Activity Patterns in Social Network Communities: a study on scale invariance

Matteo Garbellini

Università degli Studi di Milano

matteo.garbellini@studenti.unimi.it

23 Luglio 2019

Overview

1. Introduction
   - question
   - dataset
2. Theoretical Aspects
   - community detection
   - resolution parameter
   - intertime and spike trains
3. Computational Aspects
   - tools
   - process
4. Results
   - normalized activity
   - intertime
   - node activity
5. Conclusions and Further Developments

Introduction

Question

Do differently sized communities have different patterns of activity, or are these patterns scale invariant

- Dataset: Twitter Activity before/after the announcement of the discovery of the Higgs Boson

Dataset

Twitter activity before/during/after the announcement of the discovery of the Higgs Boson on July 12th 2012

- *higgs-social-network.edges*: directed graph of following/followers twitter users
  - 450000 nodes (users)
  - 14 million edges (friendships)
- *higgs-activity-time*: timestamped interactions between users, based on type of interaction Retweet(RT), Mention (MT), and Replies (RE)
  - Time frame: July 11th 0.00am to July 12th 11.59pm
  - 500000 events (interactions)
  - Format: *UserA UserB timestamp interaction*

Community Detection

Choosing the right community detection algorithm is an important step in the dataset analysis. For large networks **modularity** based algorithms perform the best. A first analysis was done using the Louvain Algorithm, while the final results were obtained using the CPM Algorithm

- Louvain Modularity Algorithm[1]
  - fast for large graph
  - small communities tend to be merged
- Constant Potts Model[2]
  - efficient *Louvain* alternative
  - almost Resolution Limit Free
  - able to discover small sub-communities

[1] Blondel, V. D., Guillaume, J., Lefebvre, E. (2008). Fast unfolding of communities in large networks, 112.

[2] Traag, V. A., Dooren, P. Van, Nesterov, Y. (2011). Narrow scope for resolution-limit-free community detection.

Community Detection: Constant Potts Model

The Constant Potts Model compares the network to a constant parameter γ instead of a null-model like the Louvain algorithm. It works by minimizing

$$\mathcal{H} = -\sum_{i,j}(A_{ij}\omega_{ij} - \gamma)\delta(\sigma_i, \sigma_j)$$

where γ is the so-called **resolution parameter**. Follows the inequality

$$n_c > \sqrt{\frac{1}{\gamma}}$$

where n_c is the cluster size lower bound.

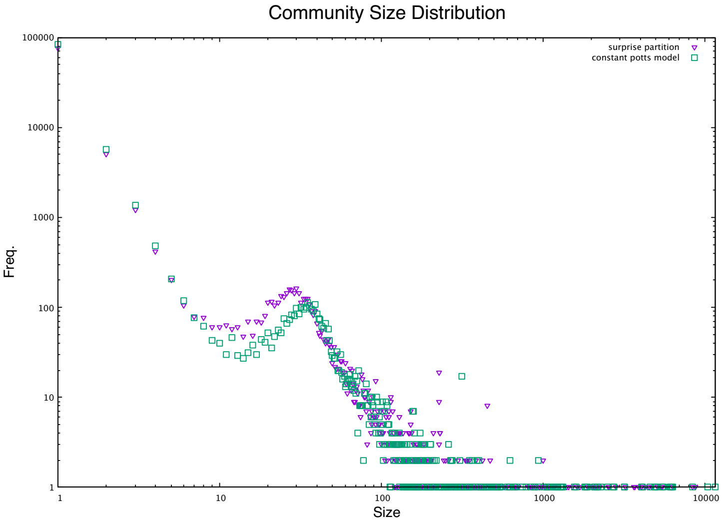
Community Detection: Choosing the Resolution Parameter

Choosing the resolution parameter is a delicate step of community detection

- resolution profile
- stable partitions
- research-oriented lower bound size
- (my case) cross-reference with *Surprise Partition Algorithm*[3]

[1]Aldecoa, R., Marn, I. (2011). Deciphering network community structure by surprise

Community Detection: size distribution CPM vs Surprise Partition



Activity Classification: Activity Index and Activation Index

Two parameters are proposed: the **activity index** and the **activation index**, defined as follows:

- Activity Index Λ

$$\Lambda = \frac{Events}{ActiveNodes}$$

- Activation Index Υ

$$\Upsilon = \frac{Activations}{ActiveNodes}$$

Activity Classification: Events Intertime

Activity Classification: Spike Trains and Local Variation

To uncover the dynamics of communications spikes (bursts), **local variation** L_v is applied, providing a local temporal measurment usually defined to characterize non-stationary neuron spike trains [4]

$$L_v = \frac{3}{N-2} \sum_{i=2}^{N-1} \left(\frac{(\tau_{i+1} - \tau_i) - (\tau_i - \tau_{i-1})}{(\tau_{i+1} - \tau_i) + (\tau_i - \tau_{i-1})} \right)^2$$

where N is the number of spikes and $\Delta\tau$ is the backward and forward delay.

[4] Sanli, C., Lambiotte, R. (2015). Temporal pattern of online communication spike trains in spreading a scientific rumor : how often , who interacts with whom?

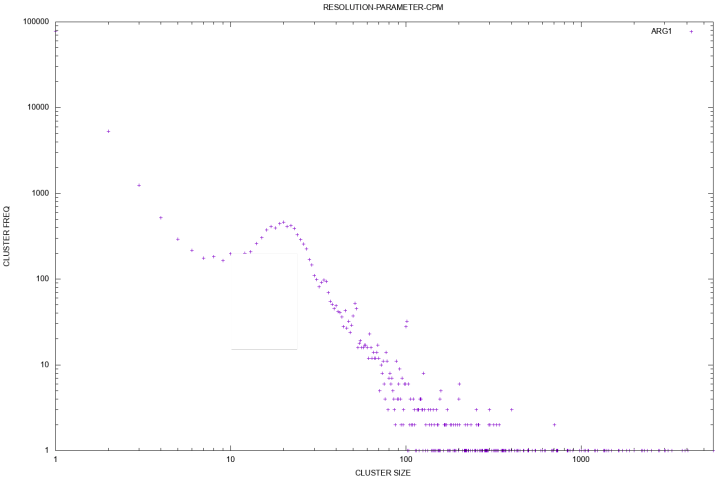
Computational Aspects: tools

- Tools Used
  - Community Detection: igraph w/ Python using **leidenalg** algorithm (Traag)
  - Community Analysis: C++ , awk and bash scripts
  - Graphs and fits: gnuplot

Computational Aspects: process

- higgs-community-detection.py: outputs detected communities
- higgs-preprocess-analysis.sh: performes basics parsing and file reformat
- higgs-analysis.cpp: builds all necessary information on the network and outputs all graph data

Insert figure here: cluster-size vs cluster-frequency



RESULTS: bin-size Definition

Communities sizes are classified following this general rule

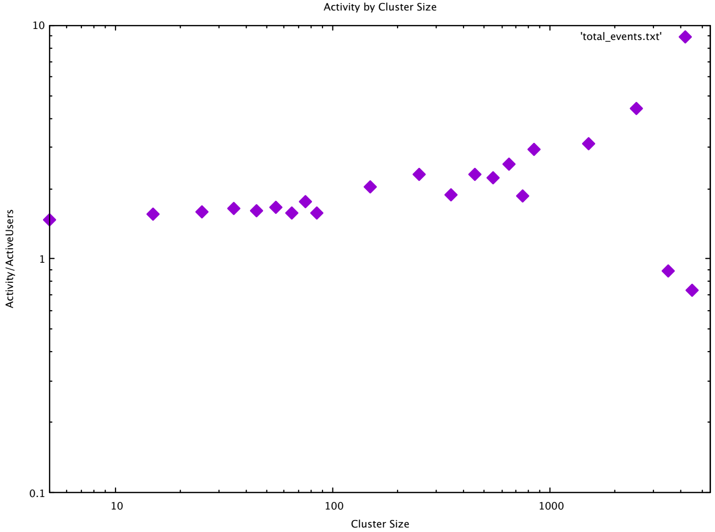
- Very Small : < 25 **not considered in analysis**
- Small: $25 - 100$
- Medium: $100 - 1000$
- Large: $1000 - 5000$
- Very Large: > 5000

The actual analysis is done considering log-sized bin averages. Each class has 10 sub-classes (10,20..100,200..1000,2000..)

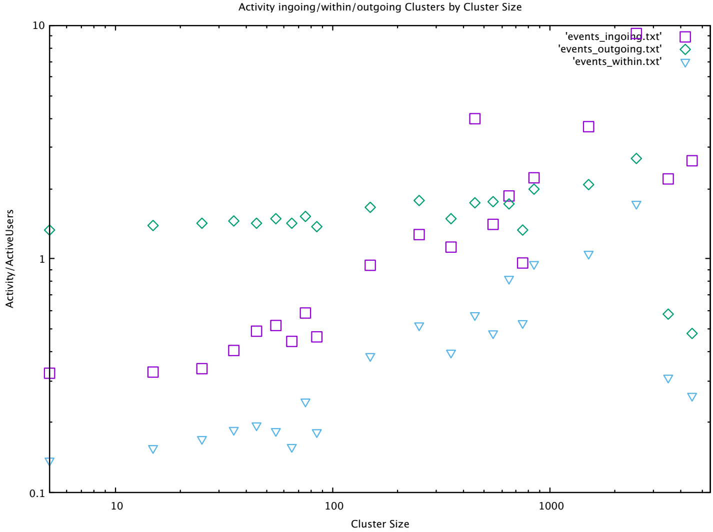
RESULTS OVERVIEW

- **Activity by cluster size**
  - ingoing / outgoing / within cluster
  - retweet / mention / reply
  - ingoing / outgoing / within by type (rt, re, mt)
- **Average node activity by cluster size**
- **Average node intertime by cluster size**

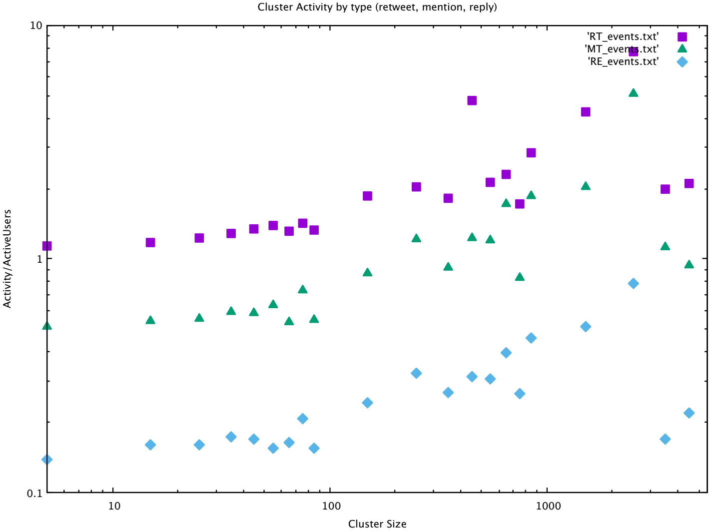
RESULTS: Cluster Activity



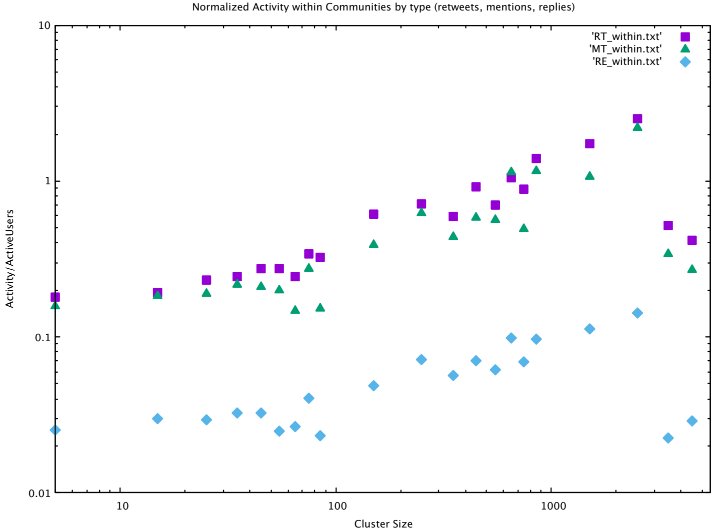
RESULTS: Cluster Activity (ingoing/outgoing/within)



RESULTS: Cluster Activity by type



RESULTS: Activity Within Cluster by type



RESULTS: Node Activation