SITIOS PREFERENCE BASED CARPOOLING,

*Aswin V T, *Aswin Manoj, *Albin Abraham George, ** Anjana J Mani

*Computer Science and Engineering, Mangalam College of Engineering, Kottayam, Kerala

** Computer Science and Engineering, Assistant Professor, Mangalam College of Engineering, Kottayam, Kerala

Abstract— Carpooling apps are gaining popularity as people seek cost-effective and environmentally-friendly ways to commute. However, several issues continue to hinder the effectiveness of these apps. Lack of trust between users is a primary issue, which can be addressed through implementing verification and rating systems. Inefficient matching algorithms can lead to long wait times for riders or empty seats for drivers, which can be addressed through the use of advanced algorithms. Limited coverage is another issue that can be addressed through incentivizing users to join and expanding the coverage area. Payment disputes can also arise between riders and drivers, and secure payment processing systems are necessary to handle these issues effectively. Safety concerns are critical in carpooling apps, and effective safety features, such as emergency buttons, driver background checks, and real-time tracking, must be implemented. Addressing these issues can enhance the effectiveness and popularity of carpooling apps.

Keywords—Pooling, attributed network representation, and personalized ranking

I. INTRODUCTION

User embedding is a machine learning and natural language processing approach for representing users in a high-dimensional vector space. User embedding's purpose is to build a numerical representation of users that incorporates their interests, preferences, behaviors, and other important attributes. There are several techniques to user embedding, but one of the most common relies on neural networks. A neural network is trained on a huge corpus of user-generated content, such as social media postings, reviews, or comments, in this technique. The neural network learns to map each user to a fixed-length vector containing the user's latent attributes.

Once collected, the user embedding may be utilized for a range of activities such as personalized recommendations, user categorization, and clustering. User embedding is especially effective when the user profile is insufficient or sparse, since it enables more precise and economical modelling of user behaviour[1].

The article proposes a user embedding approach based on deep learning for personalized ridesharing matching. The personalized matching is a key challenge in carpooling and ridesharing, as it involves finding the best matches between drivers and passengers based on their preferences, schedules, and locations. To address this challenge, a neural network model that learns user embeddings from historical ridesharing data. The user embeddings are then used to measure the similarity between drivers and passengers, and to make personalized matching recommendations.

The article provides a detailed description of the proposed model, along with experimental results that demonstrate its effectiveness in improving the quality of ridesharing matching. The potential applications of user embedding in other areas of transportation, such as traffic prediction, route planning, and vehicle routing.

Overall, the article provides a valuable contribution to the field of transportation and user embedding, and it showcases the potential of deep learning techniques for improving the efficiency and sustainability of ridesharing services.

II. EFFICIENT CARPOOLING MATCHING WITH ATTRIBUTE-BASED NETWORK EMBEDDING

A. Background

Attributed network embedding is a technique used to represent users in a network with both their social connections and attributes. It involves mapping each user to a low-dimensional vector, which captures their social relationships and individual attributes.

B. System Model

The system offers two critical phases for the attributed network embedding approach. First, a graph is created to represent user relationships based on their carpooling history and social connections. Then, using both social connections and individual attributes, an embedding model is trained to learn a low-dimensional representation of each user in the graph.

For this, the experts suggested a new approach called Social and Attribute Graph Convolutional Networks (SA-GCN), which combines graph topology and user characteristics into a neural network model. The SA-GCN approach employs many layers of graph convolutional neural networks (GCN) that operate on the graph, with each layer pooling information from the preceding layer as well as user data to build the embedding representation.

III. SA-GCN: A System for Personalized Carpooling Matching using Attribute-Based Graph Convolutional Network Embedding

To develop personalized carpooling embeddings, the system employs an attribute-based graph convolutional network (GCN) embedding methodology. To match carpooling requests and offers in a personalized manner, the system considers both social network and user characteristic information. The system's goal is to increase the efficiency and sustainability of urban transportation networks by encouraging more people to carpool.

The experts suggested a new approach called Social and Attribute Graph Convolutional Networks (SA-GCN), which combines graph topology and user characteristics into a neural network model. The SA-GCN approach employs many layers of graph convolutional neural networks (GCN) that operate on the graph, with each layer pooling information from the preceding layer as well as user data to build the embedding representation.

The resulting embeddings capture both the social relationships and individual attributes of each user, allowing for a more accurate and personalized ranking of carpooling matches. The effectiveness of their attributed network embedding method on a real-world carpooling dataset and demonstrated that it outperformed several baseline methods in terms of personalized ranking accuracy.

A. Node2vec Vs SA-GCN

Node2vec is a common network embedding approach that produces low-dimensional representations of nodes based on their local neighborhood structure. It employs a random walk technique to sample nodes in the network and develop representations that reflect both the network's local and global structure. Node properties may also be incorporated into the embedding by including them as extra features in the learning process. Whereas SA-GCN is an extension of graph convolutional networks (GCN) that incorporates both social network information and user attributes into the node embeddings.

B. Modules

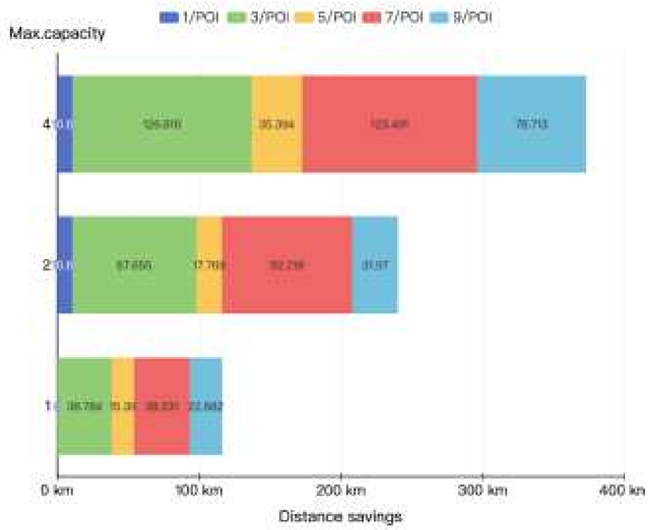
- Graph construction: The first step is to construct a graph that represents the carpooling network, where nodes represent users and edges represent previous carpooling trips between users. A threshold-based approach is used to determine which trips to include in the graph based on the similarity of trip destinations and departure times.
- User attribute representation: Each user is represented as a vector of attribute values, such as age, gender, and occupation. These attribute vectors are concatenated with the node embeddings in the GCN model.
- Social and attribute graph convolution: A new graph convolutional operation that takes into account both the graph structure and the user attributes. The convolutional operation is applied to each node's neighborhood, where the neighborhood consists of the node's direct neighbors and their attributes. The output of the convolutional operation is a new node embedding that captures both the social network information and the user attributes.
- Personalized ranking: The final step is to use the learned node embeddings to rank potential carpooling matches for each user based on their similarities in both social network structure and user attributes.

C. Experimental Result

The researchers tested the SA-GCN approach against multiple state-of-the-art norms on a real-world carpooling dataset. They assessed the personalized carpooling matching's performance in terms of accuracy, recall, and F1-score for each user. The SA-GCN technique exceeded all standards, demonstrating its efficacy in learning personalized carpooling embeddings.

- Impact of Pooling with Multiple Riders

Fig1 shows When the number of drivers and riders in the system is roughly equal, it is preferable to have single driver-multiple riders matches to maximize vehicle utilization. When 4 riders per vehicle are allowed, the number of vehicles served in matches with one rider is quite small in different vehicular distributions. This is due to an increase in the number of chances for single driver-multiple rider matching. That is, if it is possible to match four riders with the same driver, but it is also possible to match the four riders with different drivers, the former option is preferred because it results in only one driver.



- Impact of Duration Flexibility

By allowing users to select their preferred amount of flexibility in terms of journey time and destination, the proposed algorithm attempts to enhance the flexibility of carpooling. The technology then finds possible carpooling mates with comparable interests and characteristics and generates a personalized score of the best matches for each user.

- By selecting matching travel partners who can accommodate each other's tastes and demands, the personalized ranking system may be viewed as a technique to boost the flexibility of carpooling. The technology can assist users locate carpooling companions who are more flexible and responsive to their specific travel needs by pairing individuals with comparable interests and qualities.

IV. PROPOSED SYSTEM

React Native is a popular open-source mobile app development framework that enables developers to create high-performance, native mobile apps for both the Android and iOS platforms by combining JavaScript with React.

Carpooling is a common practice in which individuals share a ride with others who are travelling to the same location. React Native may be used to create a carpooling app that connects users with others who are travelling to the same location React Native provides a robust framework for building carpooling apps that can help reduce traffic congestion, save money, and provide an eco-friendlier transportation option.

. The component of React Native are:

- User Interface: React Native provides a rich set of pre-built UI components that can be used to build a carpooling app. These components are optimized for mobile devices, making it easier to create a seamless user experience. Some of the UI components that can be used in a carpooling app include buttons, text inputs, dropdowns, and maps.
- Real-time tracking: React Native has built-in support for real-time updates, which can be used to track the location of the user and other carpooling participants. This can be used to provide real-time updates on the location of the car, estimated arrival times, and other important information.
- Payment integration: Carpooling apps typically require payment integration to facilitate payment between the users. React Native provides support for payment integration through third-party libraries like Stripe, which can be used to securely process payments between users.
- Push Notifications: React Native provides support for push notifications, which can be used to send alerts to users about new ride requests, ride confirmations, and other important updates.
- Social Integration: Carpooling apps can be integrated with social media platforms like Facebook and Twitter to allow users to easily share ride requests with their social networks.

```
function maxWeightedBipartiteMatching(graph):
  // Initialize an empty matching
  matching = {}

  // Iterate over all drivers in the graph
  for each driver in graph.drivers:
    // Find the rider with the highest compatibility score
    best_rider = None
    best_score = -inf
    for each rider in graph.riders:
      if compatibilityScore(driver, rider) > best_score:
        best_rider = rider
        best_score = compatibilityScore(driver, rider)

    // Add the driver-rider pair to the matching
    matching[driver] = best_rider

  // Return the matching
  return matching
```

Preference-based carpooling involves matching riders and drivers based on their individual preferences, such as pick-up and drop-off locations, preferred travel times, and any other relevant criteria. The proposed algorithm is maximum weighted bipartite matching algorithm.

Create a bipartite graph with two sets of vertices: drivers and riders. Each driver and rider is represented by a vertex in their respective sets. For each driver, create a weighted edge to each rider, with the weight of the edge representing the compatibility score between the driver and rider (based on their preferences).

Use the maximum weighted bipartite matching algorithm to find the optimal matching between drivers and riders. This algorithm finds the combination of matches that maximizes the total weight of the edges.

Once the matching is complete, assign each rider to the driver they are matched with and generate a route for the driver to pick up and drop of fall riders. This algorithm can be further refined with additional constraints, such as limiting the maximum detour time for the driver or ensuring that each rider is matched with a driver who has a compatible vehicle type. Additionally, real-time adjustments can be made to the matching algorithm as new riders or drivers join the pool, or as preferences change over time.

CONCLUSION AND FUTURE SCOPE

Simulations have demonstrated that the recommended SA-GCN technique outperforms numerous state-of-the-art baselines. The approach suggested might be used to improve the efficiency and sustainability of urban transport networks. However, challenges remain, such as incentivizing users to participate in carpooling trips and ensuring the safety and comfort of the carpooling. Another avenue to pursue is the incorporation of additional data sources, such as user preferences and historical carpooling behavior, to increase personalized matching performance. There are additionally practical aspects to consider when adopting such personalized carpooling systems in real-world contexts. One significant problem is incentivizing people to engage in carpooling journeys, which may entail developing efficient incentive systems or giving discounts on tolls or parking costs. Another problem is ensuring the safety and comfort of carpooling passengers, which may need the creation of new safety regulations and monitoring systems.

ACKNOWLEDGEMENT

We would like to thank Principal Vinodh P Vijayan, Neethu Mariya John, H.O.D, Faculty of Computer Science, for their appropriate guidance, valuable assistance and helpful comments during the proofreading process.

REFERENCES

[1] Chao Chen, Jun Luo, Wei Wu, and Hui Xiong," User Embedding for Personalized Ridesharing Matching: A Deep Learning Approach",IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1457-1467, Apr. 2020.

[2] M. Stiglic, N. Agatz, M. Savelsbergh, and M. Gradisar, "Makingdynamic ride-sharing work: The impact of driver and rider flexibility,"Transp. Res. E, Logistics Transp. Rev., vol. 91, pp. 190–207, Jul.2016,doi:10.1016/j.tre.2016.04.010.

[3] A. K. M. M. R. Khan, O. Correa, E. Tanin, L. Kulik, andK. Ramamohanarao, "Ride-sharing is about agreeing on a destination,"in Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst., Redondo Beach, CA, USA, Nov. 2017, pp. 1–10.

[4] R. Duan and S. Pettie, "Linear-time approximation for maximum weight matching," J. ACM, vol. 61, no. 1, pp. 1–23, Jan. 2014, doi:10.1145/2529989.

[5] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, and Z. Gong, "An efficient ride-sharing framework for maximizing shared route," IEEE Trans Knowl. Data Eng., vol. 30, no. 2, pp. 219–233, Feb. 2018, doi:10.1109/TKDE.2017.2760880.