Thesis notes

6th April

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.

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

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 $\xi(G)$

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

Another possible greedy approach

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

$$\text{maximize} \sum_{ij\in E(T_k), T_k\in\mathcal{T}_C, C\in\hat{\mathcal{C}}} x_{ij}^k \tag{3}$$

$$x_{ij}^k \leq y_i \qquad \forall ij \in E(T_k), T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{4}$$

$$x_{ij}^k \leq y_j \qquad \forall ij \in E(T_k), T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{5}$$

$$x_{ij}^k \leq z_k \qquad \forall ij \in E(T_k), T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{6}$$

$$x_{ij}^k \geq -2 + y_i + y_j + z_k \qquad \forall ij \in E(T_k), T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{7}$$

$$\sum_{ij\in E^-(T_k)} x_{ij}^k - \alpha \sum_{ij\in E(T_k)} x_{ij}^k \leq 0 \qquad \forall T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{8}$$

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

$$0 \leq x_{ij}^k \leq 1 \qquad \forall ij \in E(T_k), T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{10}$$

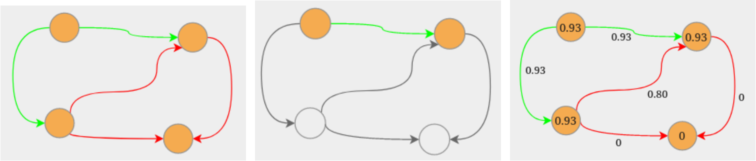
$$0 \leq z_k \leq 1 \qquad \forall T_k \in \mathcal{T}_C, C \in \hat{\mathcal{C}} \tag{11}$$

Reconstructing solutions from relaxation results

Algorithm 3: Relaxation results reconstruction

$E_{ordered}$:= Edges ordered in descending order according to their value x_{ij} ;
$U := \emptyset$;
foreach $e_{ij} \in E_{ordered}$ **do**
    Add v_i and v_j if not present in U;
    Calculate $\xi(U)$;
end

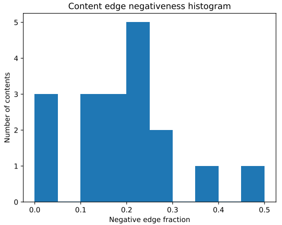
Computing exactly the score - relaxation algorithm example



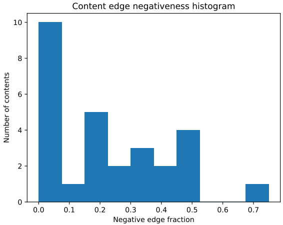
(a) Original graph (b) Integer solution (c) Relaxed solution

Figure: Model solution for a graph with a single thread, for $\alpha = 0.3$

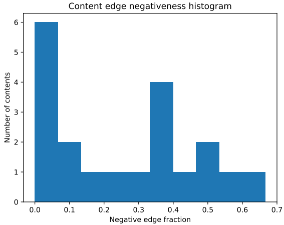
The datasets - negative edge fractions for contents



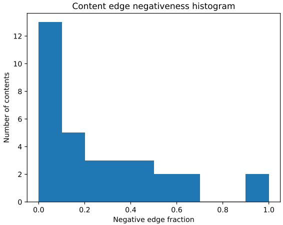
(a) @emanews



(b) @bbcscience



(c) @bbcentertainment



(d) @bbctech

An initial implementation - results

- Beta algorithm was repeated $\sqrt{n}$ times for a graph with n nodes

Table: Echo chamber scores, greedy approach

Source	|V|	|E|	$(\xi_\beta(G), \beta)$	$\xi_{peel}(G)$
@emanews	1226	1842	(0, *)	0
@bbcscience	477	388	(3, 0.9)	7
@bbcentertainment	220	183	(21, 1.0)	16
@bbctech	793	719	(101, *)	107

Table: Echo chamber scores, MIP approaches

Source	$\xi_{MIP}(G)$	$\xi_{MIPr}(G)$	$\xi_{MIPr\_alg}(G)$
@emanews	0	1.43	0
@bbcscience	7	10.76	7
@bbcentertainment	34	41.69	34
@bbctech	309	326.63	309

Computing exactly the score - relaxation algorithm counterexample

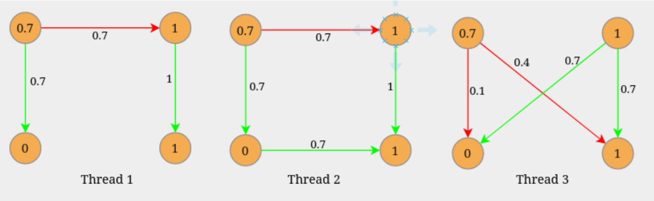


Figure: Small graph in which the algorithm may not find the optimum. The exact solution excludes the top-left vertex and scores 5.

Echo Chamber Problem hardness

Theorem

The Echo Chamber Problem is $\mathcal{NP}$-hard.