FairRankVis: A Visual Analytics Framework for Exploring Algorithmic Fairness in Graph Mining Models

Tiankai Xie, Yuxin Ma, Jian Kang, Hanghang Tong, Ross Maciejewski

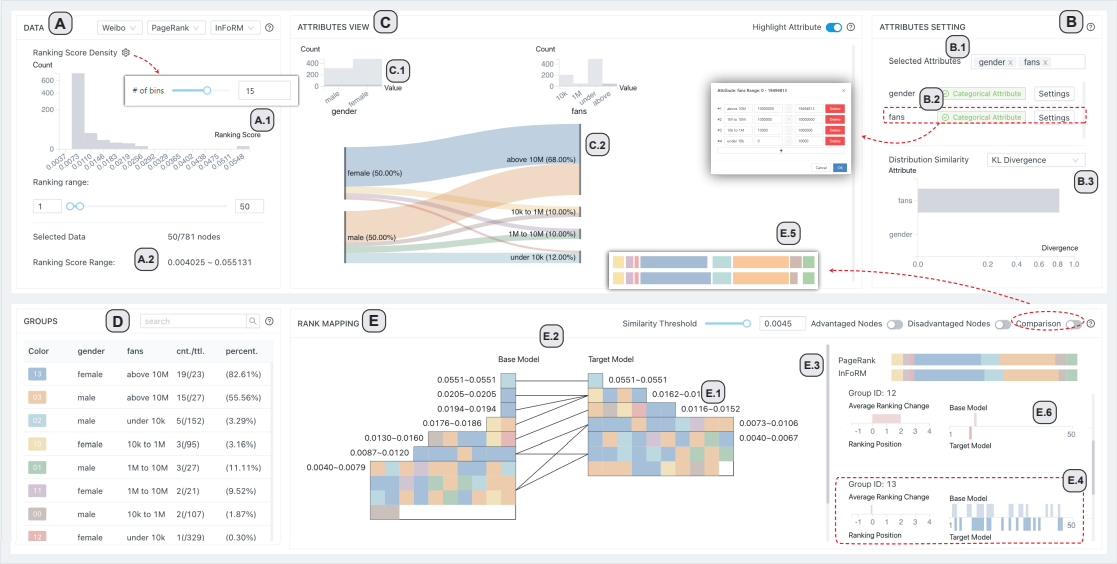

Fig. 1. Fairness diagnosis of InFoRM (a debiased ranking model) on Weibo social network data. (A) The analyst selects the top-50 ranked nodes. (B) The analyst defines the *gender* and *fans* attributes as protected classes of interest in the attributes setting panel. (C) The attributes view shows that in the top-50 ranked nodes, the *gender* attribute (`female/male`) is equally distributed. The view also shows the distribution of the *gender* attribute across the entire dataset (C.1), where it can be observed that `females` make up a larger portion of the entire dataset. The parallel sets portion of the attributes view (C.2) shows that nodes with `more than 10 million` followers make up the largest component of the top-50 nodes. (D) Selected nodes are grouped by the *gender* and *fans* attributes. (E) The rank mapping view shows that ranked nodes are clustered (E.1) based on similar ranking scores and the ranking result of the target model tends to have more similar ranking scores of top-k nodes than those of the base model. The view also supports comparison between ranking algorithms by mapping the change (E.2) in each node's rank between the two ranking algorithms being explored. The group proportion view (E.3) shows few proportional changes when comparing the original ranking algorithm to the InFoRM model. The group shift view (E.4) shows that the average ranking of the group 02 with attributes of `male` and followers `under 10 thousand` has increased by 2 positions, which may indicate that the InFoRM model has indirectly created a group preference.

Abstract—Graph mining is an essential component of recommender systems and search engines. Outputs of graph mining models typically provide a ranked list sorted by each item's relevance or utility. However, recent research has identified issues of algorithmic bias in such models, and new graph mining algorithms have been proposed to correct for bias. As such, algorithm developers need tools that can help them uncover potential biases in their models while also exploring the impacts of correcting for biases when employing fairness-aware algorithms. In this paper, we present FairRankVis, a visual analytics framework designed to enable the exploration of multi-class bias in graph mining algorithms. We support both group and individual fairness levels of comparison. Our framework is designed to enable model developers to compare multi-class fairness between algorithms (for example, comparing PageRank with a debiased PageRank algorithm) to assess the impacts of algorithmic debiasing with respect to group and individual fairness. We demonstrate our framework through two usage scenarios inspecting algorithmic fairness.

Index Terms—Graph ranking, fairness, visual analytics

- *T. Xie and R. Maciejewski are with Arizona State University. E-mail: {txie21,rmacieje}@asu.edu.*
- *Y. Ma is with Southern University of Science and Technology. E-mail: mayx@sustech.edu.cn*
- *J. Kang and H. Tong are with the University of Illinois at Urbana-Champaign. E-mail: {jiank2, htong}@illinois.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

1 Introduction

Algorithmic fairness has become increasingly important in data mining and machine learning. This has led to a proliferation of algorithmic enhancements to address potential fairness issues that can occur in black-box models. Although researchers have been developing methods to guarantee the fairness of data-driven models [2, 4, 7, 18, 28, 31, 32, 36, 37], it has been reported that biases can still be observed even after the fairness algorithms are applied [30]. The essential reason behind this phenomenon is that *it is difficult to define fairness.*

Common fairness definitions include *individual fairness* [10] and *group fairness* [24, 25], where individual fairness focuses on whether similar individuals are treated consistently and group fairness focuses

on whether or not members of a protected class have the same probability of being assigned a positive outcome (for example, the same probability of receiving a housing loan). Difficulties arise due to the fact that **the sensitive attributes**[1] **vary from task to task.** Sensitive attributes may have commonalities across tasks, and there may be legally protected classes that need to be considered when measuring fairness. However, there is no single universal definition of fairness, and different applications of an algorithm may need to alter the definition of fairness depending upon the task at hand. Furthermore, recent work [3] has found that controlling for group fairness may lead to issues in individual fairness, and it is critical to understand the various implications and trade-offs of fairness-aware machine learning tools.

Even when the task at hand has clear and explicit legal definitions of fairness, machine learning algorithms may still struggle. Take for example legal definitions of fairness that focus on gender and ethnicity attributes of the data. Here, numerous algorithms have been proposed to correct for single attribute biases. However, as noted by Wang et al. [32], algorithms might be subject to *indirect discrimination*, where a protected class attribute might be correlated to another feature in the dataset, for example, location attributes such as ZIP Code might have implicit racial information as the distribution of ethnicity is geographically unbalanced. Thus, fairness solutions that only adjust for a single data attribute can still suffer from algorithmic biases. Given such issues, **it is difficult to balance algorithmic results under potentially conflicting definitions of fairness,** and recent work [7, 11, 20] has even discussed an *impossibility theorem* for fairness noting that it may be impossible to guarantee fairness that satisfies all constraints.

Such challenges seem to necessitate a human-in-the-loop approach, where analysts can audit various definitions of fairness. Recent work in the visual analytics community has explored the development of systems for auditing machine learning algorithms with respect to fairness in classification [7, 32] and ranking [2]. However, these systems tend to focus on single attribute fairness at the group level and do not provide support for exploring the impacts of multi-attribute group fairness and individual fairness. To overcome such limitations, we have developed FairRankVis, a visual analytics framework designed to enable the exploration of multi-class bias in graph mining algorithms. The proposed framework is model agnostic, supports both group and individual fairness levels of comparison, and consists of a suite of interactive visualizations for investigating node attributes and topological features of graph elements to explore algorithmic fairness. Contributions include:

- A visual analytics framework that supports analysts in exploring multi-class bias in human-guided fairness definitions at both the group and individual levels.
- Interactive methods for auditing fairness between machine learning algorithms to help analysts diagnose model trade-offs under different fairness definitions.

2 Related Work

In this section, we review recent work in graph ranking, algorithmic fairness, and fairness in visual analytics.

2.1 Graph Ranking

Ranking is a fundamental task in graph mining and has been employed in various application domains including search engines [23], social network mining [23, 33], biology [29], and neuroscience [9]. PageRank [23], one of the most widely-used algorithms, was originally devised to retrieve relevant web pages on the Internet through hyperlinks between web pages, where the web pages were considered to be nodes and hyperlinks edges in a graph. The essential contribution of PageRank is to utilize the topological structures of the graph elements, i.e., nodes and edges, to calculate the importance of nodes:

$$\boldsymbol{r} = c\boldsymbol{A}\boldsymbol{r} + (1-c)\boldsymbol{t}, \tag{1}$$

where $\boldsymbol{r}$ is the ranking score vector for each node with size n where n represents the number of nodes. The matrix $\boldsymbol{A}$ denotes the adjacency matrix of the graph, and $\boldsymbol{t}$ the teleportation vector is initialized as $\frac{1}{n}\mathbf{1}$. The equation is computed iteratively and converges to a stationary distribution where values in $\boldsymbol{r}$ represent the importance of the nodes.

Successful applications of PageRank in web search engines have encouraged the development of numerous variants in other related research disciplines. These variants typically follow the same mechanism as PageRank but utilize extra information to enhance the traditional teleportation process. For example, a modified version of PageRank for recommendation systems, ItemRank [13], was proposed to rank items based on expected user's preferences by changing the adjacency matrix to a correlation matrix of the graph. IsoRank [29] improves the original version of PageRank by transforming the task of correspondence between nodes as an eigenvalue problem. In ranking short texts and documents, TwitterRank [33] utilizes a transition probability matrix to measure similarities between twitterers to discover influential users, and TopicRank [6] employs semantic relationships between topics as a ranking factor. AttriRank [16] differs from the previous methods by leveraging the attributes on the nodes to enhance the ranking results. However, PageRank, and its initial variations, do not consider issues of algorithmic fairness in their ranking schemes.

2.2 Fairness in Graph Mining

In order to account for potential algorithmic bias, numerous iterations of fairness-aware graph mining algorithms have been developed focusing on individual fairness and group fairness. Kamishima et al [17] employ regularization-based collaborative filtering which minimizes the average ratings among different groups to control for potential bias. In graph-based clustering, Kleindessner et al. [21] propose a fairness notion to balance the number of elements in each cluster based on different demographic groups. Bose et al. [4] employ an adversarial framework to achieve statistical parity for the learned embedding results across sensitive attributes. Kang et al. [18] study the individual fairness problem in graph mining models and propose an optimization-based framework for diagnosing and debiasing graph mining models by three individual approaches: debiasing data, debiasing model as well as debiasing result. However, these approaches only guarantee group fairness or individual fairness without considering whether applying constraints for group fairness affects individual fairness or vice versa. As such, tools that can support fairness auditing between variations of graph mining algorithms are critical for identifying algorithmic fairness.

2.3 Fairness in Visual Analytics

Given the fact that definitions of fairness can be highly task-dependent, recent work in the visual analytics community has begun exploring methods for human-in-the-loop fairness auditing and exploration. Cabrera et al. [7] propose a visual analytics framework (FairVis) for discovering intersectional bias by inspecting machine learning models' performance on different groups, where a group is defined with respect to a set of potential sensitive attributes. The analyst can select the performance metric, e.g., accuracy, F1 score, true positive rate, etc. Ahn et al. [2] propose a general visual analytics framework (FairSight) for diagnosing the fairness of top-k ranking results by considering both nodes and groups. The framework provides metrics for diagnosing both individual fairness and group fairness in terms of a single sensitive attribute. However, multi-attribute fairness diagnosis was unexplored. Wang et al. (DiscriLens) [32] also investigated issues of fairness in classification tasks by visualizing the unbalanced proportion between user-defined groups with respect to a single sensitive attribute. These approaches demonstrate the effectiveness of visual analytics in revealing and analyzing fairness-related problems. However, there are also limitations to the current approaches. FairVis only supports diagnosing biases in supervised binary classification tasks, and DiscriLens only supports exploring a single sensitive attribute. FairSight explores trade-offs between the group and individual fairness in ranking results, but multigroup fairness remains unexplored. Furthermore, none of these previous systems support model comparison as a mechanism for explaining the impacts of algorithmic debiasing.

[1] Sensitive attributes are generally defined to be traits of an individual which should not correlate with the algorithmic outcome, e.g., gender, ethnicity, age.

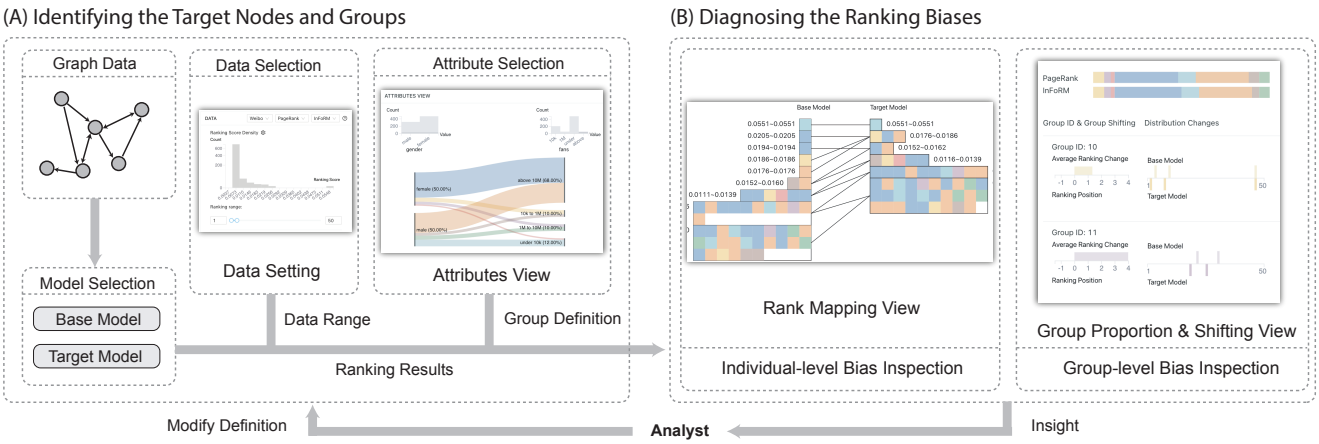

Fig. 2. The FairRankVis Framework consists of two stages: (A) the identification of target nodes and groups stage, and (B) the diagnosis of biases in ranking results stage. In stage (A), the analyst can select the base model and the target model to be inspected. The ranking results will be generated after model selection. The analyst then defines a range of nodes, either the top-k nodes or nodes who have similar ranking scores, and then defines the groups based on selected attributes. (B) The analyst can then explore and inspect both individual-level and group-level bias. The framework also supports modifying the definition of fairness at any time during the analysis process.

3 Design Overview

From our literature review on fairness in graph ranking and visualization, we have identified several research challenges and gaps in the literature. These challenges were then evaluated with three data mining researchers who specialize in debiasing algorithms for graph learning models (two of which serve as co-authors on this paper). After iterative discussions with the experts, two major research challenges for auditing fairness in graph mining algorithms were identified:

Task-oriented Definitions of Groups. In conventional debiasing approaches, the definitions of protected groups may vary across applications. Typical examples include personal attributes associated with discrimination, such as gender, ethnicity, age, etc. However, identifying sensitive attributes and characterizing protected groups is a non-trivial task and demands expert knowledge to identify potential discrimination [32]. As such, there is a need for methods that can interactively define fairness, incorporate this definition into a debiasing method, and audit the impacts of the debiasing. In this paper, we use the term *group* to denote the protected groups characterized by sensitive attributes.

Trade-offs Between Group and Individual Fairness. Ideally, fairness adjustments to a machine learning model will maintain fairness between groups of nodes with similar attribute values. However, conflicting concepts of group and individual fairness [3] can lead to cases where an algorithm that has been debiased at the group level now introduces bias at the individual level. Consider an employment recommendation system that meets the criteria for group fairness. Applicants in protected groups may receive more competitive rankings in order to keep statistical parity on selected attributes (such as gender or ethnicity). However, other candidates with similar abilities may now be de-ranked in order to ensure group fairness. Thus, it is crucial that algorithm designers have the means to explore individual and group fairness.

3.1 Analytical Tasks

We have also identified common ranking analysis and fairness auditing tasks that could benefit from a visual analytics approach. These tasks were refined through discussions with our co-authors who are the lead developers of several recent fairness aware graph mining algorithms.

T1: Define Target Nodes and Groups. Analysts should be able to specify sensitive attributes and inspect protected groups by defining:
- **T1.1:** Which portion of nodes are the most important, and;
- **T1.2:** Which attributes are critically important for fairness.

T2: Reveal the Impact of Topological Structures and Attributes on Ranking Fairness. Analysts should be able to diagnose the algorithmic fairness of graph ranking models by understanding the impacts of their topological structures and attributes on ranking fairness. Since the ranking results have no ground truth and are sensitive to changes in the graph structure [34], a base model is needed as a reference in order to explain the debiasing impact of a target model.Additionally, group fairness and individual fairness may have conflicting rule sets. When comparing models, analysts want to explore:
- **T2.1:** Which nodes are advantaged/disadvantaged by the model?
- **T2.2:** Which groups are advantaged/disadvantaged by the model?

T3: Diagnose Content Bias in Ranking Results. Display space is a bottleneck for showing all individual rankings. For example, Google searches list approximately 20 records per page, and the higher the rank, the more clicks. However, records listed on later pages may have similar relevance to the top ranked pages. This phenomenon has been studied by Pitoura et al. [26] which noted that *content bias* may occur when information is displayed in different ways. There two major analytical questions when diagnosing content bias:
- **T3.1:** Which nodes have similar relevance (ranking scores)?
- **T3.2:** What is each node's position in the ranking result, and how likely is it that content bias has occurred in similar nodes?

3.2 Design Requirements

Based on the analytical tasks, we engaged in an agile design process involving multiple iterations of the FairRankVis framework in collaboration with our domain experts. We have identified several design requirements and mapped different analytic tasks to each requirement.

D1: Visualize the Attribute Compositions of Target Nodes. The system should support the selection of target nodes from the graph (T1.1). To enable the inspection of node attributes, the system should interactively visualize the composition of attribute values among selected nodes and visualize necessary metrics to assist analysts in selecting attributes for future diagnosis (T1.2).

D2: Visualize the Algorithmic Bias and Content Bias. The system should visualize both algorithmic bias (T2) and content bias (T3) for selected nodes and attributes with the following views:
- **D2.1:** *Rank Mapping View*, which integrates ranking results that are mapped from the base model to the target model (T2.1) as well as the summary of nodes that have similar ranking scores. (T3.1, T3.2)
- **D2.2:** *Group Proportion View*, which compares the proportional difference in terms of analyst-defined groups. The view should support a global proportion overview and a pair-wise proportion difference in terms of each group. (T2.2)
- **D2.3:** *Group Shift View*, which shows how analyst-defined group rankings shift from the base model to the target model. (T2.2)

4 Visual Analytics Framework

Based on the analytic tasks and design requirements, we have developed a visual analytics framework (Figure 2) to support fairness auditing in graph-based ranking algorithms. The framework is designed to first load the graph data and then compute the ranking results using the analyst selected targeted model and base ranking model (Figure 2 A). Then the analyst can interactively define the target attributes for fairness auditing (Figure 2 B). As the definition of group and target nodes are updated by the analyst, the ranking results are updated across all views to support bias inspection. Analysts can modify the group definitions at any time to explore issues of algorithmic fairness.

The framework supports two major functionalities: 1) identifying the target nodes and groups, and 2) diagnosing potential ranking biases. By identifying the target nodes and groups, the analyst can select a portion of nodes according to their specific analytical goals and explore the attribute distributions. The selected nodes are automatically categorized by the analyst-defined groups. The analyst can also explore the ranking results of both the base ranking model and the target ranking model to explore group/node shifts, proportions, and distributions of similar nodes. The analyst can flexibly modify the definition of a group at any time to explore both single and multi-attribute fairness. Our modular design enables analysts to freely integrate any graph-based ranking models for use as the target or base model. For demonstration purposes, we apply PageRank as the base model and AttriRank and a debiased PageRank (InFoRM) as the target models.

4.1 Background of Graph Ranking Models

AttriRank [16] is a PageRank-based model that uses the topological information and node attributes to compute the ranking vector $\boldsymbol{r}$:

$$\boldsymbol{r} = c\boldsymbol{Q}\boldsymbol{r} + (1-c)\boldsymbol{P}\boldsymbol{t} \tag{2}$$

where

$$P_{ij} = \begin{cases} \frac{1}{\delta_j}, \text{ if directed edge}(j,i) \in E \\ \frac{1}{N}, \text{ if} \delta_j = 0 \\ 0, \text{ otherwise} \end{cases} , Q_{ij} = \frac{s_{ij}}{\Sigma_{k \in V} s_{kj}} \tag{3}$$

δ_j denotes the out-degree of node j, and s_{ij} the degree of similarity with respect to the attribute values of the nodes. In AttriRank, the Radial Basis Function (RBF) kernel is defined as the similarity measure:

$$s_{ij} = e^{-\gamma ||x_i - x_j||_2^2} \tag{4}$$

where γ denotes the distance influence. In this way, external attribute values are integrated into the ranking procedure which is more robust for handling nodes that have missing edge information.

InFoRM [18] is a generic individual fairness framework for quantitatively measuring the potential bias in graph mining tasks including graph ranking, clustering and graph embedding. The InFoRM framework can perform three types of debiasing methods including (1) debiasing the input graph, (2) debiasing the graph mining model, and (3) debiasing the mining result. We employ InFoRM to debias the ranking results of PageRank to simulate a situation where the debiased model does not have access to the input data and the model. Mathematically, this process is realized with the following objective function:

$$Y^* = \arg\min_Y J = ||Y - \bar{Y}||_F^2 + \alpha Tr(Y^T L_S Y) \tag{5}$$

where Y^* denotes the debiased ranking result, $\bar{Y}$ denotes the original ranking result. $\alpha > 0$ is the regularization parameter, and L_S is the Laplacian matrix of the similarity matrix S[2]. This equation minimizes the sum of the squared Frobenius distance between ranking results and the regularized tethnicity of the matrix produced by $Y^T L_S Y$ so that both the difference of the ranking results before and after debiasing (Y and $\bar{Y}$) and the bias (defined as $Tr(Y^T L_S Y)$) are minimized.

[2]The similarity matrix S uses cosine similarity and Jaccard similarity.

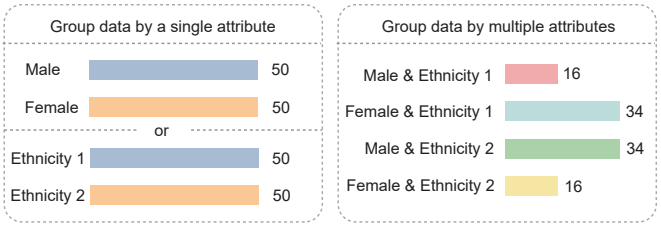

Fig. 3. Different group definitions can lead to different fairness insights. Suppose there are 100 nodes who have similar ranks. If we group the nodes only by ethnicity or gender, the proportions are equal, which might imply that the outcome is fair in terms of both ethnicity and gender. However, if we group the nodes by both ethnicity and gender, we may find potential inequalities at the intersection of the two attributes.

4.2 Identifying the Target Nodes

Our framework is designed to enable a flexible definition of ranks and attributes to be considered when diagnosing fairness. Recent research [8, 35, 37] emphasizes that the top-k elements will receive more attention, and ranking bias is typically explored with respect to the top-k ranks. In our proposed framework, a data setting panel (Figure 1.A) is configured to enable the analyst to select the top-k nodes. This is facilitated by the **Ranking Score Density Histogram** (Figure 1.A.1), which shows the ranking score distribution for the target ranking model. The analyst can interactively modify the number of bins by clicking the gear icon, and the histogram supports brush selection to select a specific ranking range **(T1.1)**. For example, if the analyst cares about potential biases of nodes who have similar ranking scores, then the analyst can brush a particular bin on the histogram and all the nodes within that ranking score range are selected. If the analyst wishes to select a specific ranking position, a slider is configured to enable the analyst to select nodes from rank m to n. In this way, the analysts can explore how attributes are distributed for any specific range of ranks.

4.3 Defining Groups

Once a range of nodes is selected, the analyst is able to explore attribute information and define groups through the attribute setting panel (Figure 1.B) and attribute view (Figure 1.C). Recent work [10, 19] suggests that a general fairness principle is based on whether *similar nodes will have a similar ranking*. In other words, defining a group means defining individuals that are similar. Wang et al. [32] note that the definition of similarity is not easy to obtain and may vary from task to task. The wrong definition of similar nodes can lead to wrong conclusions with respect to bias and fairness. Figure 3 shows a simple example of this phenomenon: nodes who have similar ranks are distributed evenly if we only group them by either ethnicity **or** gender. However, the data reflects a disproportionate distribution when we group the nodes by ethnicity **and** gender. Our framework enables analysts to explore all available attributes and across combinations of attributes. In our framework, the analyst selects one or more categorical attributes, and each combination of category is now considered a group. From the example in Figure 3, if the analyst selects gender and ethnicity, there would be four groups to be audited for fairness.

Attributes View. To support the interactive definition of groups **(T1.2)**, we have designed an attribute setting panel (Figure 1.B) and an attribute view panel (Figure 1.C). The attributes view panel employs a parallel set where each selected attribute is visualized with multiple bars. Selected nodes are encoded as curves with different widths. Both the height of bars and the width of the curves encode the number of nodes mapping to a specific attribute value. Additionally, the distribution of attributes across the selected nodes is visualized with a histogram (Figure 1.C.1). We use a light grey color to show the attribute distribution for the entire dataset, and the dark grey color histogram shows the distribution of attributes for the selected nodes. The attribute setting panel (Figure 1.B) enables the flexible selection of one or more attributes by clicking on the multiple selection area (Figure 1.B.1). All corresponding views including the attributes view (Figure 1.C), the group

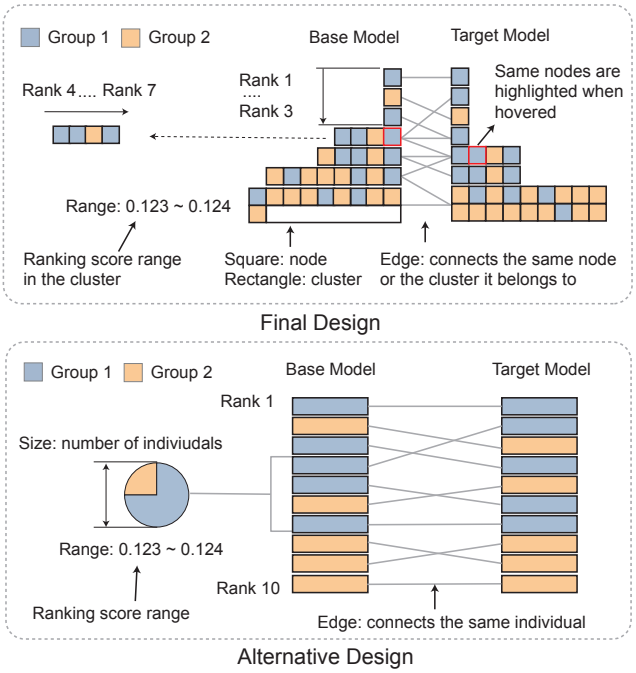

Fig. 4. Final design of the rank mapping view (top). The ranking results of the base and target model are listed separately. Small squares represent nodes and are colored with respect to the analyst-defined groups. These squares are organized into large rectangles, and each rectangle represents a cluster that contains nodes with similar ranking scores. From top to bottom, the nodes are ranked from m to n (in the example $m = 1$ and $n = 30$), and, in a cluster, the nodes' ranks from high to low are mapped from left to right. Each cluster from the base model is connected to a corresponding cluster in the target model by a grey line when they share the same node(s). Alternative design (bottom). Each rectangle is a node ranked from m to n displayed vertically. The left column shows the ranking results of the base model, and the right column shows the ranking results of the target model. Each node is connected to its counterpart by a grey line to illustrate how the ranking changes between models. The color of the bar maps to the analyst-defined group. The pie chart shows the proportion of groups in a cluster containing nodes with similar ranking scores and is proportional to the number of nodes.

table view (Figure 1.D), the rank mapping view (Figure 1.E), the group proportion view (Figure 1.E.3) and the group shift view (Figure 1.E.4) are automatically updated as the selected attributes are changed. Since group fairness is most often based on categorical attribute values, we also include a customization feature that allows analysts to categorize attributes that may have continuous values. For example, protected classes for age are often grouped into ranges, e.g. under 18, 65+, etc..

We also provide another histogram (Figure 1.B.3) to facilitate the comparison of distribution similarities on selected attributes between selected nodes and the entire dataset. The metric for measuring distribution similarities can be customized based on the analysts' needs. Currently, the framework supports Kullback-Leibler divergence for demonstration purpose. The height of the bars are mapped to the differences of the between the distributions of the selected nodes and the entire dataset on a specific attribute.

4.4 Diagnosing the Ranking Biases

Once the nodes are selected and sensitive attributes defined, the corresponding groups are automatically generated, assigned a label and unique color, and displayed in the group table (Figure 1.D). Once groups are defined, the fairness audit can begin. Here, it is important to note that biases in machine learning models can arise due to issues with the *Data* and/or issues with the *Model*.

Diagnosing Data Bias. Real-world data can be either insufficiently sampled or reflect existing prejudices. Thus, it is inappropriate to ask

Algorithm 1 Clustering Similar Ranking Scores

Inputs: similarity threshold δ; selected nodes V;
Outputs: clusters C with maximum ranking score difference $d \leq \delta$
for k in range $(1, V.\text{length})$ **do**
  $\mathbf{C} \leftarrow k_{means}(k, V)$
  if $d_c <= \delta, \forall c \in \mathbf{C}$ **then**
    return **C**
  end if
end for
Return C

models to be fair when being optimized on biased data. In terms of graph ranking, it is critical to understand how groups are distributed prior to applying a debiased ranking model. Our system first ranks the data with what we refer to as the *base* model. For demonstration purposes, we employ PageRank as the base model.Exploring the base model can help reveal the underlying topological features of the data. What we are interested in is if there are already signs showing disproportional distributions for each group. From the base model ranking, we can explore whether certain groups have higher ranking scores than others. For example, if the base model (PageRank) shows that when evaluating node ranking based on gender, nodes that are marked as male are ranked relatively higher than female nodes, then other PageRank-based models are very likely to observe a similar distribution between the male group and the female group. In this case, the gender bias is not inherited from the model but the data.

Diagnosing Model Bias. Our framework supports diagnosing three types of bias: Content **(T3.2)**, Group **(T2.2)** and Individual Bias **(T2.1)**.

1. *Content Bias.* In real-world applications, a full ranking of millions of items simply cannot be displayed, and is typically culled to some top-k rank. In this setting, even the nodes who have the same ranking scores can have a large difference in ranking positions, and this problem is referred to as content bias. For example, imagine a list of items where the second through the seventh item have identical ranking scores. The method of display implies inequality in ranking even though ranks two through seven have equal ranking scores. Here, the implicit ordering can lead to significant differences in their exposure rates. To help analysts explore this phenomenon, we group nodes into clusters based on their ranking scores **(T3.1).** Algorithm 1 shows the k-means-based clustering algorithm. The idea of the clustering algorithm is to group as many nodes as possible into a cluster such that the maximum difference between ranking scores in this cluster is less than the analyst-defined similarity threshold. The algorithm outputs the minimum number of clusters to satisfy the analyst-defined rules of similar ranking scores. The analyst can inspect the cluster for signs of possible content bias.
2. *Group Bias.* Many fairness metrics have been proposed for measuring group fairness [10, 14]. These methods attempt to measure the degree of discrimination or bias [27]. However, there is no single term that universally represents bias. We denote *group bias* as the bias that reflects the ability of the model to achieve statistical parity between groups, where a group is defined with respect to the analysts' selected sensitive attributes. The goal of the framework is to enable analysts to audit whether the ranking results of a model exhibit direct or indirect preferences towards one or more groups, resulting in lower ranking scores for the disadvantaged groups. Compared with the content bias, where disadvantages can be due to display constraints, group bias can be mitigated algorithmically. To observe the impact on groups' ranking between the base and the target model, we formalize the ranking changes for each group by computing the average ranking position change **(T2.2)**:

$$\Delta_g = \frac{1}{n} \Sigma_{v \in V_s, v \in g} (r'_v - r_v), \tag{6}$$

where Δ_g is the average ranking change of group g among selected nodes V_s, r'_v is the ranking position of node v in the target model, r_v is the ranking position in the base model, and n is the number of

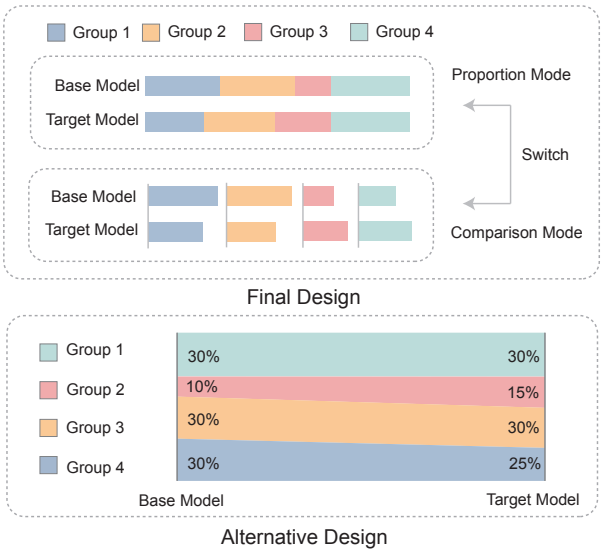

Fig. 5. Final design of the group proportion view (top). The bar chart on the top shows the average ranking change of each group. The color of the bar encodes the identity of the group. The bar chart on the bottom shows the distribution of group members in the base and target model. The x-axis maps to the ranking position of selected nodes. Alternative design (bottom). Two axes on the left and right represent the group proportion of the base and target model colored by group.

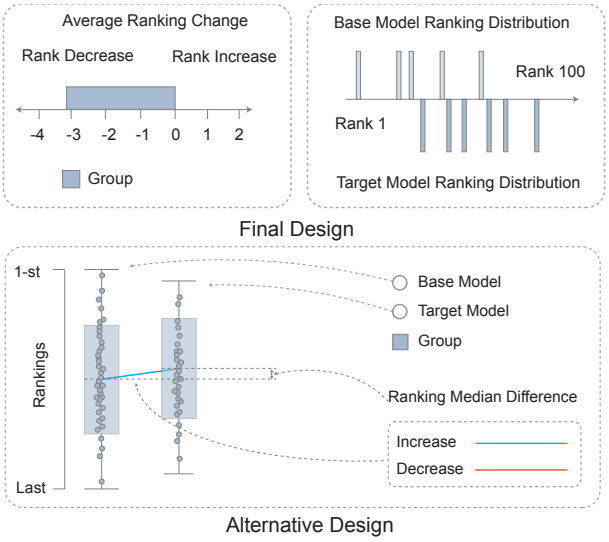

Fig. 6. Final design of the group shift view (top). The bar chart on the left shows the average ranking change of each group. The color of the bar encodes the identity of the group. The bar chart on the right shows the distribution of group members in the base model and target model, where the x-axis maps to the ranking position of selected nodes. Alternative design (bottom). We employ two box plots to show the ranking distribution in the base model and the target model respectively. A colored rectangle highlights the distributed nodes from the first quartile to the third quartile, where the color is encoded as an analyst-defined group. We link the median of the distribution of a given group from the base model to the target model and color the line with light green/red if the median rank increases/decreases.

nodes in both the selected nodes and group g.

3. *Individual Bias.* Individual bias represents how the model guarantees that nodes with similar attributes will receive similar rankings. It is important to understand if individual nodes have been "sacrificed" or privileged by the model in order to reduce group bias. To help analysts explore the individual biases among selected nodes, we label the selected nodes as advantaged/disadvantage nodes according to their ranking position changes (increase/decrease) between the base and target model **(T2.1)**.

Rank Mapping View. The rank mapping view (Figure 4 (top)) consists of two columns of stacked rectangles, where the left column shows the ranking results of the base model, and the right column shows the ranking results of the target model. For each column, small squares that represent nodes of the analyst-defined groups are organized into large rectangles, where each rectangle represents a cluster (from Algorithm 1) that contains nodes with similar ranking scores **(T3.1, T3.2)**. From top to bottom, the nodes are ranked from m to n, and in a cluster, the nodes are mapped from left to right according to their rank (high to low). Each cluster from the base model is connected to a corresponding cluster from the target model by a grey line when they share the same node(s), which illustrates how the ranking of this node changes between models **(T2.1)**. The color of the square maps to the analyst-defined groups. In this example, we can observe that members in group 1 have a relatively higher rank position than those in group 2 from the top-1 to top-10 ranks, In Figure 4 (top), we can observe that there are four nodes belonging to a cluster with a ranking score from the base model ranging from 0.123 to 0.124. Three of the nodes are in group 1, while only one is in group 2. The node from group 2 has been ranked in the sixth position. This means that even though their ranks are functionally equivalent, the node belonging to group 2 will likely have a lower exposure rate than equivalent nodes in group 1, indicating that content bias may occur. An alternative design is shown in Figure 4 (bottom).

Such phenomena can be significant when the size of the cluster is larger. Imagine a cluster with 30 nodes whose ranking scores are functionally equivalent. The 30th node is so far below the 1st node of this cluster that the differences in exposure are extremely uneven. Such content bias is inevitable given the traditional ranked list displays in real-world applications; however, the analyst should at least be aware of any content bias and can consider modifications to the display list to adjust for such biases. There are also switch buttons that allow the analyst to highlight advantaged/disadvantaged nodes (Figure 1.E.5), and analysts can hover on a single node to see the same node in the ranking result of another model. The tooltip is used to show node attributes on mouseover. If analysts click on a single node, the view will highlight all nodes in the corresponding cluster and their corresponding ranking positions in the debiased model.

Group Proportion View. The group proportion view is designed to illustrate the target ranking model's effects on each group's proportion **(T2.2)**. The group proportion view consists of two sets of bars and each set shows the composition of selected nodes sorted by both ranking models respectively (Figure 5 (top)). To facilitate inspection, we support switching the view mode between the proportion mode and the comparison mode. The proportion mode displays the stacked bars to summarize the overall group distribution of the selected nodes, while the comparison mode supports a direct comparison of group proportions between models. In other words, the comparison mode helps analysts perform pair-wise comparisons of the same group proportions between different models. Analysts can toggle between the proportion and comparison mode by using the switch button on the right side of the title bar of the rank mapping view (Figure 1.E). An alternative design (that was ultimately discarded) is shown in Figure 5 (bottom).

Group Shift View. The group shift view (Figure 6 (top)) is designed to inspect both group bias **(T2.2)** and individual bias **(T2.1)**. For group bias, the bar chart on the left shows the average ranking change of each group. The color of the bar encodes the identity of the group. The bar chart on the right shows the distribution of group members in the base model and target model, and the analyst can diagnose group shifts in selected nodes to understand the corresponding fairness trade-offs between models. For individual bias, analysts can hover on the squares of the ranking mapping view to trigger the highlighting of that node on the right side of the group shift view. This can help analysts explore if individual bias occurs when applying debiased algorithms to achieve group fairness. An alternative design is shown in Figure 6 (bottom).

Interactions. Our framework employs multiple coordinated views to allow analysts to inspect group, individual, and content biases. These views are supported by a set of rich interactions. The selection of data in the data summary view (Figure 1.A) directly updates the content of the attribute view (Figure 1.C) and the rank mapping view area which includes the group proportion view and the group shift view (Figure 1.E). The attribute view (Figure 1.C) is dynamically updated based on selected attributes in the attribute setting panel (Figure 1.B), and selections in the attribute setting panel also updates the colors of the entire system as the colors encode the analyst-defined groups. For the rank mapping view (Figure 1.E), analysts can adjust the similarity threshold slide bar to define how nodes are clustered based on the ranking scores, and analysts can toggle advantaged/disadvantage nodes to highlight nodes that have the rank increase/decrease. Analysts can also hover over squares in the rank mapping view to highlight and locate the node's position in both the vanilla and debiased algorithm, and tooltips are used to provide details of the node attributes. Along with hovering, analysts can also click on a node to show how the ranking positions of all nodes in the cluster change from the base model to the target model. Finally, analysts can toggle the comparison model to enable pairwise comparison between models, Figure 1.E.5.

5 Usage Scenarios and Expert Review

In this section, we present two usage scenarios to demonstrate how our framework supports fairness audits in graph-based ranking models. We first show how graph ranking model developers analyze the potential bias in AttriRank model. Next, we illustrate how fair ranking model developers inspect the trade-off by applying a debiased ranking model (InFoRM). Finally, we report on an expert review of the system conducted with four domain experts.

5.1 AttriRank Bias Inspection on Facebook

In social network analysis, ranking algorithms utilize an account's topological structure and demographic information for a variety of tasks including link prediction [12], advertising [15] and recommendation [13]. Biases in rankings can have a huge impact with regard to content exposure, personal opportunities, and business strategies. As such, auditing the ranking algorithms used in such systems can help analysts understand whether the ranking results can comprehensively be considered to be unbiased under a variety of fairness definitions. For example, in the recommendation-based social network application, if accounts of male users are more likely to be recommended than female users, those male users will have more opportunities for content exposure and networking opportunities. Even in the case where male and female users have equal rankings, their level of exposure might still be affected by the ranking position arrangement. Here, content bias can effectively drive more clicks to the top-1 account, when, in reality, the top-10 account may have an equal ranking. Furthermore, when exploring group-level fairness, single attribute fairness audits may show that results are balanced. However, the intersection of sensitive attributes, e.g. gender, ethnicity, age, might reveal further biases in the rankings as it is possible for a ranking model to learn biased patterns both implicitly or explicitly so that certain groups are treated with advantages while others are disadvantaged. In this usage scenario, we audit AttriRank [16] when applied to a Facebook social network dataset [22]. The data is subsampled to a subgraph with 734 nodes and 74254 edges. Each node has 24 attributes that describe the demographics of a user. All identifiable information is anonymized and some attribute values are suppressed. In this usage scenario, we assume that the model developers have no prior knowledge about the data.

Identifying the Target Nodes and Groups (T1): The data setting view displays the ranking score density distribution (Figure 7.1.a). The majority of the nodes have a ranking score ranging from 0.0017 to 0.0020. The analyst selects the top-25 nodes to explore the results of AttriRank. The analyst inputs 25 into the right-hand input box of the ranking range section, and the bottom of the data setting view shows the information of the selected nodes. Next, the analyst explores the attribute distributions in the attributes view and see that attributes political and region have been suppressed for the majority the nodes. The analyst chooses to remove such attributes from the analysis. Among the top-25 nodes, the analyst finds that there are two attributes with heavily non-uniform distributions: (1) the ratio of the *gender* value, which has two classes - `feature 78` and `feature 77`, and the ratio between the two classes is 88% to 12% respectively. (2) The *locale* has five classes, and the selected nodes with the *locale* value of `feature 127` make up a greater portion of the dataset than other *locale* values (Figure 7.3). The analyst then select *gender* and *locale* to serve as the sensitive attributes that form the basis of our fairness audit.

Diagnosing the Ranking Biases (T2): After selecting *gender* and *locale* as the criteria for defining target groups, all possible groups are generated and displayed in the groups view as shown in Figure 7.4. Among the 6 generated groups, the analyst identified that group 78127 (*gender* value `feature 78` and *locale* value `feature 127`) has 16 members in the top-25 ranks, thereby occupying the majority of the top-k ranks. Given the disproportionate representation by group 78127, the analyst decides to further explore the effects that AttriRank had on the ranking distributions compared with the base model (PageRank). By exploring the group proportion view (Figure 7.5.a), the analyst observes that group 78127 was also disproportionately favored in the top-25 rankings by the base model, PageRank. This indicates that *the reason that the nodes in* group 78127 have a higher rank is due to their topological features as opposed to the attribute rank based adjustments from AttriRank. The group proportion view also shows that the proportion for each group in the top-25 rankings saw no significant changes between the PageRank and AttriRank rankings, with only group 77127 and group 78127 seeing small changes in representation.

Next, the analyst inspects for content bias in the ranking results (Figure 7.6). Here, the analyst considers nodes with ranking scores that are within $\varepsilon = 0.0005$ of each other to have the same rank and sets this threshold number as the similarity threshold. By inspecting the rank mapping view, the analyst observes that the top-25 nodes can be grouped into 5 clusters for both PageRank and AttriRank. The top-3 nodes have substantially different rankings and form 3 unique clusters in both ranking models. For the remaining clusters, two clusters (the fourth ones) of both models in Figure 7.6.a and Figure 7.6.b cover the same ranking score range from 0.0024 to 0.0027. In other words, nodes in these clusters have approximately the same relevance or utility. However, their ranking positions range from 4th to 9th in PageRank and 4th to 10th in AttriRank, indicating that content bias is occurring and it is slightly more pronounced in AttriRank than PageRank.

Finally, the analyst inspects the effect of AttriRank's behavior on the top-25 nodes. By exploring nodes of rank mapping view (Figure 7.6.c), the analyst finds that node 1199 experiences a significant ranking drops from 11th to 21, which indicates that the AttriRank sacrifices the node during the ranking process, which may lead to individual bias. From the group shift view (Figure 7.5) the analyst also observes that the group 78127 is the only group that has an average ranking decrease. AttriRank is designed to adjust rankings such that nodes with similar attributes have similar ranking scores; however, this optimization may bias the results towards specific groups. Thus, auditing tools, such as FairRankVis, can help analysts evaluate tradeoffs between algorithms, inspect for biases, and audit fairness definitions.

5.2 InFoRM Bias Inspection on Sina Weibo

In the second usage scenario, the analyst compares a debiased ranking model (InFoRM [18]) to the vanilla version (PageRank [23]) and explores tradeoffs between individual and group fairness. Overemphasizing group fairness can propagate issues of individual fairness, and it is difficult to balance the ranking positions to guarantee both group fairness and individual fairness. As such, it is necessary for model developers to understand the trade-offs of a debiased ranking model when applied to a given dataset. Here, the analyst explores a social network dataset collected from Weibo [1] where each node consists of four social attributes (*gender*, *fans*, *account level*, and *location*) about the demographic information of a Weibo user. For demonstration purposes, we subsampled the data down to 781 nodes and 2315 edges. Again, the analyst has no prior knowledge about the dataset.

Fig. 7. AttriRank Bias Inspection on Facebook. (1) We select the top-25 nodes with ranking scores ranging from 0.002105 to 0.005230. (2) We avoid selecting attributes that are suppressed for most nodes, and choose to use *gender* and *locale* as our sensitive attributes to divide nodes into groups. (3, 4) We notice that the group 78127 in which members have *gender* value `feature 78` and locale value `feature 127` has the largest proportion, and (5) the group proportion view shows that a large portion of group 78127 is found in the top-k ranking results from both models. (6) From rank mapping view, we find that cluster (6.a) and (6.b) contains nodes with similar ranking scores from 0.0024 to 0.0027, while these nodes have different ranking positions the difference in ranking scores is less than the analyst-defined threshold $\epsilon = 0.0005$, which has implications for content bias. (6.c) Finally, we find that node `1199` experiences a large ranking drop from 11th to 21st, which has potential implications for individual fairness. For group bias, we find that group 78127 in the group shift view (6.c) was negatively influenced by AttriRank, with the average ranking and top-k proportion decreasing when compared to their rankings from PageRank.

Identifying the Target Nodes and Groups (T1): After the models and the dataset are loaded, the analyst inspects the Ranking Score Density Histogram and observes that the nodes are concentrated at a ranking score of around 0.0073 (Figure 1.A.1). The analyst is interested in how top-k nodes with different ranking scores are affected by InFoRM. The analyst selects the top-50 nodes as the target nodes. In the data setting view (Figure 1.A.2), it can be observed that most of the ranking scores for the selected nodes lie in the range between 0.004025 and 0.055131.

Then, the analyst explores the attributes in the attributes view. Here, the analyst observes that the proportions of `males` and `females` are nearly identical, i.e., 50%:50% (Figure 1.C.2). However, the global *gender* distribution on the entire dataset shows a completely different pattern where the proportion of `females` is larger than `males` (Figure 1.C.1). Next, when inspecting the attribute *fans*, which describes the number of followers for the user, the attributes view shows that the majority of users (88%) have over 10 thousand followers, resulting in mismatched distributions between the selected group and the entire dataset (Figure 1.C.2). Since these two attributes show contrasting proportions between the full dataset and the top-50 nodes, the analyst decides to explore the ranking effects of nodes who have the same *gender* class and the same *fans* class. The group table view shows that there are 8 groups generated by this split, and the analyst finds that the nodes with `more than 10 million` followers have the largest population in the top-50 rankings (Figure 1.D). Furthermore, most of the `female` users (82.61%) and the `male` users (55.56%) who have `more than 10 million` followers appear in the top-50 user list.

Diagnosing the Ranking Biases (T2): To further understand how groups are affected by a debiased ranking model which focuses on maintaining individual fairness, the analyst first inspects how groups are distributed among the top-k nodes. By exploring the group shift view (Figure 1.E.6), the analyst observes that the group 13 (representing `female` users who have `more than 10 million` followers) tends to have higher rankings than group 03 (representing `male` users who have `more than 10 million` followers). The analyst wonders whether it is the target model that favors group 13 by increasing the ranking scores of the nodes in group 13. By observing rank mapping view (Figure 1.E.2), the analyst finds the group 13 also has higher rankings than the group 03 when nodes are ranked by the PageRank model. This indicates that group 13 is not favored by the target model.

The analyst further inspects the changes of the group's proportions in the group proportion view (Figure 1.E.3), and the analyst observes the proportion of groups are quite similar between ranking results in PageRank and those in InFoRM. By toggling the comparison mode to enable pair-wise comparison, the analyst finds that the proportion of group 13 has slightly increased, and the proportion of group 00 (representing male users who have followers between 10 thousand and 1 million) slightly decreased. Other groups distribution across the top-50 rankings maintain relatively the same proportion. When observing the group shift view, the analyst finds that group 03's average ranking decreased by 1 position and group 02 (representing `male` users who have `less than 10 thousand` followers) increased by 2 positions.

Here, the analyst wants to inspect the content bias of the ranking results from the target model. By tweaking the similarity threshold, the analyst finds that given the similarity threshold 0.0035, the top-50 nodes are clustered into 6 clusters (Figure 1.E.1) and each cluster has relatively more nodes compared with clusters of the base model. This indicates that the debiased ranking results tend to manipulate nodes to have similar ranking scores and reduce the potential for individual bias. However, this results in a larger content bias.

5.3 Expert Review

To further evaluate our framework, we conducted an interview with our collaborators (E0, E1), two domain experts (E2, E3) in graph mining and two domain experts (E4, E5) in machine learning and artificial intelligence. For the interview, we first introduced our system by describing the analytical tasks supported in the framework. We then demonstrate the components of our framework by walking through one of the two usage scenarios described in Section 5.1 and 5.2. Finally, the experts were allowed to freely explore the two datasets in the usage

scenarios. The duration of the interview was approximately 90 to 100 minutes. On average, experts spent approximately 7 to 10 minutes to master the system and were able to explore bias information based on their own. All experts were able to fully understand the major functionalities of the system by asking a few questions during the exploration phase. After the free exploration stage was finished, we collected feedback from the experts using the following questions:

1. How well are the proposed analytical tasks supported? (Q1)
2. What are the traditional ways of addressing such tasks in conventional fairness audits of graph mining models? (Q2)
3. How effective is this framework in supporting fairness audits? (Q3)
4. How would the framework fit into your development circle? (Q4)
5. Rate the user experience from 1 to 5 (poor to good) considering the views, interactions and effectiveness of the workflow. (Q5)

Framework: We received positive feedback on the overall design of the framework. The experts noted that it is necessary to have such a framework to explore and identify fairness issues in graph mining models. E0 and E1 considered that the flexibility in defining target nodes and groups vastly facilitates the task-oriented analysis by swapping the nodes and groups on the fly. E1 appreciated the design of the rank mapping view, especially the support for individual-level bias inspection. *"Usually, only an aggregated measure is reported for the individual biases on all the nodes in a graph, and we also have to visualize the biased result for each node to fully obtain the information of individual bias (Q2). With the help of interactive visualizations, we can clearly observe the biases for each node in a detailed manner, which benefits the in-depth analysis and reasoning of fairness issues in different ranking algorithms. (Q3)" E2 and E5 appreciated that the framework fits the general process of fairness auditing since the fairness issues have attracted much attention; however, the definition of fairness is subjective and context-dependent. Thus, by enabling an interactive definition of sensitive attributes, this framework can support a quick reanalysis of fairness under different constraints. (Q4)*

Visualizations: The experts all agreed on the effectiveness of the visualization design in our framework (Q1). They noted that the combination of rank mapping view, group proportion view and group shift view can illustrate the impact of the ranking models on defined groups and nodes. E2 noted that the two modes of the group proportion view can reveal information in both group proportions and group-wise comparisons between models. E3 appreciated the design of the rank mapping view which depicts both individual bias and group bias simultaneously. *"This view could assist us in checking how effective the debiasing methods can be. The result can be easily observed in the rank mapping view." The average score for the user experience question is 4 out of 5 (with the lowest score being a three and the highest a 5) (Q5). Experts agreed that the workflow is clear and were enthusiastic about the ability to flexibly define protected groups.*

Limitations: The experts also offered several suggestions for improvements to the framework. E0 discussed the possibility of supporting comparisons between more than two models. *"This can speed up the fairness-oriented model selection procedure if a number of models can be compared and analyzed at once."*. E2 recommended that for groups in the rank mapping view, the details of the advantaged and disadvantaged nodes can be queried. For example, the analyst would like to highlight specific nodes in a group. E3 and E4 found the interface to be initially challenging, and these experts required the longest amount of time for training (10 minutes). They also often needed a reintroduction to views, and E5 commented that the framework has a relatively high learning curve for analysts who are not in the field of graph mining. E5 suggested adding information panels for each view may, and we have updated the system to incorporate this. Each view now contains a small question mark that provides a description of the view on mouse-over.

Scalability: Other limitations include the scalability with respect to computational complexity, visual elements and color encoding.

Computational Complexity: Our framework utilizes pre-computed data to show the analytical results. The pre-processing time varies as the data is computed by different ranking models and the time complexity depends on the linear system solver. In the usage scenarios described in this paper, pre-processing took 3 minutes for the Weibo dataset and 4 minutes for the Facebook dataset. Although the framework is able to support larger-sized data, we chose to subsample all data in our usage scenarios to be compatible with the limited memory configurations of a generic desktop. Another computational bottleneck occurs in the clustering algorithm (Algorithm 1) applied in our Rank Mapping View (Figure 4). The overall complexity of Algorithm 1 is $O(n^3)$, where n is the number of selected nodes. Although the clustering algorithm is only applied to the selected data, the performance will suffer if the number of selected nodes becomes large. However, most ranking audits are primarily concerned with a relatively small number of the top-k ranks as beyond a certain k, nodes typically remain unexplored in practice.

Number of Visible Elements: Since the analyst can define the range of ranking scores to audit, this could result in hundreds of nodes being selected. Although we provide a cluster-level abstraction with Algorithm 1, it could still result in an extremely long list that exceeds the canvas size of the rank mapping view. A possible solution is to further aggregate the nodes in the same cluster into a glyph. A similar issue occurs in the group creation as well, where the combination of sensitive attributes used to define a group could result in hundreds of groups. This would ultimately affect the rank mapping view where too many groups segment the axes into many tiny pieces and cause visual clutter. Interactive filtering on the attribute axes can be adopted to temporarily remove the inessential value ranges. Given that most ranking results on the web show a top-10 or top-20 group, our current design seems reasonable for auditing fairness within the top-ranked elements.

Color Encoding: The categorical color encoding is shared between all the views in our framework to represent different groups. One issue is that due to the limit of available colors in the color scheme, the maximum number of displayed groups may not exceed 10. However, for groups, as the number of sensitive attributes chosen expands, the number of nodes that belong to a specific group becomes very small, and issues of fairness at this level may be artifacts of under-representation in the data. After discussing with our experts, general practice is to start with one sensitive attribute (e.g., gender), explore issues of fairness. Switch to another sensitive attribute (e.g., ethnicity), and then explore the combination of these two attributes. Our experts greatly appreciated the ability of our framework to support a multi-class definition of fairness. They did note that the number of groups being audited could quickly become unwieldy; however, they felt this design likely would fall into the 80% solution category, where the majority of fairness definitions would not be covering hundreds of protected groups. One feasible way to reduce the number of visible groups is to provide an extra list for preliminary group filtering and selection.

6 Conclusions

In this work, we propose a visual analytics framework for exploring and diagnosing algorithmic fairness in graph mining models. The visualization components of the framework are implemented with D3.js [5]. The backend is supported by the NetworkX library [3] and Python Flask[4]. The source code is currently available on Github[5]. In the future, we plan to extend our framework to reveal potential fairness issues in other types of graph mining models, such as graph embedding, clustering, and classification. We also plan to support the comparisons of ranking results between the base model and more than one target model to facilitate fairness-oriented model selection.

Acknowledgments

This work was supported by the U.S. Department of Homeland Security under Grant Award 2017-ST-061-QA0001 and 17STQAC00001-03-03, and the National Science Foundation Program on Fairness in AI in collaboration with Amazon under award No. 1939725. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

[3] https://networkx.org/

[4] https://palletsprojects.com/p/flask/

[5] https://github.com/VADERASU/fairrankvis

REFERENCES

[1] UCI machine learning repository: microblogPCU data set. http://archive.ics.uci.edu/ml/datasets/microblogpcu. (Accessed on 03/11/2021).

[2] Y. Ahn and Y. R. Lin. Fairsight: Visual analytics for fairness in decision making. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1086–1095, 2020. doi: 10.1109/TVCG.2019.2934262

[3] R. Binns. On the apparent conflict between individual and group fairness. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 514–524, 2020.

[4] A. Bose and W. Hamilton. Compositional fairness constraints for graph embeddings. In *Proceedings of the International Conference on Machine Learning*, vol. 97, pp. 715–724, 2019.

[5] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[6] A. Bougouin, F. Boudin, and B. Daille. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the International Joint Conference on Natural Language Processing*, pp. 543–551, 2013.

[7] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. FAIRVIS: Visual analytics for discovering intersectional bias in machine learning. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 46–56, 2019.

[8] T. P. Chartier, E. Kreutzer, A. N. Langville, and K. E. Pedings. Sensitivity and stability of ranking vectors. *Society for Industrial and Applied Mathematics Journal on Scientific Computing*, 33(3):1077–1102, 2011.

[9] J. J. Crofts and D. J. Higham. Googling the brain: Discovering hierarchical and asymmetric network structures, with applications in neuroscience. *Internet Mathematics*, 7(4):233–254, 2011.

[10] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the Innovations in Theoretical Computer Science Conference*, pp. 214–226, 2012.

[11] S. A. Friedler, C. Scheidegger, and S. Venkatasubramanian. The (Im)Possibility of fairness: Different value systems require different mechanisms for fair decision making. *Communications of the ACM*, 64(4):136–143, 2021.

[12] D. F. Gleich. Pagerank beyond the web. *Society for Industrial and Applied Mathematics Review*, 57(3):321–363, 2015.

[13] M. Gori and A. Pucci. Itemrank: A random-walk based scoring algorithm for recommender engines. In *Proceedings of the International Joint Conference on Artifical Intelligence*, pp. 2766–2771, 2007.

[14] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 3323–3331, 2016.

[15] J. Heidemann, M. Klier, and F. Probst. Identifying key users in online social networks: A pagerank based approach. In *Proceedings of the International Conference on Information Systems*, p. 79, 01 2010.

[16] C.-C. Hsu, Y.-A. Lai, W.-H. Chen, M.-H. Feng, and S.-D. Lin. Unsupervised ranking using graph structures and node attributes. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 771–779, 2017.

[17] T. Kamishima, S. Akaho, and H. Asoh. Enhancement of the neutrality in recommendation. In *Proceedings of the 2nd Workshop on Human Decision Making in Recommender Systems*, pp. 8–14, 2012.

[18] J. Kang, J. He, R. Maciejewski, and H. Tong. InFoRM: Individual fairness on graph mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 379–389, 2020.

[19] N. Kilbertus, M. Rojas-Carulla, G. Parascandolo, M. Hardt, D. Janzing, and B. Schölkopf. Avoiding discrimination through causal reasoning. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 656–666, 2017.

[20] J. Kleinberg, S. Mullainathan, and M. Raghavan. Inherent trade-offs in the fair determination of risk scores. arXiv:1609.05807, 2016.

[21] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern. Guarantees for spectral clustering with fairness constraints. In *Proceedings of the International Conference on Machine Learning*, vol. 97, pp. 3458–3467, 2019.

[22] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 539–547, 2012.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[24] D. Pedreschi, S. Ruggieri, and F. Turini. Measuring discrimination in socially-sensitive decision records. In *Proceedings of the SIAM International Conference on Data Mining*, pp. 581–592, 2009.

[25] D. Pedreshi, S. Ruggieri, and F. Turini. Discrimination-aware data mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 560–568, 2008.

[26] E. Pitoura, P. Tsaparas, G. Flouris, I. Fundulaki, P. Papadakos, S. Abiteboul, and G. Weikum. On measuring bias in online information. *Special Interest Group on Management of Data Record*, 46(4):16–21, Feb. 2018.

[27] S. Prince. Tutorial #1: bias and fairness in ai. https://www.borealisai.com/en/blog/tutorial1-bias-and-fairness-ai. (Accessed on 03/19/2021).

[28] T. Rahman, B. Surma, M. Backes, and Y. Zhang. Fairwalk: Towards fair graph embedding. In *Proceedings of the International Joint Conferences on Artifical Intelligence*, pp. 3289–3295, 2019.

[29] R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Proceedings of the Annual International Conference on Research in Computational Molecular Biology*, pp. 16–31, 2007.

[30] T. Shr, S. Hilgard, and H. Lakkaraju. Does fair ranking improve minority outcomes? understanding the interplay of human and algorithmic biases in online hiring. arXiv:2012.00423, 2020.

[31] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, and N. Mamoulis. Fairness-aware pagerank. arXiv:2005.14431, 2020.

[32] Q. Wang, Z. Xu, Z. Chen, Y. Wang, S. Liu, and H. Qu. Visual analysis of discrimination in machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1470–1480, 2021.

[33] J. Weng, E.-P. Lim, J. Jiang, and Q. He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pp. 261–270, 2010.

[34] T. Xie, Y. Ma, H. Tong, M. T. Thai, and R. Maciejewski. Auditing the sensitivity of graph-based ranking with visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1459–1469, 2021.

[35] K. Yang and J. Stoyanovich. Measuring fairness in ranked outputs. In *Proceedings of the International Conference on Scientific and Statistical Database Management*, 2017.

[36] S. Yao and B. Huang. Beyond parity: Fairness objectives for collaborative filtering. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 2925–2934, 2017.

[37] M. Zehlike, F. Bonchi, C. Castillo, S. Hajian, M. Megahed, and R. Baeza-Yates. FA*IR: A fair top-k ranking algorithm. In *Proceedings of the ACM on Conference on Information and Knowledge Management*, pp. 1569–1578, 2017.