Route Prediction for Instant Delivery

YAN ZHANG, Peking University, China
YUNHUAI LIU*, Peking University, China
GENJIAN LI, Alibaba Local Services Company, China
YI DING, University of Minnesota, United States and Alibaba Local Services Company, China
NING CHEN, Alibaba Local Services Company, China
HAO ZHANG, Alibaba Local Services Company, China
TIAN HE, University of Minnesota, United States and Alibaba Local Services Company, China
DESHENG ZHANG, Rutgers University, United States

Instant delivery has drawn much attention recently, as it greatly facilitates people's daily lives. Unlike postal services, instant delivery imposes a strict deadline on couriers after a customer places an order online. Therefore it is critical to dispatch the order to an appropriate courier to guarantee the timely delivery. Ideally couriers should choose the optimal routes with the lowest overdue rate (i.e., the rate of the deliveries that are not finished in time) and the minimal distance. In practice, however, decision-making of the couriers is quite complex because individuals have different psychological perception of the environments (e.g., distance) and delivery requirements (e.g., deadline). To well predict their behaviors, we design multiple features to model the decision-making psychology of individual couriers and predict couriers' route with a machine learning algorithm. In particular, we reveal that perceived distance is the main factor influencing couriers' decision, which should be modeled based on the subjective understanding of the actual distances. Our design is implemented, deployed and evaluated on Ele.me, which is one of the largest instant delivery platforms in the world. Experimental results show that the overdue rate can be reduced by 48.02%, which is a significant improvement.

CCS Concepts: • **Networks → Location based services**.

Additional Key Words and Phrases: route prediction, instant delivery, perceived distance, machine learning

ACM Reference Format:
Yan Zhang, Yunhuai Liu, Genjian Li, Yi Ding, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. 2019. Route Prediction for Instant Delivery. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 124 (September 2019), 25 pages. https://doi.org/10.1145/3351282

*Corresponding author.

Authors' addresses: Yan Zhang, Peking University, No.5 Yiheyuan Rd. Haidian District, Beijing, China, yan.zhang_bibdr@pku.edu.cn; Yunhuai Liu, Peking University, No.5 Yiheyuan Rd. Haidian District, Beijing, China, yunhuai.liu@pku.edu.cn; Genjian Li, Alibaba Local Services Company, Jinshajiang Rd. Putuo District, Shanghai, China, genjian.lgj@alibaba-inc.com; Yi Ding, University of Minnesota, Minneapolis, Minnesota, United States, Alibaba Local Services Company, Jinshajiang Rd. Putuo District, Shanghai, China, dingx447@umn.edu; Ning Chen, Alibaba Local Services Company, Jinshajiang Rd. Putuo District, Shanghai, China, daniel.cn@alibaba-inc.com; Hao Zhang, Alibaba Local Services Company, Jinshajiang Rd. Putuo District, Shanghai, China, zh178187@alibaba-inc.com; Tian He, University of Minnesota, Minneapolis, Minnesota, United States, Alibaba Local Services Company, Jinshajiang Rd. Putuo District, Shanghai, China, tianhe@umn.edu; Desheng Zhang, Rutgers University, New Jersey, United States, desheng.zhang@cs.rutgers.edu.

2474-9567/2019/9-ART124 $15.00
https://doi.org/10.1145/3351282

1 INTRODUCTION

With the rapid development of mobile Internet and O2O businesses (i.e., online to offline, which refers to the use of online marketing to improve offline deals), new service models based on instant delivery are becoming increasingly popular, which enables many new applications such as takeaway delivery [12], supermarket fresh express[27], and city express [25]. In 2017, mainland China has over 10 billion instant delivery orders with a 314% year-on-year increase [3], accounting for 25% of the logistic volume[4].

Taking "Ele.me" [12], one of the largest takeout order delivery system, as an example. Ele.me has more than 168M (Million) registered users (11% of China population) and 2M cooperative merchants covering over 2000 cities in China. Among them, 34.9M are Monthly Active Users (MAU), and more than 3M couriers are finishing 28.8M orders every day. Taking Shanghai as an example, the location distribution of the merchants is shown in the Fig. 1.

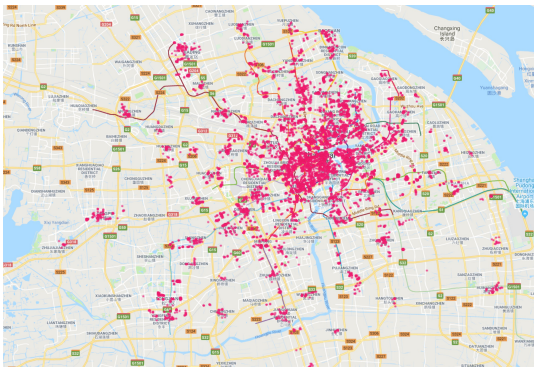


Fig. 1. Merchants location distribution in Shanghai

Fig. 2. An Ele.me courier

In a typical process of instant delivery service, customers place orders through the Ele.me's online platform. The platform will notify the merchants to prepare the takeaway packages. Couriers will be dispatched to pick up the packages and deliver them to the customers.

Usually, the delivery packages contain foods or urgent items. To promote the user experiences and improve the service quality, there is often a Time-of-Delivery (ToD), which is a deadline for each order. For example, Ele.me promises to deliver the food within 30 minutes[12]. Amazon Prime Now promise the one-hour delivery time [7]. If overdue (i.e., the courier does not finish delivery within the ToD), Ele.me will give a big discount or even make the order free. In 2017, "Ele.me" lost hundreds of million yuan due to overdue compensations. Accordingly, to reduce the overdue rate becomes the fundamental goal for instant delivery to ensure their business success.

Among many technical issues, a critical one is the order dispatch, i.e., to assign orders to appropriate couriers. A naive solution to guarantee the low overdue rate is to assign one courier for each order, which is too costly due to the high labor cost. As orders and customers are often close in vicinity areas, several nearby orders can be assigned to one courier who can finish all the orders by one trip. Notice that in traditional logistic managements, there is no strict time limitation, and thus they can always find enough nearby orders to minimize the cost. In instant delivery services, however, order dispatching has to be well designed to balance the overdue rate and the cost.

Online order dispatch has also been studied in other application scenarios such as ride-sharing [6, 38]. A taxi or uber driver may need to pick up several passengers and let them share the fare. Comparing these two service models, there are several key differences.

First, in ride-sharing, time is sensitive but there is no strict limitation. The passengers usually wait at the pick-up locations and drivers try their best to pick the passengers up. But if drivers come late, there is often no penalty to anyone. The cost of long pick-up time is actually taken by the customer and driver but not the platform.

Second, in ride-sharing, drivers usually take the route recommended by the navigation software or the platform. And thus we can find an optimized dispatch solution, which will largely be followed by the drivers, making the solution very effective in practice. In instant delivery, the couriers are individuals who travel on foot or motorcycles, as shown in Fig.2. They have their free will and belief of the short-cut route. They prefer to select the routes by their own. As we will show later in Sec.3.2, couriers hardly take the routes recommended by the platform. To well understand the behaviors of the couriers becomes an essential part of the order dispatching design.

To address the above challenges, in this paper, we propose OSquare, an instant delivery order dispatch system, to assign online orders to couriers with the best match. The basic idea is that every individual courier has his/her perceived distance rather than the physical distance. This perceived distance will take the number of orders, remaining time to order overdue, and many other factors into account. A machine learning algorithm is applied to accurately predict the courier's behavior. With such models, best match of order dispatch can be conducted.

To summarize, the main contributions of this paper are as follows.

- We conduct large scale of empirical studies for courier's behavior from one of the largest real instant delivery service systems in the world. The dataset contains 2.32 million order records of 6K couriers in one month. To the best of our knowledge, this is the first study in literature to have such scale of mobility data for couriers in instant delivery systems.
- We apply machine learning techniques to build the couriers' mobility models to predict the couriers' routes in instant delivery. We find that couriers are likely to select the next stop based on their perceived distance, which takes many psychological factors into account.
- Based on this couriers' mobility model, We develop OSquare, an instant delivery order dispatch system and implement it in real instant delivery production system. Real traces are collected for one month for experimental purposes.
- We compare OSquare Route, i.e., route prediction module in OSquare, with three baseline algorithms. (i) route plan algorithm in ride-sharing [6], (ii) an distance-based algorithm with no courier model, and (iii) Optimal Route algorithm. The four route algorithms are evaluated under four dispatch algorithms, i.e., OSquare Dispatch, Hungraian Method, Minimum Spanning Tree Method and Optimal Dispatch. Experimental results show that the prediction error of OSquare Route decrease by 16.5% and the overdue rate can be reduced by 48.02%.

The rest of the paper are organized as follows: In Sec. II, we describe the application background and problem formulation for our work. Sec. III shows the challenges of order dispatch in instant delivery. Sec. IV presents our OSquare system design, focus on the courier mobility models and their route prediction. Sec. V gives several dispatch algorithms to help evaluate OSquare Dispatch, and Sec. VI makes performance evaluations. Related works will be reviewed in Sec. VII, and we give some discussions in Sec.VIII. We conclude our work in Sec. IX.

2 BACKGROUND

In this section, we first provide more details of the instant delivery systems, and then give some necessary definitions and formulate our problem.

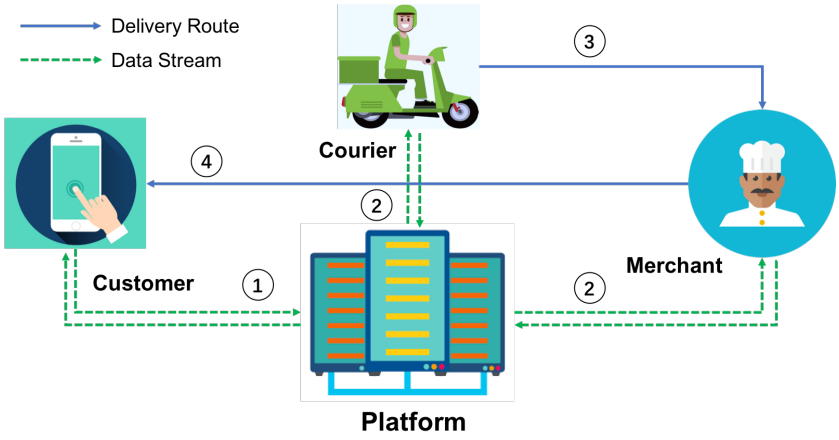


Fig. 3. Instant delivery diagram

2.1 Instant Delivery Scenario

The instant delivery services are quite simple, i.e., to deliver some items, products, or packages within a short period. As illustrated in Fig.3, a typical instant delivery service involves four roles and four steps. (i) *Customers* place orders online through the *platform*; (ii) The platform notifies the *merchants* to prepare the products and make the packages; (iii) The platform assigns a *courier* to pick up the packages; and (iv) the courier delivers the packages to customers.

When the customer is browsing to place orders, the platform shall provide the order's due time ToD for customer's reference, e.g., a KFC in 35 minutes. This ToD is based on the prediction of the real Delivery Time (DT). An accurate prediction of DT is essential as if the real DT is longer, i.e., DT>ToD, it is overdue, and the platform shall compensate the customers to avoid complains. On the other hand, too conservative ToD is not desired either. For example, suppose we take the strategy that a 35min package is claimed to be delivered in 45 minutes. In this case, we will have no overdue risk but may lose the customer as he may go to other platforms with shorter ToD. In that sense, accurate prediction of DT becomes critical for the success of the instant delivery services.

In real systems, an accurate prediction of DT is quite challenging. DT is an end-to-end measurement, from time that customers place orders to that of final delivery. Fig. 4 and Fig. 6 illustrates exampled route and timeline of two orders. At 12:00, the customer place the order O_1, and the merchant accepts the order. It takes 14 minutes, i.e., at 12:14, to make the package ready. Since the courier comes to the merchant earlier than 12:14, he has to wait until the package is ready. And if he comes later than the package ready time, it is the case of O_2 where the package is ready at 12:05 and the courier arrives at 12:21. The courier can just pick up and go.

In other words, the real DT includes the travel time from courier's current location to the merchant, the package preparation time, the package waiting time if the courier comes too early, and travel time from the merchant to the customer. All these time periods are highly dynamic in real world. And the platform shall consider all these factors when the customer is browsing orders and checking their ToD. When there are multiple orders (which is common), the courier shall pick them up and deliver in sequence, making the DT prediction even more challenging.

2.2 Problem Formulation

In this section, we summarize definitions and notations used in this paper.

2.2.1 Order. In instant delivery, an *order* is the goods which are placed by the customer online, picked up from the merchant and delivered to the customer within a given time.

Each order has several attributes, i.e., $O_i = \{m_i, c_i, a_i, p_i, t_i, e_i\}, i = 1, ..., N$, where m_i is the location of the merchant and c_i is the location of the customer. a_i is the time when the order is placed and p_i is the time when the package is ready in the merchant, which is predicted by the platform. t_i is the ToD of the order, which is given by the system to promise the service quality, e_i is the real finish time, when the orders is delivered to the customer.

2.2.2 Courier. *Couriers* are responsible for order delivery. In instant delivery, orders are continuously dispatched to the courier by the system, while couriers pick up and deliver orders in turn.

A *route* or *trip* of a courier starts when he is dispatched the first order ends when all his orders are delivered, i.e., visiting sequence of merchants and customers. Notice that orders are continuously dispatched to the courier during the trip, instead of being dispatched just at the beginning of the trip.

2.2.3 Task Definition. **Dispatch Problem:** Order dispatch is conducted by region. Usually dozens of couriers are responsible for the delivery of all orders in a region, which is about a few square kilometers. For one region with M couriers and N orders placed at a certain moment, order dispatch is to find a proper match between the M couriers and N orders, aimed to minimize the overdue rate of orders, where overdue rate is $\frac{|\{O_i | e_i > t_i\}|}{N}$.

Route Prediction: A core part for order dispatch is route prediction, which will be detailed in Sec.4. Suppose the courier c is dispatched K orders, $0 \leq K \leq N$, route prediction is to predict courier's visiting sequence of merchants and customers of the K orders.

3 CHALLENGES

In this section, we use some examples to show why traditional algorithms are not very effective in the instant delivery scenarios. We also conduct some empirical studies to show the impact of couriers' free will to order dispatch, which is a main challenge for us.

3.1 Ride-sharing and Instant Delivery

The instant delivery service model shares some similarities with that of ride-sharing [6], a quite successful business model in recent years. In this part, we make a comparison with these two new emerging applications, revealing the unique features of the instant delivery.

In a ride-sharing service, passengers (customers) place ride-sharing orders through the platform, e.g., Didi[11] or uber[29]. And the platform will aggregate those close-by orders and assign the drivers to take the passengers one by one, and then deliver them in turn. During the service, the driver would like to minimize the entire travel distance or service time so that their profit can be maximized. To the contrast, the goal of instant delivery is to minimize the overdue rate rather than the travel distance. Notice these two goals are fundamentally different.

To illustrate this, Fig. 4 and Fig. 5 give an example. Suppose we have two orders O_1={m_1, c_1 t_1=30min} and O_2={m_2, c_2 t_2=25min}. In the Fig. 4 approach, the courier travels to m_1, m_2 to pick up and to c_1 and c_2 to deliver. In this case, the DT for the first order T_1=5+7+10=22 min and that of O_2 is T_2=27 min. Though this approach has the shorter distance and lower service time, but the second order O_2 is overdue. An alternative approach is by Fig. 5 where T_1=29 min and T_2=24 min. Though the total service time becomes longer of 29 min>27 min, no order is overdue and the customers are more satisfied, i.e., DT is more closed to ToD.

Besides these concerns on overdue rate, the order preparation has also great impact on the system design. Suppose the preparation time for the two orders are p_1=14 min and p_2=5 min respectively (e.g., O_2 is some fast

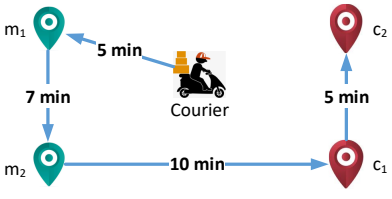


Fig. 4. Greedy route for ride-sharing

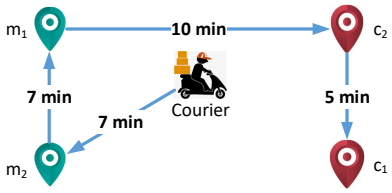


Fig. 5. Optimal route for instant delivery

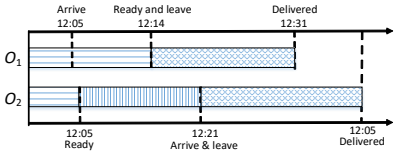


Fig. 6. Timeline of orders

food while O_1 is more complex), as illustrated in Fig. 6. When the courier arrives at m_1 at 12:05, he has to wait until 12:14 when the package is ready. In this case, the service time of Fig. 4 become T_1= 31 min and T_2=36 min. Both orders are overdue. The performance of the two approaches with and without package preparation time is summarized in Tab. 1

To show the impact of these observations, we study the routes of 6K couriers in a month. As shown in Tab. 2, if the couriers make route decisions purely based on distance, (i.e., couriers follow the solution of traveling salesman problem with pick-up and delivery, which is a variant of traveling salesman problem, as each order needs pick-up before delivery), the overdue rate is up to 2.32%. The averaged delivery time is 46.7 min, which is too long. In real system, couriers have only 1.31% overdue rate and the averaged delivery time is 36.6 min, 21.4% lower than routes with the minimal distance.

We further analyzed the route generated by ride-sharing approach and the courier's route selection. Fig.7 depicts the CDF of number of carried orders in one trip. The median number of orders per trip is two (locations to go are four), while certain couriers may carry up to 32 orders at a time. We use edit distance[24], a metric to counting the minimum number of operations required to transform a route sequence to another one, to quantify the difference between two routes. Tokens of edit distance refer to the locations of merchants or customers. Specifically, it refers m_i, when the courier picks up O_i, or c_i, when he delivers O_i. Fig.8 shows the CDF of edit distances between the ride-sharing routes, and the real routes taken by the couriers. The median of edit distance is 2, meaning that on average 50 percents of route are wrong (i.e., the route of ride-sharing is different courier's route). And the edit distance is up to 58, which occurs when the trip contains 32 points. The two trips of 32 points are totally different from the other, which indicates the courier dose not follow the route that ride-sharing method provides.

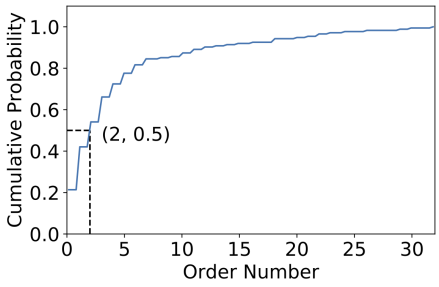


Fig. 7. Cumulative Probability of Order Number

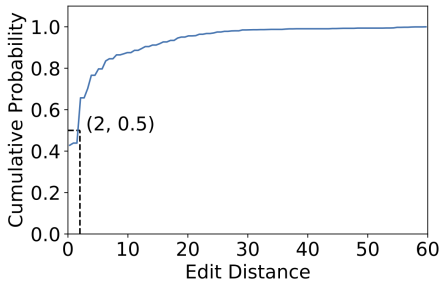


Fig. 8. Cumulative Probability of Edit Distancer

In summary, the route selection method in traditional ride-sharing applications is neither effective (by high overdue rate), nor taken by the couriers (by high edit distance). In fact, every minute in instant delivery is accounted. The customers only care about the overdue, and so as the couriers. In ride-sharing, passengers and drivers are more tolerant to delay because that can be blamed to many factors such as traffic jam, bad weather, or even the network latency. In instant delivery, there is no excuse.

Table 1. Service w/ and w/o preparation

		Order O_1	**Order** O_2	Overdue
Due		12:30	12:25	
w/o prep.	Fig. 4	12:22	12:27	1
	Fig. 5	12:29	12:24	0
w prep.	Fig. 4	12:31	12:36	2
	Fig. 5	12:29	12:24	0

Table 2. Performance of ride-sharing and courier's decision

	Ride-sharing	**Courier's decision**
Overdue rate	2.32%	1.31%
Averaged DT(mins)	46.7	36.6

3.2 Free Will of Couriers

In instant delivery, the recommendation for optimal route is another challenge. The main goal for instant delivery is to minimize overdue rate. However, the calculation of overdue rate highly depends on much real-time information, such as weather and traffic condition. These factors greatly affect couriers' speed, but are almost impossible to be obtained accurately. Moreover, as orders are continuously placed online, it is difficult to get a global optimization on overdue rate. Without the future order information, the dispatch and recommendation based on current orders are far from global optimization. Third, even all of the data was available, the solution for order dispatch and route recommendation would be NP-hard, so it is difficult to for us to get the optimal route.

So we implement a sub-optimal route recommendation, whose goal is to minimize total delivery distance[35]. We conduct experiments to study how couriers are incentivised to follow the sub-optimal routes. We compare the route recommended and couriers' actual route. Kendall rank correlation coefficient, i.e.,a statistics used to measure the ordinal association between two sequence[14], is used to measure the difference. Let $(O_1, x_1, y_1), (O_2, x_2, y_2), ..., (O_n, x_n, y_n)$ be a set of observations, i.e., O_i is the $i-th$ order of the courier, x_i is the order of O_i in the recommended pick-up order, and y_i is the order of O_i in the actual route. Any pair of (O_i, x_i, y_i) and (O_j, x_j, y_j) is said to be concordant, if both $x_i > x_j$ and $y_i > y_j$ or if both $x_i < x_j$ and $y_i < y_j$. Otherwise, it is said to be a discordant pair. The Kendall τ is defined as:

$$\tau = \frac{\text{number of concordant pairs - number of discordant pairs}}{n(n-1)/2}$$

In the evaluation, we consider two scenarios, with and without incentive. The result was shown in Fig.9. When the trip is shorter than 5, Kendall τ is no more than 0.6, which means at least 20% percent of couriers' decisions are different from the recommended route. What's more, when couriers are incentivised to take the route recommended, the difference between two routes is further increased.

We further speculate the reason why couriers do not follow the optimal route. The reasons are listed: 1) Although the recommended route is optimal under current conditions, couriers can judge from experience and predict the future order distribution, believing there are better route to achieve global optimization when considering future orders. 2) Couriers cannot judge whether the recommended route is optimal, so they don't want to take it. The reason why couriers more deny the route recommended when they are incentivised is as follows. Couriers are usually incentivised with more money under bad weather conditions. However, the real-time data under such conditions cannot be obtained accurately and is far from normal, which makes route recommendation less credible. So couriers would not take the route.

Since it is difficult to get the optimal recommendation, and sub-optimal recommendation is not taken by the courier, we consider predicting the route of couriers. Based on an accurate route prediction, we can achieve an effective dispatch.

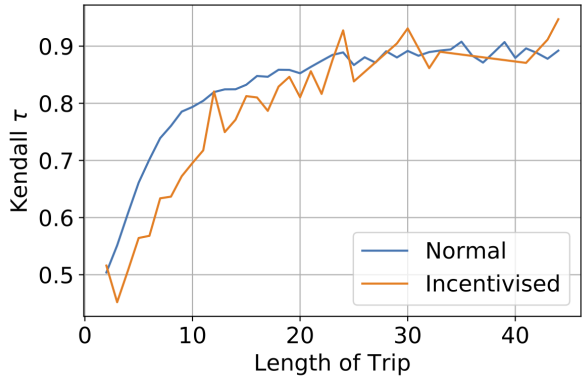


Fig. 9. Difference between recommended route and actual route

4 ROUTE PREDICTION

In this section, we introduce the route prediction algorithm which is the key to our OSquare system, named OSquare Route. We first introduce the concept of *Perceived Distance*, which brings the insight to our design. And then we show how to select features and train the route prediction model. In the last, we give the procedure to generate the route sequence.

4.1 Perceived Distance Concept

Given a number of orders in a courier's service, the goal of route prediction is to predict the visiting sequence of the merchants and customers for the courier. Couriers are usually limited to make a global optimization for their route. Instead, they prefer to determine only the next stop and visit merchants and customers one by one. Intuitively, they would like to select a near next stop, while this is not always the case. For example, orders with less ToD shall be serviced earlier, which makes these orders feel closer. And there are may other factors that may alter the courier's perceptions on the distances.

Basically, Perceived Distance is a concept borrowed from the psychology [8]. It tells us that the distance that an individual estimates from one place to another is subject to the individual's understanding of the actual distance, which is based on his/her memory, knowledge and recognition. For example, we may have the experience that when we go somewhere strange, the going trip is felt farther and slower, but the return trip is closer and faster.

Though the two trips are exactly the same distance, our perceptions are different and distorted. It is mainly because the going trip is strange, which makes us feel farther. And when we return, the return trip becomes familiar.

Perceived Distance mainly depends on an individual's social, cultural background and life experience. Invisible factors such as people's goals and physiological states, also influence their distance perceptions. Lee et. al.[16] proposed that the distortion in distance estimation is influenced by the desirability of the destination. A more desirable place feels closer. Proffitt [22] claimed that objects appear farther as the energy required to act on them increases. For example, when people throw balls to targets, targets become more distant when the balls are heavier.

As people's perception on distance is affected by many factors, we extract features from these factors and build our Perceived Distance Model(PDM) to predict a courier's next stop, and finally generate his/her whole route sequence. Fig.10 shows the overview for route prediction. With real-time courier and order data input, features related to perceived distance, e.g., spatial features, temporal features and environmental features, are extracted to predict courier's next stop. Based on PDM, route sequence is generated.

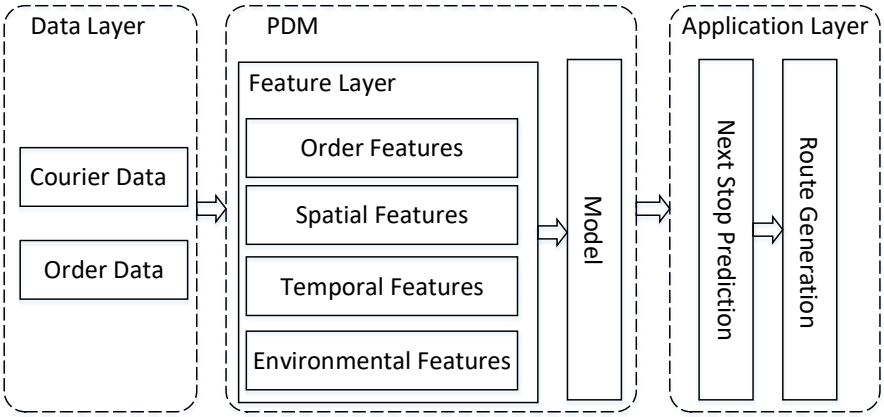


Fig. 10. Overview for route prediction

4.2 Next Stop Prediction Task

As a basis of route prediction, next stop prediction is introduced at first.

For each order, there are three statuses: *dispatched, picked-up and delivered*, corresponding to the status when order was dispatched to a courier, picked up from the merchant, and delivered to the customer. Based on this, *Courier state* refers to the state of all order the courier is carrying. *Available locations* refer to the locations that are available in certain courier state. Take Fig.11 as a example. At a_2, with the two dispatched orders O_1 and O_2, the courier will select the next stop from available locations, i.e., $\{m_1, m_2\}$. After he picked up O_2 at p_2, with order status changed, available locations are changed to $\{m_1, c_2\}$. It is obvious that during the whole trip, with courier's pick-up and deliver action, his order status and available locations are continuously changing.

The task of next stop prediction is to predict **which stop the courier is most likely to go when he has several available locations**. So it is a ranking task, i.e., rank all the accessible locations and predict the courier to choose the top one. Notice that for each decision of the courier, there is only m_i or c_i in the available locations.

This is because if the O_i hasn't been picked up, only m_i is accessible; if the O_i has been picked up and is pending delivery, only c_i is accessible.

The training data is collected from courier's historical data. Each time courier is dispatched, picks up or delivers an order, the timestamp, his current location and order status are uploaded to the system. So we reproduce the scene when the courier makes a decision, and use courier's actual decision as label. Suppose at time t, courier stays at l_t, with available locations $v_t = \{v_t^r | r = 1, ..., t_n\}$, t_n is the number of available locations. So the training sample is $< t, l_t, v_t^1, v_y >, ..., < t, l_t, v_t^n, v_y >$. v_y is 1 if the courier choose v_t^r as his next stop, otherwise v_y is set to 0. Take Fig.11 as an example, the train sample contains $< a_2, l_{a_2}, m_1, 0 >, < a_2, l_{a_2}, m_2, 1 >, < p_2, l_{p_2}, m_1, 1 >, < p_2, l_{p_2}, c_2, 0 >$ and so on.

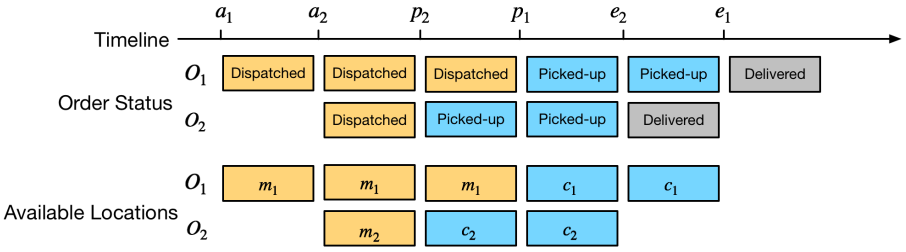


Fig. 11. An example for courier's trip

4.3 Feature Extraction

Based on the concept of the Perceived Distance, the features of the PDM are categorized to four groups, i.e., order, spatial, temporal, and environmental features. For courier's decision at time t, we extract multiple features to characterize v_t^r in available locations ,i.e., $v_t = \{v_t^r | r = 1, ..., t_n\}$ and v_t^r is the merchant or customer location for O_i.

4.3.1 Order Features. Order features provide the basic information for the orders in consideration.

Location type (B_picked) can be a merchant or a customer. Couriers normally behave differently for merchant and customer locations as the ToD constraint is strict for customer but not for the merchant.

Location number (N_d) is the number of accessible locations. As more orders are serviced at the same time, couriers has more options to make decisions, which may affect the courier's perceived distance.

The number of the same type locations (N_sameType) also helps understand the details of the couriers' decisions and their perceived distances.

Order amount (N_dish) is the number of dishes.

Order price (O_p) is the price of the order.

Merchant ID (O_m) is used to identify merchants.

4.3.2 Spatial Features. Spatial features describe spatial relationship between locations.

Location distance (D_sd) $d(l_t, v_t^r)$ is the great circle distance from the courier l_t to a location v_t^r. Perceived distance highly relies on this geographic distance[8].

Nearest mark(B_nearest) can be 0 or 1, indicating whether v_t^r is the nearest location for courier's current location among available locations v_t.

Order distance (D_mc) $d(m_i, c_i)$ is the distance between the merchant and its corresponding customer for a given order. The order distance reflects the difficulty of the order delivery.

To-all distance (D_other) $D(v_t^r)$ is the sum of distances from the v_t^r to all other locations in the available locations, i.e., $D(v_t^r) = \sum_j d(v_t^r, v_t^j)$. This feature indicates how far the location v_t^r is to other locations.

4.3.3 Temporal Features. Temporal features describe the time requirements for each location.

Pickup time (T_pick) is the remaining time till pickup, i.e., $t - p_i$, where t is the current time, and p_i is the time when the package is ready in the merchant. This feature indicates current goods preparation condition. Notice that p_i is a predicted one, which is beyond this work. If the p_i isn't predicted, the courier will estimate it based on the his experience of delivery to help make decisions.

Overdue time (T_ToD) means the remaining time till the order is overdue, i.e., $t_i - t$ where t_i is the ToD. This feature is just the ToD constraint, which has great influence on couriers' decision.

Urgent mark(B_urgent) can be 0 or 1, indicating whether v_t^r is the location with the least T_ToD among available locations v_t.

Placement time (T_a) means the time from the order is placed, i.e., $t - a_i$, where a_i is the time when the order is placed. This feature shows how long the orders has been on service.

Time budget (T_budget) is the time budget for a location. If the location is a merchant, this budget equals the ToD subtracting the the time cost from the merchant to its customer. And if it is a customer, it equals to the current time to ToD, i.e., $t_i - t$. A smaller budget implies that the order is nearly overdue and is urgent.

Urgency (T_left) is the difference between the location's time budget and the maximal of time budgets of all other locations. Relative urgency shows whether the location v_t^r is the most urgent one.

Rank for overdue (R_ToD) means the rank for remaining time till order is overdue. The less the remaining time is, the higher the rank is. Notice that the rank is for all accessible locations.

4.3.4 Environmental Feature. Environmental feature gives some contextual content about couriers' decision.

Weather condition (C_weather) describes the real-time weather condition, which is classified into five categories based on meteorological conditions, i.e., sunny, light rain, heavy rain, windy and snowy.

4.4 Feature Selection

Not all features are equally important for courier's route decision. In this part, we apply the feature engineering techniques to select the appropriate set of features. This selection can effectively reduce the possibility of model over-fitting and make the model with better generalization performance. It can also help enhance the understanding between features and classes[9, 28].

4.4.1 Feature Selection Method. **1) Variance Analysis.** For numerical features (e.g., Time Budget), usually features with large variance can provide more information and are more helpful. Based on this observation, we calculate the variance for each feature as follows.

$$Var(x) = E[x - E(x)]^2 \tag{1}$$

where where x is a feature and $E(x)$ is the expected value of x.

2) χ^2 Test For categorical features (e.g., Location Type), we use χ^2 test to analyze the independence between features and the classes[21, 33]. In our problem, the class is whether the location is chosen by the courier. For a feature x with two categories (i.e., $x = 0$ or $x = 1$), and binary classification label (i.e., $y = 0$ or $y = 1$), we count the number of sample in different categories as Tab.3, where c_{ij} is the number of sample which $x = i$ and $y = j$. Then we can get the theoretical value as Eq.2. The χ^2 value denotes the deviation between observed and theoretical values as Eq.3, where, $c_{i,j}$ is the observed frequency, $f_{i,j}$ is the theoretical frequency (when the feature has no impact on class). Then the degree of freedom can be calculated as $n - 1$, where n is the number of categories. We can get the $p - value$ from χ^2 value and degree of freedom, and features with $p - value \leq 0.05$ is considered statistically significant[18].

Table 3. Frequency table

Frequency	$y = 0$	$y = 1$
$x = 0$	$c_{0,0}$	$c_{0,1}$
$x = 1$	$c_{1,0}$	$c_{1,1}$

$$f_{ij} = \frac{(\sum_i c_{i,j})(\sum_j c_{i,j})}{\sum_{i,j} c_{i,j}} \tag{2}$$

$$\chi^2 = \sum \frac{(c-f)^2}{f} = \sum_{i,j} \frac{\left(c_{i,j} - f_{i,j}\right)^2}{f_{i,j}} \tag{3}$$

3) Feature Subset Search Given the feature set $\{f_1, f_2, ..., f_k\}$, We treat each feature as a candidate subset and evaluate the k single-feature candidate subsets. Assuming $\{f_2\}$ gets the best prediction performance, we take $\{f_2\}$ as the selected set of the first round. Then we adds a feature to the selected subset to make candidate subsets containing two features. Assuming that the results of $\{f_2, f_5\}$ in the k-1 candidate dual-feature subsets are optimal, $\{f_2, f_5\}$ is then used as the selected set of the current round. It is assumed that in the $(z+1)-th$ round, the optimal candidate (z+1)-feature subset is worse than the selected set of the previous round, then we stop the process, and the selected z-feature set in the previous round is selected as the feature set.

4.4.2 Feature Selection Process. We summarize the whole feature selection process. First, for each numerical feature, variance of the feature is calculated. Features with variance equal to 0 are eliminated. Second, we conduct χ^2 Test for each categorical feature. Features with p-value greater than 0.05 are eliminated. Finally, we use feature subset search, to find the feature set which can improve prediction performance.

After all these three steps, we are left with 12 features. They are B_nearest, B_urgent, B_picked, T_budget, T_left, N_sameType, D_sd, D_other, D_mc, T_pick, T_ToD, N_d.

4.5 Model Training

For the next stop prediction, we use XGBoost[10] to achieve the ranking task. XGBoost is a scalable tree boosting system with a large number of decision trees to learn function sequentially, i.e., new predictors are learning from mistakes committed by previous predictors. Compared with other classification model, e.g., logistic regression, SVM, KNN and GBDT, XGBoost model has the following advantages:

- Compared with models such as logistic regression and SVM, XGBoost is able to learn the nonlinear relationship between features and classes, which adapts to our problem.
- XGBoost can provide the importance of features on dataset, which sheds the light on the courier behavior patterns. For example, the location distance and time urgency have great influence on the courier's decisions. Methods like KNN can hardly provide more information of the features.
- The model can deal with a large mount of data and can be efficiently trained compared to other tree boosting algorithm such as GBDT. Thus it is suitable for next stop prediction modeling, for in our application scenario, the data comes fast with millions of data each day.

In our experiments, we used the logarithmic loss function in Eq.4. In Eq.4, θ is the model parameter, m is the number of samples, $h_\theta()$ is the model learned, $y^{(i)}$ is the label for sample $x^{(i)}$. We use gbtree as the fitting function instead of gblinear, for the models based on tree have more explanation capacity for data[10]. All the other parameters can be got through grid search[1]. The model is trained with 5-fold cross-validation.

$$L(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta\left(x^{(i)}\right) + \left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right] \tag{4}$$

4.6 Route Sequence Generation

With the model learned, we can do the route sequence generation. Suppose the courier has N dispatched orders, we will predict his visiting sequence of merchants and customers one by one. Take fig.11 as an example. At first, courier has 2 dispatched orders. For his first stop, available locations contain m_1 and m_2. PDM is used to rank the two stops. Assume that m_2 with higher rank, i.e., more likely to be chosen by the courier, then first stop of the route sequence is set as m_2. Then we adjust courier's order status and get available locations for the second stop, which contain m_1 and c_2. Then assume PDM predicts m_1 with higher rank, so the second stop of sequence is m_1. We continue this process until the available locations are empty. In the example of Fig. 11, route generated is (m_2, m_1, c_2, c_1).

Notice that during the process, courier's location and timestamp are continuously changing, which will affect the feature calculation. Suppose courier stands at l_t at time t initially, then we select the first stop from $\{m_1, m_2\}$. After m_2 is set as courier's first stop, we will change courier's location as m_2 and timestamp as $t + \frac{d(l_t, m_2)}{v}$ for the second iteration, where v is the speed of the courier. Then for the third iteration, courier's location is changed as m_1 and timestamp as $t + \frac{d(l_t, m_2)+d(m_2, m_1)}{v}$. We give details of the entire process in Tab.4.

Table 4. An example for route sequence generation

Iteration	Available Locations	Courier's Location	Timestamp	Target Stop
1	$\{m_1, m_2\}$	l_t	t	m_2
2	$\{m_1, c_2\}$	m_2	$t + \frac{d(l_t, m_2)}{v}$	m_1
3	$\{c_1, c_2\}$	m_1	$t + \frac{d(l_t, m_2)+d(m_2, m_1)}{v}$	c_2
4	$\{c_1\}$	c_2	$t + \frac{d(l_t, m_2)+d(m_2, m_1)+d(m_1, c_2)}{v}$	c_1

5 DISPATCH BASED ON PREDICTION

In instant delivery services, the orders keep coming and the couriers are online to wait for new orders. Suppose at each time instance, there are M new orders and N couriers in nearby areas (some may already have orders in hand). The goal of the order dispatch, is to assign these M new orders to the N couriers so that when these orders are serviced, overdue rate is minimized. In this section, we first propose OSquare Dispatch, which is the dispatch algorithm implemented in the real system. And then we show three classical dispatch algorithms, i.e., Hungarian Method, Minimum Spanning Tree Method and Optimal Dispatch. These four methods take route predicted as data basis to make order dispatch. In the later Sec.6, we will show how the difference of dispatch with and without route prediction.

5.1 OSquare Dispatch

In this method, M new orders are dispatched one by one. For each new order, we try to dispatch the order to every courier $c_i, 1 = 1, ..., N$ and then predict the courier's route based on this new set of orders. With the route predicted, we can estimate his overdue rate and the increased journey distance, i.e., journey distance with the new order subtracting the original journey distance. After all these N couriers' routes are analyzed, the order will be dispatched to the courier with least overdue rate and least increased journey distance. Notice that the

overdue rate is more prioritized. The increased journey distance is used only when two dispatch schemes have the same estimated overdue rate. Fig. 12 depicts the order dispatch process for OSquare.

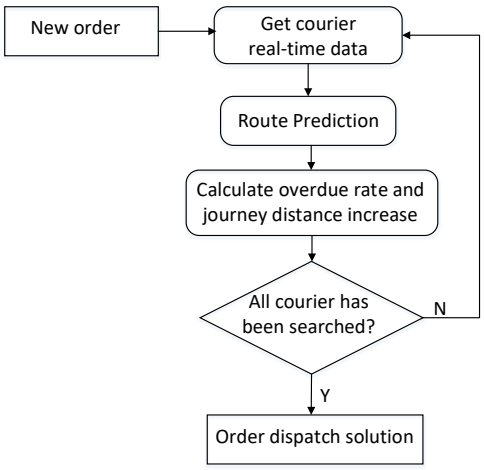


Fig. 12. OSquare Dispatch process

5.2 Hungarian Method

The Hungarian Method is a combinatorial optimization algorithm that solves the dispatch problem in polynomial time. In the method, the dispatch problem is represented as a directed graph. In the graph, there is a set U of order vertices u, a set C of courier vertices c, and a set E of edges $e = (u, c)$, connects an order vertex and an courier vertex. The weight of e is journey distance with the new order, which can be calculated from route predicted. Since Hungarian Method requires a perfect match, i.e., $M = N$, however, in the general instant delivery scenario, $M \geq N$, so we need to run the method for several iterations.

For the first iteration, the graph contains vertices of the first N placed order and N couriers, and the dispatch problem is set as a matrix of weights of edges between them. Then we uses 4 steps to solve this problem. Step 1: for each row, we finds the lowest element and subtract it from each element in that row. Step 2: for each column, we finds the lowest element and subtract it from each element in that column. Step 3, we cover all zeros in the resulting matrix using a minimum number of horizontal and vertical lines. The algorithm stops when it needs N lines. Otherwise, Hungarian Method continues with Step 4. In step 4, Hungarian Method finds the smallest element (call it k) that is not covered by a line in Step 3. Then it subtracts k from all uncovered elements, and add k to all elements that are covered twice. Hungarian Method will repeat step 3 and step 4 until all elements can be covered by N lines. Then first iteration is finished, and the first N placed order has been dispatched to the courier. Then we will continue this process with next N orders. If the left orders is no more than N, we add some dummy vertices u_d into the order set. For all edges of $e = (u_d, c)$, we set the weight as 0. The process is continued until all the orders have been dispatched.

5.3 Minimum Spanning Tree Method

In this method, for each courier, we calculate the minimum spanning tree between the courier and the merchant locations of all orders pending dispatch with Kruskal algorithm[15]. For multiple subtrees with the courier as

the root node, we calculate the average distance of courier's with orders in the subtree. The delivery distance is based on the route predicted. Then the subtree orders with shortest average travel distance are dispatched to the courier. The algorithm is then continued for the next courier. If there are still undispatched orders after all couriers have been traversed, then we continue the process with the first courier.

5.4 Optimal Dispatch

In the optimal dispatch, we consider all the MN dispatch schemes. For each scheme, we predict the route for each courier and calculate the overall overdue rate and delivery distance of the dispatch scheme. Then we choose the scheme with smallest overdue rate and shortest delivery distance.

6 EVALUATION

6.1 Evaluation Methodology

6.1.1 Dataset. The dataset is obtained from Ele.me instant delivery system during the time period of 2019.03.15 - 2019.04.15. The dataset contains 2.32 million order records of 6K couriers in Shanghai and Chengdu. Since the trip with only one order does not reflect the performance of the model, we have excluded such sample in the experiment.

6.1.2 Evaluation Metric. As there are three steps in our dispatch system, i.e., next stop prediction, route generation and order dispatch, we have different evaluation metric for each of them.

- **Accuracy:** For the total N decisions, the actual target is l_i^a, the predicted target is l_i^p, accuracy = $\frac{\sum_{i=1}^{N} I(l_i^a, l_i^p)}{N}$, where $I(x, y)$ is the index function, $I(x, y) = 1$ when $x = y$, otherwise $I(x, y) = 0$.
- **Kendall τ:** Kendall rank correlation coefficient, i.e.,a statistics used to measure the ordinal association between two sequence[14], is used to measure the difference between two sequences. Let (l_1, x_1, y_1), ..., (l_n, x_n, y_n) be a set of observations, i.e., l_i is the merchant or customer location, x_i is the order of l_i in the couriers' actual route, and y_i is the order of l_i in the predicted route. Any pair of (l_i, x_i, y_i) and (l_j, x_j, y_j) is said to be concordant, if both $x_i > x_j$ and $y_i > y_j$ or if both $x_i < x_j$ and $y_i < y_j$. Otherwise, it is said to be a discordant pair. The Kendall τ is defined as:

$$\tau = \frac{\text{number of concordant pairs - number of discordant pairs}}{n(n-1)/2}$$

- **Edit Distance:** Edit distance[24] is a way of quantifying how dissimilar two sequences (e.g., words) are to one another by counting the minimum number of operations required to transform one sequence into the other. The operations include insertion, deletion, and substitution. The edit distance between $a = a_1 \dots a_n$ and $b = b_1 \dots b_m$ is given by d_{mn} as follows. The cost of insertion, deletion, and substitution are w_{ins}, w_{del} and w_{sub}. When we compare two route sequence, w is set as 1.

$$d_{i0} = \sum_{k=1}^{i} w_{\text{del}}(b_k) \qquad \text{for } 1 \le i \le m \tag{5}$$

$$d_{0j} = \sum_{k=1}^{j} w_{\text{ins}}(a_k) \qquad \text{for } 1 \le i \le n \tag{6}$$

$$d_{ij} = \begin{cases} d_{i-1,j-1} & \text{for } a_j = b_i \\ min \begin{cases} d_{i-1,j} + w_{del}(b_i) \\ d_{i,j-1} + w_{ins}(a_j) \\ d_{i-1,j-1} + w_{sub}(a_j, b_i) \end{cases} & \text{for } a_j \neq b_i \end{cases} \qquad \text{for } 1 \le i \le m, 1 \le j \le n \tag{7}$$

In our problem, tokens evaluated refer to the locations of merchants or customers. Specifically, it refers m_i, when the courier picks up O_i, or c_i, when he delivers O_i.

- **Overdue Rate:** Overdue rate is used to measure the performance of order dispatch. For N orders, overdue rate = $\frac{|\{O_i | e_i > t_i, i=1,\ldots,N\}|}{N}$, where t_i is the ToD of O_i and e_i is the actual finish time of O_i.
- **Average Overdue Time:** Average Overdue Time also measures the performance of dispatch. It is the average overdue time of all the overdue orders, as average overdue time = $\frac{\sum_i e_i - t_i}{M}$, for M overdue orders.

6.1.3 Baseline Approach. We regard three route calculation algorithm as baselines.

- Distance-based Method: Couriers always go to the nearest locations, without considering time requirements and other factors.
- Ride-sharing Method: Given courier's current location and his orders, with the pick-up and delivery constraint, a heuristic method is used to calculate the route with minimum travel distance.[13].
- Optimal Route: Given courier's current location and the orders, with all the constraints, we calculate an oracle route with minimum overdue rate and delivery distance. Notice that the overdue rate will be optimized first, followed by the delivery distance.

6.2 Route Prediction Evaluation

In this section, we evaluate OSquare Route with large real dataset. We first evaluate the next stop prediction and route sequence performance of our model. Then we study the tread of model performance during a day, and finally we further analyze varying behaviors by region.

6.2.1 Overall Evaluation. We evaluated the performance of OSquare Route and baseline algorithms, as shown in Tab.5. The results show that our model is better than the baseline model in both next prediction accuracy and sequence performance. Compared with distance-based method, our algorithm can reduce the error rate by 16.5%.

Table 5. Result statistics

Metric	Distance-based Method	Ride-sharing	OSquare Route	Optimal Route
Accuracy	0.6634	0.6857	0.7222	0.6977
Kendall	0.7060	0.7262	0.7486	0.7337
Edit Distance	3.3986	3.1920	2.9069	3.0904

6.2.2 Performance During a Day. In order to have more detailed evaluation of our model, we also evaluate OSquare Route during a day in different cities. The cities we choose are Shanghai and Chengdu. Shanghai is a city with developed economy in China, and also an international financial trade center. Shanghai is one of the largest cities in the world, with developed urban transportation and a large population. Located in western China, Chengdu is the capital of Sichuan Province, a business center in southwestern China with good economic development. The statistics about two cities [20] is listed in Tab.6.

Different demographics characteristics cause different order distribution in Shanghai and Chengdu. The order amount of the two cities is shown in Fig.13. We can get some observations from Fig.13: i) Order amount in Shanghai is always greater than Chengdu during the whole day, for Shanghai has larger population than Chengdu. ii) There are *lunch peak* and *dinner peak* in both cities, but the peak in Shanghai is much sharper than Chengdu. With better economic environment, Shanghai has more companies than Chengdu. Office workers in Shanghai prefer to eat takeaway food, which causes the sharp peak.

Fig.14 and Fig.15 shows the prediction accuracy trend in Shanghai and Chengdu during a day. It can be seen that: i) In both cities, the performance of OSquare is always better than the three baseline algorithms. It shows that OSquare adapts to different environments. Moreover, OSquare can even outperform Optimal Route. It illustrates that couriers do not take the optimal route, which leads to a lower accuracy of Optimal Route. ii) During a day, the larger order amount is, the lower prediction accuracy is. Obviously, with more orders serviced at the same time, couriers' route sequence becomes longer, which increase the difficulty for route prediction. iii) Compared with Shanghai, route performance of Chengdu is better. The reason is that Chengdu has less order amount than Shanghai, i.e., less orders serviced at the same time, making it easier to have a more accurate prediction.

Table 6. City statistics

Index	Shanghai	Chengdu
GDP($1 \times 10^8 yuan$)	30632.99	13889.39
Area(km^2)	6340.5	4684
Population($million$)	24.18	14.35

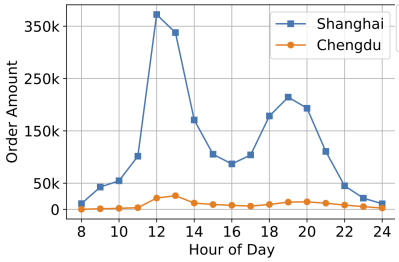


Fig. 13. order amount during a day

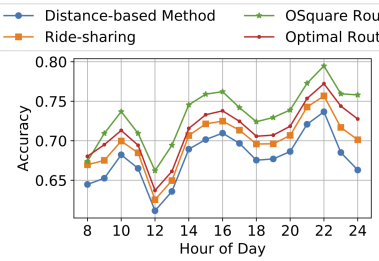


Fig. 14. prediction accuracy in Shanghai

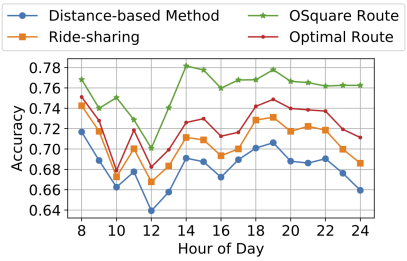


Fig. 15. prediction accuracy in Chengdu

6.2.3 Varying Behaviors by Region. We further study the differences between models in different cities and the generalization capability of models between cities. The models trained in Shanghai and Chengdu, called Shanghai Model and Chengdu Model, are both tested in these two cities. The performance is listed in Tab.7.

It can be seen that: i) For both models, the test results in Chengdu are better than those in Shanghai. It is because Chengdu's order amount is less than Shanghai's, making it less difficult to predict accurately.

Table 7. Model Comparison

Test City	Metric	Shanghai Model	Chengdu Model
Shanghai	Accuracy	0.7199	0.7162
	Kendall	0.6584	0.6675
	Edit Distance	3.0339	2.9369
Chengdu	Accuracy	0.7537	0.7439
	Kendall	0.7914	0.7522
	Edit Distance	2.6087	2.4347

ii) When we test with Shanghai's data, Shanghai Model was better than Chengdu model. When we tested with Chengdu's data, the result is the similar, i.e., Shanghai Model still performs better than Chengdu Model. We further analyze the feature importance analysis of the two models, as shown in Fig.16 and Fig.17. It can be seen that Shanghai Model balance both distance and time factors, while Chengdu Model pays more attention to distance factor than other factors. We speculate that the training data in Chengdu is relatively small, which makes the model only learns some simple characteristic of couriers. While the training data in Shanghai is much more and the data quality is higher than Chengdu (the decision-making scene of the courier is much more complicated due to the high order amount), it makes the model learns more from the couriers. So Shanghai Model has better performance than Chengdu Model in both cities.

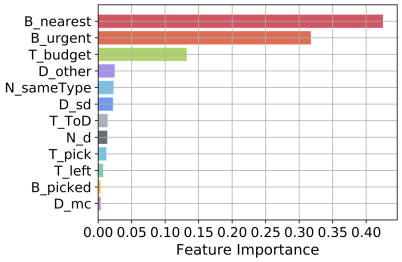


Fig. 16. Shanghai Model

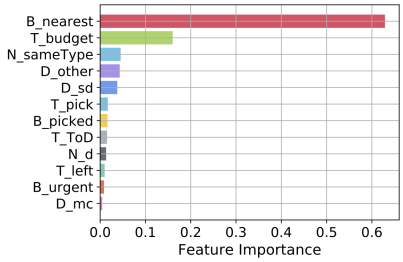


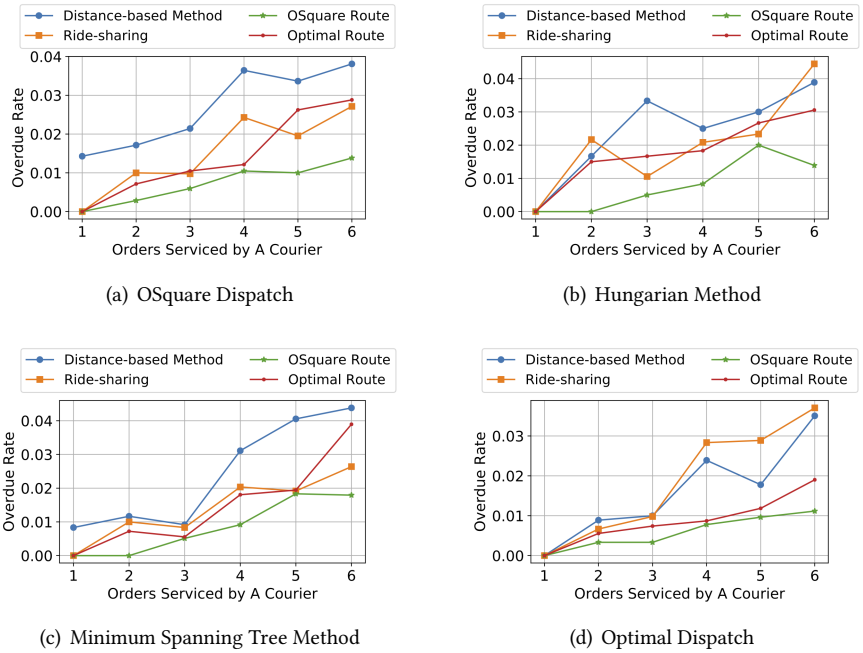
Fig. 17. Chengdu Model

6.3 Prediction-based Dispatch Evaluation

In this section, we evaluate the performance of our algorithm and baseline methods under different order dispatch algorithms. Because a bad order dispatch will greatly influence there venue of company and blamed by couriers, it is impossible to conduct online with other order dispatch algorithm. So we conduct an offline simulation. We generate the simulation data which has the same distribution with real data, and compare the performance of different order dispatch algorithms.

6.3.1 Overdue Rate. We evaluate the overdue rates of different methods under different order dispatch algorithms, as shown in Fig.18. We can get some conclusions: i) Under different dispatch algorithms, the overdue rate of OSquare Route, which is 1.31% on average, is lower than that of other methods. The overdue rate of the distance-based method, i.e., nearly 2.52%, is usually higher, and the result of the ride-sharing algorithm and the optimal route are similar. Compared with distance-based method, the overdue rate is reduced by 48.02%. Although Optimal Route is the optimal for couriers, its overdue rate is higher than OSquare, due to couriers less following it. The

result is consistent with the previous prediction performance. It indicates that the more accurate route prediction is, the lower overdue rate can be. Although there is a small improvement on next stop prediction, errors in next stop prediction will accumulate during the process, affecting the effect of order dispatch. So even small changes improvements on next stop prediction will make a difference. ii) On the other hand, comparing the results of OSquare Route in different dispatch algorithms, we can find that the optimal dispatch can achieve the best results, but it has great complexity and cannot be realized online. Compared to Hungarian method and Minimum Spanning Tree method, the current OSquare Dispatch performs better.



(a) OSquare Dispatch

(b) Hungarian Method

(c) Minimum Spanning Tree Method

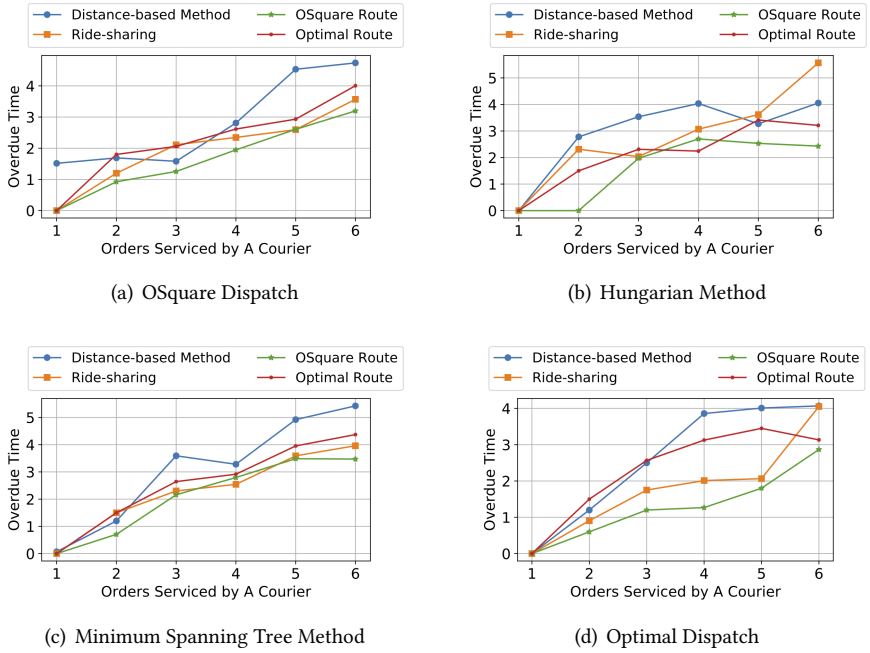
(d) Optimal Dispatch

Fig. 18. Overdue rate under different dispatch algorithms

6.3.2 Average Overdue Time. Furthermore, we evaluate the overdue time of different methods under different order dispatch algorithms, as shown in Fig.19. We conclude that: i) The gap between each algorithm in the overdue time is not as large as in the overdue rate. Under each different dispatch algorithms, the result of OSquare Route is slightly better than other baseline route algorithms. ii) Comparing the different order dispatch algorithms, it can be seen that the overdue time of the Hungarian method and Minimum Spanning Tree method is high, and the results of OSquare Dispatch and Optimal Dispatch are relatively low. Although the performance of optimal Dispatch is better than OSquare, considering the operation efficiency, OSquare Dispatch is much better in practice.

7 RELATED WORK

Compared with traditional logistics research, instant delivery more relies on the decision of the couriers. Route planning for traditional logistics is treated as traveling salesman problem with pick-up and delivery [19]. As the optimal solution to tsp problem is NP-hard, heuristic algorithms are often used to get a good enough solution[23].



(a) OSquare Dispatch

(b) Hungarian Method

(c) Minimum Spanning Tree Method

(d) Optimal Dispatch

Fig. 19. Average overdue time under different dispatch algorithms

Some commercial solvers such as Cplex [2] and gurobi[5] can be used to solve the problem. Lots of attributes must be taken into account when it applies to real world circumstances, such as vehicle's capacity, time limitation of delivery time and so on. Vidal et. al.[30, 31] made a comprehensive overview for algorithms to solve travelling sales problem with multiple attributes.

Mobility pattern modeling and prediction is another related topic that has been studied in recent years. The basic idea is to uncover the mobility pattern by mining the massive data produced by the crowd movement. Yang et. al.[34] builds a spatio-temporal bicycle mobility model using the historical bike sharing data. Similar bike sharing mobility prediction are proposed to solve the bike balancing problem between stations[32, 36]. Some network technology can also be used to help improve mobility pattern, such as LoRa[17]. Deep learning is also introduced in mobility pattern prediction where spatial region are divided into grids and inflow and outflow are predicted at certain time period[37]. Three main factors are considered: external factors (weather, events...), temporal dependency, spatial dependency. There are also some theoretical studies on the topic of human mobility pattern. The upper bound of the predictability of the human mobility is discussed by Song et. al.[26] using long term call detail records. The major difference between our work and mobility pattern study lies in that our work is more fine-grained and focus on the prediction of individuals instead of large crowd.

8 DISCUSSION

Since different couriers have different decision-making psychology, theoretically, personalized model for couriers can better reflect the characteristics of couriers. However, our experiments found that the personalized model does not perform better due to insufficient training data for each courier. Therefore, we consider that the couriers with similar behavior patterns should be clustered and modeled. In the later appendix, couriers are clustered into

three groups and the model of each group achieves a better portrayal of the couriers' characteristics. How to better cluster couriers is the key to the effect of the model.

What's more, time of delivery relies on a accurate prediction on order's actual delivery time, which is greatly affected by merchants' package preparation time. A more precise model for the product pickup time estimation, is essential for the overdue rate estimation of the instant delivery.

9 CONCLUSION

In this work, we studied the route prediction in the scene of instant delivery. The core part of route prediction is to portray the decision-making psychology of the courier and build the Perceived Distance Model. The χ^2 test is used select the appropriate features and we design a route prediction algorithm based on XGBoost model. We implement our design in Ele.me, one of the largest instant delivery platforms in the world. The experimental results show that the route prediction error is reduced by 16.5%, and the overdue rate can be significantly reduced from 2.52% to 1.31%.

ACKNOWLEDGMENTS

This work is supported partly by the National Key R&D Program of China 2018YFB0803400, 2018YFB2100300, and National Natural Science Foundation of China (NSFC) 61772046.

REFERENCES

[1] 2016. XGBoost Document. https://xgboost.readthedocs.io/en/latest/. Jan. 28th, 2018.
[2] 2018. Clpex Website. http://www.clpex.com/. Dec. 1st, 2018.
[3] 2018. Consulting Statistics. http://www.chyxx.com/industry/201806/650205.html. Dec. 23rd, 2018.
[4] 2018. Economic Statistics. https://wallstreetcn.com/articles/3343667. Dec. 23rd, 2018.
[5] 2018. Gurobi Website. http://www.gurobi.com/. Dec. 1st, 2018.
[6] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences* 114, 3 (2017), 462–467.
[7] Amazon. 1996. Amazon Website. https://primenow.amazon.com/. Dec. 31st, 2018.
[8] David Canter and Stephen K Tagg. 1975. Distance estimation in cities. *Environment and behavior* 7, 1 (1975), 59–80.
[9] Girish Chandrashekar and Ferat Sahin. 2014. A survey on feature selection methods. *Computers & Electrical Engineering* 40, 1 (2014), 16–28.
[10] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.
[11] Didi. 2019. Didi Website. https://www.didiglobal.com/. Jan. 28th, 2019.
[12] Ele.me. 2008. Ele.me Website. https://www.ele.me/. Dec. 31st, 2018.
[13] Michel Gendreau, Gilbert Laporte, and Daniele Vigo. 1999. Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research* 26, 7 (1999), 699–714.
[14] Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika* 30, 1/2 (1938), 81–93.
[15] Joseph B Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society* 7, 1 (1956), 48–50.
[16] Terence Lee. 1970. Perceived distance as a function of direction in the city. *Environment and Behavior* 2, 1 (1970), 40–51.
[17] Jansen C Liando, Amalinda Gamage, Agustinus W Tengourtius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks (TOSN)* 15, 2 (2019), 16.
[18] William Mendenhall, Robert J Beaver, and Barbara M Beaver. 2012. *Introduction to probability and statistics*. Cengage Learning.
[19] Gur Mosheiov. 1994. The travelling salesman problem with pick-up and delivery. *European Journal of Operational Research* 79, 2 (1994), 299–310.
[20] National Bureau of Statistics. 2017. National Bureau of Statistics Data. http://data.stats.gov.cn/. Aug. 2nd, 2018.
[21] Karl Pearson. 1900. X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50, 302 (1900), 157–175.
[22] Dennis R Proffitt. 2006. Distance perception. *Current Directions in Psychological Science* 15, 3 (2006), 131–135.

[23] César Rego, Dorabela Gamboa, Fred Glover, and Colin Osterman. 2011. Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research* 211, 3 (2011), 427–441.
[24] Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 5 (1998), 522–532.
[25] Shunfeng. 2017. Shunfeng Rush Website. http://www.sf-express.com/. Dec. 31st, 2018.
[26] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018–1021.
[27] TMall. 2003. TMall Website. https://www.tmall.com. Dec. 31st, 2018.
[28] C Reid Turner, Alfonso Fuggetta, Luigi Lavazza, and Alexander L Wolf. 1999. A conceptual basis for feature engineering. *Journal of Systems and Software* 49, 1 (1999), 3–15.
[29] Uber. 2012. Uber Website. https://www.didiglobal.com/. Jan. 28th, 2019.
[30] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. 2013. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research* 231, 1 (2013), 1–21.
[31] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. 2014. A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* 234, 3 (2014), 658–673.
[32] Shuai Wang, Tian He, Desheng Zhang, Yuanchao Shu, Yunhuai Liu, Yu Gu, Cong Liu, Haengju Lee, and Sang H Son. 2018. BRAVO: Improving the Rebalancing Operation in Bike Sharing with Rebalancing Range Prediction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 44.
[33] Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In *Icml*, Vol. 97. 412–420.
[34] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. 2016. Mobility modeling and prediction in bike-sharing systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 165–178.
[35] Desheng Zhang, Tian He, Fan Zhang, Mingming Lu, Yunhuai Liu, Haengju Lee, and Sang H Son. 2016. Carpooling service for large-scale taxicab networks. *ACM Transactions on Sensor Networks (TOSN)* 12, 3 (2016), 18.
[36] Jiawei Zhang, Xiao Pan, Moyin Li, and S Yu Philip. 2016. Bicycle-sharing system analysis and trip prediction. In *Mobile Data Management (MDM), 2016 17th IEEE International Conference on*, Vol. 1. IEEE, 174–179.
[37] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction.. In *AAAI*. 1655–1661.
[38] Meng Zhao, Jiateng Yin, Shi An, Jian Wang, and Dejian Feng. 2018. Ridesharing Problem with Flexible Pickup and Delivery Locations for App-Based Transportation Service: Mathematical Modeling and Decomposition Methods. *Journal of Advanced Transportation* 2018 (2018).

A PERCEIVED DISTANCE MODEL ANALYSIS

In this section, we apply PDM to analyze characteristics of couriers. Based on these, couriers are clustered into different groups and we have a detailed analysis for commonalities and differences between different groups.

A.1 Courier Clustering

We randomly select 132 couriers and train the model for each person. The sum of the times when the feature is used in tree node splitting is used as the feature importance[1]. The feature importance of these 132 models are analyzed. The two most important features for couriers are B_nearest and T_budget. We draw a scatter plot of models on these two features as shown in Fig.20.

Intuitively, couriers can be classified into three groups. The first group has relatively higher T_budget and lower B_nearest than other two groups. And the third group has relatively lower T_budget and higher B_nearest than other two groups. The feature values of the second group is between the first and third model. We further analyze the average working time for these three groups, and they are 127.5, 553.2 and 346 days. It seems that the courier's decision preference has a great relationship with working time.

Furthermore, we select three groups of couriers according to their working time, i.e.,junior courier, senior courier and expert courier, and train the model for each group. To make the results more distinguishable, we enlarge gap of working time between different groups. There are 1000 couriers for each group. The statistics of the three groups of couriers in one month are listed in the Tab.8. The table presents that with the courier's

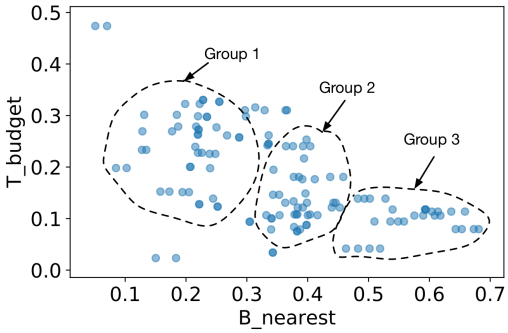


Fig. 20. Model scatter graph

working time increasing, the number of his orders increases significantly while his overdue rate decreases. This shows the gap of delivery ability between different groups of couriers is very large.

Table 8. Sample statistics

Group	Working Time	Order Amount	Overdue Rate
Junior Courier	<16 days (5% quantile)	499.0	2.6%
Senior Courier	235 days (50% quantile)	1311.9	0.5%
Expert Courier	> 827 days (95% quantile)	1872.9	0.4%

A.2 Group Characteristic Analysis

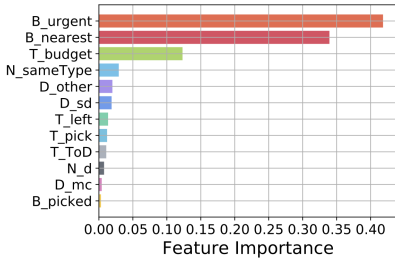


Fig. 21. Junior courier model

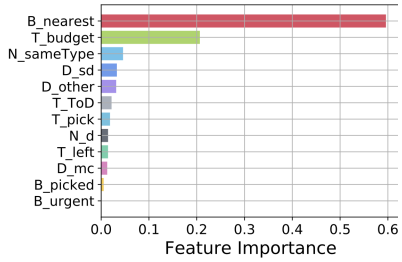


Fig. 22. Senior courier model

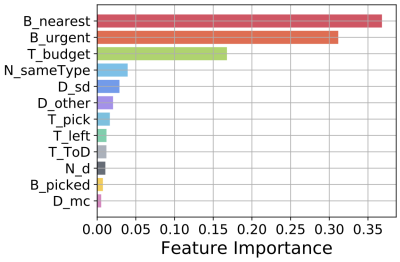


Fig. 23. Expert courier model

We then analyze group models to study the decision-making psychology of couriers. The feature importance analysis of three models is illustrated in Fig.21, Fig.22 and Fig.23. We can draw some commonalities for groups: 1) The three most significant features, which affect the decision of three groups of couriers, are those representing distance and time, i.e., B_nearest, B_urgent and T_budget. This means the courier takes both of the order's time and distance into consideration to make his decision. 2) Besides, other features take about 20% of the whole importance value, which means some other factors may also be considered by the courier to revise his decision.

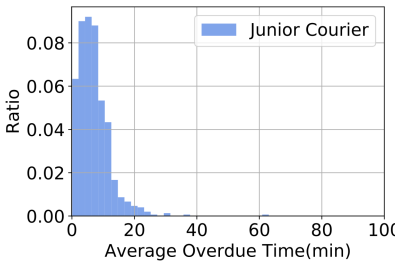


Fig. 24. Junior courier model

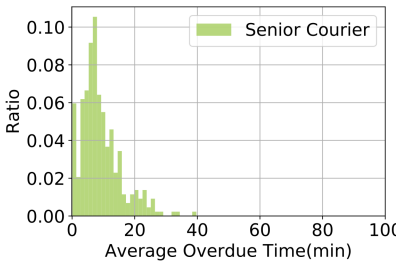


Fig. 25. Senior courier model

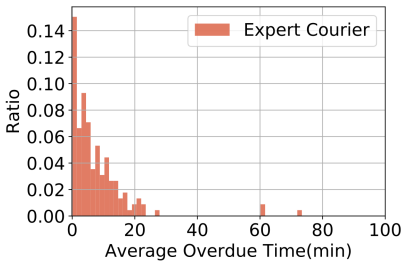


Fig. 26. Expert courier model

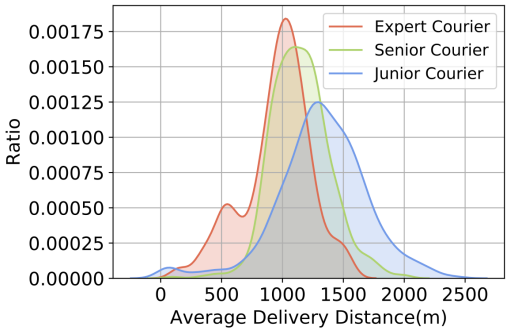


Fig. 27. PDF of average delivery distance

We then analyze the difference between three groups of couriers: 1) Junior couriers pay more attention to the order's time, while senior couriers pay more attention to the order's distance and expert couriers balance both of them. However, in spite of the prior attention to the order's time, junior couriers show a higher overdue rate. This can be explained from two aspects. a) We make a further analysis on the average delivery distance of three groups of couriers, as Fig.27 depicts. The average delivery distance of junior couriers is considerably larger than that of senior and expert couriers. This means junior couriers is not familiar with the map and their delivery experience is insufficient which leads to a high overdue rate. b) Besides, we analyze the overdue time of all the overdue orders. Fig.24 , Fig.25 and Fig.26 illustrate that most overdue time of three groups of couriers is no more than 20 minutes, which is reasonable. However, the overdue time can be up to 30-40 minutes for senior couriers and up to 60 or even 70 minutes for expert couriers, which almost never happens to the junior couriers. We speculate that when senior and expert couriers make decisions, they may give up a small amount of long-distance orders to insure that most of their orders can be delivered without overtime, which leads to the reduction of the number of overdue orders and the extension of the overdue time for some certain orders. On the other hand, junior couriers treat all orders equally without discrimination, so although the amount of overdue orders increases, the overdue time of them tends to be short.

We further analyze the difference between senior and expert couriers. Although senior couriers pay more attention to the distance of orders, but they have more delivery distance than expert couriers. As Fig.27 shows, expert couriers deliver more short-distance orders than senior couriers. It is speculated that expert couriers are more professional so they can make precise predictions on the order distribution in a certain period of time. They

often arrive at the merchant before the order is dispatched, which make their delivery with shorter distance and promotes their working efficiency. However, this ability is exactly what senior couriers lack.