Reducing Polarization in Social Media

A Thesis

submitted to the designated
by the General Assembly
of the Department of Computer Science and Engineering
Examination Committee

by

Leonidas Boutsikaris

in partial fulfillment of the requirements for the degree of

COMPUTER SCIENCE AND ENGINEERING

University of Ioannina
February 2021

Examining Committee:

- Panayiotis Tsaparas, Associate Professor, Department of Computer Science and Engineering, University of Ioannina (Supervisor)
- ? ?, Associate Professor, Department of Computer Science and Engineering, University of Ioannina
- ? ?, Associate Professor, Department of Computer Science and Engineering, University of Ioannina

Table of Contents

Chapter 1

Introduction and the Theory Behind Polarization

1.1 Introduction

Polarization describes the division of people into two contrasting groups or sets of opinions or beliefs. The term is used in various domains such as politics and social studies. For example political polarization refers to the divergence of political attitudes to ideological extremes. Social studies use this term to describe the segregation within a society in terms of income inequality or social and class status.

Currently social media are one the biggest sources for news and information as more and more people go online to discuss and socialize. Polarization is linked with harmful effects such as intensifying stereotypes and creating echo chambers.

In echo chambers an individual get their news only from like-minded people as they share and reinforce one another's opinions. Additionally, the fact that people tend to ignore opposing views, in combination with algorithmic personalization, results in a significant increase of polarization.

1.2 Social and Psychological Factors

Individuals experience discomfort when given data that actively challenge their opinions. In the field of psychology, cognitive dissonance occurs when a person holds two or more contradictory beliefs, ideas, or values and experiences psychological stress because of that. In simple terms dissonance is defined as a the lack of agreement.

Individuals want to reduce the discomfort that is caused from cognitive dissonance. Reduction occurs by strengthening opinions that come in agreement with their own and downplaying everything that challenges them. This leads individuals to a selective exposure on information [1]. Selective exposure is also demonstrated in groups where people assign themselves with certain social identities.

We can understand this better with the self-categorization theory. This theory stems from the social identity theory, which holds that conformity stems from psychological processes. Accordingly, proponents of the self-categorization model hold that group polarization occurs because individuals identify with a particular group and conform to a prototypical group position that is more extreme than the group mean. It is shown that groups of people tend to make decisions that are more extreme than the initial inclination of its members [2].

1.3 Polarization online

Online entities such as news or social media platforms are aware of their users opinions and aim to maximize their satisfaction. As discussed above, platforms will present content in a way that minimizes psychological stress. This leads to media bias.

Media bias is the bias or perceived bias of journalists and news producers within the mass media in the selection of many events and stories that are reported and how they are covered. When this happens online, personalization of the content creates algorithmic bias.

Algorithmic bias describes systematic and repeatable errors in a computer system that create unfair outcomes, such as privileging one arbitrary group of users over others.

Bias can emerge due to many factors like the design of the algorithm. Due to personalization we don't see the same content and this is the main reason for the formation of filter bubbles.

1.4 Filter Bubbles

Filter bubbles are the echo chambers of social media. In news media, an echo chamber is a metaphorical description of a situation in which beliefs and opinions are strengthened by communication and repetition inside a closed system. It is important to distinguish the difference between echo chambers and filter bubbles. This two concepts are almost identical, however, filter bubbles are a result of algorithms that choose content based on previous online behaviour, as with search histories or online shopping activity.

1.5 Polarization and Society

Political polarization can be defined as the difference in ideological extremes but in political science almost in every context polarization is considered as the gap between the political parties of a society.

Most of the time political parties disagree on policy issues and that is the main drive of democracy. With heightened polarization the followers of each political party start fearing that the other will destroy their society with their agendas.

Destroying the other side becomes their only objective and this is how democracies fall apart.

Social networks are frequently liable for terrorism. Terrorist leaders create communities of individuals that have the same opinions and fuel them with each other. As mentioned in 1.2 when like-minded people discuss with each other they tend to move toward extreme positions. This has a bigger effect when people are already quite extreme.

Terrorist leaders know this and they try to make sure that all individuals inside this community will speak and interact with people that have the same extreme direction. If members of the community think that they have a shared identity the polarization will grow. Terrorist leaders will also repress opposing views and will not tolerate internal disagreement. They take every step needed to ensure unity.

Most individuals lack in confidence on their own views or have more confidence that are willing to show. Fear of marginalization or being proven wrong make them present a moderate version of theirselves. In either case, group dynamics can push people toward a more extreme position.

Social influence also plays a great role. People have a certain image of themselves and how they want to be viewed by others. Most people like to think of themselves as not identical to but as different from others, but only in the right direction and to the right extent. There is evidence that social influence is an independent factor behind group polarization; consider in particular the fact that mere exposure to the views of others can have this effect, even without any discussion at all [2].

Combining these factors we obtain a highly dangerous and highly polarized community for everyday life.

Chapter 2

Related Work

2.1 Measuring the Polarization of a Network

At first we have to measure the opinion polarization in a social Network. The actions and information of a user can give us insights about his opinions on a topic e.g. accounts a user follows, content they repost, comments they make , etc. Using this information we can measure the polarization.

Assume a graph $G = (V, E)$ representing a network that is connected and undirected. Z will be the vector of expressed opinions for the whole network. Each value Z_i of the vector will represent a node and can be computed with the opinion-formation model of Friedkin and Johnsen.

The length of the opinion vector $||z||^2$ measures the polarization and $\pi(z) = \frac{||z||^2}{n}$ is defined as the polarization index of the network, where n is the number of nodes in the graph so the polarization index can be independent of the network size.

There is a direct link between this opinion model and random walks. Given the graph $G = (V, E)$ we can construct the augmented graph $H(V \cup X, E \cup R)$. For each vertex of V we will add a new vertex on X and a directed edge (v_i, x_i) in R.

The node x_i corresponds to the internal opinion of the node v_i. In the model we follow z_j or else the expressed opinion of a user that can be computed by the probability of $P(x_i|v_j)$. This probability represents that a random walk on the augmented graph that started from the node V_j ended at the node X_i or else how much likely the probability of user V_j adopting the opinion of user V_i. This probability depends on the structure of the graph.

Two problems are introduced, the $ModerateInternal$ and the $ModerateExpressed$. When moderating opinions a small set of nodes T_s is being set to zero, in each problem, as their names suggests, internal or external opinions are set to zero. Two algorithms are proposed for the $ModerateInternalproblem$.

A greedy algorithm that finds the set T_s of nodes iteratively according to the biggest decrease it causes and the Binary Orthogonal Matching Pursuit (BOMP) algorithm. For the $ModerateExpressed$ problem the same greedy algorithm is used. [3]

2.2 Polarization and Disagreement

Another way of looking at polarization is by combining it with disagreement. The main problem of minimising polarization and disagreement lies in the opinions of each user and how targeted ads and recommendations influence their opinions.

Considering the disagreement in combination with polarization a network can choose how to respond in different situations. Their recommendation system could choose between keeping the disagreement low or exposing users to radically different opinions. There are situations that this optimisation can reduce the overall polarization-disagreement in the network by recommending edges in different parts of the network than the ones that agree with the human confirmation bias.

Given a social network $G = (V, E, w)$ and initial opinions $s : V \to [0, 1]$ the equilibrium vector according to the Friedkin-Johnsen model is defined as $z^* = (I + L)^1 s$ where L is the laplacian matrix of the graph and I the identity matrix. Disagreement of $d(u, v)$ of edge (u, v) is defined as the squared difference between the opinions of u, v at equilibrium: $d(u, v) = w_{uv}(Z_u^* - Z_v*)^2$.

The total disagreement is defined as $D_{G,s} = \sum_{(u,v)\epsilon E} d(u, v)$. With $\bar{z} = z^* - \frac{z*^T \overrightarrow{1}}{n} \overrightarrow{1}$ polarization is measured as a deviation from the average with the standard definition of variance as $P_{G,s} = \sum_{u\epsilon V} \bar{z}_u^2 = \bar{z}^T \bar{z}$.

The polarization-disagreement index is defined as follows $I_{G,s} = P_{G,s} + D_{G,s}$. The objective is to minimize this index.

Muco and Tsourakakis have shown that minimising $\bar{z}^T \bar{z} + \bar{z}^T L \bar{z}$ is the same to minimising the polarization-disagreement index. Here, L is a matrix among the set of valid combinatorial Laplacians of connected graphs.[4]

2.3 Quantifying and Minimizing Risk of Conflict in Social Networks

We know for a fact that opinions are formed through social interactions and in every interaction conflict arises. Online networks offer public access to social disputes on controversial matters that allows the study and moderation of them. The majority of studies are based in the Friedkin-Johnsen model.

The main problem is with the Friedkin-Johnsen model metrics. The external opinion of a user, which by definition is hard to measure, combined with the internal opinion which is impossible to be measured. Another problem occurs in the editing of the social graph. We edit the social graph in a way that minimises the conflict of a certain social issue. This can lead to an increased conflict of one or more social issues inside the network.

Chen, Lijffijt and De Bie still use the Friedkin-Johnsen model to evaluate the network conflict but the quantifications depend only on the network topology in a way that the conflict can be reduced over all issues. Worst-case(WCR) conflict risk and average-case conflict risk(ACR) are defined to represent two separate problems, how the network can be minimised in the worst case or in the average case scenario by altering the social graph.

These problems consider the measures of internal conflict, external conflict, and controversy. Internal conflict (ic) measures the difference of the internal and the expressed opinion of a user. $ic = \sum_i (z_i - s_i)^2$.

External conflict (ec) measures how different are the opinions of the neighbours with each other. $ec = \sum_{(i,j)\epsilon E} w_{ij}(z_i - z_j)^2$.

Controversy (c) measures the variation of the opinions in the network and is independent of the social graph structure. $c = \sum_i {z_i}^2$.

These measures are not independent. Reducing one of them results in the increase of another. This leads to the conservation law of conflict. $S^T S = ic + 2ec + c$.

There are two methods of minimising the conflict of the network for each of the ACR and WCR problems. One is a gradient method that considers deleting and adding edges simultaneously and the other is a descent method that suggests deleting or adding a single edge. Chen, Lijffijt and De Bie used small world random networks and random networks with binomial and power law degree distribution to find out what types of networks have the highest risks for every conflict measure they defined.

A small world network is a type of graph in which most nodes are not neighbours of one another, but the neighbours of any given node are likely to be neighbours of each other and most nodes can be reached from every other node by a small number of hops or steps. They found that the small world networks are the most high-risk for the ic metric. For c and r the most high-risk network depends on the density.[5]

2.4 Reducing Controversy by Connecting Opposing Views

Garimella et al. relly on a measure of controversy that is shown to work reliably in multiple domains in contrast with other measures that focus on a single topic. The controversy measure consists of the following steps:

1. Given a topic t they create an endorsement gragp $G = (V, E)$. This graph represents users who have generated content relevant to t. For example hashtags of a user.

2. The nodes of this graph a re partitioned in two disjoint sets X and Y. The partition is obtained using a graph-partition algorithm.

3. The last step, is computing the controversy measure through a random-walk, thus creating the controversy score RWC. This score is defined as the difference of the probability that a random walk starting on one side of the partition will stay on the same side and the probability that the random walk will cross to the other side. A personalized PageRank is used where the restart probabilities are set to a random vertex of each side.

Garimella et al. states that real graphs often have a star-like structure. Small number of higly popular vertices have a lot of incoming edges. These nodes can be seen as thought leaders and their followers. It is shown that connecting the high degree vertices minimises the RWC score.

Probabilities are also incorporated in the sense that a new edge addition may be not accepted by the user. The polarity here is defined as $R_u = p^X(u) - p^Y(u)\epsilon[-1, 1]$.

The definition of $p^X(u)$ and $p^Y(u)$ is the fraction of other vertices u' for which $lu'^X < lu^X$ and $lu'^Y < lu^Y$.

In addition lu^X and lu^Y stand for the expected time a random walk needs to hit the high degree vertices of X and Y respectively starting from u. Considering u and v as 2 different and not connected users $P(u, v)$ is defined as the probability that u accepts a recommendation to connect with v.

Let R_u and R_v the polarity of these users respectively. $P(u, v)$ is estimated from the training data by obtaining $N_{Endorsed(R_u,R_v)}/N_{Exposed(R_u,R_v)}$.

The $Endorsed(R_u, R_v)$ and $Exposed(R_u, R_v)$ values represent the number of times a user with polarity R_v was exposed/endorsed content generated by a user with R_u. For example v follows u, thus v is exposed to all content u generates.

Finally we can re-define the problem as the expected decrease of RWC. $E(u, v) = p(u, v) * RWC_{u \to v}$

Chapter 3

Premilinaries and Problem Definition

3.1 The Friedkin and Johnsen Model

This model uses information about the opinion of the user, internal and external, but also the constant update of the external opinions of the neighbourhood of the user e.g. the friend list or the accounts the user follows to compute an opinion vector. This vector is a metric for the whole social graph and can give us insights about its current situation. The vector values range from [-1,1]. Values closer to the range limits indicate bigger polarization. Polarized graphs create groups of nodes that are strongly connected with each other and feedback to one another the same extreme opinion of a certain topic. These groups can be seen clearly in the illustration of filter bubbles and often associated with politics and controversial issues of our society. Using a certain number of users we can achieve a reduction in the polarization of a network.

We can educate a group of users with the opposite view, and in terms of our model that means that we can modify the social graph by adding a connection between users of different opinions.

Let $G = (V, E)$ be a connected undirected graph representing a network. Let z be the vector of expressed opinions for the whole network. Each value of the vector represents a node and can be computed with the opinion-formation model of Friedkin and Johnsen as follows.

$$z_i = \frac{w_{ii} * si + \sum_{j \epsilon N(i)} w_{ij} * z_j}{w_{ii} + \sum_{j \epsilon N(i)} w_{ij}} \tag{3.1}$$

Where s_i denotes the internal and z_i the expressed opinion of a user. The internal opinion of a user corresponds to the views that inherently holds for a controversial topic while the expressed refers to the views that the user shares on a social network with his friends. The length of the opinion vector $||z||^2$ measures the polarization of the network. To make the polarization independent of its network we can normalize it by dividing it with the length of the vector z. An equivalent way of obtaining the vector z from a graph is the following: if L is the laplacian matrix of a graph $G = (V, E)$, and I is the identity matrix, then $z = (L + I)^{-1}S$ [6].

3.2 Measuring the polarization

We measure the polarization by its distance from a neutral opinion. We can quantify this with the length of the vector of the second norm L_2^2 [3].

$$\pi(z) = ||z||_2^2 \tag{3.2}$$

This value can be independent of the network if we normalize it by dividing with the size of the graph.

3.3 A small example of the Friedkin and Johnsen Model

We will now present a small example so we can build a basic understanding of the Friedkin and Johnsen Model. Consider a small graph that consists of two nodes, u and v with internal opinions of 1 and -1 and $w_{uu} = w_{vu} = w_{vv} = 1$.

$$z_u = \frac{1*1+1*1+1*(-1)}{1+2} = \frac{1}{3} \quad , \quad z_v = \frac{1*(-1)+1*(-1)+1*1}{1+2} = -\frac{1}{3} \tag{3.3}$$

$$\pi(z) = ||z||_2^2 = \sqrt{(\frac{1}{3})^2 + (-\frac{1}{3})^2}^{\,2} = \frac{2}{9} \tag{3.4}$$

We did not normalize the polarization index here by dividing with the size of the graph as we have a simple example.

3.4 Problem Definition

Real world events such as Brexit and the 2016 U.S. presidential elections gives us a clear hint about the polarization our society is witnessing.

Social media polarization has a strong effect on politics, opinion formation and how people interact with each other in a society. Users of social media are now receiving biased information that amplify their own viewpoints.

Enclosed in their filter bubble, they will ignore everyone else and only acknowledge opinions that fit their own reality.

In combination with fake news a malicious entity could use social media as a tool to polarize certain groups of people for their own interest.

Problem 1 [k-Addition]. Let $C \subseteq V \times V$ a set of edges that are not in the graph. We want to find a subset of $S \subseteq C$ of k edges whose addition to a graph G leads to the greatest reduction of $\pi(z)$.

3.4.1 Including probabilities into the problem

Problem 1 is trying to find edges that will minimize the polarization index. We must not take for granted that these edges will be accepted.

For example a social media user could reject a new follow/friend request. This leads us to consider additions with the expectation of being accepted. Let $E[\pi(z)]$ the expected polarization probability score that an edge (u, v) is accepted as a recommendation from u.

Problem 2 [K-Addition-Expected]. Given a graph $G = (V, E)$ and an integer k, we want to find a set of k edges $E' \subseteq V \times V \ E$ that when added to G creates a new graph $G' = (V, E \cap E')$ so that the expected polarization score $E[\pi(z)]$ is minimized.

3.5 Monotonicity of the Problem

We observe that $\pi(z)$ is not monotone with respect to the edge additions. This means that adding an edge will not necessarily decrease the polarization index.

We will show that this is true with a counter example. In the network 3.1 nodes 0, 2 and 3 have a value of $s_i = -1$, and nodes 2 and 4 have a value of $s_i = +1$. For both examples we assume that $w_{ii} = w_{ij} = w_{ji} = 1$ and n the number of nodes.

We will now compute the polarization index of the original graph

(a) (b)

Figure 3.1: Edge addition between opposed opinions.

$$L+I = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix}, \qquad (L+I)^{-1} = \begin{pmatrix} \frac{34}{55} & \frac{13}{55} & \frac{1}{11} & \frac{2}{55} & \frac{1}{55} \\ \frac{13}{55} & \frac{26}{55} & \frac{2}{11} & \frac{4}{55} & \frac{2}{55} \\ \frac{1}{11} & \frac{2}{11} & \frac{5}{11} & \frac{2}{11} & \frac{1}{11} \\ \frac{2}{55} & \frac{4}{55} & \frac{2}{11} & \frac{26}{55} & \frac{13}{55} \\ \frac{1}{55} & \frac{2}{55} & \frac{1}{11} & \frac{13}{55} & \frac{34}{55} \end{pmatrix},$$

$$s = \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}, \qquad (L+I)^{-1}s = \begin{pmatrix} \frac{34}{55} & \frac{13}{55} & \frac{1}{11} & \frac{2}{55} & \frac{1}{55} \\ \frac{13}{55} & \frac{26}{55} & \frac{2}{11} & \frac{4}{55} & \frac{2}{55} \\ \frac{1}{11} & \frac{2}{11} & \frac{5}{11} & \frac{2}{11} & \frac{1}{11} \\ \frac{2}{55} & \frac{4}{55} & \frac{2}{11} & \frac{26}{55} & \frac{13}{55} \\ \frac{1}{55} & \frac{2}{55} & \frac{1}{11} & \frac{13}{55} & \frac{34}{55} \end{pmatrix} \times \begin{pmatrix} -1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{-27}{55} \\ \frac{1}{55} \\ \frac{-5}{11} \\ \frac{-21}{55} \\ \frac{17}{55} \end{pmatrix}$$

$$\pi(z) = \frac{||z||_2^2}{n} = \frac{\sqrt{(\frac{-27}{55})^2 + (\frac{1}{55})^2 + (\frac{-5}{11})^2 + (\frac{-21}{55})^2 + (\frac{17}{55})^2}^2}{5} = 0.13785123966 \tag{3.5}$$

We will now compute the polarization index after the addition of the edge $1 \to 3$.

$$L+I=\begin{pmatrix}2&-1&0&0&0\\-1&4&-1&-1&0\\0&-1&3&-1&0\\0&-1&-1&4&-1\\0&0&0&-1&2\end{pmatrix},\qquad (L+I)^{-1}=\begin{pmatrix}\frac{59}{99}&\frac{19}{99}&\frac{1}{11}&\frac{8}{99}&\frac{4}{99}\\\frac{19}{99}&\frac{38}{99}&\frac{2}{11}&\frac{16}{99}&\frac{8}{99}\\\frac{1}{11}&\frac{2}{11}&\frac{5}{11}&\frac{2}{11}&\frac{1}{11}\\\frac{8}{99}&\frac{16}{99}&\frac{2}{11}&\frac{38}{99}&\frac{19}{99}\\\frac{4}{99}&\frac{8}{99}&\frac{1}{11}&\frac{19}{99}&\frac{59}{99}\end{pmatrix},$$

$$s=\begin{pmatrix}-1\\1\\-1\\-1\\1\end{pmatrix},\qquad (L+I)^{-1}s=\begin{pmatrix}\frac{59}{99}&\frac{19}{99}&\frac{1}{11}&\frac{8}{99}&\frac{4}{99}\\\frac{19}{99}&\frac{38}{99}&\frac{2}{11}&\frac{16}{99}&\frac{8}{99}\\\frac{1}{11}&\frac{2}{11}&\frac{5}{11}&\frac{2}{11}&\frac{1}{11}\\\frac{8}{99}&\frac{16}{99}&\frac{2}{11}&\frac{38}{99}&\frac{19}{99}\\\frac{4}{99}&\frac{8}{99}&\frac{1}{11}&\frac{19}{99}&\frac{59}{99}\end{pmatrix}\times\begin{pmatrix}-1\\1\\-1\\-1\\1\end{pmatrix}=\begin{pmatrix}\frac{-53}{99}\\\frac{-7}{99}\\\frac{-5}{11}\\\frac{-29}{99}\\\frac{35}{99}\end{pmatrix}$$

$$\pi(z)=\frac{||z||_2^2}{n}=\frac{\sqrt{(\frac{-53}{99})^2+(\frac{-7}{99})^2+(\frac{-5}{11})^2+(\frac{-29}{99})^2+(\frac{35}{99})^2}^{\,2}}{5}=0.14180185695 \tag{3.6}$$

We can see an increase of the polarization index after adding this particular edge. This example was discovered after brute-forcing different graph topologies with different combinations of opinion values.

Lemma 3.1. The polarization index does not necessarily decrease after an edge addition between opposing views.

Chapter 4

Algorithms

4.1 Intuition

To solve this problem we have to evaluate all possible edge combinations. Even for greedy heuristics we need to limit the edge candidates. The algorithm considers nodes that have high expressed value. According to our model the smallest decrease is happening when we connect different and extreme opinions.

We will now see why this statement holds by examining how the expressed opinion of the Friedkin and Johnsen model changes with an edge addition.

Consider an arbitrary example with two nodes inside a network. Node a has $z_a = -0.02$ and node b has $z_b = 0.5$. Also for this example we assume that $w_{ii} = w_{ij} = w_{ji} = 1$. If we connect these two nodes with an edge and recalculate the expressed opinions, both of the z_i denominators will be increased by one. This emerges from the fact that both nodes will have a new neighbour and that all weights equal with one. The numerator of the node a will be increased by a lot and the numerator of the node b will be decreased by a small value.

The new z_a will not change a lot because the addition of the numerator will approach the +1 addition of the denominator. On the other hand the new z_b will see a big change, the numerator will have a small decrease, the denominator will have the +1 addition, thus creating a big decrease overall for this node. We can clearly see that only one of the two nodes will have a big decrease.

Now consider a second example with two nodes, node c has $z_c = -0.8$ and node d has $z_d = 0.9$. After the addition node d will see a big decrease because we add two conflicting values that almost neutralise each other on the numerator but the addition of the +1 on the denominator stands still. On the other hand node c will also see a big decrease for the same reason. With this type of connection both of the nodes have a significant decrease.

In a setting that $w_{ii} \neq w_{ij} \neq w_{ji} \neq 1$ the same intuition holds but the weights must be also considered to create a strategy for neutralising each other.

4.2 Heuristics

In this section we consider a greedy algorithm and some heuristics for minimising $\pi(z)$. All the heuristics use the intuition that connecting the most extreme opinions of each community draw both of them into neutrality. The algorithms use two lists. One for each viewpoint sorted according to their opinion value.

We can classify these algorithms based on how they perceive the network when exploring new edge additions. This is derived from the fact that when adding an edge to the network the structure of the graph is changes.

This leads to a different Z vector. The heuristics can choose to recompute the Z vector or not.

4.2.1 Heuristics that consider network changes

We begin with a Greedy algorithm. Greedy algorithms work in stages and during each stage a choice is made which is locally optimal.

After finishing, the totality of all these choices produce a globally optimal solution.

If a greedy algorithm does not lead us to a globally optimal solution we can refer to it as a heuristic or a greedy heuristic. Heuristics provide shortcuts to a solution that are not necessarily optimal.

The Greedy algorithm computes the decrease in $\pi(z)$ and selects the edge with the largest decrease every time.

Algorithm 4.1 Greedy minimization of $\pi(z)$

INPUTS: Graph $G(V, E)$; k number of edges to add; OUTPUT: Graph G' with k new edges that minimize the polarization index $\pi(z)$

for $i = 1 : k$ do
    $Decrease \leftarrow EmptyList$;
    for each edge in $|V| \times |V| \setminus E$ do
        Compute the decrease of $\pi(z)$ if edge is added to the graph;
        Append the decrease on the $Decrease$ list;
    end for
    Select the edge with the largest decrease from the $Decrease$ list.
    Add this edge to the graph.
end for

The $FirstTopGreedy$ is a greedy heuristic we consider that takes into account changes in the graph structure.

This heuristic is taking the first $K \times K$ items of each set of nodes for each opinion respectively. As mentioned before these two sets are sorted by the external opinion value.

Then a greedy search is performed in this smaller space. This allows the $FirstTopGreedy$ to reduce the amount of time spend searching for the best edge to add by considering opinions that lead to greater reduction.

Algorithm 4.2 First Top Greedy

INPUTS: Graph $G(V, E)$; k number of edges to add;
X, Y, the sorted set of vertices according to each viewpoint ϵ [-1,0] and [0,1] respectively.

OUTPUT: List of k edges that minimize the polarization index $\pi(z)$

```
A ← first k items of X
B ← first k items of Y
for i = 1 : k do
    Decrease ← EmptyList;
    for each edge in |A| × |B|\E do
        Compute the decrease of π(z) if edge is added to the graph;
        Append the decrease on the Decrease list;
    end for
    Select the edge with the largest decrease from the Decrease list.
    Add this edge to the graph.
end for
```

4.2.2 Heuristics that do not consider network changes

The running time of a greedy algorithm is determined by the ease of maintaining an ordering of the candidate choices in each round.

This is accomplished via sorting the candidates. In our case each choice is an edge.

After an addition the network changes. Computing the $\pi(z)$ is an expensive operation due to the computation of the inverse matrix.

Bellow we explore some heuristics that do not consider the network changes and thus can run more easily in larger datasets.

At first we can see a variation of the *Greedy* algorithm. Its implementation is similar to the *Greedy* but without recomputing the $\pi(z)$ at each round.

Algorithm 4.3 Greedy Batch

INPUTS: Graph $G(V, E)$; k number of edges to add; OUTPUT: List of k edges that minimize the polarization index $\pi(z)$

$EdgesToAdd \leftarrow EmptyList$;
for each edge in $|V| \times |V| \backslash E$ do
    Compute $\pi(z)$, the decrease if the edge (u, v) is added;
    Append edge (u, v) to EdgesToAdd;
end for
$Sorted \leftarrow sort(EdgesToAdd)$ by the decrease of $\pi(z)$ by decreasing order;
Return top k from $Sorted$

We continue by using a variation of the $FirstTopGreedy$. The $FirstTopGreedyBatch$ heuristic.

This heuristic does not take into consideration changes in the structure of the graph.

$FirsTopGreedy$ works exactly like $GreedyBatch$ with the difference that only the first $K \times K$ items of each set of nodes will be used. The heuristic is omitted due to the fact that is similar with the previous.

Last we consider two heuristics that choose edges based on the value of the expressed opinion of their nodes.

We can use the absolute distance of these values or their multiplication.

Algorithm 4.4 Expressed opinion (Distance/Multiplication)

INPUTS: Graph $G(V, E)$; k number of edges to add

OUTPUT: List of k edges that minimize the polarization index $\pi(z)$

$EdgesToAdd \leftarrow EmptyList$;
for each edge in $|V| \times |V| \backslash E$ do
    Append to $EddgesToAdd$ the absolute distance (or multiplication) of z values of the nodes of the edge;
end for
if Absolute Distance then
    $Sorted \leftarrow sort(EdgesToAdd)$ by increasing order;
else
    $Sorted \leftarrow sort(EdgesToAdd)$ by decreasing order;
end if
Return top k from $Sorted$

4.3 Including Probabilities in the heuristics

Link prediction is the problem of predicting the existence of a link between two entities in a network in the future. In our setting, social media networks, the entity represents a person. For example the "People you may know" section on Facebook.

Link prediction algorithms are based on how similar two different nodes are, what features they have in common, how are they connected to the rest of the network or how many other nodes are connected to a single node.

Link prediction is also used in recommendation systems and information retrieval.

By using the acceptance probabilities of a link prediction model we can define the expected decrease of the polarization. For computing these probabilities we will use graph embeddings.

4.3.1 Graph Embeddings

A graph embedding is the transformation of the properties of the graphs to a vector or a set of vectors. The embedding will capture the topology of the graph and will consider the relationship between nodes. The embedding will be used to make predictions on the graph.

Machine learning on graphs is limited while vector spaces have a much bigger toolset available. In essence embeddings are compressed representations in a vector that has a smaller dimension.

4.3.2 Word2Vec

At first we have to define Word2Vec. Suppose we have a sentence of words. For a certain task a simple neural network with a single hidden layer is created.

The trained neural network is not actually used for the task that we trained it on. The goal is to learn the weights of the hidden layer. These weights represent the "word vectors".

After giving the neural network a word in the middle of a sentence, it is trained to look for nearby words and pick a random one. The network is going to give the probability for every word in our vocabulary of being inside a window size we set.

The output probabilities are going to relate to how likely it is to find each vocabulary word near our input word. The neural network is trained by feeding it word pairs found in training examples.

The hidden layer of this model is operating as a lookup table. The output of the hidden layer is just the "word vector" for the input word.

The word vector will then get fed to the output layer. The output layer is a softmax regression classifier. Each output neuron will produce an output between 0 and 1, and the sum of all these output values will add up to 1.

If two different words have very similar context then our model needs to output very similar results for these two words.

4.3.3 DeepWalk

After defining Word2Vec we can use its logic in graphs. DeepWalk uses random walks to produce embeddings. The random walk starts in a selected node and then moves to a random neighbour from a current node with certain number of steps. The method consists of three steps.

- Sampling: A graph is sampled with random walks. Authors show that it is sufficient to perform from 32 to 64 random walks from each node.
- Training skip-gram: Random walks are comparable to sentences in word2vec approach. The skip-gram network accepts a node from the random walk a vector as an input and maximizes the probability for predicting neighbour nodes.
- Computing embeddings: Embedding is the output of a hidden layer of the network. The DeepWalk computes embedding for each node in the graph.

DeepWalk method performs the walks randomly and that means that embeddings do not preserve the local neighbourhood. Node2vec approach fixes that [7].

4.3.4 Node2Vec

Node2vec is a modification of DeepWalk with a small difference in the implementation of random walks. There are two parameters introduced, P and Q.

Parameter Q defines how probable is that the random walk will explore the undiscovered part of the graph, while parameter P defines how probable is that the random walk will return to the previous node and retain a locality[8].

4.3.5 Methodology

Our objective is to predict whether there would be a link between 2 unconnected nodes. At first we will find the pairs of nodes that don't have a link between them.

The next step is to label these pairs. This is needed for preparing a training dataset.

The edges that are present in the graph will be labeled as 1 (positive samples) and the unconnected node pairs as 0 (negative samples).

After the labelling we will use the node2vec algorithm to extract node features from the graph. For computing the features of an edge we can add up the features of the nodes of that pair. These features will be trained with a logistic regression model.

After the model is trained we will obtain a dictionary containing the probability of an edge being accepted. The expected decrease can now be defined.

$$E[\pi(z)] = P(u, v) * Val \tag{4.1}$$

Where $P(u, v)$ is the probabilty of u and v forming an edge and Val is a value that can be either the polarization decrease, the absolute distance of the expressed opinions of nodes u and v or the multiplication of their values.

Chapter 5

Experiments

5.1 Datasets

In this section we consider datasets that are separated in two opposing communities. The information about the opinions of each member of this community is known. Thus, we can assign internal opinions -1 and 1 to the nodes depending on their community membership[3]. We consider the following.

1. The Karate dataset, that represents the friendships between the members of a karate club at a US university. This network is split in two equal size polarized communities around two rival karate instructors.

2. The Books dataset, that is a network of US politics books. These books were published near the 2004 presidential election and sold by Amazon. These Books are classified as ”Liberal”, ”Conservative”, or ”Neutral”. There are in total 43 liberal books, 49 conservative and 13 neutral.

3. The Blogs dataset, a network of hyperlinks between online blogs on US politics. Blogs are classified as either Liberal or Conservative.

4. The Elections dataset, this dataset is the network between the Twitter followers of Hillary Clinton and Donald Trump collected in the period 15/12/2016-15/01/2017 – around the time of the 2016 presidential elections. Members of this network are assigned an internal opinion of 1 or -1 based on which one of the two candidates they follow. We took a subsampled portion that has be done by Matakos, et al [3].

5. The beefban dataset, a hashtag that Twitter users used in March 2015 to signal that their posts referred to a decision by the Indian government about the consumption of beef meat in India.

6. The GermanWings dataset, a hashtag that Twitter users used after the crash of GermanWings Flight 9525.

5.2 Dataset statistics

Table 5.1: Stats

Name	# of Nodes	# of Edges	Avg. Degree	$\pi(z)$
Karate	34	78	4.5882	0.33964
books	105	441	8.4	0.43429
beefban	799	6026	15.0839	0.30326
polblogs	1490	16718	22.4403	0.30983
GermanWings	2111	7329	6.9436	0.44479
ClintonTrump	2832	18551	13.1010	0.07582

5.3 A Visualisation of Edge Additions

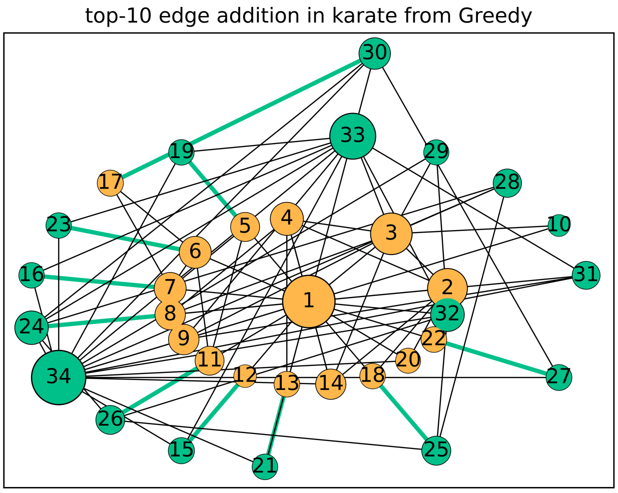

Figure 5.1: the top-10 edges proposed by the greedy algorithm

5.4 Results

All experiments were made with an 2,7 GHz Dual-Core Intel Core i5 on the PyCharm IDE. We can only experiment with the $Karate$ and the $Books$ dataset on all the heuristics. The $Greedy$ algorithm cannot run on the rest of the datasets because they contain thousands of nodes. The $Greedy$ algorithm needs to consider changes in the network structure so it is impossible to compute the polarization so many times.

The same applies for the $GreedyBatch$ algorithm. Although the $GreedyBatch$ algorithm can run on the $beefban$ dataset that contains 799 nodes it would take a lot of time even for the $polblogs$ dataset that has 1490 nodes.

The $FirstTopGreedy$ and the $ExpressedOpinion$ can run in all datasets but provide us with a small decrease in polarization. This decrease can be greater if we consider edge additions that are proportional to the size of the dataset. Greater number of edges would make $FirstTopGreedy$ nonrunnable. The $ExpressedOpinion$ can run in our larger datasets even with a big number of additions.

5.4.1 Experiments with Heuristics

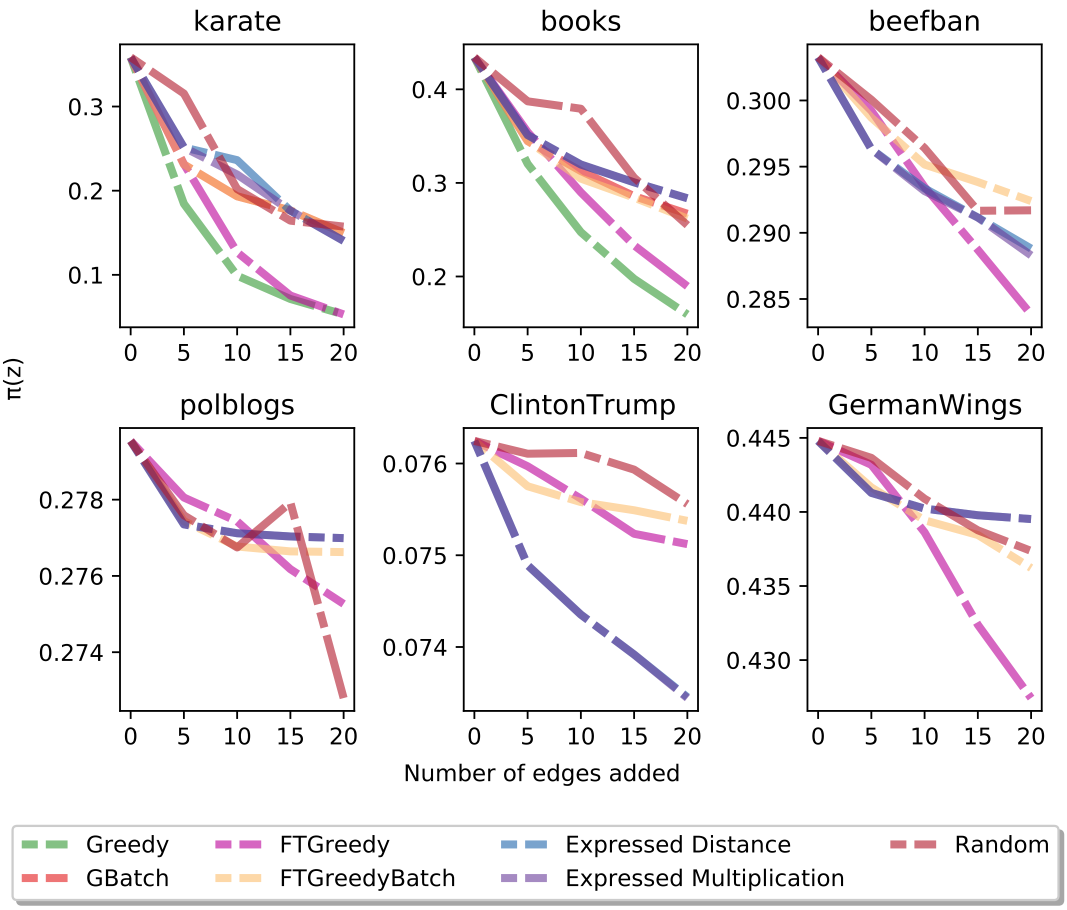

Figure 5.2: Comparison of the heuristics between datasets

We start by applying the heuristics described at 4.2 in these 6 datasets. The algorithms perform as expected. The greedy ones have better results but are expensive in time. The *GreedyBatch* algorithms perform slightly worse but need less time to run. Finally the *Distance*/*Multiplication* algorithms are cheap on time but do not perform well. In almost all cases they are worse than the *Random* addition algorithm. This is due to the fact that they do not take into consideration the structural changes of the graph after an addition.

Their choice is based on the vector of expressed opinions. This vector after some additions will be completely different from what the algorithm knew and thus will make decisions with a false view. The price we pay to recompute this vector is the same as the greedy one.

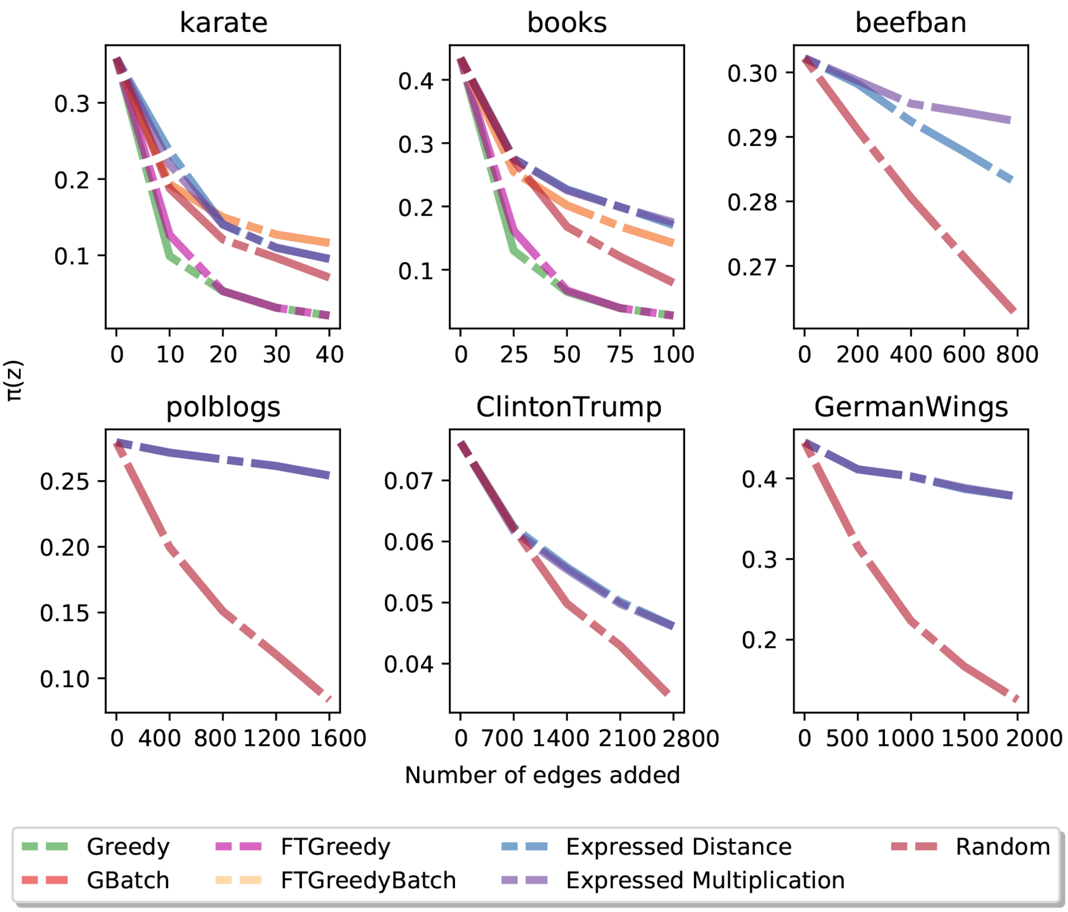

Figure 5.3: Comparison of the heuristics between datasets

This behaviour can be seen more clear when we attempt to add edges proportional to the size of the graph.

The random addition will almost drop to zero and the *Distance/Multiplication* algorithm will still struggle to have a decent decrease. The *Greedy* ones still perform as expected.

We can not run the *Greedy* algorithms in larger datasets due to time limitations.

5.4.2 Experiments by including propabilities

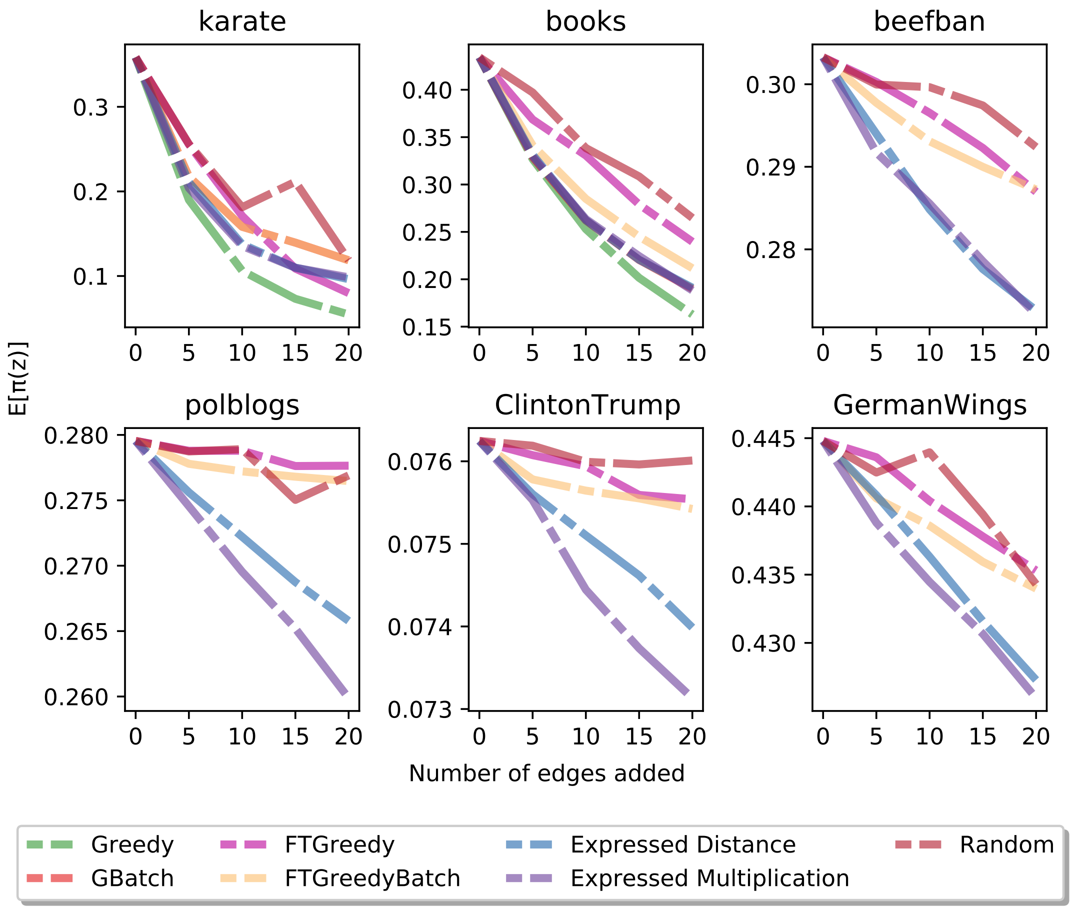

Figure 5.4: Comparison of the Expected Decrease between datasets

Bibliography

[1] Jonas, Schulz-Hardt, Frey, and Thelen, "Confirmation bias in sequential information search after preliminary decisions: an expansion of dissonance theoretical research on selective exposure to information. journal of personality and social psychology," American Psychological Association, 2001.

[2] C. Sunstein, "The law of group polarization." journal of political philosophy, 2002.

[3] Matakos, Terzi, and Tsaparas, "Measuring and moderating opinion polarization in social networks," Data Mining and Knowledge Discovery, vol. 15, 2017.

[4] C. Musco, C. Musco, and Tsourakakis. (2017) Minimizing polarization and disagreement in social networks. [Online]. Available: https://arxiv.org/abs/1712.09948

[5] Chen and D. B. Lijffijt. (2018) Quantifying and minimizing risk of conflict in social networks. [Online]. Available: https://dl.acm.org/doi/proceedings/10.1145/3219819

[6] Bindel, Kleinberg, and Oren. (2015) How bad is forming your own opinion? [Online]. Available: https://arxiv.org/pdf/1203.2973.pdf

[7] S. Perozzi, Al-Rfou. (2014) Deepwalk: Online learning of social representations. [Online]. Available: https://arxiv.org/pdf/1403.6652.pdf

[8] G. Leskovec. (2016) node2vec: Scalable feature learning for networks. [Online]. Available: https://cs.stanford.edu/~jure/pubs/node2vec-kdd16.pdf