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
March 2021

Examining Committee:

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

- Evaggelia Pitoura , Associate Professor, Department of Computer Science and Engineering, University of Ioannina

- Nikos Mamoulis, Associate Professor, Department of Computer Science and Engineering, University of Ioannina

Table of Contents

List of Algorithms

Abstract

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

Users of online communities are receiving biased information that amplify their own viewpoints. This creates a fragmented community and users interact only with individuals that hold the same opinions. In this thesis we use a metric, the polarization index, to measure the polarization of a social graph.

We try to reduce the polarization by connecting individuals. We propose new social connections between different and extreme opinions by following the intuition of the Friedkin-Johnsen model.

Finally, probabilities are adopted to our heuristic algorithms that now make selections based on how probable is the acceptance of a recommendation of a new social connection.

Περιληψη

Γνωρίζουμε ως γεγονός το ότι οι γνώμες διαμορφώνονται μέσω των κοινωνικών συναναστροφών. Οι διαδικτυακές κοινότητες προσφέρουν δημόσια την πρόσβαση σε συζητήσεις για αμφιλεγόμενα ζητήματα που επιτρέπουν την μελέτη αλλά και τον έλεγχο τους. Η πλειοψηφία των ερευνών για τα κοινωνικά δίκτυα βασίζονται στο μοντέλο του Friedkin-Johnsen.

Οι χρήστες των διαδικτυακών κοινοτήτων λαμβάνουν μεροληπτικές πληροφορίες που ενισχύουν την δική τους οπτική. Αυτό δημιουργεί μία κατακερματισμένη κοινότητα και οι χρήστες αλληλεπιδρούν μόνο με άτομα που έχουν τις ίδιες γνώμες με αυτούς. Σε αυτήν την διπλωματική εργασία θα χρησιμοποιήσουμε μια μετρική, τον δείκτη πόλωσης, για να μετρήσουμε το πόσο πολωμένο είναι ένα κοινωνικό γράφημα.

Προσπαθούμε να μειώσουμε την πόλωση με το να συνδέσουμε τα άτομα μεταξύ τους. Προτείνουμε νέες κοινωνικές συνδέσεις μεταξύ ατόμων που έχουν διαφορετικές και ακραίες γνώμες ακολουθώντας τον τρόπο που λειτουργεί το μοντέλο του Friedkin-Johnsen.

Τελικά, ενσωματώνουμε τις πιθανότητες στους ευριστικούς μας αλγορίθμους, που τώρα κάνουν επιλογές με βάση το πόσο πιθανό είναι να γίνει αποδεκτή μία πρόταση για μια νέα κοινωνική σύνδεση.

Chapter 1

Introduction

1.1 Motivation

Real world events such as Brexit and the 2016 U.S. presidential elections give 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.

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. In social media settings, users tend to join communities of like-minded individuals. In these settings the opinions of the users are amplified and reinforced by the continuous communication and recycling of the same view. These communities are referred to as echo chambers. Inside an echo chamber users can easily find information that reinforces their existing opinion without being exposed to opposing views. Echo chambers can be created where information is exchanged, whether it's online or in real life.

On social media almost anyone can quickly find like-minded people and countless news sources. This has made echo chambers far more numerous and easy to fall into. Echo chambers online are referred to as filter bubbles. Filter bubbles are created by algorithms that keep track of the online behaviour of a user such as search histories, shopping activity and many more. Social media will then use those algorithms to show content that is similar to what the user is already aligned with. This can lead users to adopt a more extreme version of their opinions. 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.

Our goal is to decrease the polarization by proposing new social connections. These additions are computed using heuristic algorithms. In a real world setting, new social connections are not always accepted. For example we would not accept friend requests from people we barely know. This is relevant with link prediction. Link prediction is the problem of predicting the existence of a link between two entities in a network in the future. 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 they are 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. For computing these probabilities we will use graph embeddings.

1.2 Thesis Outline

We begin by exploring the Friedkin and Johnsen Model. This model uses the terms of internal and external opinion. By repeated averaging and combining these two values we can get the opinion vector of the graph. This is a vector that contains information about the opinions of the whole network. Then, by using a metric that quantifies the polarization we can get an image of the social graph.

We then proceed and define our two problems. First, we want to find the best k edges that will lead to the greatest reduction of the polarization metric we use. The selection of the edges is mage with heuristics. The second problem incorporates acceptance probabilities. We also observe that the addition of new edges between opposing opinions will not necessary decrease the polarization metric and prove it with a counter-example.

Our heuristics are based on the intuition that the Friedkin-Johnsen model has the biggest decrease when we connect different and extreme opinions. We classify our heuristics in two categories. In these two categories the heuristics do or do not recompute the opinion vector after the addition of an edge. This is derived from the fact that when adding an edge to the network the structure of the graph changes. The heuristics that consider network changes are the $Greedy$ the $FTGreedy$ and the $ExpressedOpinion$. These three are then modified into a batch version that does not consider network changes. We continue by using Graph Embeddings and the $Node2Vec$ algorithm to compute acceptance probabilities. We use these probabilities in a modified version of our heuristics to compute how much we expect the polarization metric to drop. Our heuristics are applied in 6 datasets of various topics and compared with each other. The Greedy heuristics cannot run on graphs that contain a lot of nodes due to time limitations. Chapter 7 concludes in a briefly manner.

1.3 Roadmap

Chapter 2 addresses the related work around polarization and decreasing polarization. We see how polarization is measured, the relation of polarization and random walks and how polarization can be combined with disagreement and conflict. Chapter 3 defines the Friedkin and Johnsen model and the polarization metric we use. Our 2 problems are also defined there, the $k - Addition$ problem and the $k - Addition - Expected$. Then, a counter-example is provided for the monotonicity of the polarization metric we use. Chapter 4 defines our heuristics and chapter 5 presents our datasets and the results of the polarization decrease. Chapter ?? continues by including acceptance probabilities in the heuristics and presenting the results of the experiments that include probabilities. Chapter 6 briefly concludes.

Chapter 2

Related Work

We will now present some work that is related to the work in this thesis.

2.1 Opinion models

How people form their opinions has long been the subject of research in the field of social sciences. Models of opinion formation and dynamics are being used by computer scientists to explore and quantify polarization, conflict and disagreement on social networks. Opinion models study these quantities and how they change by manipulating the opinions and by changing the network structure of a set of nodes of the social graph. Many of these models are modelling the influence that goes with social interaction and the Friedking-Johnsen model is a very popular one.

2.2 Measuring the Polarization of a Network

In this paper the polarization index is defined [1]. The direct link between the Friedkin-Johnsen model and random walks is also explored. 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.

2.3 Polarization and Disagreement

Another way of looking at polarization is by combining it with disagreement [2]. 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.

2.4 Quantifying and Minimizing Risk of Conflict in Social Networks

This paper addresses the main problem in the Friedkin-Johnsen model metrics [3]. 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. When the social graph is edited its is done 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.

2.5 Reducing Controversy by Connecting Opposing Views

Garimella et al. [4] rely 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 graph $G = (V, E)$. 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 controversy.

Garimella et al. also states that real graphs often have a star-like structure. Small number of highly 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 controversy score. Probabilities are also incorporated in the sense that a new edge addition may be not accepted by the user.

2.6 Graph Embeddings and Node2Vec

A graph embedding [5] 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.

Node2vec uses random walks to compute acceptance probabilities. 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.

Chapter 3

Premilinaries and Problem Definition

3.1 The Friedkin and Johnsen Model

The Friedkin-Johnsen model is a very popular opinion dynamic model [6]. This model uses information about the opinion of the user assuming there is an internal and external opinion. The internal opinions cannot change and is the specific opinion of an individual for a certain matter. On the other hand the expressed opinion is influenced by social interactions. The internal opinion of a user corresponds to the views that inherently holds for a controversial topic while the expressed opinion refers to the views that the user shares on a social network with his connections. The internal opinions of a user is denoted as s_i and the expressed opinion as z_i.

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

The expressed opinion z_i is computed as a weighted average of the external opinions of the neighbourhood of the user, for example, the opinions of the users friend list or the opinions of the accounts the user follows. The opinions of the users are stored in a vector z. The opinion vector z 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.

There are two ways of obtaining the z vector of opinions. The first is to use repeated averaging until the model converges. An equivalent way is by computing the following: if L is the laplacian matrix of a graph $G = (V, E)$, and I is the identity matrix, then $z = (L + I)^{-1}S$ [7].

3.2 Measuring the polarization

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-Johnsen. We use the definition of the polarization index [1]. The polarization is measured by its distance from a neutral opinion.

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

To make the polarization index independent of its network we can normalize it by dividing it with the length of the vector z.

3.3 Problem Definition

Problem 1 [k-Addition]. Let $G = (V, E)$ be a connected undirected graph representing a network and k a given number of edges. Let z be the vector of expressed opinions for the whole network and $\pi(z) = ||z||_2^2$ the polarization index of this social graph. Let also $C \subseteq V \times V$ 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 will reduce the polarization index $\pi(z)$.

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.

Problem 2 [K-Addition-Expected]. Let $G = (V, E)$ be a connected undirected graph representing a network and k a given number of edges. Let z be the vector of expressed opinions for the whole network and $\pi(z) = ||z||_2^2$ the polarization index of this social graph. Let also $C \subseteq V \times V$ set of edges that are not in the graph and $P(u, v)$ a probability that the edge addition u, v is accepted. We want to find a subset of $S \subseteq C$ of k edges whose addition to a graph G will show us how much we expect the polarization index $\pi(z)$ to be reduced.

3.4 Monotonicity of the Problem

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

We will show this with a counter example. In the network 3.1 nodes 0, 2 and 3 have a value of $s_i = -1$, and nodes 1 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.

$$(L+I)^{-1}s = z = \begin{pmatrix} \frac{-27}{55} \\ \frac{1}{55} \\ \frac{-5}{11} \\ \frac{-21}{55} \\ \frac{17}{55} \end{pmatrix}, \qquad \pi(z) = 0.13785123966 \tag{3.3}$$

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

$$(L+I)^{-1}s = z = \begin{pmatrix} \frac{-53}{99} \\ \frac{-7}{99} \\ \frac{-5}{11} \\ \frac{-29}{99} \\ \frac{35}{99} \end{pmatrix}, \qquad \pi(z) = 0.14180185695 \tag{3.4}$$

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.

Chapter 4

Algorithms

In this section we consider a greedy algorithm and some heuristics for reducing $\pi(z)$. All the heuristics use the intuition that connecting the most extreme opinions of each community can result in great reduction. When a new edge is introduced, the graph structure changes. This leads to changes in the opinion vector z. The recomputation of the z vector is expensive on time due to the computation of the inverse matrix in the $(L+I)^{-1}S$ formula. This is why we consider two types of algorithms, those that recompute the z vectors and those that do not. All the heuristics run on $\mathcal{O}(n^2)$. This is due to the fact that they need to explore all edge combinations.

4.1 Algorithms that recompute the opinion vector

We begin with a Greedy algorithm. Greedy algorithms work in stages and during each stage a choice is made which is locally optimal. The Greedy algorithm computes the decrease in $\pi(z)$ and selects the edge with the largest decrease every time.

After finding the best edge, the $Greedy$ algorithm adds this edge to the graph. This result in a change of the network structure. Then a recomputation of the z vector is happening and the procedure is repeated.To reduce running times, we use repeated averaging instead of computing the inverse matrix and limit the accuracy of the convergence.

Algorithm 4.1 Greedy

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

OUTPUT: A set S of k edges to be added to G that minimize the polarization index $\pi(z)$

```
for i = 1 : k do
    Compute the opinion vector z
    for each edge in |V| × |V|\E do
        Compute the decrease of π(z) if edge is added to G
    end for
    Select the edge with the largest decrease and add it to G
end for
Return the set of edges that were selected
```

A second heuristic we consider is the $FirstTopGreedy$. Let X be the set of nodes of expressed opinions ϵ [-1,0) sorted by increasing order and Y the set of nodes of expressed opinions ϵ (0,1] sorted by decreasing order. This heuristic use the first k nodes of X and Y, resulting in a $k \times k$ smaller search space. This allows the $FirstTopGreedy$ to reduce the amount of time spend searching for the best edge to add.

Last we consider two heuristics that choose edges based on the value of the expressed opinion of their nodes. The Distance of their opinions can be defined as $D = |z_u - z_v|$. This heuristic computes the distance between every edge candidate and then chooses to add the edge with the maximum distance.

Algorithm 4.2 FirstTopGreedy

INPUT: Graph $G(V, E)$; k number of edges to add;
X, the set of nodes that their expressed opinions ϵ [-1,0) sorted by increasing order
Y, set of nodes that their expressed opinions ϵ (0,1] sorted by decreasing order

OUTPUT: A set S of k edges to be added to G that minimize the polarization index $\pi(z)$

```
A ← first k items of X
B ← first k items of Y
for i = 1 : k do
    Compute the opinion vector z
    for each edge in |A| × |B|\E do
        Compute the decrease of π(z) if edge is added to the graph
    end for
    Select the edge with the largest decrease and add it to G
end for
Return the set of edges that were selected
```

Algorithm 4.3 ExpressedOpinion

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

OUTPUT: A set S of k edges to be added to G that minimize the polarization index $\pi(z)$

for $i = 1 : k$ do
    Compute the opinion vector z
    Compute the z values
    for each edge in $|V| \times |V| \setminus E$ do
        Compute the value $D = |z_u - z_v|$.
    end for
    Sort the distance values by decreasing order
    Add the edge with the biggest distance to G
end for
Return the set of edges that were selected

4.2 Algorithms that do not recompute the opinion vector

We continue by exploring some heuristics that do not consider the network changes and thus can run more easily in larger datasets. We will also refer to them as batch heuristics. Computing the $\pi(z)$ is an expensive operation due to the computation of the inverse matrix. At first we can see a variation of the *Greedy* algorithm. Its implementation is similar to the *Greedy*. We compute the opinion vector only once and we sort the edges according to the decrease. Then we select the top k edges.

We continue by using a variation of the $FirstTopGreedy$. The $FirstTopGreedyBatch$ heuristic. $FirsTopGreedy$ works exactly like $GreedyBatch$ with the difference that the opinion vector is not recomputed. Let X be the set of nodes that their expressed opinions ϵ [-1,0) sorted by increasing order and Y the set of nodes that their expressed opinions ϵ (0,1] sorted by decreasing order. This heuristic is taking the first k nodes of X and Y, resulting in $k \times k$ nodes.

Algorithm 4.4 GreedyBatch

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

OUTPUT: A set S of k edges to be added to G that minimize the polarization index $\pi(z)$

```
Compute the z values
for each edge in |V| × |V|\E do
    Compute the decrease of π(z) if edge is added to G
end for
Sort the values computed by decreasing order;
Select the k edges with the largest decrease and add it to G
Return the set of edges that were selected
```

Algorithm 4.5 FirstTopGreedyBatch

INPUT: Graph $G(V, E)$; k number of edges to add;
X, the set of nodes that their expressed opinions ϵ [-1,0) sorted by increasing order
Y, set of nodes that their expressed opinions ϵ (0,1] sorted by decreasing order

OUTPUT: A set S of k edges to be added to G that minimize the polarization index $\pi(z)$

```
A, B ← first k items of X , Y
Compute the z values
for each edge in |A| × |B|\E do
    Compute the decrease of π(z) if edge is added to the graph
end for
Sort the values computed by decreasing order
Select the k edges with the largest decrease and add it to G
Return the set of edges that were selected
```

4.3 Computing edge probabilities

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 the probabilities of an edge being accepted for every edge. We use the default settings for the $Node2Vec$ algorithm and a 80/20 training/test size for the logistic regression model.

Computing the actual expected decrease in the polarization, and selecting the k best edges is a difficult problem. Our goal is to incorporate the probabilities in the operation of the algorithms. We do not expect that the decrease in the polarization index will improve. We will do this as follows.

Each algorithm computes a value $Val(u, v)$ for each candidate edge, and selects greedily edges with the best value. We will replace this value in the algorithm by $P(u, v) * Val(u, v)$. The quantity $Val(u, v)$ can be either the polarization decrease or the absolute distance of the expressed opinions of nodes u and v. In the case that $Val(u, v)$ is the polarization decrease the product $P(u, v) * Val(u, v)$ corresponds to the expected polarization decrease.

In addition in section 5.5 we measure the median probabilities and in section 5.6 we display the results of the heuristics and edited heuristics together including a new algorithm called $maxProb$ that chooses an edge if it is among the edges with the highest acceptance probabilities.

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[1]. 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 created by Matakos, et al [1].

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: Statistics

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 Experiments

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 $FirstTopGreedy$ and the $ExpressedOpinion$ can run in all datasets and us with a fairly good decrease in polarization compared to the batch algorithms. 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 with big number of additions.

In addition to the heuristics we use two random algorithms. The $Random$, that chooses random edges from all possible combinations and the $RandomDifferent$. The second one chooses random edges between different z opinions. More specifically edges that the multiplication of their expressed opinions is negative ($z_u * z_v < 0$).

We start by applying the heuristics described at 4 in our 6 datasets. The algorithms perform as expected. The greedy ones have better results but are expensive in time. We can not run the $Greedy$ algorithms in larger datasets due to time limitations. In figure 5.1 we can see the charts that compare the decrease of the polarization index of the heuristics.

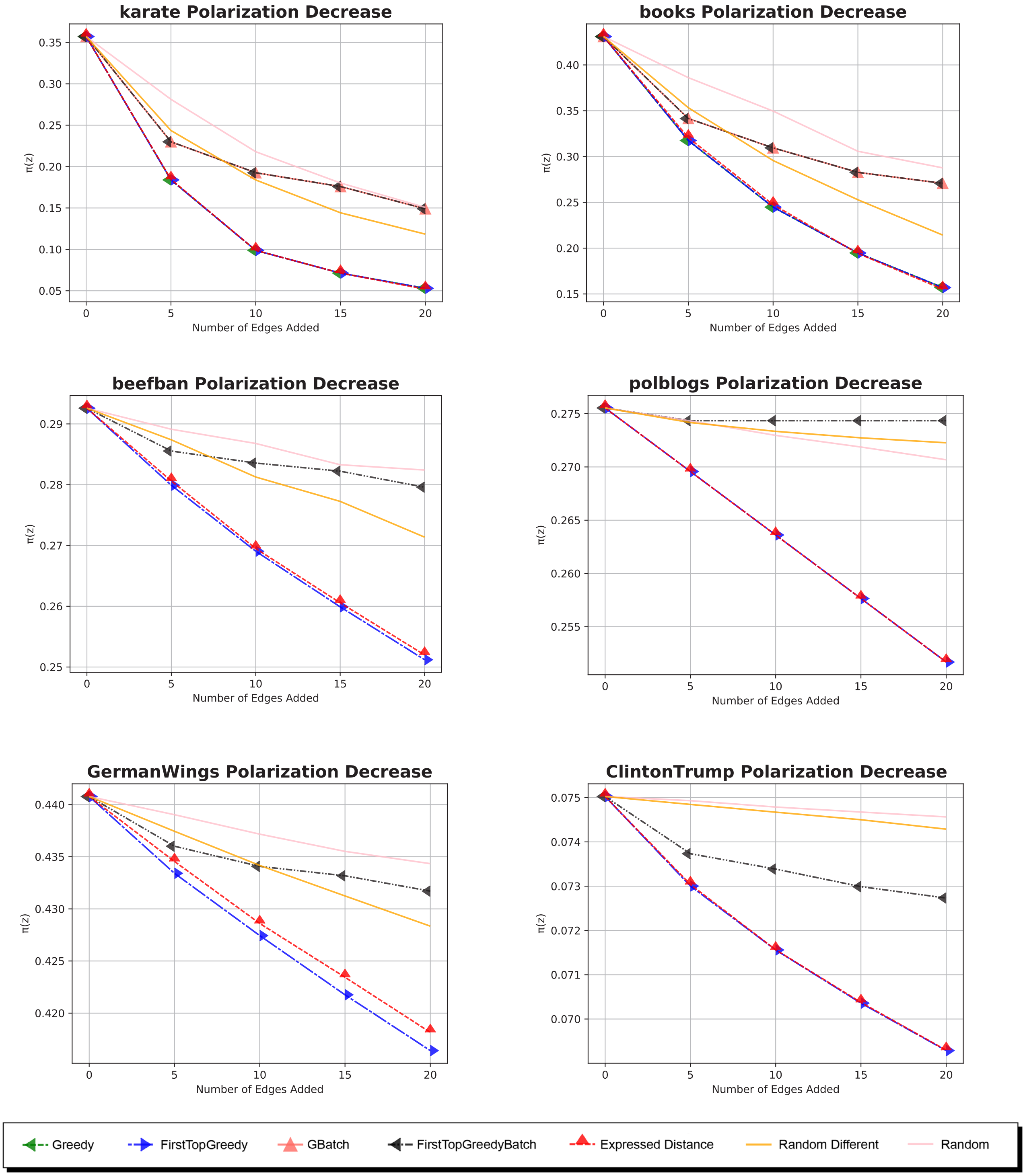


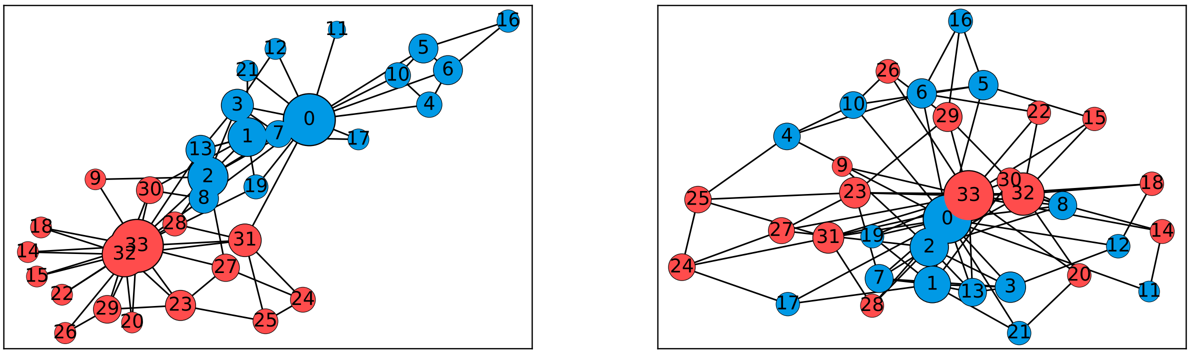
Figure 5.1: Comparison of the heuristics between datasets

- The Expressed Opinion heuristic, that is based on the distance of the opinions, performs very well, similar to *Greedy* and is cheap on time.

- Batch algorithms perform very poorly, even worse than Random. When adding a new edge the Batch algorithms do not recompute the z vector. That means that the heuristic has a false view of the opinions of the network. For example in a batch version of the *ExpressedOpinion* the nodes that have the most extreme opinions of each side are reused even if their value is changed after an addition and are no longer the ones with the most extreme values.

5.4 A Visualisation of Edge Additions

In the figures bellow we can see the karate graph before and after adding the top 10 edges proposed by the *Greedy* algorithm. Blue nodes represent expressed opinions $\epsilon[-1, 0)$, red nodes represent expressed opinions $\epsilon(0, 1]$ and size shows how central a node is. The size has been computed with the help of the pagerank algorithm. The green edges are the additions proposed by the algorithm. We can clearly see that before adding edges of different opinions the network is polarized in two communities.



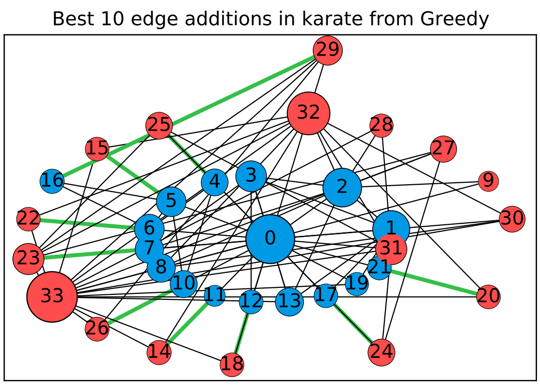
(a) The karate graph

(b) The karate graph after adding 10 edges proposed by *Greedy*

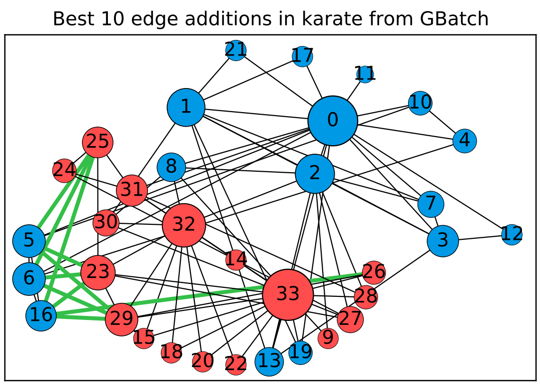
Figure 5.2: Edge addition between opposed opinions.

We can continue by visualising and comparing edges additions between the rest of the algorithms. We observe in the comparison of the choices of $Greedy$ and $GreedyBatch$ that the second reuses the same nodes. This is aligned with the way batch algorithms work. By not recomputing the z vector they will always choose the same nodes thinking they are the best candidates. We proceed to compare the $ExpressedOpinion$ at with $FirstTopGreedy$ and $Greedy$. We can clearly see that the nodes that are being selected are discrete and they are not reused.

To conclude with these visualisations, when we compare heuristics that do not recompute the z vector we will observe that some nodes participate in edges that will be added multiple times. If the heuristics recompute the z vector the nodes will not be reused.

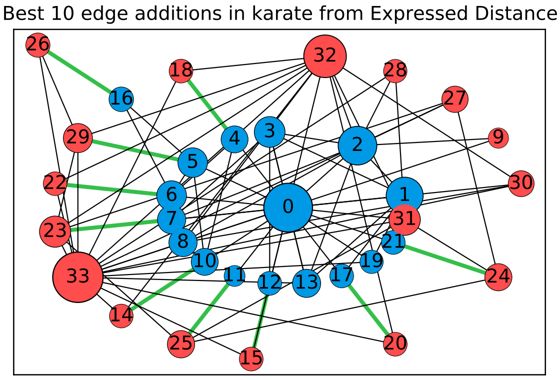


(a) The karate graph after adding 10 edges proposed by $Greedy$

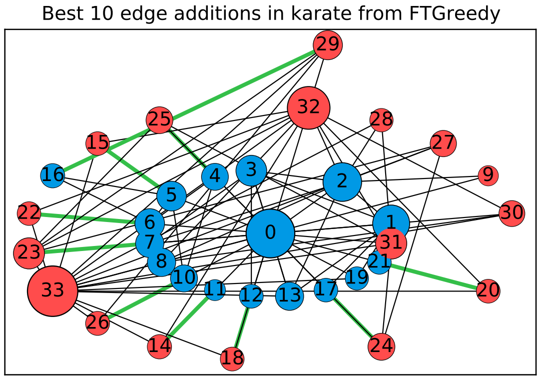


(b) The karate graph after adding 10 edges proposed by $GreedyBatch$

Figure 5.3: Difference of edges between $Greedy$ and $GreedyBatch$

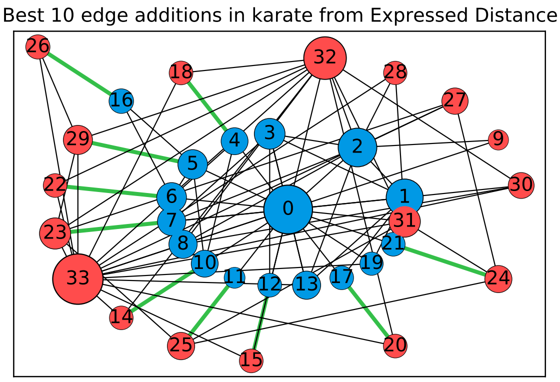


(a) The karate graph after adding 10 edges proposed by *ExpressedOpinion*

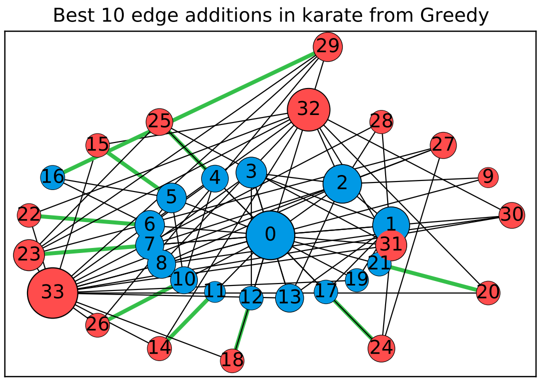


(b) The karate graph after adding 10 edges proposed by *FirstTopGreedy*

Figure 5.4: Difference of edges between *ExpressedOpinion* and *FirstTopGreedy*



(a) The karate graph after adding 10 edges proposed by *ExpressedOpinion*



(b) The karate graph after adding 10 edges proposed by *Greedy*

Figure 5.5: Difference of edges between *ExpressedOpinion* and *Greedy*

5.5 Measuring the median probability

We measure the median probability of the edges selected by the heuristics and compare it to the ones that were edited to include acceptance probabilities. We expect that the edited heuristics will have a higher median value. A new algorithm called $maxProb$ is included that chooses an edge if it is among the edges with the highest acceptance probabilities. This sets an upper limit for the mean probability by having the highest mean probability among all heuristics. This will help us compare the edited heuristics that use acceptance probabilities. We want them to have a relatively higher acceptance probability among the edges they choose. There is a clear increase in the mean probability of the edges the edited heuristics choose.

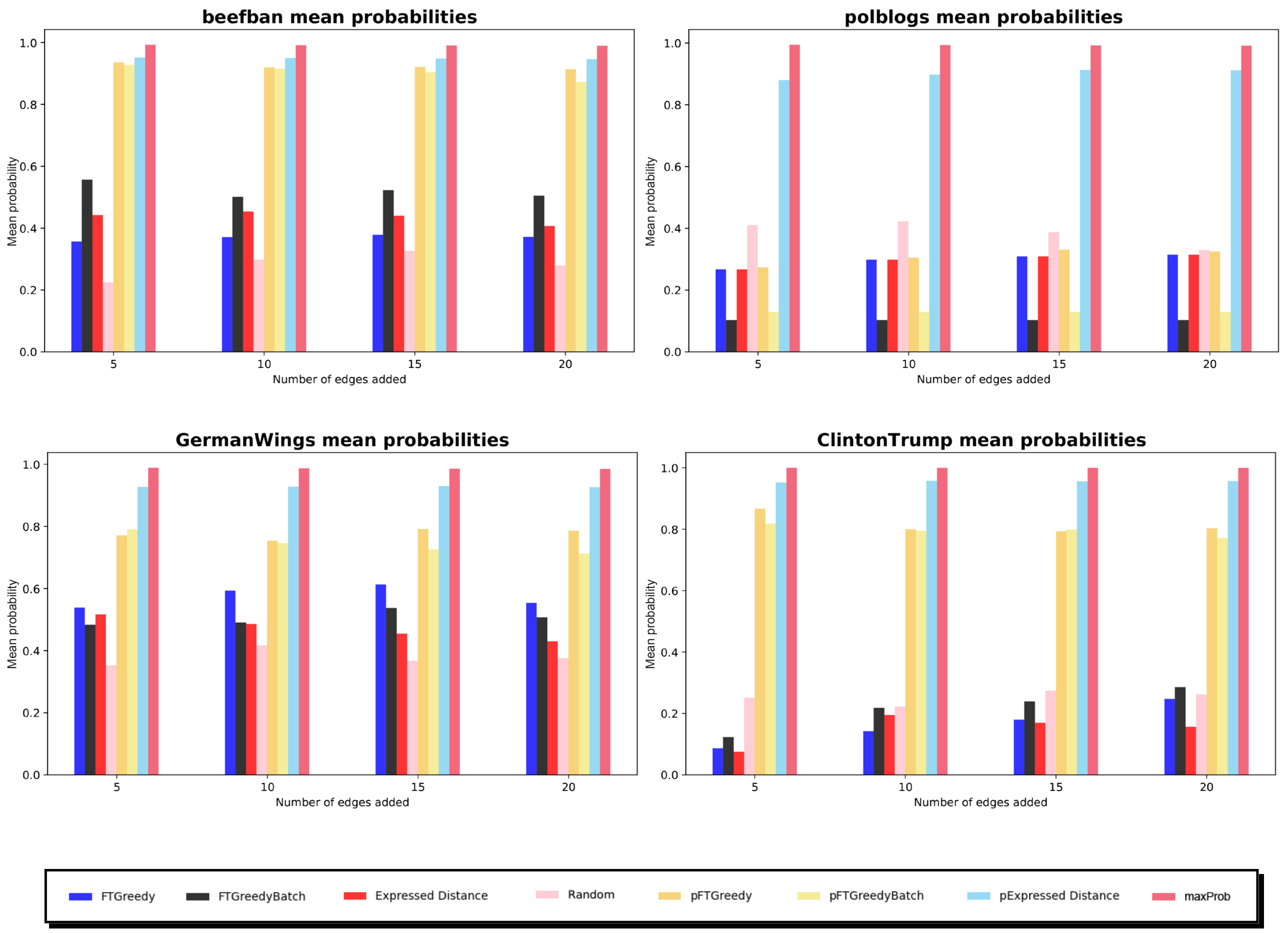


Figure 5.6: Comparison of the median probability

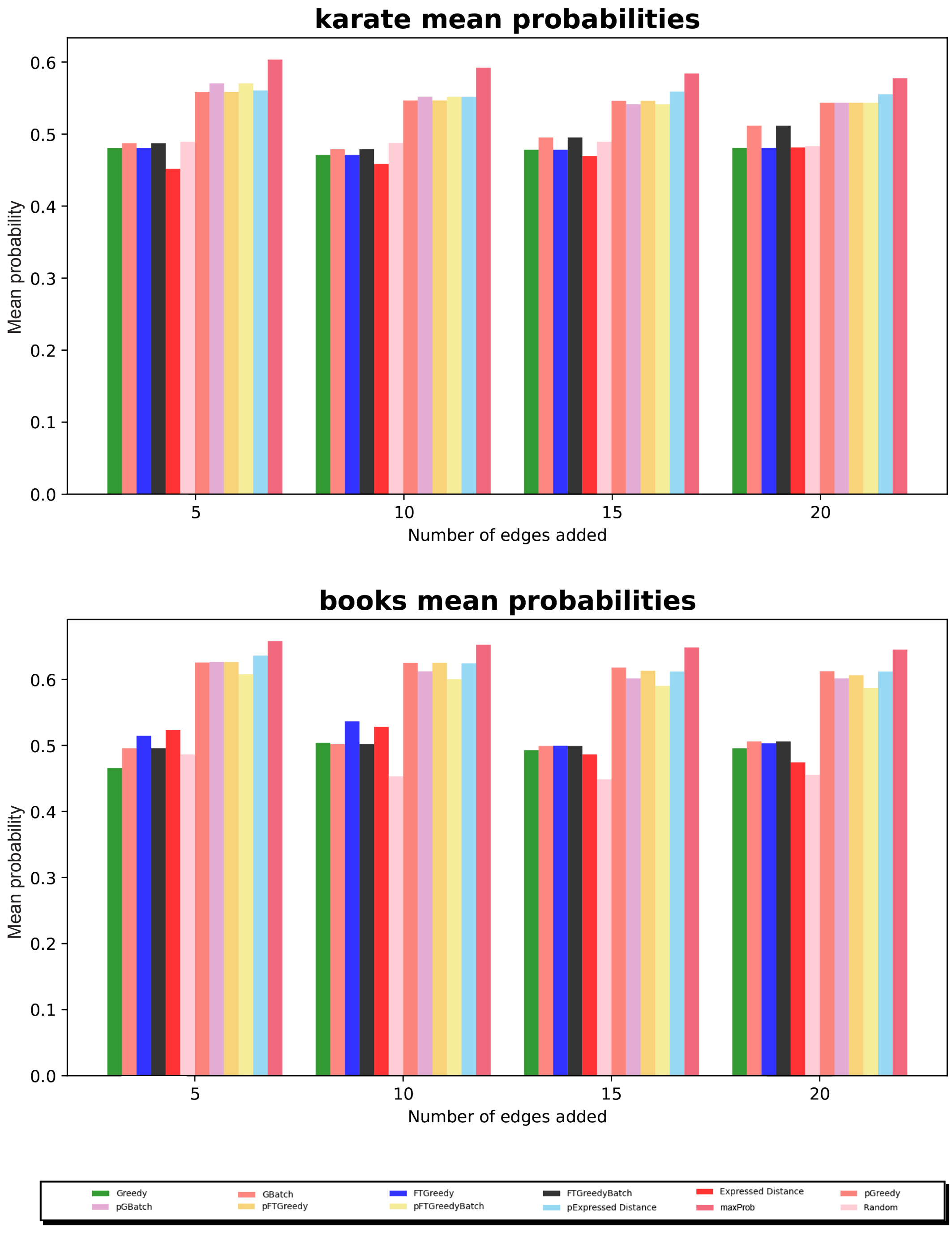


Figure 5.7: Comparison of the median probability

5.6 Comparing all the heuristics

In this section we compare all the algorithms together, $maxProb$ is also displayed in the chart. As mentioned before we want the edited heuristics to have a higher acceptance probability among the edges they choose and this can be seen in the figures 5.7 and 5.6. In these graphs we can see that the polarization is not reduced as much as the original heuristics. This is the tradeoff that the acceptance probabilities create. If we try to reduce the polarization index in a network by not including acceptance probabilities there would be a chance that the decrease would not be as good because individuals could reject these recommendations.

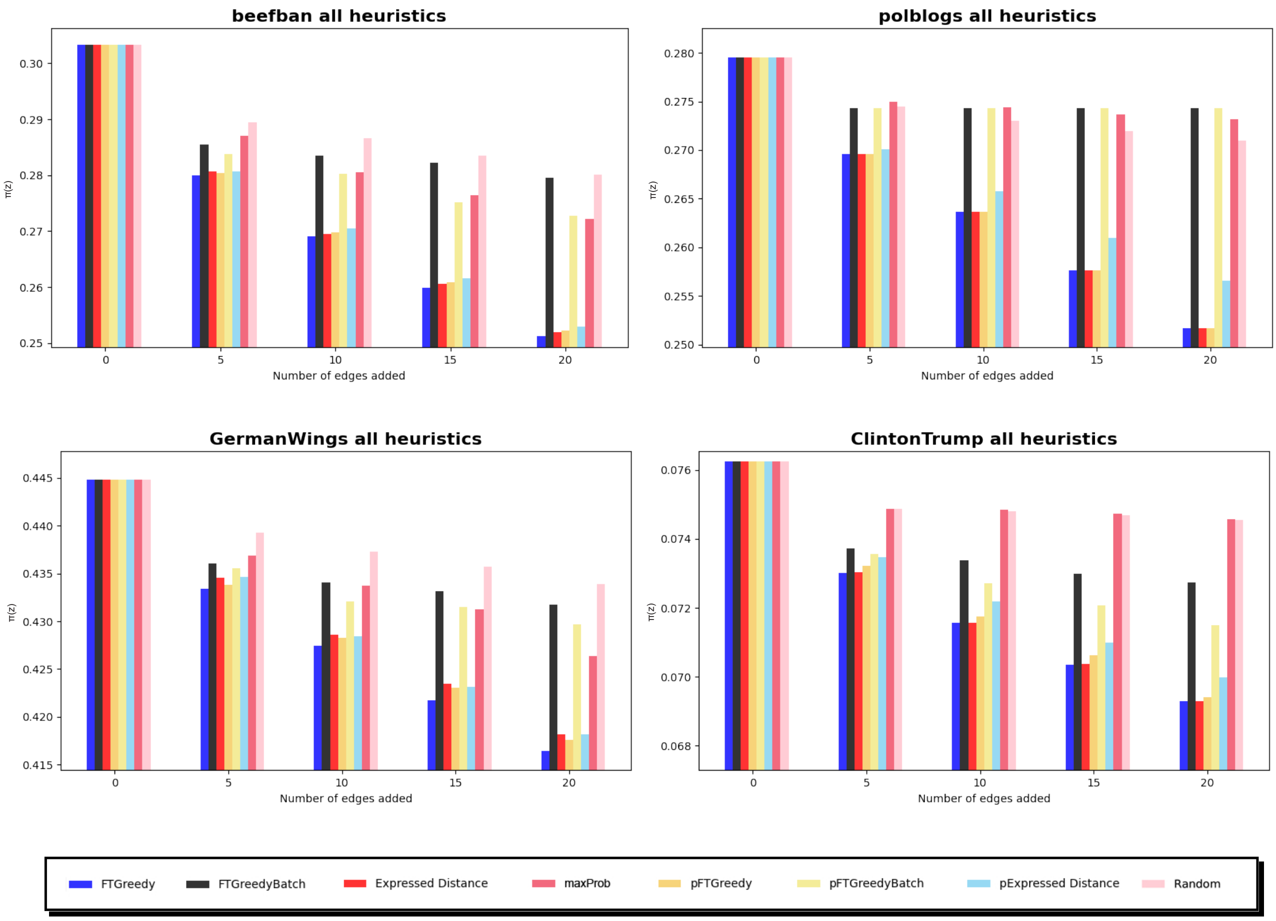


Figure 5.8: Comparison of all the heuristics

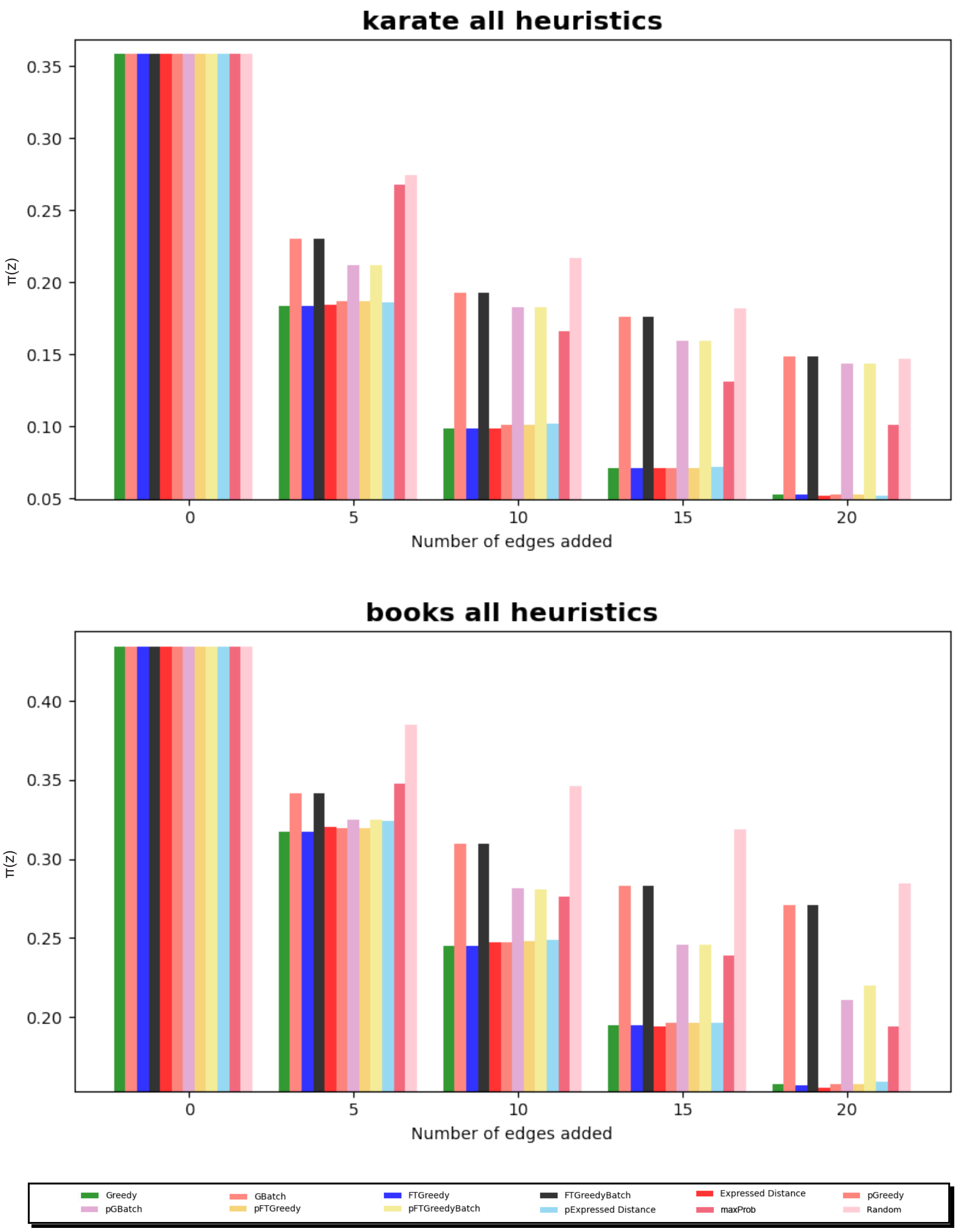


Figure 5.9: Comparison of all the heuristics

Chapter 6

Conclusions

At first we explored some heuristics to reduce the polarization. The $Greedy$ heuristic cannot run in large datasets. We tried to limit the search space by using the $FirstTopGreedy$ heuristic. Even in a smaller space larger datasets need a lot of time to compute the inverse matrix and can only run for a small k. The batch heuristics were used to save time by not recomputing the opinion vector z but perform very poorly and in some cases even worse than the random algorithm. This is derived from the fact that when adding a new edge the opinion vector z changes and without recomputation the batch algorithms will not choose a good candidate for reducing the $\pi(z)$. On the other hand, even though the problem is hard to solve the $ExpressedOpinion$ performs very well with a performance that matches the $Greedy$ algorithm and is also cheap on time. This happens because it chooses to add an edge based on the expressed opinions and not the reduction of $\pi(z)$ if the edge is added, thus, avoiding computing the quantity of the inverse matrix.

We continued by adopting acceptance probabilities in our heuristics. We measured the mean probability of the edges selected by the heuristics that do and do not consider the acceptance probabilities. In this case we wanted to set an upper bound for the acceptance probabilities with the $maxProb$ algorithm and see if the edited heuristics reach it. We then confirmed that the edited heuristics have a higher mean probability when adding an edge.

Finally we compared the reduction of the polarization index with both versions of the heuristics. We observed that the ones that consider acceptance probabilities might have smaller reductions in the $\pi(z)$. This is due to the tradeoff between adding the best candidates to reduce the polarization without knowing if they will be accepted or selecting edges that will most likely be accepted but not have the greatest effect on reducing the $\pi(z)$.

Bibliography

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

[2] C. Musco, C. Musco, and C. E. Tsourakakis, "Minimizing polarization and disagreement in social networks," WWW '18: Proceedings of the 2018 World Wide Web Conference 369-378, 2018.

[3] X. Chen, J. Lijffijt, and T. De Bie, "Quantifying and minimizing risk of conflict in social networks," KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 1197–1205, 2018.

[4] K. Garimella, G. D. F. Morales, A. Gionis, and M. Mathioudakis, "Reducing controversy by connecting opposing views," Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence Best Sister Conferences 5249-5253, 2018.

[5] J. Leskovec and A. Grover, "node2vec: Scalable feature learning for networks," KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 855-864, 2016.

[6] N. E. Friedkin and E. Johnsen, "Social influence and opinions," J Math Soc 15(3–4):193–206, 1990.

[7] D. Bindel, J. Kleinberg, and S. Oren, "How bad is forming your own opinion?" in 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, 2011, pp. 57–66.