Web3Recommend

Decentralised Web3 social recommendations with trust and relevance balance

Rohan Madhwal
Delft University of Technology
Delft, The Netherlands
R.Madhwal@student.tudelft.nl

Johan Pouwelse
Delft University of Technology
Delft, The Netherlands
J.A.Pouwelse@tudelft.nl

***Abstract*—Social Recommender Systems have emerged as a solution to the "information overload" problem on the Internet where the proliferation of massive amounts of data makes it increasingly hard for users to find content that they like or find useful. Almost all social media platforms in the modern age rely on such a system for their popularity by ensuring that users remain engaged with the platform. These platforms conventionally rely on centralized infrastructure to create these systems. Creating a Social Recommender System for Web3 platforms on the other hand is a non-trivial problem because of the lack of trust amongst users in decentralized networks. Notably, sybil and spam attacks allow malicious users to unfairly manipulate the ranking of items, leading to low quality, untrustworthy recommendations being generated. In this paper, we present a novel decentralized Recommender System that balances trust and relevance in its recommendations by combining GraphJet, a graph-based system for generating content recommendation in Twitter with decay parameters and trusted exploration techniques from MeritRank, a decentralized reputation scheme. Our systems ranks items using a Monte Carlo estimation method which involves random walks performed in a Sybil resistant manner ensuring that non-trustworthy items receive a low ranking while also ensuring that recommendations are personalized and relevant to users. In order to quantifiably demonstrate the trust-relevance balance in our system, we conducted experiments on our system using data from Kaggle's "Million Song Dataset" and showed that with the right hyper-parameters our system can effectively remove sybils/spam from top recommendations while not greatly harming the relevancy of non-malicious recommendations. Further, we show the feasibility of using the system in edge devices acting as decentralized nodes by demonstrating scalability of the system to over a million users at a relatively low resource cost which is achieved through novel compact serialization and edge gossiping mechanisms.**

I. Introduction

The cardinal objective of a social media platform that aims to be successful and vibrant is an active and engaged user base. Achieving user engagement boils down to presenting the most attractive and relevant content to each user. However, popularity and success is a double edged sword since the abundance of users and content on these platforms floods users with huge amounts of information and hence poses a great challenge in terms of information overload. While search capabilities slightly alleviate the problem, often users are unable to express keywords that convey requirements about the type of content they would be interested in. Further, users tend to have diverse taste and the quality of content may be subjective depending on the user searching for it, therefore, beyond simple searching capabilities, personalisation is also required to make the content attractive and relevant to each user.

A Social Recommender System is an intelligent system that filters the massive amounts of information on social media platforms and recommends useful items and information to users based on their personalised needs which are inferred through unique explicit and implicit interactions within the social network. In this way, Social Networks and their Recommender Systems tend to have a symbiotic relationship since the quality of recommendations catered to users allows the networks to grow in size, which in turn provides more interactions, allowing higher quality recommendations.

Decentralized networks, unlike centralized networks, lack a central authority to maintain and manage user data, making it challenging to establish trust amongst users. The absence of a central authority and the distribution of power amongst the network's participants result in a lack of consensus, making it difficult to achieve a reliable system that is resistant to malicious attacks.

Recommender systems in centralized networks are relatively more secure from Sybil attacks due to the centralized nature of the system. Centralized systems often require user authentication and verification, making it more difficult for malicious actors to create multiple fake accounts to manipulate the ranking of items. Additionally, centralized systems have the advantage of being able to monitor user behavior and detect anomalies such as unusual activity patterns or highly repetitive actions, which could indicate the presence of a Sybil attack. This detection is possible because of the large number of skilled attendants dedicated to maintaining and improving system capabilities in centralized systems. These attendants can develop measures to prevent Sybil attacks, such as restricting the number of user actions that can be performed within a certain time frame, or introducing identity verification requirements.

In contrast, creating a trustable and reliable decentralized recommender system for Web3 platforms is a challenging task due to the lack of centralized infrastructure, making it easier for malicious users to create and control multiple identities, manipulate the ranking of items, and compromise the trustworthiness of recommendations. Therefore, the creation of decentralized recommender systems requires new approaches that can address the challenges of decentralized networks, including sybil attacks, limited resources, and lack of trust among users.

To tackle these challenges, we propose a novel decentralized recommender system that leverages a combination of GraphJet, a graph-based recommendation system, and MeritRank, a decentralized reputation scheme, to generate personalized and relevant recommendations while addressing the challenges of sybil attacks and limited resources. Our proposed system employs a Monte Carlo estimation method that involves random walks performed in a sybil-resistant manner, ensuring that non-trustworthy items receive a low ranking while personalized and relevant recommendations are generated for users. To demonstrate the effectiveness of our approach, we conduct experiments on our system using the Million Song Dataset from Kaggle, where we show that our system can effectively remove sybils and spam from top recommendations while maintaining the relevancy of non-malicious recommendations.

Our system also addresses the challenge of resource limitation by demonstrating its scalability to over a million users at a relatively low resource cost, achieved through novel compact serialization and edge gossiping mechanisms. We show that our system is feasible for use in edge devices acting as decentralized nodes, making it possible to create reliable and efficient recommender systems for Web3 platforms that can provide personalized and relevant recommendations to users while maintaining trust and addressing the challenges of sybil attacks and limited resources.

II. Background

The system presented in this paper relies on the (incremental) computation of personalized PageRank and SALSA augmented with principles from MeritRank. We also build on top of the GraphJet recommender system by Twitter. In this section, we provide a quick review of these methods.

A. PageRank

One of the most widely known ranking systems in the world is Google's PageRank which is still used (along with other algorithms) in order to rank websites for user queries on Google. PageRank determines a rough estimate of the relative importance of a website by computing a ranking for every web page. The underlying assumption of PageRank is that a website that is more important is more likely to receive links from other websites than a website that is less important.

The calculation of PageRank of a website can be simplified to the below equation:

$$\sum \frac{PageRank\ of\ Inbound\ Link}{Number\ of\ Outgoing\ Links\ on\ that\ Page} \tag{1}$$

PageRank can also be viewed as the stationary distribution of a random walk in a graph of websites connected by hyperlinks, which at each step with probability ϵ jumps to a random node and with probability $1 - \epsilon$ jumps to a randomly chosen node through an edge from the current node. **Personalized Page Rank**

Despite PageRank being a well studied problem, making some simple assumptions on the structure of data layout and network leads to a dramatic improvement in its runtime using the Monte Carlo estimation technique.

B. SALSA

C. GraphJet

D. Sybil Attacks

E. MeritRank

F. Distributed Network and Web3

III. System Design

Web3Recommend is a Recommender System designed to provide recommendations in any application running on a distributed network. Hence, we use a peer to peer architecture which assumes each user operates their own node and interacts with items being shared inside the network. The central data structure in the network is the **TrustNetwork** graph which stores information about node to node and node to item relationships across the entire network. Recommendations in the system are generated by performing random walks inside this network. Each node maintains a personal copy of a TrustNetwork and updates to the network are synchronized through a timestamp biased edge gossiping mechanism which ensures that recommendations are based on recent, global information inside the network. The system design also includes a simple bootstrapping mechanism which allows new users to find similar users in the network, however, it is worth noting that this bootstrap mechanism can be exploited by malicious users and in a real application, we assume that users are able to bootstrap through social discovery of trusted peers or through the provision of trustworthy nodes by the application itself. The following is an in-depth description of the various components of the system:

A. TrustNetwork

B. Recommendation Algorithm

C. Timestamp Biased Edge Gossiping

D. Bootstrap

- Graph Storage: High Level Design consists of each node storing the entire network's state in the form of two graphs: a node to node network which describes trust relationships between nodes in the network and a node to song network which describes each node's song preferences calculated as a proportion of the song's play count. These graphs are then synchronized between nodes using a gossiping mechanism. - Baseline Recommendation System: Influenced by GraphJet, runs on the node to song network in the form of a personalized SALSA, create illustration to show how it works, songs that have the most song visits receive highest ranking - Modified Trustworthy Recommendation System: Exploration probability added to the random walk, beta decays from merit rank - Custom Compact Serialization - Gossiping Mechanism - Bootstrap

IV. Dataset and Experiments

- Background about what we need to prove, how the dataset was constructed - Large Sybil Attack - Similarity

V. Performance Analysis

VI. Conclusion

VII. Future Work

References