Splitting the Expenditure by Intermediator Drive Conveying

Praveen Kumar[1], Gokul Modi[2], Surendra Singh[3], Rohan Choksey[4], Bipin Chaudhary[5]

Department of Computer Science & Engineering

Parul University, Vadodara, Gujarat

Email: *Praveen.kumar270052@paruluniversity.ac.in[1], 170303105173@paruluniversity.ac.in[2], 170303105048@paruluniversity.ac.in[3], 170303105043@paruluniversity.ac.in[4], 170303105038@paruluniversity.ac.in[5]*

DOI:- https://doi.org/10.47531/SC.2022.20

Abstract

Ride-sharing (RS) has great value in saving energy and alleviating traffic pressure. Existing studies can be improved for better efficiency. Therefore, we propose a new ride-sharing model, where each driver has a requirement that if the driver shares a ride with a rider, the shared route percentage exceeds the expected rate of the driver. Customers are both price and delay-sensitive, and drivers are strategic and self-scheduling. We prove that optimizing the matching decisions have a first-order effect on the system performance. We show that fixing the matching decisions and optimizing only the pricing decisions does not generally maximize matchings.

Similarly, we show that fixing the pricing decisions and optimizing only the matching decisions is not optimal in general. Finally, we show that optimizing in only one dimension (either pricing or matching) has no benefit to the firm under some conditions. In contrast, joint pricing and matching optimization can lead to a significant performance increase.

Keywords: *- Vehicles, Roads, Real-time ride-sharing, Machine Learning, Recommender System, Dynamic Pricing.*

INTRODUCTION

In this paper, we study ride-sharing platforms such as Lyft, Sidecar, and Uber. Since their founding in the last several years, these platforms have experienced extraordinary growth. At their core, the platforms reduce the friction in matching and dispatch for transportation. A typical transaction on these platforms is as follows: a potential rider opens the app on her phone and requests a ride; the system matches her to a nearby driver if one is available, else blocks the ride request. These platforms typically do not employ drivers but deliver a share of the earnings per ride to the driver, to incentivize driver participation Ride-sharing platforms are thus two-sided markets: drivers on one side and passengers on the other. Consequently, a central goal of the platform's intermediation is to calibrate supply and demand

relative to each other while ensuring relatively high satisfaction to both sides. An essential tool used by these platforms to manage supply and demand is dynamic pricing – the platform can adjust ride prices in real time to react to changes in ride requests and available drivers. The central focus of this paper is in understanding how these two features of ride-sharing platforms, their two-sided nature and the ability to price based on real-time state – influence the volume of trade and the platform's revenue.

To capture the fast-timescale dynamics of ride-sharing platforms, we employ a queueing theoretic approach. Our primary modelling contribution lies in combining this queueing model for the underlying stochastic dynamics with an equilibrium analysis that captures drivers and passengers' incentives and throughput/revenue maximization by the platform.

The general model we consider is one where a geographic area is divided into regions. Each ride involves a driver picking up a passenger in one region and dropping her off in another. For simplicity, we analyze this model first for a single region; subsequently, using tools from classical queueing theory, we generalize some of our results to networks of regions. Key to our model is an intrinsic timescale separation in the behaviour of drivers and passengers. In ride-sharing platforms driven by mobile apps, passengers are typically sensitive to the immediate price of the ride they request. If this price is too high, price-sensitive customers will abandon it. On the other hand, drivers are typically sensitive to the average wages they earn over a longer period, usually several days or a week. Thus, drivers often select specific periods when they are "active" during the week and adjust their activity levels based on their earnings assessment during the last week.

Two key questions common to all ride-sharing firms concern pricing and matching. Pricing determines customer demand and driver supply because lower prices attract customers, and higher prices attract drivers. Meanwhile, matching connects a customer requesting a ride with a driver, determining how long the customer must wait for driver pick-up. Together, the pricing and matching decisions affect the geospatial distribution of the drivers at any given point in time. In turn, that geospatial distribution determines the set of available drivers to be matched with an arriving customer.

However, a customer offered a far-away driver may not accept the ride due to the long pick-up time. This paper aims to maximize the number of matched customers, meaning the number of customers who accept the offered rides, by jointly optimizing the pricing and matching decisions. Email address: erhozkan@ku.edu.tr There are many papers in the literature (e.g., Besbes, Castro, & Lobel, 2018; Bimpikis, Candogan, ˘ & Saban, 2019; Castillo, Knoepfle, & Weyl, 2016; Guda & Subramanian, 2019; Riquelme, Banerjee, & Johari, 2015) that optimize the pricing decisions under an assumed matching policy. The price is varied across locations to influence the locations of drivers waiting to be assigned to a customer. The matching policy is fixed and could, for example, offer an arriving customer the closest available driver. However, as the following example shows, ignoring matching optimization can result in subpar overall performance.

LITERATURE SURVEY

1. Popular Ride-Sharing Applications and Their Limitations

Our study began with an in-depth inspection of several famous Ride Sharing applications like UberPool, LyftLine, Juno, Curb, Wingz, Via, Flywheel, Zimride, and Waze. Some of the common limitations and reasons for disputes observed in all the applications are that drivers get to know the count of passengers at the pick-up point.

On many trips, the vehicle is occupied with only one passenger, which is entirely against the essence of Ride Sharing. Additional limitations include users' lack of basic details of other users they are travelling with, unfair pricing, and the sudden addition of riders, which adds a significant amount of time to complete trips due to distant locations.

Reference Test Specimens

Noting the limitations in applications, we have designed our model considering most of the discovered limitations. While matching, we first perform the exact match, which finds riders with exactly matching characteristics. If the pool is incomplete, we find riders with little different or closer characteristics. If the pool remains incomplete, we incorporate the current Uber or Lyft model of matching riders irrespective of characteristics [8]. Utilizing the three types of matching in the system ensures that we serve most broadcasting rider requests and complete the pool for a maximum number of trips. After having every passenger's details, we provide the trip itinerary to every rider, including the driver, before commencing the trip, which assists in reducing the social barriers among riders. The types of matching are illustrated in Table 1.

Tracking Rider Characteristics

The designed model allows the riders to provide feedback only to the users they have travelled before on trips. The method for tracking the data characteristic utilizes rider feedback records. For example, a user may constantly rate a high score of 4 or a low score of 0 to a specific characteristic for several trips, implying users are less interested in that specific characteristic. Our motive is to find the characteristics the user is most interested in and recommend riders based on the tracked characteristics. The methodology selected for tracking rider characteristics is variance and is demonstrated with the help of lists, L1 = [1, 0, 5, 4, 0], L2 = [0, 0, 0, 0, 2], and L3 = [4, 4, 4, 4, 4].

Table-1

Riders%	Chatty	Safety	Punctuality	Friendliness	Comfortability
Broadcasting Rider Characteristics	2	3	4	4	2
Search Riders Having Exact Characteristics	2	3	4	4	2
Search Riders Having Closer Characteristics	3(+1)	3	4	3(-1)	2
Search Riders of Irrespective of Characteristics	4	1	5	5	5

The total number of sample points in each list is given by N, and the mean is denoted by xi. The difference x distance of data-point x to the mean xi is computed by x − xi [12]. The general definition of the variance is the average of squared differences from the mean [12]. Variance indicates each sample point's spread level in a data set [12] and is given by Equation 1. The larger the variance of a data set, the higher is the data variety [12]. If the variance is applied to all three lists, the highest score is computed for L1 as the data variety in L2 and L3 is notably low [12]. If a similar methodology is applied for every characteristic feedback, the characteristic feedback with the highest variance is the rider's most focus.

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_{distance})^2}{N} = \frac{\sum_{i=1}^{N}(x - x_i)^2}{N}$$

Machine Learning Module Selection:-

After researching several methods for matching, we selected the Machine Learning Content-Based recommendation system. In this system, the features are converted to vectors and represented in d-dimensional space, where d is the number of features [3]. The angular distance or the cosine of the angle, θ between the vectors is calculated using the dot product equation [3, 13]. Vectors with the highest cosine values are deemed as the best match. We made a similar use where the features represented the registered rider characteristics, and riders with higher cosine similarities are paired up on a trip. The classification module we selected for classifier prediction is the Support Vector Machines (SVM) because of SVM's Radial Bias Function (RBF) Kernel. The RBF kernel is a highly non-linear curve that is used for distinguishing classes [5]. SVM works on the principle of placing the line to the closest data point with maximum distance. The line placement is changed through the regularization parameter, C, and the gamma parameter, γ [5]. The process of governing the curve placement is called Kernelization [5, 9]. The regularization allows a slight error of fitting a class, including fewer data points of other classes and is best suited for imbalanced data sets [5, 9].

PROPOSED MODEL

1. System Architecture:-

The critical elements of the proposed architecture are the broadcasting rider request, the closest driver, matching layers, the feedback system, and the Machine Learning module. Figure 1 reflects the system architecture. The broadcasting rider request consists of rider source, destination, and the rider user-id. The registered five characteristics are retrieved from the data server and are referenced throughout the trip while searching for more riders using the user-id. For the best simulation practices, we have utilized the New York Cab location database [14]. The NYC Cab locations are divided into 263 zones that are useful to avoid a larger search. The closest available driver is retrieved from the same source broadcasting rider's source zone using the Google Map Distance Matrix API. The key element in driver association is noting the vehicle seating capacity. The next step is the execution of the characteristics matching layer. All the broadcasting riders having Exact or Closer characteristics from the same source zone are retrieved. Riders go through the ML-based recommendation system and are trip while searching for more riders using the based recommendation system and are the user-id. For the best simulation practices, we have utilized the

New York Cab location database [14]. The NYC Cab locations are divided into 263 zones that are useful to avoid a larger search. The closest available driver is retrieved from the same source broadcasting rider's source zone using the Google Map Distance Matrix API. The key element in driver association is noting the vehicle seating capacity. The next step is the execution of the characteristics matching layer. All the broadcasting riders having Exact or Closer characteristics from the same source zone are retrieved. Riders go through the ML-based recommendation system and are later added to the final trip itinerary.

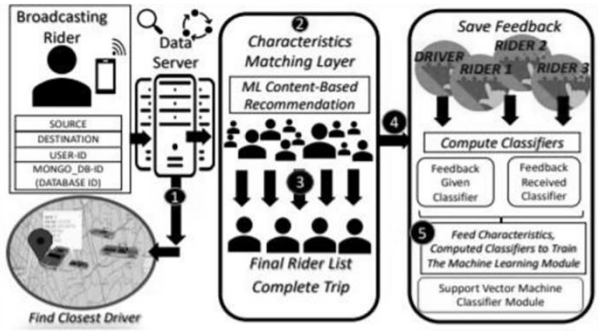

Fig. 1. The System Architecture.

The next step is of saving the feedback and computing two classifiers for every user. Riders are classified into two classes, and these classes are referred for rider recommendations in future trips. The following step is the training and testing of the Machine Learning classification model. For newly registering riders, the Machine Learning module predicts classifiers and provides better rider recommendations. The phase of Machine Learning is the final step of our architecture.

Machine Learning Recommendation System:-

Initially, rider characteristics are converted to a vector. For example, a broadcasting rider by with characteristics be chatty:3, safety:4, punctuality:3, friendliness:3 and comfortability:4 is represented as char v br = [3, 4, 3, 3, 4]. In Figure 2, 'O' represents the origin and points 'B', '1', and '2' represent the points plotted by the vectors for broadcasting and other riders. The next step is to compute the angular distance between vectors or the cosine of angle θ abusing the dot-product Equation 2

The case of rider matching by exact characteristics is when the θab equal to 0 or cosθab is 1. Therefore, a greater cosine value means a greater match. In Figure 2, char v1 or Rider1 seems to be a better match than char v2 due to a smaller angle θ1B. In rider matching simulations, we accept riders only with a cosθab value above 0.85 or 85%, which we felt is enough percentage to indicate a good match.

Computation of Classifiers:-

The first classifier is called the Feedback- Given-Classifier and uses the first part of the feedback data, which includes the ratings given by a rider to other riders. For example,

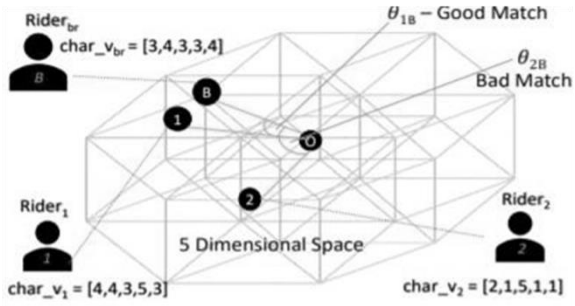

Fig. 2. Rider matching using Content-Based recommendation.

$$cos\theta_{ab} = \frac{(char\_v_a \cdot char\_v_b)}{\|char\_v_a\|\|char\_v_b\|}$$

Let the feedback given by Rider1 to other riders be as shown in Table 2. The data given by the rider is segregated characteristic wise and appended in new lists as follows: chattyRider1 = [0, 0, 1], sa f etyRider1 =

[2,3,5], punctuality Rider1 = [1, 0, 0], friendliness Rider1 = [4, 4, 4], and comfortability Rider1 = [0, 0, 0]. The observation made from the five lists is that Rider1 may continue to give a friend rating of 4 or comfort or a punctual rating of 0 in future trips. The only data variety observed is in the safety equation is stated in Equation3.

$$\sigma^2_{char} = \frac{\sum_{i=1}^{n_{char}} (x - x_{char\_i})^2}{data\_count_{char}}$$

The characteristic list with the highest variance is selected as this proves that the user is more diverse in rating the characteristic and therefore focuses more on the specific characteristic. After getting the first classifier, the system computes the Feedback-Received-Classifier is the safety class. The computation is done using the variance equation. Total number of elements in a characteristic list is n char or data count char.

The mean is denoted by x char i and variance Classifier using the second part of the feedback data-set, which is the feedback received by other riders to a rider.

Let the feedback given to Rider1 be as shown in the Table 3. Each element in the column has two values. The first value is the rating given by the rider for a specific characteristic, and the second value is the characteristic variance ((σi char) 2) computed for the individual rider's characteristics. Every time a rider provides feedback, the feedback value is multiplied by the corresponding characteristic variance. To exemplify, Rider2 variance for safety is 4.31, and the safety rating to Rider1 is 2.

The computed feedback to Rider1 safety characteristic is $2 * 4.31$, which is 8.62. In the end, for every rider's search criteria is dynamically defined by the system. The scenario at this stage resembles a practical use case as riders classify other riders based on their past experiences, which assists in better and real- time recommendations to riders

Table 2

Riders%	Chatty	Safety	Punctuality	Friendliness	Comfortability
Rider1	0	2	1	4	0
Rider2	0	3	0	4	0
Rider3	1	5	0	4	0

Table 3

Riders(%)	Chatty	Safety	Punctuality	Friendliness	Comfortability
Rider1	4*0.32	2*4.31	0*2.10	2*0.1	4*1.73
Rider2	3*3.45	1*0.15	1*0.55	0*5.72	3*3.34
Rider3	3*9.21	0*3.21	3*0.02	0*0.21	0*1.32
Total	39.26	8.77	0.61	0.2	16.92

Model Evaluation & Results:-

At first, we evaluate the Machine Learning model by computing the F1 score, precision, and recall for the five classifiers. The F1 score, recall, precision, and confusion matrix provides a comparison between computed scores and predicted scores. A higher score means the predicted classes match the computed classes for the same set of inputs. Tables 4 and Table 5 presents the characteristic, all multiplications are added and compared to get the highest characteristic value, which forms the Feedback-Received- Classifier. In the same example, the classifier is chatty, as the value is highest, which is 39.26. After computing the two classifiers, performance measures for both SVMs

The confusion matrix provides a two- dimensional array, which states how much error the module makes in predictions. The matrix reflects how much the model predicted correctly, also called as the true positive scores. In the case of the confusion matrix, if the diagonal elements have the highest values, the prediction of the model is notably accurate. From Figure 3, we conclude that our true positive scores for both SVMs are high, and the model predicts accurately. From Tables 5 and 6, the precision and recall is above 85% which is a good measure for a Machine Learning classification model. We indeed got an overall accuracy of 90% for both SVMs. Also, we computed the Root Mean.

Table 5

Measurement (%)	Chatty	Safety	Punctuality	Friendliness	Comfortability
F1 Score	87.85	89.02	90.63	93.22	93.21
Precision	86.13	87.52	92.58	91.97	95.48
Recall	89.85	88.82	89.67	94.49	96.96

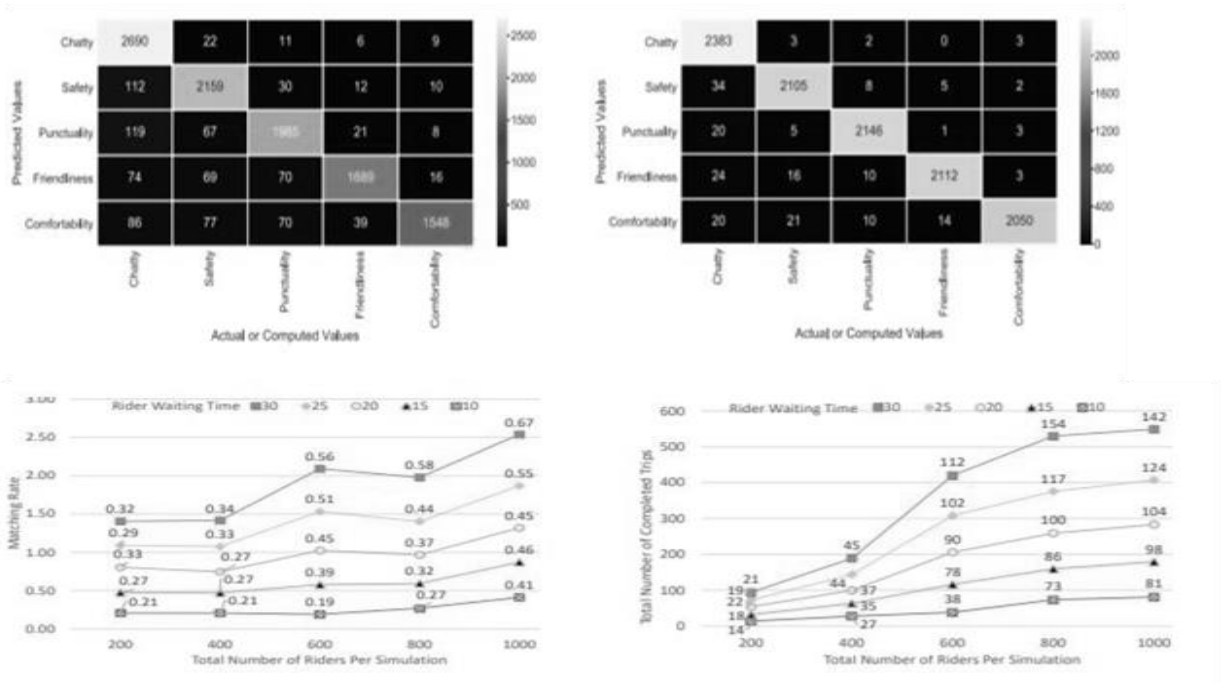

Fig. 4. Simulation matching rate (left) and total number of computed trips (right).

CONCLUSIONS

We implemented our designed and proposed model of Ride Sharing based on rider characteristics and Machine Learning Content-Based recommendation system. We subjected the model to an extensive simulation to test system performance. The matching rate and the number of completed trips continue rising with the progressing simulations or increasing number of traversed riders. Also, the SVM modules run with an accuracy of 90% and precisely predict classifiers for newly registering riders, which is crucial in providing better and real- time rider recommendations. Based on observations, the overall trip formation time rounds up to a minute. Indeed, we achieved our major goals of completing a maximum number of trips with pool completion and getting maximum rider matches by Exact and Closer characteristics matching. Our future work includes building a full-fledged Android or Web application as well as a sophisticated pricing model for users. Also, riders may be allowed to add riders as “Favourites,” and the system will recommend the added riders if they are broadcasting at the same time on a similar commuting trajectory.

REFERENCES

1. Apte, J.S., Messier, K.P., Gani, S., Brauer, M., Kirchstetter, T.W., Lunden, M.M., Marshall, J.D., Portier, C.J., Vermeulen, R.C., Hamburg, S.P., 2017. High-resolution air pollution mapping with Google street view cars: exploiting big data. ACS Environmental Science & Technology 51, 6999–7008.
2. Cramer, J., Krueger, A.B., 2016. Disruptive change in the taxi business: The case of Uber. American Economic Review 106, 177–82.
3. Dehak, N., Dehak, R., Glass, J.R., Reynolds, D.A., Kenny, P., et al., 2010. Cosine similarity scoring without score normalization techniques., in: Odyssey, p. 15.
4. Duan, Y., Mosharraf, T., Wu, J., Zheng, H., 2018. Optimizing carpool scheduling algorithm through partition merging, in: IEEE International Conference on Communications (ICC), pp. 1–6.
5. Han, S., Qubo, C., Meng, H., 2012. Parameter selection in SVM with RBF kernel function, in: IEEE World Automation Congress, pp. 1–4.
6. He, Y., Ni, J., Wang, X., Niu, B., Li, F., Shen, X., 2018. Privacy-preserving partner selection for ride-sharing services. IEEE Transactions on Vehicular Technology 67, 5994–6005.
7. Huang, S.C., Jiau, M.K., Lin, C.H., 2014. A genetic-algorithm-based approach to solve carpool service problems in cloud computing. IEEE Transactions on intelligent transportation systems 16, 352–364.
8. Inc., U.T., 2019. How does Uber match riders with drivers? URL:https://marketplace.uber.com/matching. [Accessed November 1, 2019].
9. Jiang, S., Hartley, R., Fernando, B., 2018. Kernel support vector machines and convolutional neural networks, in: IEEE Digital Image Computing: Techniques and Applications (DICTA), pp. 1–7.
10. Li, Z., Hong, Y.,Zhang, Z., 2016. An empirical analysis of on-demand ride sharing and traffic congestion, in: AIS International Conference on Information Systems.
11. Mallus, M., Colistra, G., Atzori, L., Murroni, M., Pilloni, V., 2017. A persuasive real-time carpooling service in a smart city: A case-study to measure the advantages in urban area, in: 20th IEEE Conference on Innovations in

Clouds, Internet and Networks (ICIN), pp. 300–307.

12. Math, Science, . What is variance in statistics? learn the variance formula and calculating statistical variance! URL: https://www.youtube.com/watch?v=sOb9b_AtwDg.
13. Nguyen, H.V., Bai, L., 2010. Cosine similarity metric learning for face verification, in: Springer Asian conference on computer vision, pp. 709–720.
14. NYC, 2019. NYC open data. URL: https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc. [15] Rodriguez, G., 2019. Autonomous vehicles and unmanned aerial systems: Data collection and liability [leading edge]. IEEE Technology and Society Magazine 38, 14–16.
15. Shaheen, S., Cohen, A., 2019. Shared ride services in North America: definitions, impacts, and the future of pooling. Transport Reviews 39, 427–442.
16. Teubner, T., Flath, C.M., 2015. The economics of multi- hop ride sharing. Springer Business & Information Systems Engineering 57, 311–324.
17. Wang, X., 2019. Preparing the public transportation workforce for the new mobility world, in: Elsevier Empowering the New Mobility Workforce, pp. 221–243.
18. Wang, Y., Gu, J., Wang, S., Wang, J., 2019. Understanding consumers' willingness to use ride-sharing services: The roles of perceived value and perceived risk. Elsevier Transportation Research Part C: Emerging Technologies 105, 504–519.