Addressing User Activity Biases in Bipartite Graph Ranking using BiRank Algorithm

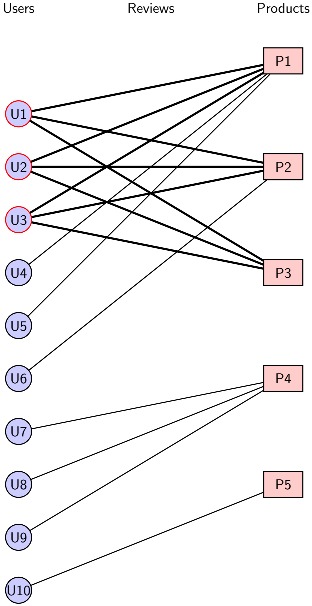
Layan Kaushik & Trent Tubyfill

17th December 2023

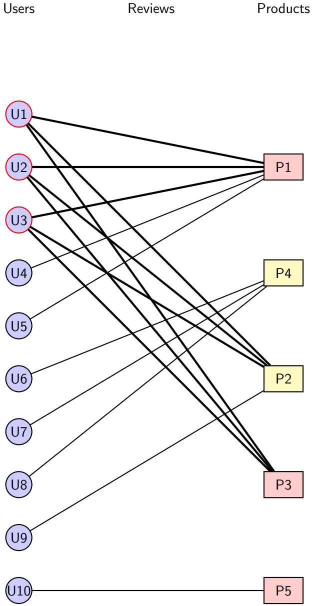
User Activity Bias

- **User Activity Bias**: Distortion that can occur in data analysis and model outcomes due to the disproportionate influence of highly active users
- A small subset of users tends to contribute a large portion of the content or interactions, which can skew analysis and model predictions if not accounted for:
  1. Product Reviews: hyperactive reviewers disproportionately influence the overall ratings of products, which may not be representative of the general user base.
  2. Social Media: a minority of users might post frequently, which could lead to an overestimation of certain opinions or topics.
  3. Content Creation Platforms: a small number of content creators might produce a large volume of the content consumed, which can bias recommendation algorithms towards their content.
- Regularization can help account for this bias.

User Activity Bias Example



User Activity Bias Example



Introducing BiRank for Bipartite Graphs [1]

- BiRank: A prominent algorithm for ranking nodes in bipartite graphs, commonly used for recommendation systems, search engines, and social network analysis.
- BiRank is an iterative algorithm that computes the steady-state probability distribution of a random walker navigating the bipartite graph.
- This approach inherently ranks nodes based on their connectivity and the connectivity of their adjacent nodes.
- Given a bipartite graph $G(U, V, E)$ with an adjacency matrix W, where W_{ij} represents the weight of the edge between node i in U and node j in V.
- The Normalized Adjacency Matrix $W' = D_U^{-1/2} W D_V^{-1/2}$ is used, where D_U and D_V are diagonal matrices containing the degrees of nodes in U and V, respectively.
- Normalization is crucial as it balances the influence of nodes with varying degrees.

BiRank for Bipartite Graphs

- Iterative Update Rules:
  - For nodes in U:

$$u^{(k+1)} = \alpha W' v^{(k)} + (1-\alpha) q_U$$

  - For nodes in V:

$$v^{(k+1)} = \alpha (W')^T u^{(k+1)} + (1-\alpha) q_V$$

- Variables:
  - α: Damping factor, controlling the balance between structure-driven and query-driven ranking.
  - q_U, q_V: Query vectors, representing prior knowledge or inherent node importance.
  - u, v: Rank vectors, iteratively updated to represent node importance in sets U and V.
- Convergence: The process iterates until the change in rank vectors falls below a defined tolerance level.

Limitation of BiRank: User Activity Bias

- While BiRank effectively ranks nodes, it does not inherently account for user activity bias.
- Some users (nodes in U) are more active than others, creating a skew in the representation and influence within the graph.
- Our Contribution: We extend BiRank by incorporating a regularization mechanism to mitigate this bias, ensuring a more balanced and representative ranking across the network.

BiRank with User Activity Regularization

- Addressing user activity bias in bipartite graphs using regularization.
- Regularization Vector r defined as $r_i = \gamma \frac{1}{\text{user\_activity}[i]+\epsilon}$.
  - Penalizes nodes based on activity level, balancing ranking.
  - γ controls regularization strength; ϵ avoids division by zero.
- User activity measure: $\text{user\_activity}[i] = \frac{\text{degree of user } i}{\sum_j \text{degree of user } j}$.
  - Relative activity measure ensures balanced view of user influence.
  - Effectively reduces bias by penalizing hyperactive users.

BiRank with User Activity Regularization

- Modified Update Rules with Regularization [2]:
  - For nodes in U:

$$u^{(k+1)} = \alpha W' v^{(k)} \odot r + (1-\alpha) q_U$$

  - For nodes in V:

$$v^{(k+1)} = \alpha (W')^T u^{(k+1)} + (1-\alpha) q_V$$

- Impact: Regularization diminishes the ranking influence of hyperactive users for a more balanced network representation.
- Application: Enhances recommendation systems by mitigating bias from varied user interaction levels.

Fairness Metric

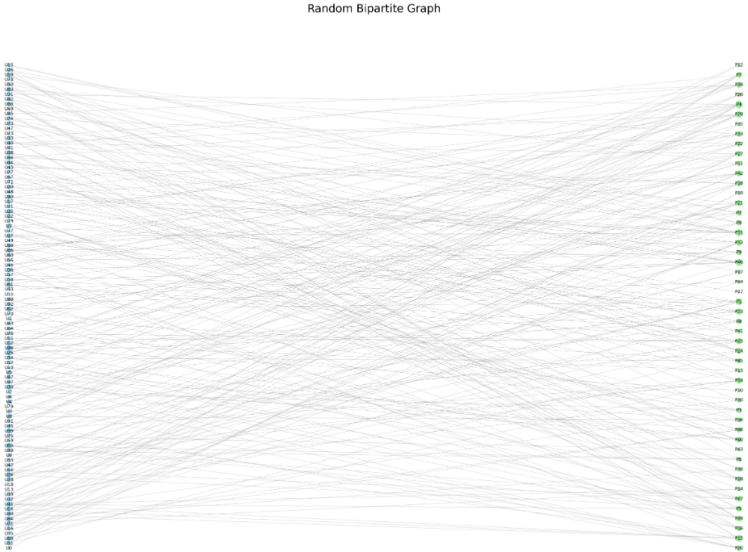
Objective: Balance minimizing disparity in activity ranks with maintaining a fair Gini index. (Smaller the fairer)
Formula:

$$\text{Fairness Metric} = \frac{1}{\left(\frac{1-w}{|\text{Disparity Ratio}-1|+\epsilon} + \frac{w}{\text{Gini Coefficient}+\epsilon}\right)}$$

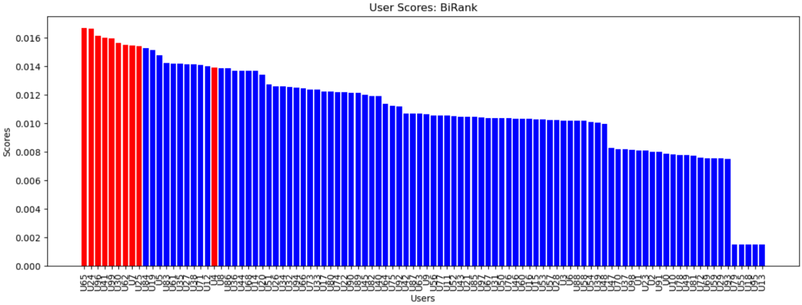
$$\text{where Disparity Ratio} = \frac{\textit{High Activity Average Rank}}{\textit{Low Activity Average Rank} + \epsilon}$$

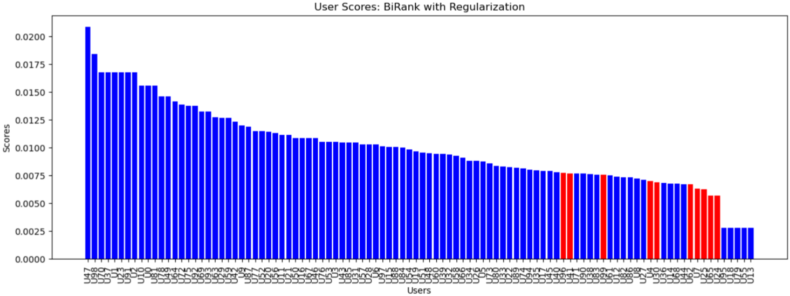
- High Activity Avg Rank is the mean rank of users above a specified activity threshold, while Low Activity Avg Rank is for those at or below it. The Gini Coefficient is calculated for the user score distribution.
- The Fairness Metric penalizes deviations from ideal Disparity Ratios (close to 1) and high Gini Coefficients (indicating inequality), aiming for a more equitable system.
- This metric emphasizes the importance of both minimizing disparity among user activity levels and ensuring an equitable distribution of ranks, with a lower metric value indicating greater fairness.

Random Bipartite Graph (100 Users, 50 Products)

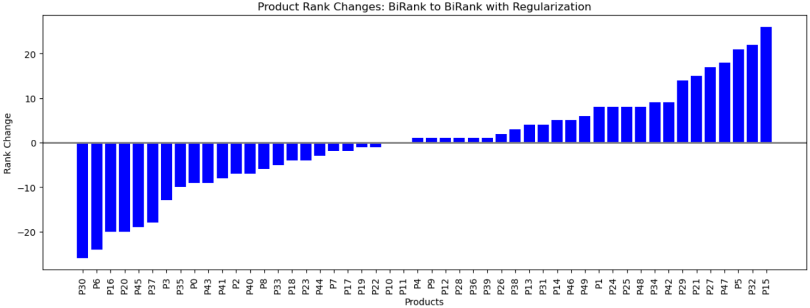


User Scores for Random

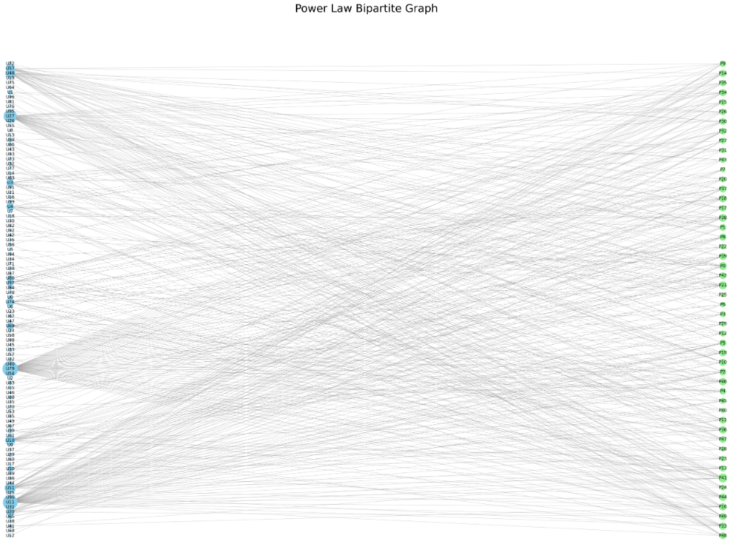




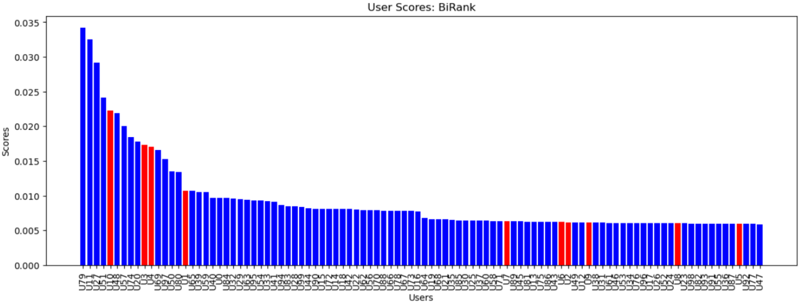
Product Rank Changes for Random

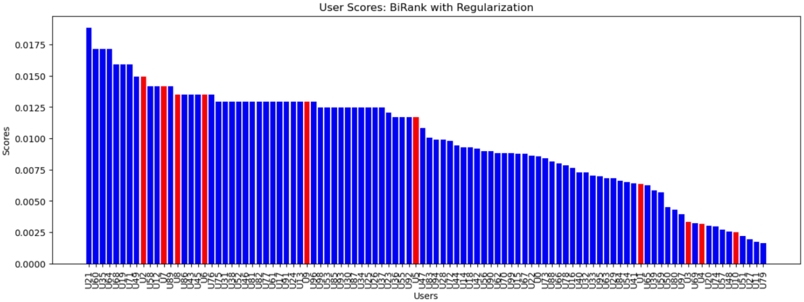


Power-Law Bipartite Graph (100 Users, 50 Products)

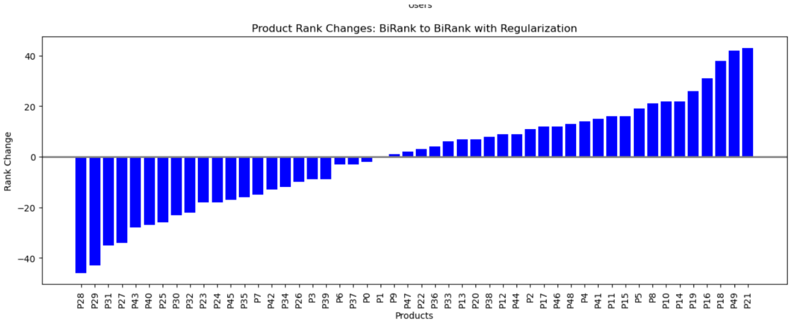


User Scores for Power Law





Product Rank Changes for Power-Law



Results (Average Fairness Metric Values)

Graph Type / (#Users, #Products)	BiRank	BiRank with Reg
Random Graph (100,50)	0.508	0.857
Random Graph (1000,100)	0.404	0.701
Random Graph (10000,500)	0.421	0.711
Random Graph (100,500)	0.722	1.211
Power-Law Graph (100,50)	0.906	0.604
Power-Law Graph (1000,100)	0.747	0.711
Power-Law Graph (10000,500)	0.701	0.724
Power-Law Graph (100,500)	1.241	1.259

Table: Fairness Metrics for BiRank and BiRank with Regularization

References

[1] X. He, M. Gao, M.-Y. Kan, and D. Wang. Birank: Towards ranking on bipartite graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):57–71, 2017.

[2] K.C. Yang, B. Aronson, and Y.Y. Ahn. Birank: Fast and flexible ranking on bipartite networks with r and python. *J Open Source Softw*, 5(51):2315, 2020.

Appendix

Hyperparameter Combinations used to calculate the results done for 4 (User, Product) combinations.

Combination	Gamma (γ)	Weight (w)	Exponent	Graph Type
1	0.1	0.5	N/A	Random
2	0.2	0.6	N/A	Random
3	0.3	0.4	N/A	Random
4	0.4	0.3	N/A	Random
5	0.5	0.5	N/A	Random
6	0.1	0.5	1.5	Power-Law
7	0.2	0.6	2.0	Power-Law
8	0.3	0.4	2.5	Power-Law
9	0.4	0.3	1.8	Power-Law
10	0.5	0.5	1.8	Power-Law

Table: Different Combinations of Hyperparameters for Random and Power-Law Graphs with Regularization