Fairness Aware Ranking & Recommendations in Networks

A Thesis

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

by

Sotirios Tsioutsiouliklis

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN DATA AND COMPUTER
SYSTEMS ENGINEERING

WITH SPECIALIZATION
IN DATA SCIENCE AND ENGINEERING

University of Ioannina
September 2020

Examining Committee:

- **Evaggelia Pitoura**, Professor, Department of Computer Science and Engineering, University of Ioannina (Advisor)
- **Panayiotis Tsaparas**, Associate Professor, Department of Computer Science and Engineering, University of Ioannina
- **Nikolaos Mamoulis**, Professor, Department of Computer Science and Engineering, University of Ioannina

Dedication

To my family.

Acknowledgements

I have to start by expressing my gratitude to my supervisor Prof. Evaggelia Pitoura for the guidance, the support and the valuable remarks and advices she provided my during the completion of this thesis. Furthemore, I would like to thank my co-supervisor Prof. Panayiotis Tsaparas for his significant contributions and the excellent collaboration. Last, I want to thank Prof. Nikolaos Mamoulis for being always willing to help me with his useful insights.

Special thanks to my friend and my fellow student Ilias Kleftakis for his remarks and his contribution in the implementation of the algorithms.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Sotirios Tsioutsiouliklis, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, September 2020.
Fairness Aware Ranking & Recommendations in Networks.
Advisor: Evaggelia Pitoura, Professor.

Algorithmic fairness has attracted significant attention in the past years. Surprisingly, there is little work on fairness in networks. In this work, we consider fairness for link analysis algorithms and in particular for the celebrated PageRank algorithm. We provide definitions for fairness, and propose two approaches for achieving fairness. Furthermore, we explore how a recommendation system can affect the fairness of a network. We define objective for a fair recommender and we propose two recommendation policies in this direction. We present experiments with real and synthetic graphs that examine the fairness of PageRank, demonstrate qualitatively and quantitatively the properties of our fair algorithms and evaluate the impact of the different recommendation systems.

Εκτεταμενη Περιληψη

Σωτήριος Τσιουτσιουλικλής, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, Σεπτέμβριος 2020.
Αλγόριθμοι Κατάταξης και Συστήματα Συστάσεων σε Κοινωνικά Δίκτυα Ενάντια στις Διακρίσεις.
Επιβλέπων: Ευαγγελία Πιτουρά, Καθηγήτρια.

Στην εποχή μας, λόγο του συνεχούς αυξανόμενου όγκου των δεδομένων προς επεξεργασία, χρησιμοποιούνται καθημερινά συτήματα και αλγόριθμοι για την ολοκλήρωση διάφορων διαδικασιών που μέχρι πρόσφατα διεξάγονταν από ανθρώπους. Συνήθεις διαδικασίες τέτοιων αλγορίθμων είναι η κατάταξη και η κατηγοριοποίηση των δεδομένων. Η εφαρμογή τέτοιων αλγορίθμων σε διαδικασίες που σχετίζονται με ανθρώπους (π.χ. 10 καλύτεροι ερευνητές για το 2020) είχαν ως αποτέλεσμα την εμφάνιση του ζητήματος των άκριτων διακρίσεων διαφόρων μορφών (π.χ. φυλετικές διακρίσεις) και της άνισης μεταχείρησης ανθρώπων από αλγορίθμους. Παρ' ότι το φαινόμενο έχει απασχολήσει την ερευνητική κοινότητα σε διάφορες κατηγορίες αλγορίθμων, όπως αυτών της μηχανικής μάθησης, και τα δίκτυα χρησιμοποιούντε στη μοντελοποίηση πληθώρας καθημερινών καταστάσεων και προβλημάτων, ύπαρχει ελάχιστη δραστηριότητα προς αυτή τη κατεύθυνση στον τομέα των αλγορίθμων δικτύων.

Σε αυτη την εργασία επιχειρούμε μια προσέγγιση στη καταπολέμιση των διακρίσεων σε αλγορίθμους που δρουν σε δίκτυα. Αρχικά, ορίζουμε τις έννοιες της δικαιοσύνης και του δίκαιου αλγορίθμου για δίκτυα. Επικεντρωνόμαστε στον δημοφιλή αλγόριθμο PageRank (Αν και η ανάλυση και οι αλγόριθμοι μπορούν να επεκταθούν κατά φυσικό τρόπο σε διάφορους άλλους αλγορίθμους για δίκτυα) και σε δυαδικά προστατυεόμενα χαρακτηριστικά (π.χ. άντρας - γυναίκα), μελετάμε τις ιδιότητες του δικτύου που το κάνουν άδικο και προτείνουμε διαφορετικές προσεγγίσεις προς τη παραγωγή ενός δίκαιου αποτελέσματος διατηρώντας παράλληλα εκείνα τα χαρακτηριστικά του αρχικού αλγορίθμου που τον ξεχωρίζουν και του προσδίδουν ιδιαίτερη αξία. Η πρώτη προσέγγιση χρησιμοποιεί τον διάνυσμα "άλματος" του PageRank για την επίτευξη ενός δίκαιου αποτελέσματος, ενώ η δεύτερη επιχειρεί μέσο της ατομικής συμπεριφοράς κάθε κόμβου αναγκάζοντας τον, κατά κάποιον τρόπο, να λειτουργήσει δίκαια. Επίσης, αξιολογούμε τους διαφορετικούς αλγορίθμους βάση της αλλαγής που φέρνουν σε σύγκριση με τον PageRank και τη χρησιμότητα τους. Οι αλγόριθμοι που προτείνουμε κλιμακώνουν αποδοτικά σε δεδομένα ευρείας κλίμακας.

Στη συνέχεια εξετάζουμε την επιρροή των συστημάτων συστάσεων συνδέσμων στη δικαιωσυνή ενός δικτύου. Παρατηρούμε ότι τα εώς τώρα συστήματα συστάσεων δεν επιρεάζουν το δύκτιο σε αυτή τη παράμετρο, παρά διατηρούν την αρχική κατάσταση. Προτείνουμε ένα σύστημα συστάσεων που επιτυγχάνει την ανάδειξη και προβολή της αδικημένης/προστατευόμενης κατηγορίας στο δίκτυο με εξερετικά αποτελέσματα, θυσιάζοντας όμως τη ποιότητα των συστάσεων. Διατηρούμε το σκορ που παράγεται από το σύστημα αυτό και το εφραμόζουμε σε μια υβριδική μορφή σε συνδιασμό με ένα υπάρχον σύστημα συστάσεων. Για την πειραματική αξιολόγηση του συστήματος χρησιμοποιούμε ένα σύστημα συστάσεων βασισμένο σε embeddings προερχόμενα από τον node2vec αλγόριθμο και παρατηρούμε ότι το υβριδικό σύστημα ισορροπεί με ικανοποιητικό τρόπο τους δύο αντικειμενικούς στόχους μας (ανάδειξη της αδικημένης κατηγορίας και διατήρηση ποιοτικών συστάσεων). Επιπλέον, εξετάζουμε σε συνθετικά δίκτυα την συμπεριφορά των διαφόρων συστημάτων για διαφορετικές παραμέτρους και βλέπουμε ότι το προτεινόμενο σύστημα δεν επιρεάζεται από τα χαρακτηριστικά του δικτύου και συνεχίζει να έχει όμοια αποτελέσματα. Τέλος, μελετάμε τα ποιοτικά χαρακτηριστικά των συστάσεων όλων τον αλγορίθμων και προσπαθούμε να εξηγήσουμε το σύστημα συστάσεων μέσα από απλά χαρακτηριστικά των προτεινόμενων συστάσεων.

Chapter 1

Introduction

Today, algorithmic systems driven by large amounts of data are increasingly being used in all aspects of life. Often, such systems are being used to assist, or, even replace human decision making. This increased dependence on algorithms has given rise to the field of algorithmic fairness, where the goal is to ensure that algorithms do not exhibit biases towards specific individuals, or groups of users (see e.g., [1] for a survey). We also live in a connected world where networks, be it, social, communication, interaction, or cooperation networks, play a central role. However, surprisingly, fairness in networks has received less attention.

Link analysis algorithms, such as Pagerank [2], HITS [3], or SALSA [4], take a graph as input and use the structure of the graph to determine the relative importance of its nodes. The output of the algorithms is a numerical weight for each node that reflects its importance. The weights are used to produce an ordering of the nodes and as input features in a variety of machine learning algorithms including classification [5], and search result ranking [2]. In this work, we focus on the Pagerank algorithm [2]. Pagerank performs a random walk on the input graph, and ranks the nodes according to the stationary probability of this walk. At every step, the random walk restarts with probability c. The restart node is selected according to a"jump" distribution vector **v**. Since its introduction in the Google search engine, Pagerank has been the cornerstone algorithm in several applications (see, e.g., [6]).

1.1 Thesis Scope

As in previous research, we view fairness as lack of discrimination against a protected group defined by the value of a sensitive attribute, such as, gender, or race [1]. We operationalize this view by saying that a link analysis algorithm is *ϕ-fair*, if the fraction of the total weight allocated to the members of the protected group is ϕ. The value of ϕ is a parameter that can be used to implement different fairness policies. In the simplest case, ϕ is set equal to the fraction of the protected nodes in the graph, asking that these nodes have a share in the weights proportional to their share in the population. We also consider *targeted* fairness, where we focus on a specific subset of nodes to which we want to allocate weight in a fair manner.

We revisit Pagerank through the lens of our fairness definition, and we consider the problem of defining Pagerank variants that are fair. We also define the *utility loss* of a fair algorithm as the difference between its output and the output of the Pagerank algorithm, and we pose the problem of achieving fairness while minimizing utility. We consider two approaches for achieving fairness. Our first approach, the *fairness-sensitive* Pagerank algorithm, exploits the jump vector **v**. There has been a lot of work on modifying the jump vector to obtain variants of pagerank biased towards a specific set of nodes, for example, in personalized pagerank, all jump probability is assigned to a single node, while in topic-sensitive pagerank, the probability is assigned to nodes of a specific topic [7]. In this thesis, we take the novel approach of using the jump vector to achieve ϕ-fairness. We determine the conditions under which this is feasible and formulate the problem of finding the jump vector that achieves ϕ-fairness while minimizing utility loss from the original PageRank as a convex optimization problem.

Our second approach takes a microscopic view by looking at the behavior of each individual node in the graph. Implicitly, a link analysis algorithm assumes that links in the graph correspond to endorsements between the nodes. Therefore, we can view each node, as an agent that *endorses* (or *votes for*) the nodes that it links to. The link analysis algorithm defines a process that takes these individual actions of the nodes and transforms them into a global weighting of the nodes. To this end, we introduce, the *locally fair PageRank algorithms*, where each individual node acts fairly by distributing its own PageRank to the protected and non-protected groups according to the fairness ratio ϕ. Local fairness defines a dynamic process that can be viewed as a *fair random walk*, where *at each step* of the random walk (not only at convergence), the probability of being at a node of the protected group is ϕ.

In our first locally fair PageRank algorithm, termed the *neighborhood locally fair* PageRank algorithm, each node distributes its PageRank fairly among its immediate neighbors, allocating a fraction ϕ to the neighbors in the protected group, and $1 - \phi$ to the neighbors in the non-protected group. Or, in random walk terms, at each node the probability of transitioning to a neighbor in the protected group is ϕ and the probability of transitioning to a non-protected neighbor is $1 - \phi$. The *residual-based locally fair* pagerank algorithms generalizes this idea. Consider a node i that has less neighbors in the protected group than ϕ. The node distributes an equal portion of its pagerank to each of its neighbors and a residual portion $\delta(i)$ to members in the protected group but not necessarily in its neighborhood. Or, in random walk terms, at each node i, the probability of transitioning to a neighbor is $1 - \delta(i)$ and the probability of transitioning to a node in the protected group is $\delta(i)$. The residual is allocated based on *a residual redistribution policy*, which allows us to control the fairness policy. In this thesis, we use the residual redistribution policy to minimize the utility loss.

Finally, we present a post-processing approach that given the output of a link analysis algorithm, it redistributes the weights so as to attain fairness. This gives us a lower bound on the utility loss.

We study the fairness of the original PageRank in both real and synthetic networks. We also evaluate quantitatively and qualitatively the output of our fairness-sensitive algorithms. The weights produced by the neighborhood locally fair PageRank tend to promote protected nodes lying on the boundaries of the two groups especially in homophilic networks, while the fairness-sensitive PageRank tends to jump to protected nodes especially when the requested ϕ is large.

Besides that, previous research has considered algorithms that weight nodes according to their degree, and found biases that arise as a network evolves [8, 9]. Link recommendation systems is known that can affect the evolution of a network. The first objective of a link recommendation system is to speed up the physical evolution of the network as this would have been done without any external intervention as this assumed to be more pleasant for the users. However, there are cases where changing a feature is important and recommendation systems are used for this purpose [10, 11]. In the context of fairness, as it is defined in the first part, we see that existing recommenders preserve the initial bias of the network. We explore the possibilities that recommendations systems have and we show that they can be used to improve the fairness of a network.

At first, we call a recommender fair if the recommendations it makes improve the cumulative PageRank of the protected group. In previous PageRank related perturbation analysis they have study the effect of adding a directed out edge to the individual PageRankof each node [12]. We show in our analysis how adding edges affect the cumulative PageRank of a group. We detect those edges that will raise the PageRank of the protected group and we rank them based the raise they can succeed. We use this raise as a score for a fair link recommendation system and we introduce a hybrid fair recommendation system which acts complementary to known recommendation systems to achieve link recommendation that improve our objective and are close to the natural evolution of the network. In this thesis we embed our recommendation in a recommendation system based on node2vec embeddings [13] creating the hybrid fair link recommendation system.

To evaluate the existing and the fair recommenders we use again real and synthetic data. We explore their impact both in fairness and in quality of link recommendations, meaning the score we derive from the existing recommendation system which we use to create the hybrid fair recommender. In our case this is the node2vec recommendation score. Last we examine in depth the features of the proposed links for each recommendation system and we analyze their underlying mechanisms.

In summary, in this thesis we make the following contributions:

- We propose the fairness-sensitive Pagerank algorithm that modifies the jump vector so as to attain fairness and the locally fair Pagerank algorithms that guarantee that individually each node behaves in a fair manner

- We formulate optimization problems for finding the algorithms that minimizes the utility loss and estimate a lower bound for the optimal utility loss by post-processing the output of Pagerank

- We define the notion of fair link recommendation system.

- We propose the fair and the hybrid fair link recommendation systems.

- We perform experiments on several datasets. Our experiments demonstrate qualitatively and quantitatively the properties of the fair Pagerank algorithms and of the fair recommendatin systems.

1.2 Thesis Structure

The remainder of this thesis is structured as follows. In Chapter 2 we present the required preliminaries knowledges to follow the thesis and a table with all the basic notations we use throughout the text. In Chapter 3 we define fair link analysis and our fair link analysis algorithms. In chapter 4 we present the notion of a fair recommender, we define the problem of fair link recommendations and we introduce a fair recommendation policy. Chapter 5 Includes the experimental evaluation for our fair algorithms and our fair recommendation system. Finally, we compare our work with related research in Chapter 6 and offer our conclusions and our future work ideas in Chapter 7.

Chapter 2

Preliminaries

In this chapter, we present prerequisite knowledges that would be useful for the understanding of this thesis. This includes the PageRank algorithm and its variations and basic theory of the stochastic processes and particular of the Markov chains. Also we present a table with the notation we will use in the following chapters.

2.1 PageRank

The PageRank algorithm is the best-known link analysis algorithm, popularized by its application in the Google search engine. The scoring vector of the algorithm is the stationary distribution of a random walk on the graph G. We will use $\mathbf{p}$ to denote this probability vector (which is the same as the scoring vector $\mathbf{w}$).

More formal, let $G = (V, E)$ be a graph where $V = \{0, 1, ..., n\}$ is the set of nodes and $E \subseteq VxV$ is the set of edges exist in the graph. PageRank is a link analysis algorithm $\mathcal{G}^n \to \mathbb{R}^n$, $\mathcal{G}^n$:= the set of all graphs of size n, that defines a stochastic process based on the graph structure - and particular an ergodic Markov chain.

In PageRank the nodes are the states of the stochastic process and the transition probabilities are defined by the formula $\mathbf{P}_G = (1-c)\cdot\mathbf{P} + c\cdot\mathbf{e}\mathbf{v}^T,\ 0 < c < 1$, where $\mathbf{P}_G$ is the transition matrix defined by PageRank and $\mathbf{P}$ is the transition matrix derived from the adjacency matrix of the graph with the difference that if a node has 0 out neighbors (it is an absorbing node) then we consider all nodes to be its neighbors, c is a factor known as "jump coefficient", $\mathbf{v}$ is a probability vector (commonly the uniform one) known as "jump vector" and $\mathbf{e}$ is the vector with 1 in all of its coordinates. PageRank vector can also be expressed as

$$\mathbf{p}^t = (1-c)\mathbf{p}^t \cdot \mathbf{P}_G + c\mathbf{v}^t \tag{2.1}$$

PageRank is a way of evaluating the nodes and assign to them a score that represents their importance in the network. A common problem is when we want to evaluate the nodes not by their importance in the network but by their importance in the network for a specific node $u \in V$ (or for a specific set of nodes $U \subset V$). In this case the PageRank is called personilized and this can be achieved by setting the jump vector

$$\mathbf{v}(i) = \begin{cases} 1/|U|, i \in U \\ 0, otherwise \end{cases}$$

From 2.1 we can take that

$$\mathbf{p}^t = a\mathbf{v}^t \cdot \mathbf{Q},\ \ \mathbf{Q} = [\mathbf{I} - (1-a)\cdot\mathbf{P}_G]^{-1} \tag{2.2}$$

The last relation implies that the PageRank of each node is the average personilized PageRank of that node ($U = \{u\}$), for all the nodes in the network.

2.2 Markov Chains

Markov chains are descrete stochastic processes on countable state spaces where the probability to move from one state to another depends only upon the present state. We call a state transient if the probability to leave from it is greater than 0 and absorbing otherwise. When the probability to move from every state to any other state is greater than 0 the Markov chain is called irreducible. Moreover, when for all states the gcd of the number of steps that is possible to go back to the current state is equal to 1 the Markov chain is called aperiodic. We call ergodic a Markov chain that is both irreducible and aperiodic. These stochastic processes have what we call as "stationary distribution". This means the existence of a distribution over the states, which describes the probability to visit each state, with the property that stays stable through time. We can express this as $\exists\ \mathbf{p} : \mathbf{p}^t = \mathbf{p}^t \cdot \mathbf{P}\ \wedge\ \mathbf{p}^T \cdot \mathbf{e} = 1, \mathbf{p} \in \mathbb{R}^n$.

An interesting category of stochastic processes are the absorbing Markov chains. An absorbing Markov chain is a Markov chain that has one or more absorbing states and it is possible from every transient state to reach at least an absorbing one. We will now present the basic theory of absorbing Markov chains and a common modification of a Markov chain to an absorbing one, which provides us with the tools to prove interesting properties.

As it is clear by now every state in an absorbing Markov chain is either transient or absorbing. If we consider that the transient states are the first k states and the absorbing ones the last $n-k,\ n = |V|$, then we can write the transition matrix in its "canonical form":

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_t & \mathbf{R} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.3}$$

where $\mathbf{P}_t$ is the transition matrix from transient to transient states, $\mathbf{R}$ is the transition matrix from transient to absorbing states, $\mathbf{0}$ is the zero matrix denoting the probabilities from absorbing to transient states and $\mathbf{I}$ is the identity matrix denoting the probabilities from absorbing to absorbing states.

Transition probabilities in m steps are given by $\mathbf{P}^m$, from 2.3 we get:

$$\mathbf{P}^m = \begin{bmatrix} \mathbf{P}_t^m & \mathbf{R}^* \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.4}$$

Since there is a probability to reach an absosbing state from any transient state (not necessarily in 1 step) is greater than 0 we have that $lim_{m\to\infty}\mathbf{P}_{ij}^m = 0,\ \forall i,j \in \{i|i \in V \wedge i \text{ is a transient state}\}$. Also we have that $\mathbf{P}_{ii}^m = 1,\ \forall i \in \{i|i \in V \wedge i \text{ is an absorbingt state}\}$. These implies that the $\lim_{m\to\infty}\mathbf{P}^m$ exists and it holds:

$$\lim_{m\to\infty}\mathbf{P}^m = \begin{bmatrix} \mathbf{0} & \mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{2.5}$$

where $\mathbf{B}_{ij}$ denotes the probability that has a random walk to gets absorbed from the transient state $k+j$ starting from state $i,\ 1 \le i \le k, 1 \le j \le (n-k)$

We can see that the i,j element of matrix $\mathbf{N} = \mathbf{I} + \mathbf{P}_t^1 + \mathbf{P}_t^2 + ...$ (the series converges and the matrix $\mathbf{N}$ is well defined) gives us the expected number of visits to state j if we start the random walk from state i until we get absorbed by an absorbing state. Matrix $\mathbf{N}$ is called the "foundamental matrix" of an absorbing Markov chain. Now, if we mulitply both sides of equation of matrix $\mathbf{N}$ by $(\mathbf{I} - \mathbf{P}_t)$ we get:

$$(\mathbf{I} - \mathbf{P}_t) \cdot \mathbf{N} = \mathbf{I} \tag{2.6}$$

which implies that $(\mathbf{I} - \mathbf{P}_t)$ is the invers of the matrix $\mathbf{N}$.

Knowing the foundamental matrix we are able now to calculate the absorbing probabilities for the transient states denoted as $\mathbf{B}$ in the canonical form. This because the probability to be absorbed from state j starting from the transient state i is equal with the expected times I visit each state multiplied by the probability to be absorbed in the next step from that state, or more formally:

$$\mathbf{B} = \mathbf{N}\mathbf{R} \tag{2.7}$$

$\mathbf{R}$ as defined in the canonical form.

2.3 Relation Between PageRank and Absorbing Markov Chains.

Last we must point out that for an ergodic Markov chained defined from the PageRank algorithm we can define a new absorbing Markov chain by adding some absorbing random nodes and by connecting each of the pre existing nodes in one of the new absorbing nodes with probability equal to the jump probability c. Then, the transition probabilities between transient states are defined as $\mathbf{P}_t = (1 - c)\mathbf{P}$ and $\mathbf{R}_{ij} = c$, if the transient state i has been connected to the new absorbing state $k + j$ and 0 otherwise. For this pair of Markov Chains, from equations 2.2, 2.6 it holds that the foundamental matrix of the new absorbing Markov chain is equal with the matrix of personalized PageRanks $\mathbf{Q}$.

$$\mathbf{N} = \mathbf{Q} \tag{2.8}$$

2.4 Useful Notations

Table 2.1: Useful notations.

Notation	Description
$\mathbf{A}_i$	the i_{th} row of matrix $\mathbf{A}$.
$\mathbf{e}$	vector with all coordinates 1.
$\mathbf{e}_i$	$\mathbf{e}_i(j) = 1$ for $i = j$, otherwise = 0.
$G = (V, E)$	Graph with set of nodes V and set of edges E.
E_i	Out neighbors of node i.
$G' = (V, E')$	The new graph resulting of the addition of new edges.
$\widetilde{E}$	$E' = E \cup \widetilde{E}$
$\mathbf{p}$	PageRank vector
c	Jump coefficient of PageRank.
$\mathbf{v}$	Jump vector of PageRank.
$\mathbf{P}$	Transition matrix derived from graph.
$\mathbf{P}_G$	Transition matrix defined by PageRank.
$\mathbf{P}_t$	Transition matrix between transient states.
$\mathbf{R}$	Transition matrix from transient to absorbing states.
$\mathbf{Q}$	$\mathbf{Q}_i$:= is the pesonilized PageRank of node i.
$\mathbf{D}$	Perturbation matrix of rank one.
$\mathbf{B}$	Absorption probabilities matrix of an absorbing Markov chain.
$\mathbf{N}$	The foundamental matrix of an absorbing Markov chain.

Chapter 3

Fairness Aware PageRank

In this chapter we formally define the problem of ranking and we study the fairness aware ranking in link analysis with focus on famous PageRank. We start with a quick introduction about link analysis algorithms and we define the notion of fairness for this kind of algorithms. We introduce different approaches to succeed a fairness aware ranking based on PageRank and we study the utility of the fair algorithms in cases where we care only for a subset of the nodes, we call this approach targeted.

3.1 Introduction and Problem Definition

A link analysis algorithm can be seen as a function $A : \mathcal{G}^n \to \mathbb{R}^n$ from the set $\mathcal{G}^n$ of all graphs of size n to the real vectors of size n. The function takes as input a graph $G = (V, E)$ (directed, or undirected) of size n, and produces a vector **w** of size n, which assigns a weight w_v to each node v in the graph. This weight defines the importance of the node in the graph G, and it depends on the graph structure. The best known link analysis algorithm is PageRank, which we consider in this thesis.

Given a graph $G = (V, E)$, we assume that there exists a subset of nodes that define a *protected group*. This group may be defined based on a protected attribute of the nodes in the graph, such as race or gender. In this thesis, we consider two types of nodes, the groups R and B of red and blue nodes, and we assume that R is the protected group. We denote with $r = \frac{|R|}{n}$, and $b = \frac{|B|}{n}$, the fraction of nodes that belong to the red and blue group respectively.

We will say that a link analysis algorithm is *fair*, if it assigns weights to each group according to a specified ratio ϕ. Ratio ϕ may be specified so as to implement specific affirmative action policies, or other fairness enhancing interventions. For example, ϕ may be set in accordance to the 80 percent rule advocated by the US Equal Employment Opportunity Commission (EEOC), or some other formulation of disparate impact [14].

Definition 3.1 (Fair link analysis). A link analysis algorithm $A : \mathcal{G}^n \to \mathbb{R}^n$ is ϕ-fair on graph G, if for the output $\mathbf{w} = A(G)$, it holds that: $\frac{\sum_{v \in R} w_v}{\sum_{v \in V} w_v} = \phi$, where $R \subset V$ is the protected set of nodes.

For instance by setting $\phi = r$, we ask for a fair link analysis algorithm that assigns weights proportionally to the sizes of the two groups. In this case, fairness is analogous to demographic parity, i.e., the requirement that the demographics of those receiving a positive outcome are identical to the demographics of the population as a whole [15]. It is easy to show (see Appendix) that in this case the weights produced by the fair link analysis are such that the average weight of the red nodes is the same with the average weight of the blue nodes.

We define the following problem:

Problem 1. *Given a value ϕ, a graph G, and a link analysis algorithm A, design a link analysis algorithm A_F that is ϕ-fair on graph G.*

Note that the fair variant A_F will necessarily change the original weights of algorithm A, incurring some loss in *utility*. We quantify the *utility loss* using the sum of squares loss function $L(A, A_F) = \|A(G) - A_F(G)\|^2$. We then consider the problem of designing a fair algorithm that minimizes utility loss.

Problem 2. *Given a value ϕ, a graph G, and a link analysis algorithm A, design a link analysis algorithm A_F that is ϕ-fair on graph G, such that the utility loss $L(A(G), A_F(G))$ is minimized.*

Finally, we consider an extension of the fairness definition that asks for a fair distribution of weights among a specific set of nodes S that is given as input. We assume that the set S is selected such that it contains nodes from both groups R and B.

Definition 3.2 (Targeted Fair link analysis). A link analysis algorithm $A : \mathcal{G}^n \rightarrow \mathbb{R}^n$ is targeted ϕ-fair on graph $G = (V, E)$ for a set of nodes $S \subset V$, if for the output $\mathbf{w} = A(G)$, it holds that $\frac{\sum_{v \in S \cap R} w_v}{\sum_{v \in S} w_v} = \phi$, where $R \subset V$ is the protected set of nodes.

In this thesis, we consider the PageRank link analysis algorithm.

3.2 Fairness Sensitive PageRank

Our first algorithm achieves fairness by keeping the transition matrix fixed and changing the jump vector $\mathbf{v}$ so as to meet the fairness criterion.

3.2.1 The Algorithm

First, we note that that pagerank vector $\mathbf{p}$ can be written as linear function of the jump vector $\mathbf{v}$. Solving Equation (2.1) for $\mathbf{p}$, using column vector notation, we have that $\mathbf{p} = \mathbf{Q}\mathbf{v}$, where

$$\mathbf{Q} = c\left([\mathbf{I} - (1-c)\mathbf{P}_G]^{-1}\right)^T$$

Let $\mathbf{p}_R$ denote the pagerank mass that is allocated to the nodes of the protected category. We have that

$$\mathbf{p}_R = \left(\sum_{i \in R} \mathbf{Q}\mathbf{v}\right)[i] = \left(\sum_{i \in R} \mathbf{Q}_i^T\right)\mathbf{v} = \mathbf{Q}_R^T\mathbf{v} \tag{3.1}$$

where $\mathbf{Q}_i^T$ is the i-th row of matrix $\mathbf{Q}$, and $\mathbf{Q}_R^T$ is the vector that is the sum of the rows in the set R. In order for the algorithm to be fair, we need $\mathbf{p}_R = \phi$. Our goal is to find a vector $\mathbf{v}$ such that $\mathbf{Q}_R^T\mathbf{v} = \phi$.

Does such a vector always exist? We can prove the following:

Lemma 3.1. *Given the vector $\mathbf{Q}_R^T$, there exists a vector $\mathbf{v}$ such that $\mathbf{Q}_R^T\mathbf{v} = \phi$, if and only if, there exist entries i, j in $\mathbf{Q}_R^T$, where $\mathbf{Q}_R^T(i) \leq \phi$ and $\mathbf{Q}_R^T(j) \geq \phi$*

Proof. We have that $\mathbf{p}_R = \sum_{j=1}^{N} \mathbf{Q}_R^T(j) v_j$, that is, $\mathbf{p}_R$ is the weighted average of the values $\mathbf{Q}_R^T(j)$, with weights v_j, where $0 \leq v_j \leq 1$. Since $\mathbf{Q}_R^T \mathbf{v} = \phi$, there must exist at least one entry i with $\mathbf{Q}_R^T(i) \leq \phi$, and one entry j $\mathbf{Q}_R^T(j) \geq \phi$. Conversely, if there exists two such entries i, j, then we can find values v_i and v_j, such that $v_i \mathbf{Q}_R^T(i) + v_j \mathbf{Q}_R^T(j) = \phi$ and $v_i + v_j = 1$. □

3.2.2 Optimizing Utility

An implication of Lemma 3.1 is that, in most cases, there are multiple jump vectors that give a fair pagerank vector. We are interested in the solution that minimizes the utility loss.

We first consider the case were we want fairness over all nodes. To solve this problem we exploit the fact that the utility loss function $L(\mathbf{p_v}, \mathbf{p_u}) = \|\mathbf{p_v} - \mathbf{p_u}\|^2$ is convex, and that we can express the fairness requirement as a linear function. We can then define the following convex optimization problem.

$$
\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & \|\mathbf{Q}\mathbf{x} - \mathbf{p_u}\|^2 \\
\text{subject to} \quad & \mathbf{Q}_R^T \mathbf{x} = \phi \\
& \sum_{i=1}^{n} \mathbf{x}_i = 1 \\
& 0 \leq \mathbf{x}_i \leq 1, \; i = 1, \ldots, n
\end{aligned}
$$

This problem can be solved using standard convex optimization solvers.

3.2.3 Targeted Fairness Algorithm

We will now formulate a similar convex optimization problem for the targeted fairness problem. Let $\mathbf{Q}_S^T = \sum_{i \in S} Q_i^T$ be the sum of rows of $\mathbf{Q}$ for the nodes in S, and $\mathbf{Q}_{R|S}^T = \sum_{i \in S \cap R} Q_i^T$be the sum of rows of $\mathbf{Q}$ for the R nodes in S. We define a convex optimization problem that is exactly the same as in Section 3.2.2, except for the fact that we replace the constraint $\mathbf{Q}_R^T \mathbf{x} = \phi$ with the constraint $\mathbf{Q}_{R|S}^T \mathbf{x} = \phi \mathbf{Q}_S^T \mathbf{x}$

We can model specific cases by adding additional constraints. For example, let $T_k(\mathbf{w})$ denote the k nodes with the largest weights in vector $\mathbf{w}$, and let $S = T_k(\mathbf{p_u})$, that is, the top-k nodes of the original Pagerank algorithm. We want fair redistribution of Pagerank among the nodes in S, but we also want these nodes to remain in the top-k position in the fair pagerank, that is, $T_k(\mathbf{p}) = T_k(\mathbf{p_u})$. This requirement can be achieved by adding the constraint:

$$\mathbf{Q}_i^T \mathbf{x} \geq \mathbf{Q}_j^T \mathbf{x},\ i \in T_k(\mathbf{p}), j \notin T_k(\mathbf{p}).$$

3.3 Locally Fair PageRank

The locally fair PageRank algorithms take a microscopic view, by asking that *each individual node* acts fairly, i.e., each node distributes its own pagerank to red and blue nodes fairly. In random walk terms, local fairness defines a dynamic process that can be viewed as a random walk that is fair, i.e., at each step, and not just at convergence, the probability of being at a node of the protected group is ϕ.

3.3.1 The Algorithms

In our first algorithm, the *neighborhood locally fair PageRank* algorithm, each node distributes its own pagerank fairly among the red and blue nodes in its neighbors. The *residual-based locally fair PageRank algorithms* generalize this idea, by again asking that each node allocates its pagerank fairly among the red and blue nodes, but not necessarily among the red and blue nodes in its own neighborhood. Finally, we show that the local PageRank algorithms are fair.

The neighborhood locally fair PageRank algorithm

We first consider a node that treats its neighbors fairly, that is, by allocating a fraction ϕ of its pagerank to its red neighbors and the remaining $1 - \phi$ fraction to its blue neighbors. In random walk terms, at each node the probability of transitioning to a red neighborhood is ϕ and the probability of transitioning to a blue neighborhood $1 - \phi$.

Specifically, we define the *neighborhood locally fair pagerank* (LFPR$_N$) $\mathbf{p}_N$ as follows. Each node i splits the $\phi\,\mathbf{p}_N(i)$ portion of its pagerank value evenly among its red out-neighbors and the remaining $(1-\phi)\,\mathbf{p}_N(i)$ portion of its pagerank evenly among its blue out-neighbors. Similarly, we use a modified "fair" jump vector $\mathbf{v}_N$ with $\mathbf{v}_N[i] = \frac{\phi}{|R|}$, if $i \in R$, and $\mathbf{v}_N[i] = \frac{1-\phi}{|B|}$, if $i \in B$.

Let $out_R(i)$ and $out_B(i)$ be the number of edges directed from node i to red nodes and blue nodes respectively. We define $\mathbf{P}_R$ as the normalized adjacency matrix that includes links to red nodes, or random jumps to red nodes if such links do not exist:

$$\mathbf{P}_R(i,j) = \begin{cases} \frac{1}{out_R(i)}, & \text{if } j \in R,\ out_R(i) \neq 0, \text{ and } (i,j) \in \mathrm{E} \\ \frac{1}{|R|}, & \text{if } j \in R, \text{ and } out_R(i) = 0 \\ 0, & \text{otherwise} \end{cases}$$

$\mathbf{P}_B$ is defined similarly. The transition matrix $\mathbf{P}_N$ of the LFPR$_N$ algorithm is:

$$\mathbf{P}_N = \phi\, \mathbf{P}_R + (1-\phi)\, \mathbf{P}_B$$

and, the neighborhood locally-fair pagerank vector $\mathbf{p}_N$ is defined as:

$$\mathbf{p}_N^T = (1-c)\mathbf{p}_N^T \mathbf{P}_N + c\, \mathbf{v}_N^T$$

The neighborhood locally-fair pagerank value $\mathbf{p}_N$ of a node is the stationary probability that a neighborhood-fair walker ends up at this node.

The residual-based locally fair PageRank algorithms

We consider an alternative fair behavior for individual nodes. Similarly to the LFPR$_N$ algorithm, each node i acts fairly by respecting the ϕ ratio when distributing its own pagerank to red and blue nodes. However, now node i treats its neighbors the same, independently of their color and assigns to each of them the same portion of its pagerank. When a node is in a "biased" neighborhood, i.e., the ratio of its red neighbors is different than ϕ, to be fair, node i distributes the remaining portion of its pagerank to nodes in the underrepresented group. We call the remaining portion *residual* and denote it by $\delta(i)$. How $\delta(i)$ is distributed to the underrepresented group is determined by a *residual policy*.

Intuitively, this corresponds to a fair random walker that upon arriving at a node i, with probability 1-$\delta(i)$ follows one of i's outlinks and with probability $\delta(i)$ jumps to one or more nodes in the locally underrepresented group.

We now describe the algorithm formally. We divide the nodes in V into two sets, L_R and L_B, based on the fraction of their red and blue neighbors. Set L_R includes the "blue-biased" nodes, that is, all nodes i such that $(1-\phi)\, out_R(i) < \phi\, out_B(i)$, that is, the nodes for which the ratio of red nodes in their neighborhood is smaller than the required ϕ ratio. These are the nodes having a residual that needs to be distributed to red nodes. Analogously, L_B includes all "red-biased" nodes, that is, all nodes i such that $(1-\phi)\, out_R(i) \geq \phi\, out_B(i)$.

Let us first consider a node i in L_R. Each neighbor of i gets the same portion of i's pagerank, let $\rho_R(i)$ be this portion. To attain the ϕ ratio, the residual $\delta_R(i)$ of i's pagerank goes to the red nodes. Portions $\rho_R(i)$ and $\delta_R(i)$ must be such that:

$$(1-\phi)\ (out_R(i)\,\rho_R(i) + \delta_R(i)) = \phi\ (out_B(i)\,\rho_R(i)) \tag{3.2}$$

$$out_R(i)\,\rho_R(i) + out_B(i)\,\rho_R(i) + \delta_R(i) = 1 \tag{3.3}$$

From Equations (3.2) and (3.3), we get $\rho_R(i) = \frac{1-\phi}{out_B(i)}$ and the residual is $\delta_R(i) = \phi - \frac{(1-\phi)\,out_R(i)}{out_B(i)}$.

Analogously, for a node i in L_B, we get $\rho_B(i) = \frac{\phi}{out_R(i)}$ and a residual $\delta_B(i) = (1-\phi) - \frac{\phi\,out_B(i)}{out_R(i)}$ that goes to the blue nodes.

Example. Consider a node i with 5 out-neighbors, 1 red and 4 blue, and let ϕ be 0.5. This is a "blue-biased"node, that is a node in L_R. In the original PageRank algorithm, each of the 5 neighbors gets $1/5$ of i's pagerank, resulting in red nodes getting $1/5$ and blue nodes $4/5$ of i's pagerank, which is an unfair behavior for node i. With the residual algorithm, each of i's neighbors gets $\rho_R(i)$ = $1/8$ portion of i's Pagerank, resulting in red neighbors getting $1/8$ and blue neighbors $4/8$ of i's pagerank. The residual $\delta_B(i)$ = $3/8$ goes to nodes in the red group so as to attain the ϕ ratio and make i fair. Which of the nodes in the red group will get the residual is determined by the residual policy. In terms of the random walker interpretation, a random walker that arrives at i, with probability $5/8$ chooses one (any) of i's outlinks and with probability $3/8$ jumps to nodes in the red group.

The transition matrix $\mathbf{P}_L$ is defined as

$$\mathbf{P}_L(i,j) = \begin{cases} \frac{1-\phi}{out_B(i)}, & \text{if } (i,j) \in E \text{ and } i \in L_R \\ \frac{\phi}{out_R(i)}, & \text{if } (i,j) \in E \text{ and } i \in L_B \\ 0, & \text{otherwise} \end{cases}$$

Let δ_R be the vector carrying the red residual, that is, $\delta_R[i] = \phi - \frac{(1-\phi)\,out_R(i)}{out_B(i)}$, if $i \in L_R$ and 0 otherwise. Similarly, let δ_B be the vector carrying the blue residual, that is, $\delta_B(i) = (1-\phi) - \frac{\phi\,out_B(i)}{out_R(i)}$, if $i \notin L_B$ and 0 otherwise. We have a total red residual $\Delta_R = \mathbf{p}_L^T \delta_R$ and a total blue residual $\Delta_B = \mathbf{p}_L^T \delta_B$, where $\mathbf{p}_L$ is the locally fair pagerank vector.

To express the residual distribution policy, we introduce two matrices, matrices $\mathbf{X}$ and $\mathbf{Y}$, that capture the policy for distributing the residual to red and blue nodes respectively. Specifically, $\mathbf{X}[i,j]$ denotes the portion of the $\delta_R(i)$ of node $i \in L_R$ that goes to node $j \in R$ and $\mathbf{Y}[i,j]$ the portion of the $\delta_B(i)$ of node $i \in L_B$ that goes to node $j \in B$.

The locally-fair pagerank vector $\mathbf{p}_L$ is defined as:

$$\mathbf{p}_L^T = (1-c)\mathbf{p}_L^T (\mathbf{P}_L + \mathbf{X} + \mathbf{Y}) + c\,\mathbf{v}_N^T$$

Residual Distribution Policies. The $\mathbf{X}$ and $\mathbf{Y}$ allocation matrices allow us the flexibility to specify appropriate policies for distributing the residual. In particular, the following holds (proof in the Appendix):

Lemma 3.2. *The* LFPR$_N$ *algorithm is a special case of the residual-based algorithm, with*

$$\mathbf{X}_N[i,j] = \begin{cases} \frac{1}{out_R(j)}, & \text{if } i \in R,\ j \in L_R, \text{ and } (j,i) \in E \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{Y}_N[i,j] = \begin{cases} \frac{1}{out_B(j)}, & \text{if } i \in B,\ j \in L_B, \text{ and } (j,i) \in E \\ 0 & \text{otherwise} \end{cases}$$

We also consider residual policies where all nodes follow the same policy in distributing their residual. In this case, the residual policy is expressed through two (column) vectors $\mathbf{x}$ and $\mathbf{y}$, with $\mathbf{x}[i]$ being the portion of Δ_R going to red node i, and $\mathbf{y}[i]$ the portions of Δ_B going to blue node i. In this case, we have:

$$\mathbf{p}_L^T = (1-c)\mathbf{p}_L^T (\mathbf{P}_L + \delta_R\,\mathbf{x}^T + \delta_B\,\mathbf{y}^T) + c\,\mathbf{v}_N^T.$$

We define two locally fair PageRank algorithms based on two intuitive policies of distributing the residual, namely:

(1) the *Uniform Locally Fair PageRank* (LFPR$_U$) algorithm, which distributes the residual uniformly. Specifically, for the LFPR$_U$ algorithm, we define the vector $\mathbf{x}$, as $\mathbf{x}[i] = \frac{1}{|R|}$ if $i \in R$ and 0 otherwise, and the vector $\mathbf{y}$, as $\mathbf{y}[i] = \frac{1}{|B|}$, if $i \in B$ and 0 otherwise.

(2) the *Proportional Locally Fair PageRank* (LFPR$_P$) algorithm, which distributes the residual proportionally based on the original pagerank weight $\mathbf{p_u}(i)$ of node i. Specifically, for the LFPR$_P$ algorithm, we define the vector $\mathbf{x}$, as $\mathbf{x}[i] = \frac{\mathbf{p}[i]}{\sum_{i \in R} \mathbf{p}[i]}$, if $i \in R$ and 0 otherwise, and the vector $\mathbf{y}$, as $\mathbf{y}[i] = \frac{\mathbf{p}(i)}{\sum_{i \in B} \mathbf{p}[i]}$, if $i \in B$ and 0, otherwise.

Fairness of the locally fair PageRank algorithms

In the locally fair pagerank algorithms, each node in the graph treats the red and blue nodes fairly by respecting the ϕ ratio. However, each node acts independently of the other nodes in the network. It is interesting to see how this microscopic view of fairness relates to our macroscopic view of link fairness.

We prove the following theorem.

Theorem 3.1. *The locally fair PageRank algorithms are fair.*

Proof. We must show that $\frac{\sum_{v \in R} \mathbf{p}_N(u)}{\sum_{v \in V} \mathbf{p}_N(u)} = \phi$. Since each node in the graph gives a portion ϕ of its pagerank to red nodes, we have

$$\sum_{v \in R} \mathbf{p}_N(u) = \sum_{v \in V} \phi \, \mathbf{p}_N(u)$$

which proves the theorem. □

3.3.2 Optimizing Utility

We consider how to optimally distribute the residual so as to minimize the utility loss of the fair Pagerank. We denote this algorithm as LFPR$_O$. To this end, we define appropriate $\mathbf{x}$ and $\mathbf{y}$ residual distribution vectors by formulating an optimization problem.

We can write the vector $\mathbf{p}_L$ as a function of the vectors $\mathbf{x}$ and $\mathbf{y}$ as follows:

$$\mathbf{p}_L^T(\mathbf{x}, \mathbf{y}) = c\,\mathbf{v}^T\left[\mathbf{I} - (1-c)(\mathbf{P}_L + \delta_R\,\mathbf{x}^T + \delta_B\,\mathbf{y}^T)\right]^{-1}$$

We can now define the optimization problem of finding the vectors $\mathbf{x}$ and $\mathbf{y}$ that minimize the loss function $L(\mathbf{p}_L, \mathbf{p_u}) = \|\mathbf{p}_L(\mathbf{x}, \mathbf{y}) - \mathbf{p_u}\|^2$ subject to the constraint that the vectors $\mathbf{x}$ and $\mathbf{y}$ define a distribution over the nodes in R and B respectively.

We solve this optimization problem using gradient descent. We enforce the distribution constraints by adding a penalty term $\lambda\ ((\sum_{i=1}^n \mathbf{x}_i - 1)^2 + (\sum_{i=1}^n \mathbf{y}_i - 1)^2)$. We enforce the positivity constraints through proper bracketing at the line-search step.

Note that we can also formulate a convex optimization problem asking for the jump vector that minimizes utility loss, as in Section 3.2.2. In this case, since the transition matrix is fair, we just need to constrain the jump vector to obey the ϕ ratio.

3.3.3 Targeted Fair Local Algorithms

We show how to apply the local algorithms to the targeted fairness problem. Let S_R and S_B be the red and blue nodes in the set S respectively, and let I_S be the set of in-neighbors of S. The idea is that the nodes in I_S should distribute their PageRank to S_R and S_B fairly, such that the ratio of the portion that goes to nodes in S_R and the portion that goes to nodes in S_B is equal to $\frac{\phi}{1-\phi}$. We can implement the same redistribution policies as in the case of the neighborhood local and the residual-based local fair algorithms.

We also need the (global) jump vector $\mathbf{v}$ to obey the ϕ ratio for the nodes in S. We can achieve this by redistributing the probability $|S|/n$ of the jump vector according to the ϕ ratio. Note that there is a variety of policies one could implement, depending on a specific objective. For example if we want to increase the weight of the nodes in S, we can make the jump vector allocate all probability to the nodes in S.

3.4 A Post Processing Approach

We now consider a post processing approach in which we assume that we are given a weight vector $\mathbf{w} = A(G)$ of a link analysis algorithm A on graph G. The goal is to produce a new weight vector $\mathbf{f}$ such that: (1) $\mathbf{f}$ is fair, and (2) the utility loss $L(\mathbf{w}, \mathbf{f}) = \|\mathbf{w} - \mathbf{f}\|^2$ is minimized. The post-processing algorithm is agnostic to the fact that the weight vector $\mathbf{w}$ is the result of a link analysis algorithm, much less of the specific link analysis algorithm (e.g., Pagerank). Therefore, the vector $\mathbf{f}$ that minimizes the loss $L(\mathbf{w}, \mathbf{f})$ may not be attainable by any Pagerank algorithm.

3.4.1 The Post Processing Algorithm

Given the weight vector $\mathbf{w}$, let $\mathbf{w}_R$ denote the weight vector for the nodes in R, and $\mathbf{w}_B$ the weight vector for the nodes in B. We also use W_R to denote the total weight allocated to R, and W_B to denote the total weight allocated to B. We assume that $\mathbf{w}$

Algorithm 3.1 Optimal Redistribution Algorithm

Input: Excess weight Δ, nodes B, weights $\mathbf{w}_B$
Output: Optimal weight vector $\mathbf{f}_B$

```
B_NZ ← {x ∈ B : w_x > 0}
δ = Δ/|B_NZ|
β = min_{x∈B_NZ} w_x
if β ≥ δ then
    w_x = w_x − δ   for all x ∈ B_NZ

    return  w_B
else
    w_x = w_x − β   for all x ∈ B_NZ
    Δ = Δ − |B_NZ|β

    return  REDISTRIBUTE(Δ,B,w_B)
end if
```

has non-negative entries, and it is normalized so that its entries sum to 1. Without loss of generality assume that $W_R < \phi$. Let $\Delta = \phi - W_R$. To make the vector fair we need to distribute weight Δ to the nodes in R, and remove weight Δ from the nodes in B. It is easy to show that in order to minimize the loss, the optimal redistribution will remove weight $\Delta/|B|$ from all nodes in B and add $\Delta/|R|$ from all nodes in B. This follows from the fact that among all distribution vectors the one with the smallest length is the uniform one. Therefore, we obtain the following lower-bound for the loss:

$$\text{Loss}_{LB} = \frac{\Delta^2}{|R|} + \frac{\Delta^2}{|B|}$$

Note that this lower bound does not guarantee that the new vector $\mathbf{f}$ has non-negative entries, thus it is not a valid weight vector. We now describe an optimal redistribution algorithm that ensures that when removing weight no entry becomes negative, while using the principle that whenever removing weight, the optimal way is to remove uniformly from all nodes. The pseudocode for the algorithm is shown in Algorithm 3.1.

The algorithm takes as input the value of excess weight Δ that needs to be removed, the set of nodes B from which we want to remove the weight, and the current weights $\mathbf{w}_B$ of these nodes. First, it finds the subset of nodes B_{NZ} in B that have non-zero weight. If the minimum weight β among these nodes is at least $\Delta/|B_{NZ}|$, then we can remove the weight uniformly without making the weights negative. The algorithm updates the weights and returns. Otherwise, we can remove at most β. The algorithm removes β from all nodes in B_{NZ} and makes a recursive call with the remaining excess weight $\Delta - |B_{NZ}|\beta$. Note that anytime we want to remove weight from a set of nodes, we remove it uniformly from all nodes, which guarantees optimality. The algorithm returns the updated weight vector $\mathbf{f}_B$ for the nodes in B. We can now compute the optimal loss as

$$\text{Loss}_O = \frac{\Delta^2}{|R|} + \|\mathbf{f}_B - \mathbf{w}_B\|^2$$

3.4.2 Targeted Fairness

Computing algorithmically the optimal redistribution is harder in the targeted fairness case, since there are many different options in how we can redistribute weight. We can move weight between the nodes in S, or bring in weight from outside of S, or move weight out of S, or a combination of those. In Appendix A.1 we compute analytically a lower bound for the loss, which provides some intuition on how the weight is moved in different cases .

Finding the optimal redistribution vector can be formulated as a convex optimization problem:

$$\begin{aligned} \underset{\mathbf{f}}{\text{minimize}} \quad & \|\mathbf{f} - \mathbf{p}\|^2 \\ \text{subject to} \quad & \sum_{i \in S \cap R} \mathbf{f}_i = \phi \\ & \sum_{i=1}^{n} \mathbf{f}_i = 1 \\ & 0 \leq \mathbf{f}_i \leq 1, \; i = 1, \ldots, n \end{aligned}$$

We use the solution of the optimization problem to compare the optimal redistribution with that achieved by the modified Pagerank algorithms.

Chapter 4

PageRank Fair Recommendations

Link recommendation systems are a valuable feature in online social networks. Their main use is to help a network grow faster and create a more pleasant experience for the users. Besides that, link recommendation systems can also be used to change unwilling properties of the network. We explore their possibilities on helping a network grow in a more fair way, we show how we can succeed this and we propose a link recommendation system for this purpose.

4.1 Introduction and Problem Definition

We consider a link recommendation system, as a function which accepts as input a graph $G = (V, E)$ and a node $u \in V$ and returns the probability for every possible link recommendations for node/user u, to be accepted by u. We will refer to this probability as the quality of candidate link recommendation. Usually this probability is used to rank the candidate link recommendations and propose the best k of them to the user. However there are cases where the quality of the proposed links isn't the only objective. In this direction various methods have been proposed to recommend links that satisfy some objective function without sacrificing the quality of the recommended links [10, 11, 16].

We consider again a binary sensitive attribute to nodes and a protected group based on this attribute. We want to recommend the edges that enhance the presence of the protected group in the network and are highly possible to be accepted by the user they are about to be proposed to. As in chapter 3, we use as a metric to evaluate the presence of a group in a network, the cumulative score of the group coming from the link analysis algorithm that we are interested in.

Definition 4.1. Given a graph $G = (V, E)$ and a link analysis algorithm A, we call the link recommendation system fair if the edges it proposes improve the cumulative score of the group based on algorithm A.

By the definition we understand that may exist algorithms that are both fair but with different magnitude of impact in the network. For reason of cohesion we focus again on the PageRank algorithm. We start our analysis by identifying the impact of an new edge addition to the node u and we extend our conclusions for a set of nodes being added to a single source. We consider the impact on cumulative PageRank of the group as the fair score of the candidate edge. This is the difference between the new and the old cumulative PageRank. We then define our fair recommendation system based on fair score and our hybrid fair recommendation system which acts complementary to an existing link recommendation system, taking into consideration, not only the fair score, but the recommendation score as well.

4.2 Impact of Adding Edges

Lets assume an unweighted, directed graph $G = (V, E)$ where V is the set of all nodes and E is the set of all edges. And let $G' = (V, E' = E \cup \{(u, v)\})$ where $(u, v) \notin E$. We denote with $\mathbf{p}(R)$, $\mathbf{p}'(R)$ the ratio of the pagerank that goes to Red nodes in the graph G and G' respectively. We denote by k_u the out degree of node u

Now let $\mathbf{Q}$ be the matrix where $\mathbf{Q}_{ij}$ is the personilized PageRank of node i to node u and so $\mathbf{Q}_i(R)$ is the personilized PageRank of node i to all Red nodes. We want to find the impact of the edge (u, v) to the cumulative PageRank of the red group. We prove the following.

Theorem 4.1. *If* $G' = (V, E' = E \cup \{(u,v)\})$ *then:*

$$\mathbf{p}'(R) = \mathbf{p}(R) + \mathbf{p}_u \cdot \frac{\frac{(1-c)}{c}\left[\mathbf{Q}_v(R) - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_w(R)\right]}{(k_u+1) - \frac{(1-c)}{c}\left[\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_{wu}\right]} \tag{4.1}$$

Proof. To prove this we first write the transition matrix $\mathbf{P}$ of G' as a sum of the tranistion matrix $\mathbf{P}$ of graph G and a rank one, perturbation matrix $\mathbf{D}$. This is:

$$\mathbf{P}' = \mathbf{P} + \mathbf{D}, \quad \mathbf{D}_i = \begin{cases} 0, \ i \neq u \\ (-1 + \frac{k_u}{k_u+1})\mathbf{P}_u + \frac{1}{k_u+1}\mathbf{e}_v \end{cases}$$

And then we exploit a fundamental lemma [17] that states that If G is nonsingular, H is of rank 1 and $G+H$ is nonsingular as well, then:

$$(G+H)^{-1} = G^{-1} - \frac{1}{1+g}G^{-1}HG^{-1}, \ g := tr(HG^{-1})$$

We also know from equation 3.1 that $\mathbf{p} = \mathbf{vQ}$ and [18] that $\mathbf{Q} = c \cdot \mathbf{N}$ where $\mathbf{N}$ is the foundamental matrix of an absorbing Markov chain with transition matrix for transient states equal to $(1-c)\mathbf{P}$, c is the jump probability of PageRank.

For $G = [\mathbf{I} - (1-c)\ \mathbf{P}]$ and $H = -(1-c)\cdot\mathbf{D}$, we have:

$$\mathbf{N}' = \mathbf{N} - \frac{1}{1+q}\mathbf{N}(-(1-c)\mathbf{DN}), \ q := tr(-(1-c)\mathbf{DN}) \Rightarrow$$
$$\frac{1}{c}\mathbf{Q}' = \frac{1}{c}\mathbf{Q} - \frac{1}{c^2}\frac{1}{1+q}\mathbf{Q}(-(1-c)\cdot\mathbf{D})\mathbf{Q}, \ q := tr(-(1-c)\mathbf{D}\frac{1}{c}\mathbf{Q}) \Rightarrow$$

$$\mathbf{Q}' = \mathbf{Q} + \frac{1}{c}\frac{(1-c)}{1-\frac{(1-c)}{c}q}\mathbf{QDQ}, \ q := tr(\mathbf{DQ}) \tag{4.2}$$

If we compute $\mathbf{DQ}$, $\mathbf{QDQ}$ we get:

$$\mathbf{DQ}_{ij} = \begin{cases} 0, \ i \neq u \\ \frac{1}{k_u+1}[\cdot\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_{wj}], \quad i = u \end{cases}$$
$$\mathbf{QDQ}_{ij} = \frac{1}{k_u+1}\big(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_{wj})\big)$$

And from (1):

$$\mathbf{Q}'_{ij} = \mathbf{Q}_{ij} + \mathbf{Q}_{iu} \frac{\frac{(1-c)}{c}\left(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wj})\right)}{k_u + 1 - \frac{(1-c)}{c}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu}))}$$

Since $\mathbf{p} = \mathbf{vQ}$ we have to compute

$$\mathbf{p}'(R) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j \in R}^{n} \mathbf{Q}'_{ij}$$

to obtain the formula of the theorem. □

If we examine the formula 4.1 we see that the result agrees with our intuition. The fraction that exists in the formula is the impact that the new edge will have on the PageRank of the red group. First we see that the magnitude of the impact is (by approximation) proportional to the fraction of the PageRank of the source node to the out degree of the source node. This is logical if we consider adding an edge to a node with out degree 1 and to a node with out degree 10. In the first case the node will give much more of its PageRank to the new neighbor. Also we prove right below that the quantity on the denominator is always positive. This means that if an edge will have positive or negative impact is determined by that nominator. What the nominator indicates is that if we want to have gain in the red PageRank of the network we must add an edge that the target node will have greater red personalized PageRank than the average personalized PageRanks of the current neighbors of the source node. The last quantity we didn't comment so far, that exists in the denominator affects the magnitude of the impact. Though, the impact of this quantity on formula is not important, it describes the fact that if the target node of the edge that we added gives less PageRank to the source node than the average of the current neighbors of the source node, then the impact of the edge is getting smaller as the PageRank of the source node is getting smaller.

We will show now that the denominator of the fraction is always positive. To do that we will need the following lemma.

Lemma 4.1. $\forall v, u \in V$, *it holds:*

$$\mathbf{Q}_{vu} < \mathbf{Q}_{uu}$$

Proof. Let $f_{vu}^{(i)}$ be the possibility to reach transient state u starting from the transient state v for the first time at step i. And let $f_{vu}^{*} = \sum_{i=1}^{\infty} f_{vu}^{(i)}$. For the absorbing Markov chain $\overline{X}$ it holds:

$$f_{vu}^{*} < 1 \tag{4.3}$$

That is because there is possibility c for transient state v to be absorbed at the first step to the absorbing state a_0.

Let N_u to denote the number of visits to state u. then:

$$P[N_u = m \mid \overline{X}_0 = u] = {f_{uu}^{*}}^{m-1}(1 - f_{uu}^{*})$$

$$P[N_u = m \mid \overline{X}_0 = v] = \begin{cases} 1 - f_{vu}^{*}, \ m = 0 \\ f_{vu}^{*}{f_{uu}^{*}}^{m-1}(1 - f_{uu}^{*}) \end{cases}$$

So N_u follows geometric distribution with success probability of $(1 - f_{uu}^{*})$ and so:

$$\left.\begin{array}{l} E[N_u \mid \overline{X}_0 = u] = \frac{1}{1-f_{uu}^{*}} \\ E[N_u \mid \overline{X}_0 = v] = f_{vu}^{*} E[N_u \mid \overline{X}_0 = u] \end{array}\right\} \Rightarrow$$

$$E[N_u \mid \overline{X}_0 = v] < E[N_u \mid \overline{X}_0 = u]$$

We also know that $E[N_u \mid \overline{X}_0 = v] = \mathbf{N}_{vu} = \frac{1}{c}\mathbf{Q}_{vu}, \ \frac{1}{c} > 0$ and so:

$$\mathbf{Q}_{vu} < \mathbf{Q}_{uu}$$

□

Now, we can continue with the proof. To do this we define an absorbing Markov chain $\overline{X}$. We add two adsorbing states $n+1, \ n+2$, from now on $a_u, \ a_o$ and we connect the state u to the state a_u and all other states to state a_o, all with probability c. We denote with $\mathbf{B}_{i1}$ the absorbing probability of node i to node a_u and $\mathbf{Q}_{iu}$ the personilized PageRank of node i for the node u.

Lemma 4.2. *The absorption probability of state i to state a_u of the absorbing Markov chain $\overline{X}$ is equal with the personilized PageRank of node i to node u.*

$$\mathbf{Q}_{iu} = \mathbf{B}_{i1}$$

Proof. The new transition matrix $\overline{\mathbf{P}}$ in its canonical form is:

$$\overline{\mathbf{P}} = \begin{bmatrix} (1-c)\mathbf{P} & \mathbf{R} \\ \mathbf{0}_{2xn} & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{R} \in \mathbb{R}^{nx2},$$

Where

$$\mathbf{R} = \begin{cases} c, \ (i = u \wedge j = 1) \vee (i \neq u \wedge j = 2) \\ 0, \ otherwise \end{cases}$$

We know[18] that absorption probabilities $\mathbf{B} = \mathbf{NR}$ where $\mathbf{N}$ is the Foundamental matrix of process $\overline{X}$ and as before $\mathbf{N} = \frac{1}{c}\mathbf{Q}$ So:

$$\begin{cases} \mathbf{B} = \mathbf{NR} \\ \mathbf{N} = \frac{1}{c}\mathbf{Q} \end{cases} \Rightarrow \mathbf{B}_{ij} = \sum_k \mathbf{Q}_{ik}\mathbf{R}'_{kj} = \begin{cases} Q_{iu}, \ j = 1 \\ Q_i(V \setminus \{u\}), \ j = 2 \end{cases}$$

□

Lemma 4.3. *For the personilized PageRank that the node i gives to itself, it holds:*

$$\mathbf{Q}_{ii} = c + (1-c)\frac{1}{k_i}\sum_{w \in E_i}\big(\mathbf{Q}_{wi}\big)$$

Proof. From lemma 4.2 we know that $\mathbf{Q}_{iu} = \mathbf{B}_{i1}$. It also holds [18] that $\mathbf{B}_{i1} = 1\mathbf{R}_{ij} + \sum_{j \in E_i}\mathbf{P}_{ij}\mathbf{B}_{j1}$. So:

$$\begin{aligned} \mathbf{Q}_{iu} &= \mathbf{B}_{i1} \\ &= \mathbf{R}_{ij} + \sum_{j \in E_i}(1-c)\mathbf{P}_{ij}\mathbf{B}_{j1} \\ &= \mathbf{R}_{i1} + (1-c)\sum_{j \in E_i}\mathbf{P}_{ij}\mathbf{Q}_{ju} \\ &= \mathbf{R}_{i1} + (1-c)\mu_{E_i}\big(\mathbf{Q}_{\cdot}(R)\big), \qquad \mathbf{R}_{i1} = \begin{cases} c, \ i = u \\ 0, otherwise \end{cases} \end{aligned}$$

□

From lemma 4.3 we can get now that :

$$k_u + 1 - \frac{(1-c)}{c}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu})) = k_u - \frac{1}{c}\big[(1-c)\mathbf{Q}_{vu} - \mathbf{Q}_{uu}\big]$$

This quantity is always positive because $\mathbf{Q}_{vu} - \mathbf{Q}_{uu}$ is always negative, lemma 4.1, and $1 - c < 1$.

Theorem 4.1 can be generalized by adding a set of nodes to a fixed source node. In this case it holds the following.

Theorem 4.2. *If* $G' = (V, E' := E \cup \widetilde{E})$, $\widetilde{E} = \{(u, v_i) | v_i \in V \wedge v \notin E_u \ i = 1, 2, ..., \widetilde{k}\}$, $E'_u = E_u \cup \widetilde{E}_u$, $\widetilde{E}_u = \{v | (u, v) \in \widetilde{E}\}$ *then:*

$$\mathbf{p}'(R) = \mathbf{p}(R) + \mathbf{p}_u \cdot \frac{\frac{(1-c)}{c}\left(\frac{1}{\widetilde{k}}\sum_{v \in \widetilde{E}_u}(\mathbf{Q}_v(R)) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_w(R))\right)}{\frac{k_u + \widetilde{k}}{\widetilde{k}} - \frac{(1-c)}{c}\left(\frac{1}{\widetilde{k}}\sum_{v \in \widetilde{E}_u}(\mathbf{Q}_{vu}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu})\right)} \tag{4.4}$$

Identical natural meaning as for the single edge formula 4.1 can be derived from the generalized one.

4.3 Efficient Computation of Personalized PageRank

Both the above formulas 4.1, 4.2 include quantities like the red personalized PageRank of all nodes and the personalized PageRank for the source node of all nodes that are prohibitive to calculate them for every node in the network by executing multiple PageRank algorithms. In this section, we present an efficient way to compute all the forth mentioned quantities by executing only two PageRank - like iterative algorithms.

From Lemma 4.2 we know that for the Markov chain $\overline{X}$, $\mathbf{Q}_{iu} = \mathbf{B}_{i1}$, $\mathbf{Q}_i(V \setminus \{u\}) = \mathbf{B}_{i2}$

Furthermore, we know that $\overline{\mathbf{P}}^n_{ij}$ gives as the probability to be in state j starting from state i after n steps. Also we know that $\lim_{n\to\infty} \overline{\mathbf{P}}^n$ exists. From the canonical form of $\overline{\mathbf{P}}$ we take:

$$\overline{\mathbf{P}}^n = \begin{bmatrix} (1-c)^n \mathbf{P}^n & \mathbf{R}^{(n)} \\ 0_{2xn} & \mathbf{I}_2 \end{bmatrix} \Rightarrow \mathbf{B} = \lim_{n\to\infty} \begin{bmatrix} \mathbf{R}^{(n)} \\ \mathbf{I}_2 \end{bmatrix} \Rightarrow$$

$$\mathbf{B} = \lim_{n\to\infty} \overline{\mathbf{P}}^n \cdot \mathbf{e}_{n+1}, \quad \mathbf{e}_{n+1} \in \mathbb{R}^{n+2} \tag{4.5}$$

The above expression allows us to compute the personilized PageRank of all nodes to node u with the computational cost of one PageRank.

Same as in Lemma 4.2 we can define the Markov chain $\widetilde{X}$. We add two absorbing states $n+1$, $n+2$, from now on a_r, a_b respectively. We then unite all red nodes to state a_r with probability c and all the blue nodes to state a_b also with probability c. If $\mathbf{B}_{i1}, \mathbf{B}_{i2}$ are the absorbing probabilities for nodei to a_r, a_b respectivly and $\mathbf{Q}_i(R), \mathbf{Q}_i(B)$ is the ratio of personilized PageRank of node i for Red and Blue nodes, then we can prove the following.

Lemma 4.4. *The absorption probabilities of state i to state a_r, a_b of the absorbing Markov chain $\widetilde{X}$ are equal to the ratio of personilized PageRank of node i that Red and Blue nodes receive respectively. This is:*

$$\mathbf{Q}_i(R) = \mathbf{B}_{i1}, \quad \mathbf{Q}_i(B) = \mathbf{B}_{i2}$$

Proof. proof is identical to lemma 4.2 □

Now working for $\widetilde{X}$ as before we can get a similar expression for $\mathbf{Q}_i(R)$ and compute also the personilizes Red PageRank of all nodes.

From the above is clear that we need computational cost equal to three PageRanks to compute the expressions of theorems 4.1, 4.2 for all possible future edges of the network with fixed source node.

4.4 Fair Link Recommendations

Exploiting the results of the theorems 4.1, 4.2, we propose the fair recommendation system. This saying we mean a link recommendation system that takes into consideration only the maximum gain of cumulative PageRank of the protected group.

Definition 4.2. Given a graph $G = (V, E)$, a source node u and the number of proposed edges k, the **Fair Link Recommendation System** returns the k edges that would have caused the greatest gain on the cumulative PageRank of the protected group, if they had been added independently.

This score - the **fair score** - of an edge is compute with for all candidate edges with the use of the formula 4.1. We then rank the candidate edges based on this score and we return the k best of them. Although, as we see in the chapter 5, the fair recommendation system has amazing results as to our objective of improving the presence of the protected group in the network, we observe that the top edges by the fair score lack in quality score (acceptance probability by node2vec). Furthermore, we observe that the simple recommenders lack significantly in fair score.

In order to balance this trade off between fair score and acceptance probability we propose a hybrid fair link recommendation system. This system is based on an existing link recommendation system of our choice and uses the fair score as complementary to the acceptance probability derived from the recommendation system. More specific the hybrid fair system computes the expected gain (or expected fairness) for all candidate edges for a source node and proposes to the user the best k of them. More formal we give the following definition.

Definition 4.3. Given a graph $G = (V, E)$, a source node u, a link recommendation system and the number of proposed edges k, the **Hybrid Fair Link Recommendation System** returns the k edges with the greatest expected gain on the cumulative PageRank of the protected group, if they had been added independently. Expected gain is calculated based on the acceptance probability deriving from the link recommendation system.

The hybrid fair link recommendation system can be used in any known link recommendation system and is balancing sufficient good both the objective of fairness and the objective of the acceptance probability. In our experimental evaluation we use it complementary to a node2vec based link recommendation system. Node2vec is a famous node embedding algorithm. We construct the link recommendation system as a link prediction process. For every pair of nodes (source , target) we construct the edge embedding by taking the Hadamard product of the two nodes. We then train a logistic regression classifier that returns the probability of an edge to exists in the graph. We use this score as the acceptance probability score. More details about the implementation of the node2vec link recommendation system can be found on chapter 5.

4.5 Fair Important Edges

In the road of understanding how fairness evolves in a networks, it is interesting enough to understand the role of the existing edges to the current fairness of the network. To do that we measure the impact that would have to the network if we removed an edge. This is, as before, a rank one perturbation to the transition matrix of the network and the formula of this score can be calculated equivalent with theorem

4.1 by defining properly the matrix **D**. Following the same process we can take the 2 following theorems:

Theorem 4.3. *If* $G' = (V, E' = E \setminus \{(u,v)\}),\ v \in E_u$ *then:*

$$\mathbf{P}(R)' = \mathbf{P}(R) + \mathbf{p}_u \cdot \frac{\frac{1-c}{c}(\frac{1}{k_u}\sum_{w \in \widetilde{E}_u} \mathbf{Q}_w(R) - \mathbf{Q}_v(R))}{(k_u - 1) - \frac{1-c}{c}(\frac{1}{k_u}\sum_{w \in \widetilde{E}_u} \mathbf{Q}_{wu} - \mathbf{Q}_{vu})}$$

Theorem 4.4. *If* $G' = (V, E' := E \backslash \widetilde{E}),\ \widetilde{E} = \{(u, v_i) | v_i \in V \wedge v_i \in E_u\ i = 1, 2, ..., \widetilde{k}\}$ *then:*

$$\mathbf{p}'(R) = \mathbf{p}(R) + \mathbf{p}_u \cdot \frac{\frac{(1-c)}{c}\left(\frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_w(R)) - \frac{1}{\widetilde{k}}\sum_{v \in \widetilde{E}_u}(\mathbf{Q}_v(R))\right)}{\frac{k_u - \widetilde{k}}{\widetilde{k}} - \frac{(1-c)}{c}\left(\frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu}) - \frac{1}{\widetilde{k}}\sum_{v \in \widetilde{E}_u}(\mathbf{Q}_{vu})\right)} \tag{4.6}$$

We see again that the impact of an edge/a set of edges is determined by the fraction in the formulas. These formulas indicate that important edges for the preservation of the fairness across the network are edges that source node is strong as to his (PageRank/out degree) metric and the target node is strong as to his personalized red PageRank and mainly when its personalized PageRank is far better than the average personalized red PageRank of the rest of out neighbors of the source node. Besides that, an edge is more important if the source node isn't strong at the target's personalized PageRank. This means that the strong target node was managing more PageRank of the source node than it was giving to it. An example of such a good edge would be a source node of the favorite group in a central role in a cluster of favorite group, which would had only a few out neighbors and one edge towards a node of the unfavored group with great red personalized PageRank. The target node could be a leaf in a big cluster of the unfavored team.

Chapter 5

Experimental Evaluation

5.1 Dataset Description

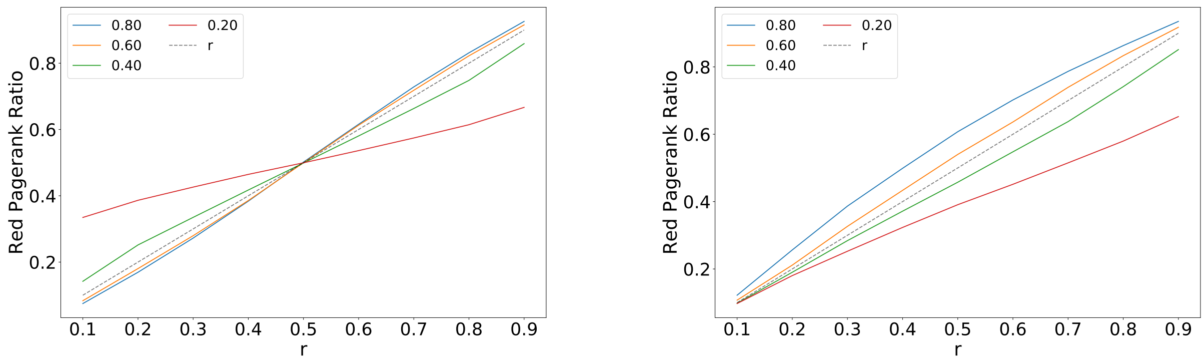
In this section, we evaluate experimentally the different fair PageRank algorithms and provide quantitative and qualitative results. In the experiments we use the following datasets:

- TWITTER: A political retweet graph from [19].
- DBLP2: An author collaboration network constructed from DBLP including a subset of data mining and database conferences.
- BOOKS: A network of books about US politics where edges between books represented co-purchasing[1].
- BLOGS: A directed network of hyperlinks between weblogs on US politic [20].

[1] http://www-personal.umich.edu/~mejn/netdata/

Table 5.1: Real dataset characteristics. r, b relative size of protected and unprotected group, respectively; p_R, p_B pagerank assigned to the red and blue group respectively

Dataset	#nodes	#edges	Protected attribute	r	b	$cross_R$	$cross_B$	p_R	p_B
BOOKS	92	748	political (left)	0.47	0.53	0.063	0.065	0.46	0.54
BLOGS	1,222	19,089	political (left)	0.48	0.52	0.284	0.036	0.33	0.67
DBLP	13,015	79,972	gender (women)	0.17	0.83	0.96	0.86	0.16	0.84
TWITTER	18,470	61,157	political (left)	0.61	0.39	0.07	0.03	0.57	0.43



(a) Symmetric

(b) Asymmetric

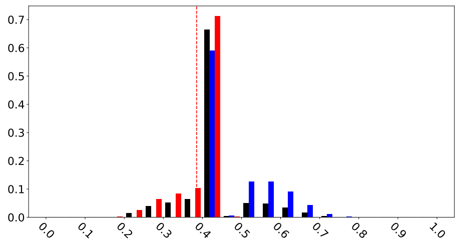
Figure 5.1: Fairness of the PageRank algorithm with the size of the protected group for varying same group preference.

The characteristics of the real datasets, and the protected groups, are shown in Table 5.1. To infer the gender in the DBLP2, we used the python *gender guesser* package[2]. We also report *homophily* which was shown to affect degree distributions among groups [8]. We measure it as the number of mixed edges, i.e., edges between nodes belonging to different groups, divided by $2\,r\,(r-1)$, i.e., the expected number of such edges. Values significantly smaller than 1 indicate that the network exhibits homophily [21].

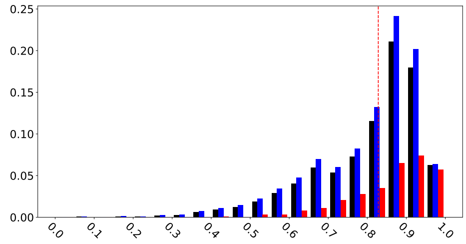
We have used various real data sets. We focus on the following four, while results for additional datasets can be found in the Appendix.

Synthetic networks are generated using a variation of the biased preferential attachment model introduced in [8]. The graph evolves with time as follows. Let $G_t = (V_t, E_t)$ and $d_t(v)$ denote the graph and the degree of node v at time t, re-

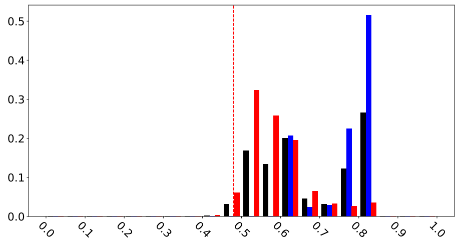
[2]https://pypi.org/project/gender-guesser/



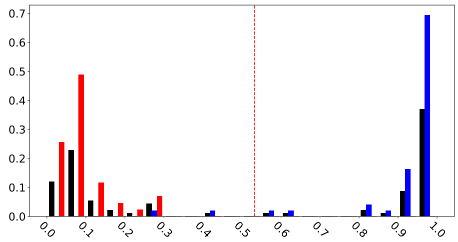
(a) Twitter



(b) DBLP2



(c) Blogs



(d) Books

Figure 5.2: Personalized pageranks starting from each of the blue nodes: histogram of the fraction of the personalized weight. The x-axis corresponds to weights fractions (blue, red and all (black bar)) and the y-axis to the percentage of the blues nodes with the corresponding fraction. The majority of nodes allocate the larger fraction of the personalized weight to blue nodes, thus being highly unfair to the opposite group.

spectively. The process starts with an arbitrary initial connected graph G_0, with $n_0\, r$ red and $n_0\,(1-r)$ blue nodes. At each time step $t+1$, $t>0$, a new node v enters the graph. The color of v is red with probability r and blue with probability $1-r$. Node v chooses to connect with an existing node u with probability $\frac{d_t(u)}{\sum_{w \in G_t} d_t(w)}$. If the color of the chosen node u is the same with the color of the new node v, then an edge between them is inserted with probability p; otherwise an edge is inserted with probability $1-p$. If no edge is inserted, the process of selecting a neighbor for node v is repeated until an edge is created.

Probability p controls the level of homophily in the network, where $p=0$ corresponds to the zero preference to same group, $p=0.5$ to the random preferences and $p=1$ to total preference to the same group. We also consider asymmetry in same group preference probability. In this case, the above procedure is followed by a node v only when v belongs to the red group. A node v in the blue group connects with the selected node u without testing u's color.

Table 5.2: Utility loss with respect to optimal utility ($\frac{LFPR_X}{OPTIMAL}$, for ϕ = 0.5)

Dataset	LFPR_N	LFPR_U	LFPR_P	LFPR_O	SFPR
TWITTER	6.576	6.683	4.218	2.1671	2.699
DBLP2	1.356	1.232	1.516	1.1792	2.6
BLOGS	5.05	5.08	3.163	1.5923	1.73
BOOKS	9.53	4.94	1.576	1.000	1

The datasets and code are available at GitHub [3].

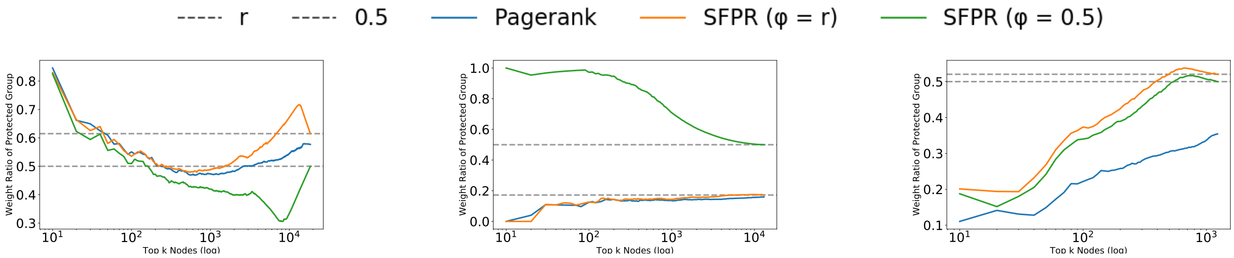
5.2 Fairness in the original Pagerank algorithm.

We use the synthetic datasets to study the behavior of Pagerank for different levels of homophily and relative sizes of the two groups. For this set of experiments, we set ϕ = r. As shown in Figure 5.1, for the symmetric case, when the groups exhibit homoplily ($h = 0.8$ and $h = 0.6$), PageRank is unfair towards the minority group. On the contrary, when the groups exhibit heterophily ($p = 0.4$ and $p = 0.2$), then PageRank is unfair towards the majority group. For the asymmetric case, i.e., when the blue group shows no homophily, being homophilic helps the red group independently of its size, while being heterophilic hurts the red group independently of its size.

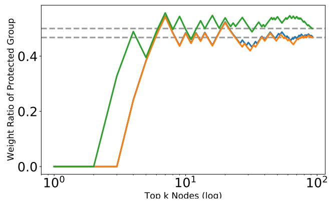
For the real dataset, we report the fraction of the total weight allocated to each of the two groups in Table 5.1. In some cases (BLOGS, TWITTER), the fraction of the weight assigned to the protected group is significantly smaller than r. In all cases, by setting ϕ to the desired level of fairness, we can redistribute weights so that we get the desired ϕ-fairness. We report quantitative and qualitative results for ϕ = 0.5 in the next section.

To get a better insight about the distribution of the weights between the two groups, we also run personalized Pagerank algorithms starting from each node i and calculated for each node i the fraction of the weight allocated to the blue and red nodes (ignoring the Pagerank allocated to the node itself). In all graphs, most of the starting nodes allocate the majority of their personalized pagerank weights to nodes in their group, resulting in highly unfair weights. We report the histogram of the

[3] https://github.com/SotirisTsioutsiouliklis/FairLaR



(a) Twitter (b) DBLP2 (c) Blogs



(d) Books

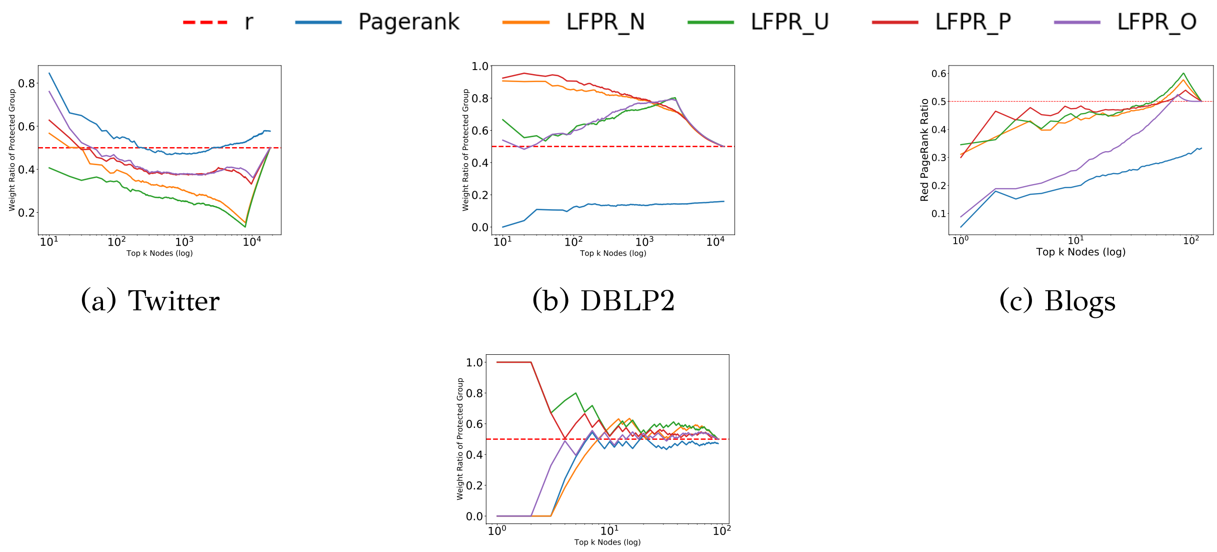
Figure 5.3: Fairness sensitive Pagerank for $\phi = r$ and $\phi = 0.5$.

fraction of the weights allocated to blue and red node for personalized pageranks starting from each of the blue nodes in Figure 5.2. Correlation with homophily can be oberved, with the most homophilic networks, i.e, BOOKS and TWITTER, showing the largest unfairness. Our locally fair Pagerank algorithms can be used to attain ϕ-fairness for personalized Pagerank as well.

5.3 Fairness Aware PageRank Ranking

The fair PageRank algorithms. We run our fair Pagerank algorithm for various values of ϕ. In Figure 5.3, we report results for $\phi = r$ and $\phi = 0.5$ for the fairness sensitive Pagerank (SFPR), while in Figure 5.4, results for $\phi = 0.5$ for the various locally fair Pagerank algorithms (i.e, neighbor ($LFPR_N$), uniform ($LFPR_U$), proportional ($LFPR_P$) and with optimized residual ($LFPR_O$)). Results for $\phi = r$ can be found in the Appendix.

Table B.1 reports the utility loss for each of the fair pagerank algorithms relative to the optimal utility loss as estimated by Algorithm 3.1. For the non-optimized algorithms, as expected taking into account the original pagerank values, the $LFPR_P$ algorithm results in the smallest utility loss. The $LFPR_N$ algorithm incurs the highest utility loss. The utility loss decreases significantly when considering the optimized



(a) Twitter (b) DBLP2 (c) Blogs

(d) Books

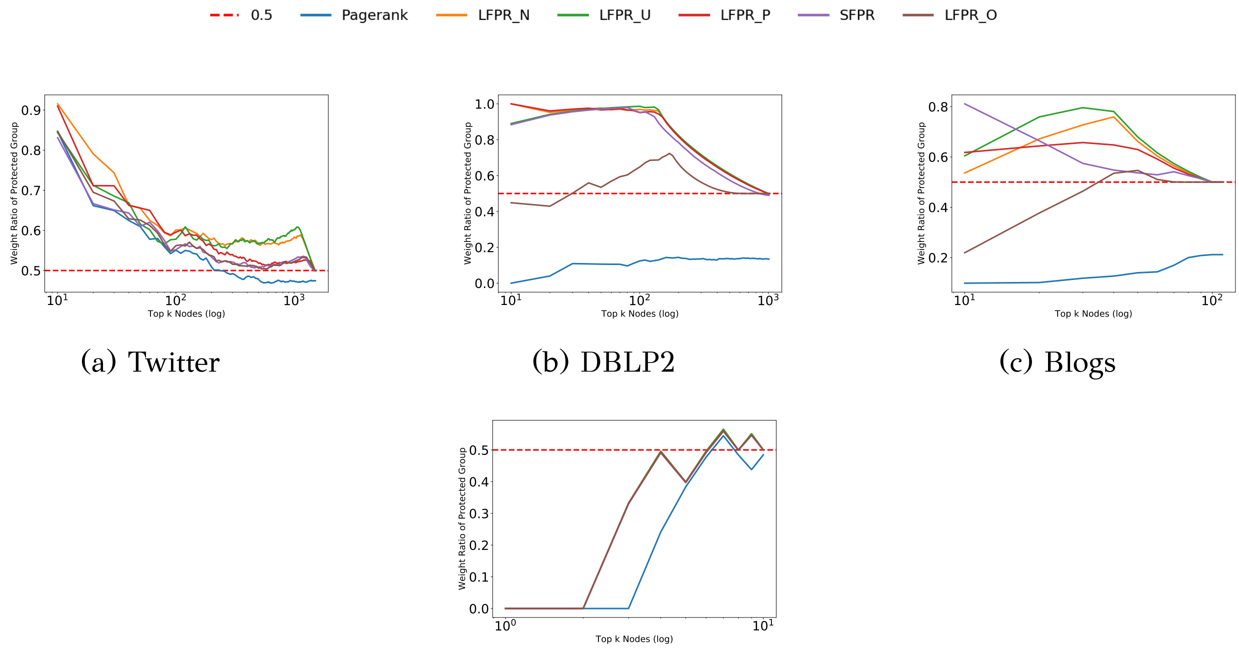
Figure 5.4: Locally fair Pagerank algorithms for $\phi = 0.5$.

algorithms. It is interesting that for different datasets different variants perform better. This suggests that the different algorithms provide different levers for adjusting fairness. Depending on the dataset one approach may be more applicable for preserving utility than another. We discuss this further in our qualitative comparison of the algorithms.

We report the results of all targeted fair PageRank algorithms in Figure B.3. The targeted fair PageRank algorithms allows us to focus on a specific set of nodes and adjust their weights in a fair manner. In this experiment, we selected the set S for each dataset, so as to include the 10% of the nodes having the highest original PageRank values. This provides us with the flexibility to adjust weights in the top positions.

Residual distribution policies. In this set of experiments, we take a closer look at the set of nodes that are mostly affected by the different residual distribution policies. To this end, we consider the sets LOSS (resp. GAIN) with the 10 red and 10 blue nodes whose PageRank decreased (resp. increased) the most. For each node i, we define its *in-neighborhood fairness*, $in\_f(i)$ as the ratio of its red in-neighbors over all its in-neighbors. Thus, $in_f(i) = 0$ corresponds to a node i i with only blue in-neighbors, $in_f(i) = 1$ to only red in-neighbors and $in_f(i) = 0.5$ to a balanced in-neighborhood.

We run the algorithms on all datasets. The results are depicted in Figure 5.6. As shown in Figure 5.6(a), with very low variance, all algorithms penalize the nodes that are in homophilic in-neighborhoods, that is, red nodes with large in_f, and blue



(a) Twitter (b) DBLP2 (c) Blogs

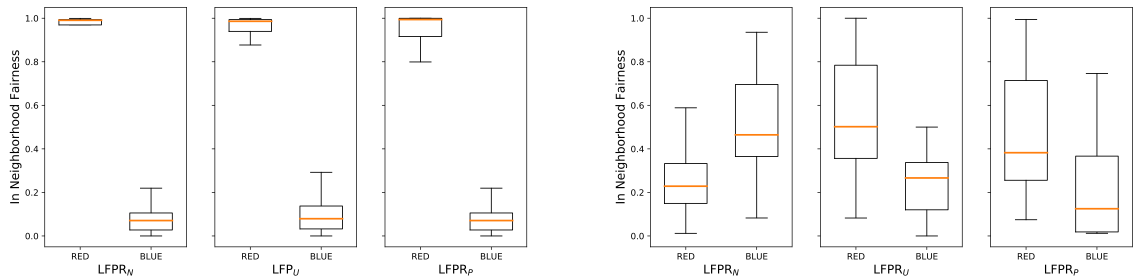
(d) Books

Figure 5.5: Targeted fair PageRank algorithms and the optimal post-processing redistribution for $\phi = 0.5$. In each dataset, the target set S includes the 10% of nodes with the highest PageRank weights.

nodes with small in_f. The algorithms exhibit different behavior regarding which nodes each algorithm promotes as shown in Figure 5.6(b). LFPR$_N$ promotes the red (i.e., protected group) nodes that are in heterophilic (i.e., protected-group dominated) in-neighborhoods. This does not hold for the blue nodes. On the other hand, LFPR$_P$ does not seem to particularity favor heterophilic red nodes, while it tends to follow the trend of the PageRank promoting homophilic blue (i.e, favored-group) nodes. Finally, for LFPR$_U$, in_f seems to play a lesser role, with the small favoritism to homophilic blue nodes most probably reflecting the homophily of the blue group in the underlying population.

Qualitative comparison. To provide some insight on the weights produced by the various algorithm, we visualize their output for $\phi = 0.5$ in Figures 5.7 and 5.8. In the visualizations, red nodes are colored red, and blue nodes are colored blue. Their size depends on the value of the quantity we visualize.

For the Twitter and the books datasets, where the fraction of the weight of the protected group is close to ϕ, the fairness sensitive pagerank is very similar to the



(a) Loss (b) Gain

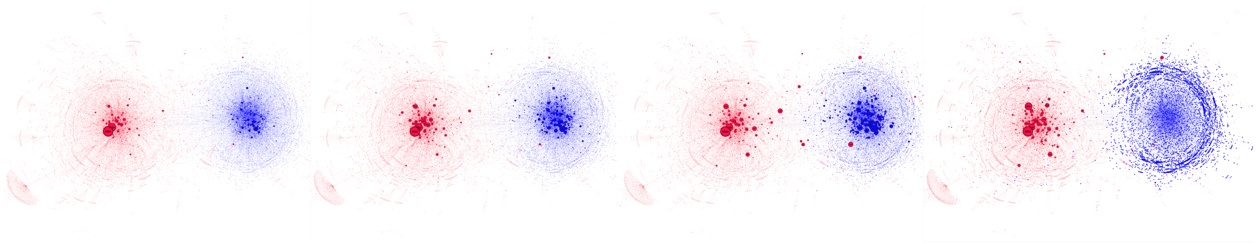
Figure 5.6: In neighborhood fairness for the nodes with the maximum loss and gain for each algorithm.

original one. For the BLOGS and especially for the DBLP2 datasets, where the fraction of the weight of the protected (red) group is much smaller, the fairness sensitive pagerank promotes red nodes. We also visualize the jump vector for the fairness sensitive pagerank. We observe that for the TWITTER and the BOOKS dataset, where the algorithm is already "almost" fair, the jump vector assigns rather uniform weights, as the original Pagerank. For the other two datasets, it gives large values to a number of red nodes. This suggests an interesting line for future work: considering these nodes in link recommendation algorithms, since it seems that these nodes play a role in fairness.

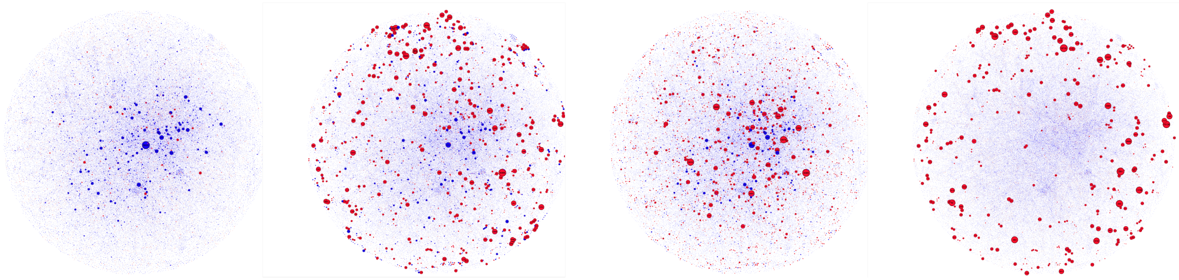
The neighborhood locally fair pagerank algorithm produces different weights from the original Pagerank for all four datasets. In all cases, it promotes nodes connecting the two opposite groups, i.e., nodes that are minorities in their neighborhoods. This is more evident in the most homophilic networks, that is, in TWITTER and BOOKS. Such nodes are also known as weak links and play an important role. They can also be useful in the context of recommendations, since research shows that it is more likely for such nodes to be accepted from the other side [22].

5.4 PageRank Fairness Aware Recommendations

In this section we compare and evaluate different known recommendation policies and our novel ones. We start by studying various known link recommendation systems and we show that they don't improve our objective (enhance the protected group in the network). Due to the fact that our policy acts complementary to an existing one



(a) Twitter: Original PageRank (b) Fairness Sensitive PageRank (c) Local Neighborhood PageRank (d) Jump Vector for Sensitive



(e) BDLP2: Original PageRank (f) Fairness Sensitive PageRank (g) Local Neighborhood PageRank (h) Jump Vector for Sensitive

Figure 5.7: Visualization for $\phi = 0.5$

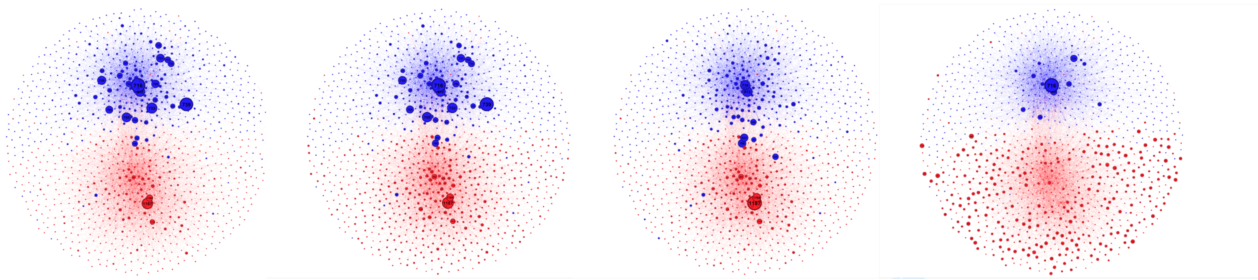
we could use any recommendation policy to apply it on. In these experiments we use node2vec recommender as it is one well known recommender that has been used extensively in research with good results.

Node2vec recommender was implemented by taking the node2vec implementation of snap [4] and train a logistic regression classifier with sklearn module in python. For both the classifier and the node2vec embeddings we used the default settings. For the training of the classifier we use as train test the 80% of the network's edges for positive examples and equal amount of edges that don't exist for negative example. We use the rest 20% of positive edges and equal amount of negative edges as test set. For all the rest recommendation algorithms we used their implementation on networkx [5] module for python

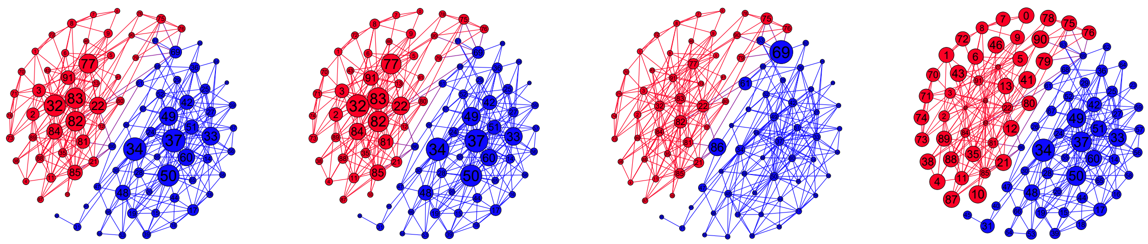
To evaluate our policy we compare both the impact to our objective and the quality of link recommendations it produces. We evaluate the quality of recommendations by assuming that the acceptance probability of candidate links coming from the original recommender (node2vec in our case) is in fact true. We proceed further the analysis of our policy by identifying the differences in quality features of link recommendations

[4]http://snap.stanford.edu/snap/index.html

[5]https://networkx.github.io/



(a) Blogs: Original PageRank (b) Fairness Sensitive PageRank (c) Local Neighborhood PageRank (d) Jump Vector for Sensitive



(e) Books: Original PageRank (f) Fairness Sensitive PageRank (g) Local Neighborhood PageRank (h) Jump Vector for Sensitive

Figure 5.8: Visualization for $\phi = 0.5$

and explaining in a level the underlying mechanism of each one.

5.4.1 Existing Recommendation Policies

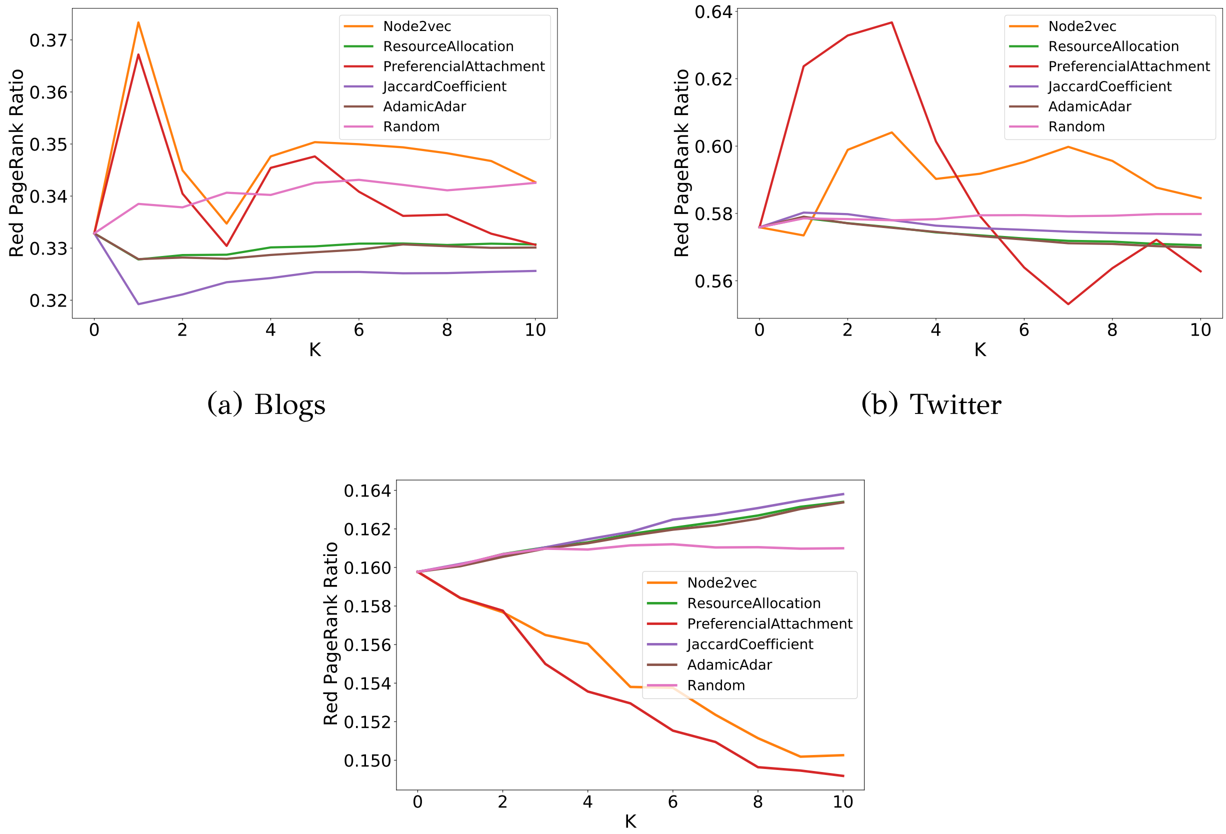
Each of the link recommendation policies has its own mechanism of selecting and proposing links to a source node. Also, as we saw in theorem 4.1, the impact of a link addition to the network it depends not only on the node that is being proposed but also on the node that is proposed to. To study the impact of the different policies to the red PageRank of the network we have decided to use the resource allocation, Jaccard coefficient, Adamic Adar, preferential attachment and node2vec recommendation policies. In this direction we conducted the following experiment.

First we choose a set of source nodes. Then we take the 10 best link recommendations for every source node by each policy. After that, we add one edge per source node (starting from the best one and continuing with the second best etc.). We continue the process until we have added all the 10 edges to all source nodes. We use K to denote the number of the links that have been added to each of the source nodes (e.g. $K = 4$ means that we have added 4 links to each of the source nodes). We conduct the experiments for 3 different sets of source nodes. In the first one we chose randomly 10% of the nodes and for the other two we chose a hundred best red and a hundred best blue nodes. By best we mean those nodes in which we expect to have the greatest expected impact on the network as derived from the formula in theorem 4.1. The expected impact for the source nodes is computed by approximation due to the prohibitive complexity of the actual computation. This approximation is $(PageRank\ )/(outdegree)$. We present the red PageRank of the network as evolves after the addition of the recommended edges for $K = 0, 1, ..., 10$. In this experiment series we excluded the Books dataset because its small size is prohibitive.

In general, we can separate the known recommendation policies into two categories. In the first we classify policies that their recommendation score is based on the number of the common neighbors between two nodes (resource allocation index, Jaccard coefficient, Addamic Adar) and In the second one policies that tend to propose strong/popular nodes in the network (preferential attachment, node2vec). Algorithms in each category tend to share common characteristics and behavior, or equivalently they affect the networks evolution in the same way.

To start with, we can see in fig. 5.9 that policies of the first group behave similar with the random policy in a smooth way, giving a small privilege to the bigger group. The policies of the second group have a slightly unpredicted behavior in the first 1 - 3 steps but they too follow the trend of the initial PageRank as the network evolves. This unpredicted behavior can be explained - as we will see below - due to the fact that policies of the second group, in contradiction with the ones of the first (see table 5.3), tend to propose common neighbors independently of the source node. Thus, the characteristic of the first few proposed nodes dominate the impact on the network. As the number of the links accepted added to the source node gets higher, the more the initial trend of the network is represented in the target nodes, so the impact of the proposed links in networks Red PageRank approaches that of the random link recommendations.

As far as the source node concerns, we can see from figures 5.10, 5.11 that the behavior of the policies of second group change slightly in the pace they converge to the random policy, meaning they are more robust to source node selection while the policies of the first group change total behavior and they enhance the team that the source nodes belong to. This is expected, if we consider that social networks tends



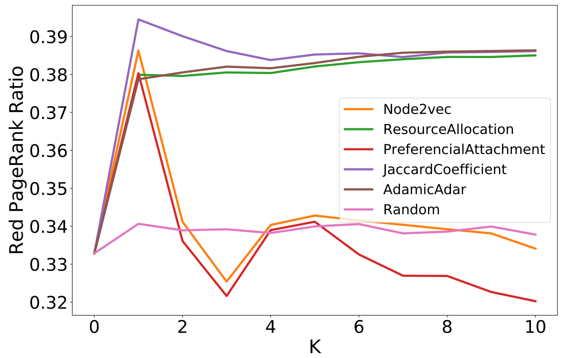
(a) Blogs (b) Twitter

(c) DBLP

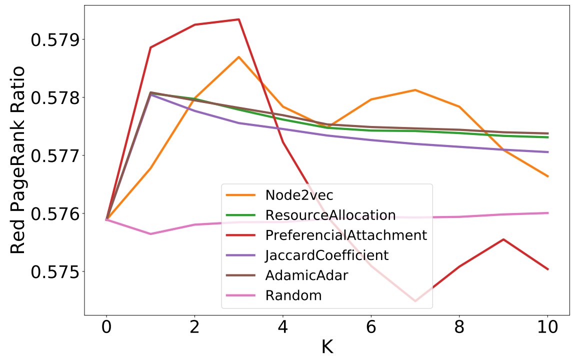
Figure 5.9: Impact on Fairness - Known Recommenders - Random Source Nodes.

to be homophilic (this also holds in our datasets) by their nature and these kind of recommendation policies propose links in distance two. The first one means that a blue source node will probably have blue neighbors and it will exists in a blue dominated cluster and that a blue node will also have a higher personalized blue PageRank. The equivalent holds for a red source node. So from our formula about edge addition impact and considering the second observation about the distance of proposed links, it is clear that the links proposed of the policies of first group are ideal to rise the PageRank of the source node's group.

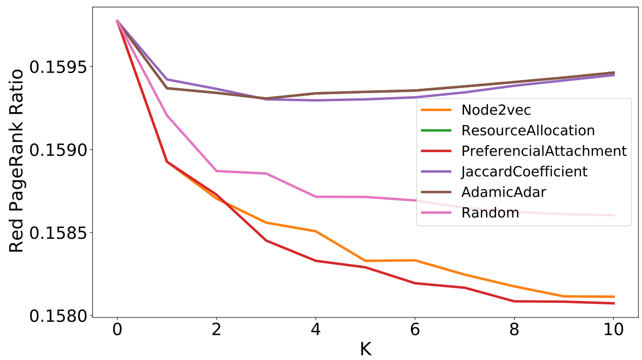
To sum up, though the link recommendation policies already exists have shown some valuable results, proposing edges in a simple way and having an expected impact on the network, they can not be used as link recommendation system if we care to restrict the discrimination on a network, improve fairness or in more general term enhance the presence of a protected group. We continue the experimental evaluation by showing how we can tackle this behavior and how our proposed fair policy can



(a) Blogs



(b) Twitter



(c) DBLP

Figure 5.10: Impact on Fairness - Known Recommenders - Red Source Nodes.

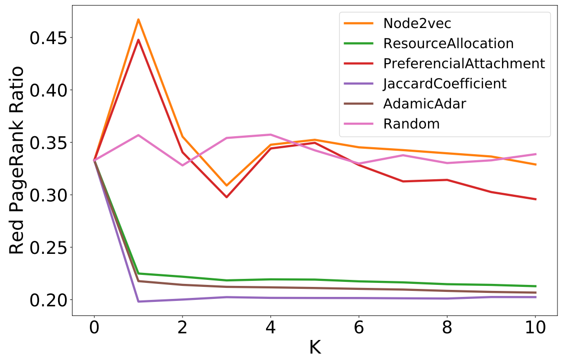
affect a recommender towards this direction.

5.4.2 Fair Recommendation Policies

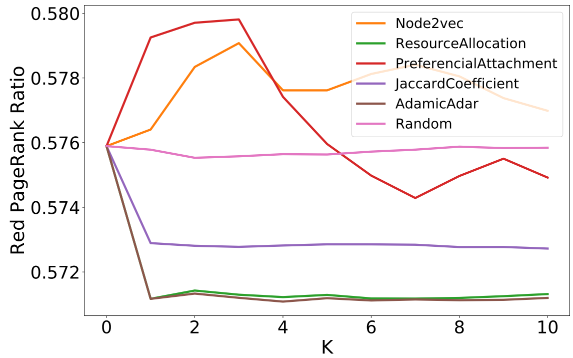
In this section we study the link recommendations provided by our fair recommender, from node2vec recommender and from the hybrid fair recommender. We observe that between node2vec and the fair recommender there is a significant trade off between the quality and the wanted impact of the links, however this trade off is nicely balanced by the hybrid fair recommender.

To compare this result we present the red PageRank ratio of each network as we did before, only this time for random, node2vec, fair and hybrid fair recommenders and for the same experiment we also present the average acceptance probability of the networks as this calculated based on node2vec recommendation score.

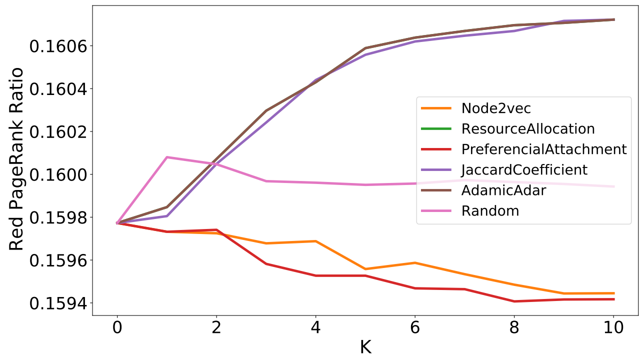
As we see in figures 5.12, 5.13, 5.14, the fair policy rises the red PageRank more



(a) Blogs



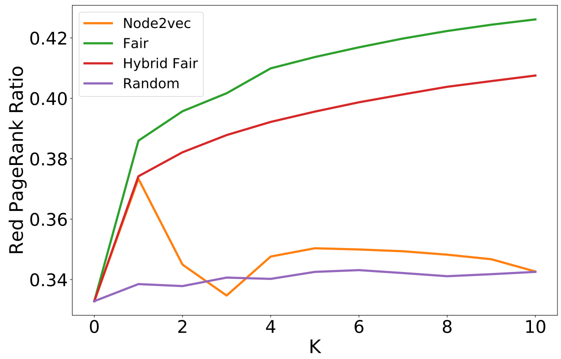
(b) Twitter



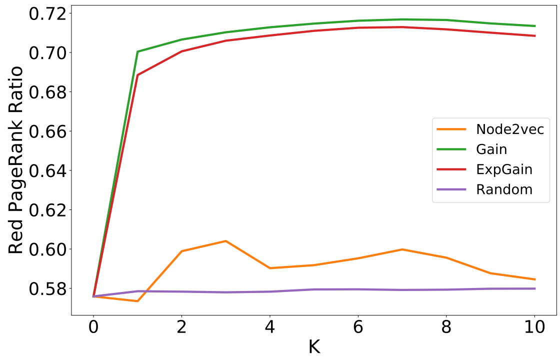
(c) DBLP

Figure 5.11: Impact on Fairness - Known Recommenders - Blue Source Nodes.

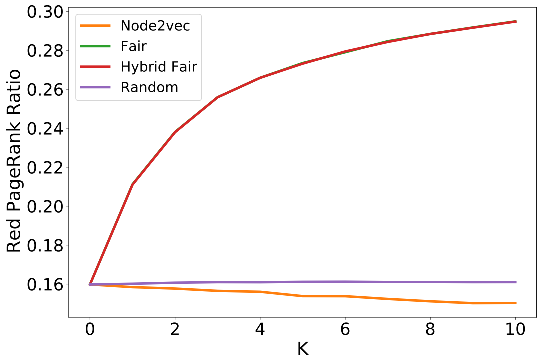
than any other and the hybrid fair policy succeeds impressive rise as well. This shows us that taking into consideration both fair and recommendation score doesn't affect significant the wanted impact on the network. Moreover by figures 5.15 5.16 5.17 we conclude that fair policy doesn't take into consideration at all the acceptance probability of the link it proposes, performing in some cases worst than the random recommendation policy. This makes these recommendation invaluable as it is highly possible not be accepted by the users. On the other side we see that the hybrid fair policy manages to restrict this phenomenon, approaching in a satisfying level the average recommendation score of the node2vec recommender. Also, we see that the results are not affected by the set of source nodes which is really important as in practice social networks differs in their demographic characteristics. Last but not least, we observe (DBLP case) that when a recommendation system provides low information about the quality of each recommended link, then as expected, the hybrid fair algorithm follows the fair algorithm as the score of the recommendation system



(a) Blogs



(b) Twitter

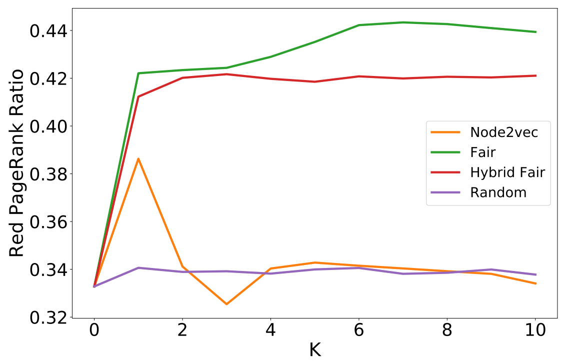


(c) DBLP

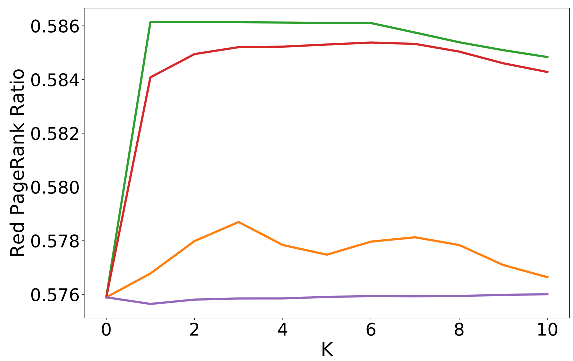
Figure 5.12: Impact on Fairness - Fair Recommenders - Random Source Nodes.

isn't valuable. This indicates that the fair algorithm takes into consideration not only the ranking of the recommendation system but also the magnitude of the score, protecting in a way the fair score - our first objective - when there isn't valuable information. The almost neutral score of the node2vec recommendation system is partially explained by the high density and the lack of homophily in the network. This can be corrected by choosing others than the default parameters to learn the node2vec embeddings.

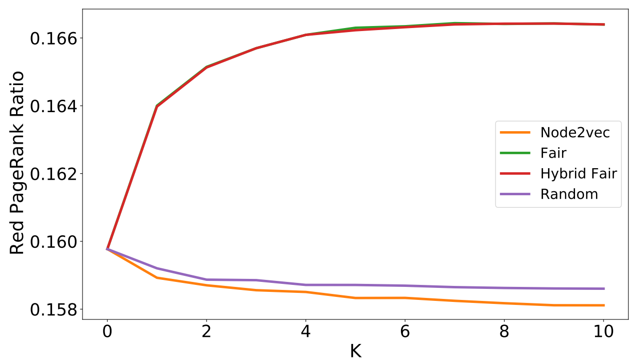
An interesting observation about the impact of fair recommendation policies is presented in figures 5.18 5.19. In this figures we see the ratio of red PageRank at top k nodes by PageRank as it has been formed at the end of the previous experiment. The interesting effect of fair policies in contrast with the node2vec is that fair policies improve the red PageRank ratio in a network while node2vec preserve the original distribution. This difference means that fair recommendations help the protected group to gain higher scores and positions in the ranking by PageRank algorithm and



(a) Blogs



(b) Twitter



(c) DBLP

Figure 5.13: Impact on Fairness - Fair Recommenders - Red Source Nodes.

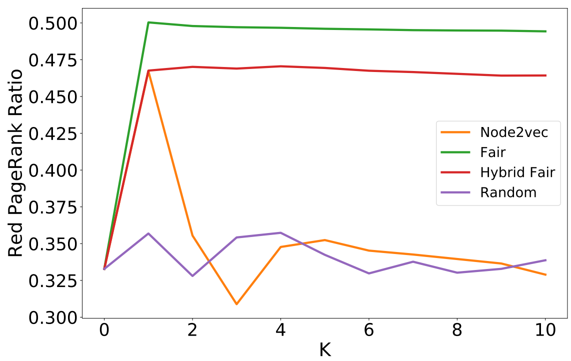
Table 5.3: Number of Total Unique Target Nodes by Policy

Dataset	Node2vec	Fair	Hybrid Fair	Resource Allocation	Jaccard Coefficient	Adamic Adar	Preferencial Attachment
Blogs	15	20	40	351	662	315	17
Twitter	51	154	105	5184	7492	4833	12

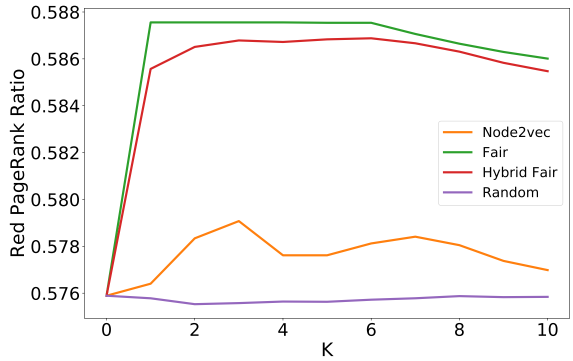
so be represented more fairly in the top positions.

5.4.3 Target Nodes Analysis

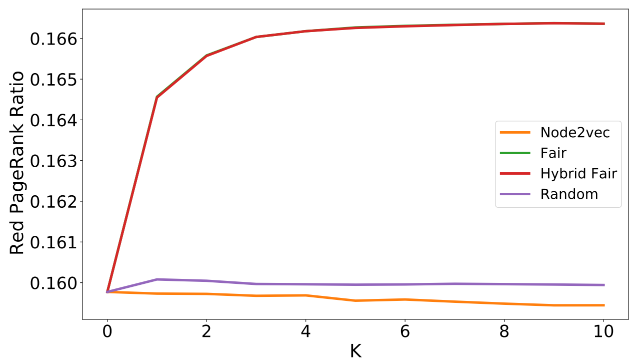
So far it seems that hybrid fair policy performs really good in all the metrics of our evaluation. However, having a link recommender in use, is always interesting to understand its link recommendation mechanism in a more simple way. That is understanding what are these node and edge characteristics that rule its decisions and which is the dynamic of each proposed target node to the network. This kind of results most of times aren't so unexpected and they provide us with useful insights based on simple metrics. In this direction we study the nodes that have been selected



(a) Blogs



(b) Twitter



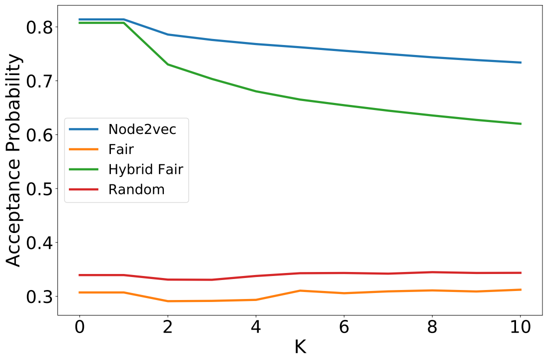
(c) DBLP

Figure 5.14: Impact on Fairness - Fair Recommenders - Blue Source Nodes.

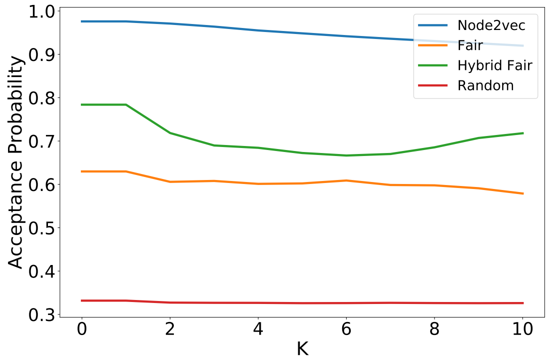
from the previous experiment. We first present in table 5.3 the total distinct number of nodes for each policy per dataset and then, in tables 5.4, 5.5 we present the quality features for the node2vec, fair and hybrid fair policies.

In table 5.3 we see that node2vec, preferencial attachment, fair and hybrid fair, are tend to propose a relative small number of distinct target nodes. On the other, side the rest of the policies propose a much larger number. This result follows our instinct as the node2vec, preferencial attachment and fair policies tends to highlight nodes strong globally in the network while the rest policies propose nodes that are locally strong. The result for the hybrid fair policy can be explained straight forward as it is the combination of the node2vec and fair policies.

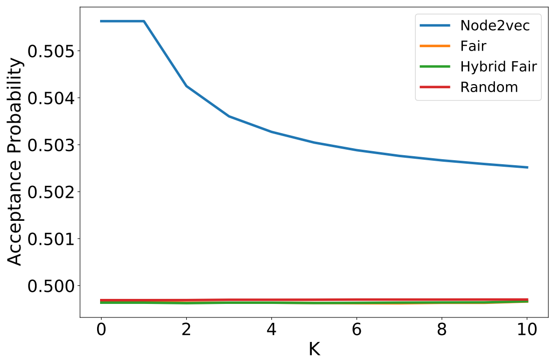
Except the fact that node2vec, fair and hybrid fair policies select a relative small number of distinct target nodes to propose, there is a subset of them that have significant more occurrences. To study further the 3 policies we keep only the top in occurrences nodes. The selection process is described in figures 5.21, 5.22. The exact



(a) Blogs



(b) Twitter



(c) DBLP

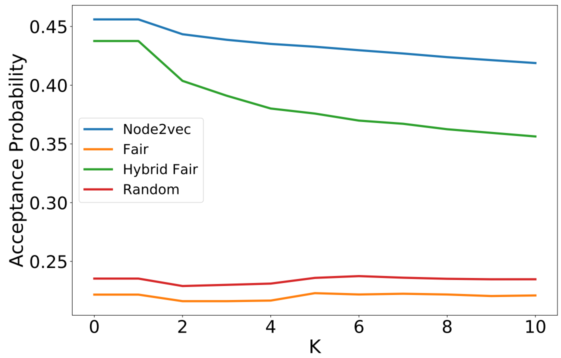
Figure 5.15: Average Acceptance Probability - Random Source Nodes.

number of minimum occurrences that we accept is found by plotting the occurrences of each node in descending order, we observe that this plot approaches a sigmoid function, something that allow us to define properly this minimum.

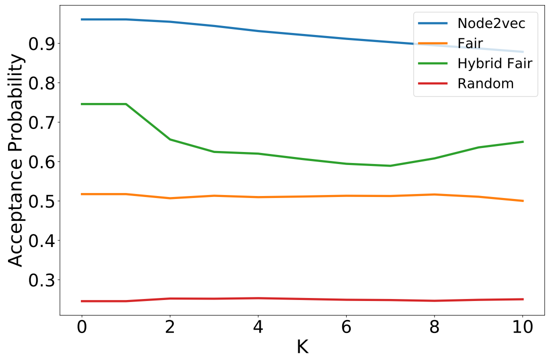
We present these results in tables 5.4, 5.5. First we observe that node2vec recommendations are characterized by nodes that are strong by PageRank and gather the distribution of distances around smaller values. Fair policy nodes are high in red personalized PageRank and in red out neighbors ratio while hybrid fair balances all the above having in general higher values in all the forthmentioned scores.

Table 5.4: Target Quality Features in Blogs Network.

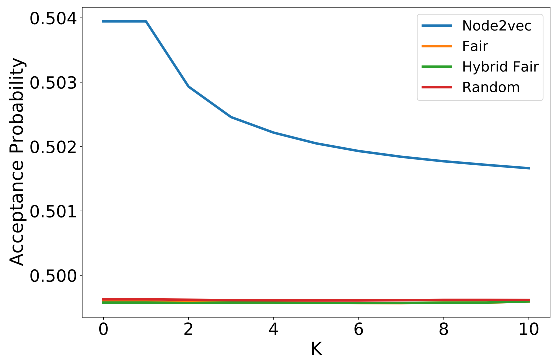
Policy	Distance			PageRank			Red PageRank			Node Homophily		
	mean	median	max	mean	median	max	mean	median	max	mean	median	max
Random	2.809016	3	7	0.000243	0.000331	0.045172	0.000000	0.282878	0.638946	0.000000	0.500000	1.000000
Node2vec	2.313158	2	4	0.000243	0.004722	0.010006	0.161032	0.313908	0.564971	0.000000	0.099480	0.957143
Fair	3.745805	4	7	0.000243	0.000243	0.000583	0.615104	0.620985	0.638946	1.0	1.0	1.0
Hybrid Fair	2.644818	3	5	0.000243	0.001271	0.006028	0.490224	0.555863	0.638946	0.834951	0.970612	1.000000



(a) Blogs



(b) Twitter



(c) DBLP

Figure 5.16: Average Acceptance Probability - Red Source Nodes.

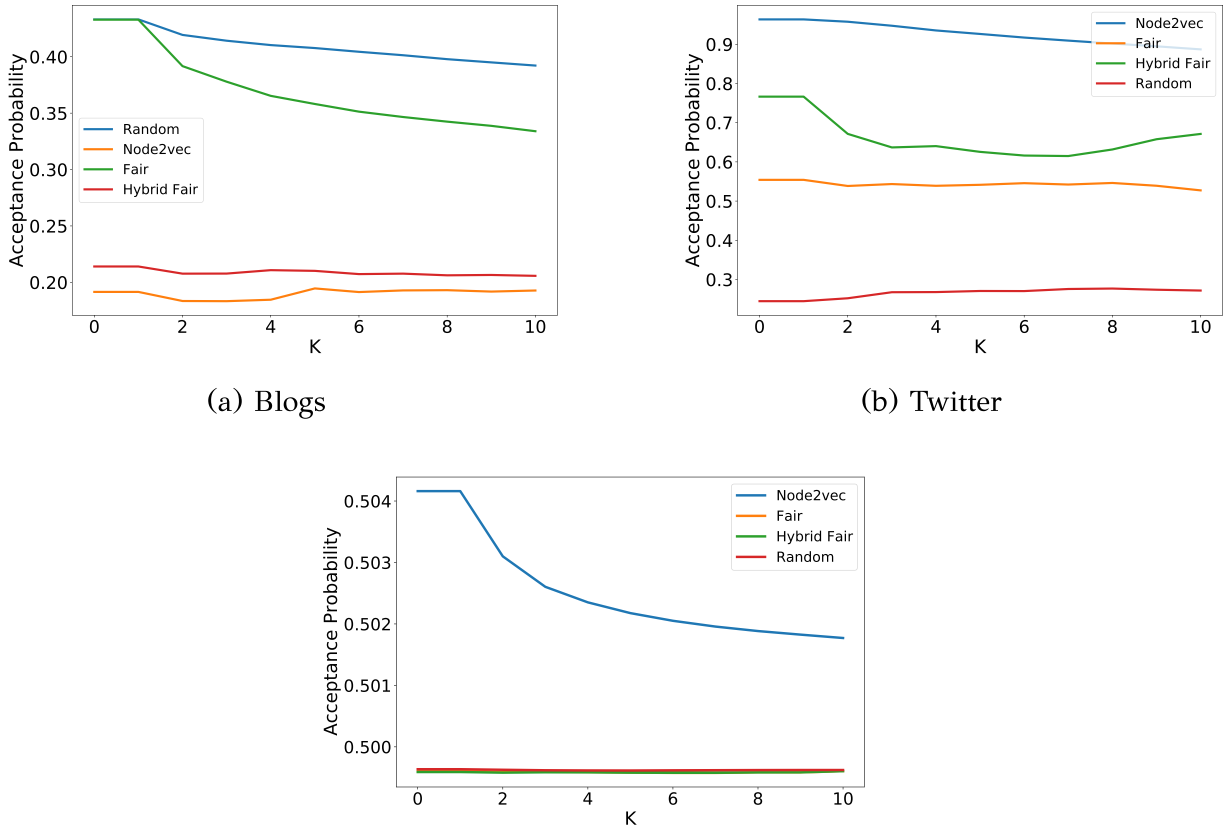
5.4.4 Homophily and Minority Size

We have already seen in figure 5.1 the effect of homophily and minority size to the Red PageRank ratio of a network. A subsequent question is if and how these factors affect the evolution of Red PageRank in a network depending on the the link recommendation policy.

We present in figures 5.23, 5.24 the impact of the different recommendation policies in the evolution of the networks for different degrees of protected group size and different degrees of symmetric and asymmetric same group preference probability.

Table 5.5: Target Quality Features in Twitter Network.

Policy	Distance			PageRank			Red PageRank			Node Homophily		
	mean	median	max	mean	median	max	mean	median	max	mean	median	max
Random	4.98	5	12	0.000054	0.000037	0.003275	0.577001	0.639552	1.0	0.512503	0.5	1.0
Node2vec	3.85	4	11	0.001410	0.001376	0.003275	0.643997	0.733387	0.817765	0.750000	1.0	1.0
Fair	5.27	5	11	0.000185	0.000228	0.000298	0.942876	1.0	1.0	1.0	1.0	1.0
Hybrid Fair	4.87	5	11	0.000457	0.000283	0.001412	0.935682	1.0	1.0	1.0	1.0	1.0



(a) Blogs

(b) Twitter

(c) DBLP

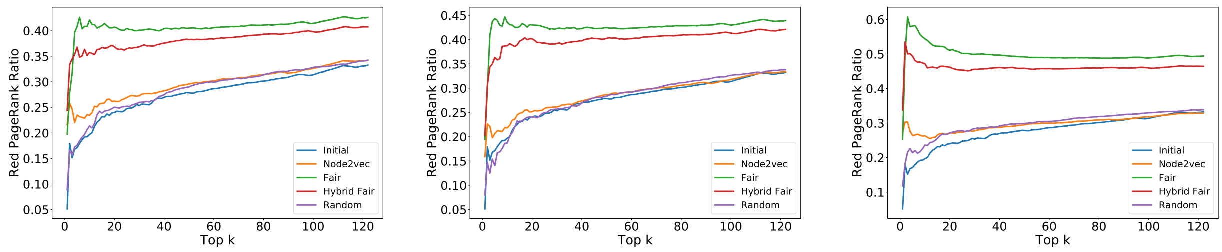
Figure 5.17: Average Acceptance Probability - Blue Source Nodes.

Synthetic network confirm the behavior we show before that node2vec doesn't change the network's red PageRankratio. Fair policies they both enhance protected group as expected but we see that their dynamics are affected both by the size and the same group preference probability. The general rule that applies to both forth mentioned quantities s the greater the values the greater the impact.

5.4.5 Batch vs Online Gain

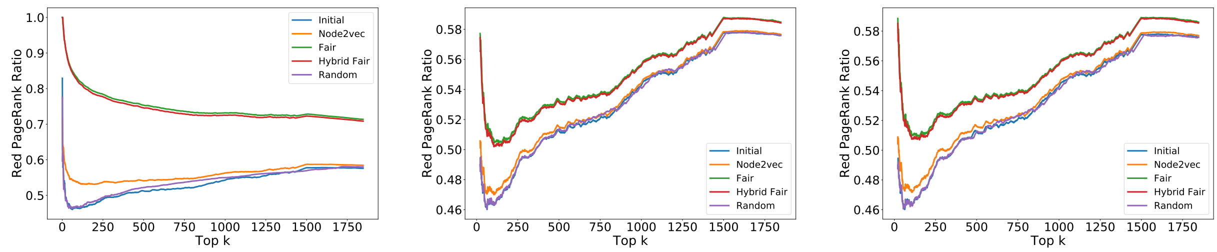
Fair policy exhibits satisfactory results but we can understand from the formula that it doesn't take into consideration the changes in the network's structure that happens from the addition of edges on other nodes. We conduct the basic experiment with a greedy algorithm recalculating the fair score before any recommendation. Figures 5.25 5.26

We see that in many cases the greedy algorithm can extend in grade level the



(a) Random (b) Red (c) Blue

Figure 5.18: Red PageRank ratio at top k nodes by PageRank - Blogs network.



(a) Random (b) Red (c) Blue

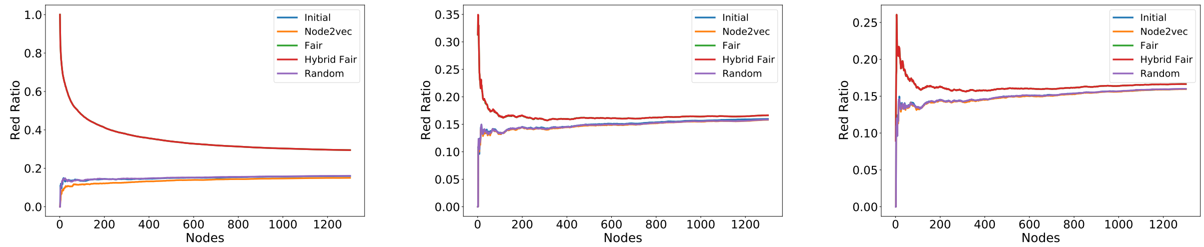
Figure 5.19: Red PageRank ratio at top k by nodes PageRank - Twitter network.

impact on the network but there are also cases where the two algorithms perform equivalently. This happens when the changes in the network doesn't change the order of best targets per source.

5.4.6 Fairness, Accepted Probability Correlation

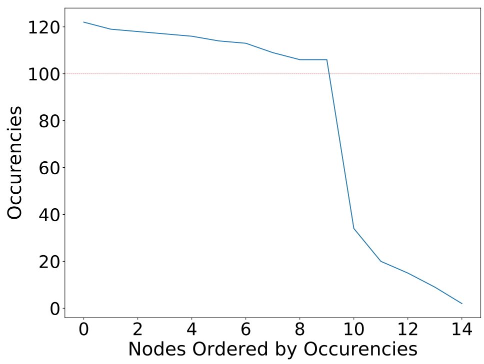
From the experiments so far it is obvious that it is difficult to combine high recommendation score with high fair impact. In fact, the next experiments show us that there is also a negative correlation between acceptance probability and fair score on the top suggestions of each.

To measure that we use the recommendations of node2vec and fair policy we had for the synthetic networks. From these sets we keep the best 50%. For the edges proposed from node2vec recommender we separate them in buckets of equal size and we plot the average fair score for all buckets. We create the corresponding plots for the edges proposed from fair recommender. The number and the size of buckets differs in every occasion depending on the range of the values. Results are presented in figure 5.27

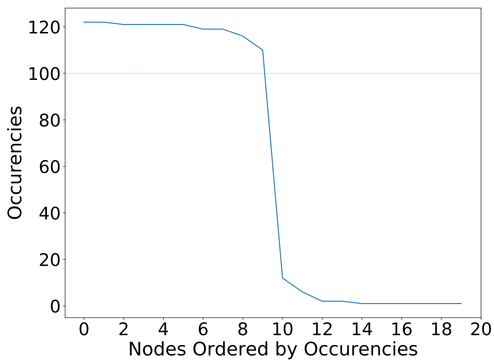


(a) Random (b) Red (c) Blue

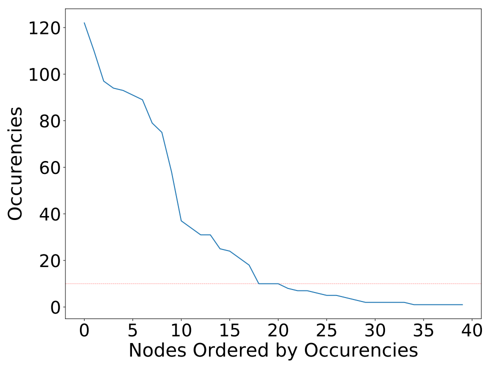
Figure 5.20: Red PageRank ratio at top k by nodes PageRank - DBLP network.



(a) Node2vec

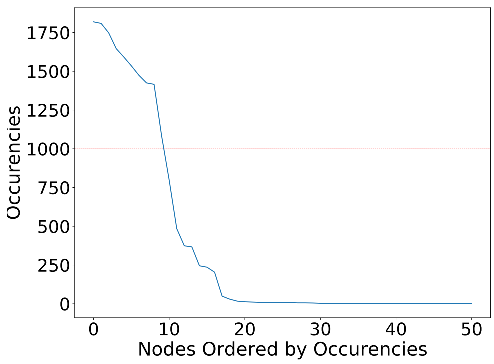


(b) Fair

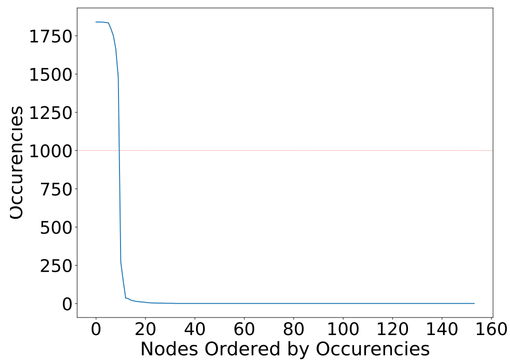


(c) Hybrid Fair

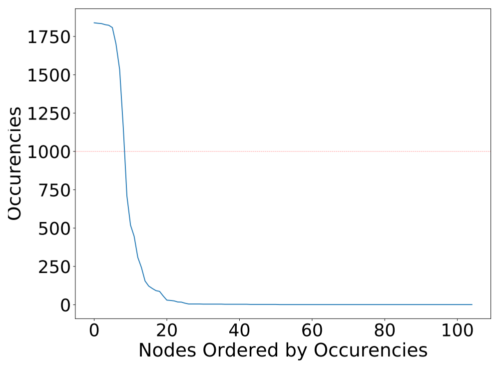
Figure 5.21: Cutting Point for Selecting Nodes.



(a) Node2vec

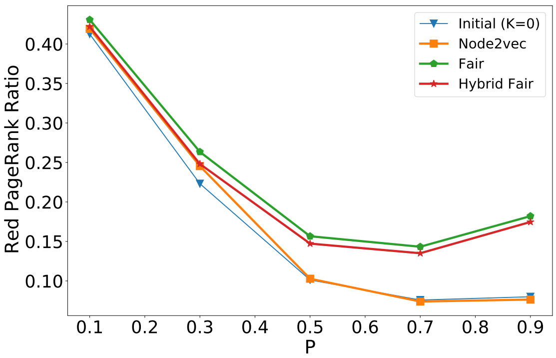


(b) Fair

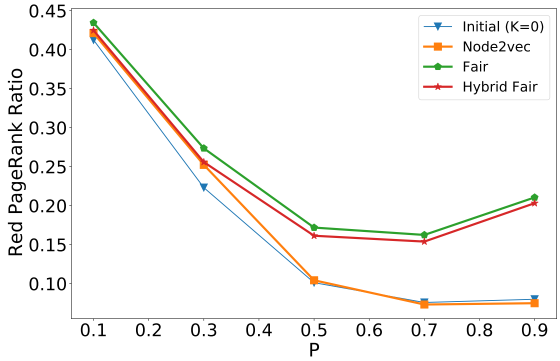


(c) Hybrid Fair

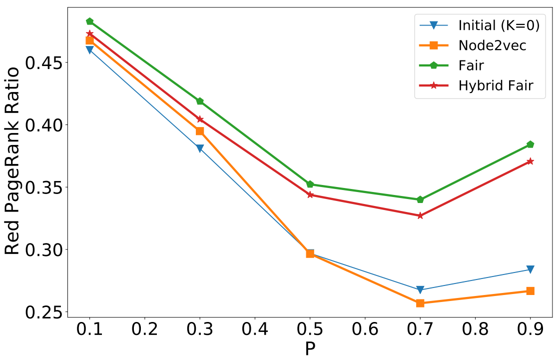
Figure 5.22: Cutting Point for Selecting Nodes



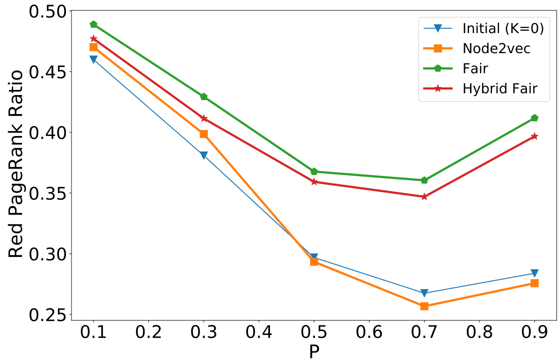
(a) Size: 0.1, K = 5



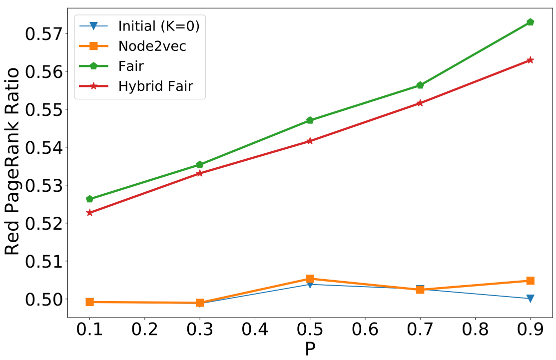
(b) Size: 0.1, K = 10



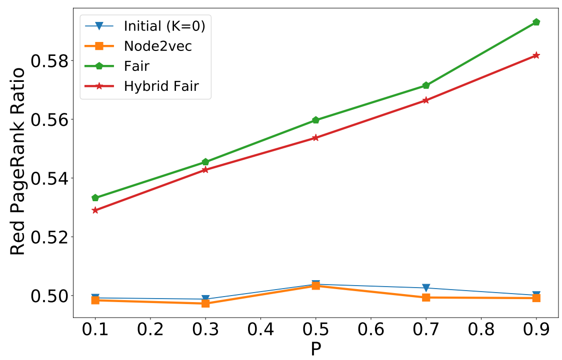
(c) Size: 0.3, K = 5



(d) Size: 0.3, K = 10

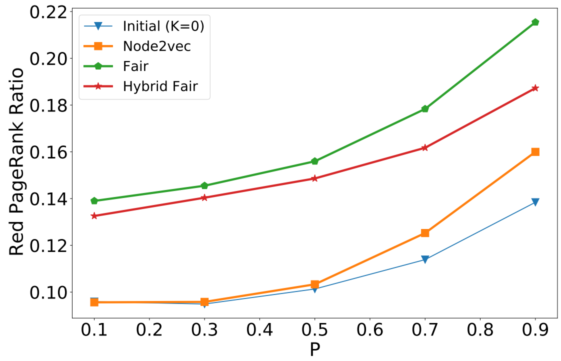


(e) Size: 0.5, K = 5

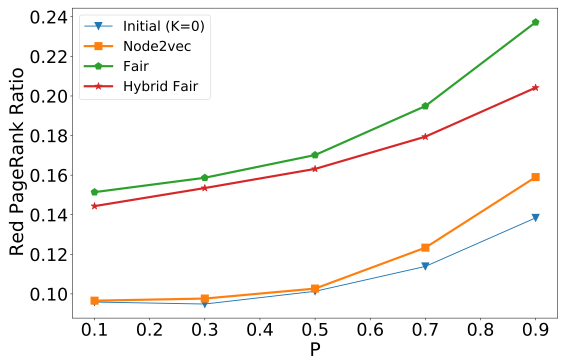


(f) Size: 0.5, K = 10

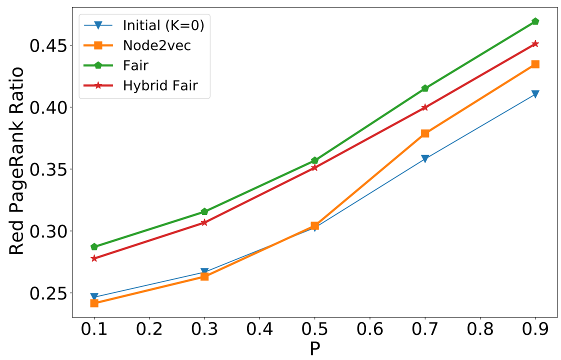
Figure 5.23: Red PageRank ratio to different same group preference probability for sizes 0.1, 0.3, 0.5 after 5 and 10 link additions for symmetric same group preference.



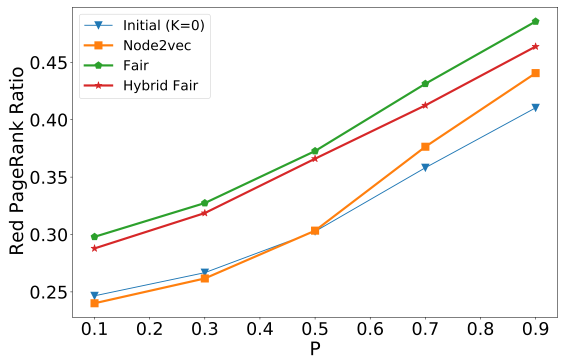
(a) Size: 0.1, K = 5



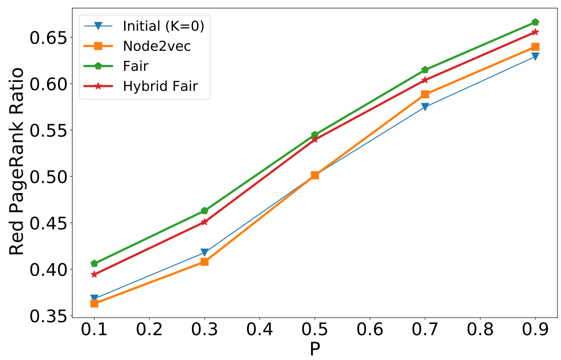
(b) Size: 0.1, K = 10



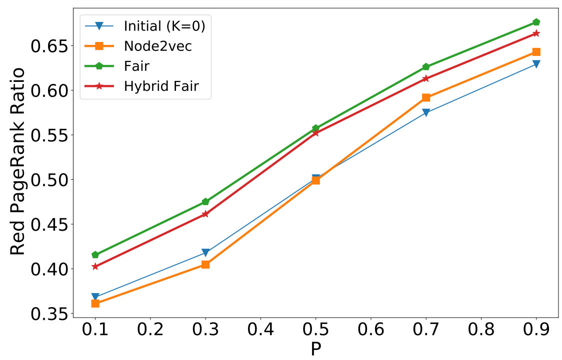
(c) Size: 0.3, K = 5



(d) Size: 0.3, K = 10

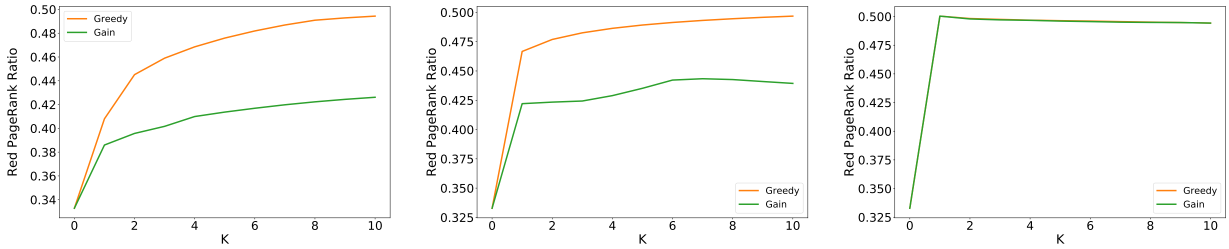


(e) Size: 0.5, K = 5



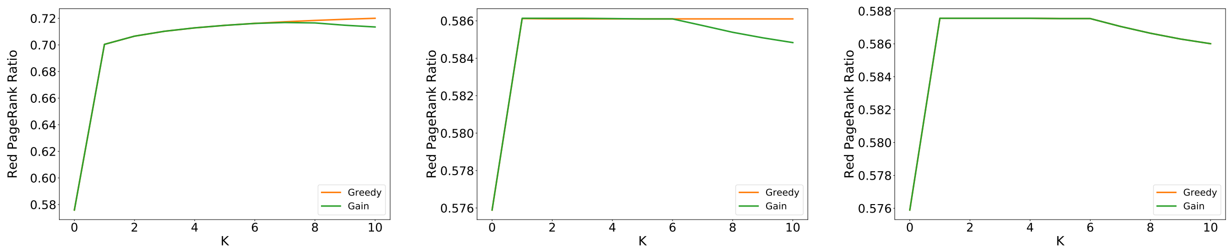
(f) Size: 0.5, K = 10

Figure 5.24: Red PageRank ratio to different same group preference probability for sizes 0.1, 0.3, 0.5 after 5 and 10 link additions for asymmetric same group preference.



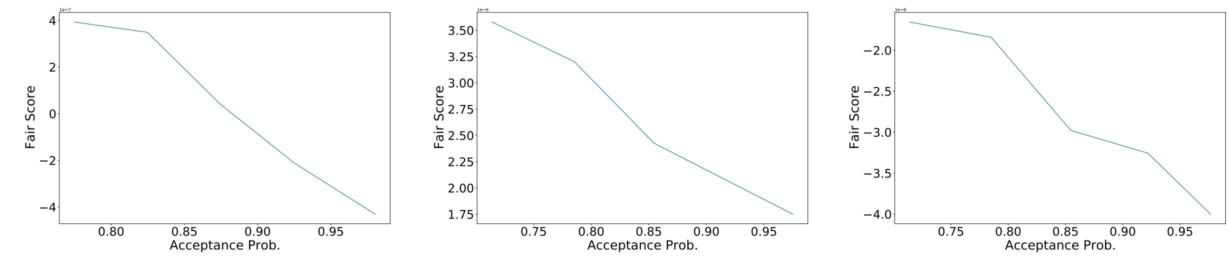
(a) Random (b) Red (c) Blue

Figure 5.25: Fairness Impact for Batch and Online Fair Policy - Blogs.

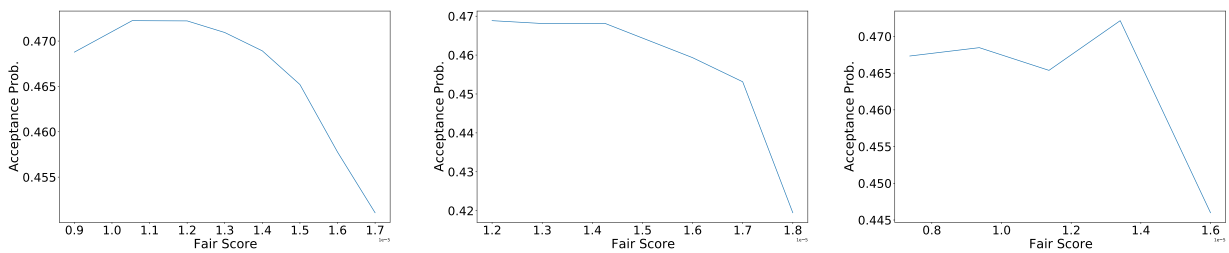


(a) Random (b) Red (c) Blue

Figure 5.26: Fairness Impact for Batch and Online Fair Policy - Twitter.



(a) Random (b) Red (c) Blue



(d) Random (e) Red (f) Blue

Figure 5.27: Recommendation Score - Fair Score Correlation.

Chapter 6

Related Work

Algorithmic fairness. Recently, there has been increasing interest in algorithmic fairness, especially in the context of machine learning. Fairness is regarded as the lack of discrimination on the basis of some protective attribute. Various definition of fairness having proposed especially for classification [15, 1, 23, 24]. We use a group-fairness definition, based on parity. Approaches to handing fairness can be classified as *pre-processing*, that modify the input data, *in-processing*, that modify the algorithm and *post-processing* ones, that modify the output. We are mostly interested in in-processing techniques.

There is also prior work on fairness in ranking [25, 26, 27, 28]. All of these works consider ranking as an ordered list of items, and use different rules for defining and enforcing fairness that consider different prefixes of the ranking [25, 26], pair-wise orderings [28], or exposure and presentation bias [29, 27].

Our goal in this paper is not to propose a new definition of ranking fairness, but rather to initiate a study of fairness in link analysis. A distinguishing aspect of our approach is that we take into account the actual Pagerank weights of the nodes, not just their ranking. Furthermore, our focus in this paper, is to design in-processing algorithms that incorporate fairness in the inner working of the Pagerank algorithm. We present a post-processing approach as a means to estimate a lower bound on the utility loss. None of the previous approaches considers ranking in networks, so the proposed approaches are novel.

Fairness in networks. There has been some recent work on network fairness in the context of graph embeddings [30, 31, 32]. The work in [30] follows an in-processing approach that extends the learning function with regulatory fairness enforcing terms, while the work in [31] follows a post-processing approach so as to promote link recommendations to nodes belonging to specific groups. Both works are not related to our approach. The work in [32] extends the node2vec graph embedding method by modifying the random walks used in node2vec with fair walks, where nodes are partitioned into groups and each group is given the same probability of being selected when a node makes a transition. The random walk introduced in [32] has some similarity with the random walk interpretation of LFPR_N. It would be interesting to see, whether our extended residual-based algorithms could be utilized also in the context of graph embeddings, besides its use in link analysis.

There are also previous studies on the effect of homophily, preferential attachment and differences in group sizes. It was shown that the combination of these three factors leads to uneven degree distributions between groups [8]. Evidence of this kind of inequality - like between degree distribution of minorities and majorities - was also found in many real networks [33]. Our work extends this line of research by looking at Pagerank values instead of degrees.

We note here that there is previous work on diversity in network ranking. In this line of research, the goal is to find important nodes that also maximally cover the nodes in the network [34, 35]. Our problem is fundamentally different, since we look for rankings that follow a parity constraint.

Fair recommendation systems. Recent work also shows that this phenomenon is exaggerated by many link recommendation algorithms [9]. Existing works have shown way to update the results of the PageRank [36], [37] but they focus on efficient computations methods. There are studies providing similar analysis towards predicting the impact of new edges based a particular objective [10], [12] but their problem is complete different and their analysis not applicable in our context. Besides that, our analysis provide an efficient way to compute the impact of the new edges.

We have seen different approaches for fair recommendation systems like [32] that suggest a link recommendations system based on node embedding which have been computed using a fair variation of node2vec, or [38] that proposing a hybrid recommender system using a set of probabilistic soft logic rules. Our approach is complete different as it exploits the fairness score of each edge which we compute based on our perturbation analysis. Last the [31] studies the close related problem of link prediction, introducing a dyadic-level fairness cryterion and provide a post processing approach to promote more heterogenous links.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

In this paper we study the fairness for link analysis algorithms. We give general definitions of fairness, and we focus on fair algorithms for the PageRank algorithm. We considered two approaches, one that modifies the jump vector, and one that imposes a fair behavior per node. We also consider the problem of attaining fairness while minimizing the utility loss of Pagerank. Besides that we examine the possibilities of link recommendation systems to affect the fairness of a network. We present a theoretical analysis on the impact of new edges to the fairness of a network and we propose fair link recommendation systems. Our experiments demonstrate the behavior of our different algorithms. Last we evaluate our recommendation systems under the lens of both impact on fairness and quality of link recommendations.

7.2 Future Work

For link analysis ranking, we would like to study and generalize our fair approaches in other link analysis algorithms than the PageRank. We want to explore the utility of our residual policies in graph embeddings. We also plan to study further their personalized and the targeted versions. Moreover, we would like to explore different objectives for the sensitive algorithms and explore its utility in different contexts. Last we like to explore the possibilities of our link recommendation mechanism in problems that can be modeled similarly, like the mitigation of polarization and the information diffusion in social networks.

Bibliography

[1] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth, “A comparative study of fairness-enhancing interventions in machine learning,” in *FAT**, 2019, pp. 329–338.

[2] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107 – 117, 1998.

[3] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” *J. ACM*, vol. 46, no. 5, p. 604–632, 1999.

[4] R. Lempel and S. Moran, “Salsa: The stochastic approach for link-structure analysis,” *ACM Trans. Inf. Syst.*, vol. 19, no. 2, p. 131–160, 2001.

[5] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri, “Know your neighbors: Web spam detection using the web topology,” in *SIGIR*, 2007.

[6] D. F. Gleich, “Pagerank beyond the web,” *SIAM Review*, vol. 57, no. 3, pp. 321–363, 2015.

[7] T. H. Haveliwala, “Topic-sensitive pagerank,” in *WWW*, 2002.

[8] C. Avin, B. Keller, Z. Lotker, C. Mathieu, D. Peleg, and Y. A. Pignolet, “Homophily and the glass ceiling effect in social networks,” in *ITCS*, 2015, pp. 41–50.

[9] A. Stoica, C. J. Riederer, and A. Chaintreau, “Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity,” in *WebConf*, 2018, pp. 923–932.

[10] V. Amelkin and A. K. Singh, “Fighting opinion control in social networks via link recommendation,” in *Proceedings of the 25th ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining, ser. KDD '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 677–685. [Online]. Available: https://doi.org/10.1145/3292500.3330960

[11] F. Masrour, T. Wilson, H. Yan, P. Tan, and A. Esfahanian, "Bursting the filter bubble: Fairness-aware network link prediction," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 2020, pp. 841–848. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/5429

[12] K. Avrachenkov and N. Litvak, "The effect of new links on google pagerank," *Stochastic Models*, vol. 22, no. 2, pp. 319–331, 2006. [Online]. Available: https://doi.org/10.1080/15326340600649052

[13] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 855–864. [Online]. Available: https://doi.org/10.1145/2939672.2939754

[14] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *KDD*, 2015, pp. 259–268.

[15] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, "Fairness through awareness," in *Innovations in Theoretical Computer Science*, 2012, pp. 214–226.

[16] N. Parotsidis, E. Pitoura, and P. Tsaparas, "Centrality-aware link recommendations," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, ser. WSDM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 503–512. [Online]. Available: https://doi.org/10.1145/2835776.2835818

[17] K. S. Miller, "On the inverse of the sum of matrices," *Mathematics Magazine*, vol. 54, no. 2, pp. 67–72, 1981. [Online]. Available: https://doi.org/10.1080/0025570X.1981.11976898

[18] C. M. Grinstead and J. L. Snell, *Introduction to Probability*. AMS, 2003. [Online]. Available: http://www.dartmouth.edu/~chance/teaching_aids/books_articles/probability_book/book.html

[19] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: http://networkrepository.com

[20] L. A. Adamic and N. S. Glance, "The political blogosphere and the 2004 U.S. election: divided they blog," in *LinkKDD*, 2005.

[21] D. A. Easley and J. M. Kleinberg, *Networks, Crowds, and Markets*. Cambridge University Press, 2010.

[22] Q. V. Liao and W. Fu, "Can you hear me now?: mitigating the echo chamber effect by source position indicators," in *CSCW*, 2014, pp. 184–196.

[23] A. Narayanan, "21 fairness definitions and their politics. (tutorial)," in *FAT*, 2018.

[24] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *CoRR*, vol. abs/1908.09635, 2019. [Online]. Available: http://arxiv.org/abs/1908.09635

[25] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates, "Fa*ir: A fair top-k ranking algorithm," in *CIKM*, 2017.

[26] K. Yang and J. Stoyanovich, "Measuring fairness in ranked outputs," in *SSDBM*, 2017.

[27] A. J. Biega, K. P. Gummadi, and G. Weikum, "Equity of attention: Amortizing individual fairness in rankings," in *SIGIR*, 2018, pp. 405–414.

[28] A. Beutel, J. Chen, T. Doshi, H. Qian, L. Wei, Y. Wu, L. Heldt, Z. Zhao, L. Hong, E. H. Chi, and C. Goodrow, "Fairness in recommendation ranking through pairwise comparisons," in *KDD*, 2019, pp. 2212–2220.

[29] A. Singh and T. Joachims, "Fairness of exposure in rankings," in *KDD*, 2018.

[30] A. J. Bose and W. Hamilton, "Compositional fairness constraints for graph embeddings," in *ICML*, 2019, pp. 715–724.

[31] F. Masrour, T. Wilson, H. Yan, P.-N. Tan, and A.-H. Esfahanian, "Bursting the filter bubble: Fairness-aware network link prediction," in *AAAI*, 2020.

[32] T. A. Rahman, B. Surma, M. Backes, and Y. Zhang, "Fairwalk: Towards fair graph embedding," in *IJCAI*, 2019, pp. 3289–3295.

[33] F. Karimi, M. Génois, C. Wagner, P. Singer, and M. Strohmaier, "Homophily influences ranking of minorities in social networks," *Nature Scientific Reports*, vol. 8, 2018.

[34] X. Zhu, A. B. Goldberg, J. V. Gael, and D. Andrzejewski, "Improving diversity in ranking using absorbing random walks," in *HLT-NAACL*, 2007, pp. 97–104.

[35] Q. Mei, J. Guo, and D. R. Radev, "Divrank: the interplay of prestige and diversity in information networks," in *KDD*, 2010, pp. 1009–1018.

[36] A. Langville and C. Meyer, "Updating markov chains with an eye on google's pagerank," *Society for Industrial and Applied Mathematics*, vol. 27, pp. 968–987, 01 2006.

[37] C. D. M. Jr and J. M. Shoaf, "Updating finite markov chains by using techniques of group matrix inversion," *Journal of Statistical Computation and Simulation*, vol. 11, no. 3-4, pp. 163–181, 1980.

[38] G. Farnadi, P. Kouki, S. K. Thompson, S. Srinivasan, and L. Getoor, "A fairness-aware hybrid recommender system," *CoRR*, vol. abs/1809.09030, 2018. [Online]. Available: http://arxiv.org/abs/1809.09030

[39] L. Takac and M. Zábovský, "Data analysis in public social networks," in *International Scientific Conference and International Workshop Present Day Trends of Innovations, Lomza, Poland*, 2012.

[40] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *KDD*, 2008, pp. 990–998.

[41] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "COSNET: connecting heterogeneous social networks with local and global consistency," in *KDD*, 2015, pp. 1485–1494.

Appendix A

Proofs

A.1 Fairness Aware PageRank Ranking

When $\phi = r$, the average weight of red nodes is equal with the average weight of the blue nodes, i.e., $\frac{\sum_{v \in R} w_v}{|R|} = \frac{\sum_{v \in B} w_v}{|B|}$.

Proof. It holds: $\frac{\sum_{v \in B} w_v}{|B|} = \frac{\sum_{v \in V} w_v - \sum_{v \in R} w_v}{|N| - |R|} = \frac{1/r \sum_{v \in R} w_v - \sum_{v \in R} w_v}{|N| - |R|} = \frac{N/|R| \sum_{v \in R} w_v - \sum_{v \in R} w_v}{|N| - |R|} = \frac{\sum_{v \in R} w_v}{|R|}$. □

Proof of Lemma 3.2.

Proof. From the transition matrix $\mathbf{P}_L$, each node $i \in L_R$ gives a portion $\frac{1-\phi}{out_B(i)}$ of each of its pagerank to its neighbors. The blue neighbors do not get any residual pagerank, thus they get an $1 - \phi$ portion as in the LFPR$_N$ algorithm. Each of the red neighbors gets an additional $\frac{1}{out_R(i)} \delta_R(i) = \frac{1}{out_R(i)} \left(\phi - \frac{(1-\phi)\, out_R(i)}{out_B(i)}\right)$, which sums to $\frac{\phi}{out_R(i)}$. Thus, the red nodes get an ϕ portion of i's pagerank as in the LFPR$_N$ algorithm. The proof is analogous for each node $i \in L_B$. □

A lower bound for the optimal weight redistribution for targeted fairness

Given the weight vector $\mathbf{w}$ and the set S, we divide the full set of nodes in three categories: the set B_S of blue nodes in S, the set R_S of red nodes in S, and the rest of nodes O not in S. In order for $\mathbf{f}$ to be fair, it must be that it moves weight between these categories. Furthermore, this movement is always in one direction, e.g., all nodes in R_S will increase their weight. It is clearly sub-optimal to increase the weight of some nodes in R_S and decrease the weight of others. We define the variables $x_B = \sum_{i \in B_S}(f_i - w_i)$, $x_R = \sum_{i \in R_S}(f_i - w_i)$, and $x_O = \sum_{i \in O}(f_i - w_i)$ to be the total change in weight for the nodes in B_S, R_S, and O respectively. Note that these values may be positive, indicating an increase in weight for the respective category, or negative, indicating a decrease in weight for the respective category. It holds:

$$x_B + x_R + x_O = 0 \tag{A.1}$$

Let $\mathbf{f}_R$ and $\mathbf{f}_B$ the weight allocated to nodes in R_S and R_B respectively, and ρ and β be their desired values according to ϕ. Also, let w_B and w_R be the weight of the nodes in B_S and R_S respectively. Since the vector $\mathbf{f}$ is fair for the nodes in S it holds that

$$\frac{w_R + x_R}{w_B + x_B} = \frac{\rho}{\beta} \tag{A.2}$$

Using Equations A.1 and A.2, we can express x_B and x_R as a function of x_O:

$$x_R = \rho w_B - \beta w_R - \rho x_O \tag{A.3}$$

$$x_B = \beta w_R - \rho w_B - \beta x_O \tag{A.4}$$

Now, let N_B, N_R, and N_O denote the number of nodes in categories B_S, R_S, and O respectively. To minimize loss, and since we allow $\mathbf{f}$ to have negative entries, the change in weight must be distributed equally in each category. Thus, the total loss is

$$\text{Loss}(\mathbf{f}, \mathbf{w}) = \frac{x_R^2}{N_R} + \frac{x_B^2}{N_B} + \frac{x_O^2}{N_O} \tag{A.5}$$

We substitute Equations A.3 and A.4 in Equation A.5, we take the derivative with respect to x_O, and we set it zero. Solving for x_O, we get:

$$x_O = \frac{N_O(\beta w_B - \rho w_R)(\beta N_R - \rho N_B)}{\rho N_B(\rho N_O + N_R) + \beta N_R(\beta N_O + N_B)} \tag{A.6}$$

Substituting x_O in Equations A.4 and A.3, we obtain:

$$x_R = \frac{(\rho w_B - \beta w_R)N_R(\beta N_O + N_B)}{\rho N_B(\rho N_O + N_R) + \beta N_R(\beta N_O + N_B)} \tag{A.7}$$

$$x_B = \frac{(\beta w_R - \rho w_B)N_B(\rho N_O + N_R)}{\rho N_B(\rho N_O + N_R) + \beta N_R(\beta N_O + N_B)} \tag{A.8}$$

There are some interesting observations in these equations. First, a factor that appears in all equations is $\beta\mathbf{w}_R - \rho\mathbf{w}_B$, which tells us how unfair the original weights are. For example, if $\beta\mathbf{w}_R - \rho\mathbf{w}_B < 0$, then we are unfair towards category R. In this case the nodes in category R will always receive weight $w_R > 0$. The origin of the weight depends on the ratio N_R/N_B of the nodes in S. If $\beta N_R - \rho N_B < 0$, then we have proportionally more nodes of B in S with an excess of weight. In this case we remove weight only from the nodes in B, and we distribute it to the nodes in R and O as defined by Equations A.8 and A.6. If $\beta N_R - \rho N_B < 0$, then we have proportionally less nodes of B in S, but they have proportionally more weight. In this case we remove weight from both the nodes in B, and O, as defined by Equations A.8 and A.6, and we distribute it to the nodes in R. If $\beta N_R - \rho N_B = 0$, then we take weight only from the nodes in B and give only to the nodes in R.

Having computed the values for x_R, x_B and x_O, we can now compute the loss using Equation A.5. Note that this is a lower bound to the optimal loss for our problem, since it does not guarantee that the resulting vector $\mathbf{f}$ has non-negative entries.

A.2 PageRankFair Recommendations

Proof. Detailed proof of Theorem 4.1

Assume that $E' = E \cup \{u, v\}$ then:

$$\mathbf{D}_i = \begin{cases} 0, \; i \neq u \\ -\mathbf{P}_u + \frac{k_u}{k_u+1}\mathbf{P}_u + \frac{1}{k_u+1}\mathbf{e}_v = (-1 + \frac{k_u}{k_u+1})\mathbf{P}_u + \frac{1}{k_u+1}\mathbf{e}_v = \frac{-1}{k_u+1}\mathbf{P}_u + \frac{1}{k_u+1}\mathbf{e}_v, \; i = u \end{cases}$$

$$\mathbf{DQ}_{ij} = \begin{cases} 0,\ i \neq u \\ \sum_{w\in E_u}[\mathbf{D}_{uw}\mathbf{Q}_{wj}] + \mathbf{D}_{uv}\mathbf{Q}_{vj} = \sum_{w\in E_u}[\frac{-1}{k_u(k_u+1)}\cdot \mathbf{Q}_{wj}] + \frac{1}{k_u+1}\cdot \mathbf{Q}_{vj}, \quad i = u \end{cases}$$
$$= \begin{cases} 0,\ i \neq u \\ \frac{1}{k_u+1}[\cdot \mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_{wj}], \quad i = u \end{cases}$$

$$\mathbf{QDQ}_{ij} = \sum_{w=1}^{n}\mathbf{Q}_{iw}[\mathbf{DQ}]_{wj} = \mathbf{Q}_{iu}[\mathbf{DQ}]_{uj} = \frac{1}{k_u+1}\mathbf{Q}_{iu}\big(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_{wj})\big) \tag{A.9}$$

$$q = tr(\mathbf{DQ}) = \sum_{i=0}^{n}\mathbf{DQ}_{ii} = \mathbf{DQ}_{uu} = \frac{1}{k_u+1}\big(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}\mathbf{Q}_{wj})\big) \tag{A.10}$$

We know[17] that If G is nonsingular, H is of rank 1 and $G + H$ is nonsingular as well, then:

$$(G + H)^{-1} = G^{-1} - \frac{1}{1+g}G^{-1}HG^{-1},\ g := tr(HG^{-1})$$

For $G = [\mathbf{I} - (1-c)\ \mathbf{P}]$ and $H = -(1-c)\cdot\mathbf{D}$, we have:

$$\begin{aligned}
\mathbf{N}' &= \mathbf{N} - \frac{1}{1+q}\mathbf{N}(-(1-c)\mathbf{DN}),\ q := tr(-(1-c)\mathbf{DN}) \Rightarrow \\
\frac{1}{c}\mathbf{Q}' &= \frac{1}{c}\mathbf{Q} - \frac{1}{c^2}\frac{1}{1+q}\mathbf{Q}(-(1-c)\cdot\mathbf{D})\mathbf{Q},\ q := tr(-(1-c)\mathbf{D}\frac{1}{c}\mathbf{Q}) \\
&= \frac{1}{c}\mathbf{Q} + \frac{1}{c^2}\frac{(1-c)}{1+q}\mathbf{QDQ},\ q := -\frac{(1-c)}{c}\cdot tr(\mathbf{DQ}) \Rightarrow \\
\mathbf{Q}' &= \mathbf{Q} + \frac{1}{c}\frac{(1-c)}{1-\frac{(1-c)}{c}q}\mathbf{QDQ},\ q := tr(\mathbf{DQ}) \\
&= \mathbf{Q} + \frac{\frac{(1-c)}{c}}{1-\frac{(1-c)}{c}q}\mathbf{QDQ},\ q := tr(\mathbf{DQ}) \Rightarrow
\end{aligned}$$

$$
\begin{aligned}
\mathbf{Q}'_{ij} &= \mathbf{Q}_{ij} + \frac{\frac{(1-c)}{c}}{1-\frac{(1-c)}{c}q}\frac{1}{k_u+1}\mathbf{Q}_{iu}\big(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wj})\big) \\
&= \mathbf{Q}_{ij} + \frac{\mathbf{Q}_{iu}}{k_u+1}\frac{\frac{(1-c)}{c}}{1-\frac{(1-c)}{c}\frac{1}{k_u+1}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wu}))}\big(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wj})\big) \\
&= \mathbf{Q}_{ij} + \mathbf{Q}_{iu}\frac{\frac{(1-c)}{c}\big(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wj})\big)}{k_u+1-\frac{(1-c)}{c}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wu}))}
\end{aligned}
$$

We want to maximize PageRank of Red (R). We know that $\mathbf{p} = \mathbf{vQ}$, so:

$$
\begin{aligned}
\mathbf{p}'(R) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in R}^{n}\mathbf{Q}'_{ij} \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in R}^{n}\mathbf{Q}_{ij} + \mathbf{Q}_{iu}\frac{\frac{(1-c)}{c}\big(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wj})\big)}{k_u+1-\frac{(1-c)}{c}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wu}))} \\
&= \mathbf{p}(R) + \frac{1}{n}\sum_{i=1}^{n}\sum_{j\in R}^{n}\mathbf{Q}_{iu}\frac{\frac{(1-c)}{c}\big(\mathbf{Q}_{vj} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wj})\big)}{k_u+1-\frac{(1-c)}{c}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wu}))} \\
&= \mathbf{p}(R) + \mathbf{p}_u\frac{\frac{(1-c)}{c}\big(\mathbf{Q}_v(R) - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_w(R))\big)}{(k_u+1)-\frac{(1-c)}{c}(\mathbf{Q}_{vu} - \frac{1}{k_u}\sum_{w\in E_u}(\mathbf{Q}_{wu}))} \overset{\text{lemma 1.2, 1.4}}{\Longrightarrow}
\end{aligned}
$$

$$
= \begin{cases}
\mathbf{p}(R) + \mathbf{p}_u\cdot\dfrac{\frac{(1-c)}{c}\big(\mathbf{Q}_v(R) + \frac{c}{1-c} - \frac{1}{1-c}\mathbf{Q}_u(R)\big)}{k_u+1-\frac{(1-c)}{c}(\mathbf{Q}_{vu} + \frac{c}{(1-c)} - \frac{1}{(1-c)}\mathbf{Q}_{uu})}, & u\in R \\[2ex]
\mathbf{p}(R) + \mathbf{p}_u\cdot\dfrac{\frac{(1-c)}{c}\big(\mathbf{Q}_v(R) - \frac{1}{1-c}\mathbf{Q}_u(R)\big)}{k_u+1-\frac{(1-c)}{c}(\mathbf{Q}_{vu} + \frac{c}{(1-c)} - \frac{1}{(1-c)}\mathbf{Q}_{uu})}, & u\in B
\end{cases}
$$

$$
= \begin{cases}
\mathbf{p}(R) + \mathbf{p}_u\cdot\dfrac{\frac{1}{c}\big((1-c)\mathbf{Q}_v(R) + c - \mathbf{Q}_u(R)\big)}{k_u+1-\frac{1}{c}((1-c)\mathbf{Q}_{vu} + c - \mathbf{Q}_{uu})}, & u\in R \\[2ex]
\mathbf{p}(R) + \mathbf{p}_u\cdot\dfrac{\frac{1}{c}\big((1-c)\mathbf{Q}_v(R) - \mathbf{Q}_u(R)\big)}{k_u+1-\frac{1}{c}((1-c)\mathbf{Q}_{vu} + c - \mathbf{Q}_{uu})}, & u\in B
\end{cases}
$$

Assume that $G' = (V, E' := E\cup\widetilde{E}),\ \widetilde{E} = \{(u,v)|v\in V\wedge v\notin E_u\},\ E'_u = E_u\cup\widetilde{E}_u,\ \widetilde{E}_u =$

$\{v|(u,v) \in \widetilde{E}\}$ then:

$$\mathbf{DQ}_ij = \begin{cases} 0, \;\; i \neq u \\ \frac{\widetilde{k}}{k_u + \widetilde{k}}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_{vj}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wj})\big) \end{cases}$$

$$\mathbf{QDQ}_{ij} = \frac{\widetilde{k}}{k_u + \widetilde{k}}\mathbf{Q}_{iu}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_{vj}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wj})\big)$$

$$tr\big(\mathbf{DQ}\big) = \mathbf{DQ}_{uu} = \frac{\widetilde{k}}{k_u + \widetilde{k}}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_{vu}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu})\big)$$

$$\mathbf{Q}'_{ij} = \mathbf{Q}_{ij} + \mathbf{Q}_{iu}\frac{\frac{(1-c)}{c}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_{vj}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wj})\big)}{\frac{k_u+\widetilde{k}}{\widetilde{k}} - \frac{(1-c)}{c}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_{vu}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu})\big)}$$

$$\mathbf{p}'(R) = \mathbf{p}(R) + \mathbf{p}_u \cdot \frac{\frac{(1-c)}{c}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_v(R)) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_w(R))\big)}{\frac{k_u+\widetilde{k}}{\widetilde{k}} - \frac{(1-c)}{c}\big(-\frac{1}{\widetilde{k}}\sum_{w \in \widetilde{E}_u}(\mathbf{Q}_{vu}) - \frac{1}{k_u}\sum_{w \in E_u}(\mathbf{Q}_{wu})\big)}$$

□

Proof. Lemma 4.4

The new transition matrix is:

$$\widetilde{\mathbf{P}} \in \mathbb{R}^{(n+2)x(n+2)} := \widetilde{\mathbf{P}}_{ij} = \begin{cases} (1-c)\mathbf{P}_{ij}, \;\; 0 \leq i,j \leq n \\ c, \;\; (i \in R \wedge j = a_r) \vee (i \in B \wedge j = a_b) \\ 1, \;\; (i = j = a_r) \vee (i = j = a_b) \\ 0, \;\; otherwise \end{cases}$$

$\widetilde{\mathbf{P}}$ can also be written in its canonical form:

$$\widetilde{\mathbf{P}} = \begin{bmatrix} (1-c)\mathbf{P} & \mathbf{R} \\ \mathbf{0}_{2xn} & \mathbf{I}_2 \end{bmatrix}, \quad \mathbf{R} \in \mathbb{R}^{nx2}, \quad \mathbf{R} = \begin{cases} c, \;\; (i \in R \wedge j = 1) \vee (i \in B \wedge j = 2) \\ 0, \;\; otherwise \end{cases}$$

We know[18] that absorption probabilities are $\mathbf{B} = \mathbf{NR}$ where $\mathbf{N}$ is the Fundamental matrix of $\widetilde{X}$ defined as $N = [\mathbf{I}_n - (1-c)\mathbf{P}]^{-1}$.

We observe that $\mathbf{N} = \frac{1}{c}\mathbf{Q}$ So:

$$\begin{cases} \mathbf{B} = \mathbf{N}\mathbf{R} \\ \mathbf{N} = \frac{1}{c}\mathbf{Q} \end{cases} \Rightarrow \begin{cases} \mathbf{B} = \mathbf{Q}\mathbf{R}' \\ \mathbf{R}' = \frac{1}{c}\mathbf{R} \end{cases} \Rightarrow \mathbf{B}_{ij} = \sum_k \mathbf{Q}_{ik}\mathbf{R}'_{kj} = \begin{cases} Q_i(R), \ j = 1 \\ Q_i(B), \ j = 2 \end{cases} , \Rightarrow$$

$$\mathbf{Q}_i(R) = \mathbf{B}_{i1}, \quad \mathbf{Q}_i(B) = \mathbf{B}_{i2}$$

□

Appendix B

Experiment Evaluation

Additional datasets and experiments

In Table B.2, we present statistics for additional datasets.

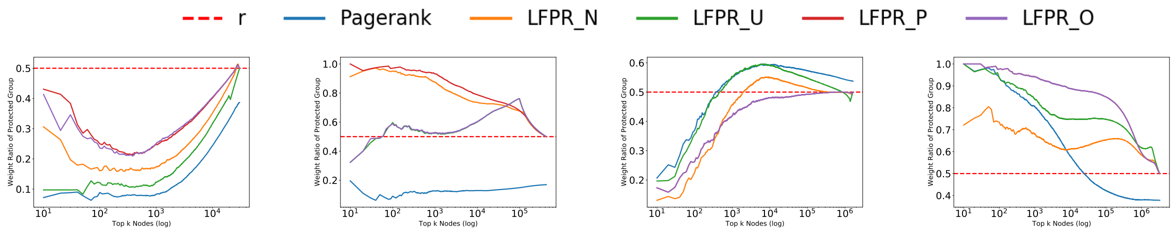
- POKEC [39]: This is a Slovak social network. Nodes correspond to users, and links to friendships. Friendship relations are directed.
- DBLP1: An author collaboration network constructed by the Arnetminer academic search system [40] using publication data from dblp. Two authors are connected if they have co-authored an article.

Table B.1: Utility loss with respect to optimal utility ($\frac{LFPR_X}{OPTIMAL}$)

Dataset	LFPR$_N$	LFPR$_U$	LFPR$_P$	SFPR
POKEC	30.57	35.31	15.39	-
TWITTER	152.08	156.02	66.92	6.94
DBLP1	99.12	41.66	21.80	-
DBLP2	95.84	47.81	25.18	6.13
LINKEDIN	4,913	1,787	1,149	-
PHYSICS	9.56	9.04	8.21	50.24

Table B.2: Real dataset characteristics. r, b relative size of protected and unprotected group, respectively; p_R, p_B pagerank assigned to the red and blue group respectively

Dataset	#nodes	#edges	Protected attribute	r	b	homophily	p_R	p_B
POKEC	1,632,803	30,622,564	gender (women)	0.51	0.49	1.11	0.54	0.46
DBLP1	423,469	2,462,422	gender (women)	0.19	0.81	0.83	0.13	0.87
LINKEDIN	3,209,448	13,016,453	gender (women)	0.37	0.63	0.72	0.37	0.63
PHYSICS	30,359	347,235	year (after 1997)	0.66	0.34	0.76	0.39	0.61



(a) Physics (b) DBLP1 (c) Pokec (d) linkedin

Figure B.1: Locally fair Pagerank algorithms for the additional datasets with $\phi = 0.5$.

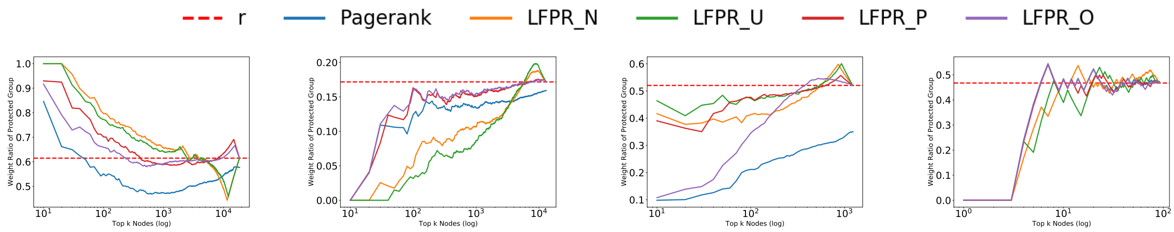
- LINKEDIN [41]: Nodes correspond to LinkedIn profiles. Two profiles are linked if they were co-viewed by the same user.
- PHYSICS: This is the Arxiv HEP-PH (high energy physics phenomenology) citation graph from the SNAP dataset[1]. Nodes correspond to papers and there is an edge from a paper to another, if the first paper cites the second one.

Again, there are cases where the fraction of the weight assigned to the protected group is even smaller than r.v

B.1 Fairness Aware PageRank Ranking

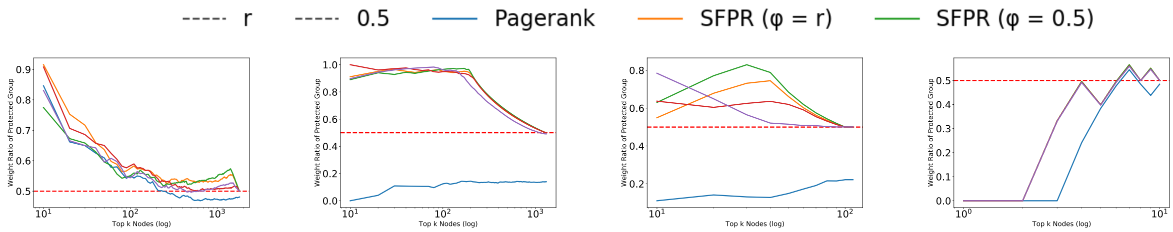
In Figure B.1, we report results for the original and the locally fair PageRank algorithms for the additional datasets and in Figure B.2, we report results for the locally fair PageRank algorithms for $\phi = r$.

[1] http://snap.stanford.edu/data



(a) Twitter (b) DBLP2 (c) Blogs (d) Books

Figure B.2: Locally fair Pagerank algorithms for $\phi = r$.



(a) Twitter (b) DBLP2 (c) Blogs (d) Books

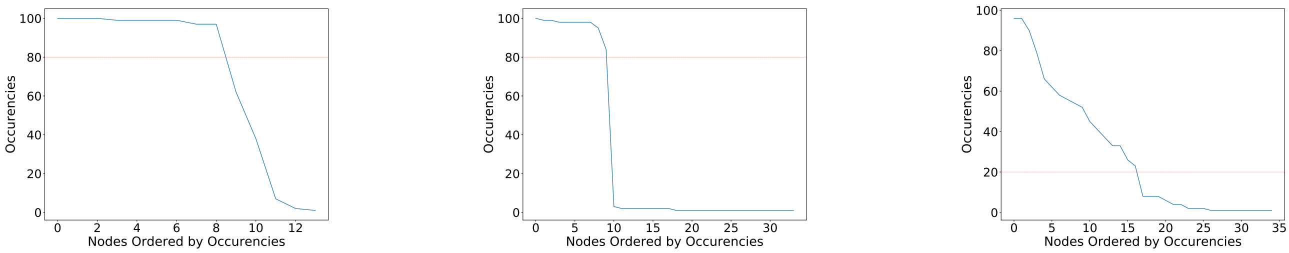
Figure B.3: Targeted locally fair PageRank algorithms and the optimal post-processing redistribution for $\phi = 0.5$. The size of the set S is set to 10% of the size of the dataset.

B.2 PageRankFair Recommendations

Similar results with those in analysis for random source nodes can derived for the other two source nodes sets from the tables B.3, B.4, B.5, B.6. Selection threshold for each set is presented in figures B.4, B.5, B.6, B.7.

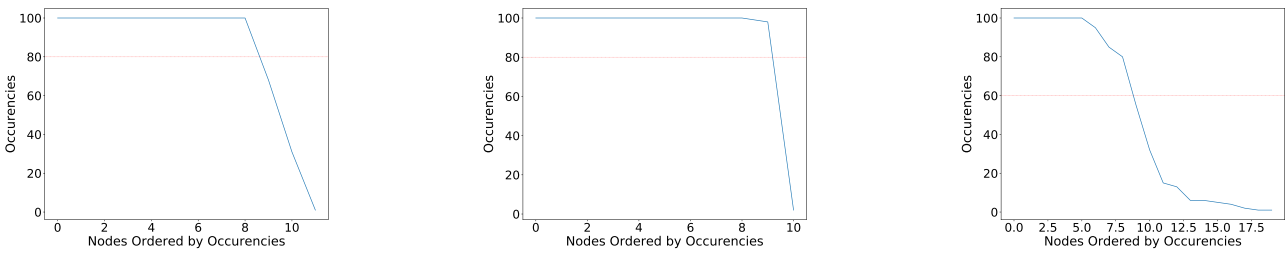
Table B.3: Target Quality Features in Blogs - Red Source Nodes.

Policy	Distance			PageRank			Red PageRank			Node Homophily		
	mean	median	max	mean	median	max	mean	median	max	mean	median	max
Random	3.380000	3	5	0.000822	0.000339	0.045172	0.332817	0.282878	0.638946	0.435733	0.500000	1.000000
Node2vec	2.722222	3	4	0.004934	0.004793	0.010006	0.340760	0.321328	0.564971	0.308273	0.160000	0.957143
Gain	3.890000	4	7	0.000284	0.000243	0.000583	0.622608	0.620985	0.638946	1.000000	1.000000	1.000000
ExpGain	3.040816	3	5	0.000940	0.000583	0.002620	0.580573	0.590254	0.638946	0.969143	1.000000	1.000000



(a) Node2vec (b) Fair (c) Hybrid Fair

Figure B.4: Cutting Point for Selecting Nodes.

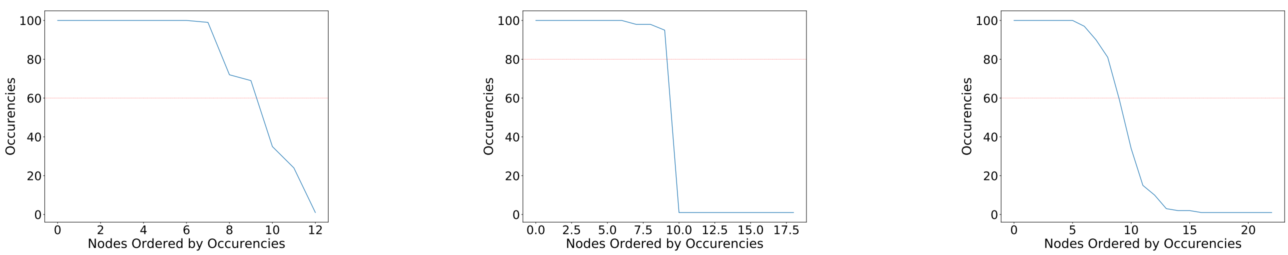


(a) Node2vec (b) Fair (c) Hybrid Fair

Figure B.5: Cutting Point for Selecting Nodes.

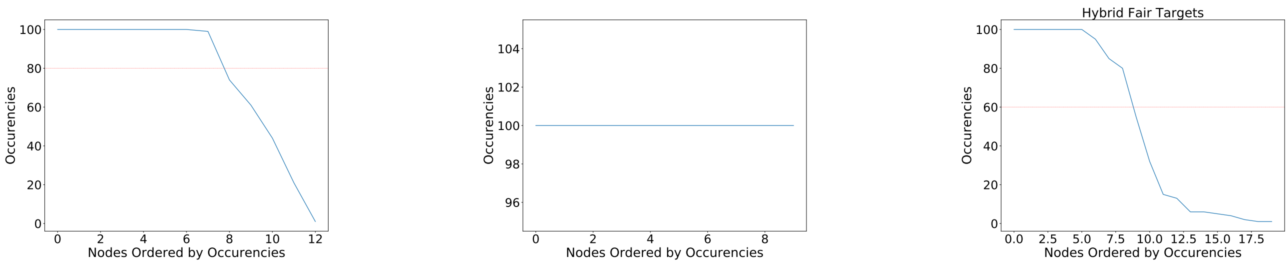
Table B.4: Target Quality Features in Blogs - Blue Source Nodes.

Policy	Distance			PageRank			Red PageRank			Node Homophily		
	mean	median	max	mean	median	max	mean	median	max	mean	median	max
Random	2.846667	3	6	0.000840	0.000331	0.045172	0.337866	0.282878	0.638946	0.445203	0.500000	1.000000
Node2vec	2.437037	2	4	0.004934	0.004793	0.010006	0.340760	0.321328	0.564971	0.308273	0.160000	0.957143
Gain	4.200000	4	7	0.000284	0.000243	0.000583	0.622608	0.620985	0.638946	1.000000	1.000000	1.000000
ExpGain	3.160839	3	5	0.000982	0.000742	0.002620	0.578018	0.577612	0.638946	0.967215	0.989131	1.000000



(a) Node2vec (b) Fair (c) Hybrid Fair

Figure B.6: Cutting Point for Selecting Nodes.



(a) Node2vec (b) Fair (c) Hybrid Fair

Figure B.7: Cutting Point for Selecting Nodes.

Table B.5: Target Quality Features in Twitter - Red Source Nodes.

Policy	Distance			PageRank			Red PageRank			Node Homophily		
	mean	median	max	mean	median	max	mean	median	max	mean	median	max
Random	4.762500	5	7	0.000054	0.000037	0.001418	0.579776	0.639552	1.000000	0.511345	0.500000	1.000000
Node2vec	3.636364	4	7	0.001447	0.001376	0.003275	0.590534	0.721374	0.817765	0.650000	1.000000	1.000000
Gain	4.775000	5	7	0.000185	0.000228	0.000298	0.942876	1.000000	1.000000	1.000000	1.000000	1.000000
ExpGain	4.442857	4	7	0.000457	0.000283	0.001412	0.935682	1.000000	1.000000	1.000000	1.000000	1.000000

Table B.6: Target Quality Features in Twitter - Blue Source Nodes.

Policy	Distance			PageRank			Red PageRank			Node Homophily		
	mean	median	max	mean	median	max	mean	median	max	mean	median	max
Random	4.754545	5	7	0.000058	0.000037	0.001418	0.575291	0.639552	1.000000	0.511240	0.500000	1.000000
Node2vec	3.829545	4	5	0.001574	0.001376	0.003275	0.651073	0.736853	0.817765	0.750000	1.000000	1.000000
Gain	5.372727	5	7	0.000185	0.000228	0.000298	0.942876	1.000000	1.000000	1.000000	1.000000	1.000000
ExpGain	4.904255	5	7	0.000457	0.000283	0.001412	0.935682	1.000000	1.000000	1.000000	1.000000	1.000000

Short Biography

Sotiris Tsioutsiouliklis was born in Ioannina, Greece in 1989. He received his B.Sc. degree from the Mathematics department of the university of Ioannina in 2018. From 2018 he is a M.Sc. student at the Computer Science & Engineering department of the university of Ioannina. His research interests include graph mining and social network analysis.