COSN 2014
CONFERENCE ON ONLINE SOCIAL NETWORKS

SAPIENZA
UNIVERSITÀ DI ROMA

Spreading Rumours without the Network

Alessandro Epasto

P. Brach*, A. Panconesi°, P. Sankowski*.

*U. of Warsaw °Sapienza U. Rome

Rumour Spreading

Diffusive processes on graphs are an important paradigm in several fields:

- **Systems:** How to spread information on network?
- **Social Networks:** Why posts become viral?
- **Sociology:** What makes innovations/products accepted?
- **Epidemiology:** How diseases spread?

We consider various models of information diffusion: **Push, Pull and SIR**.

Background

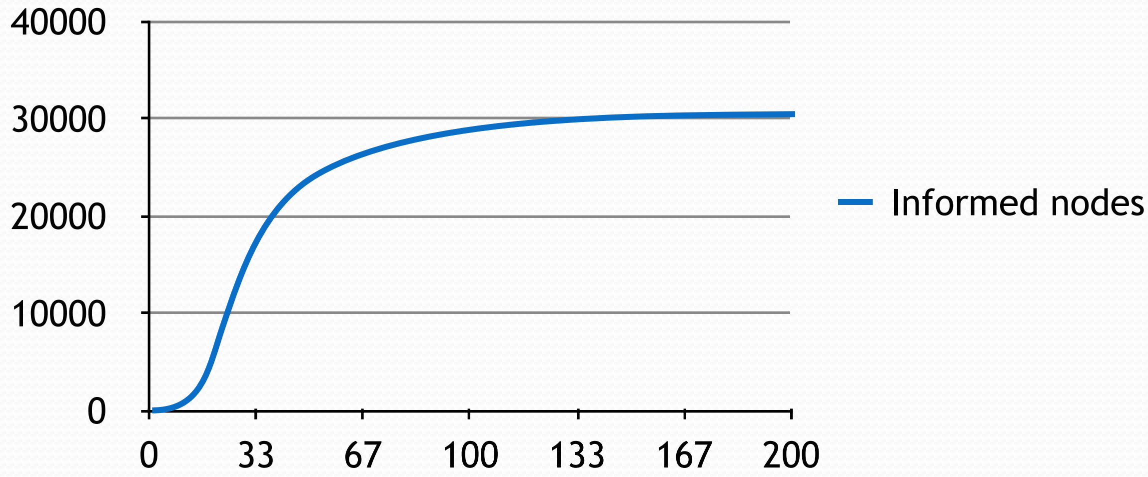
Most results known are asymptotic bounds on the competition time:

- At most O(n log(n)) (Feige et. al, 90)
- Fast in Erdos Reyni and Preferential Attachement (Elsasser et al. 2006, Chierichetti et al. 2009).
- Fast in high conductance graphs. (Chierichetti et al. 2010, Giakkoupis et al. 2011)

Our Goal

Goal #1: Beyond asymptotics

We are interested in the expected number of informed nodes for each time step of the process

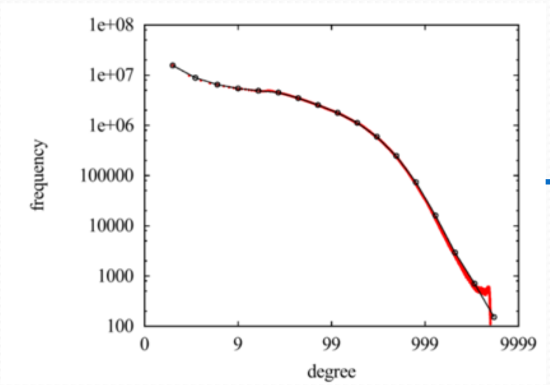


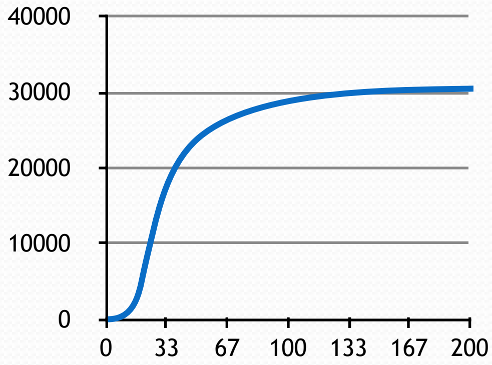
Notice: this is known only for very simple graphs (e.g. Clique, Pittel '87)

Our Goal

Goal #2: Prediction with limited information

Motivation: real networks are often unavailable



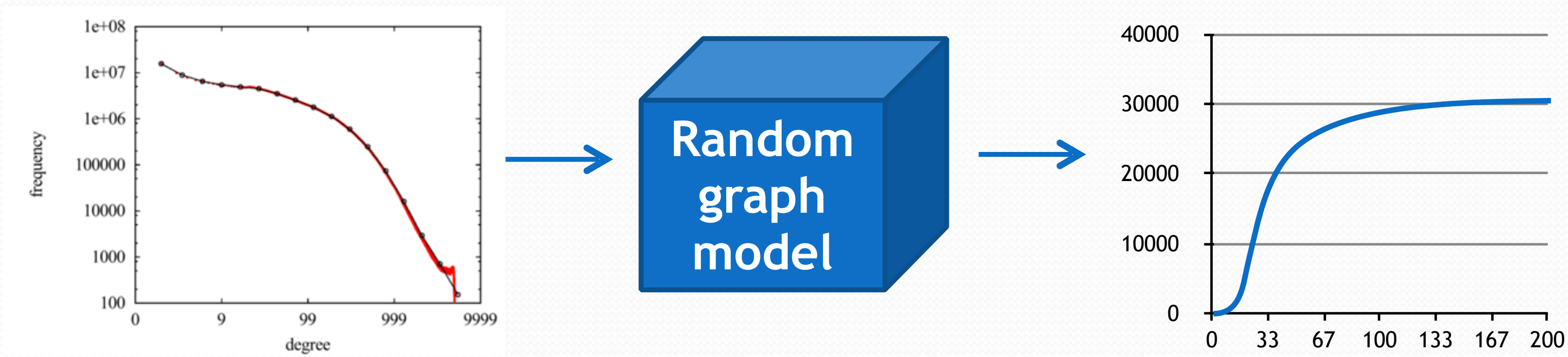


Caveat: this is clearly an ill-posed question…

… But surprisingly, it is possible for **real** social network

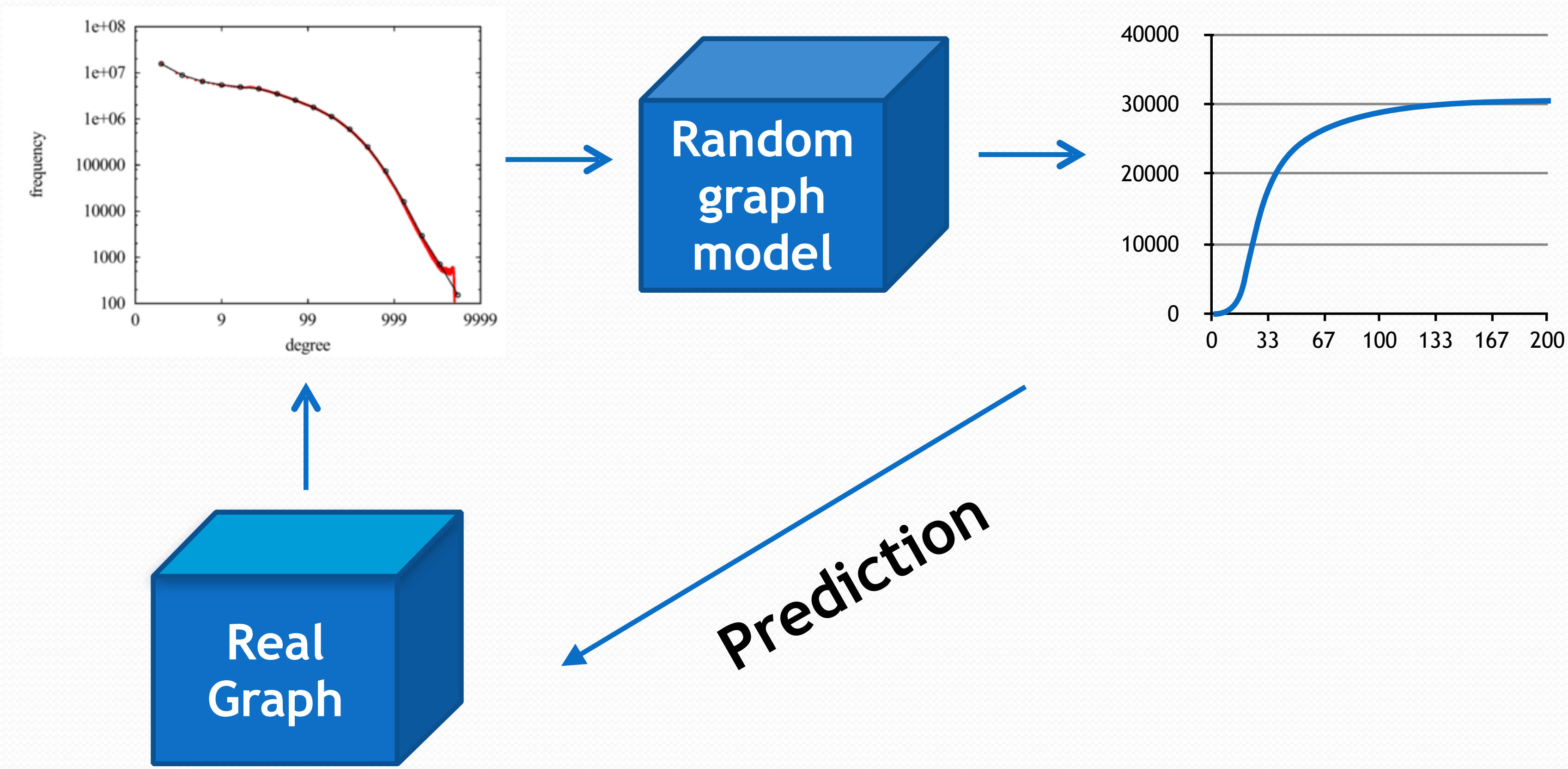
How Can we Achieve this?

A simpler problem: model the *unknown graph* by a *known random* graph generation process.



How Can we Achieve this?

A simpler problem: model the *unknown graph* by a *known random* graph generation process.



Which Graph Model?

We use the **configuration model** as random graph model.

SIR on configuration model matches real post diffusions in **Twitter** (Goel et al., 2013):

- Distribution of **popularity** of posts.
- **Virality** of the diffusion.

Our Contribution

A predictor algorithm for the **configuration model** for the **Push, Pull** and **SIR** Processes:

- Space efficient: very large graphs can fit in memory.
- Provably exact on random graphs.

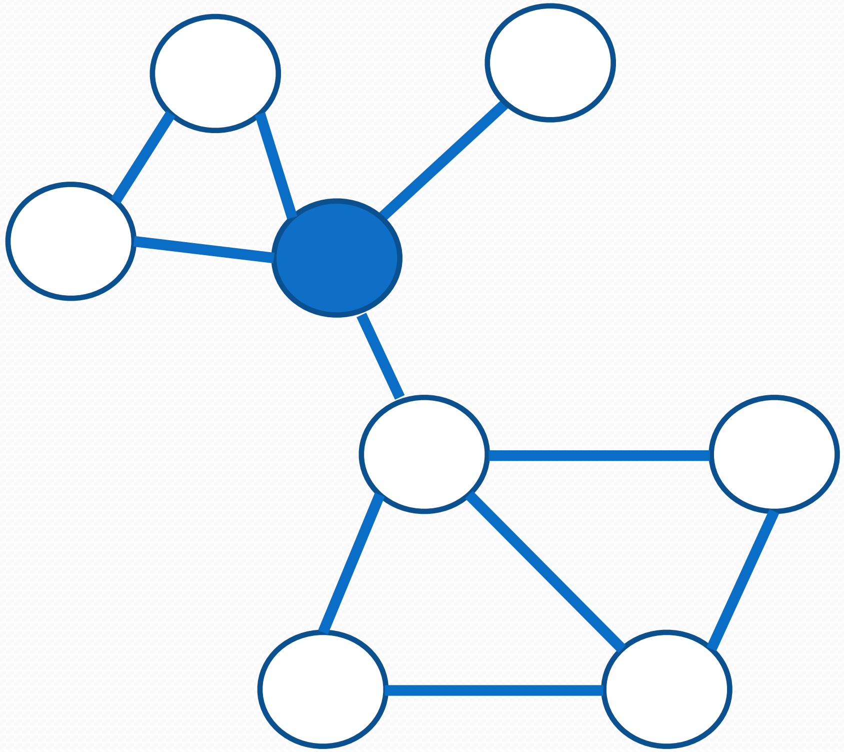
The algorithm predicts accurately the both the popularity and the virality on real social networks.

Outline of the Talk

- The diffusion processes;
- Our algorithm(s);
- Experimental evaluation;
- Conclusions.

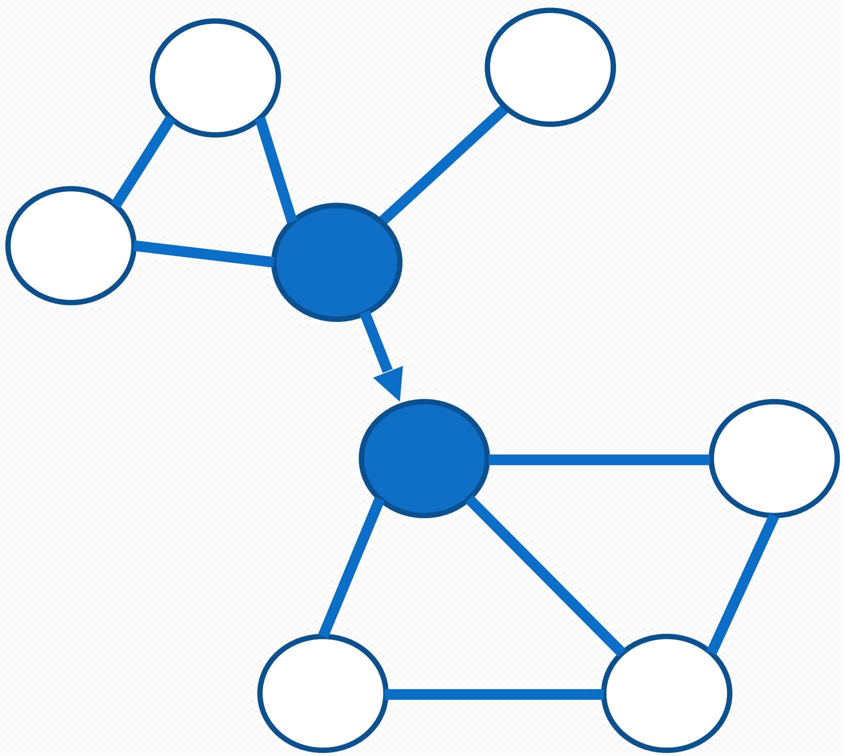
The Push-Pull Process

Push-Pull Protocol



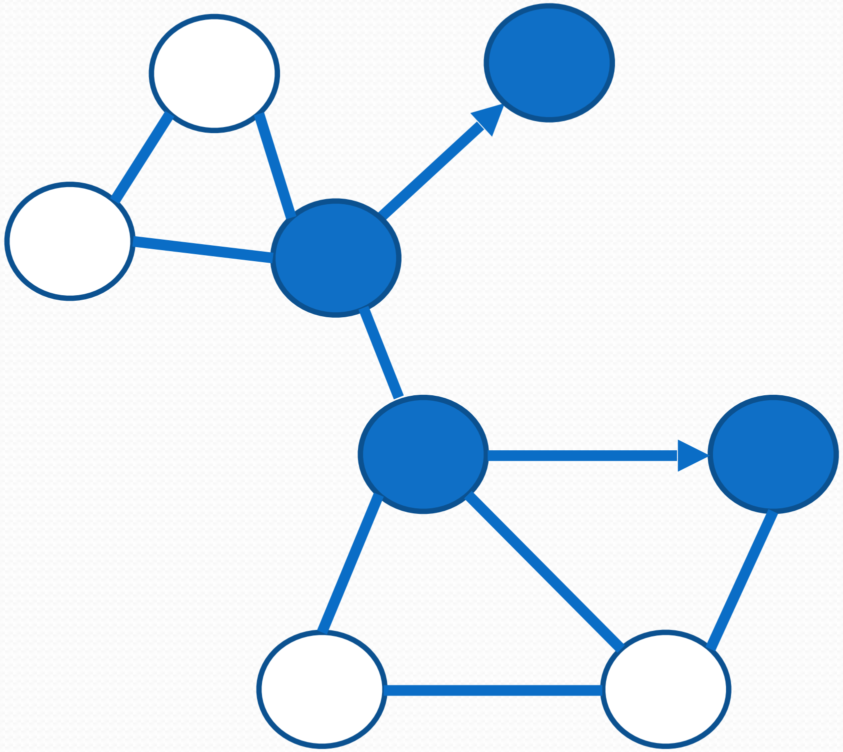
PUSH

Push-Pull Protocol



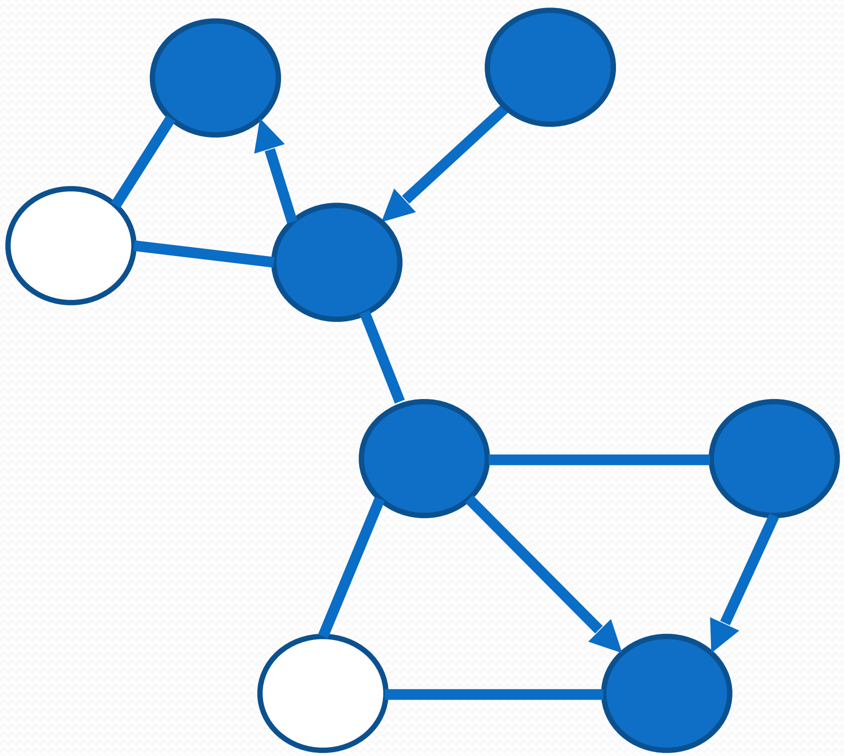
PUSH

Push-Pull Protocol



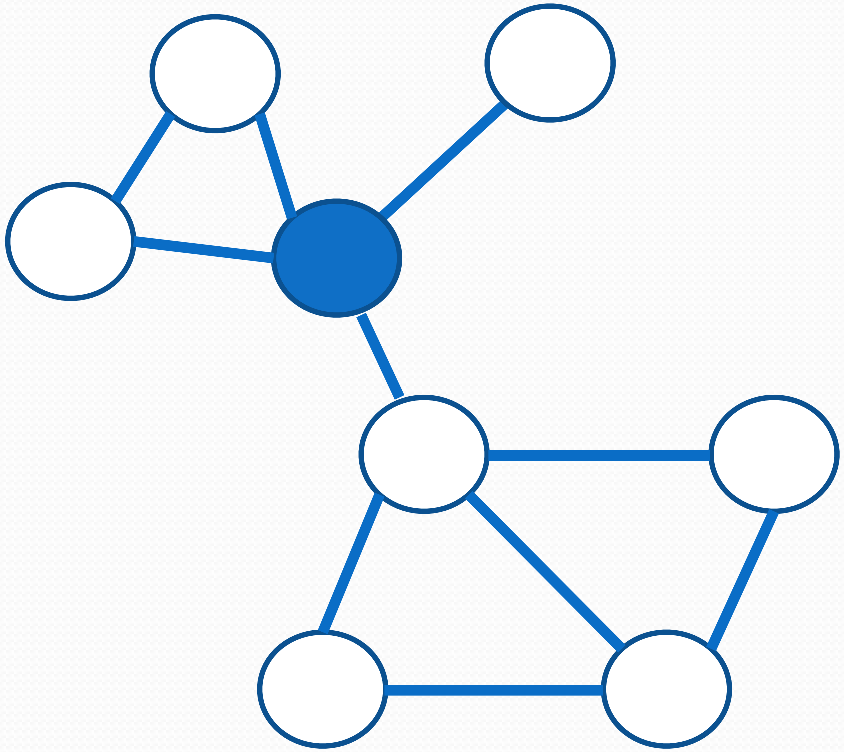
PUSH

Push-Pull Protocol



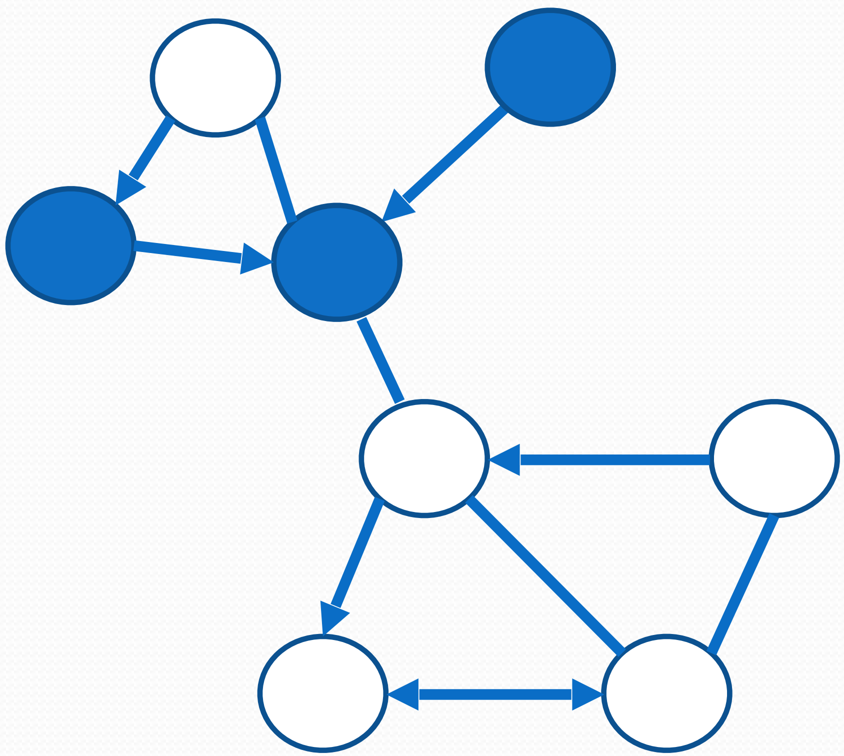
PUSH

Push-Pull Protocol



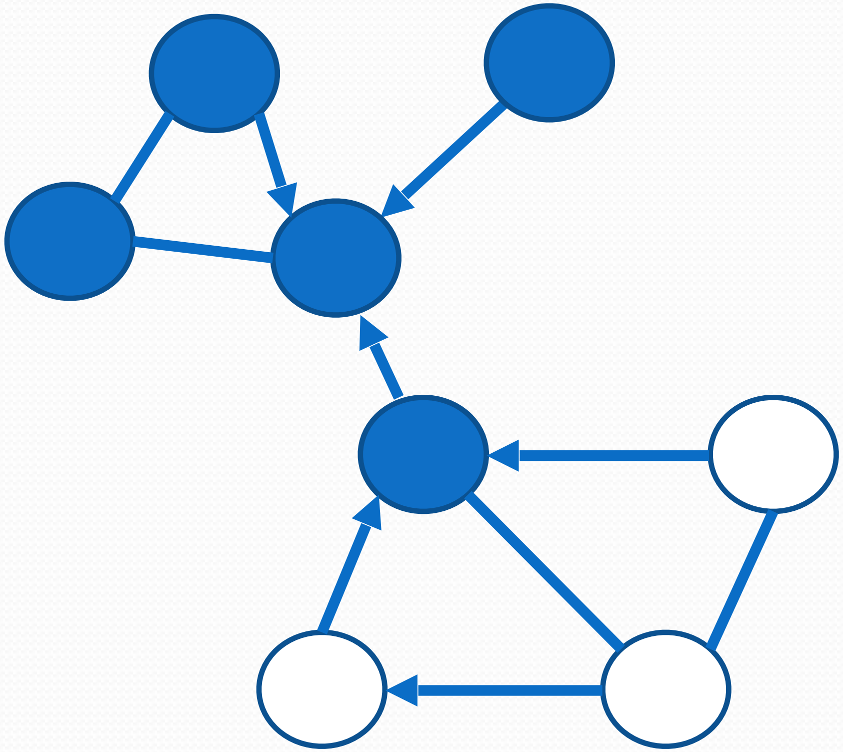
PULL

Push-Pull Protocol



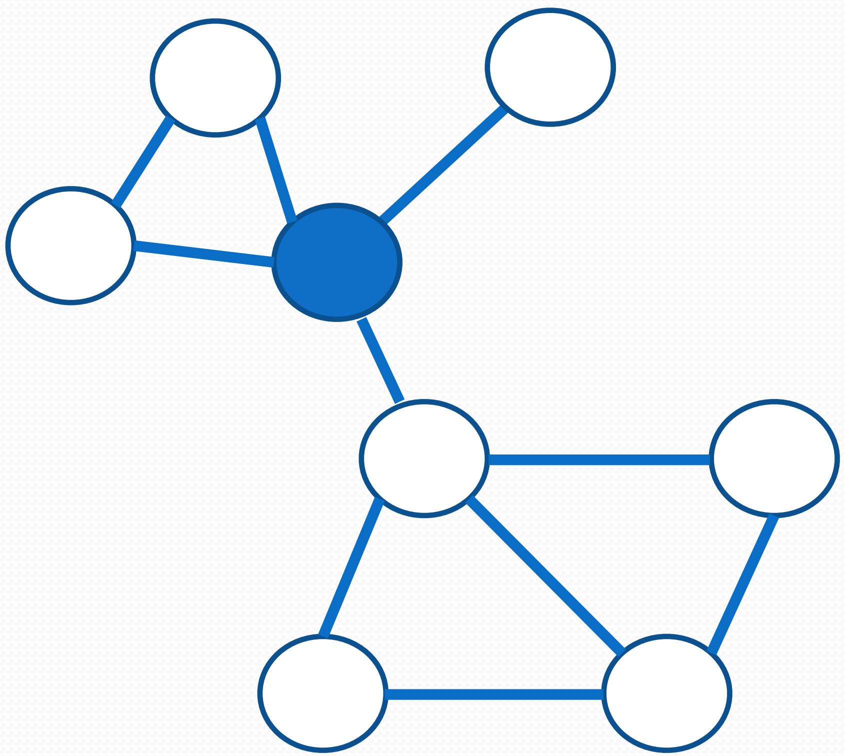
PULL

Push-Pull Protocol



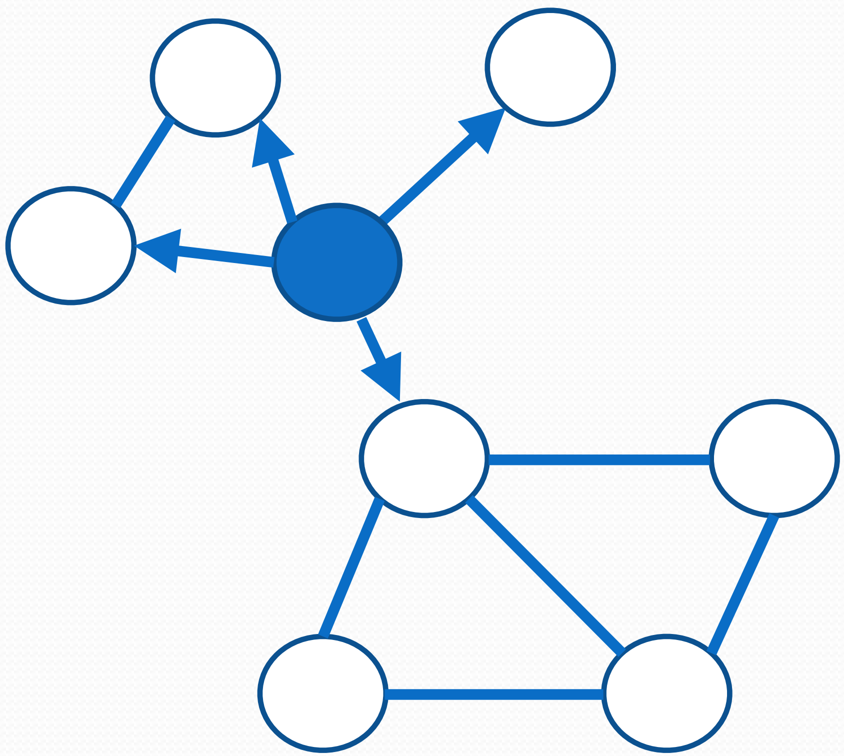
PULL

SIR Process



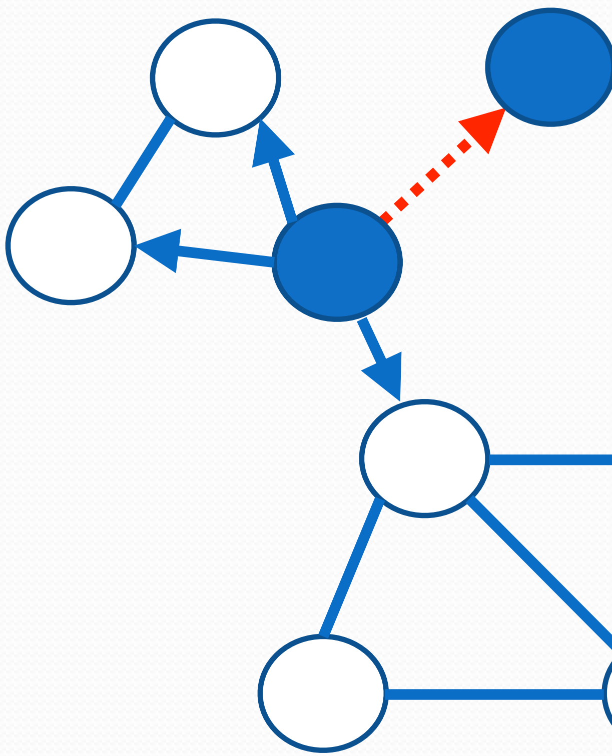
SIR

SIR Process



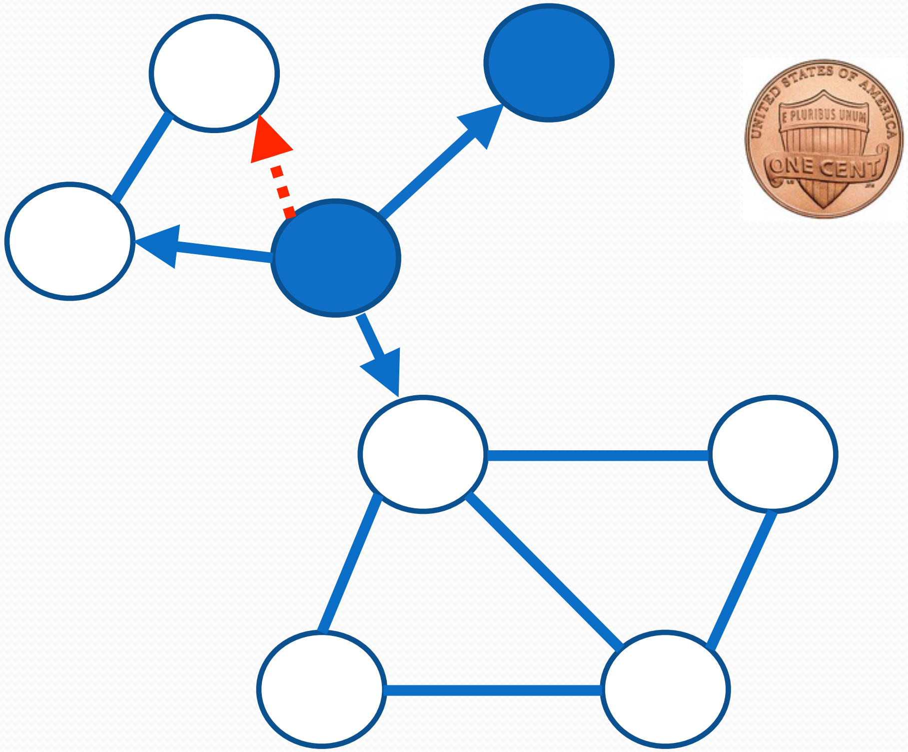
SIR

SIR Process



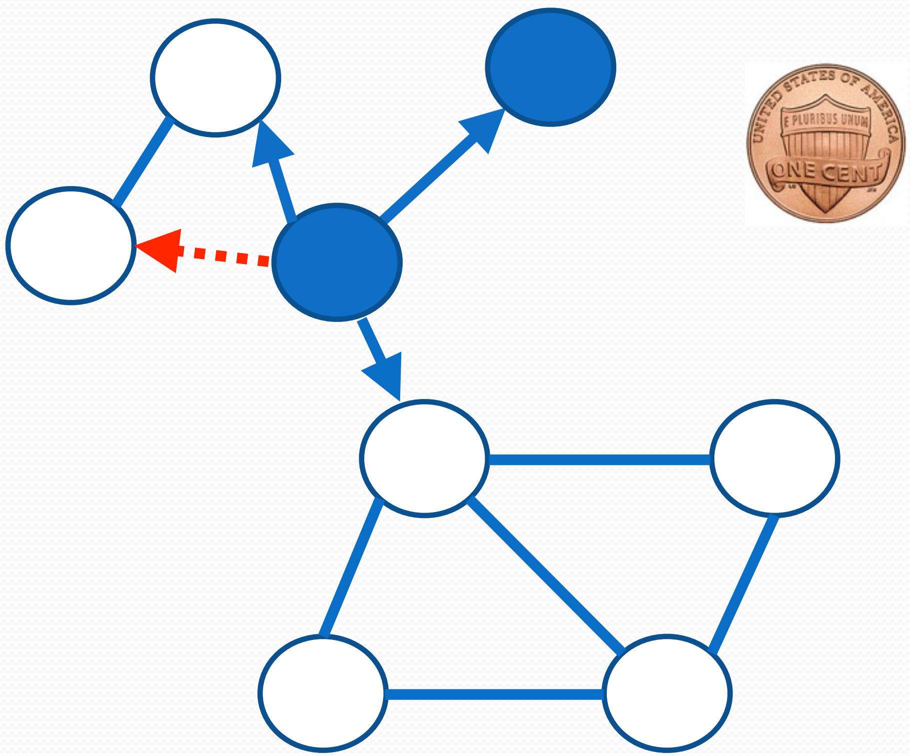
SIR

SIR Process



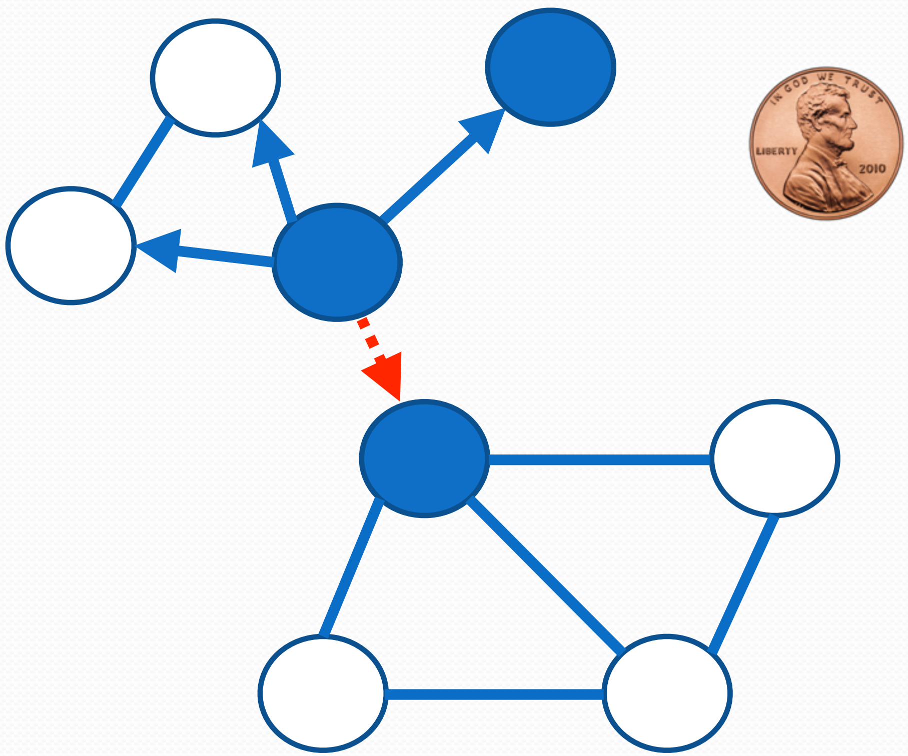
SIR

SIR Process



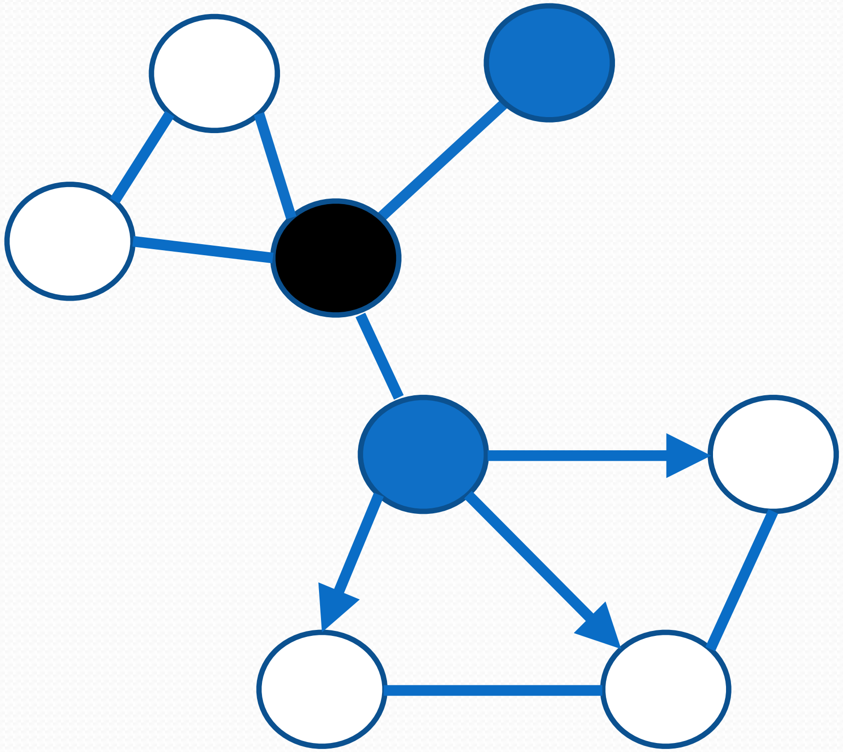
SIR

SIR Process



SIR

SIR Process



SIR

Our Algorithm

Naive Solution

Simulate two random processes: the network generation and the rumour spreading.

Naive algorithm:

- Generate a random network G.
- Simulate rumour spreading on G.
- Run several times **in parallel** and average.

Space bottleneck: Real networks are too large to fit in main memory!

Our Approach

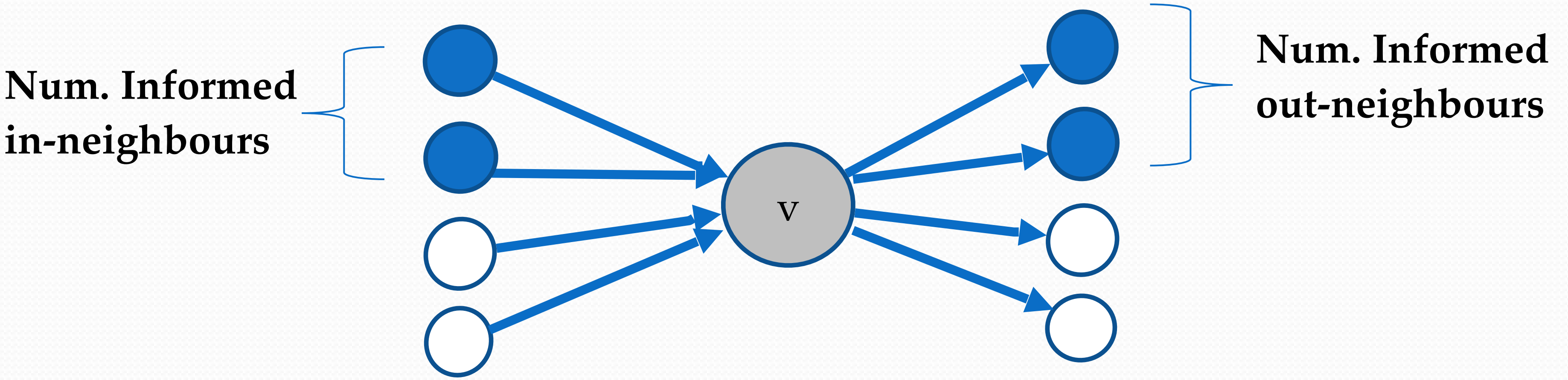
We can reduce the space to O(n) vs O(n+m) in directed graphs and even o(n) in undirected ones.

This is a significant reduction not only in asymptotic!

Deferred decision principle: the topology is *discovered* as nodes are involved in the rumor spreading process and immediately *forget*.

Intuition

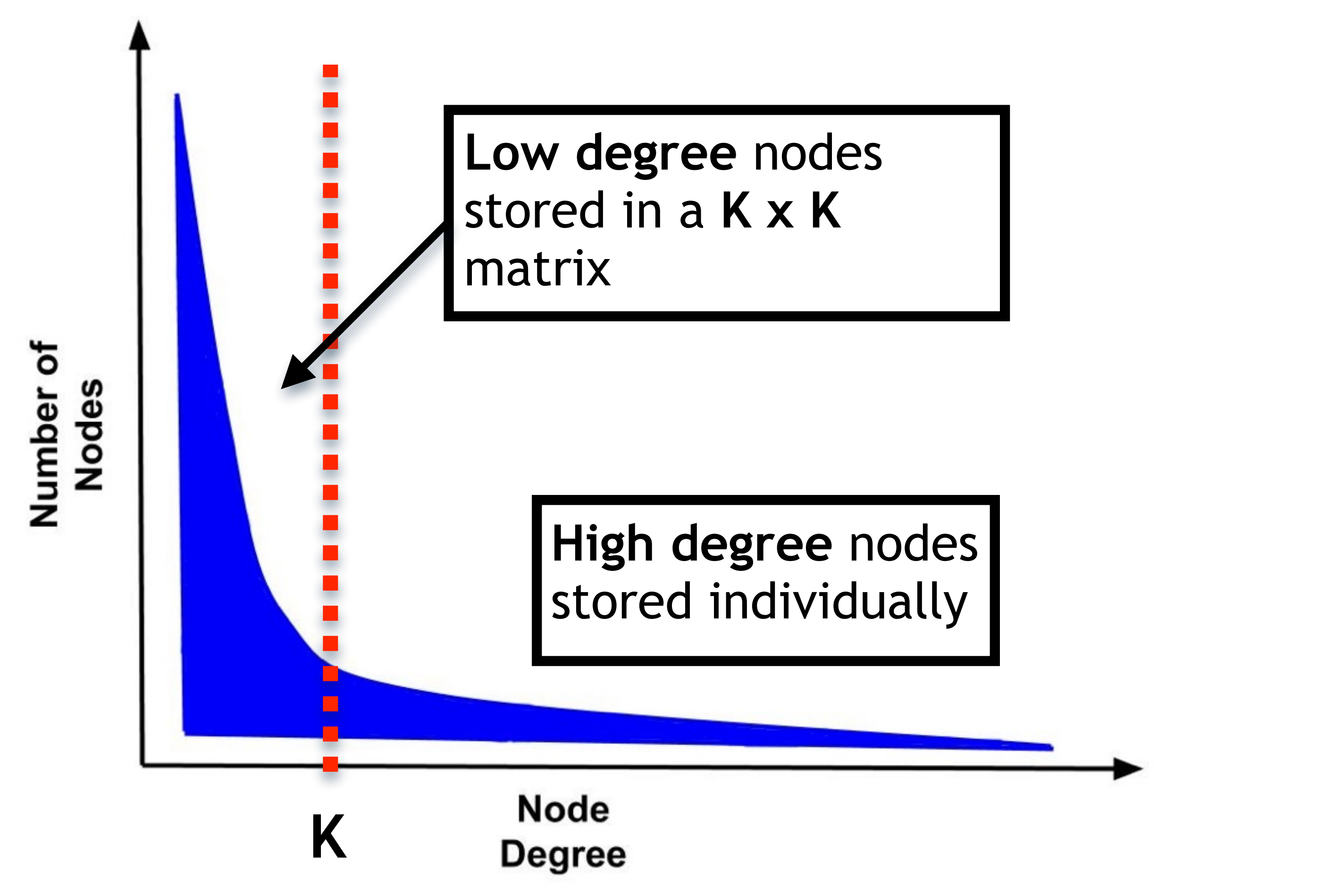
Only the local neighbourhood determines the evolution of the process.



We do not store the **edges** of the graph.

Undirected Graphs

We use an efficient matrix representation.



Undirected Graphs

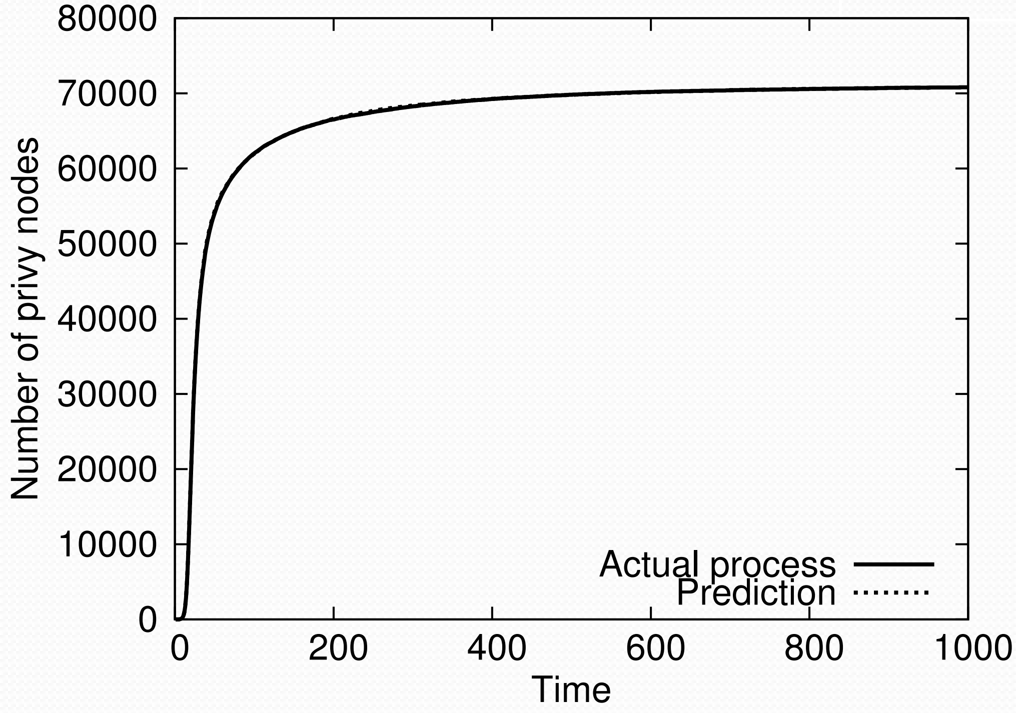
Graph	Nodes	Matrix SIze	Saving in space
Livejournal	*5M*	*176*	*98%*
Facebook (estimates)	*720M*	*<5000*	*>97%*

For power law graphs of exponent α the cost is $n^{\frac{2}{1+\alpha}}$

In practice the entire Facebook graph could fit in few gigabytes.

Results on Random Graphs

Results on Random Graphs

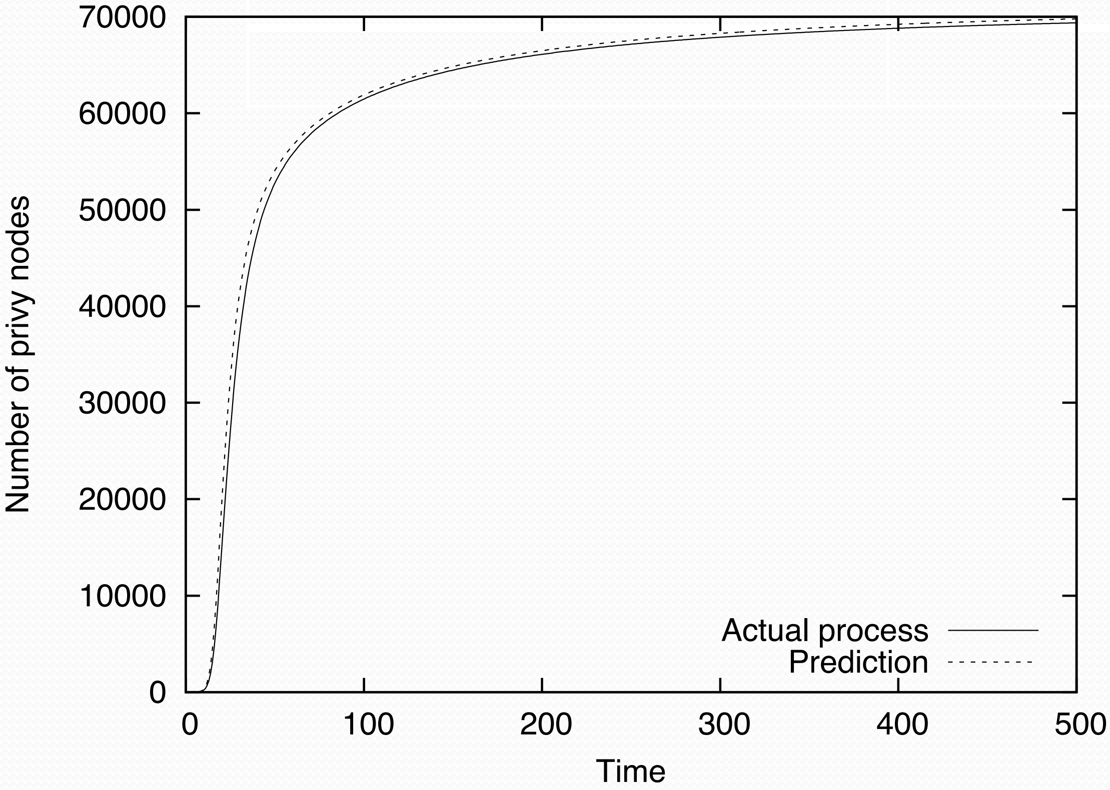


The model prediction is perfect

This can be proved formally.

Results on Real Graphs

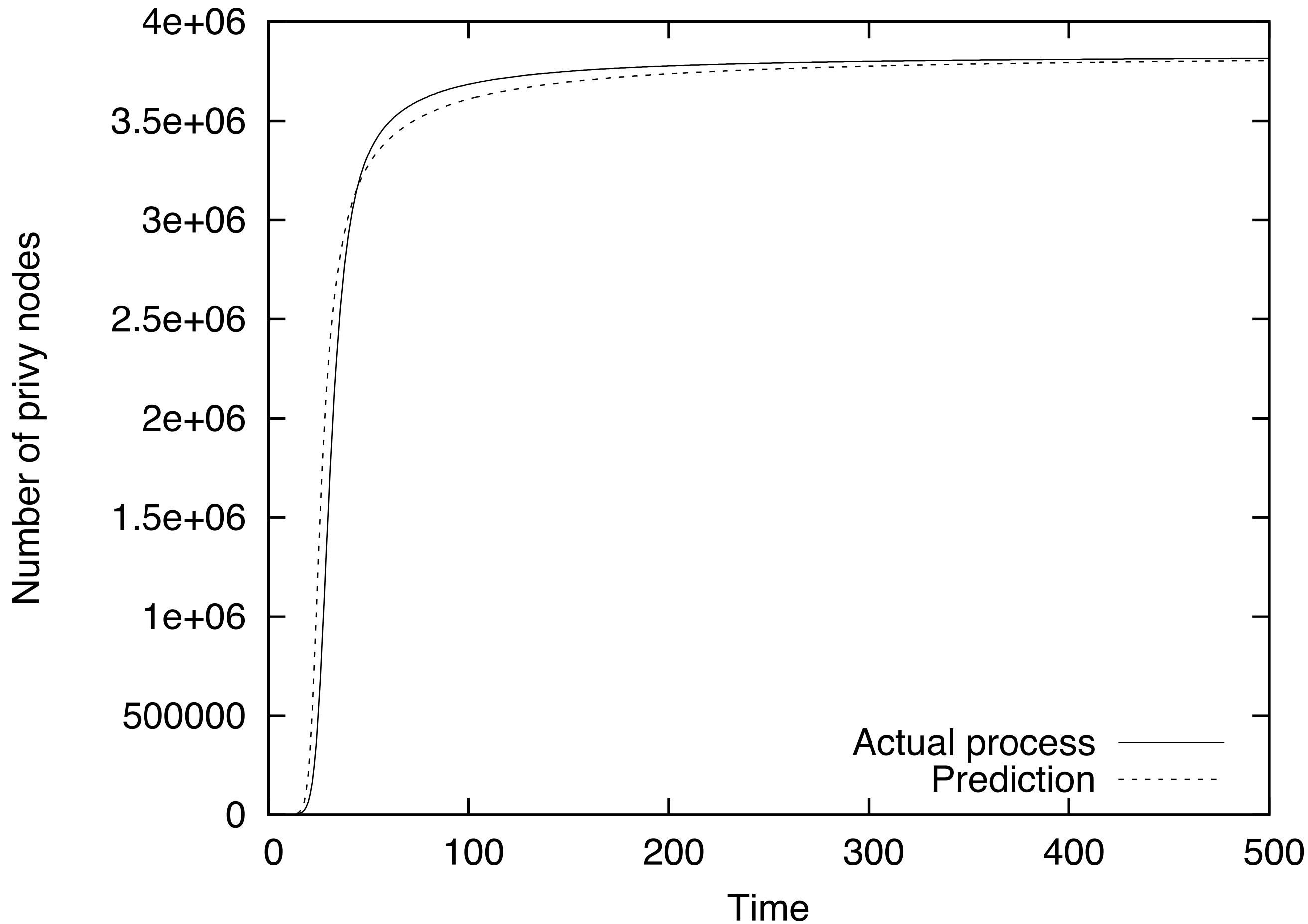
Social Networks - Push



Slashdot

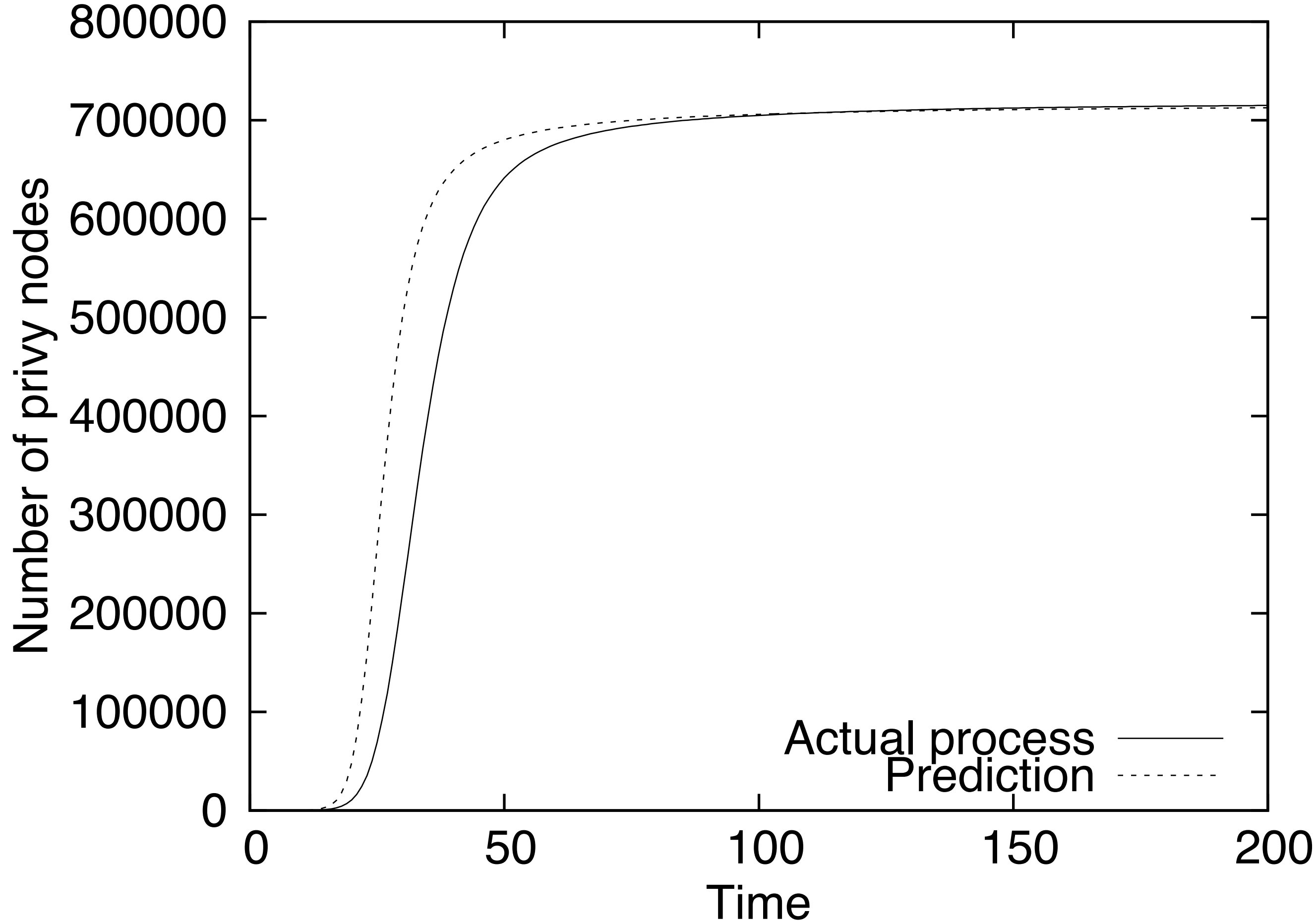
The model is qualitatively accurate for the social network we tested

More Social Networks - Push



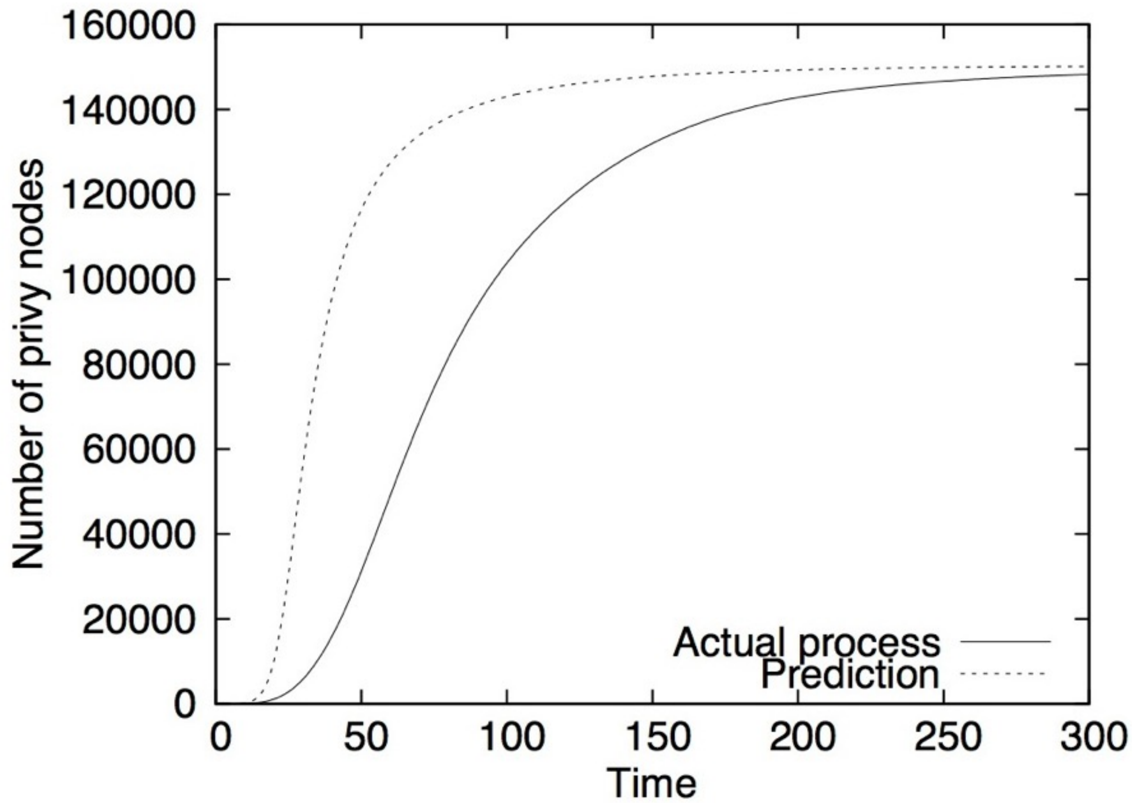
Livejournal

More Social Networks - Push



DBLP

Non-Social Networks - Push



Web Stanford

For non-social networks the prediction is not accurate.

Results

Prediction performances strongly depends on the network class:

- Very good for **social networks**: friendship graphs, trust networks, collaboration networks.
- Poor for **non-social networks**: web graphs, road networks, etc.

This dichotomy has been observed in other contexts: degree correlations, graph compressibility, etc.

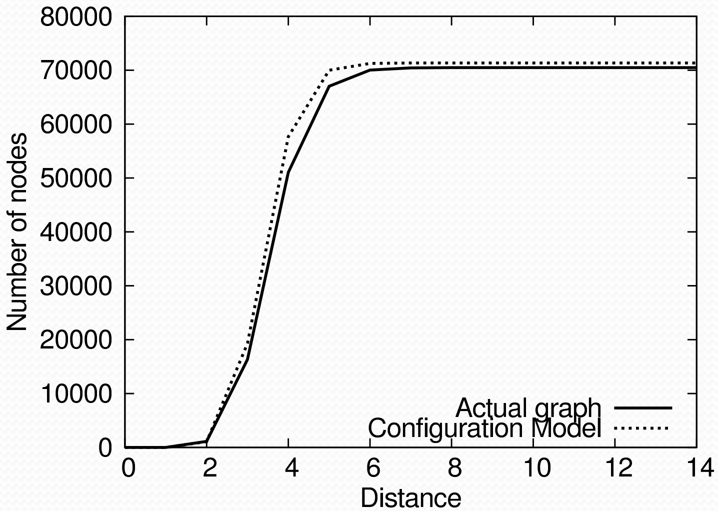
What is the reason for this phenomenon?

Neighbourhood Function

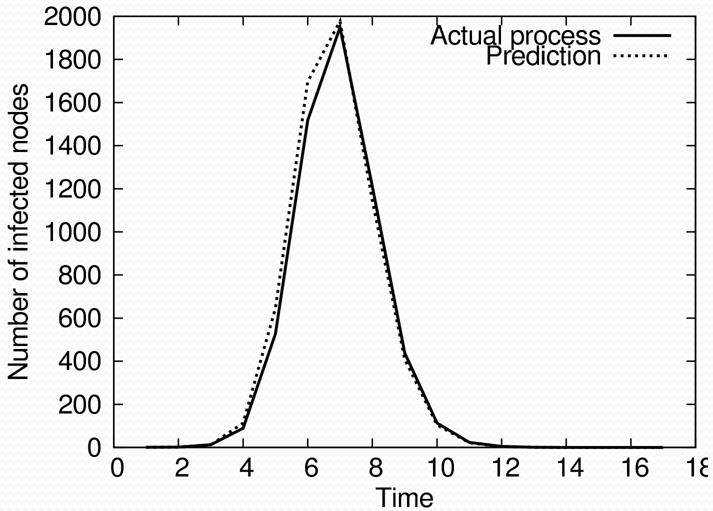
The neighbourhood function F(t) of graph measures how many pairs of nodes are at distance <= t

This measure has been shown to tell apart social and non-social graphs.

Neighbourhood F. vs Prediction Quality



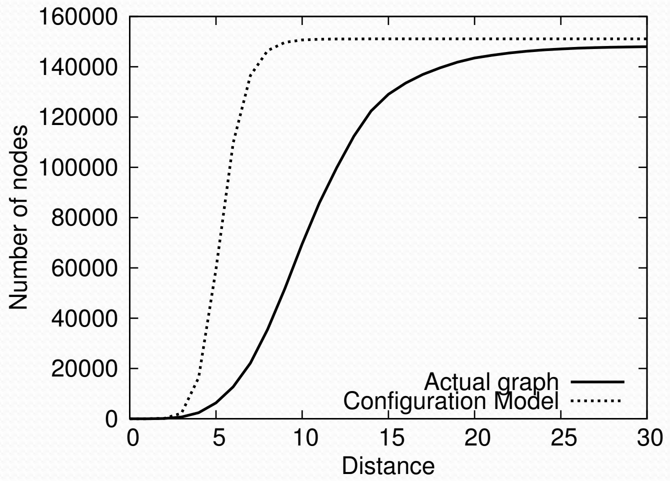
Slashdot Neighbourhood F.



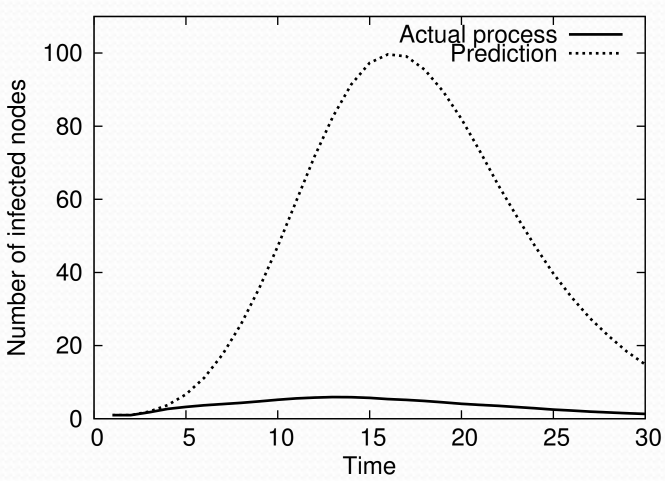
Slashdot Prediction - SIR

Social graphs have a neighbourhood function close to the configuration model.

Neighbourhood F. vs Prediction Quality



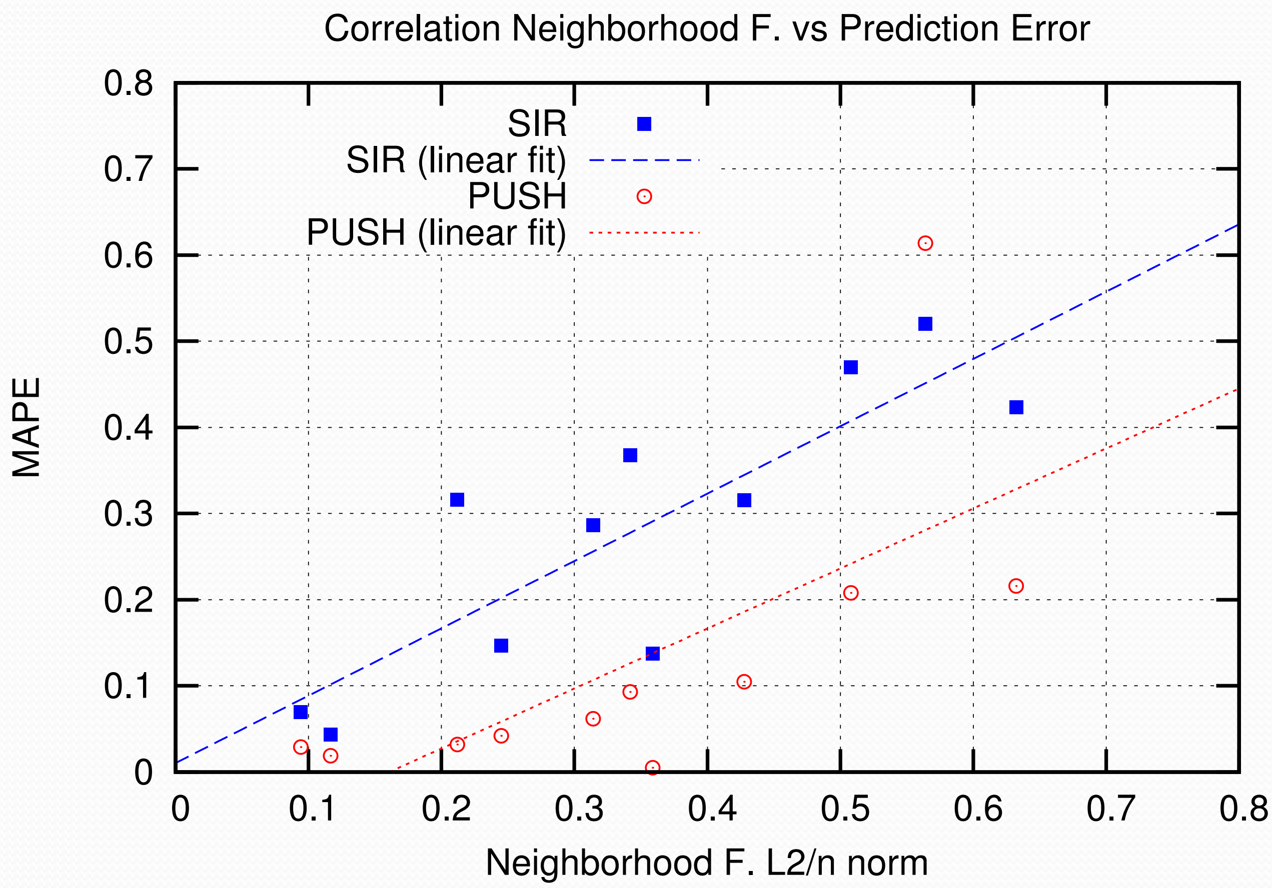
Web Graph Neighbourhood F.



Web Graph Prediction - SIR

Non-Social graphs have a neighbourhood function far from the configuration model.

Neighbourhood F. vs Prediction Quality



The correlation is strong and statistically significant.

Conclusion

- Rumour spreading processes can be predicted accurately in social graphs based on very limited information on the graph.
- Our predictor is provably correct and space efficient.
- We characterise the class of graph that can be predicted based on the Neighbourhood Function.
- We would like to extend our model to more nuanced diffusion processes.

Thank you for your attention!