CrossWalk: Fairness-enhanced Node Representation Learning

Ahmad Khajehnejad[1] , **Moein Khajehnejad**[2] , **Mahmoudreza Babaei**[3] ,
Krishna P. Gummadi[4] , **Adrian Weller**[5,6] , **Baharan Mirzasoleiman**[7]

[1]Dept of Computer Engineering, Sharif University of Technology
[2]Dept of Data Science and AI, Monash University
[3]Max Planck Institute for Human Development
[4]Max Planck Institute for Software Systems
[5]University of Cambridge
[6]The Alan Turing Institute
[7]Dept of Computer Science, University of California Los Angeles

khajenezhad@ce.sharif.edu, moein.khajehnejad@monash.edu, babaei@mpib-berlin.mpg.de

Abstract

The potential for machine learning systems to amplify social inequities and unfairness is receiving increasing popular and academic attention. Much recent work has focused on developing algorithmic tools to assess and mitigate such unfairness. However, there is little work on enhancing fairness in graph algorithms. Here, we develop a simple, effective and general method, CrossWalk, that enhances fairness of various graph algorithms, including influence maximization, link prediction and node classification, applied to node embeddings. CrossWalk is applicable to any random walk based node representation learning algorithm, such as DeepWalk and Node2Vec. The key idea is to bias random walks to cross group boundaries, by upweighting edges which (1) are closer to the groups' peripheries or (2) connect different groups in the network. CrossWalk pulls nodes that are near groups' peripheries towards their neighbors from other groups in the embedding space, while preserving the necessary structural information from the graph. Extensive experiments show the effectiveness of our algorithm to enhance fairness in various graph algorithms, including influence maximization, link prediction and node classification in synthetic and real networks, with only a very small decrease in performance.

1 Introduction

Fairness in machine learning is receiving growing attention as algorithmic systems are increasingly deployed across society in ways that have significant impact on people's lives. Decisions made by such systems often affect different population subgroups disproportionately [Kirchner and Mattu, 2016; Osoba and Welser IV, 2017]. As a result, fairness — the absence of prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics — has received much recent interest (see Mehrabi *et al.*, 2019 for a survey).

Despite the emerging body of efforts to enhance algorithmic fairness in machine learning algorithms, there has been little work on enhancing fairness in graph algorithms. Considering the far-reaching application of graphs to many important problems in sociology, finance, computer science, and operations research [Easley *et al.*, 2010], it becomes crucial to develop methods to enhance fairness in graph algorithms, to prevent outcomes skewed for or against a particular group of people. A concrete example is promoting fairness of influence maximization in social networks, which aims ultimately to reach different population groups roughly equally. This becomes critical for receiving important information, such as job opportunities or loan advertisements, by various communities. Another example is enhancing fairness in node classification in social networks, which aims to limit disparity in the prediction accuracy across different groups or social communities.

Here, we develop a simple, intuitive and effective approach, CrossWalk, to promote fairness in the results of graph algorithms applied to node embeddings produced by random walk-based node representation learning algorithms. The key idea of our method is to bias random walks to cross group boundaries, by upweighting edges which are (1) closer to the groups' peripheries or (2) connecting different groups in the network. Several methods for representation learning on graphs, including DeepWalk [Perozzi *et al.*, 2014] and Node2Vec [Grover and Leskovec, 2016], leverage random walks to preserve higher-order proximity between nodes and learn the corresponding representations. When transition probabilities in random walks are chosen by CrossWalk, random walks initiated from a particular group will be pulled toward the group boundaries and have a higher probability of crossing groups' peripheries and visiting nodes from other groups in the network. In doing so, CrossWalk pulls closer nodes that are near groups' peripheries towards their neighbors from other groups in the embedding space, while preserving the necessary structural information from the graph. The resulting representation promotes fairness for the original graph in the result of various graph algorithms, including influence maximization, node classification, and link prediction. Beyond representation learning, our method is applicable to any graph algorithm which works by stochastic traversal of network edges, such as the classical influence maximization based on the Independent Cascade (IC) model, which we discuss in the Appendix.

We conduct experiments on synthetic and real-work networks to evaluate the effectiveness of our proposed method. We first apply our method to learn node representations, using DeepWalk and Node2Vec in a number of synthetic networks, as well as two real-world networks, namely, Rice-Facebook [Mislove *et al.*, 2010] and a subset of Twitter [Babaei *et al.*, 2016; Cha *et al.*, 2010]. We then apply various graph algorithms, including influence maximization, node classification, and link prediction to the obtained node representations. We show that CrossWalk is very effective in enhancing fairness of the aforementioned graph algorithms in synthetic and real networks, without a significant compromise in the total performance of the algorithms.

2 Related Work

There has been an emerging body of efforts to enhance algorithmic fairness in machine learning algorithms. However, few works have focused on detecting and enhancing unfairness in graph algorithms such as influence maximization, node classification, and link prediction. In this section, we review recent works on network embedding, and fairness in influence maximization, node classification, and link prediction.

2.1 Representation Learning on Graphs

Node representation learning algorithms attempt to map the nodes in a graph to a lower dimensional space, such that the network structure is preserved. Among the most well-known approaches are embedding methods based on random-walks [Khajehnejad, 2019; Grover and Leskovec, 2016], deep learning architectures [Wang *et al.*, 2016], and graph neural networks [Hamilton *et al.*, 2017]. Most related methods to our work are DeepWalk and Node2Vec, that are two widely used random walk based methods for deriving node representations.

DeepWalk DeepWalk [Perozzi *et al.*, 2014] takes a graph G and iteratively (1) initiates a random walk from a randomly sampled vertex, and (2) updates the node representations, by optimizing the Skip-gram likelihood objective [Mikolov *et al.*, 2013], using a hierarchical soft-max. DeepWalk preserves higher-order proximity between nodes by maximizing the probability of observing the last d nodes and the next d nodes in the random walk centered at v_i, i.e. maximizing $\log P(v_{i-d}, \cdots, v_{i-1}, v_{i+1}, \cdots, v_{i+d}|\Phi_i)$, where $2d+1$ is the length of the random walk and Φ_i is the representation of node v_i. The model performs the optimization over sum of log-likelihoods for each random walk.

Node2Vec Similar to DeepWalk, Node2Vec [Grover and Leskovec, 2016] preserves higher-order proximity between nodes by maximizing the probability of occurrence of subsequent nodes in fixed length random walks. The crucial difference from DeepWalk is that Node2Vec employs biased-random walks that provide a trade-off between breadth-first (BFS) and depth-first (DFS) graph searches, and hence produces higher-quality and more informative embeddings than DeepWalk. Choosing the right balance enables Node2Vec to preserve community structure and structural equivalence between nodes.

2.2 Fair Influence Maximization

Influence maximization aims at finding an initial set of nodes to maximize the number of further adopters [Richardson and Domingos, 2002]. While finding the optimal solution for influence maximization is NP-hard [Kempe *et al.*, 2003], a simple greedy algorithm provides a $1-1/e$-approximation guarantee under different cascade models such as Linear Threshold (LT) and Independent Cascade (IC). Since then, extensive research has focused on studying different variations [Goyal *et al.*, 2013; Carnes *et al.*, 2007] among which Keikha *et al.* [2020] take advantage of a network embedding approach by applying a k-means method on the embedding space to select the resulting k cluster centroids as initial seeds.

Among the wide range of recent works on influence maximization, only a few has considered the fairness. After Babaei *et al.* [2016] showed that users in social media select their sources sub-optimally in the sense of receiving diverse information, a group of studies have worked on diversifying the initial seeds [Benabbou *et al.*, 2018; Aghaei *et al.*, 2019]. However, these works still fail to take into account the fairness criterion. Network embedding approaches for fair influence maximization which have been introduced recently, [Khajehnejad *et al.*, 2020] proposed adversarial network representation learning for enhancing fairness of influence maximization. We use this method as a baseline in our work.

2.3 Fair Node Classification

Node classification determines the labelling of nodes by looking at the labels of their neighbours. Recent node classification methods work by classifying the learned nodes representations. A few existing work considered finding fair representations for classification [Zemel *et al.*, 2013; Lahoti *et al.*, 2019]. The key idea of the above work is that individuals who are deemed similar according to a task-specific similarity metric should receive similar outcomes. However, we are not aware of any fariness-enhanced node representation learning algorithm for classification in graphs.

2.4 Fair Link Prediction

Link prediction infers new or previously unknown relationships of a network. However, existing algorithms are susceptible to promoting links that may lead to increased segregation. Among the few existing methods to enhance fairness of link prediction, [Rahman *et al.*, 2019] introduced FairWalk, a modified random walk, which results in a more diverse network neighborhood representation thereby producing less biased graph embedding. FairWalk, however, fails to enhance fairness in graphs where the majority of nodes are more than one hop away from group peripheries. More recently, Masrour *et al.* [2020]; Bose and Hamilton [2019] proposed methods that combine adversarial network representation learning with supervised link prediction to enhance fairness of link prediction, and Buyl and De Bie [2020] proposed a Bayesian method which utilizes a biased prior in the embedding phase to generate fair node representations.

Existing methods to enhance fairness of node representation learning are specific to a specific graph algorithm. On the other hand, we propose a general and effective method to enhance fairness of node representation learning that is applicable to various graph algorithms. While FairWalk Rahman *et al.* [2019] was originally proposed to enhance fairness of link prediction, we use it as a baseline for various graph algorithms in our work.

3 Problem Formulation

Consider a directed network $G = (V, E)$ with a set V of nodes and a set E of edges. We denote by $w_{uv} \in \mathbb{R}$ the weight of an edge $(u, v) \in E$. Assume the nodes are partitioned into C groups $\{V_1, \cdots, V_C\}$. Furthermore, let l_v indicate the group that node v belongs to, and $\mathcal{N}(v) = \{u|(v, u) \in E\}$ be the set of nodes in v's immediate neighborhood. Many algorithms for representation learning on graphs, including DeepWalk and Node2Vec, leverage random walks to preserve higher-order proximity between nodes and learn their representations.

Random Walk: Given a source node u_i, a random walk $\mathcal{W}_{u_i}$ of length d rooted at u_i is a sequence of vertices $u_i, u_{i+1}, \cdots, u_{i+(d-1)}$, not necessarily distinct, such that (u_i, u_{i+1}) is an edge in the graph. Formally, nodes u_i are generated according to the following distribution:

$$P(u_i = u|u_{i-1} = v) = \begin{cases} \pi_{vu} & if (i,j) \in E \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where π_{ij} is the normalized transition probability between nodes i and j. In a weighted network, normalized edge weights w_{uv} can be used as transition probabilities π_{uv} of the random walk.

Fairness Metric (Disparity): Let $Q \in \mathbb{R}$ be the performance of algorithm A on the entire graph, and $Q_i \in \mathbb{R}$ be the performance of A on group $i \in [C]$ in the graph. For instance, Q can be the fraction of nodes that are infected at the end of a network diffusion process, and Q_i can be the fraction of infected nodes in group i. Our goal is to modify the weight w_{uv} of every edge $(u, v) \in E$ to w'_{uv} so that when used by random walks to produce node embedding, the performance of A applied to the produced representations has a higher fairness and smaller discrepancy on different groups of the underlying graph. More formally, we will propose a mapping from the original edge weights $\{w_{uv}\}$ to modified weights $\{w'_{uv}\}$ which will be used by random walks to produce node representations. The aim is that A applied to node embeddings achieves a low value of

$$\text{disparity}(A) = Var(\{Q_i\} : i \in [C]) \quad (2)$$

Note that the performance Q, e.g., the percentage of infected nodes is measured on the graph with original edge weights $\{w_{uv}\}$.

4 Our Method: CrossWalk

The key idea of our method is to assign larger weights to (1) the edges that are connected to nodes closer to the groups' periphery, and (2) the edges that are connecting nodes from different groups. Intuitively, our reweighting method biases random walks initiated in a given group towards visiting nodes on the group boundary and eventually crossing the boundaries and visiting nodes from other groups in the graph. A schematic diagram of the proposed method is illustrated in Figure 1.

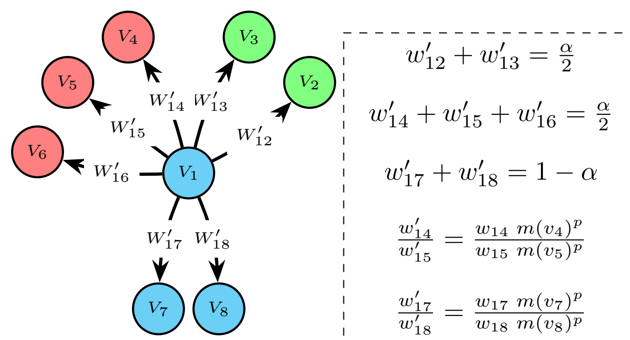


Figure 1: Illustrating CrossWalk. Different colors represent nodes from different groups. Our method upweights edges which are (1) closer to the groups' peripheries or (2) connecting different groups in the network.

4.1 Bias towards Group Boundaries

For every node v we define a measure of *proximity* to other groups in the graph. Intuitively, for every node v, its proximity $m(v)$ to other groups indicates the fraction of nodes from other groups in v's close proximity. To calculate $m(v)$, we initiate a set of r truncated random walks of length d from a node v based on the original edge weights. We define v's proximity to other groups, $m(v)$, as the expected number of times nodes from other groups are visited in r random walks of length d rooted at v. Formally, we have

$$m(v) = \frac{\sum_{j \in [r]} \sum_{u \in \mathcal{W}_v^j} \mathbb{I}[l_v \neq l_u]}{r \times d}. \quad (3)$$

Nodes that are closer to group boundaries and have a larger number of nodes with a different label in their close proximity has a higher value of m. Assigning larger weights to the edges connecting nodes with larger proximity values biases reweighted random walks towards visiting nodes on group boundaries in the graph.

4.2 Bias towards Other Groups

In addition, to bias the stochastic edge traversal procedure towards visiting nodes from other groups, we upweight the edges connecting different groups in the graph. More specifically, for a node v we denote the set of groups in v's immediate neighborhood by $R_v = \{\cup_{u \in \mathcal{N}(v)} l_u | l_v \neq l_u\}$. Moreover,

Algorithm 1 CrossWalk: Fairness-enhanced node embedding

Require: Graph $G = (V, E)$, Edge weights w_{uv} $\forall (v, u) \in E$, Parameters α, p.
Ensure: weights w'_{vu} $\forall (v, u) \in E$.
for $v \in V$ **do** ▷ Calculating closeness to boundary
  Run r random walks $\mathcal{W}_v^j, j \in [r]$ rooted at v.
  $m(v) = \sum_{j \in [r]} \sum_{u \in \mathcal{W}_v^j} \mathbb{I}[l_v \neq l_u]/(r \times d)$.
end for
for $v \in V$ **do** ▷ Reweighting edges
  $N_v = \{u \in \mathcal{N}(v) | l_u = l_v\}$
  $N_v^c = \{u \in \mathcal{N}(v) | l_v \neq l_u = c\}$
  $R_v = \{\cup_{u \in \mathcal{N}(v)} l_u | l_v \neq l_u\}$
  $Z = \sum_{u \in N_v} w_{vu} \times m(u)^p$
  for $u \in N_v$ **do** ▷ Edges in same group
    $w'_{vu} = w_{vu} \times (1 - \alpha) \times m(u)^p / Z$
  end for
  for $c \in R_v$ **do**
    $Z = |R_v| \times \sum_{u \in N_v^c} w_{vu} \times m(u)^p$
    for $u \in N_v^c$ **do** ▷ Edges connecting different groups
      if $N_v \neq \emptyset$ **then**
        $w'_{vu} = w_{vu} \times \alpha \times m(u)^p / Z$
      else
        $w'_{vu} = w_{vu} \times m(u)^p / Z$
      end if
    end for
  end for
end for

we denote the set of v's neighbors within the same group by $N_v = \{u \in \mathcal{N}(v) | l_u = l_v\}$, and the set of v's neighbors that belong to another group $c \neq l_v$ by $N_v^c = \{u | u \in \mathcal{N}(v) | l_v \neq l_u = c\}$. Now, for parameters $\alpha \in (0, 1)$, $p > 0$, we weight v's outlinks to its neighbors $u \in \mathcal{N}_v$ as follows: if u belongs to the same group, i.e. $l_u = l_v$, we multiply w_{vu} by $(1 - \alpha) \times m(u)^p$, and if u belongs to another group, i.e. $l_v \neq l_u$, we multiply w_{vu} by $\alpha \times m(u)^p$. The parameter p controls the degree of biasness of random walks towards visiting nodes at group boundaries. We normalize the edge weights such that the sum of the weights of edges connecting v to its neighbors from the same group is $1 - \alpha$, and the sum of the weights of edges connecting v to its neighbors from any other group is $\alpha / |R_v|$. Formally, the new edge weights can be derived as follows:

$$w'_{vu} = \begin{cases} w_{vu}(1-\alpha) \times \frac{m(u)^p}{\sum_{z \in N_v} w_{vz} m(z)^p}, & if u \in \mathcal{N}(v), l_v = l_u \\ w_{vu} \alpha \times \frac{m(u)^p}{|R_v| \sum_{z \in N_v^c} w_{vz} m(z)^p}, & if u \in \mathcal{N}(v), l_v \neq l_u = c. \end{cases} \tag{4}$$

Larger α upweights edges connecting different groups, and biases the stochastic edge traversal procedure towards visiting nodes belonging to other groups in the graph. The pseudocode of CrossWalk is shown in Alg.1.

4.3 CrossWalk Enhances Fairness

Our reweighting method upweights the edges that are closer to the groups' periphery and those connecting different groups. When transition probabilities in random walks are chosen based on our reweighting strategy, the random walks initiated by representation learning algorithms spend more time visiting nodes at group boundaries. Hence, CrossWalk pulls nodes that are near groups' peripheries towards their neighbors from other groups in the embedding space, while preserving the necessary structural information from the graph. Figure 2 compares the DeepWalk projection of node embeddings of the original and the reweighted graph by CrossWalk on a 2-D space. We observe that the embeddings of the two groups are pulled towards each other in the reweighted graph. For networks in which the majority of nodes are more than one hop away from group peripheries, larger value of p bias random walks towards other groups and result in fairness-enhanced representations. For networks with dense connected components, larger values of α have a similar effect. We experimentally study the effect of α, p in the Appendix.

5 Applications

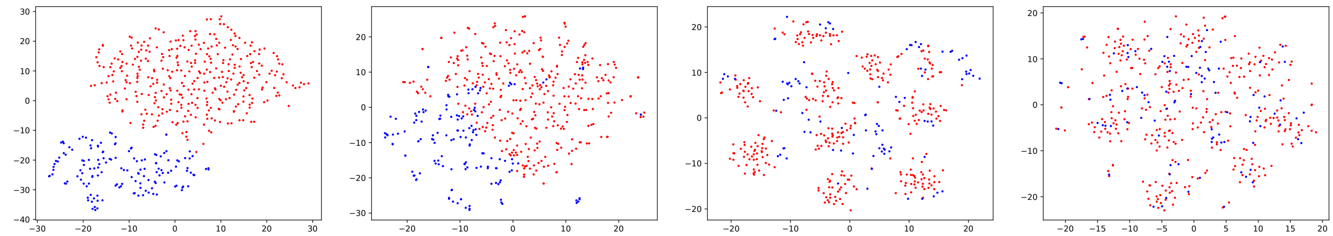
In this section, we discuss applications of our proposed reweighting strategy to enhance fairness in the result of random walk based graph algorithms, including influence maximization, node classification, and link predictions in graphs. In the Appendix, we consider the classical influence maximization problem based on the Independent Cascade (IC) model, and discuss the application of our reweighting method to reduce disparity of influenced individuals from various groups.

Influence Maximization Having the node representations learned by CrossWalk applied to DeepWalk or Node2Vec, the most influential individuals can be found as the set of most centrally located nodes in the representation space. These nodes are medoids of the node representations, and can be found by minimizing the pairwise dissimilarities between nodes within a cluster and a node designated as the center of that cluster in the representation space. For a specific value of k, the set of k-medoids can be found as follows:

$$S^* \in \arg \min_{\substack{S \subseteq V, \\ |S| \leq k}} \sum_{i \in V} \min_{j \in S} \|\Phi_i - \Phi_j\|_2 \tag{5}$$

A common algorithm to find the set of k-medoids starts by selecting k nodes uniformly at random. Then, it iteratively assigns each node to the cluster defined by the nearest medoid, updates the medoids within the new clusters, and repeats this procedure as long as the sum of distances between the nodes and their corresponding medoid in the representation space is decreasing.

We calculate the number of infected individuals by simulating diffusion started from the selected seeds S^* in the original graph that is not reweighted by CrossWalk. We use Independent Cascade (IC) model, in which at every time step $t > 0$, a node $u \in V$ which was activated at time $t - 1$ can activate its inactivated neighbor v with probability P_{uv}. The diffusion stops at time $t > 0$ if no new node gets activated. In a weighted network, a multiplication of the original edge weights w_{uv} can be used as transmission probabilities P_{uv} in the IC model.



(a) Synthetic Layered Dataset DeepWalk

(b) Synthetic Layered Dataset CrossWalk ($\alpha = 0.5, p = 2$)

(c) Rice-Facebook Dataset DeepWalk

(d) Rice-Facebook Dataset CrossWalk ($\alpha = 0.5, p = 4$)

Figure 2: Distributions of the DeepWalk embedded nodes from the two groups. Embeddings of the two groups are pulled towards each other in the graph reweighted by CrossWalk.

Node Classification Node representations learned by CrossWalk applied to DeepWalk and Node2Vec can be used for node classification in graphs. Here, we consider Label Propagation (LP) to classify nodes based on the obtained representations. LP first constructs a k-nearest graph from the data and initializes all the nodes with a unique label. Then, it iteratively assigns to every node the label with the highest frequency among its neighbors. This process is repeated until every node has the same label as the majority of its neighbors in the graph.

Link Prediction Node representations obtained by CrossWalk applied to DeepWalk and Node2Vec can be used to detect new or formerly unknown connections in a network. To do so, we train a logistic regression on edges. For nodes v and u with representation vectors r_v and r_u, the feature vector used for the link (v, u) is $(r_v - r_u)^{\circ 2}$, where $\circ$ denotes Hadamard power. For evaluation, we randomly select 10% of the existing edges as positive test data and do not use them when obtaining node embedding. We also select equal number of non-existing edges as negative test data. The rest of the existing edges in the graph and an equal number of non-existing edges are used for training.

CrossWalk & Graph Algorithms As discussed in Sec. 4.3, CrossWalk pulls together representation of nodes belonging to different groups. In the influence maximization problem, the most centrally located nodes in the representation space will have a larger proximity to multiple groups. Therefore, the selected seeds can spread the information more effectively in multiple groups. Similarly, in the node classification and edge prediction problems, as similar nodes or edges belonging to various groups become closer in the representation space, they share the same label irrespective of their group. This enhances fairness and decreases the disparity of test accuracy in different groups.

6 Experiments

In this section we evaluate the effectiveness of CrossWalk for reducing disparity of influence maximization, link prediction and node classification on two real and two synthetic datasets. We use DeepWalk [Perozzi *et al.*, 2014], FairWalk [Rahman *et al.*, 2019] and Adversarial graph embedding [Khajehnejad *et al.*, 2020] as baseline methods. In all experiments CrossWalk and FairWalk use DeepWalk embedding, unless we explicitly mention that Node2vec is used. Our experiments confirm the superiority of CrossWalk over baselines in different tasks.

6.1 Dataset

Rice-Facebook Dataset: This dataset [Mislove *et al.*, 2010] is an undirectd graph of friendships between students at the Rice university. The dataset contains 1205 nodes and 42443 edges. Node features include age, college and major. We use students' ages as the sensitive attribute. We consider the students with age 20 as group A and the students with ages 18 and 19 as group B, and exclude the nodes with ages higher than 20. Group A has 344 nodes and 7441 inner-group connections. Group B has 97 nodes and 513 inner-group connections. There are 1779 connections between the two groups.

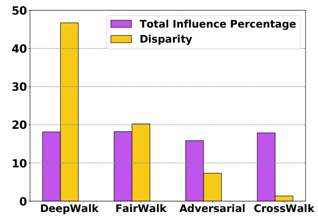
Twitter Dataset: We consider an undirected and connected sub-graph of the Twitter dataset [Babaei *et al.*, 2016; Cha *et al.*, 2010] with 3560 nodes. The nodes can be divided into three groups based on their political learning: neutrals (group A) with 2598 nodes, (group B) liberals with 782 nodes, and (group C) conservatives with 180 nodes. The number of intra-group and inter-group connections are $e^{A}_{intra} = 3724$, $e^{B}_{intra} = 950$, $e^{C}_{intra} = 74$, $e^{AB}_{inter} = 1461$, $e^{AC}_{inter} = 359$ and $e^{BC}_{inter} = 109$.

Synthetic Datasets: We consider two undirected synthetic datasets. Our first synthetic network consists of two groups with $n_A = 350$ and $n_B = 150$ nodes that are connected with intra-group probabilities $P^{A}_{intra} = P^{B}_{intra} - 0.025$ and inter-group probability $P^{AB}_{intere} = 0.001$.

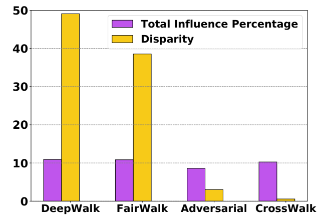
Our second synthetic network consists of three groups with $n_A = 300$, $n_B = 125$ anad $n_C = 75$ nodes. Nodes are connected with intra-group probabilities of $P^{A}_{intra} = P^{B}_{intra} = P^{C}_{intra} = 0.025$ and inter-group probabilities of $P^{AB}_{inter} = 0.001$ and $P^{AC}_{inter} = P^{BC}_{inter} = 0.0005$.

6.2 Influence Maximization

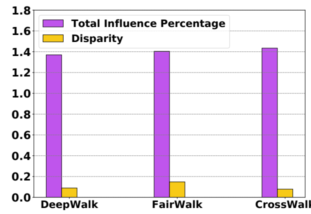
To find the set of most influential nodes, we apply k-medoids with $k = 40$ to node representations obtained by DeepWalk and Node2vec. We also report the performance of Adversarial Embedding [Khajehnejad *et al.*, 2020] on Rice-Facebook



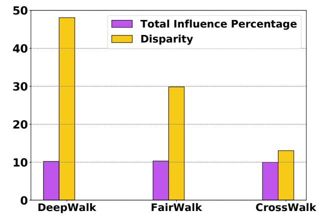
(a) Rice-Facebook, $\alpha = 0.5, p = 4, k = 40$.



(b) 2-grouped synthetic dataset, $\alpha = 0.7, p = 4$.



(c) Twitter dataset, $\alpha = 0.5, p = 2$.



(d) 3-grouped synthetic dataset, $\alpha = 0.7, p = 4$.

Figure 3: Influence Maximization

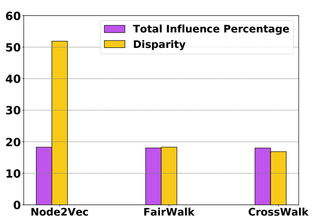


Figure 4: Influence Maximization - CrossWalk and FairWalk on Node2vec

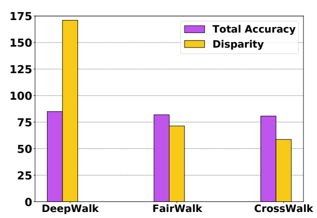
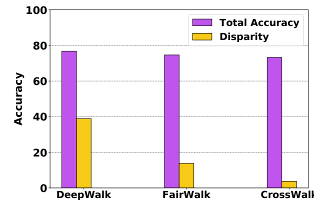


Figure 5: Node Classification - Rice-Facebook dataset. $\alpha = 0.5, p = 1$.

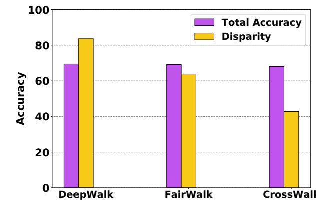
and our first synthetic network, as this method is only applicable to networks of 2 groups. To calculate the number of infected individuals, we consider Independent Cascade model (IC) with a constant activation probability for all the edges in the original network (0.01 for real datasets and 0.03 for synthetic networks). We calculate disparity according to Eq. (2), and report the averaged results of 5 runs for each method.

Figures 3a and 3b show total influence and disparity of different methods on the Rice-Facebook dataset and the 2-grouped synthetic dataset, respectively. We observe that CrossWalk results in a significant decrease in disparity compared to FairWalk and Adversarial embedding, with a very small decrease in the total influence. Figures 3c and 3d demonstrate the same results for the Twitter and our second synthetic dataset, each with 3 groups. Interestingly, while FairWalk increased the disparity, CrossWalk decreased the disparity and improved the total influence, compared to DeepWalk.

We also apply CrossWalk to Node2Vec on Rice-Facebook dataset. Figure 4 compares the performance of CrossWalk with $\alpha = 0.5, p = 4$, and FairWalk applied to Node2Vec with $p = 0.5, q = 0.5$. We observe that CrossWalk outperforms FairWalk in reducing disparity.



(a) Rice-Facebook Dataset, $\alpha = 0.5, p = 2$.



(b) Twitter Dataset, $\alpha = 0.5, p = 2$

Figure 6: Link Prediction

6.3 Node Classification

We use the Rice-Facebook dataset for classification, using the students' ages as the sensitive attribute and their college IDs as their class labels. We measure disparity according to Equation (2). Figure 5 shows the accuracy and disparity of Label Propagation (LP) with $k = 7$, applied to the representations obtained by FairWalk and CrossWalk applied to DeepWalk. We see the superiority of CrossWalk in reducing disparity of node classification.

6.4 Link Prediction

In the Rice-Facebook dataset with two groups of nodes, A and B, there exist three types of links; A to A, B to B, and A to B connections. Similarly, in the Twitter dataset with three groups of nodes there exist six types of links. For each connection type, we select equal number of positive and negative test samples (10% of each group). We train a logistic regression on the embeddings obtained by FairWalk and CrossWalk applied to DeepWalk. Figures 6a and 6b illustrate the total accuracy and disparity according to Eq. (2), for link prediction. It confirms the superiority of CrossWalk over FairWalk in reducing disparity with a slight decrease in accuracy.

7 Conclusion

In this work, we developed a simple, and effective approach to enhance fairness in the results of graph algorithms which work on random walk based node embeddings. The key idea of our method is to upweight the edges that are (1) closer to the groups' peripheries or (2) connecting different groups in the network. We applied our reweighting strategy to DeepWalk and Node2Vec, and used the obtained representations to address fairness-enhanced influence maximization (for the original graph), node classification, and link prediction. Our extensive experiments confirmed the effectiveness of our algorithm to enhance fairness of various graph algorithms on synthetic and real networks.

Acknowledgements

Adrian Weller acknowledges support from a Turing AI Fellowship under grant EP/V025379/1, The Alan Turing Institute under EPSRC grant EP/N510129/1 and TU/B/000074, and the Leverhulme Trust via the CFI.

References

Sina Aghaei, Mohammad Javad Azizi, and Phebe Vayanos. Learning optimal and fair decision trees for nondiscriminative decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1418–1426, 2019.

Mahmoudreza Babaei, Przemyslaw Grabowicz, Isabel Valera, Krishna P Gummadi, and Manuel Gomez-Rodriguez. On the efficiency of the information networks in social media. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, pages 83–92, 2016.

Nawal Benabbou, Mithun Chakraborty, Xuan-Vinh Ho, Jakub Sliwinski, and Yair Zick. Diversity constraints in public housing allocation. In *AAMAS*, pages 973–981, 2018.

Avishek Joey Bose and William L Hamilton. Compositional fairness constraints for graph embeddings. *arXiv preprint arXiv:1905.10674*, 2019.

Maarten Buyl and Tijl De Bie. Debayes: a bayesian method for debiasing network embeddings. *arXiv preprint arXiv:2002.11442*, 2020.

Tim Carnes, Chandrashekhar Nagarajan, Stefan M Wild, and Anke Van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *EC*, pages 351–360. ACM, 2007.

Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring user influence in Twitter: the million follower fallacy. In *Proceedings of AAAI Conference on Weblogs and Social Media (ICWSM'2010)*, May 2010.

David Easley, Jon Kleinberg, et al. *Networks, crowds, and markets*, volume 8. Cambridge university press Cambridge, 2010.

Amit Goyal, Francesco Bonchi, Laks VS Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social network analysis and mining*, 3(2):179–192, 2013.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

Mohammad Mehdi Keikha, Maseud Rahgozar, Masoud Asadpour, and Mohammad Faghih Abdollahi. Influence maximization across heterogeneous interconnected networks based on deep learning. *Expert Systems with Applications*, 140:112905, 2020.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

Moein Khajehnejad, Ahmad Asgharian Rezaei, Mahmoudreza Babaei, Jessica Hoffmann, Mahdi Jalili, and Adrian Weller. Adversarial graph embeddings for fair influence maximization over social networks. *arXiv preprint arXiv:2005.04074*, 2020.

Moein Khajehnejad. Simnet: Similarity-based network embeddings with mean commute time. *PloS one*, 14(8):e0221172, 2019.

Jeff Larson Lauren Julia Angwin Kirchner and Surya Mattu. Machine bias: Thereâăźs software used across the country to predict future criminals. and itâăźs biased against blacks.(may 2016), 2016.

Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. Operationalizing individual fairness with pairwise fair representations. *arXiv preprint arXiv:1907.01439*, 2019.

Farzan Masrour, Tyler Wilson, Heng Yan, Pang-Ning Tan, and Abdol Esfahanian. Bursting the filter bubble: Fairness-aware network link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 841–848, 2020.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Alan Mislove, Bimal Viswanath, Krishna P Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 251–260, 2010.

Osonde A Osoba and William Welser IV. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation, 2017.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

Tahleen A Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *IJCAI*, pages 3289–3295, 2019.

Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.

Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1225–1234. ACM, 2016.

Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

Appendix

A Source Codes and Datasets

To access the source codes and datasets used in our paper please refer to https://github.com/ahmadkhajehnejad/CrossWalk.

B Table of Notations

Table 1: Notations used in this paper

Notation	Definition
G	Original Graph
V, E	Nodes and edges of G
V_i	The i^{th} group of G
S	Initial seed set for IC model
Q	Performance of a graph algorithm on the entire graph
Q_i	Performance of a graph algorithm on group $i \in [C]$
π_{uv}	Normalized transition probability between nodes u and v
$\mathcal{W}_u$	A random walk with fixed length d rooted at node u
$m(u)$	Closeness of node u to group boundaries
$\mathcal{N}(u)$	The set of neighbors of node u
l_u	The group to which node u belongs
N_u	The set of u's neighbors within l_u
R_u	The set of u's neighbors within groups other than l_u
N_v^c	The set of u's neighbors that belong to another group $c = l_v \neq l_u$
p	Tuning parameter controlling how much information propagates from one group to the others.
α	Multiplication factor in the reweighting process controlling the probability of a boundary node being connected to other groups.
w_{uv}	Edge weight between nodes u and v in the original graph
w'_{uv}	New edge weight between nodes u and v after reweighting method

C Classical Influence Maximization

In the classical influence maximization problem, the goal is to find the most influential subset of k nodes that can maximize the spread of a piece of information through the network. Formally, the set of k most influential seeds can be found by solving the following maximization problem:

$$S^* \in \arg\max_{S \subseteq V} f(S), \tag{6}$$

where $f : 2^V \to \mathbb{R}$ quantifies the expected number of individuals infected at the end of the diffusion process. Using IC to model information diffusion, the utility function f in Problem 6 is a non-negative and monotone submodular set function Kempe *et al.* [2003]. The submodularity is an intuitive notion of diminishing returns, stating that for any given sets $A \subseteq A' \subseteq V$ and any node $a \in V \setminus A'$, it holds that:

$$f(A \cup \{a\}) - f(A) \geq f(A' \cup \{a\}) - f(A').$$

Although problem (6) is NP-hard in general [Wolsey, 1982], for maximizing a submodular function the following greedy algorithm provides a logarithmic approximation guarantee. The greedy algorithm starts from an empty set, adds a new node to the set which provides the maximal marginal gain in terms of utility, and stops when k nodes are selected.

Edge-reweighting Enhances Fairness Our reweighting method increases the weights of in-links to the nodes that are closer to the groups' periphery. Therefore, such nodes have a higher chance of being activated in the IC model. Furthermore, our reweighting increases the weight of out-links from the nodes that are on the boundary of their corresponding groups. As a result, the boundary nodes activate nodes from other groups with a higher probability. This encourages the greedy seed selection algorithm to include nodes that are closer to group boundaries and can spread the information more effectively to other groups. This results in reducing the disparity of influenced individuals from various groups.

Figure 7 compares the fraction of influenced individuals in different groups, when seeds are greedily selected from a network reweighted by our proposed method. The greedy algorithm estimates the expected influence of each set of seeds by simulating the diffusion process based on IC model. This diffusion process can be simulated as a set of stochastic traversals of the edges on the graph. We scale the transmission probability of every edge $(v, u) \in E$ by the weights w'_{vu} proposed by our reweighting strategy. Formally, for an edge with transmission probability τ in the original graph, the scaled transmission probability can be calculated as $\tau'_{vu} = \tau_{vu} \times w'_{vu}/\mathcal{M}_v$, where $\mathcal{M}_v = \max_{u \in \mathcal{N}(v)} w'_{vu}$ is a normalization factor. It is worth mentioning that although the greedy algorithm is run on the reweighted graph, the final evaluation of each seed selection algorithm is performed on the original graph.

D Effect of Bias parameters α and p

To study the effect of bias parameters α and p, we use different two-grouped graphs with $P^A_{intra} = P^B_{intra} = P_{intra} = 0.025$ and different values of P^{AB}_{inter}. Figure 8 shows that our proposed reweighting strategy enhances fairness of influence maximization on networks with loosely inter-connected groups (networks with smaller values of P^{AB}_{inter}) to a larger extent. When the value of P^{AB}_{inter} is close to P^A_{intra} and P^B_{inter}, the network becomes a single connected component, in which each node connects to all the other nodes with the same probability. In this settings, the greedy selection simply infects all the nodes with the same probability, and consequently, results in a fair influence.

As the number of inter-group connections grows, we need to use larger values for α. For instance, consider a network in which a β-fraction of neighbors of each boundary node belongs to other groups. Here, using larger values for β than α increases the probability of visiting nodes from other groups in random walks. The first row of Figure 8 ($\alpha = 1$) shows that growing p does not compensate for using small values of α for graphs with $P^{AB}_{inter} = 0.01$ and $P^{AB}_{inter} = 0.015$. Therefore we suggest selecting larger values of α for graphs with larger number of inter-group connections.

Figure 8 confirms that using sufficiently large values for p (larger than 2 in this experiment) can considerably improve the resulting fairness. Note that p controls the degree of bias of the stochastic edge traversal procedure towards visiting boundary nodes.

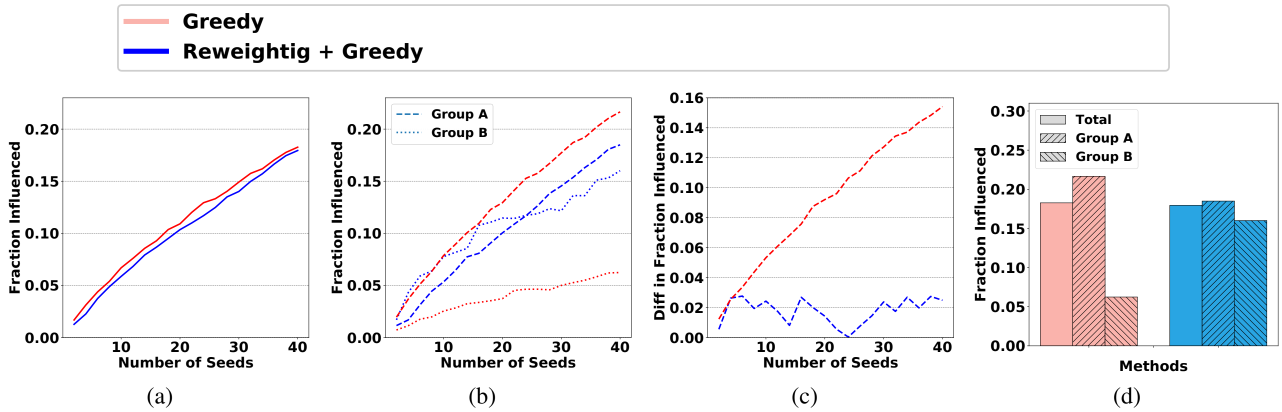


Figure 7: Effect of our proposed reweighting (with parameters $\alpha = 0.4$ and $p = 1$) on the greedy algorithm for classical influence maximization based on IC model, over the Rice-Facebook dataset. Our proposed reweighting method improves the fairness of the greedy seed selection algorithm.

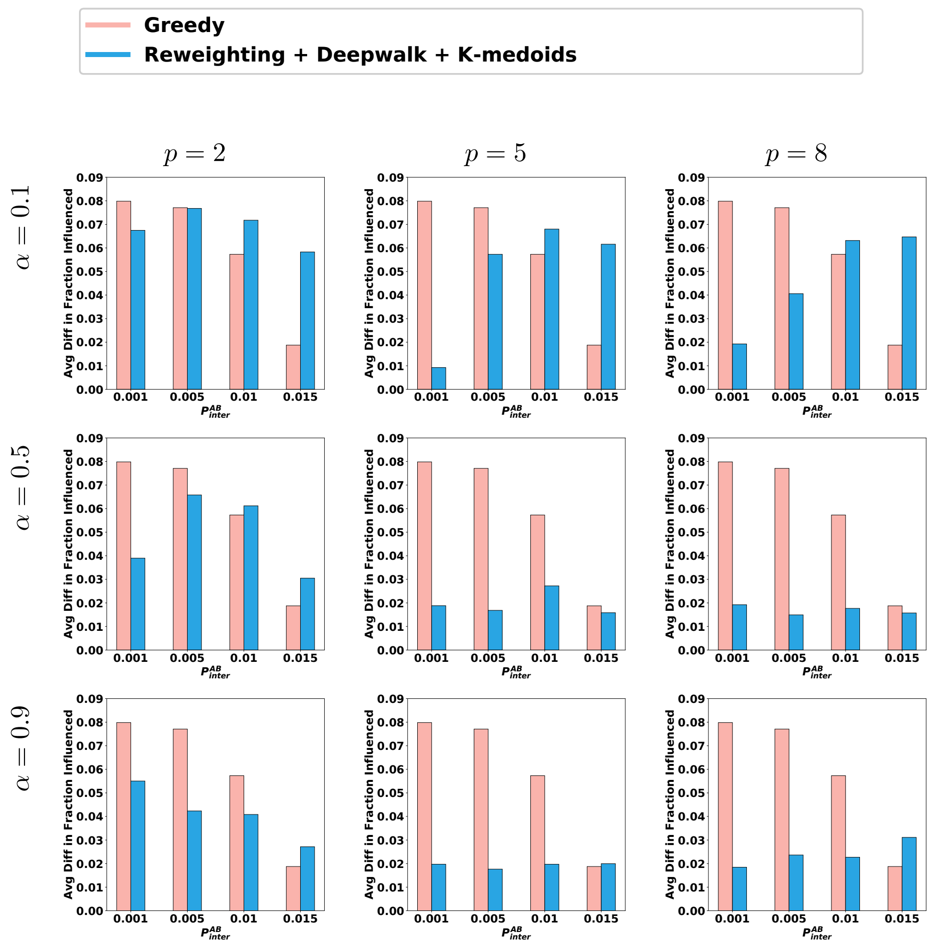


Figure 8: The effect of the parameters α and p on the fairness of the influence, when seeding k-medoids of the DeepWalk representations (blue) of synthetic graphs with $P^A_{intra} = P^B_{intra} = 0.025$ and different values of P^{AB}_{inter}. The results of the greedy algorithm (red) are also illustrated to evaluate the fairness promotion in our method. The y axis is the averaged difference in the fraction of influenced individuals in the two groups for different values of k (number of seeds) from 1 to 40.