Social Network Analysis

FairMIA: The heuristic based Fair Influence Maximization

Presented by:
Nikhila Dhulipalla, Sasikanth Kotti and Adhun Thalekkara

Presentation Sub-sections

- Introduction
- Discussion about our proposals/contributions
- Experiments
- Results
- Conclusion

Introduction

- People and their relationships make up a social network.
- As a model for spreading information across social networks, Influence Maximization is gaining traction.
- Algorithmic bias and fairness, has surprisingly garnered little attention thus far.
- This issue is important in studying how information spreads in a community of networks, with application areas like marketing, news dissemination, vaccination, and generating online trends.
- Because historical prejudices may be encoded in human networks, algorithms that use them to automate outcomes may capture and recreate such biases

Brief about our proposals

In our current work, we attempted to address the problem of fair influence maximization by defining fairness with heuristics.

1. We proposed a new heuristic based algorithm for fair influence maximization, called FairMIA (fair maximum influence arborescence) which is inspired from MIA algorithm.
2. We demonstrated that our algorithm performs satisfactorily via the 'Price of Fairness', 'Fairness Score' and 'Maximin' metrics.
3. We proposed new 'Fairness Score' metric for determining how equitably a particular attribute has been influenced.
4. We proposed a model to construct synthetic networks to evaluate fairness influence maximization.

Algorithm 4 FairMIA, $FairMIA(G, k, \theta)$

/* initialization */
Set seed set $S = \emptyset$
set $IncInf(v) = 0$ for all nodes $v \in V$
for each node $v \in V$ **do**
  compute $MIIA(v.\theta)$ and $MIOA(v, \theta)$
  set $ap(u, S, MIIA(v, \theta)) = 0, \forall u \in MIIA(v, \theta$ /* since $S = \emptyset$*/
  compute $\alpha(u, v), \forall u \in MIIA(v, \theta$ (Algo.3)
  for each node $u \in MIIA(v, \theta)$ **do**
    $IncInf(u) += \alpha(v, u).(1 - ap(u, S, MIIA(v, \theta)))$
  end for
  compute subgraph G where node attribute value $\in v$
  compute d dijkstra path between v and random node of subgraph G
  compute subgraph H where $node \in G$ and vompute $MIIA(v, H, \theta)$
  for each node $w \in MIIA(v, H, \theta)$ **do**
    $IncInf(w) += \alpha(v, w).(1 - ap(w, S, MIIA(v, H, \theta)))$
  end for
end for
/* main loop */
for $i = 1 \dots k$ **do**
  pick $u = argmax_{v \in V \setminus S}\{IncInf(v)\}$
  /* update incremental influence and fair influence spreads*/
  for $v \in MIOA(u, \theta) \setminus S$ **do**
    /* subtract previous incremental influence and fair influence spreads*/
    for $w \in MIIA(v, \theta) \setminus S$ **do**
      $IncInf(w) -= \alpha(v, w).(1 - ap(w, S, MIIA(v, \theta)))$
    end for
    compute $MIIA(v, H, \theta)$
    for each node $w \in MIIA(v, H, \theta)$ **do**
      $IncInf(w) -= \alpha(v, w).(1 - ap(w, S, MIIA(v, H, \theta)))$
    end for
  end for
  $S = S \cup \{u\}$
  for $w \in MIOA(u, \theta) \setminus S$ **do**
    compute $ap(w, S, MIIA(v, \theta)), \forall w \in MIIA(v, \theta)$ (Algo. 2)
    compute$\alpha(v, w), \forall w \in MIIA(v, \theta)$ (Algo. 3)
    /* add new incremental influence and fair influence spreads */
    for $w \in MIIA(v, \theta) \setminus S$ **do**
      $IncInf(w) += \alpha(v, w).(1 - ap(w, S, MIIA(v, \theta)))$
    end for
    compute $MIIA(v, H, \theta)$
    for each node $w \in MIIA(v, H, \theta)$ **do**
      $IncInf(w) += \alpha(v, w)(1 - ap(w, S, MIIA(v, H, \theta)))$
    end for
  end for
end for
return S

1. We proposed a new heuristic based algorithm for fair influence maximization, called FairMIA (fair maximum influence arborescence) which is inspired from MIA algorithm.

- Algorithm has the below structure:
  - ➢Initialization
  - ➢Main Loop:
    - ➢Update incremental influence and fair influence spreads.
    - ➢Subtract incremental influence and fair influence spreads.
    - ➢Add new incremental influence and fair influence spreads.

    Return seed set S

2. We demonstrated that our algorithm performs satisfactorily via the ‘Price of Fairness’, ‘Fairness Score’ and ‘Maximin’ metrics.

- Maximin Fairness - Maximin Fairness encapsulates the simple objective of enhancing the lives of the poorest population. That is, aim to maximize any group’s minimal influence as a fraction of its population.

$$U^{Maximin}(A) = min_i \frac{I_{G,C_i}(A)}{|C_i|}$$

- Price of Fairness - calculated the Price of Fairness, which would be the ratio of optimal influence to best achievable influence, to determine the cost of guaranteeing a reasonable outcome for the diversified community.

$$PoF^{Maximin} = \frac{I^{OPT}}{I^{Maximin}}$$

3. We proposed new 'Fairness Score' metric for determining how equitably a particular attribute has been influenced.

- Fairness score - The fairness score is a new metric we've developed for determining how equitably a particular attribute has been influenced. This fairness score ranges from 0 to 1, with 1 indicating the highest fairness and 0 indicating no fairness in the influence maximization approach

Algorithm 6 Fairness Score Calculation

```
Input
    n:list of number of nodes in each category
    nInfluenced: list of number of nodes influenced in each category
Output
    returns fairness score
numberOfGroups ← length of n
percentageOfInfluence ← new list formed by nInfluenced divided by n
Find the mean of percentageOfInfluence
create list called deviation with the distance from mean to percentageOfInfluence
sum the deviation list and devide by mean value
return (1 − deviation/numberOfGroups)
```

$$\text{Fairness score} = \left[\sum_{i=0}^{n} \frac{\left| \mu - \frac{numberOfNodes_i}{numberOfNodesInfluenced_i} \right|}{\mu * n} \right]$$

4. We proposed a model to construct synthetic networks to evaluate fairness influence maximization.

Algorithm 5 Barabasi Albert Attribute Model Generator

```
Generate a random star graph with m number of nodes
Initialize the nodes with some sample attributes as given by the user
Initialize a list called repeatedItems
for each node in graph do
    add degree times the node to repeatedItems list
end for
source=length of the graph
while source numberOfNodes do
    initialize new key source to the dictionary graphAttribute
    initialize a blank array called repeatedItems
    for each Attribute in Attributes do
        generate a random choice for the attribute of the node
        extend repeatedItems list with dependencyIndex times the node
    end for
    remove the current handling nodes from repeatedItems to avoid self loops
    Add repeatedItems to the repeatedNodes
    get a random subset of repeatedNodes and form connection
    remove the repeatedItems from the repeatedNodes list
    extend the repeatedNodes with the new subset
    source= source+1
end while
```

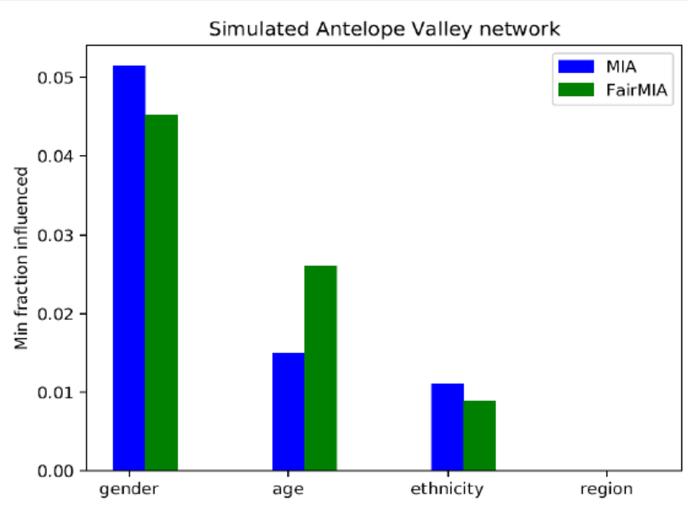
- The Barabasi-Albert Model is a straightforward approach for generating scale-free networks.
- These are extensively utilized as they closely resemble real-life social networks.
- In this paper, we present an addition to the Barabasi-Albert that combines user-defined subgroups or selections such as ’25-30,’40-50,’male,’female,’ with user-defined attribute values such as Age, Gender, Ethnicity.

Experiments

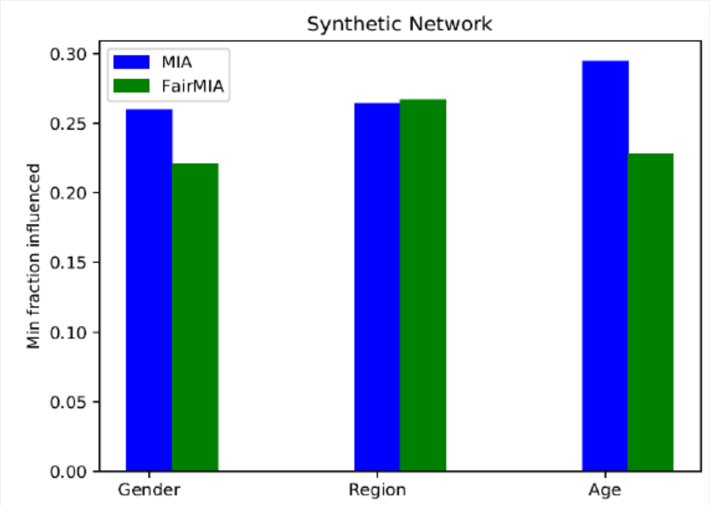
- We utilized two datasets or social networks in the experiment - Antelope Valley network and synthetic dataset.
- IC model is employed. 10000 simulations were executed on the IC model to obtain optimal influence and fair influence. A total of 15 seeds were considered and influence threshold was set to 0.01.
- Our own BA enhanced approach also generates a networkx graph from a synthetic dataset. To create the synthetic dataset, we used the following parameters

Attribute Name	Attribute Type	Choices	Weights	Dependency Index
Gender	Choice	Male, Female	50, 50	3
Region	Choice	India, US	90, 10	2
Age	Choice	20-25, 50-59	60, 40	1

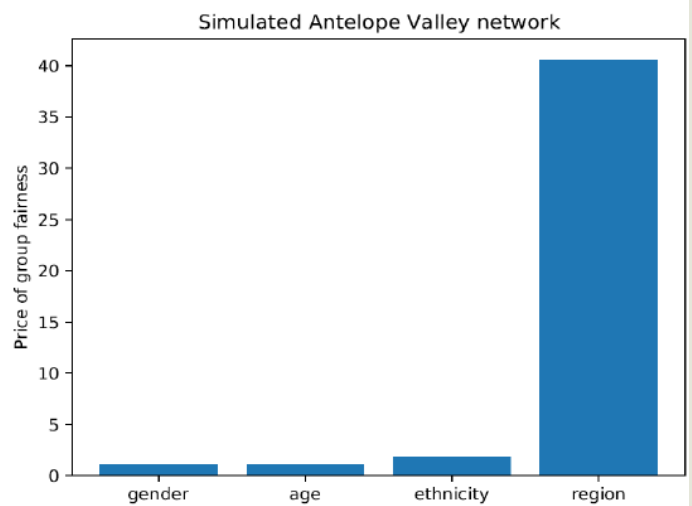
Results



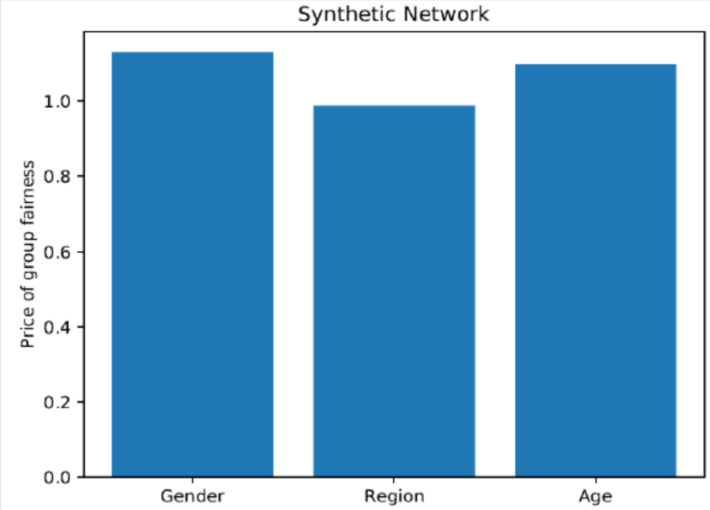
Simulated Antelope Valley network
MIA
FairMIA
Min fraction influenced
0.00
0.01
0.02
0.03
0.04
0.05
gender
age
ethnicity
region



Synthetic Network
MIA
FairMIA
Min fraction influenced
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Gender
Region
Age



Simulated Antelope Valley network
Price of group fairness
0
5
10
15
20
25
30
35
40
gender
age
ethnicity
region



Synthetic Network
Price of group fairness
0.0
0.2
0.4
0.6
0.8
1.0
Gender
Region
Age

Attribute Name	Simulated Antelope Valley network	Synthetic network
Gender	0.8457	0.7675
Age	0.5892	0.7975
Region	-0.0361	0.9315
Ethnicity	0.5251	-

```
Results on Simulated Antelope Valley network :
Attributes of the graph are : {'gender': ['male', 'female'], 'age': ['30-39', '50-59', '18-24', '65+', '25-29', '40-49', '60-64'], 'ethnicity': ['other', 'asian', 'black',
'white', 'latino'], 'region': ['desert_view_highlands', 'lake_los_angeles', 'quartz_hill', 'littlerock', 'acton', 'palmdale', 'northwest_antelope_valley', 'lancaster',
'leona_valley', 'sun_village', 'northwest_palmdale', 'northeast_antelope_valley', 'southeast_antelope_valley']}
Initialization Completed
Seeds for IM with MIA : [263, 17, 32, 268, 13, 319, 271, 28, 423, 265, 288, 464, 238, 155, 18]
Initialization Completed
Price of group fairness : {'gender': 1.0243039717433224}
Attribute fairness score : {'gender': 0.8457}
Min fraction influenced with fair MIA : {'gender': 0.04521595330739021}
Initialization Completed
Price of group fairness : {'gender': 1.0243039717433224, 'age': 1.082901458192911}
Attribute fairness score : {'gender': 0.8457, 'age': 0.5892}
Min fraction influenced with fair MIA : {'gender': 0.04521595330739021, 'age': 0.026018867924530786}
Initialization Completed
Price of group fairness : {'gender': 1.0243039717433224, 'age': 1.082901458192911, 'ethnicity': 1.7849592399553273}
Attribute fairness score : {'gender': 0.8457, 'age': 0.5892, 'ethnicity': 0.5251}
Min fraction influenced with fair MIA : {'gender': 0.04521595330739021, 'age': 0.026018867924530786, 'ethnicity': 0.008923076923076853}
Initialization Completed
Price of group fairness : {'gender': 1.0243039717433224, 'age': 1.082901458192911, 'ethnicity': 1.7849592399553273, 'region': 40.56411838912598}
Attribute fairness score : {'gender': 0.8457, 'age': 0.5892, 'ethnicity': 0.5251, 'region': -0.0361}
Min fraction influenced with fair MIA : {'gender': 0.04521595330739021, 'age': 0.026018867924530786, 'ethnicity': 0.008923076923076853, 'region': 0.0}
Price of group fairness : {'gender': 1.0243039717433224, 'age': 1.082901458192911, 'ethnicity': 1.7849592399553273, 'region': 40.56411838912598}
Min fraction influenced with fair MIA : {'gender': 0.04521595330739021, 'age': 0.026018867924530786, 'ethnicity': 0.008923076923076853, 'region': 0.0}
Min fraction influence with MIA : {'gender': 0.051510699588477385, 'age': 0.01503448275862024, 'ethnicity': 0.010976923076923169, 'region': 0.0}
/root/fairMIA/utils.py:78: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels(keys)
Attribute fairness score with fair MIA: {'gender': 0.8457, 'age': 0.5892, 'ethnicity': 0.5251, 'region': -0.0361}
Results on Synthetic network G :
Attributes of the graph G are : {'Gender': ['male', 'female'], 'Region': ['India', 'US'], 'Age': ['20-25', '50-59']}
Initialization Completed
Seeds for IM with MIA : [9, 23, 24, 31, 76, 3, 12, 48, 88, 6, 99, 11, 34, 1, 60]
Initialization Completed
Initialization Completed
Initialization Completed
Price of group fairness : {'Gender': 1.12847983005883, 'Region': 0.9858330884429136, 'Age': 1.0975268222025953}
Min fraction influenced with fair MIA : {'Gender': 0.22098199999999943, 'Region': 0.26681666666666504, 'Age': 0.22808285714284954}
Min fraction influence with MIA : {'Gender': 0.25942800000000008, 'Region': 0.26409999999999981, 'Age': 0.2945184615384677}
/root/fairMIA/utils.py:78: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels(keys)
Attribute fairness score with fair MIA: {'Gender': 0.7675, 'Region': 0.9315, 'Age': 0.7975}
```

Conclusion

- We demonstrated in our work to address the problem of fair influence maximization by defining fairness with heuristics. Ours was the first attempt to induce fairness with heuristics for influence maximization.
- We could successfully show that our proposed new heuristic-based algorithm for fair influence maximization performed comparatively better than MIA.
- Our experimental evaluations demonstrated that our proposed algorithm performs satisfactorily via the 'Price of Fairness', 'Fairness Score' and 'Maximin' metrics.
- The fairness score metric was a new attempt to determine how equitably a particular attribute was influenced.
- Our novel model constructed to generate synthetic networks for evaluating fair influence maximization has also produced a satisfactory result.


THANK YOU!