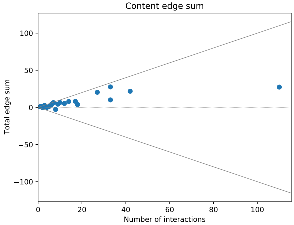
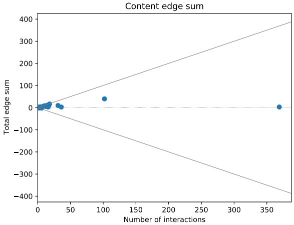
Thesis notes

23rd March

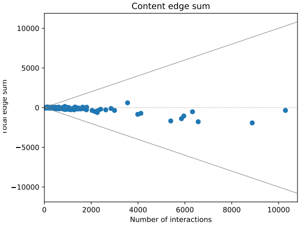
Detecting controversial content



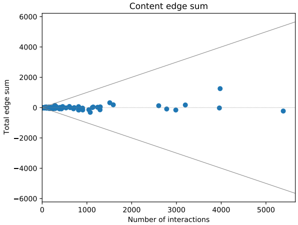
(a) @bbcscience



(b) @bbctech

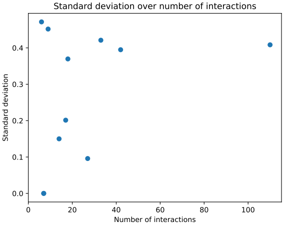


(c) @foxnews

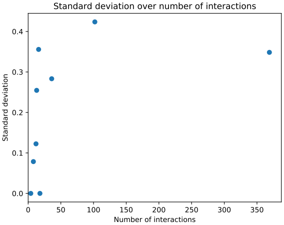


(d) r/politics

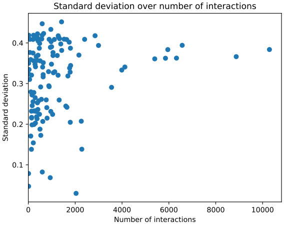
Detecting controversial content



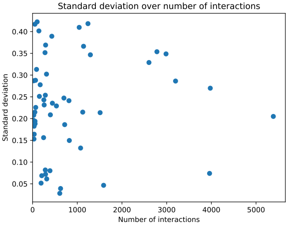
(a) @bbcscience



(b) @bbctech



(c) @foxnews



(d) r/politics

Detecting controversial content

- Controversial content usually receives many more replies
- Another possibility for detecting it (needs to be verified)
    1. select content C with high standard deviation (of the fraction of negative edges $\eta(C)$), which may be associated with an higher number of interactions
    2. keep content C whose $\eta(C) > \alpha$

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.

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

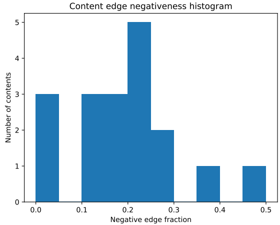
The echo chamber problem

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

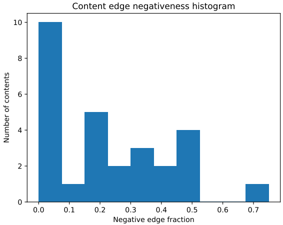
$$\xi(U) = \sum_{C \in \hat{\mathcal{C}}} \sum_{T[U] \in S_C(U)} |T[U]| \tag{2}$$

The set of users maximing the expression is denoted as $\hat{U}$ and the corresponding score is $\xi(G)$

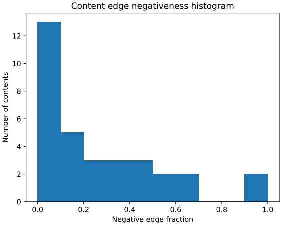
The datasets - negative edge fractions for contents



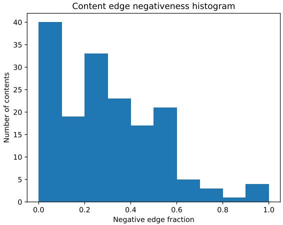
(a) @emanews



(b) @bbcscience



(c) @bbctech

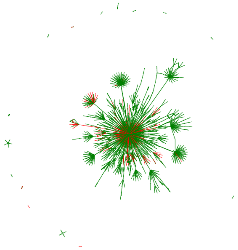


(d) @bbcsports

Echo chamber scores of connected components

Table: Echo chamber scores, by components

Source	|V|	|E|	$\xi(G)$	$\|\hat{U}\|$	$\xi(G)/\|\hat{U}\|$
@emanews	1226	1842	0	0	-
@bbcscience	447	388	4	2	0.5
@bbctech	793	719	26	12	2.17
@bbcsports	1645	2457	0	0	-



An initial implementation

Algorithm 1: 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

An initial implementation - results

Process was repeated $\sqrt{n}$ times for a graph with n nodes

Table: Echo chamber scores, greedy approach

Source	$\|V\|$	$\|E\|$	β	$\xi(G)$	$\|\hat{U}\|$	$\xi(G)/\|\hat{U}\|$
@emanews	1226	1842	0.6	0	0	-
			0.7	0	0	-
			0.8	0	0	-
			0.9	0	0	-
			1	0	0	-

An initial implementation - results

Table: Echo chamber scores, greedy approach

Source	$\lvert V \rvert$	$\lvert E \rvert$	β	$\xi(G)$	$\lvert \hat{U} \rvert$	$\xi(G)/\lvert \hat{U} \rvert$
@bbcscience	447	388	0.6	2	2	1
			0.7	0	0	-
			0.8	6	3	2
			0.9	3	2	1.5
			1	2	3	0.67
@bbctech	793	719	0.6	28	9	3.11
			0.7	28	9	3.11
			0.8	28	9	3.11
			0.9	34	14	2.42
			1	28	9	3.11

An initial implementation - results

Table: Echo chamber scores, greedy approach

Source	|V|	|E|	β	$\xi(G)$	$\|\hat{U}\|$	$\xi(G)/\|\hat{U}\|$
			0.6	173	16	10.8
@bbcsports	1645	2457	0.7	159	12	13.25
			0.8	220	32	6.87
			0.9	224	36	6.22
			1	228	40	5.7

Note: algorithm was stopped at the 30th iteration.

Another possible greedy approach

Inspired to the greedy algorithm proposed in [Cha00]

Algorithm 2: Greedy approach

$U = \{$ all nodes $\}$;
$S = \xi(U)$;
while *U is not empty* **do**
  remove from U the node contributing less to the score $\xi(U)$;
  update S if the current score is higher;
end

Computing exactly the score

$$\textit{maximize} \sum_{ij \in E(\hat{\mathcal{C}})} x_{ij} \tag{3}$$

$$x_{ij} \leq y_i \quad \forall ij \in E(\hat{\mathcal{C}}) \tag{4}$$

$$x_{ij} \leq y_j \quad \forall ij \in E(\hat{\mathcal{C}}) \tag{5}$$

$$\sum_{ij \in E^-(T_k)} x_{ij} - \alpha \sum_{ij \in E(T_k)} x_{ij} \leq M_k(1 - z_k) \quad \forall T_k \in \mathcal{T}_C, C \in \mathcal{C} \tag{6}$$

$$\sum_{ij \in E(T_k)} x_{ij} \leq N_k z_k \tag{7}$$

$$x_{ij} \in \{0, 1\} \quad \forall ij \in E(\hat{\mathcal{C}}) \tag{8}$$

$$y_i \in \{0, 1\} \quad \forall i \in V \tag{9}$$

$$z_k \in \{0, 1\} \quad \forall T_k \in \mathcal{T}_C, C \in \mathcal{C} \tag{10}$$

Computing exactly the score

A thread T_k is non controversial if $\eta(T) \leq \alpha$, i.e.

$$\frac{\sum_{ij \in e^-(t_k)} x_{ij}}{\sum_{ij \in e(t_k)} x_{ij}} \leq \alpha \tag{11}$$

which can be written as

$$\sum_{ij \in e^-(t_k)} x_{ij} - \alpha \sum_{ij \in e(t_k)} x_{ij} \leq 0 \tag{12}$$

Computing exactly the score

So, for controversial content

$$\sum_{ij \in e^-(t_k)} x_{ij} - \alpha \sum_{ij \in e^((t_k)} x_{ij} > 0 \tag{13}$$

and, for the constraint

$$\sum_{ij \in E^-(T_k)} x_{ij} - \alpha \sum_{ij \in E(T_k)} x_{ij} \leq M_k(1 - z_k) \quad \forall T_k \in \mathcal{T}_C, C \in \mathcal{C} \tag{14}$$

it will be $z_k = 0$. So controversial $T_k \implies z_k = 0$.

Computing exactly the score

$$\sum_{ij \in E(T_k)} x_{ij} \leq N_k z_k \qquad (15)$$

will set to 0 edges associated to controversial threads T_k.
So controversial $T_k \implies z_k = 0 \implies x_{ij} = 0 \quad \forall ij \in E(T_k)$.

Computing exactly the score

N_k and M_k can be simply m, the number of edges in the graph.

Bibliography

[Cha00] Moses Charikar. "Greedy Approximation Algorithms for Finding Dense Components in a Graph". In: *Approximation Algorithms for Combinatorial Optimization*. Ed. by Klaus Jansen and Samir Khuller. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 84–95. ISBN: 978-3-540-44436-7.