Thesis notes

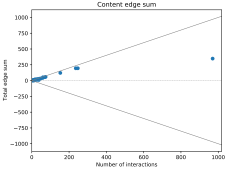
16th March

Negative edge fractions for many datasets

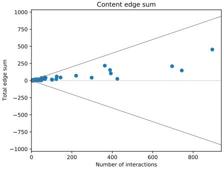
Table: Negative edge fractions for graphs built on 200 contents of different subreddits

r/cats	Pictures and videos about cats	0.169 23
r/Covid19	Scientific discussion of the pandemic	0.298 14
r/programming	Computer Programming discussions	0.302 65
r/climate	News about climate and related politics	0.391 79
r/Football	News, Rumours, Analysis about football	0.411 03
r/Economics	News and discussion about economics	0.417 30
r/Politics	News and discussion about U.S. politics	0.511 22
r/AskTrumpSupporters	Q&A between Trump supporters and non supporters	0.532 99

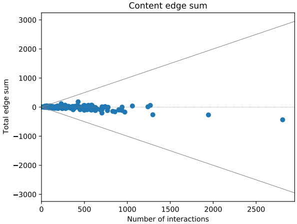
Negative edge fraction for number of interactions



(a) r/cats



(b) r/programming



(c) r/asktrumpsupporters

The echo chamber problem - notation

- $G = (V, E^+, E^-)$ interaction graph
- $\mathcal{C}$ set of contents
- $C \in \mathcal{C}$ content, $\mathcal{T}_C$ set of threads associated with C. A thread $T \in \mathcal{T}_C$ is a subgraph of G
- $U \subseteq V$ subset of users, $T[U]$ subgraph of T induced by U. $|T(U)|$ is the number of edges of this subgraph

The echo chamber problem - notation

- $\eta(C)$ fraction of negative edges associated with C (analogous definition for a thread T). Content (or thread) controversial if $\eta \in [\alpha, 1]$
- $\hat{\mathcal{C}} \subseteq \mathcal{C}$ set of *controversial* contents
- $\mathcal{S}_C(U)$ set of *non controversial* threads induced by U, for *controversial* contents, i.e.

$$\mathcal{S}_C(U) = \{T[U] \; s.t. \; T[U] \; non \; controversial, T \in \mathcal{T}_C, C \in \hat{\mathcal{C}}, U \subseteq V\} \tag{1}$$

The echo chamber problem

Goal: given an interaction graph G, find $U \subseteq V$ maximing

$$\xi(U) = \sum_{C \in \hat{\mathcal{C}}} \sum_{T[U] \in S_C(U)} |T[U]| \tag{2}$$

A possible initial implementation

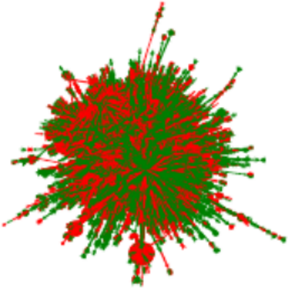
Algorithm 0: Greedy approach

$U = \{$ random node $\}$;
while $\xi(U)$ *can be increased by adding a node* **do**
  With probability β add to U the node increasing more the score $\xi(U)$ (taking into account variations in $S_C(U)$);
  With probability $(1-\beta)$ remove from U the node increasing less the score $\xi(U)$. This node will be ignored in the next iteration;
end

- Process is repeated for many nodes and maximum score is selected
- Final score is divided by the number of nodes of the graph.
- Set of users is *compacted* by the random node removal
- β regulates *density* of the user group

About community detection

- Follow graph may be too sparse and communities may end up corresponing to connected components in the interaction graph



(a) Giant Component, people following @nytimes

(b) Another component, actual group of friends discussing one or more contents

An alternative to community detection: social balance

Distance from social balance is measured by counting *frustrated* edges [1].

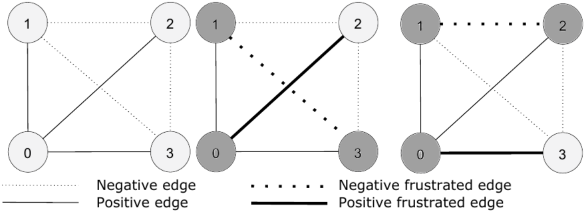
Each node gets a binary label; if x_i and x_j labels of nodes, an edge x_{ij} is frustrated if

- x_{ij} is negative and $x_i = x_j$
- x_{ij} is positive and $x_i \neq x_j$

The problem tries to find optimal label assignments to minimize the number of frustrated edges with Linear Programming.

[1] *From Balance and Frustration in Signed Networks* by Aref, Wilson

An alternative to community detection: social balance



(a) An example signed graph (b) Two node colourings (both optimal) and their resulting frustrated edges

- Labels can be used as to identify group of users
- Can be called recursively to find *inner* groups