: *Let u in X with* $d'(u)$ *maximal*
5: *Remove u from X*
6: *Test all untested pairs* (u, v)*, for any* $v \in V$
7: **if** (u, v) *exists and is the first link of v discovered* **then**
8: *Add v to X*
9: **end if**
10: **end while**



Complete strategy

Strategy (COMPLETE$_k$)

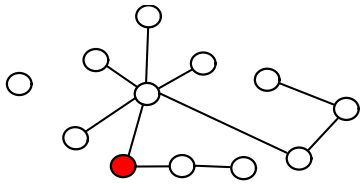
1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: *Let* $X = V'$
3: **while** X *is nonempty* **do**
4: *Let* u *in* X *with* $d'(u)$ *maximal*
5: *Remove* u *from* X
6: *Test all untested pairs* (u, v)*, for any* $v \in V$
7: **if** (u, v) *exists and is the first link of* v *discovered* **then**
8: *Add* v *to* X
9: **end if**
10: **end while**



Complete strategy

Strategy (COMPLETE$_k$)

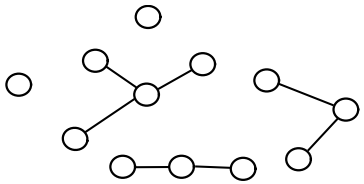
1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: *Let* $X = V'$
3: **while** X *is nonempty* **do**
4: *Let* u *in* X *with* $d'(u)$ *maximal*
5: *Remove* u *from* X
6: *Test all untested pairs* (u, v)*, for any* $v \in V$
7: **if** (u, v) *exists and is the first link of* v *discovered* **then**
8: *Add* v *to* X
9: **end if**
10: **end while**



Test-Between-Found strategy

Strategy (TBF$_k$)

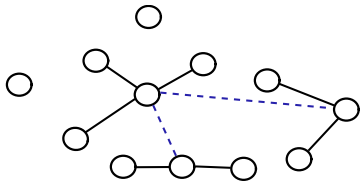
1: *Apply* RANDOM$_k$ *(or* $\vee$-RANDOM$_k$*)*
2: **for** $(u, v) \in V' \times V'$ *in decreasing order of* $d'(u) + d'(v)$ **do**
3: *Test* (u, v) *if it was untested*
4: **end for**



Test-Between-Found strategy

Strategy (TBF$_k$)

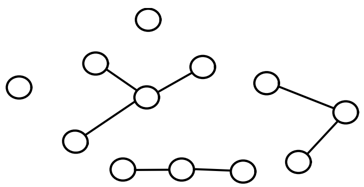
1: *Apply* RANDOM$_k$ *(or* $\vee$-RANDOM$_k$*)*
2: **for** $(u, v) \in V' \times V'$ *in decreasing order of* $d'(u) + d'(v)$ **do**
3: *Test* (u, v) *if it was untested*
4: **end for**



Test-Between-Found strategy

Strategy (TBF_k)

Apply $\textsc{random}_k$ *(or* $\vee$-$\textsc{random}_k$*)*
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

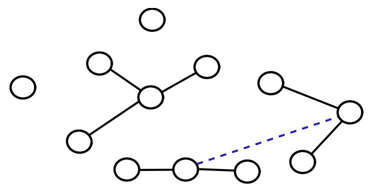


Efficient measurements using link queries
Measurement strategies
Simple strategies

Test-Between-Found strategy

Strategy (TBF_k)

Apply $\textsc{random}_k$ *(or* $\vee$-$\textsc{random}_k$*)*
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

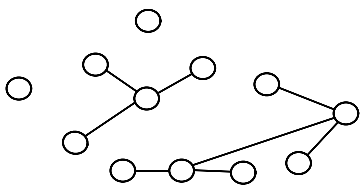


Efficient measurements using link queries
Measurement strategies
Simple strategies

Test-Between-Found strategy

Strategy (TBF_k)

Apply $\textsc{random}_k$ *(or* $\vee$-$\textsc{random}_k$*)*
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

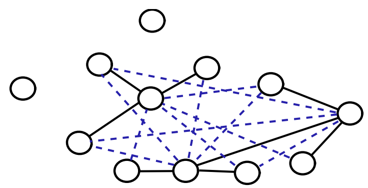


Efficient measurements using link queries
Measurement strategies
Simple strategies

Test-Between-Found strategy

Strategy (TBF_k)

Apply $\textsc{random}_k$ *(or* $\vee$-$\textsc{random}_k$*)*
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

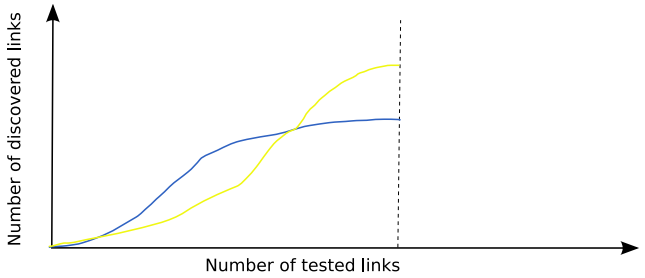


Mixing the approaches

Strategy (TBFC_k)

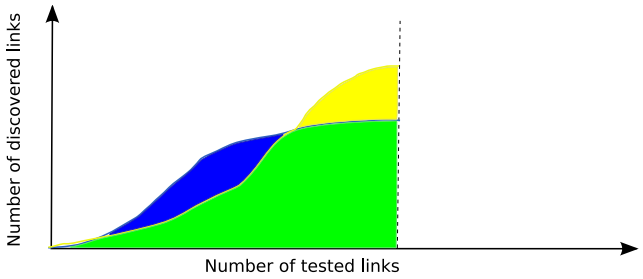
1: *Apply* TBF_k *(or* $\vee$-TBF_k*)*
2: *Apply* $\textsc{Complete}_0$

Absolute and relative efficiency



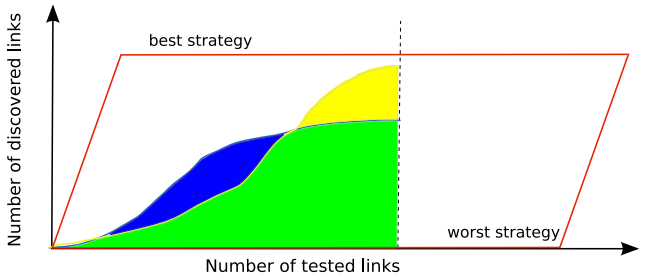
How to compare those two strategies ?

Absolute and relative efficiency



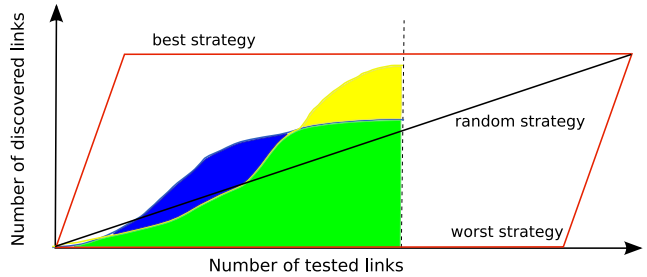
- Efficiency : $\mathcal{E}_q(S) = \sum_{i=1}^{q} m'_S(i)$

Absolute and relative efficiency



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

Absolute and relative efficiency



- 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 : $\mathcal{R}_q(S) = \frac{\overline{\mathcal{E}}_q(S)}{\overline{\mathcal{E}}_q(\text{ran})}$

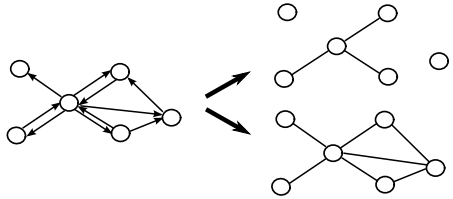
Based on Flickr website

Data used for the tests :

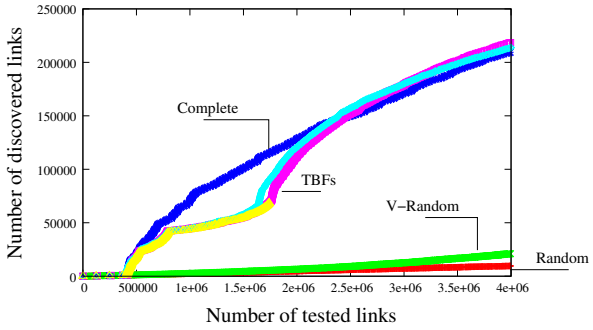
- Largest group (31 523 users) of Flickr (Aug 2006)
- Each user has a list of contacts
- Each user can post a comment on a photo

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



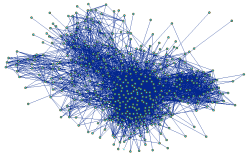
Evolution of the number of discovered links according to the number of tested links for RANDOM_k, V-RANDOM_k, COMPLETE_k, TBFC_k and V-TBFC_k (for $k = 1\,000$ and $q = 4.10^6$ tests)

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
$V\text{-}TBFC_{1000}$	214 175	1.04	22.9	0.134	22.7

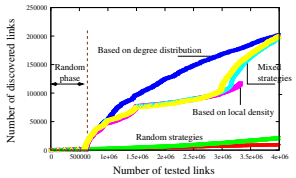
Efficiency of each strategy after 4.10^6 links queries : the number m' of discovered links ; the percentage of tested pairs of nodes ; the percentage of existing links found ; and the efficiency coefficients $\mathcal{E}$ and $\mathcal{R}$.

Summary

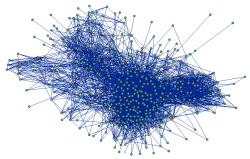


Extract a sample **efficiently**

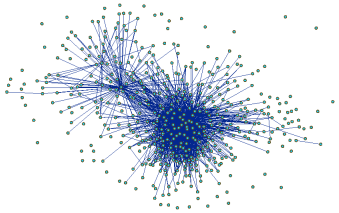
1. Random phase
2. Statistical properties
3. Prediction of existing links



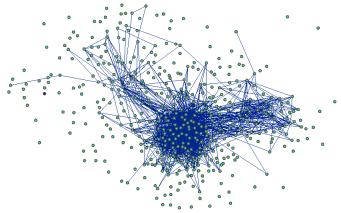
Summary



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 and transitivity ratio.

Going further

- Incorporate qualitative aspects when assessing the efficiency
- Elaborate more complex strategies
- Explore different contexts (Internet measurements, web pages explorations) which induce different primitives
- **Proving** the good properties of the strategies

Going further : Ordering the link queries

Strategy (COMPLETE$_k$)

1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: *Let* $X = V'$
3: **while** X *is nonempty* **do**
4: *Let* u *in* X *with* $d'(u)$ *maximal*
5: ...
6: **end while**

Strategy (TBF$_k$)

1: *Apply* RANDOM$_k$ *(or* $\vee$-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**

Going further : Ordering the link queries

Strategy (COMPLETE$_k$)

1: *Apply* RANDOM$_k$ *(or* V-RANDOM$_k$*)*
2: *Let* $X = V'$
3: **while** X *is nonempty* **do**
4: *Let* u *in* X *with* $f(u)$ *maximal*
5: ...
6: **end while**

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

Strategy (TBF$_k$)

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

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

Going further : Ordering the link queries

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

Going further : Ordering the link queries

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

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

- $f(u) = \frac{d'_1(u)}{d'_0(u)+d'_1(u)}$

Going further : Ordering the link queries

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

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

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

Efficient measurements using link queries
Perspectives

Going further : Ordering the link queries

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

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

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