Exploring Fairness in Heterogenous Graph Embeddings

Course project for CSE 6240: Web Search and Text Mining, Spring 2021

Sanjana Garg
sgarg96@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Rohit Mujumdar
rohitmujumdar@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

Rohit Gajawada
rohitgajawada@gatech.edu
Georgia Institute of Technology
Atlanta, Georgia, USA

KEYWORDS

Graph Embeddings, Bipartite Networks, Fairness, Feature Learning

1 ABSTRACT

With the increased in usage of artificially intelligent systems around us, it becomes important that these systems do not reinforce existing social biases. Specifically, when using graph embeddings for solving real world tasks based on networks, it is paramount that these embeddings do not carry inherent social and cultural biases. While there have been methods to generate fair embeddings for homogeneous graphs using the notion of unbiased random walks, there has not been any existing work where this notion is used to generate embeddings for bipartite graphs. If we develop information rich node embeddings for networks that are robust to bias, they can be stored and used for multiple downstream tasks. Towards this end, we explore the idea of unbiased random walks to improve the fairness in the node embeddings, using the MovieLens 100k dataset [6]. We first construct graph embeddings for this data using node2vec [3] and metapath2vec [2] algorithms, use them for the downstream task of movie recommendation. After demonstrating bias in the baselines, we introduce fairness mitigation method based on Fairwalk [7] and design new walk strategies by to demonstrate lesser biased results. We also impelement GraphSAGE algorithm and incoporate fairness into it. Out of all the strategies implemented, using the Fairwalk strategy with metapath2vec gives the best results, and show the lowest bias. GrpahSAGE with fairness also shows a lower bias than its vanilla counterpart. These results pave the way for further exploring fairness in heterogeneous graph embeddings.

2 INTRODUCTION

Our project explores the idea of unbiased random walks to improve the fairness in the node embeddings in heterogeneous graphs. We primarily focus on bipartite graphs and use the generated embeddings for the downstream task of recommendation to evaluate the fairness. Towards this, we use MovieLens 100k dataset [6] which consists of 100,000 edges obtained by connections between 943 users and 1682 movies. The dataset also contains demographic information for users and several attributes for each movie, which will form the ground to design and evaluate the biases in our walk strategies and the resulting recommendation.

Our objectives are to explore if existing graph embedding approaches can have their training strategy modified so that they are fair compared to vanilla counterparts. We assess our results by measuring performance on the downstream task of recommendation in terms of fairness metrics like statistical parity and also precision and recall.

We implement the node2vec [3] and metapath2vec [2] baselines, which do not incorporate any bias mitigation measures. We then try 6 different strategies, by building over the baselines and using the Fairwalk [7] walk strategy. In each of the strategies we also experiment with selecting the next node in the walk by sampling from the probability distribution generated from ratings as weighted edges. We also explore the inductive training strategy of GraphSAGE[4] to incorporate fairness.

We calculate precision and recall at various values of k for both of our baselines, shown in Table 2. We note that our metapath2vec + fairwalk approaches with random sampling is able to achieve similar precision and recall with vanilla node2vec and vanilla metapath2vec.

In terms of overall statistical parity over the entire graph i.e all genres which we denote as bias of the algorithm. metapath2vec with fairwalk and random sampling achieves the best bias (**0.17**) in its category beating node2vec (0.29) and metapath2vec with random sampling (0.19). GraphSAGE with fairness is also able to get an improved bias (**0.17**) over its non fair counterpart (0.21)

Impact. We see that introducing bias mitigation techniques, such as Fairwalk, to generate graph embeddings does help reduce bias. This sets the ball rolling for (atleast) a discussion around fairness in graph embeddings. Discussion around fairness in AI, especially language models, and an attempt to reduce bias began much later than the inception of the said language models or their foundations, causing significant damage. Since graph and node embeddings are relatively newer, research of equivalent techniques in graphs, would be more impactful in terms of preventing inherent socio-cultural biases in downstream tasks.

3 LITERATURE SURVEY AND BASELINES

One of the first attempts to study potential bias issues inherent within graph embeddings in a transductive way was when Rahman et. al. proposed Fairwalk [7], a fairness-aware embedding method that extended node2vec [3]. While node2vec uses a random walk over a graph to generate walk traces and then extracts features based on the learned traces, Fairwalk first partitions neighbors into groups based on their sensitive attribute values and samples nodes for random walk from each group with the same probability, thereby enforcing equality of representation. However, Fairwalk is designed for homogeneous networks and would invariably lose out on the semantics of bipartite (heterogeneous) graphs. We aim to improve upon this by employing metapath2vec[2] which extends node2vec [3] by conditioning the random walk sampling based on the node types. This demonstrate improvements over using naive homogeneous approaches for heterogeneous networks. We aim to introduce fairness in the embeddings generated by metapath2vec[2].

GraphSAGE [4] is an inductive approach for generating learnt graph embeddings and it can be done in an unsupervised, semi-supervised or supervised way. The benefit of this method for generating embeddings is that it can scale well to unseen networks and nodes as it does not require all of them to exist during training. Popular works like PinSage [8] have adopted this for large scale recommender systems for bipartite graphs. However, there is no study about inherent bias present in the embeddings.

Bose et al.[1] introduced another inductive approach for learning unbiased embeddings by using filtering via adversarial training with respect to the sensitive attributes. It does not use the notion of unbiased random walks for unsupervised feature learning and the embeddings generated can only be used for that particular downstream task.

4 DATASET DESCRIPTION

4.1 Dataset Preparation

Source. We use the MovieLens dataset [5], which is rating data set collected and made available on the Grouplens datasets web site by GroupLens Research. The data was collected from the website during the seven-month period from September 19, 1997 through April 22, 1998. We use MovieLens 100K Dataset version of the dataset. The dataset consists of 100,000 ratings on a scale of 1 (lowest) to 5 (highest) from 943 users on 1682 movies. Each user has rated at least 20 movies. The dataset provides demographic information like age, gender, occupation, zip and features like movie title, release date, video release date, IMDb URL and genre for each movie. While our data does not have explicit ground truth labels, the presence of an edge (rating by a user to a movie) is considered positive for the downstream task of movie recommendation. We generate negative samples based on what is not present in the data. (no recommendation.)

Data preprocessing. We construct an *weighted, undirected, bipartite* graph of movies and users as nodes and the edges between representing a user who rated a movie. To distinguish between the IDs of the user and movie we append the strings 'user' and 'movie' reflectively as prefixes to the numeric IDs. We add the demographic features of users viz. age, gender, occupation, zip as *node attributes* and the ratings given by a user to a movie as the *edge weights* between the two types of nodes.

Significance To measure the fairness of the embeddings we need a graph dataset that has sensitive attributes so that the performance of the unbiased embeddings can be compared against that biased embeddings in a downstream task. Since our dataset has sensitive attributes of gender and occupation we use it for our project.

4.2 Raw Data Statistics

The constructed graph has the properties described in Table1:

Property	Value
Number of nodes	2625
Number of edges	100000
Average degree	76.1905
Radius	3
Diameter	5
Density	0.02903

Table 1: Dataset (Bipartite Graph) properties

4.3 Data Analysis

After exploring our data, we had some interesting findings, which are articulated below:

- To check the distinct network connectivity difference between male and female users, we measured how many users, females are connected to, via user - movie - user (2 hop) links (shown in Figure 1) and did the same for males. The proportion of users connected to higher number of unique users is more in males compared to females (by 4% normalized percentage).
- We further assesse the occupations of users which are 2 hops away for both males and female and find that males are connected more to engineers, programmers, students and technicians by roughly 0.2% normalized percentage and females are connected more to librarians, educator and other category by roughly 0.2% normalized percentage. Other occupations seem to be almost same for both male and female.
- We also assessed if the movie genre affects the proportion of a gender being connected to movies in that genre (shown in Figure 2). We noticed some genres like Action have over 2% more male users and genres like Romance and Drama have over 2% more female users.

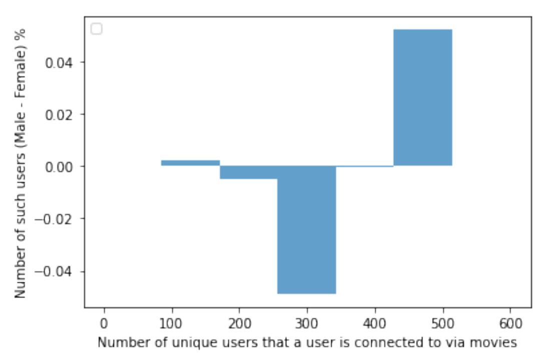

Figure 1: Data Insight 1

Note that in the above observations, we consider percentages by normalizing by the total count of that gender for a meaningful comparison since there are more male users than female users. For

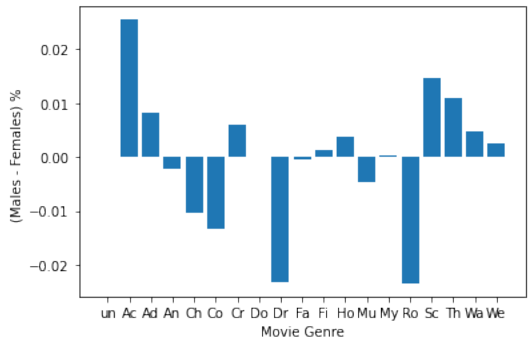

Figure 2: Data Insight 3

calculating percentage difference between male and female, we normalize the male and female histograms individually and then subtract them.

These findings give us an insight into what (potential) biases the graph can learn with respect to the sensitive attributes. For example, the third insight could mean that a biased recommender system could learn to recommend actions films to more men than women.

5 EXPERIMENTAL SETTING AND BASELINES

5.1 Classification Task.

Movie Recommendation. For both baselines, we first generate node embeddings of the MovieLens graph. We use these embeddings for the downstream task of movie-user recommendation. We adopt a supervised approach for this, where our classifier takes as input two embedding vectors, one for user and one for item and *predict the presence or absence of an edge between them.* The presence of an edge implies recommendation and vice versa. We use the positive class probability as the recommendation score. We train a Logistic Regression model where the probability score of an edge being present is used for ranking recommendations. An item v is recommended to a user u, if the recommendation score for the pair $(u, v) \notin E$ is within the top $k\%$ of all the scores received by all candidate pairs. The resulting set of recommendations given to a user u is denoted by $\rho(u)$.

5.2 Evaluation Metrics.

5.2.1 *Evaluating of recommendation system performance.*

To evaluate the performance of each recommendation system, we use the metrics in terms of precision@k and recall@k. In the context of recommendation systems we are most likely interested in recommending top-N movies to the user. In the notion of precision and recall at k, k is a user definable integer that is set by the user to match the top-N recommendations objective.

5.2.2 *Evaluating Fairness.*

We have a bipartite graph $\mathbf{G} = (V, E)$ where $V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$, and the set of edges $E \subseteq \{(u, v) : u \in V_1, v \in V_2\}$. The function $\zeta : V_1 \rightarrow \mathcal{Z}^{\mathcal{S}}$ maps users to their attribute values. For example, for a male u, $\zeta^{g}(u)$ ="male". A sensitive attribute is denoted by $\mathcal{S}$. User-item pairs $(u, v) \in V_1 \times V_2$ are partitioned into groups $G_{ij}^{\mathcal{S}}$ based on the attribute values of u, where $G_{ij}^{\mathcal{S}} = \{(u, v) : \zeta(u) = i, u \in V_1, v \in V_2\}$. The set of all groups based on a sensitive attribute $\mathcal{S}$ is denoted by $\mathcal{G}^{\mathcal{S}}$ where in this case, S = (male, female). For example $\mathcal{S} = g$ for gender. $\mathcal{Z}^{\mathcal{S}}$ denotes the set of all possible values of $\mathcal{S}$. A set of items recommended to user u as $\rho : V_1 \longrightarrow 2^{V_2}$. For evaluating fairness, we use the metric, Statistical Parity. This measure requires the acceptance (recommendation) rates from two groups to be equal. Let $P(G_{ij}^{\mathcal{S}})$ denote the acceptance rate for group $G_{ij}^{\mathcal{S}}$. Here in our case, i would be either the male or female group and j would be the genre of the movies predicted. Then the statistical difference for a particular genre j would be :

$$P(G_{ij}^{\mathcal{S}}) = \frac{|\{(u, v) : v \in \rho(u) \wedge (u, v) \in G_{ij}^{\mathcal{S}}\}|}{|G_{ij}^{\mathcal{S}}|}$$

The bias or statistical parity between male and female is basically the difference between their acceptance rates. Extending this to multiple groups (an overall statistical parity of the whole gender), the overall statistical parity or overall bias for that algorithm can be calculated as the standard deviation of the acceptance rate of each group. $bias(\mathcal{G}^{\mathcal{S}}) = stddev(\{P(G_{ij}^{\mathcal{S}})\} : G_{ij}^{\mathcal{S}} \in \mathcal{G}^{\mathcal{S}})$

Justification of metrics. Precision and Recall help us identify the relevance of the recommendations done for users. Since our proposed methods are supposed to be less biased than node2vec, precision and recall for the former would probably be lower than that for the latter, indicating that we deviate more from the original biased recommendations compared to thereby *not* amplifying the initial bias in the network. Statistical parity measures how independent the algorithm is of a particular protected attribute and would worsen in the scenario of higher bias.

5.2.3 *Experiment Parameters.*

Train-test parameters. For training the embeddings, we split our dataset into train and test set. Out of the 100,000 user-movie pairs, 80% pairs were used for training and 20% were kept for testing. For movie recommendation, we used link prediction task. Since we only had positive pairs from the dataset, ie the movies which were rated by the users. We sampled equal number of negative pairs from pairs which were not present in the dataset. These positive and negative pairs combined were used for training the Logistic Regression classifier.
Model Hyperparameters. For the logistic regression model we used the following hyperparameters : (1) Penalty = L2 Norm, (2) Inverse of regularization strength = 1.0. The rest were default parameters of *sklearn.linear_model.LogisticRegression.*
System Details. We used a 16GB RAM CPU system for our data processing and experiments. The CPU details are : Intel(R) Core i7 CPU, with 1.890 GHz. For GraphSAGE experiments, we used a NVIDIA 1060 GPU.

5.3 Baselines

We implement the two baselines we plan to use for our downstream recommendation task.

- node2vec [3]: node2vec uses a random walk over a graph to generate walk traces and then extracts features based on the learned traces. For node2vec we used the following open-source repository : .

- metapath2vec [2] : This approach builds the random walk sampling based on the node types. Since we have two node types - movie (M) and user (U), our random walk strategy would be either $MUMU$... or $UMUM$....We implement metapath2vec using 2 strategies. In both the strategies, we ensure that only an M is chosen after a U and the other way round. However, in the first strategy, the next node in the walk is sampled randomly. Whereas, in the second strategy, we use the ratings between the nodes as edge weights and use them as the probability distribution for sampling the next node in the random walk.

The results and discussion for the baseline are detailed in Section 6.

5.4 Proposed Methods

5.4.1 New Walk Strategies. We extend the idea used in Fairwalk [7] to our baseline of metapath2vec [2] approach. Assume there are two types of nodes V_1 and V_2 in a bipartite graph G and V_1 has sensitive demographic attributes like gender, race, etc (usually a user) and V_2 has attributes like genre, etc. We can enforce an unbiased step from V_2 to V_1 such that there is an equal likelihood of sampling users from different demographics $v_1 \in N(v_2)$ where $v_1 \in V_1$ in the neighbourhood of the node $v_2 \in V_2$. Let's say a_i are the distinct attributes present in the neighbors of v_i and $f_j(v_i)$ be the jth subgroup of neighbors grouped by a_i.
For V_2 to V_1:

$$p\left(v_1 \mid v_2\right) = \begin{cases} \frac{1}{|f_j(N(v_2))|}\frac{1}{|a_2|}. & (v_1, v_2) \in E,\ v_1 \in f_j(N(v_2)) \\ 0 & \text{otherwise} \end{cases}$$

Thus, the sampling of the walk from user to movie in metapath2vec would aim to be unbiased in terms of gender with respect to genre and similarly the sampling of the walk from movie to user in metapath2vec would aim to be unbiased in terms of genre with respect to gender.

Based on the above, we propose 4 new walk strategies delineated below:

(1) **metapath2vec + Fairwalk + random next node sampling (RS)** : While choosing the next node we follow the $MUMU$... or $UMUM$.. strategy but also partition neighbors into groups based on their gender values and sample nodes for random walk from each group with the same probability, thereby enforcing equality of representation. Within this strategy, the next node in the walk is sampled randomly.

(2) **metapath2vec + Fairwalk + weighted next node sampling (WS)** : Similar to the above strategy, except, we use the ratings between the nodes as edge weights and use them as the probability distribution for sampling the next node in the walk.

(3) **metapath2vec + Fairwalk + node2vec next node sampling (NS)** : While choosing the next node we follow the $MUMU$... or $UMUM$.. strategy but also partition neighbors into groups based on their gender values and sample nodes for random walk from each group with the same probability, thereby enforcing equality of representation. Within this strategy, if we are at a node c after having visited node s, the next node d in the walk is sampled using the node2vec sampling strategy, using the return and in-out paramaters.

(4) **metapath2vec + Fairwalk + weighted next node sampling (WS) + node2vec next node sampling (NS)** : Similar to the above strategy, except, we use the ratings between the nodes as edge weights and use them as the probability distribution for sampling the next node in the walk, along with the node2vec walk strategy. We multiply the rating weights with the probability distribution obtained from node2vec for each node and apply the softmax function to it to get the final combined probability distribution, from which node d, after having visited node s and c would be sampled from.

Justification of Approach. Our first baseline, node2vec, does not address the potential biases that may be learnt by the embeddings, nor does it take into account the heterogenity of the graph. While the second baseline metapath2vec does use a walk approach that alternates between movie and user nodes, potential biases may still be introduced because of a biased choice of next node in the walk. While Fairwalk tries to rectify this, it is still designed for homogeneous networks, and captures only structural correlations between different "types" of nodes, making the embeddings invariably lose out on the semantics of bipartite graphs. For example, in our graph, during the random walk, if the walker is at node of type *user*, then it is biased towards nodes of type *movie* for the next step. This makes metapath2vec more suitable for heterogenous networks than Fairwalk. Adding the Fairwalk sampling strategy in metapath2vec, we can ensure that the semantic relationships between different types of nodes is captured, *and* we also possibly get less biased embeddings by enforcing equality of representation. Hence, we can expect our method to perform better than baselines. We further use the node2vec walk strategy with the return and in-out parameters to exploit the freedom of defining neighbourhoods for nodes and controlling the BFS-DFS exploration in the graph.

Walk Strategy Parameters. We chose to use the same hyperparameters for the baselines and the proposed strategies while generating embeddings, so that the comparison would be fair. The recommendation system result metrics and the fairness metrics for each of these are shown in Table 2

5.4.2 ***GraphSage with Fairness***. After our approach for metapath2vec with fairness, we wanted to experiment with a similar unbiased sampling step via the AGGREGATE step in GraphSAGE. Our aim is that such fairness based approaches can be added on to inductive approaches like GraphSAGE. The main idea of using GraphSAGE in our scenario, is that for every movie node embedding, a certain number of user neighbours of the movie node are sampled at random and aggregated.

However, some movie nodes might have significantly more male users as neighbours and some other movie nodes might have significantly more female users as neighbours. In order to possibly introduce fairness in this sampling step, when aggregating the user neighbours of a movie we make sure we take the same number of male user nodes and female user nodes in the GraphSAGE with fair AGG approach.

Strategy	k=50		k=100		k=500		k=1000			
(P = Precision, R = Recall, A = Accuracy, B = Bias)	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**A**	**B**
node2vec	0.05	0.16	0.04	0.29	0.02	0.79	0.01	0.93	0.67	0.29
metapath2vec + RS	0.03	0.1	0.03	0.2	0.02	0.67	0.01	0.93	0.67	0.19
metapath2vec + WS	0.03	0.09	0.03	0.2	0.02	0.61	0.01	0.89	0.67	0.25
metapath2vec + fairwalk + RS	0.04	0.12	0.03	0.22	0.02	0.65	0.01	0.92	0.67	**0.17**
metapath2vec + fairwalk + WS	0.03	0.11	0.03	0.19	0.02	0.63	0.01	0.91	0.67	0.18
metapath2vec + fairwalk + NS	0.03	0.1	0.03	0.23	0.02	0.67	0.01	0.93	0.67	**0.17**
metapath2vec + fairwalk + NS + WS	0.04	0.12	0.03	0.22	0.02	0.67	0.01	0.92	0.67	0.23

Table 2: Baseline and Proposed Walk Strategy Results

Parameter	Value
Embedding Dimension size	64
Number of Walks	20
Walk Length	30
node2vec return parameter (p)	0.50
node2vec in-out parameter (p)	0.25

Table 3: Experiment strategies hyperparameters

Method	Accuracy	F1-Score	Bias
GraphSAGE	0.83	0.91	0.21
GraphSAGE with fair AGG	0.83	0.90	**0.17**

Table 4: GraphSAGE Results.

Algorithm 1: Heterogenous GraphSAGE with fair aggregation

Input : Graph G(V_u, V_m, E), depth $K = 2$,
weight matrices W^k, non-linearity σ,
differentiable aggregator function AGG,
neighbourhood function for movies
$N_u : v_m -> 2^{V_u}$ and
neighbourhood function for users $N_m : v_u -> 2^{V_m}$,
NNS (number of neighbour samples for AGG)

Output : node embeddings for every node (movie and user)

for $k \leftarrow 1$ ***to*** K **do**
  for $v_m \in V_m$ **do**
    if *Fairness Sampling == True* **then**
      $h^k_{N_u(v_m)} \leftarrow AGG_k(\mathrm{h}^{k-1}_{u-male} \leftarrow N_u(v_m)$ where gender(u) is Male + $h^{k-1}_{u-female} \leftarrow N_u(v_m)$ where gender(u) is Female)
      where $count(\mathrm{h}^{k-1}_{u-male}) == count(\mathrm{h}^{k-1}_{u-female})$
    end
    if *Fairness Sampling == False* **then**
      $h^k_{N_u(v_m)} \leftarrow AGG_k(\mathrm{h}^{k-1}_u \leftarrow N_u(v_m))$
    end
    $h^k_{v_m} \leftarrow \sigma W^k.CONCAT(\mathrm{h}^{k-1}_{v_m}, h^k_{N_m(v_m)})$
  end
  $h^k_{v_m} \leftarrow h^k_{v_m} ||h^k_{v_m}||_2 \ \forall v_m \in V_m)$
end
for $k \leftarrow 1$ ***to*** K **do**
  for $v_u \in V_u$ **do**
    $h^k_{N_m(v_u)} \leftarrow AGG_k(\mathrm{h}^{k-1}_m \leftarrow N_m(v_u))$
    $h^k_{v_u} \leftarrow \sigma W^k.CONCAT(\mathrm{h}^{k-1}_{v_u}, h^k_{N_m(v_u)})$
  end
  $h^k_{v_u} \leftarrow h^k_{v_u} \ ||h^k_{v_u}||_2 \ \forall v_u \in V_u)$
end

Hyperparameters. We use mean aggregation with K = 2. The number of neighbours sampled is equal to 10 and in the fairness AGG approach, we sample maximum 5 for male and maximum 5 for female. For the input embedding in the network encoder, we use an embedding dimension of 100.

Training and experiment setup. While the embeddings of our node2vec and metapath2vec based approaches are trained in an unsupervised fashion, we obtain our GraphSAGE embeddings by training the network in a supervised fashion. This is because the unsupervised loss proposed in Hamilton et al. [4] does not make sense intuitively for heterogenous graph because the embeddings of nearby movie and user nodes need not be similar. We train the GraphSAGE network to predict 1 if the rating between a user embedding and a movie embedding is >= 3 and to predict 0 otherwise. This is also why we do not compare GraphSAGE approaches directly with our node2vec and metapath2vec approaches.

6 EXPERIMENTS, RESULTS AND DISCUSSION.

6.0.1 Recommendation system metrics. We calculate precision and recall at various values of k for both of our baselines, shown in Table 2. We note that our metapath2vec + fairwalk approaches with random sampling is able to achieve similar precision and recall with vanilla node2vec and vanilla metapath2vec.

6.0.2 Fairness Metrics. [7] In terms of overall statistical parity over the entire graph i.e all genres which we denote as bias of the algorithm. metapath2vec with fairwalk and random sampling achieves the best bias (**0.17**) in its category beating node2vec (0.29) and metapath2vec with random sampling (0.19). GraphSAGE with fair AGG is also able to get an improved bias (**0.17**) over its non fair counterpart (0.21)

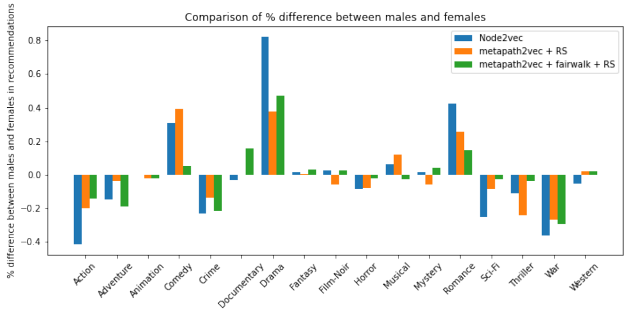

Figure 3: Random sampling - metapath2vec: Difference of % of recommendations from each genre between Males and Females

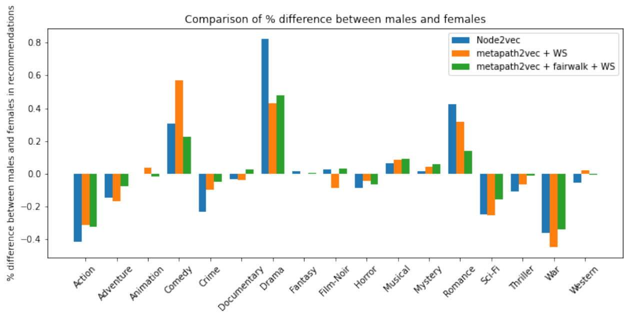

Figure 4: Random sampling - metapath2vec: Difference of % of recommendations from each genre between Males and Females

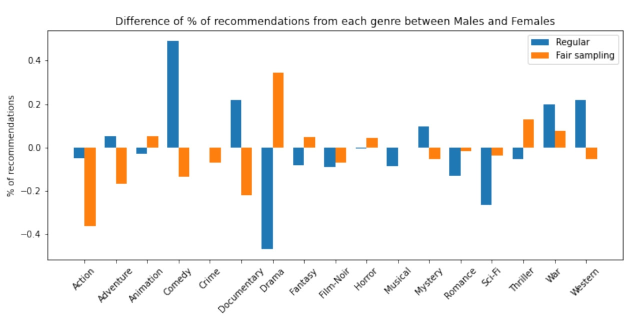

Figure 5: GraphSAGE: Difference of % of recommendations from each genre between Males and Females

7 CONCLUSION

Our project aimed at exploring techniques to introduce fairness in graph embeddings. We first established the bias that get introduced in conventional algorithms such as node2vec [3] and metapath2vec [2]. We explored hybrid walk strategies that build upon either of metapath2vec and node2vec, or both. We demonstrated, using the downstream task of movie recommendation, that adopting bias mitigation strategies reduces bias in the embeddings. We further implement GraphSAGE [4] and demonstrate that by adding fairness sampling in it, the performance of the rating model does not reduce while improving the bias of the model.

One shortcoming of the project is that we only assess fairness with respect to one sensitive attribute (gender). We also were not able to try unsupervised deep learning based approaches for generating embeddings. Besides, the results of our analyses are based only on dataset with a bipartite graph structure, which makes it to generalise our findings to all heterogeneous graphs.

In the future, more deep learning based unsupervised approaches can be explored using fairness constraints. These approaches should be tested on even larger datasets (for example entire Pinterest, Instagram, Twitter, etc) to assess their fairness capability at scale. This would have a much larger impact by ensuring certain groups of people are not treated unfair. Other extensions of the project include experimenting with different combinations or groups of sensitive attributes. For example, assessing how our proposed strategies work with age + gender combinations as compared to occupation + gender. It would be worth performing correlation analyses amongst such combinations to see what insights can be drawn.

8 CONTRIBUTION

All team members have contributed a similar amount of effort.

REFERENCES

[1] A. Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *ICML*.

[2] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *KDD '17*. ACM, 135–144.

[3] Aditya Grover and J. Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016).

[4] William L. Hamilton, Zhitao Ying, and J. Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*.

[5] F. M. Harper and J. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5 (2015), 19:1–19:19.

[6] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4, Article 19 (Dec. 2015), 19 pages. https://doi.org/10.1145/2827872

[7] Tahleen A. Rahman, Bartlomiej Surma, M. Backes, and Y. Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *IJCAI*.

[8] Rex Ying, Ruining He, K. Chen, Pong Eksombatchai, William L. Hamilton, and J. Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining* (2018).