Modularity-Based Fairness in Network Communities

A Thesis

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

by

Konstantinos Manolis

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
School of Engineering
Ioannina 2024

Examining Committee:

- **Evangelia Pitoura**, Professor, Department of Computer Science and Engineering, University of Ioannina (Advisor)
- **Panagiotis Tsaparas**, Assoc. Professor, Department of Computer Science and Engineering, University of Ioannina
- **Aristidis Likas**, Professor, Department of Computer Science and Engineering, University of Ioannina

Dedication

This thesis is dedicated to those who have been the pillars of support and love throughout my journey, not only during this postgraduate program but throughout all my 26 years.

To my family, who have nurtured me with love and support that has shaped the person I am today. Your sacrifices, encouragement, and unwavering belief in me have been my foundation and strength. Your presence is a constant reminder of what it means to love unconditionally and strive unrelentingly.

To my partner, whose endless emotional and physical support throughout this postgraduate program has been nothing short of extraordinary. Your patience, understanding, and companionship have been my sanctuary. Your belief in me and my aspirations has been a source of motivation and courage, pushing me to surpass my limits and achieve beyond my dreams.

To my friends, who have illuminated the darkest of days with their unwavering support and infectious optimism. Your ability to make difficult days seem brighter and challenges more surmountable has been a beacon of hope and joy in my life. The laughter, conversations, and moments of solace we've shared have been invaluable in this journey.

This dedication is a tribute to all of you, for without your love, support, and belief, this journey would have been a far different experience. Thank you for being my light during the darkest times and my anchor throughout the storms. This achievement is as much yours as it is mine.

Acknowledgements

I extend my deepest gratitude to my supervisor, Professor Evangelia Pitoura, for her invaluable guidance, patience, and unwavering support throughout the entirety of my thesis preparation and execution. Her expertise and insightful feedback were instrumental in shaping both the direction and substance of this research. Her dedication to excellence and her commitment to nurturing my academic and professional growth have left an indelible mark on my journey, for which I am profoundly grateful.

I am also immensely thankful to Christos Gkartzios for his substantial contributions and the robust support he has offered me. His expertise and constructive suggestions have significantly enriched my work, providing me with a broader perspective and deepening my understanding of the subject matter. His encouragement and readiness to assist at every turn have been sources of motivation during challenging moments.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Abstract

Konstantinos Manolis, M.Sc. in Data and Computer Systems Engineering, Department of Computer Science and Engineering, School of Engineering, University of Ioannina, Greece, 2024.
Modularity-Based Fairness in Network Communities.
Advisor: Evangelia Pitoura, Professor.

In this thesis, we study the fairness of community structures in networks from a group-based perspective. Specifically, we assume that individuals in a social network belong to different groups based on the value of one of their sensitive attributes, such as their age, gender, or race. We view community fairness as the lack of discrimination towards any of the groups. For simplicity, let us assume that nodes belong to two groups, the blue and the red group. We introduce three fairness metrics. The first metric, termed balance-fairness, equitably represents communities by ensuring an equal distribution of red and blue nodes in each community. The second, termed modularity-fairness, refines the notion of modularity to demand equal intra-community connectivity for the groups. The third metric, termed diversity-fairness, promotes intra-community edges between nodes of different color thus addressing the filter-bubble phenomenon. We have modified the Louvain algorithm, a well-known community detection algorithm, to produce communities that are both well-connected and fair. We present an extensive evaluation using several real-world and synthetic networks. The goal of our evaluation is twofold: (1) to study the fairness of communities in networks and the causes of unfairness and (2) to evaluate the effectiveness of our fairness-enhanced Louvain algorithm.

Εκτεταμενη Περιληψη

Κωνσταντίνος Μανώλης, Δ.Μ.Σ. στη Μηχανική Δεδομένων και Υπολογιστικών Συστημάτων, Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Πολυτεχνική Σχολή, Πανεπιστήμιο Ιωαννίνων, 2024.
Δικαιοσύνη βασισμένη στην αρθρωτότητα σε δικτυακές κοινότητες.
Επιβλέπων: Ευαγγελία Πιτουρά, Καθηγήτρια.

Στην παρούσα διατριβή, μελετάμε τη δικαιοσύνη των κοινοτήτων στα δίκτυα από μια οπτική γωνία που βασίζεται σε ομάδες. Συγκεκριμένα, υποθέτουμε ότι τα άτομα σε ένα κοινωνικό δίκτυο ανήκουν σε διαφορετικές ομάδες με βάση την τιμή ενός από τα ευαίσθητα χαρακτηριστικά τους, όπως η ηλικία, το φύλο ή η φυλή τους. Θεωρούμε τη δικαιοσύνη της κοινότητας ως την έλλειψη διακρίσεων προς οποιαδήποτε από τις ομάδες. Για λόγους απλότητας, ας υποθέσουμε ότι οι κόμβοι ανήκουν σε δύο ομάδες, την μπλε και την κόκκινη ομάδα. Εισάγουμε τρεις μετρικές δικαιοσύνης. Η πρώτη μετρική, που ονομάζεται ισορροπία-δικαιοσύνη, αντιπροσωπεύει δίκαια τις κοινότητες εξασφαλίζοντας την ισοκατανομή των κόκκινων και μπλε κόμβων σε κάθε κοινότητα. Η δεύτερη, που ονομάζεται modularity-fairness, βελτιώνει την έννοια της αρθρωτότητας ώστε να απαιτεί ίση συνδεσιμότητα εντός της κοινότητας για τις ομάδες. Η τρίτη μετρική, που ονομάζεται δικαιοσύνη ποικιλομορφίας, προάγει τις ενδοκοινοτικές ακμές μεταξύ κόμβων διαφορετικού χρώματος, αντιμετωπίζοντας έτσι το φαινόμενο του φίλτρου-φούσκας. Τροποποιήσαμε τον αλγόριθμο Louvain, έναν γνωστό αλγόριθμο ανίχνευσης κοινοτήτων, για να παράγουμε κοινότητες που είναι τόσο καλά συνδεδεμένες όσο και δίκαιες. Παρουσιάζουμε μια εκτεταμένη αξιολόγηση χρησιμοποιώντας διάφορα δίκτυα του πραγματικού κόσμου και συνθετικά δίκτυα. Ο στόχος της αξιολόγησής μας είναι διττός: (1) να μελετήσουμε τη δικαιοσύνη των κοινοτήτων στα δίκτυα και τις αιτίες της αδικίας και (2) να αξιολογήσουμε την αποτελεσματικότητα του ενισχυμένου με δικαιοσύνη αλγορίθμου Louvain.

Chapter 1

Introduction

Social networks play a pivotal role in shaping opinions and influencing decision-making processes. However, despite the extensive research on algorithmic fairness, the fairness issues stemming from the interconnection between individuals in a network have received comparatively less attention [1]. In this paper, we examine the community structures formed within social networks through the lens of fairness.

We take a group-based approach in which we assume that individuals in a social network belong to one of two groups based on the value of one of their attributes, for example their gender, age, or race. Most previous work on group-based fairness in clustering takes a representation based approach that asks that a sufficient percentage of nodes in each cluster belongs to the protected group [2, 3, 4].

In this paper, we introduce a novel fairness notion for communities, termed *modularity fairness*. Modularity is a characterization of the quality of communities based on the connectivity of each community. Specifically, high values of modularity indicate that there are many edges within communities and few edges between them [5]. The proposed modularity fairness asks that the nodes of the protected group in each community are well-connected, that is, in each community, the nodes that belong to the protected group have many intra-community edges and few inter-community ones.

We also introduce a fairness notion for communities, termed *diversity fairness*, inspired by the work of [6] which addresses the filter bubble problem in link predictions. Filter bubble is a common problem in social network where an individual's exposure to information on the internet becomes limited to ideas and viewpoints similar to their own. This occurs because algorithms predict and select content based on past behavior, effectively isolating users from contrasting perspectives. This can reinforce biases and reduce the diversity of content encountered, leading to a narrower understanding of the world. Diversity fairness evaluates the variety within a community by examining the connections between differently categorized nodes, such as those labeled by different colors. Enhancing the number of connections between nodes of distinct colors within the community leads to greater diversity.

Then, we provide insights of the balance, modularity fairness and diversity fairness present in real networks by evaluating the balance, modularity fairness and diversity fairness of several real world networks. In addition, we seek to explore the confounding factors that may lead to unfairness. Previous research has shown that the relative size of the groups and homophily (i.e., the tendency of nodes to connect with similar nodes) affect various properties in the network, such as the degree and Pagerank distribution of the groups [7, 8]. To study the effect of these parameters on community fairness, we propose a new extension of the stochastic block model [9] and use it to create various synthetic networks. We report findings from experiments conducted on these synthetic networks, aiming at evaluating the impact of size imbalance, homophily, and connectivity on balance and modularity fairness.

The remainder of this paper is structured as follows. In Chapter 2, we define the three types of community fairness, in Chapter 3 we describe the Louvain algorithm and our proposed modifications using the introduced fairness metrics, Fair Modularity and diversity modularity, in Chapter 4, we descrcibe the real networks and we introduce the synthetic models used for the experimental evaluation. In Chapter 5, the document presents the outcomes of our empirical assessment, showcasing the findings from community detection, the evaluation of fairness across various networks, and the analysis of correlations among different fairness metrics. Chapter 6 discusses the existing literature, while Chapter 7 concludes with our findings and outlines potential future research to further develop the concept introduced.

Chapter 2

Fairness Metrics

2.1 Community Detection

Community detection was vastly studied over the past years in computer science field. With the rapid popularity and development on the social networks, community detection procedure and classifying users is of considerable importance in network analysis. This evolution had a crucial impact in optimization of classical, state of the art methods but also the development of new techniques, using modern technologies.

Most classical methods [?] in community detection are based on the network topological structure and morphology. They are using statistical models in order to calculate and determine similarities between the components of the graph and try to cluster them in groups. Depending on the model used, there are different approaches to achieve the community partitioning. Traditional methods, such Kernighan-Lin algorithm and spectral bisection method are trying to find an optimal cut and divide the network in two pieces and belong to the Graph Partitioning family of algorithms. Other well known techniques use similarity or divisive metrics in order to merge or divide the nodes between them (Hierarchical Clustering and Girvan-Newman algorithm). But the needs of the rapid evolution in social networks created a new problem. These algorithms are very dependent on edge density and modern social networks with thousands and millions connections between its users, designed strange and confusing patterns that created the need for new techniques.

Deep learning techniques have been developed to bring the solution in the above problem. It offers a flexible solution for handling high-dimensional network data with improved accuracy compared to traditional methods like spectral clustering and statistical inference. The use of deep neural networks (DNNs), deep nonnegative matrix factorization and deep sparce filtering as deep learning models has significantly advanced the field of community detection. DNNs are further divided into convolutional networks, graph attention networks, generative adversarial networks and autoencoders. The latest developments in deep learning for community detection include the use of popular benchmark datasets, evaluation metrics and open-source implementation for experimentation settings. With its various practical applications in various domains, deep learning continues to be a fast-growing field in community detection.

In our work we use three community detection algorithms to evaluate our procedures.

2.2 Fairness in Community Detection

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes and $E \subseteq V \times V$ the set of edges. We assume that the nodes belong to one of two groups based on the value of one their attributes, namely the blue group, denoted as B, and the red group, denoted as R, where $B \cup R = V$ and $B \cap R = \emptyset$. Let us call *protected* group the red group. We will use ϕ to denote the ratio of the red nodes in the overall population, that is, $\phi = \frac{|R|}{|V|}$

Let $\mathcal{C} = \{C_1, C_2, \ldots C_k\}$ be the set of communities. For a community C_i, abusing slightly the notation, we use $B(C_i)$ and $R(C_i)$, for respectively the blue and red nodes that belong to C_i. We assess the fairness of each community from two distinct perspectives. Firstly, we analyze whether every group is adequately represented within each community. Secondly, we evaluate whether the members of each group are sufficiently well-connected within their respective communities.

2.3 Balance Fairness

Our first notion of fairness is based on balance. For a community $C_i \in \mathcal{C}$, its balance, $balance(C_i)$, is defined in [2]:

$$balance(C_i) = min\left(\frac{|R(C_i)|}{|B(C_i)|}, \frac{|B(C_i)|}{|R(C_i)|}\right). \tag{2.1}$$

Balance focus on the representation of each group in each community. A community with an equal number of red and blue nodes has balance 0.5 (perfectly balanced), while a monochromatic community has balance 1 (fully unbalanced). Balance encapsulates a specific notion of fairness.

We adopt a demographic parity approach [?] to balanced-based fairness. For a community C_i to be fair towards the protected group R, we ask that the percentage of red nodes in the community is at least equal to ϕ, i.e., the percentage of red nodes in the overall population.

Definition 2.1. For a community $C_i \in \mathcal{C}$, the balance fairness of C_i, $f_{balance}(C_i)$, is defined as:

$$f_{balance}(C_i) = \frac{|R(C_i)|}{|C_i|} - \phi. \tag{2.2}$$

Negative values of $f_{balance}(C_i)$ indicate unfairness towards the red group, that is, the fact that the red nodes are less well-represented in the community than the blue ones. Positive values indicate the opposite.

2.4 Fair Modularity

Our second notion of fairness evaluates for each community how well the members of each group are connected within the community emphasizing the importance of strong connections. Our definition is based on modularity [5]. Modularity measures the divergence between the number of intra-communities edges from the expected such number assuming random connections. Specifically, the modularity of community C_i, Q_{C_i}, is defined as:

$$Q_{C_i} = \frac{1}{2m} \sum_{u \in C_i} \sum_{v \in C_i} (A_{uv} - \frac{d_u \, d_v}{2m}) \tag{2.3}$$

where A is the adjacency matrix, m the number of edges in G and d_u, d_v the degree of node u, and v respectively.

Modularity provides a measure of how well all nodes in a community are connected with each other. We are interested in the connectivity of the red nodes in particular. To this end, for each red node u in C_i we take the difference between the actual number of its intra-community edges and the expected such number. We call this measure *red modularity* ($Q^R_{C_i}$):

$$Q^R_{C_i} = \frac{1}{2m} \sum_{u \in R(C_i)} \sum_{v \in C_i} (A_{uv} - \frac{d_u \, d_v}{2m}) \tag{2.4}$$

We define similarly the *blue modularity* ($Q^B_{C_i}$) as:

$$Q^B_{C_i} = \frac{1}{2m} \sum_{u \in B(C_i)} \sum_{v \in C_i} (A_{uv} - \frac{d_u \, d_v}{2m}) \tag{2.5}$$

An equivalent more efficient way to express modularity was derived in [?]:

$$Q_{C_i} = \frac{L_c}{m} - \left(\frac{d_{C_i}}{2m}\right)^2 \tag{2.6}$$

where L_{C_i} is the number of intra-community edges, and d_{C_i} the sum of the degrees of nodes in community C_i.

Following a similar approach, we derive the following more efficient formulas for the red and blue modularity:

$$Q^R_{C_i} = \frac{L^R_c}{m} - \frac{d_{C_i} \, d^R_{C_i}}{(2m)^2} \tag{2.7}$$

$$Q^B_{C_i} = \frac{L^B_c}{m} - \frac{d_{C_i} \, d^B_{C_i}}{(2m)^2} \tag{2.8}$$

where $L^R_{C_i}$ ($L^B_{C_i}$) is the number of intra-community edges with at least one red (blue) endpoint and $d^R_{C_i}$ ($d^B_{C_i}$) is the sum of the degrees of the red (blue) nodes in C_i.

We now define *modularity fairness* by comparing the red and the blue modularity.

Definition 2.2. For a community $C_i \in \mathcal{C}$, the modularity fairness of C_i, $f_{modularity}(C_i)$ is defined as:

$$f_{modularity}(C_i) = \frac{RQ_{C_i} - BQ_{C_i}}{|Q_{C_i}|}. \tag{2.9}$$

Negative values of $f_{modularity}(C_i)$ indicate unfairness towards the red group, that is, the fact that the red nodes are less connected within the community than the blue ones. Positive values indicate the opposite.

2.5 Diversity Fairness

Our third notion of fairness expands the modularity family definitions and evaluates how well are connected the members of each group in the whole network. Inspired by [6] they are addressing the filter bubble problem through the lens of algorithmic fairness, particularly focusing on dyadic-level fairness criteria. It introduces two criteria, namely subgroup dyadic-level protection and mixed dyadic-level protection, to assess fairness in link formation with respect to protected group memberships. The former ensures representativeness of protected subgroups in link creation, while the latter evaluates homogeneity in nodes participating in each link to prevent segregation and filter bubble effects.

Building upon this foundation, the paper incorporates the network modularity measure as a tool to assess mixed dyadic-level protection. Network modularity, initially developed for community detection, quantifies homophily in social networks by analyzing link density within communities. The modularity measure is adapted for fairness evaluation by considering protected attribute values, such as gender, in the calculation. A higher modularity value indicates a network biased towards intra-group links, signaling potential unfairness.

This Diversity Modularity metric is inspired by foundational work documented in [6] and is designed to quantify the degree of modularity within networks, taking into account the attributes among nodes. The metric is defined as follows:

$$Q_D = \frac{1}{2m} \sum_{i,j \in C} (A_{i,j} - \frac{d_i d_j}{2m}) \delta(c_i, c_j)(1 - \delta(P_i, P_j)) \tag{2.10}$$

Here, Q_D represents the diversity modularity, where $A_{i,j}$ denotes the adjacency matrix element indicating the presence (or absence) of an edge between nodes i and j. The term, d_i and d_j refer to the degrees of nodes i and j respectively, while m represents the total number of edges in the network. The $\delta(P_i, P_j)$ is the kronecher function for the attribute we study each time, so it is 1 if nodes i and j have the same value for the selected attribte and 0 otherwise.

Integrating the $(1 - \delta(P_i, P_j))$ we can rewrite 2.10 as:

$$Q_D = \frac{1}{2m} \sum_{j \in P(C_i)} \sum_{i \in C_i} (A_{i,j} - \frac{d_i d_j}{2m}) \tag{2.11}$$

The sets C_i and $P(C_i)$ denote the community to which node i belongs and the subset of nodes within C_i that possess an attribute value opposite to that of node i, respectively.

Definition 2.3. To facilitate a more efficient computation and analysis of the modularity metric, an equivalent formulation is proposed:

$$Q_D = \frac{L_{C_i}^P}{m} - \frac{d_{C_i} d_{C_i}^P}{(2m)^2} \tag{2.12}$$

In this expression, $L_{C_i}^P$ signifies the sum of the weights of edges connecting nodes of differing attributes within the same community C_i. Meanwhile, d_{C_i} and $d_{C_i}^P$ represent the cumulative degrees of nodes within C_i and the aggregate edge weights connecting nodes of opposite attributes within the community (inclusive of inter-community connections), respectively. This reformulation not only retains the original metric's intent but also enhances computational efficiency, enabling a more streamlined analysis of modularity in networks characterized by attribute polarization.

In alignment with this concept, our proposed fairness-aware framework extends this evaluation by introducing a metric that gauges the reduction in modularity measure. This metric serves as a means to appraise whether the link prediction results contribute to bias by favoring either inter-group or intra-group links. The framework aims to quantify and mitigate unfairness in the modified network resulting from link predictions, enhancing the understanding of the network's fairness in terms of mixed dyadic-level protection.

Chapter 3

Algorithms

3.1 Louvain Algorithm

The first algorithm we use to detect communities in our experiments is the Louvain algorithm as proposed by [10] and [11]. Louvain Community Detection algorithm is a simple method to detect a network's community structure. The algorithm is divided in 2 phases. On the first phase, it assigns every node in its own community, so the in the first partition there are as many communities as the total number of nodes in the network. Next, for every node i, they evaluate the gain of modularity by moving each node to the neighbouring communities and try to find the maximum positive modularity. If there is no positive modularity the node remains in its community. This process is repeated for all nodes until there is no other improvement.

Lemma 3.1. *The modularity gain obtained by moving a node i into a community C can be calculated by the following formula as introduced by [10]:*

$$\Delta Q = \left[\frac{\sum_{in} + 2\sum_{i \in C_i} d_{i,in}}{2m} - \left(\frac{\sum_{tot} + \sum_{i \in C_i} d_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{\sum_{i \in C_i} d_i}{2m}\right)^2\right] \tag{3.1}$$

In 3.1, ΔQ is the change in modularity, $\sum_{in}$ is the sum of the weights of the edges inside the community, $\sum_{i \in C_i} d_{i,in}$ is the sum of the weights of the edges from node i to nodes in the community, m is the sum of the weights of all the edges in the network, $\sum_{tot}$ is the sum of the weights of the edges incident to nodes in the community, $\sum_{i \in C_i} d_i$ is the sum of the weights of the edges incident to node i.

From 3.1 with simple Algebra (combining [10][11]) we can derive in the formula:

$$\Delta Q = \frac{\sum_{i \in C_i} d_{i,in}}{2m} - \frac{\sum_{tot} \cdot \sum_{i \in C_i} d_i}{2m^2} \tag{3.2}$$

where m is the size of the graph, $\sum_{i \in C_i} d_{i,in}$ is the sum of the weights of the links from i to nodes in C, d_i is the sum of the weights of the links incident to node i and $\sum_{tot}$ is the sum of the weights of the links incident to nodes in C. The weight of a link is considered as 1 unless specified otherwise.

The goal of the second phase is to build a new network, whose nodes are now the communities found in the first phase. For that cause, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Once this phase is complete, it is possible to reapply the first phase creating bigger communities with increased modularity.

Algorithm 3.1 Louvain algorithm

Input: Graph $G(V, E)$ where V is the set of vertices, E is the set of edges.
Output: List of N clusters detected.
repeat
  Assign every vertex $v \in V$ a unique community number, calculate the modularity Q for the initial partition.
  for each vertex $v \in V$ **do**
    Calculate the modularity gain ΔQ by removing v from its current community and placing it in the community of each neighbor.
    if $\Delta Q > 0$ **then**
      Move v to the community with the highest modularity gain.
    end if
  end for
  Aggregate every node that belongs to the same community and create a new ”meta-node” representing them. Now the new V is the set of those ”meta-nodes”.

  Recalculate the weight of the edges between these new ”meta-nodes”.
until there is no change in modularity
=0

3.2 Fair Modularity Louvain Modifictaiton

We propose a modification to Louvain Algorithm for community detection named Fair Modularity Louvain. Our modification optimizes the fair modularity metric introduced earlier in this work and its purpose is to build communities of equal red and blue connectivity thus, there is no group that is more connected than the other inside the community. For that purpose it is crucial to balance the modularity between the red connectivity and blue connectivity. The algorithm also works in 2 steps as the original procedure.

Before we describe the algorithm steps, we need to introduce a new metric named Fair modularity gain ΔQ_F. That value is adjacent to the modularity gain ΔQ 3.1 from Louvain and is responsible for calculating the increase of fair modularity when a node moves from a community to another.

Lemma 3.2. *The Fair Modularity gain for red nodes perspective is defined as:*

$$\Delta Q_F^R = \left[\frac{\sum_{in}^R + 2\sum_{i\in C} d_{i,in}^R}{2m} - \left(\frac{\sum_{tot}^R + \sum_{i\in C_i} d_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}^R}{2m} - \left(\frac{\sum_{tot}^R}{2m} \right)^2 - \left(\frac{\sum_{i\in C_i} d_i}{2m} \right)^2 \right] \tag{3.3}$$

$$\Delta Q_F^R = \frac{\sum_{i\in C} d_{i,in}^R}{2m} - \frac{\sum_{tot}^R \cdot \sum_{i\in C_i} d_i}{2m^2} \tag{3.4}$$

$$\Delta Q_F^B = \frac{\sum_{i\in C} d_{i,in}^B}{2m} - \frac{\sum_{tot}^B \cdot \sum_{i\in C_i} d_i}{2m^2} \tag{3.5}$$

$$\Delta Q_F = |\Delta Q_F^R - \Delta Q_F^B| \tag{3.6}$$

In 3.3 ΔQ_F^R is the change in fair modularity for red nodes, $\sum_{in}^R$ is the sum of the weights of the edges inside the community with at least one red node, $\sum_{i\in C} d_{i,in}^R$ is the sum of the weights of the edges from node i with at least one red node, m is the sum of the weights of all the edges in the network, $\sum_{tot}^R$ is the sum of the weights of the edges incident to nodes inside the community with at least one red node, $\sum_{i\in C_i} d_i$ is the sum of the weights of the edges incident to node i.

Again with simple calculations from 3.3 we can derive to 3.4. Adjacently for the blue node perspective, replacing the adjacent variables from the 3.3 we can conclude to the 3.5. The algorithm optimizes the 3.6.

Again the goal of the second phase is to build a new network, whose nodes are now the communities found in the first phase. For that cause, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Once this phase is complete, it is possible to reapply the first phase creating bigger communities with increased fair modularity.

Algorithm 3.2 Fair Modularity Louvain algorithm

Input: Graph $G(V, E, A, \lambda)$ where V is the set of vertices, E is the set of edges and A is the attributes of each node in the graph and $\lambda \in [0, 1]$ controlling the participation rate of each modularity value.

Output: List of N clusters detected.

repeat

Assign every vertex $v \in V$ a unique community number, calculate the modularity Q and the fair modularity Q_F for the initial partition and calculate the $Q_{diff} = \lambda Q - (1 - \lambda)Q_F$

for each vertex $v \in V$ **do**

Calculate the modularity gain ΔQ and the fair modularity gain ΔQ_F by removing v from its current community and placing it in the community of each neighbor.

if $\lambda \Delta Q - (1 - \lambda)\Delta Q_F > 0$ **then**

Move v to the community with the highest gain.

end if

end for

Aggregate every node that belongs to the same community and create a new ”meta-node” representing them. Now the new V set is those ”meta-nodes”.

Recalculate the weight of the edges between these new ”meta-nodes”.

until there is no change in Q_{diff} =0

3.3 Diversity Louvain Modification

The Diversity Louvain Algorithm represents an innovative modification of the traditional Louvain Algorithm, specifically tailored for enhanced community detection within networks. This advanced algorithm endeavors to optimize the segregation of communities by meticulously adjusting the distribution of Red-Blue edges within these groups without losing the community structure that the original Louvain algorithm offers. The algorithm works in 2 steps as the original procedure.

Before we describe the algorithm steps, we need to introduce a new metric named Diversity Modularity Gain ΔQ_D. This value is adjacent to the modularity gain ΔQ from Louvain and is responsible for calculating the increase in Diversity modularity when a node moves from a community to another.

Lemma 3.3. *The Diversity Modularity gain is defined as:*

$$\Delta Q_D = \left[\frac{\sum_{in}^{P} + 2\sum_{i\in C} d_{i,in}^{P}}{2m} - \left(\frac{\sum_{tot}^{P} + \sum_{i\in C_i} d_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}^{P}}{2m} - \left(\frac{\sum_{tot}^{P}}{2m}\right)^2 - \left(\frac{\sum_{i\in C_i} d_i}{2m}\right)^2\right] \tag{3.7}$$

$$\Delta Q_D = \frac{\sum_{i\in C} d_{i,in}^{P}}{2m} - \frac{\sum_{tot}^{P} \cdot \sum_{i\in C_i} d_i}{2m^2} \tag{3.8}$$

In 3.7 ΔQ_D is the change in diversity modularity, $\sum_{in}^{P}$ is the sum of the weights of the edges inside the community between nodes with different attribute, $\sum_{i\in C} d_{i,in}^{P}$ is the sum of the weights of the edges from node i to nodes with different attribute in the community, m is the sum of the weights of all the edges in the network, $\sum_{tot}^{P}$ is the sum of the weights of the edges incident to nodes with different attribute in the community, $\sum_{i\in C_i} d_i$ is the sum of the weights of the edges incident to node i. Again with simple calculations from 3.7 we can derive to 3.8.

Initially, the algorithm designates each node to a single community, establishing an initial partition. Subsequent to this assignment, the algorithm calculates the modularity Q, the diversity modularity Q_D and their difference $Q_{diff} = \lambda Q - (1-\lambda)Q_D$. This λ controls participation rate of each modularity value ensuring that the final communities will not lose much of the structure.

Next for every node i, they evaluate the gain of modularity ΔQ and the diversity modularity gain ΔQ_D by moving each node to the neighbouring communities and try to find the maximum positive gain. We assume the final gain as $final\_gain = \lambda \Delta Q - (1-\lambda)\Delta Q_D$. If there is no positive gain the node remains in its community. This process is repeated for all nodes until there is no other improvement of Q_{diff}.

The subsequent phase is characterized by the construction of a novel network framework, wherein the nodes are constituted by the communities identified during the initial phase. In this restructured network, the linkage weight between newly formed nodes is determined by the aggregate weight of connections spanning nodes across the respective communities. This model presupposes a unitary weight assignment $weight = 1$ for edges interlinking nodes of divergent colors. Upon the completion of this phase, the algorithm permits a reiteration of the initial phase, facilitating the amalgamation of communities into larger entities with augmented diversity modularity.

This iterative process—comprising the initial assignment and optimization phase, followed by the network reconstruction and community integration phase—persists until no further gains in Diversity Modularity are realized.

Algorithm 3.3 Diversity Louvain algorithm

Input: Graph $G(V, E, A, \lambda)$ where V is the set of vertices, E is the set of edges and A is the attributes of each node in the graph and $\lambda \in [0, 1]$ controlling the participation rate of each modularity value.

Output: List of N clusters detected.

repeat

  Assign every vertex $v \in V$ a unique community number, calculate the modularity Q and the Diversity modularity Q_D for the initial partition and calculate the $Q_{diff} = \lambda Q - (1 - \lambda) Q_D$

  for each vertex $v \in V$ **do**

    Calculate the modularity gain ΔQ and the diversity modularity gain ΔQ_D by removing v from its current community and placing it in the community of each neighbor.

    if $\lambda \Delta Q - (1 - \lambda) \Delta Q_D > 0$ **then**

      Move v to the community with the highest gain.

    end if

  end for

  Aggregate every node that belongs to the same community and create a new ”meta-node” representing them. Now the new V set is those ”meta-nodes”

  Recalculate the weight of the edges between these new ”meta-nodes”.

until there is no change in Q_{diff} =0

Chapter 4

Dataset Description

In this section, we study the balance and modularity fairness of several real and synthetic networks. To detect communities, we use the Louvain algorithm, a greedy algorithm that optimizes modularity [10].

4.1 Real World Data

We study the following real datasets:

- **Pokec**[1] Nodes are the users of the Pokec social network and edges are friendship relationships between them. We study both the gender attribute (**Pokec-g**) and the age attribute (**Pokec-a**). For the age attribute, we remove nodes that have no value for this attribute, or the value was not a possible value for age. We use the median value of the remaining nodes for splitting the nodes into two (almost) equal-sized groups.
- **Deezer**[2] Nodes are Deezer users from European countries and edges are mutual follower relationships between them.

[1]https://snap.stanford.edu/data/soc-Pokec.html
[2]https://snap.stanford.edu/data/feather-deezer-social.html

- **Facebook**[3] The dataset consists of friends list from Facebook.
- **Twitch Gamers**[4] Nodes are twitch users and edges are mutual follower relationship between them.
- **NBA**[5] Nodes are NBA players and data collected from Twitter.

Note that for Deezer and Facebook the actual values of the sensitive attribute are hidden in the datasets. We also report homophily values that indicate the tendency of nodes to connect with nodes with similar attribute values, in our case, with nodes of the same color. We report separately the homophily of the red (Rh) and the homophily of the blue nodes (Bh). Rh is computed as the ratio of the number of the actual edges connecting two red nodes and the expected number of such edges (estimated as ϕ^2). $Rh > 1$ indicates homophily, while $Rh < 1$ heterophily (tendency to connect with nodes of the opposite color). Similarly, we compute Bh as the ratio of the number of the actual edges between two blue nodes and the expected such number (estimated as $(1 - \phi)^2$).

In Table 4.2(left), we report the results of the Louvain algorithm: the number of communities detected, their average size, and the average modularity of all communities.

Table 4.1: Network characteristics, AvRd (AvBd): average degree of the red (blue) nodes, Rh (Bh): red (blue) homompily.

Network	# Nodes	# Edges	Attribute	# Blue nodes	# Red nodes	AvRd	AvBd	ϕ
Pokec-g	1,632,803	22,301,964	Gender	804,474	828,289	28.28	26.32	0.51
Pokec-a	1,095,590	10,779,932	Age	546,212	549,381	14.20	25.20	0.50
Pokec-a asym	1,095,590	10,779,932	Age ≤ 30	250,863	844,705	35.5	15.08	0.77
Deezer	28,281	92,752	Gender	12,535	15,738	6.73	6.34	0.57
Facebook	4,039	88,234	Gender	1,532	2,507	42.09	46.30	0.62
Twitch Gamers	168,114	6,797,557	Maturity	79,033	89,081	74.31	88.26	0.53
NBA	400	10,621	Age ≤ 26	204	196	43.54	61.9	0.49

[3]http://snap.stanford.edu/data/ego-Facebook.html
[4]https://snap.stanford.edu/data/twitch_gamers.html
[5]https://github.com/EnyanDai/FairGNN/tree/main/dataset/NBA

Table 4.2: Communities detected and their aggregated balance and modularity fairness.

Network	# Comms	Size	Mod	Balance				Modularity			
				Min	Avg	Per. fair R	Per. fair B	Min	Avg	Per. fair R	Per. fair B
Pokec-g	38	42,967.5	0.71	0.19	0.49	0.57	0.58	-1	0.1	0.85	0.15
Pokec-a	47	23,309.9	0.72	0	0.36	0.65	0.50	-1	-0.46	0.02	0.99
Pokec-a asym	47	23,309	0.72	-0.5	0.13	0.78	0.45	-1	0.44	0.02	0.99
Deezer	88	348.86	0.69	0.11	0.56	0.58	0.50	-1	0.18	0.92	0.10
Facebook	16	252.19	0.83	0.43	0.62	0.58	0.58	-0.05	0.23	0.95	0.09
Twitch Gamers	21	8,005.43	0.42	0.37	0.63	0.52	0.65	-0.40	0.22	0.52	0.65
NBA	6	23.33	0.18	-0.4	-0.03	0.54	0.55	-0.86	-0.1	0.67	0.66

4.2 Synthetic Data

4.2.1 Symmetric Synthetic Model

To study the factors that may lead to unfairness, we introduce a new model based on the stochastic block model [4] to create networks with nodes of different colors and connectivity behavior. The model has three important parameters:

- p_R: the probability that a node belongs to the red group. This parameter controls the relative size of the two groups. By setting $p_R = 0.5$, we get groups of equal size, while when $p_R < 0.5$, the red group is the minority group.
- p_h: the probability that a node connects with a a node that has the same color with it. This parameter controls homophily. With $p_h = 1$, we have perfect homophily, with $p_h = 0$, nodes connect only with nodes of the opposite color and we get heterophily, while $p_h = 0.5$ results in neutral behavior.
- p_c: the probability that a node connects with a a node that belongs to the same community with it. This parameter controls modularity. Large values of p_c lead to well-connected communities, while when $p_c \leq 0.5$, there is no community structure.

We start by an initial assignment of nodes in h communities and then generate edges between them. Note that the actual number of communities created differs from h, depending on the values of the parameters. First, we assign an equal number of nodes to each community. Then, for each of the nodes, we use p_R to determine its color. Finally, we generate edges as follows. For each node u, we create h edges on average. To create edge $e = (u, v)$, we first select the community that v will belong to and then its color. Specifically, we use p_c to determine whether e will be an inter-community or an intra-community edge. Then, we use p_h to determine the color of node v. Let $X \in \{R, B\}$ be the selected color. If e is an intra-community edge, we choose as v uniformly at random a node of color X in the community of u. In case of an inter-community edge, we first choose uniformly at random one of the $h-1$ communities other than the community of u. Let C be this community. We choose as v uniformly at random a node of color X in C.

Table 4.3 summarizes the parameters. We study the influence of size imbalance (p_R), homophily (p_h) and intra-cluster connectivity (p_c) in fairness. In each case, we vary one of the three parameters and use the default values for the other. We run each experiment 5 times.

Table 4.3: Synthetic dataset characteristics.

Parameter	Meaning	Default
N	number of nodes	4000
p_R	ratio of red nodes	0.5
l	edges per new node	4
h	initial number of communities	5
p_h	homophily	0.5
p_c	prob. of intra-cluster edge	0.7

4.2.2 Assymetric Synthetic Model

In this subsection, we present an extension of our previously introduced model, aimed at further advancing the evaluation framework for network fairness. Building upon the foundation laid out in our earlier work, which introduced a novel synthetic network generation model to assess fairness based on various criteria, we now introduce an enhanced and more intricate model. This new iteration incorporates a heightened complexity, featuring an expanded set of parameters designed to capture and evaluate a broader spectrum of behaviors within the synthesized networks. Through this augmentation, we aim to deepen our understanding of fairness considerations in network structures and provide a more comprehensive tool for researchers and practitioners to assess and address nuanced aspects of fairness in network models. The model has these important parameters:

- p_R: the probability that a node belongs to the red group. This parameter controls the relative size of the two groups. By setting $p_R = 0.5$, we get groups of equal size, while when $p_R < 0.5$, the red group is the minority group.

- p_{hR}: the probability of a red node connects with a a node that has the same color with it. This parameter controls homophily of the red group. With $p_{hR} = 1$, we have perfect homophily, with $p_{hR} = 0$, nodes connect only with nodes of the opposite color and we get heterophily, while $p_{hR} = 0.5$ results in neutral behavior.

- p_{hB}: the probability of blue node connects with a node that has the same color with it. This parameter controls homophily of the blue group. With $p_{hB} = 1$, we have perfect homophily, with $p_{hB} = 0$, nodes connect only with nodes of the opposite color and we get heterophily, while $p_{hB} = 0.5$ results in neutral behavior.

- p_{cR}: the probability that a red node connects with a node that belongs to the same community with it. This parameter controls red group modularity. Large values of p_{cR} lead to well-connected communities, while when $p_{cR} \leq 0.5$, there is no community structure among red nodes.

- p_{cB}: the probability that a blue node connects with a node that belongs to the same community with it. This parameter controls blue group modularity. Large values of p_{cB} lead to well-connected communities, while when $p_{cB} \leq 0.5$, there is no community structure among the blue nodes.

We start by an initial assignment of nodes in k communities and then generate edges between the nodes. Note that the actual number of communities created differs from k, depending on the values of the other parameters. First, we assign an equal number of nodes to each community. Then, for each of the nodes, we use p_R to determine the color of the node. Finally, we generate edges as follows. For each node u, we create l edges on average. To create an edge $e = (u, v)$ for u, we first select the community that target node v will belong to and then the color of v. Specifically, we use p_{cR} and p_{cB} values if the initial node u is red or blue adjacently, to determine whether e will be an inter-community or an intra-community edge. Then, we use p_{hR} and p_{hB} values if the initial node u is red or blue adjacently, to determine the color of target node v. Let $X \in \{R, B\}$ be the selected color. If e is an intra-community edge, we choose as v uniformly at random a node of color X in the same community as u. If e is an inter-community edge, we first choose uniformly at random one of the $k-1$ communities other than the community of u. Let C be this community. We choose as v uniformly at random a node of color X in community C.

Chapter 5

Experiments

5.1 Evaluation of Fairness

We initiated the analysis by performing community detection on various social networks, encompassing both real-world and synthetic instances. Two widely employed algorithms, Louvain and Spectral Clustering, were employed for this task. The primary objective of community detection was to identify cohesive groups within the network.

Subsequently, we evaluated the identified community structures using the modularity value. Modularity serves as a metric to assess the quality of community structures, measuring the extent to which the network is partitioned into distinct, internally connected groups.

In Table 5.1, we report results about the communities detected in the real networks using the Louvain algorithm, specifically, we report the number of communities detected, their average size, and their average modularity.

Having established the community structures, our focus shifted to the examination of fairness within these communities. We employed two distinct fairness definitions

Table 5.1: Detected communities.

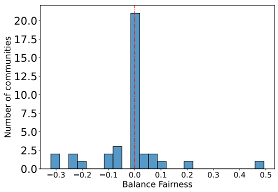
Network	Number	Avg. Size	Avg. Mod
Pokec-g	38	42,967	0.71
Pokec-a	47	23,309	0.72
Pokec-a asym	47	23,309	0.72
Deezer	88	348	0.69
Facebook	16	252	0.83
Twitch Gamers	21	8,005	0.42
NBA	6	66,7	0.18

to comprehensively assess the distribution of resources and opportunities among community members.

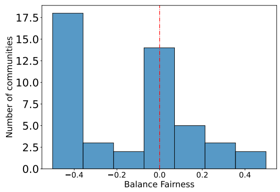
5.1.1 Real World Networks

In Figures 5.1 and 5.2, we plot respectively the distribution of $f_{balance}$ and $f_{modularity}$ of the communities found in the real networks. Negative values correspond to communities unfair towards the red group, while positive values to communities unfair towards the blue group. Unfair communities exist in all networks both in the case of $f_{balance}$ and $f_{modularity}$, as indicated by the large number of communities having non-zero $f_{balance}$ and $f_{modularity}$ values. In terms of $f_{balance}$, Pokec is almost gender balanced (Pokec-g), while it is unfair towards the younger individuals (Pokec-a). Pokec is also modularity fair for gender (Pokec-g) and modularity unfair towards the younger individuals (Pokec-a). Deeezer is almost gender balanced, but the members of the red group are more connected in their communities as indicated by the majority of positive $f_{modularity}$ values. Facebook is slightly balanced unfair towards the red group, which is also the largest group, but heavily modularity unfair towards the blue group. Finally, Twitch is both balanced and modularity unfair towards the more mature users.

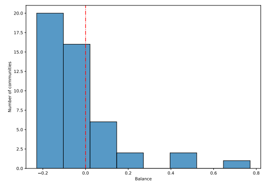
When examining diversity and fairness through Figure 5.3, we define a community as fair if the actual count of edges connecting Red and Blue nodes matches the expected count of such connections, with a baseline value of 0.5. Communities scoring above this threshold are deemed fair, whereas those below are considered less fair. The analysis reveals that the Facebook network exhibits complete fairness, with every community aligning with diversity expectations. Similarly, the Deezer network predominantly hosts communities that are considered fair. In contrast, the Pokec network, across two selected attributes, displays a mix of fair and less fair communities, albeit with a majority leaning towards fairness. Conversely, communities within the NBA and Twitch networks show a notable lack of diversity, particularly within the NBA network, where no community meets the diversity criterion.



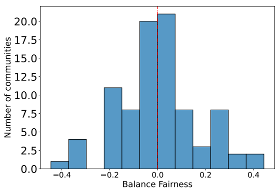
(a) Pokec-g



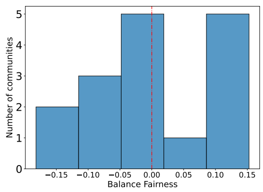
(b) Pokec-a



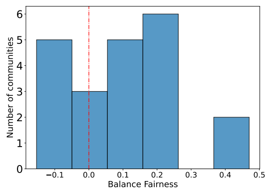
(c) Pokec-a asym



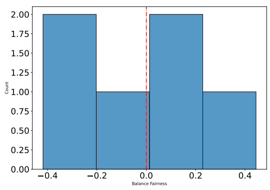
(d) Deezer



(e) Facebook

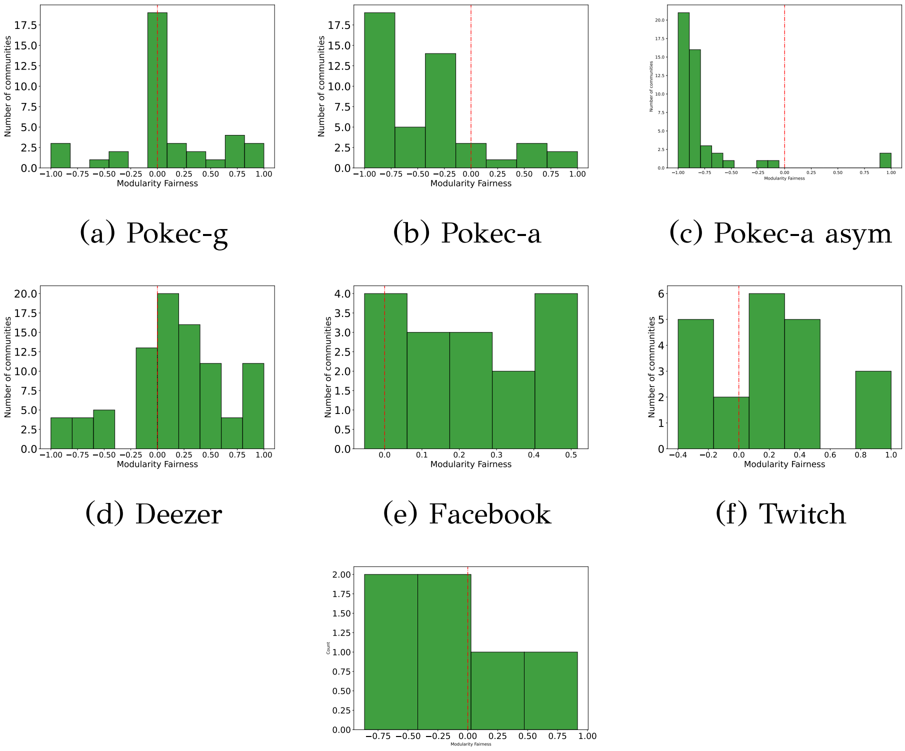


(f) Twitch



(g) NBA

Figure 5.1: Distribution of $f_{balance}$ in the real datasets. The red line corresponds to 0.

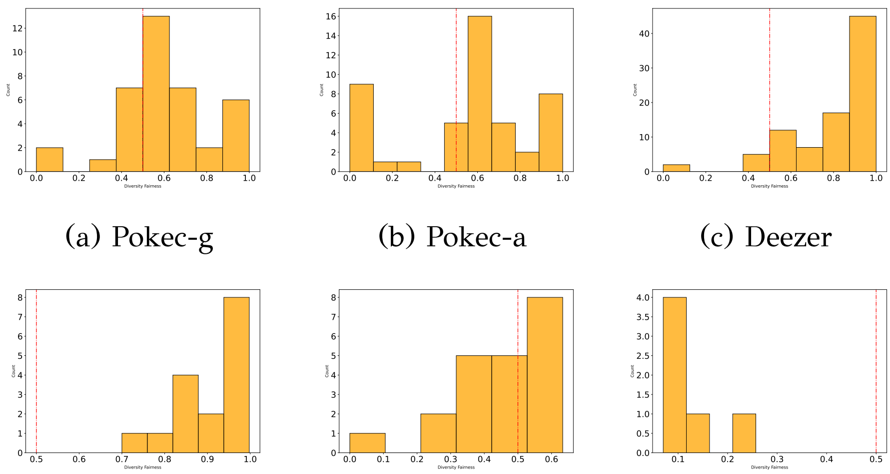


(a) Pokec-g (b) Pokec-a (c) Pokec-a asym

(d) Deezer (e) Facebook (f) Twitch

(g) NBA

Figure 5.2: Distribution of $f_{modularity}$ in the real datasets. The red line corresponds to 0.



(a) Pokec-g (b) Pokec-a (c) Deezer

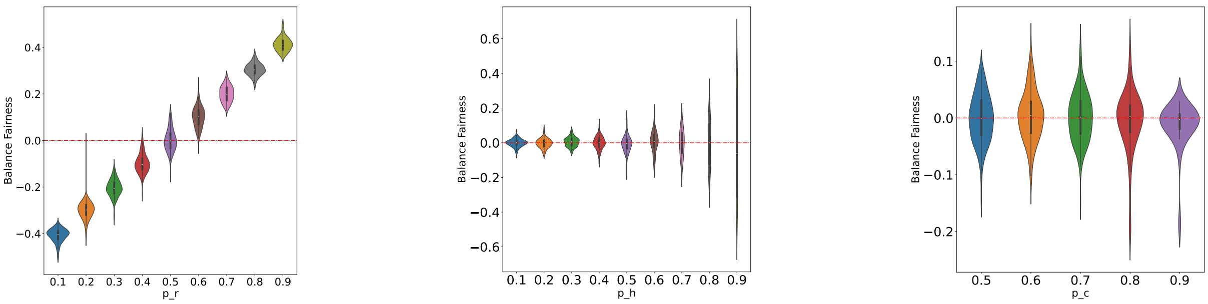
(d) Facebook (e) Twitch (f) NBA

Figure 5.3: Distribution of diversity fairness in the real datasets. The red line corresponds to 0.5

5.1.2 Synthetic Networks

In Figure 5.4, we report results regarding $f_{balance}$. We use violin plots to depict the distribution of $f_{balance}$ in the communities for different values of our parameters. Again negative values indicate unfairness towards the red group, while positive values unfairness towards the blue group. Size imbalance (p_B) is directly reflected in $f_{balance}$, since there is balance unfairness towards the smaller group. In terms of homophily (p_h), when the networks have low homophily, i.e., $p_h < 0.5$, all communities are almost balanced, i.e., their $f_{balance}$ is very close to 0. As networks become homophilic (p_h increases), the $f_{balance}$ of many communities deviates from 0 towards values corresponding to monochromatic communities (recall that $\phi = p_R = 0.5$). Finally, intra-cluster connectivity (p_c) has a very small effect on $f_{balance}$. While p_c affects the quality of the communities, it does not affect fairness, indicating that quality may not have a direct influence on fairness.

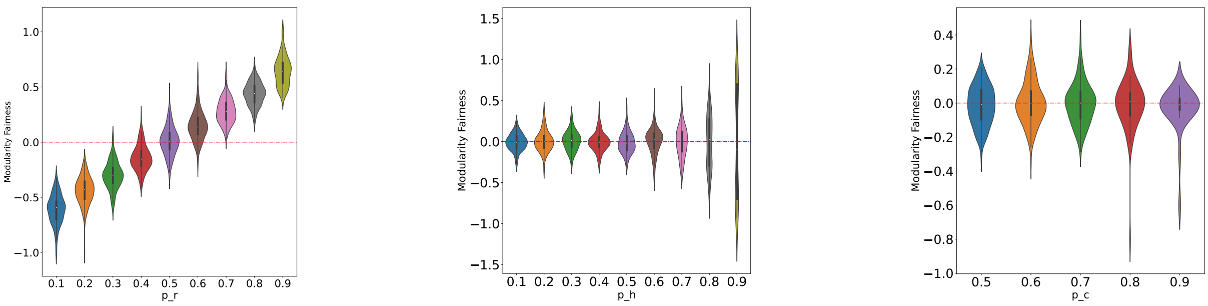
In Figure 5.5, we report results regarding $f_{modularity}$. In terms of p_R, we get the best fairness, when the two groups have equal sizes ($p_R = 0.5$). In all other cases, the algorithm favors the larger group, creating better connected communities (positive values) for this group. Homophily (p_h) seems to also affect $f_{modularity}$, since nodes end up in communities in which they are less connected. Again p_c has a smaller effect on fairness.



(a) p_R (b) p_h (c) p_c

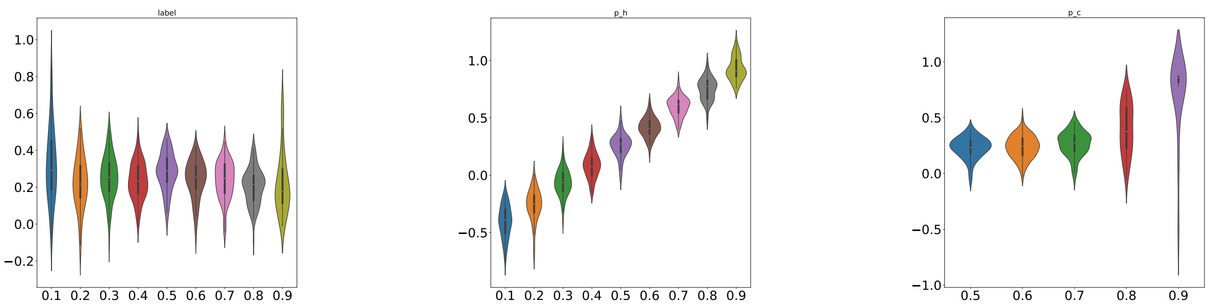
Figure 5.4: Distribution of $f_{balance}$ in synthetic datasets. The red line corresponds to 0.

In Figure 5.7, we report results regarding $f_{balance}$ compared with p_h changes. We use violin plots to depict the distribution of $f_{balance}$ in the communities for different values of our parameters. Again negative values indicate unfairness towards the red group, while positive values unfairness towards the blue group. In terms of ho-



(a) p_R (b) p_h (c) p_c

Figure 5.5: Distribution of $f_{modularity}$ in synthetic datasets. The red line corresponds to 0.



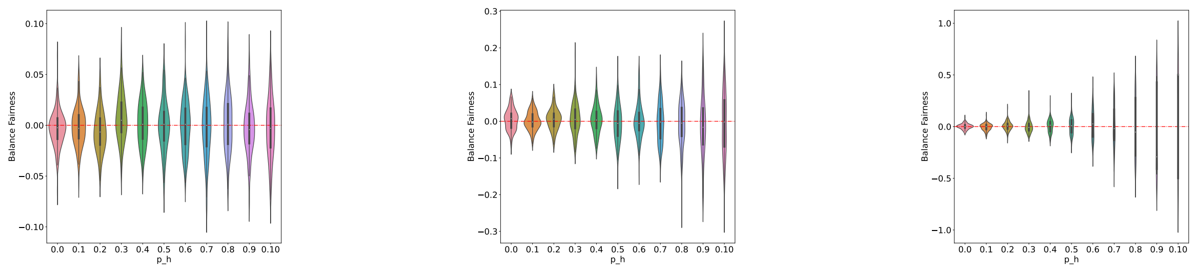
(a) p_R (b) p_h (c) p_c

Figure 5.6: Distribution of Filter Bubble fairness in synthetic datasets. The red line corresponds to 0.

mophily it seems when the nodes of one group tend to connect only with the opposite group ($p_{hB} = 0$) meaning no signs of homophily all communities are almost close to 0 thus considered balanced. When both groups are homophilic and connect only with nodes of their group, as it is obvious the networks show great variance on their fairness, making them unbalanced. We can conclude that heterophily leads to balanced networks.

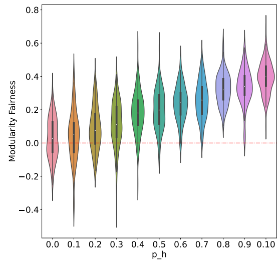
In Figure 5.8, we report results regarding $f_{modularity}$ compared with p_h changes. Homophily, seems to also affect $f_{modularity}$, since nodes end up in communities in which they are less connected. The group that is most connected with itself seems to be favored in the network. In the case of one group being fully homophilic, we see that as the other group becomes more homophilic there is a significant variance in the modularity value of the communities.

In Figure 5.10 and 5.11, we report results of $f_{balance}$ and $f_{modularity}$ compared with p_c changes respectively. For both fairness definitions it seems that there is low contribution of the connectivity in the changes of fairness.

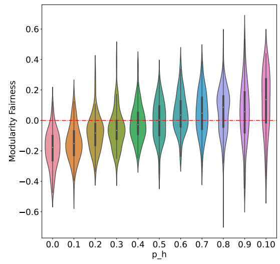


(a) $p_{hB} = 0$ (b) $p_{hB} = 0.5$ (c) $p_{hB} = 1.0$

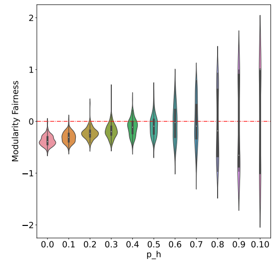
Figure 5.7: Distribution of $f_{balance}$ in asymetric synthetic datasets by changing p_{hR}. The red line corresponds to 0.



(a) $p_{hB} = 0$

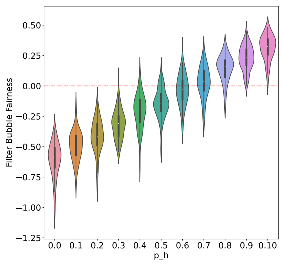


(b) $p_{hB} = 0.5$

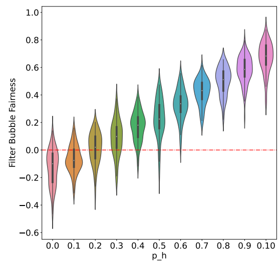


(c) $p_{hB} = 1.0$

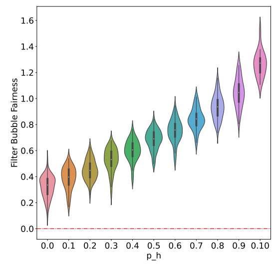
Figure 5.8: Distribution of $f_{modularity}$ in asymetric synthetic datasets by changing p_{hR}. The red line corresponds to 0.



(a) $p_{hB} = 0$

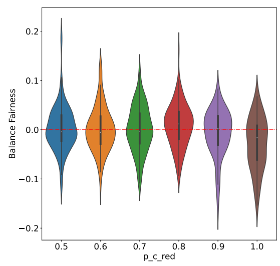


(b) $p_{hB} = 0.5$

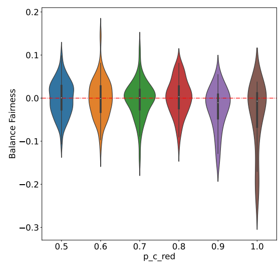


(c) $p_{hB} = 1.0$

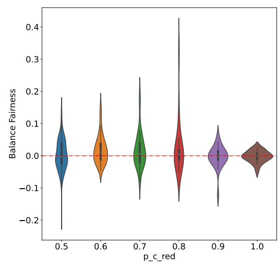
Figure 5.9: Distribution of Filter Bubble fairness in asymetric synthetic datasets by changing p_{hR}. The red line corresponds to 0.



(a) $p_{cB} = 0.5$

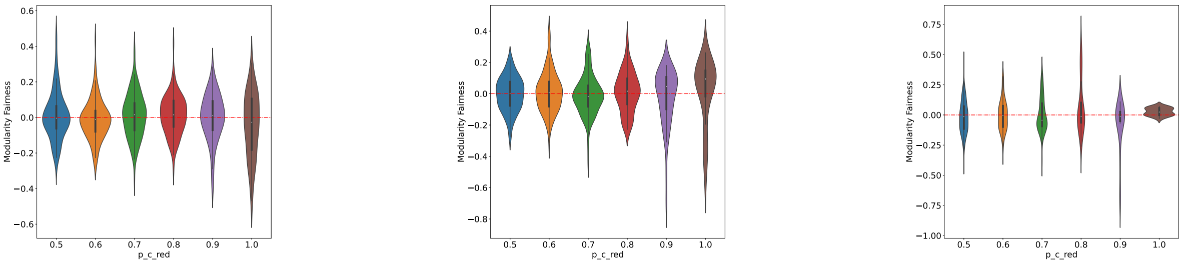


(b) $p_{cB} = 0.7$



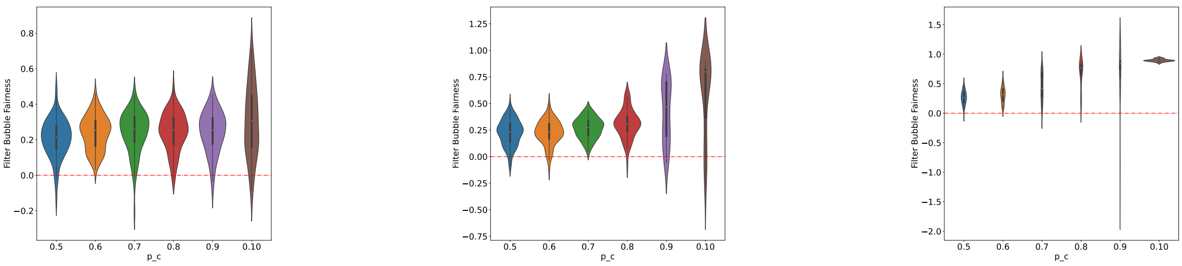
(c) $p_{cB} = 0.9$

Figure 5.10: Distribution of $f_{balance}$ in asymetric synthetic datasets by changing p_{cR}. The red line corresponds to 0.



(a) $p_{cB} = 0.5$ (b) $p_{cB} = 0.7$ (c) $p_{cB} = 0.9$

Figure 5.11: Distribution of $f_{modularity}$ in asymetric synthetic datasets by changing p_{cR}. The red line corresponds to 0.



(a) $p_{cB} = 0.5$ (b) $p_{cB} = 0.7$ (c) $p_{cB} = 0.9$

Figure 5.12: Distribution of Filter Bubble fairness in asymetric synthetic datasets by changing p_{cR}. The red line corresponds to 0.

5.2 Fairness Metrics Correlation

Across all networks analyzed, there is a notable relationship between Balance and Modularity fairness, indicating a strong correlation. However, for other metrics, there is no significant correlation observed. In figures 5.13, 5.14, 5.15, 5.16, 5.17 we can see the results for the real networks.

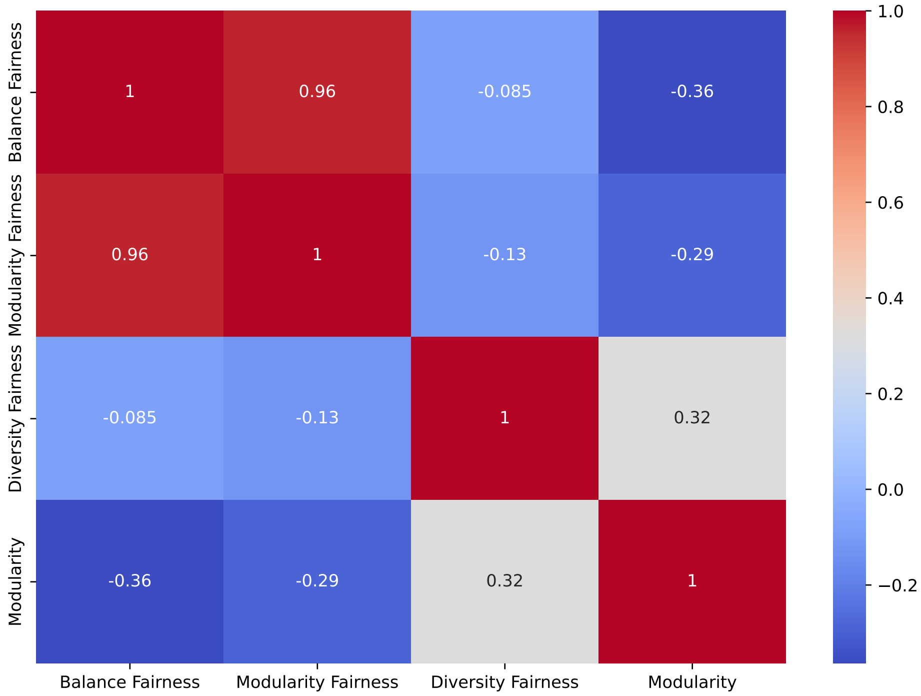


Figure 5.13: Facebook-Fairness Metrics Correlation

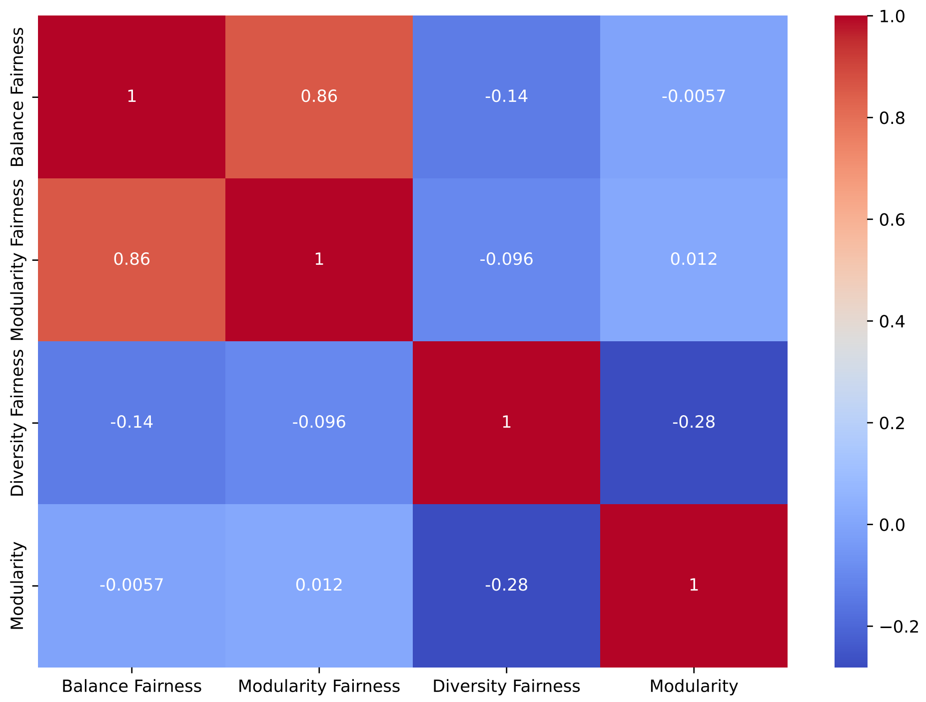


Figure 5.14: Deezer-Fairness Metrics Correlation

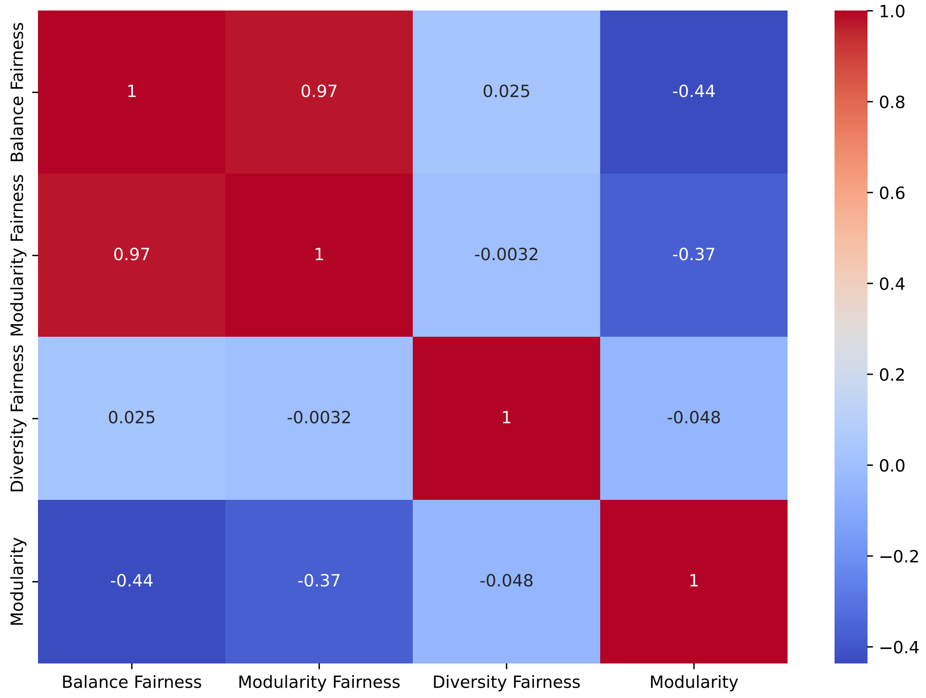


Figure 5.15: Twitch-Fairness Metrics Correlation

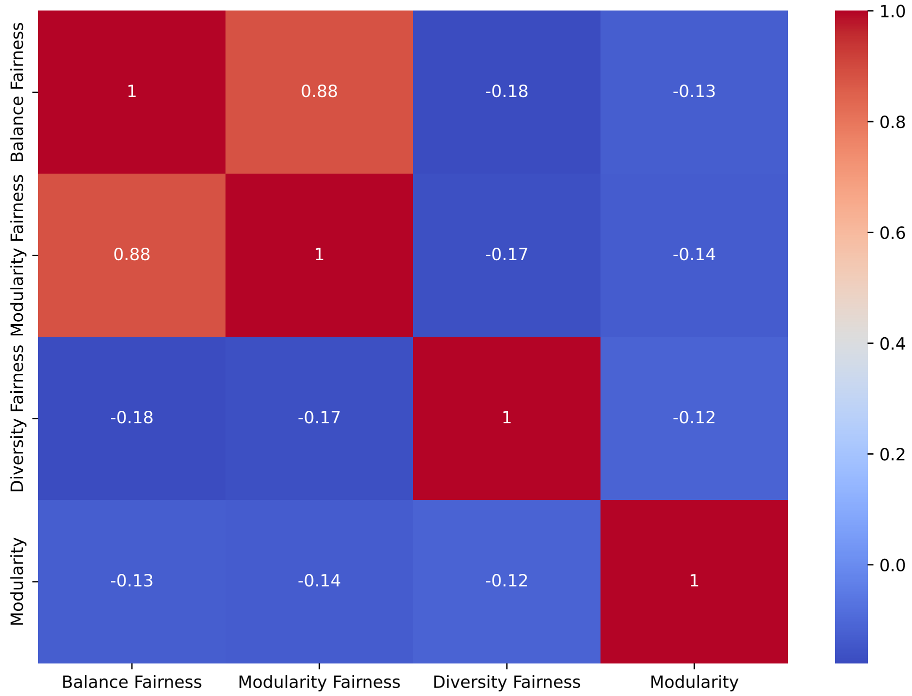


Figure 5.16: Pokec-Fairness Metrics Correlation

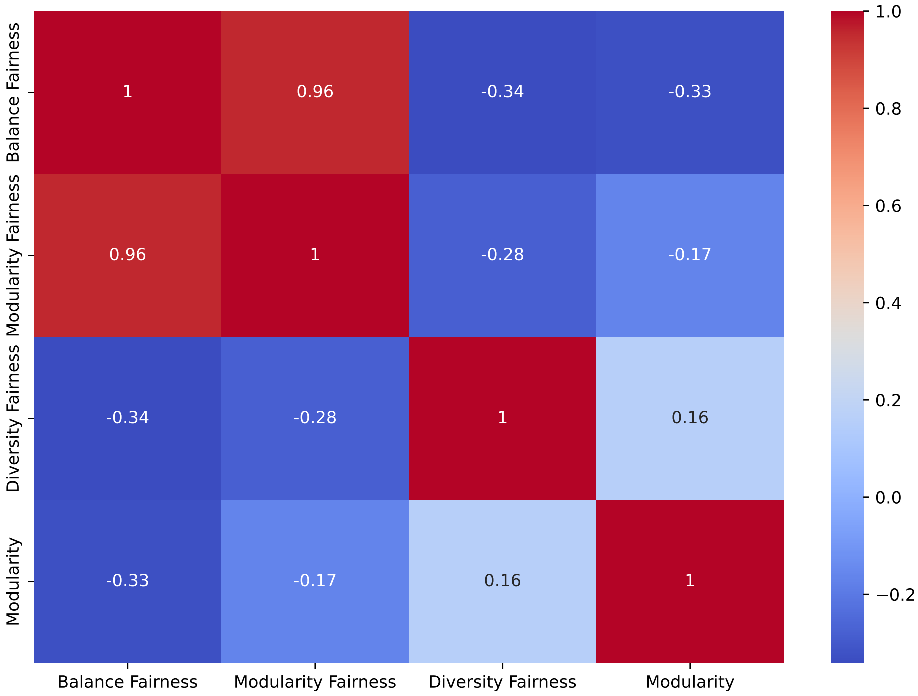


Figure 5.17: Pokec(age)-Fairness Metrics Correlation

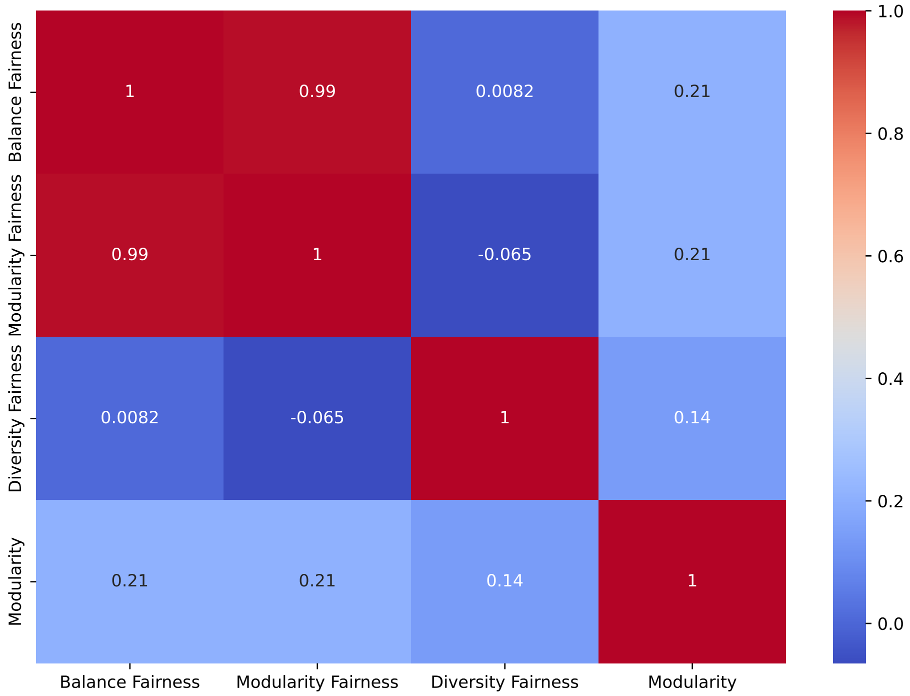


Figure 5.18: NBA - Fairness Metrics Correlation

5.3 Comparative Analysis of Modified Louvain Algorithms

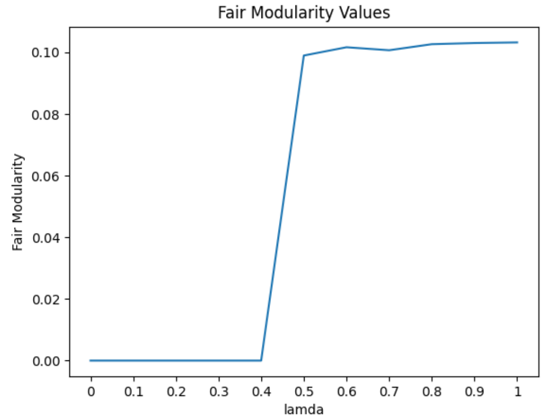
5.3.1 Fair Modularity Analysis

In this section, we delve into the comparative analysis of the modified Louvain algorithm, which is tailored to foster equal connectivity within communities, especially between nodes identified as red and blue. As outlined in Section 3.2, a community achieves fair connectivity when the connectivity density between red and blue group members is equivalent. This adaptation is grounded in the concept of Fair Modularity, referenced in 2.4 and 3.2, which acts as the guiding objective function for the algorithm's optimization process.

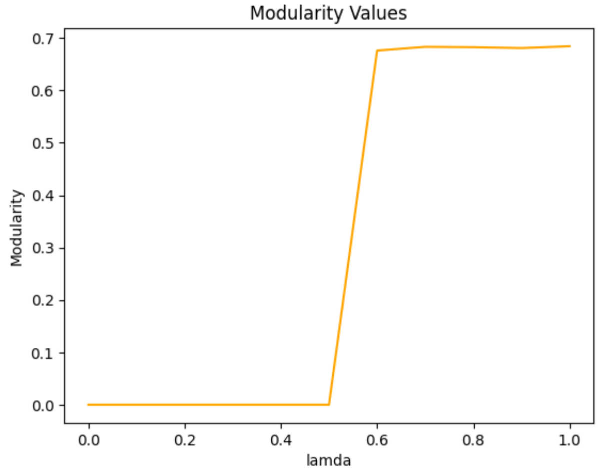
Throughout our experiments, we concurrently evaluate both modularity and fair modularity metrics, adjusting their influence in the optimization process according to the value of $\lambda \in [0, 1]$. This adjustment is governed by the equation:

$$Q_{diff} = 1 \times Q + (1 - \lambda) \times Q_F \tag{5.1}$$

where Q_{diff} represents the combined metric used for optimization, blending the traditional modularity Q and fair modularity Q_F based on the λ. A λ value of 0 focuses the algorithm purely on optimizing fair modularity Q_F, whereas a λ of 1 shifts the focus entirely to traditional modularity Q, mirroring the original Louvain algorithm. This framework allows us to examine the impact of varying degrees of fair metric optimization on community structure, highlighting the balance between different optimization metrics. The outcomes of these experiments and the effects of this metric balance are illustrated in Figures 5.19, 5.20, 5.21, and 5.22, showcasing the results of this trade-off.

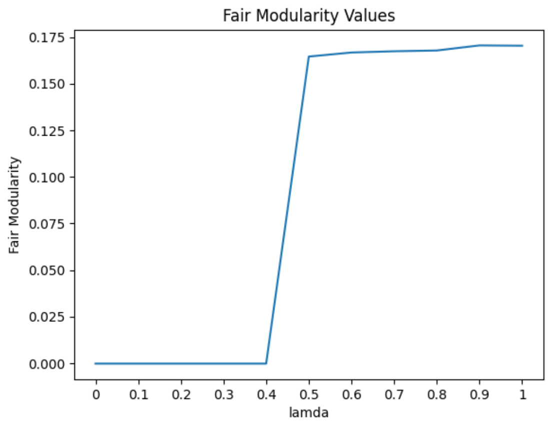


(a) Fair Modularity

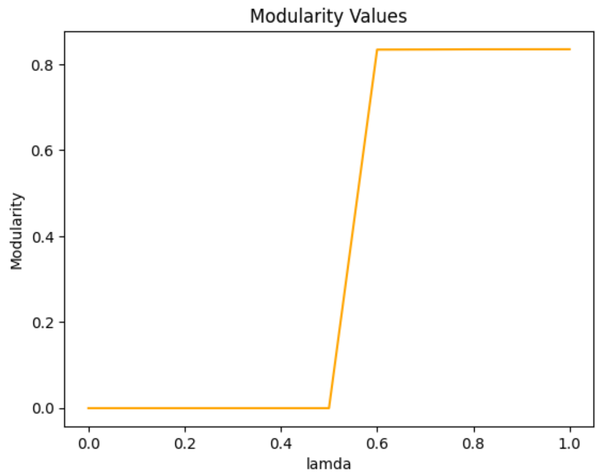


(b) Modularity

Figure 5.20: Fair Modularity Deezer

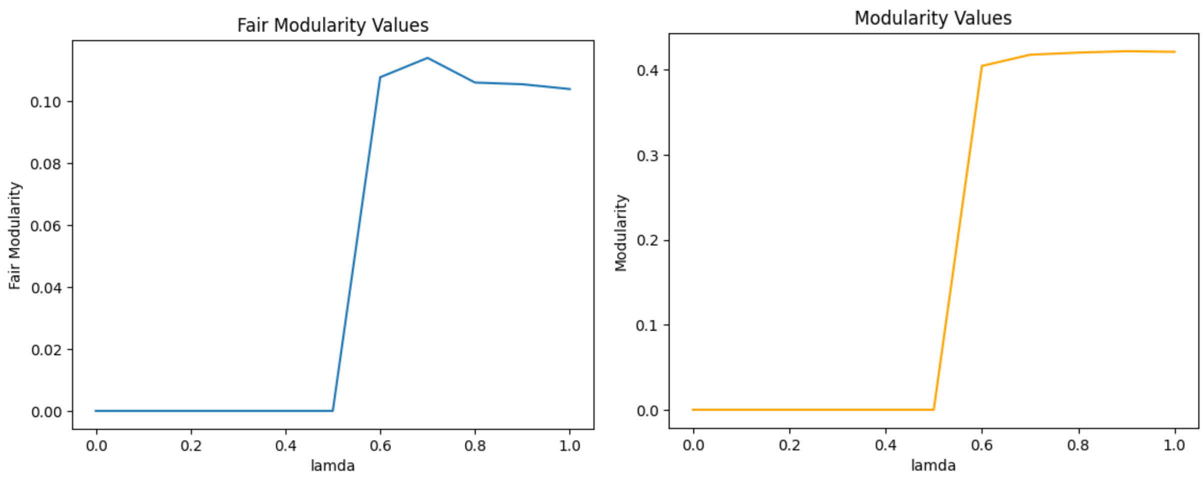


(a) Fair Modularity



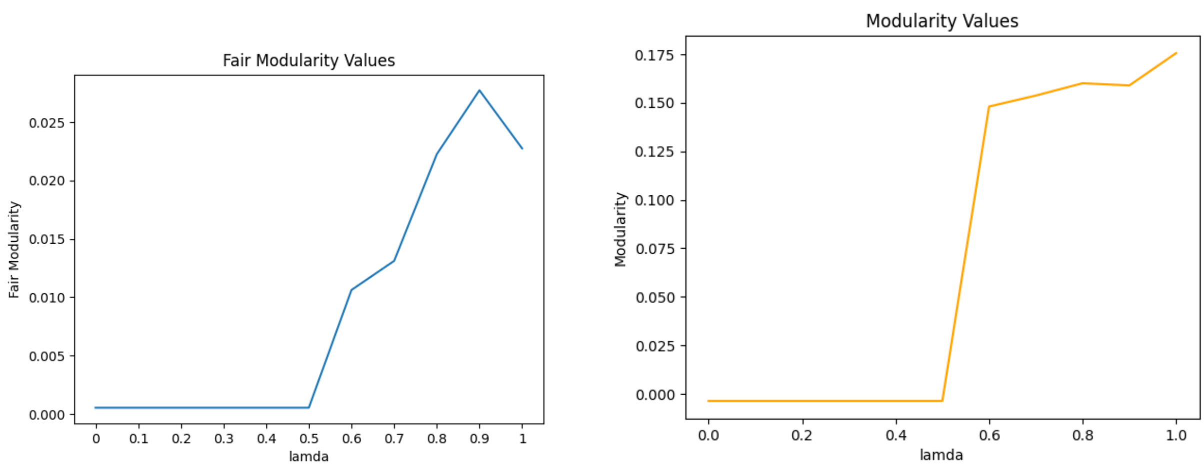
(b) Modularity

Figure 5.19: Fair Modularity Facebook



(a) Fair Modularity (b) Modularity

Figure 5.21: Fair Modularity Twitch



(a) Fair Modularity (b) Modularity

Figure 5.22: Fair Modularity NBA

In our results, we analyze the difference in connectivity density between red and blue nodes, represented by $|Q^R - Q^B|$ alongside the modularity value's impact on the network. Our findings reveal that for $\lambda \leq 0.5$ the algorithm tends to identify a number of communities equal to the number of nodes in the graph, thereby achieving the highest possible fairness in community connectivity. Conversely, for $\lambda > 0.5$, where the modularity value has a greater influence on the optimization process, the resulting community structure significantly improves compared to scenarios with lower λ. As the λ increases, the fair modularity value rises, indicating that the communities become more coherent. However, it's noteworthy that an increase in fair modularity (bearing in mind that values approaching 0 denote fairness) suggests a deviation from absolute fairness. This deviation implies that as communities become more defined and structured, the fairness in modularity—compared to that of communities identified by the original Louvain algorithm—starts to diverge from the ideal of zero, indicating a shift towards structured yet less evenly connected communities.

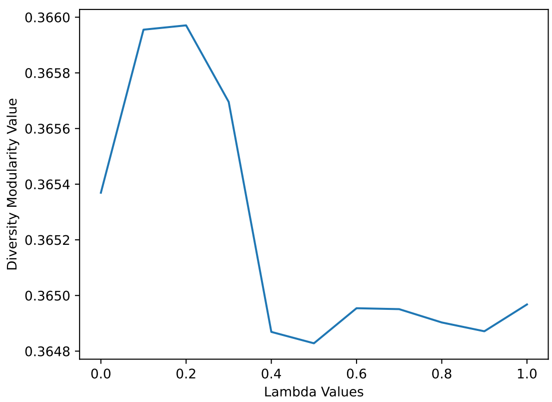
5.3.2 Diversity Modularity Analysis

In this section we compare the modified Louvain algorithm for community detection, aimed at enhancing community diversity by prioritizing edges between nodes of differing attributes, such as gender. The core of this modification is Diversity Modularity as introduced in 2.5, and 3.3 which serves as the objective function for optimization within the algorithm.

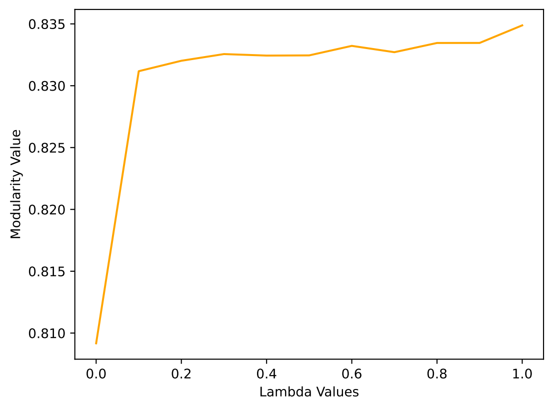
Diversity Modularity is designed to quantify the degree of diversity within detected communities by evaluating the distribution of edges connecting nodes with varied attributes. This metric is integrated into the optimization process of the Louvain algorithm, alongside the traditional Modularity metric, to guide the formation of diverse communities. The experimental framework incorporates a lambda (λ) parameter, which operates within a range of 0 to 1. This parameter dictates the trade-off between traditional Modularity and Diversity Modularity in the community detection process. The final optimization function is defined as:

$$Q_{diff} = 1 \times Q + (1 - \lambda) \times Q_D$$

The experiment evaluates the impact of varying λ values on the resulting community structures, specifically observing changes in both Modularity and Diversity Modularity metrics. This analysis aims to understand how the emphasis on diversity modularity influences the overall graph structure and the balance between achieving high modularity and enhancing diversity within communities. In the figures 5.23 5.24 5.25 5.26 we can see the results on real social networks.

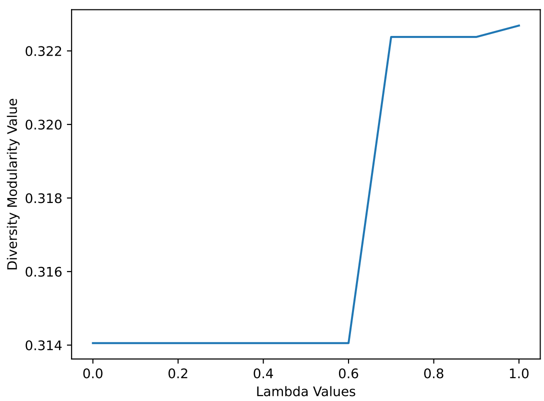


(a) Diversity Modularity

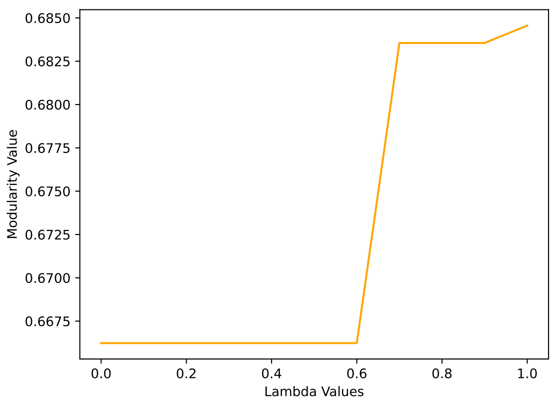


(b) Modularity

Figure 5.23: Fair Diversity Facebook

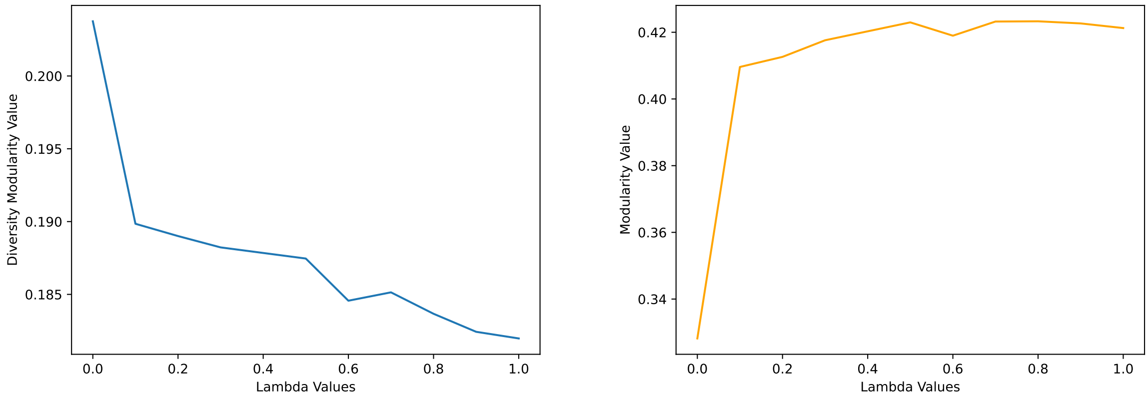


(a) Diversity Modularity



(b) Modularity

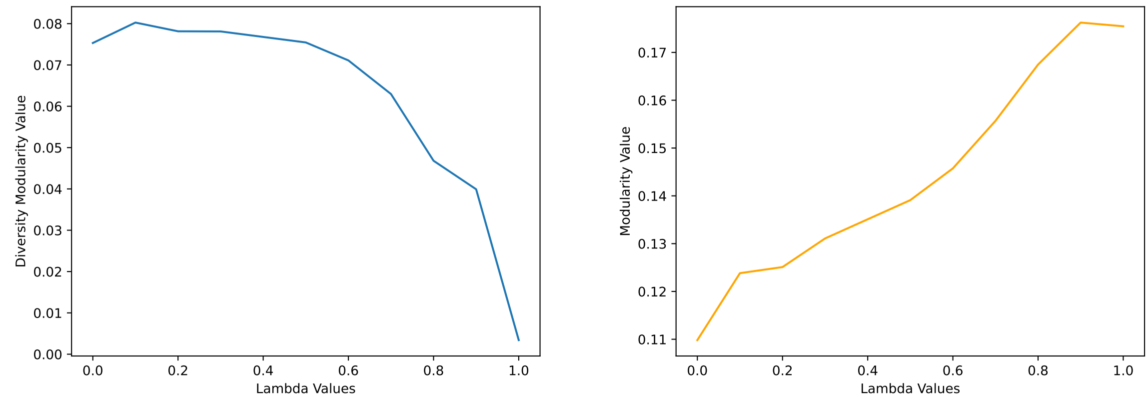
Figure 5.24: Fair Diversity Deezer



(a) Diversity Modularity

(b) Modularity

Figure 5.25: Fair Diversity Twitch



(a) Diversity Modularity

(b) Modularity

Figure 5.26: Fair Diversity NBA

In our observations, even with a $\lambda = 0$, where optimization is solely focused on diversity modularity, the impact on traditional modularity metrics is minimal. This phenomenon is particularly pronounced in well-structured networks, which appear to be less susceptible to alterations induced by prioritizing diversity. Consistent with our expectations, increasing lambda values correlate with a decrease in diversity modularity and an enhancement in traditional modularity metrics across real social networks. This trend validates our hypothesis regarding the interplay between diversity enhancement and structural integrity within community detection algorithms.

Chapter 6

Related Work

There has been a lot of recent research in fairness in machine learning [12, 13], although fairness for graph data has been less explored [1]. The balanced view approach of fairness was introduced in the seminal work of fairlets [2] for the case of clustering of non-graph data. It has been extended in various directions, such as for supporting more than one protected group [3], and for improving performance [14]. A balanced view has also been considered for graph data through a spectral clustering algorithm with fairness constraints [4].

Another view of fairness is that of ensuring results of equal quality for both groups by minimizing the clustering cost. This view is taken for non-graph data in the *socially fair* k-means clustering approach that seeks to minimize the maximum of the average k-means objective applied to each group [15] and in *equitable* clustering that seeks to minimize the distance of each point to its nearest center [16]. In a sense, modularity-based clustering follows this view, since its goal is maintaining good clustering quality in terms of intra-cluster connectivity.

There is also research on individual fairness in graphs e.g., [17]. Modularity-based fairness can be also applied at the individual node level, by looking at the connectivity of individual nodes; we leave this as future work. Finally, modularity has been refined to promote *mixed links*, i.e., links connecting nodes of different color in link recommendations [6]. In this paper, we focus on the connectivity of the protected group inside each community. It would be interesting to also look into promoting mixed links inside each community.

[18] research on community detection in weighted networks introduces a nuanced perspective on algorithmic fairness and modularity. They emphasize the complexity of weighted connections in networks, proposing intra-centrality and inter-centrality metrics to better understand community structures. This work aligns with the thesis's exploration of fairness in modularity-based community detection, offering a methodological foundation for evaluating the impact of algorithm modifications on fairness metrics. Their findings on the applications in Delay Tolerant Networks (DTN) and Online Social Networks (OSN) illustrate the practical implications of these algorithms, underscoring the relevance of fairness considerations in diverse network contexts.

One prominent method for evaluating community detection fairness is through the application of fairness metrics such as balance, distance definition [15], and individual fairness [19]. These metrics assess how well community detection algorithms represent protected attributes within identified communities, thus providing a nuanced understanding of algorithmic fairness. For instance, the balance metric evaluates the proportionate representation of different groups, offering insights into the demographic parity achieved by community detection algorithms.

Recent advancements in fair clustering have introduced novel perspectives and methodologies for integrating fairness into unsupervised learning tasks, such as community detection. A significant contribution in this domain has been made by [19], who proposed a local search-based algorithm for k-median and k-means clustering that addresses individual fairness. This work defines individual fairness in terms of ensuring that each point in a dataset is clustered with a center within a certain expected radius, thereby guaranteeing fair treatment across all data points. The algorithm achieves a bicriteria approximation that balances clustering quality with fairness, presenting a notable step forward in the pursuit of equitable algorithms in machine learning. This research complements the traditional focus on group fairness by highlighting the importance of treating individuals equitably within the clustering process, thereby enriching the discourse on fairness in community detection and offering new pathways for algorithmic development.

Another work that is related with in-processing procedure for community detection algorithm [20] introducing "I-Louvain," an algorithm for attributed graph clustering that enhances the original Louvain method by incorporating a new measure based on inertia alongside Newman's modularity. This dual approach allows for more efficient community detection in graphs by considering both the relationships and the attributes of vertices, showcasing an advance over existing methods that either focus solely on relational information or require categorical attributes. The experiments highlight I-Louvain's superior performance in creating meaningful clusters compared to methods like ToTeM, especially under conditions of data degradation or when handling numerical attributes. This demonstrates I-Louvain's robustness and its potential for broader application in community detection tasks.

Another related work on Louvain modification introduced by [21], discussing the integration of node attributes into the Louvain algorithm to enhance community detection. It introduces two modified algorithms, LAA (Louvain-AND-Attribute) and LOA (Louvain-OR-Attribute), focusing on how the inclusion of node attributes alongside traditional modularity optimization can improve the detection of cohesive groups within networks. This approach is compared against standard methods like Newman's Eigenvector, showcasing its effectiveness in achieving higher modularity and denser community partitions, especially in scenarios where both node attributes and structural properties are considered.

Chapter 7

Epilogue

7.1 Conclusions

This thesis has explored the intricate dynamics of fairness in modularity-based community detection within social networks. Through a meticulous examination of the modified Louvain algorithms tailored for fairness in connectivity, we have unveiled the delicate balance between optimizing for modularity and ensuring equitable representation within identified communities. Key findings from our comparative analysis underscore the potential for algorithmic adjustments to foster inclusivity without substantially compromising the structural integrity of community partitions. Specifically, the introduction of Fair Modularity and Diversity Modularity metrics has illuminated the path towards more nuanced and equitable community detection methodologies. Our research not only contributes to the burgeoning discourse on fairness in machine learning but also provides a practical framework for implementing fairness-aware algorithms in social network analysis.

7.2 Future Work

As part of the ongoing exploration into the fairness of community detection algorithms, a significant avenue for future research lies in the examination of alternative modularity approaches. Specifically, investigating the applicability and fairness outcomes of community detection algorithms from different modularity families in comparison to our work with the Louvain method. This approach not only broadens the spectrum of fairness evaluation but also contributes to a more holistic understanding of how different algorithms perform in real-world scenarios.

Moreover, there is a compelling need to extend our analysis to encompass a wider variety of real-world networks. Such an expansion would not only serve to validate our existing findings but also provide a deeper insight into the practical implications and effectiveness of fairness-aware community detection across diverse settings. This endeavor is crucial for bridging the gap between theoretical models and their real-world applicability, ensuring that our approaches are robust and versatile.

The development of new fairness metrics also represents a critical area for future exploration. Moving beyond the confines of Fair Modularity and Diversity Modularity, it is essential to investigate new dimensions of fairness. This exploration will enable a more comprehensive understanding of fairness in community detection, facilitating the integration of these considerations into algorithmic designs. Such advancements are key to evolving the field and ensuring that fairness is intricately woven into the fabric of community detection methodologies.

Finally, revisiting and optimizing our proposed algorithms is imperative. This involves a meticulous refinement of the mathematical foundations to better align with our fairness objectives, enhancing both the efficiency and intuition behind these algorithms. By focusing on the optimization of fairness-aware algorithms such as the modified Louvain method, future work can ensure that these approaches are not only theoretically sound but also practically applicable across a variety of network structures and datasets. This holistic approach to refinement and optimization is essential for advancing the pursuit of fairness in community detection, ensuring that our methods are both effective and aligned with the principles of equity and inclusivity.

Through these focused areas of future work, we aim to further the discourse and implementation of fairness in community detection, bridging theoretical models with practical applications to foster more inclusive and equitable digital communities.

Bibliography

[1] Y. Dong, Y. Ma, S. Wang, C. Chen, and J. Li, "Fairness in graph mining: A survey," *TKDE*, vol. To appear, 2023.

[2] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *NeurIPS*, 2017, pp. 5029–5037.

[3] S. K. Bera, D. Chakrabarty, N. Flores, and M. Negahbani, "Fair algorithms for clustering," in *NeurIPS*, 2019, pp. 4955–4966.

[4] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern, "Guarantees for spectral clustering with fairness constraints," in *ICML*, vol. 97, 2019, pp. 3458–3467.

[5] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, 2004.

[6] F. Masrour, T. Wilson, H. Yan, P. Tan, and A. Esfahanian, "Bursting the filter bubble: Fairness-aware network link prediction," in *AAAI*. AAAI Press, 2020, pp. 841–848.

[7] A. Stoica, C. J. Riederer, and A. Chaintreau, "Algorithmic glass ceiling in social networks: The effects of social recommendations on network diversity," in *WWW*. ACM, 2018, pp. 923–932.

[8] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, and N. Mamoulis, "Fairness-aware pagerank," in *WWW*, 2021.

[9] P. W. Holland, K. Laskey, and S. Leinhardt, "Stochastic blockmodels: First step," *Social Networks*, vol. 5, pp. 109–137, 1983.

[10] V. D. Blondel, J.-L. Guillaume, and E. L. R. Lambiotte, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, 2008.

[11] V. A. Traag, L. Waltman, and N. J. van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific Reports*, vol. 9, no. 1, p. 5233, Mar 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-41695-z

[12] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 115:1–115:35, 2022.

[13] E. Pitoura, K. Stefanidis, and G. Koutrika, "Fairness in rankings and recommendations: an overview," *VLDB J.*, vol. 31, no. 3, pp. 431–458, 2022.

[14] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner, "Scalable fair clustering," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 405–413.

[15] M. Ghadiri, S. Samadi, and S. S. Vempala, "Socially fair k-means clustering," in *FAccT*. ACM, 2021, pp. 438–448.

[16] M. Abbasi, A. Bhaskara, and S. Venkatasubramanian, "Fair clustering via equitable group representations," in *FAccT*, 2021, pp. 504–514.

[17] J. Kang, J. He, R. Maciejewski, and H. Tong, "Inform: Individual fairness on graph mining," in *KDD*. ACM, 2020, pp. 379–389.

[18] Z. Lu, X. Sun, Y. Wen, G. Cao, and T. L. Porta, "Algorithms and applications for community detection in weighted networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 11, pp. 2916–2926, 2015.

[19] S. Mahabadi and A. Vakilian, "Individual fairness for k-clustering," in *ICML*, vol. 119, 2020, pp. 6586–6596.

[20] D. Combe, C. Largeron, M. Géry, and E. Egyed-Zsigmond, "I-Louvain: An Attributed Graph Clustering Method," in *Intelligent Data Analysis*. Saint-Etienne, France: LaHC, University of Saint-Etienne, France, Oct. 2015. [Online]. Available: https://ujm.hal.science/ujm-01219447

[21] Y. Asim, R. Ghazal, W. Naeem, A. Majeed, B. Raza, and A. K. Malik, "Community detection in networks using node attributes and modularity," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 1, 2017. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2017.080148

Short Biography

Konstantinos Manolis, born in 1997, successfully completed his undergraduate studies in Applied Informatics at the University of Macedonia in 2021. Driven by a passion for Big Data Analysis, he conducted his undergraduate thesis on "Entity Resolution on Heterogeneous Big Data." This endeavor fueled his decision to further his academic pursuits through a post-graduate program in "Data and Computer Systems Engineering" at the University of Ioannina. During his academic journey, Konstantinos cultivated a keen interest in Social Network Analysis, leading to the exploration of the topic in his thesis titled "Modularity-Based Fairness in Network Communities." Concurrently, his professional endeavors as a Software Engineer equipped him with expertise in AI technologies and frameworks. Notably, he contributed significantly to the field by developing a Text to SQL framework for the QRPatrol application. This framework proved invaluable for efficient data extraction, benefiting both Terracom S.A. staff and its clientele. Furthermore, his scholarly achievements include a collaborative publication titled "Modularity-Based Fairness in Community Detection" at the ASONAM 23 convention, co-authored with his supervisor Evangelia Pitoura, showcasing his dedication to advancing the field of Social Network Analysis through rigorous academic research.