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. 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 proposed in [1] for measuring the polarization of a social graph, which relies on the popular Friedkin and Johnsen model.

We try to reduce the polarization by connecting individuals. We propose new social connections between different and extreme opinions, following the intuition of the Friedkin and Johnsen model. We adapt our algorithms to incorporate the probability of acceptance in their selection. We perform experiments with 6 real datasets. We observe that there is a decrease on the polarization when we connect users with opposing views. As the datasets get larger we need to increase the number of edges we add to see a significant decrease. When incorporating probabilities we observe that the decrease is not as great as the previous experiments. 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 polarization.

Περιληψη

Είναι γεγονός ότι οι γνώμες διαμορφώνονται μέσω των κοινωνικών συναναστροφών. Οι διαδικτυακές κοινότητες προσφέρουν δημόσια πρόσβαση σε συζητήσεις για αμφιλεγόμενα ζητήματα, επιτρέποντας την μελέτη αλλά και τον έλεγχο τους. Οι χρήστες των διαδικτυακών κοινοτήτων λαμβάνουν μεροληπτικές πληροφορίες που ενισχύουν την οπτική τους. 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 που προτείνεται στο [1], για να μετρήσουμε το πόσο πολωμένο είναι ένα κοινωνικό γράφημα, σύμφωνα με το δημοφιλές μοντέλο των Friedkin και Johnsen.

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 των Friedkin και Johnsen. Αρχικά, προσαρμόζουμε τους αλγορίθμους μας με σκοπό να ενσωματωσουμε την πιθανότητα αποδοχής στην επιλογή τους. Στη συνέχεια πραγματοποιουμε πειραματα με 6 διαφορετικα σετ δεδομένων. Παρατηρούμε μείωση στη πόλωση όταν συνδέουμε χρήστες με αντιτιθέμενες απόψεις. Όσο τα σετ δεδομένων γίνονται μεγαλύτερα, χρειάζεται να αυξήσουμε τον αριθμό των ακμών που προσθέτουμε, ώστε να διακρίνουμε σημαντική μείωση. Όταν ενσωματώνουμε πιθανότητες παρατηρούμε ότι η μείωση δεν είναι τόσο μεγάλη όσο τα προηγούμενα πειράματα. Αυτό οφείλεται στον συμβιβασμό που θα πρέπει να κάνουμε μεταξύ της προσθήκης των καλύτερων υποψηφίων, ώστε να μειώσουμε τη πόλωση χωρίς να γνωρίζουμε εάν θα γίνουν αποδεκτοί, και της επιλογής ακμών που πιθανότατα θα γίνουν αποδεκτοί, αλλά δεν έχουν τη μεγαλύτερη επίδραση στη μείωση της πόλωσης.

Chapter 1

Introduction

1.1 Motivation and Thesis Goal

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. In a polarized social network, users receive biased information that amplifies 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, users tend to join communities of like-minded individuals and the opinions of the users are amplified and reinforced by the continuous communication and recycling of the same views. These communities are referred to as echo chambers. Echo chambers can be created where information is exchanged, whether it is 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.

An echo chamber leads its members to distrust everybody on the outside of that chamber. This can lead to countless problems on politics, public discourse and poses a threat to the way democracies work. To shield our societies, there is a need for tools for reducing polarization.

1.2 Thesis Contributions

In this thesis, we will consider the problem of reducing polarization by proposing new social connections. For measuring polarization we consider the polarization index measure introduced in [1]. This metric is based on the popular Friedkin and Johnsen model. This model assumes that users have an internal and an expressed opinion, and that the expressed opinion of a node is computed through repeated averaging of her internal opinion and the expressed opinions of their social circle. The polarization index is defined as the measure of the vector of the expressed opinions. The idea is that the closer this measure to zero, the closer the network to neutrality. We then proceed and define two problems. We ask for the best k edges that, if introduced in the network, they will lead to the greatest reduction of the polarization.

The second problem takes into account that in a real social network, new social connections are not always accepted. For example, we would not accept friend requests from people we barely know. We thus assume that every missing edge has some probability to appear. These probabilities can be estimated using link recommendation algorithms. The second problem we consider, $k-expected-addition$, asks for the best k edges that if introduced in the network, they will lead to the greatest expected reduction of the polarization index.

Our heuristics are based on the intuition that the Friedkin and Johnsen model has the biggest polarization 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.

1.3 Roadmap

Chapter 2 addresses the related work around polarization and decreasing polarization. We see how polarization is measured, the relation between polarization and random walks and how polarization can be combined with disagreement and conflict. We also briefly review the work on the $Node2Vec$ algorithm. Chapter 3 defines the Friedkin and Johnsen model and the polarization metric we use. Our two 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. In Chapter 4, we describe our algorithms for both problems. Chapter 5 presents our experiments for the two problems we consider.

Chapter 2

Related Work

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

2.1 Opinion Models and Polarization

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 Friedkin-Johnsen model is a very popular one. The polarization index we will use for measuring polarization relies on the Friedkin and Johnsen model. We describe both the model and the polarization index in Chapter 3.

The polarization index we use in this paper 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 $ModerateInternalproblem$.

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.

Chen et al. [3] addresses the main problem in the Friedkin-Johnsen model metrics. The problem is that the external opinion of a user is hard to measure and the internal opinion is impossible to be known. Another problem occurs in the editing of the social graph. When the social graph is edited to decrease conflict, it 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 et al. [3] 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.

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. 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.

2.2 Graph Embeddings and Node2Vec

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.

A graph embedding [5] is the transformation of the properties of the graphs to a vector or a set of vectors. The embedding captures the topology of the graph and considers the relationships 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 of 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 and Johnsen model is a very popular opinion dynamics model [6]. Each user has an internal and an external opinion. The internal opinion 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 their connections. The internal opinion of a user is denoted as s_i and the expressed opinion as z_i. The expressed opinion z_i is computed as a weighted average of the external opinions of the neighbourhood of the user. Let $G = (V, E)$ be a graph, and for a node i let $N(i)$ be the neighbours of the node. We have that:

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

Let s and z denote the vectors of the internal and external opinions of all the nodes in the graph. 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]. The vector values range from [-1,1]. Values closer to the range limits indicate extreme viewpoints, while values close to zero indicate moderation and neutrality.

3.2 Measuring the Polarization

We use the definition of the polarization index [1]. Let $G = (V, E)$ be a connected undirected graph representing a network. Let z be the vector of expressed opinions for the whole network as computed by the Friedkin and Johnsen model. The polarization index $\pi(z)$ is defined as the measure of the distance of vector z from the vector of all zeros, which corresponds to the neutral opinion. Specifically:

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

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

3.3 Problem Definition

We now define the problems that we will consider in this Thesis.

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 maximizes the reduction of 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 that have a probability of being accepted. For the following we assume that for each missing edge (u, v) we have an estimate of the probability $P(u, v)$ that this edge is accepted if recommended.

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 maximizes the expected reduction of the polarization index.

3.4 Monotonicity of the Problem

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

Proof: 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.

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

We will now add an edge between nodes $1 \to 3$. Note that these nodes have expressed opinions of opposite sign. We recompute the polarization index as follows.

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

We can see an increase of the polarization index after adding this particular edge.

Chapter 4

Algorithms

In this section we consider greedy and heuristic algorithms for the problems we defined in Chapter 3. All the heuristics use the intuition that connecting the most extreme expressed 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.

4.1 Algorithms for the k-Addition Problem

The first algorithm we consider is 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 $\pi(z)$ after adding (u, v) to the graph and selects the edge with the largest decrease every time.

The Greedy algorithm recomputes the z vector for each candidate edge, and also at the beginning of each iteration. To reduce running times, we use repeated averaging instead of computing the inverse matrix and limit the accuracy of the convergence. Since we are interested in the relative order of the edges we do not expect a significant difference. The pseudocode of the algorithm is shown in Algorithm 4.1. The complexity of the algorithm is $\mathcal{O}(k * n^2 * n^3) = \mathcal{O}(n^3)$. k refers to the k edges of the addition problem, n^2 to all the possible edge combinations and n^3 is needed for the inverse of the $L + I$ matrix.

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 $\pi(z)$

1: $S \leftarrow$ empty set
2: for $i = 1 : k$ do
3: Compute the opinion vector z
4: for each edge in $|V| \times |V| \backslash E$ do
5: Compute the decrease of $\pi(z)$ if edge is added to G
6: end for
7: Select the edge with the largest decrease, add it to G and to S
8: end for
9: Return S

The $Greedy$ algorithm is very slow for large, or medium sized datasets. To reduce the running time we propose the $FirstTopGreedy$ algorithm. 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. The algorithm considers the first k nodes of X and Y, resulting in a $k \times k$ search space. This allows the $FirstTopGreedy$ to reduce the amount of time spend searching for the best edge to add. The pseudocode of the algorithm is shown in Algorithm 4.2.

The complexity of the algorithm is $\mathcal{O}(k * k^2 * n^3) = \mathcal{O}(n^3)$. k refers to the k edges of the addition problem, k^2 to the reduced edge combinations space and n^3 is needed for the inverse of the $L + I$ matrix.

Last we consider a heuristic, $ExpressedOpinion$, that chooses edges based on the value of the expressed opinion of their nodes. For a candidate edge (u, v) we compute the distance of the opinions of the endpoints, defined as $D = |z_u - z_v|$. The algorithm computes the distance between every edge candidate and then chooses to add the edge with the maximum distance. The pseudocode of the algorithm is shown in Algorithm 4.3. The complexity of the algorithm is $\mathcal{O}(k * n^2) = \mathcal{O}(n^2)$. k refers to the k edges of the addition problem and n^2 to all the possible edge combinations. There is no need to recompute the inverse of the $L + I$ matrix here, thus we see a decrease in the time complexity.

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 $\pi(z)$

1: $A, B \leftarrow$ first k items of X , Y
2: $S \leftarrow$ empty set
3: for $i = 1 : k$ do
4: Compute the opinion vector z
5: for each edge in $|A| \times |B| \backslash E$ do
6: Compute the decrease of $\pi(z)$ if edge is added to the graph
7: end for
8: Select the edge with the largest decrease, add it to G and to S
9: end for
10: Return S

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 $\pi(z)$

```
S ← empty set
for i = 1 : k do
    Compute the opinion vector z
    for each edge in |V| × |V|\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 and to S
end for
Return S
```

Computing the reduction of $\pi(z)$ for each candidate edge at each iteration is expensive even for medium sized graphs. We will now consider variants of the algorithms we described that compute the reduction only once, and sort the edges according to this value and select the top-k edges. We will refer to them as batch algorithms.

At first we can see a variation of the $Greedy$ algorithm, the $GreedyBatch$. Its implementation is similar to the $Greedy$. The pseudocode of the algorithm is shown in Algorithm 4.4. The complexity of the algorithm is $\mathcal{O}(n^2 * n^3) = \mathcal{O}(n^3)$. We continue by using a variation of the $FirstTopGreedy$, the $FirstTopGreedyBatch$, in a similar manner. The pseudocode of the algorithm is shown in Algorithm 4.5. The complexity of the algorithm is $\mathcal{O}(k^2 * n^3) = \mathcal{O}(n^3)$.

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 $\pi(z)$

1: $S \leftarrow$ empty set
2: Compute the z values
3: for each edge in $|V| \times |V| \backslash E$ do
4: Compute the decrease of $\pi(z)$ if edge is added to G
5: end for
6: Sort the values computed by decreasing order;
7: Select the k edges with the largest decrease, add it to G and S
8: Return S

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 $\pi(z)$

1: $A, B \leftarrow$ first k items of X , Y
2: $S \leftarrow$ empty set
3: Compute the z values
4: for each edge in $|A| \times |B| \backslash E$ do
5: Compute the decrease of $\pi(z)$ if edge is added to the graph
6: end for
7: Sort the values computed by decreasing order
8: Select the k edges with the largest decrease, add it to G and to S
9: Return S

4.2 Algorithms for the k-Addition-Expected Problem

For the $k - Addition - Expected$ problem we assume that for every candidate edge (u, v) we have computed a probability $P(u, v)$ of the edge being accepted as a recommendation. We want to maximize the expected reduction in the polarization index. Computing the actual expected decrease in the polarization, and selecting the k best edges is a difficult problem. We thus design heuristics that incorporate the probabilities in the operation of the algorithms we described before.

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. The complexity of the algorithms stay the same. These algorithms will have the following names: $pGreedy$, $pFirstTopGreedy$, $pExpressedOpinion$, $pGreedyBatch$, $pFirstTopGreedyBatch$. We will refer to them as edited algorithms.

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.

We present some statistics for the datasets in the Table 5.1.

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

5.2 Experiments

We now present experiments with our algorithms for the $k - Addition$ problem. 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 can experiment with all algorithms only for the $Karate$ and the $Books$ datasets. The $Greedy$ algorithm, that recomputes $\pi(z)$ for every candidate edge cannot run on the rest of the datasets that contain thousands of nodes.The $FirstTopGreedy$ and the $ExpressedOpinion$ run in all datasets.

In figure 5.1 we plot the decrease of the polarization index of the heuristics for all our algorithms. We make the following observations.

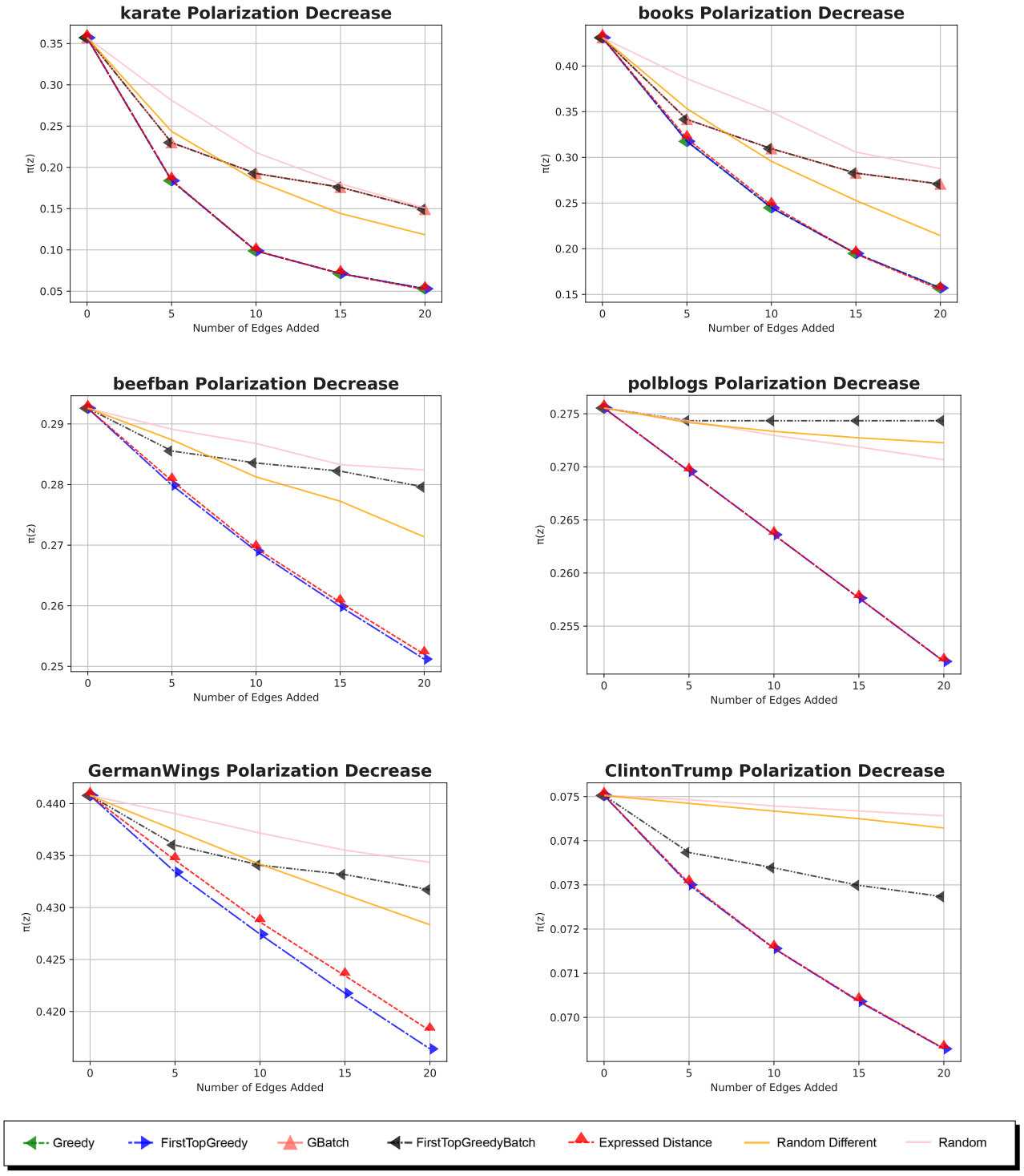

Figure 5.1: Comparison of the heuristics between datasets

The Expressed Opinion heuristic, that is based on the distance of the opinions, performs very well. It achieves a decrease very close to $Greedy$ while being significantly faster. The $FirstTopGreedy$ does not perform well for medium and large datasets. Even with a small k there is a noticeable amount needed for the algorithm to run. When increasing the k the $FirstTopGreedy$ cannot run due to time limitations. 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.

To better understand the behaviour of our algorithms, we will visualize the edge additions. In figure 5.2 we can see the karate graph before and after adding the top 10 edges proposed by all the algorithms. 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 observe that the choices of $Greedy$ and $GreedyBatch$ are different. $GreedyBatch$ 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. 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.

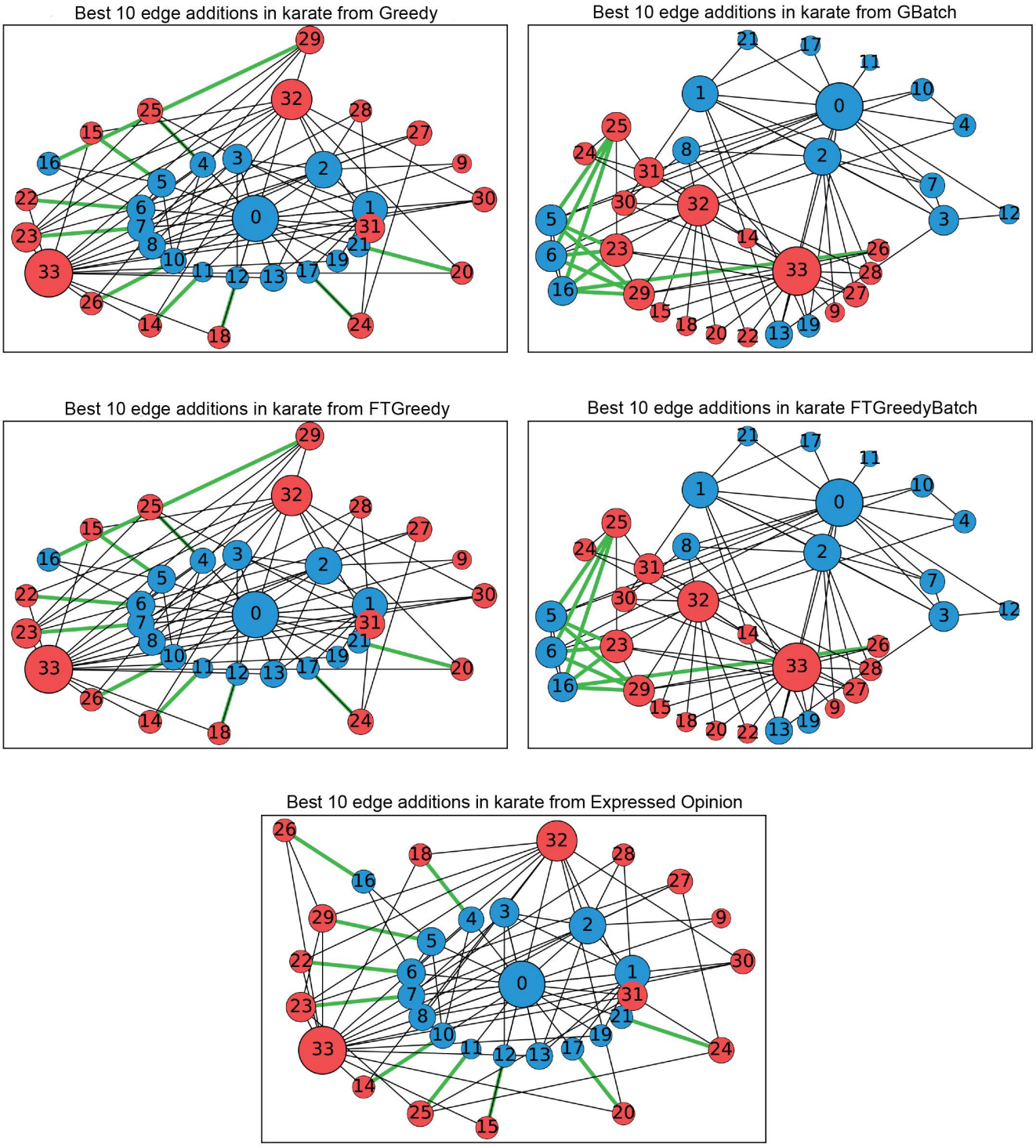

Figure 5.2: Difference of edge additions between the algorithms

5.3 Measuring the median probability

In these experiments we also measure the median probability of the edges selected by the heuristics and compare it to the ones that were edited to include acceptance probabilities.

5.3.1 Estimating Acceptance 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.

We also consider a new algorithm called $MaxProb$ that chooses the top-k edges with the highest acceptance probabilities. This defines an upper bound for the mean probability and we will use it to 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.

5.3.2 Experimental Results

In figure 5.3 we show the mean acceptance probability for all our algorithms, the ones that incorporate probabilities (edited heuristics) and the ones that do not, as well as $MaxProb$. The algorithms have small probability compared to the upper bound, the edited algorithms are better and pretty close to the $MaxProb$. The $ExpressedDistance$, that is fast, displays good performance.

We also compare all the algorithms and $MaxProb$ with respect to the polarization decrease they achieve. In Figure 5.4 we see the polarization index reduction for all the algorithms. In this graph 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 good because individuals might reject these recommendations.

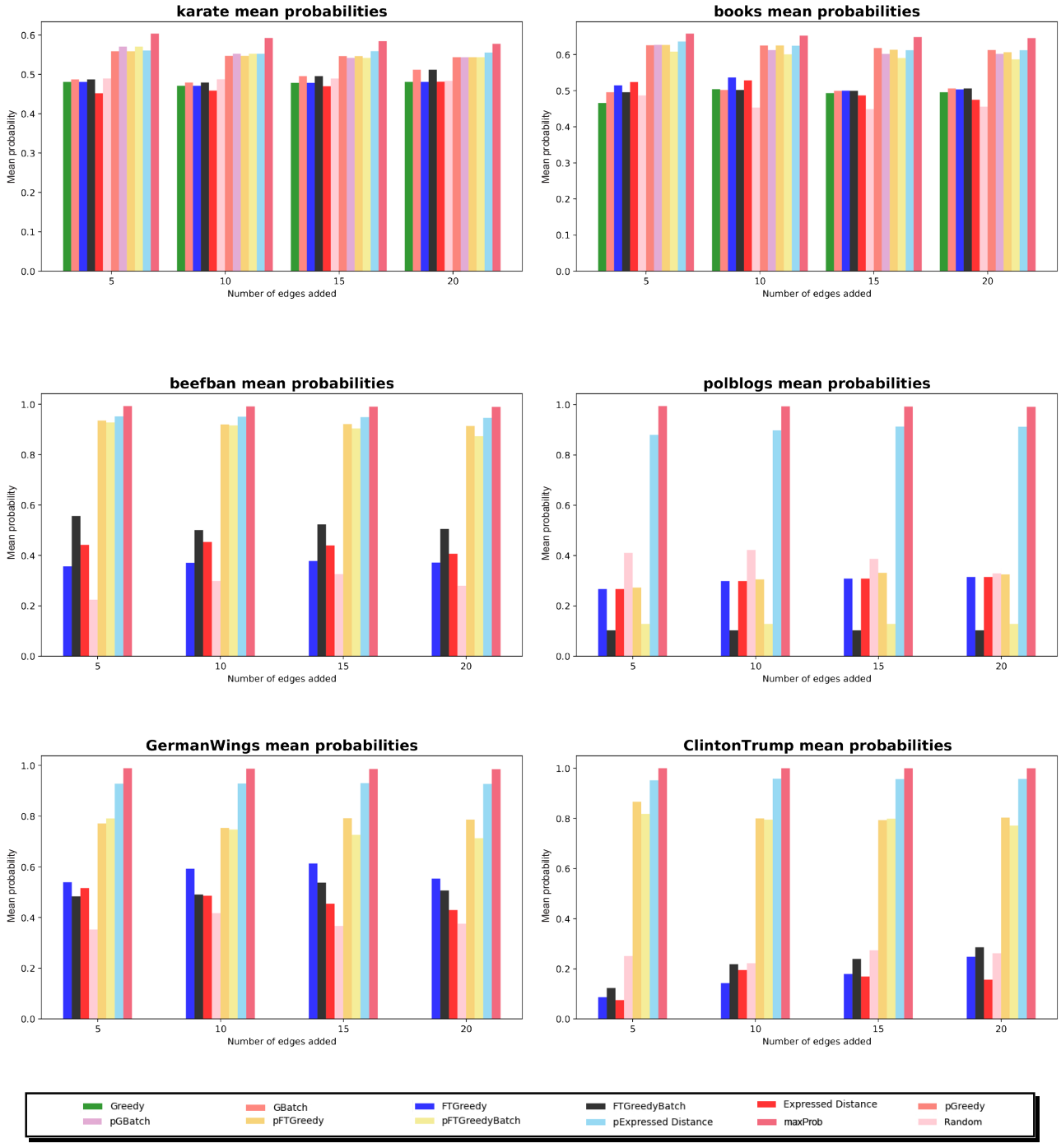

Figure 5.3: Comparison of the mean probability

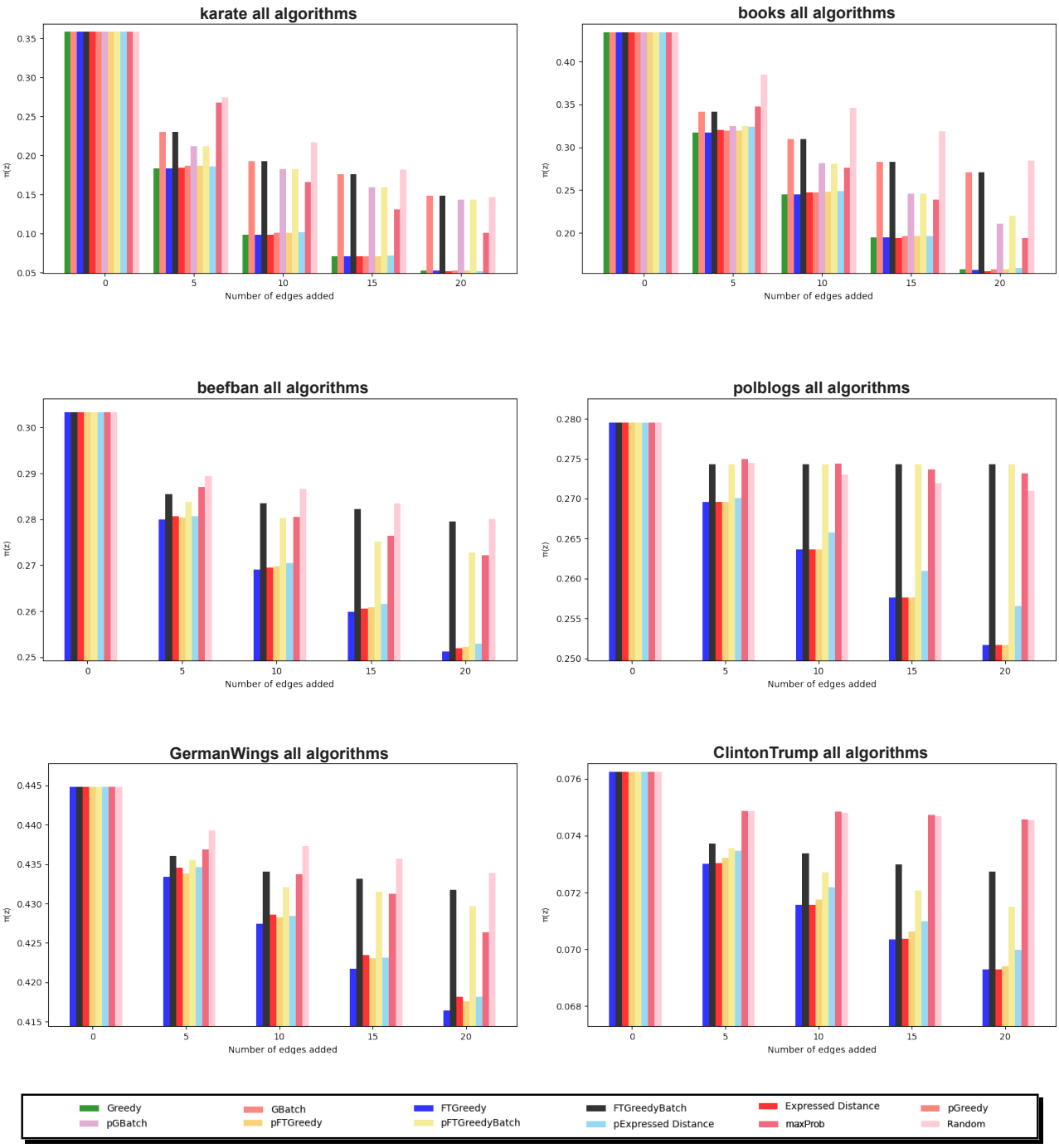

Figure 5.4: Polarization index reduction

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, a lot of time is needed to compute the inverse matrix for medium and large datasets. In addition the algorithm 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.