Fairwalk: Towards Fair Graph Embedding

Tahleen Rahman, Bartlomiej Surma, Michael Backes and Yang Zhang
@ CISPA Helmholtz Center for Information Security

D6: Online Social Networks and Media; paper presentation

Papazis Stergios (483)
Zisopoulos Georgios (505)

Wednesday, 17 January 2024

Department of Computer Science & Engineering
School of Engineering
University of Ioannina

Overview

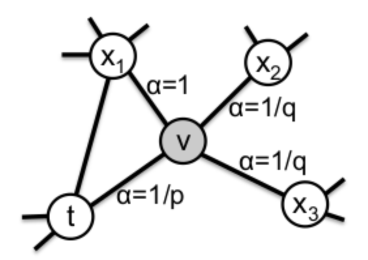
- Assume a graph task that requires ML
  (e.g. building a link/friend recommendation system)
- ML algorithms can only be used with vector representation
  $\Rightarrow$ graph embedding are needed
- Node2vec generates embeddings, *but*:
  - fairness through unawareness
  - ignores minority structure
  - creates echo chambers, bubble effect
- How Node2vec is unfair and how to modify to fix it
  (Fairwalk)

Nodes2vec

- state of the art algorithm
- captures structure of network (node similarity, hubs, neighbourhood structure)
- extendable to edge embedding

Walk generation

- second order random walks
- p: search width
- q: search depth
- parameters: d, walk_num, walk_len

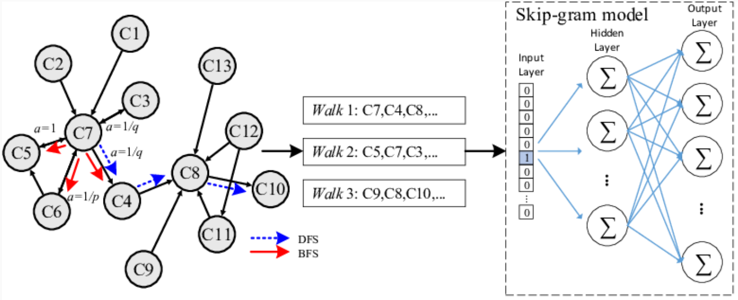


Nodes2vec

Train a ML model that creates the vector representations:

1. sample node neighbourhoods through random walks;
2. maximise likelihood that a node neighbourhood appear given the feature vector of the node

$$\max_{f} \sum_{u \in V} \log \mathrm{P}\left(N_S(u) \mid f(u)\right)$$



Fairness

- How to measure fairness in this setting?

Fairness

- How to measure fairness in this setting?
- **Sensitive attribute**: S (e.g. g: gender, r: race, …)

Fairness

- How to measure fairness in this setting?
- **Sensitive attribute**: S (e.g. g: gender, r: race, …)
- **Group** G^S_{ij} : pair of users with attribute values i, j

Fairness

- How to measure fairness in this setting?
- **Sensitive attribute**: S (e.g. g: gender, r: race, …)
- **Group** G^S_{ij} : pair of users with attribute values i, j
- **Number of recommendations for** G^S_{ij} :

$$N_{rec}(G^S_{ij}) := \left| \left\{ (u, v) \in G^S_{ij} : \; v \in \rho(u) \right\} \right|$$

Fairness

- How to measure fairness in this setting?
- **Sensitive attribute**: S (e.g. g: gender, r: race, ...)
- **Group** G_{ij}^{S} : pair of users with attribute values i, j
- **Number of recommendations for** G_{ij}^{S} :

$$N_{rec}(G_{ij}^{S}) := \left|\left\{(u, v) \in G_{ij}^{S} : \; v \in \rho(u)\right\}\right|$$

- **Acceptance rate**:

$$\mathrm{P}(G_{ij}^{S}) = \frac{N_{rec}(G_{ij}^{S})}{\left|G_{ij}^{S}\right|}$$

fraction of j-user suggested to i-users over total number of users with i, j attributes

Fairness measurements: Statistical Parity

- **Statistical Parity** or **Demographic Parity** or **Independence**: acceptance rates of the candidates from all groups should be somewhat equal
- bias wrt G^S_{ab}, G^S_{cd} :

$$\mathrm{bias}^{\mathrm{SI}}(G^S) := \mathrm{P}(G^S_{ab}) - \mathrm{P}(G^S_{cd})$$

- bias for multiple groups :

$$\mathrm{bias}^{\mathrm{SI}}(G^S) := \mathrm{Var}\left(\left\{\mathrm{P}(G^S_{ij}) : \; G^S_{ij} \in G^S\right\}\right)$$

Equality of Representation

- **network level**:

$$\text{bias}^{\text{ERg}}(G^S) := \text{Var}\left(\left\{N_{rec}(G^S_{ij}) : \; G^S_{ij} \in G^S\right\}\right)$$

- **user level**:

$$\text{bias}^{\text{ERu}}(z) := \frac{1}{|Z^S|} - \frac{1}{|U|}\sum_{u \in U} z\text{-share}(u)$$

  - measures recommendation fairness *independent* of the ground truth (existing friendships contain bias)
  - fair fraction minus average z-share over all users
  - positive: underrepresented, negative: overrepresented

z-share: fraction of recommended users with attribute z

$$z\text{-share}(u) := \frac{|\rho_z(u)|}{|\rho(u)|}$$

Node2vec unfairness example

Dataset

- nodes: *Instagram users* in London & LA
- edges: *mutual follows*
- sensitive attributes: *gender*, *race*
  (autogenerated from profile photos)
- is biased

Groups

- $G^g_{00}, G^g_{01}, G^g_{10}, G^g_{11},$
  (0: female, 1: male)
- $G^r_{00}, G^r_{01}, G^r_{10}, G^r_{11}, G^r_{12}, G^r_{21}, G^r_{22}, G^r_{02}, G^r_{20}$
  (0: african, 1: caucasian, 2: asian)

Node2vec unfairness example

	LA	London
No. users	82,607	53,902
No. social links	482,305	165,184
gender 0	62.6%	62.3%
gender 1	37.4%	37.7%
race 0	21.9%	15.9%
race 1	72.2%	80.7%
race 2	5.9%	3.4%

Table 2: Statistics of both datasets.

	Gender groups				Race groups								
$i-j$ for G_{ij}	0-0	0-1	1-0	1-1	0-0	0-1	1-0	1-1	0-2	1-2	2-0	2-1	2-2
LA	37.78	21.00	21.99	19.21	5.97	12.55	12.65	57.92	1.17	3.87	1.16	3.74	0.96
London	38.96	18.19	20.74	22.10	3.55	9.52	9.83	70.31	0.47	2.57	0.56	2.76	0.41

Table 1: Percentage of existing friendships in each group in our original dataset

Node2vec unfairness example

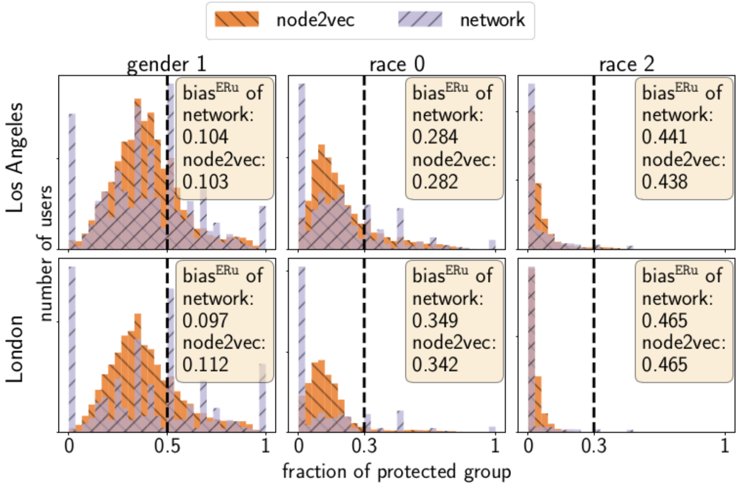
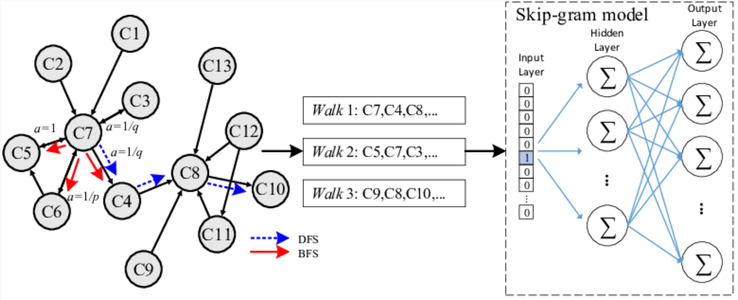


Figure 1: z-share distributions of node2vec and original network. The vertical line shows the *fair fraction* (0.5 and 0.3)

Modify random walk generation



Fair walk generation

Algorithm 1 Fair random walk trace generation

procedure RAND_WALK(U, ω, `walk_num`, `walk_len`)
    traces $\leftarrow$ empty_list
    for all $u \in U$ **do**
        for $i \leftarrow 0,$ `walk_num` **do**
            trace $\leftarrow$ empty_list
            $u_1 \leftarrow u$
            for $j \leftarrow 0,$ `walk_len` **do**
                trace.append(u_1)
                $\mathcal{Z}_u \leftarrow \{z : z \in \mathcal{Z} \wedge \left|\omega_z(u_1)\right| > 0\}$
                $z_1 \xleftarrow{R} \mathcal{Z}_u$
                $v \xleftarrow{R} \omega_{z_1}(u_1)$
                $u_1 \leftarrow v$
            end for
            traces.append(trace)
        end for
    end for
    return traces
end procedure

Random walks in the example

- Node2vec mirrors underlying distribution, not enough info about minorities
- Fair walks create higher network diversity, greater representation of minorities
  $\Rightarrow$ ML gets better understanding of sensitive attributes

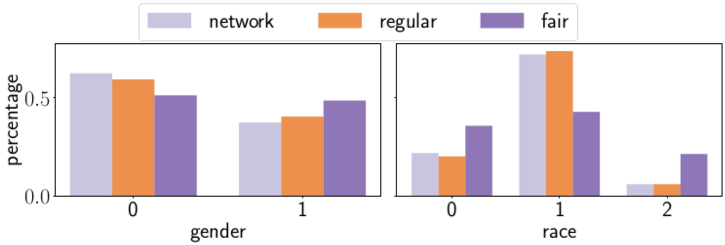


Figure 2: Ratio of each gender and race in the original network and regular and fair random walk traces in Los Angeles dataset

Experiment

Experimental Setup

- 5 iterations
- 80% training set
- 20% test set
- generate node embeddings
- train random forest on embedded pairs of users in order to learn presence or absence of friendships (links)
- recommend k% most similar non-friend users

Dataset reminder

Dataset

- nodes: *Instagram users* in London & LA
- edges: *mutual follows*
- sensitive attributes: *gender*, *race*
  (autogenerated from profile photos)
- is biased

Groups

- G^g_{00}, G^g_{01}, G^g_{10}, G^g_{11},
  (0: female, 1: male)
- G^r_{00}, G^r_{01}, G^r_{10}, G^r_{11}, G^r_{12}, G^r_{21}, G^r_{22}, G^r_{02}, G^r_{20}
  (0: african, 1: caucasian, 2: asian)

Dataset reminder

	LA	London
No. users	82,607	53,902
No. social links	482,305	165,184
gender 0	62.6%	62.3%
gender 1	37.4%	37.7%
race 0	21.9%	15.9%
race 1	72.2%	80.7%
race 2	5.9%	3.4%

Table 2: Statistics of both datasets.

	Gender groups				Race groups								
$i-j$ for G_{ij}	0-0	0-1	1-0	1-1	0-0	0-1	1-0	1-1	0-2	1-2	2-0	2-1	2-2
LA	37.78	21.00	21.99	19.21	5.97	12.55	12.65	57.92	1.17	3.87	1.16	3.74	0.96
London	38.96	18.19	20.74	22.10	3.55	9.52	9.83	70.31	0.47	2.57	0.56	2.76	0.41

Table 1: Percentage of existing friendships in each group in our original dataset

Results: biases

		LA		London	
		gender	race	gender	race
ERg	regular	$1.3e^{10}$	$2.5e^{7}$	$6.5e^{9}$	$2.4e^{7}$
	fair	$0.8e^{10}$	$1.9e^{7}$	$4.8e^{9}$	$1.9e^{7}$
SI	regular	$4.7e^{-9}$	$1.4e^{-12}$	$1.1e^{-8}$	$7.1e^{-11}$
	fair	$1.7e^{-9}$	$0.4e^{-12}$	$0.2e^{-8}$	$2.8e^{-11}$

Table 3: $\text{bias}^{\texttt{SI}}$ and $\text{bias}^{\texttt{ERg}}$ for both cities (lower, the better)

Results: statistical imparity

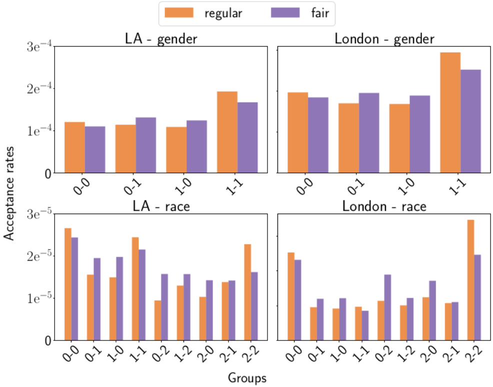


Figure 3: Fraction of recommended users pairs out of all possible pairs in each group. The x-axes marks the Type-2 groups G_{z_i,z_j} with the corresponding $i - j$

Results: equality of representation (network)

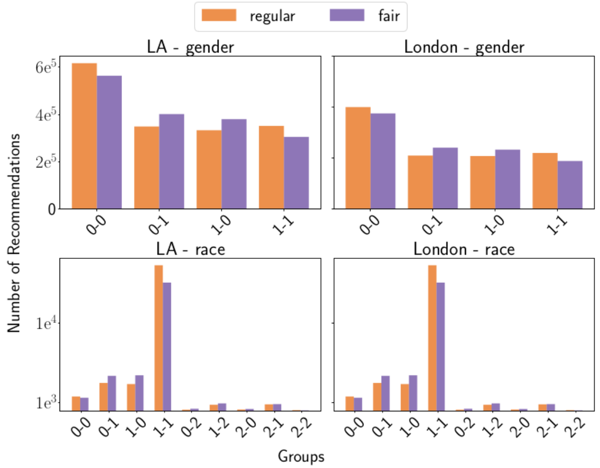


Figure 4: Number of recommended users pairs from each group. The x-axes marks groups G_{ij} with the corresponding $i - j$

Results: equality of representation (user)

		gender		race		
		0	1	0	1	2
LA	network	0.104	0.104	0.117	0.392	0.275
	node2vec	0.103	0.103	0.115	0.387	0.272
	fairwalk	0.068	0.068	0.054	0.288	0.234
London	network	0.097	0.097	0.183	0.481	0.298
	node2vec	0.112	0.112	0.176	0.474	0.298
	fairwalk	0.095	0.095	0.135	0.417	0.282

Table 4: Bias by *Equality of Representation* at user level for both genders and all three races (lower, the better).

Results: equality of representation (user)

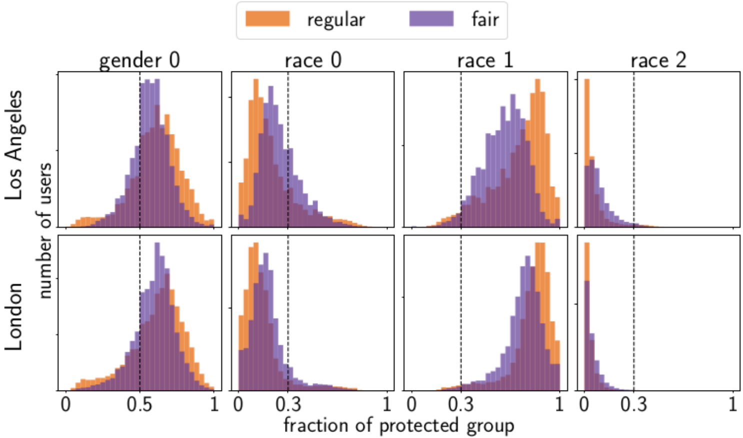


Figure 5: z-share distributions of node2vec and *Fairwalk*. The vertical line shows the *fair fraction*.

Fairwalk utility

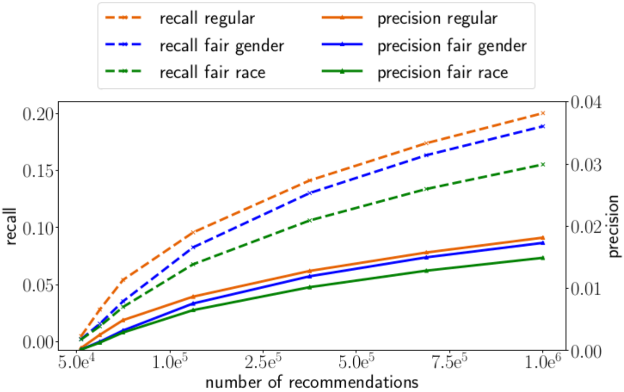


Figure 6: Precision and recall for different number of recommendations.

$$\text{recall} = \frac{TP}{TP + FN}, \qquad \text{precision} = \frac{TP}{TP + FP}$$

Thanks for your attention!

References

[1] 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*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 855–864. DOI: `10.1145/2939672.2939754`.

[2] Tahleen Rahman et al. “Fairwalk: Towards Fair Graph Embedding”. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 3289–3295. DOI: `10.24963/ijcai.2019/456`.