Incorporating Multi-Source Urban Data for Personalized and Context-Aware Multi-Modal Transportation Recommendation

Hao Liu, *Member, IEEE*, Yongxin Tong, *Member, IEEE*, Jindong Han, Panpan Zhang, Xinjiang Lu, *Member, IEEE*, and Hui Xiong, *Fellow, IEEE*

Abstract—Transportation recommendation is one important map service in navigation applications. Previous transportation recommendation solutions fail to deliver satisfactory user experience because their recommendations only consider routes in one transportation mode (uni-modal, e.g., taxi, bus, cycle) and largely overlook situational context. In this work, we propose Hydra, a multi-task deep learning based recommendation system that offers multi-modal transportation planning and is adaptive to various situational context (e.g., nearby point-of-interest (POI) distribution and weather). We leverage the availability of existing routing engines and big urban data, and design a novel two-level framework that integrates uni-modal and multi-modal (e.g., taxi-bus, bus-cycle) routes as well as heterogeneous urban data for intelligent multi-modal transportation recommendation. In addition to urban context features constructed from multi-source urban data, we learn the latent representations of users, origin-destination (OD) pairs and transportation modes based on user implicit feedbacks, which captures the collaborative transportation mode preferences of users and OD pairs. Moreover, we propose two models to recommend the proper route among various uni-modal and multi-modal transportation routes: (1) a light-weight gradient boosting decision tree (GBDT) based recommendation model; and (2) a multi-task wide and deep learning (MTWDL) based recommendation model. We also optimize the framework to support real-time, large-scale route query and recommendation. We deploy Hydra on Baidu Maps,[1] one of the world's largest map services. Real-world urban-scale experiments demonstrate the effectiveness and efficiency of our proposed system. Since its deployment in August 2018, Hydra has answered over a hundred million route recommendation queries made by over ten million distinct users. The GBDT based model and MTWDL based model achieve 82.8 and 96.6 percent relative improvement of user click ratio, respectively.

Index Terms—Multi-modal transportation, personalized recommendation, multi-task learning, deployment

1 Introduction

Transportation recommendation is a core component in various map services and has deeply penetrated into the everyday life of citizens. Transportation recommendation refers to a set of routes recommended to users given the specific OD pair input by users. Online map services such as Baidu Maps answer over a hundred million transportation recommendation queries made by over ten million distinct users in China per day.

Despite its popularity and frequent usage, existing transportation recommendation solutions still fail to deliver satisfactory user experience. After analyzing the user query log in Baidu Maps, we find a strong requirement of inter-modal transportation comparison. For example, over 15 percent of the users in Beijing tend to request transportation recommendations on different uni-modal routing engines (e.g., taxi and bus) for the same origin and destination pair. Furthermore, 89.1 percent of routing queries from users in Beijing are answered with feasible transportation recommendations, but over 58.5 percent of the transportation recommendation list has no user clicks (see Table 1 for detail), indicating none of the recommended transportation plans is satisfactory.

The above observations indicate two limitations of current transportation recommendation solutions. *(i) Ignorance of situational context*. For instance, when a big concert lets out, it is difficult to call a taxi. A better solution may simultaneously consider multiple alternative transportation modes (as illustrated in Fig. 1) and recommend the most efficient one. *(ii) Uni-modal transportation recommendation*. For example, imagine the following scenario that the distance of the OD pair is relatively large, and the trip purpose is in no emergency. In this case, a cost-effective route that includes multiple transport modes, e.g., taxi-bus, maybe more attractive (as illustrated in Fig. 1b). Hence, the transportation recommendation should adapt to the situational context e.g., whether there is a concert, and provides more flexible recommendations, e.g., combining buses and taxis.

1. https://maps.baidu.com/

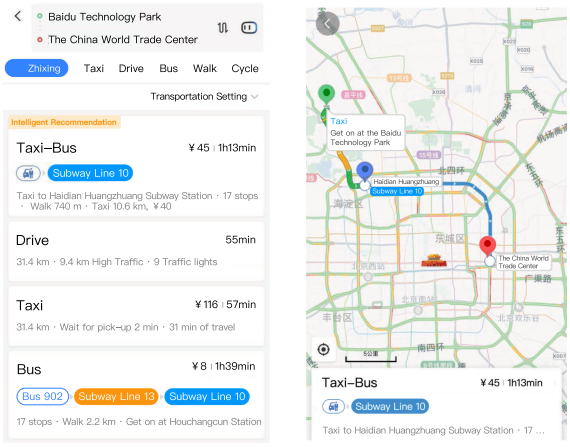
- *H. Liu, P. Zhang, X. Lu, and H. Xiong are with the Business Intelligence Lab, Baidu Research, National Engineering Laboratory of Deep Learning Technology and Application, Beijing, China. E-mail: liuhaoscut@gmail.com, {zhangpanpan, luxinjiang}@baidu.com.*
- *Y. Tong is with the SKLSDE Lab, Beihang University, Beijing 100083, China. E-mail: yxtong@buaa.edu.cn.*
- *J. Han is with the Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: hanjindong@bupt.edu.cn.*
- *H. Xiong is with the Management Science and Information Systems Department, Rutgers University, Newark NJ 07102 USA. E-mail: hxiong@rutgers.edu.*

Manuscript received 15 Oct. 2019; revised 1 Mar. 2020; accepted 30 Mar. 2020. Date of publication 10 Apr. 2020; date of current version 11 Jan. 2022. (Corresponding author: H. Xiong.)
Recommended for acceptance by Raymond Chi-Wing Wong.
Digital Object Identifier no. 10.1109/TKDE.2020.2985954

TABLE 1
Statistics of Datasets

Data description		BEIJING	SHANGHAI
User behavior data	# of queries	5,956,596	5,628,921
	# of displays	5,308,127	4,993,350
	# of clicks	2,205,091	1,980,870
Geographical data	# of POIs	900,669	1,061,399
	# of road segments	812,195	768,336
	# of bus stations	44,830	45,052
Meteorological data	# of weather records	34,944	32,760
User profile data	# of distinct users	1,199,399	1,217,140

To address these limitations, we did some preliminary work [1]. First, we propose Hydra, a personalized and context-aware multi-modal transportation recommendation system. Inspired by the availability of existing routing engines and big urban data, we design a novel framework that integrates route plans in different transportation modes (including both uni-modal and multi-modal transportation plans) and heterogeneous urban data. To the best of our knowledge, this is the first product level intelligent routing engine that integrates various transportation modes in a unified service. Second, we design GBDT based recommendation model that is adaptive to the situational context. We extract a rich set of features from multi-source urban data to sense the context variation and adopt a graph embedding based algorithm to capture the transportation preferences of users and OD pairs. Third, in web-scale recommendation, the service scalability and online recommendation latency are also curial for user experience [2]. To address the service efficiency concern, we build a distributed offline data pipeline as well as an RPC based online web service framework. Besides, we propose a dedicated region index structure in online feature processing to reduce the online recommendation latency. Extensive real-world urban-scale experiments on real datasets show that our proposed framework outperforms baseline algorithms in four metrics. The online recommendation service achieves less than 250 ms latency on average and scales well in the production environment.



(a) Recommendation list. (b) Details of the first route.

Fig. 1. An example of user interfaces of Hydra on Baidu Maps. The left figure shows the list of plans in various transportation modes ordered by our recommendation model. The right figure shows the details of the top-1 recommendation, which is a multi-modal transportation plan (i.e., first take taxi and then bus). The first recommended plan is 26.3 percent faster than the pure bus plan and 61.2 percent cheaper than the pure taxi plan.

In this paper, we further improve our Hydra framework, and deliver the following four major contributions. First, we reformulate the multi-modal transportation recommendation problem as multiple binary classification problems and adopt the multi-task learning paradigm to decide the final recommendation across different transportation modes. Second, we propose a novel deep learning based recommendation model, *Multi-Task Wide and Deep Learning* (MTWDL), which extends the well-known wide and deep model [3] to the multi-task learning paradigm in the transportation area. Compared with the lightweight GBDT based model, MTWDL is more complex but powerful. Third, we provide two deployment strategies for MTWDL (i.e., server mode and mobile mode) and discuss some deployment trade-offs in a hundred million user level online map service. Fourth, we evaluate the efficiency and effectiveness of the MTWDL model and explore its influence on the whole system. Compared with six existing baselines and two new deep learning based baselines, MTWDL achieves the best performance in four metrics and shows impressive online latency and scalability performance.

2 Data Description and Analysis

This section introduces the datasets that will be used in the following sections, with a preliminary data analysis. All user behavior data, geographical data and user profile data are acquired from Baidu Maps (https://maps.baidu.com/), a large-scale navigation app. All meteorological data are crawled from the China government website (http://www.weather.com.cn/). Table 1 summarizes the statistics of the datasets.

2.1 User Behavior Data

User behavior data captures the user interactions with navigation applications. Our user behavior data are collected from Baidu Maps, from September 2018 to November 2018. According to a user interaction loop, the user behavior data can be further categorized into *query records*, *display records* and *click records*. In short, a query record represents one route search from a user on Baidu Maps; a display record is the routes recommended by Baidu Maps shown to the user; and a click record indicates the user feedback of different recommendations (i.e., a user may click on specific routes displayed to him/her for details, as in Fig. 1). Please refer to Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2020.2985954, for a detailed data description.

We briefly explain the distributions of our user behavior dataset in BEIJING (see Fig. 2a, 2b, 2c, 2d, and 2e). Note that similar observations held in SHANGHAI, which we omit due to the page limit. Fig. 2a and 2b depict the spatial distributions of origins and destinations in the query records. As can be seen, most origins and destinations are within the 6th ring road, i.e., the central area of Beijing. We further employ Moran's I [4] to quantify the spatial auto-correlation. Specifically, the auto-correlation of origin and destination are 0.23

(a) Spatial distribution of origins. (b) Spatial distribution of destinations. (c) Temporal distribution.

(d) Distance distribution. (e) Transport mode distribution. (f) User age distribution.

Fig. 2. Distributions of the BEIJING dataset.

and 0.49, respectively. The larger auto-correlation of destinations is possibly because most queries are about specific POIs such as transport stations and city landmarks. The spatial distribution patterns of origins and destinations motivate us to use geographical data to capture the spatial dependency for transportation route recommendation. Fig. 2c plots the temporal distributions of query, display and click records (i.e., numbers per day). We observe significant temporal fluctuations where peaks often correspond to weekends and holidays. For example, the peaks on the 22nd and 31st days correspond to the mid-autumn festival and the National day, two public holidays in China. Statistically, the 1st-order and 7th-order temporal auto-correlation [5] of queries are respectively 0.42 and 0.37, indicate the strong temporal dependency and week periodicity. Fig. 2d shows the distribution of trip distance from the queries. Here the trip distance is measured by the spherical distance on earth. Over 60 percent trips are within 10 Kms and 80 percent trips are within 20 Kms. This indicates short-distance and mid-distance trips are the major query demand on online navigation applications. Fig. 2e shows the distribution of clicks on different recommended routes. Above 54.64 percent clicks involve buses (i.e., bus and bus-bicycle) and 25.12 percent clicks are drive or taxi, indicating public and car-based transportation are more preferable.

2.2 Geographical Data

Intuitively, geographical characteristics of origins and destinations partially reflect the situational context, and thus affect user preferences on transportation modes. Accordingly, we use a large-scale geographical dataset collected from *(i)* professional surveyors employed by Baidu Maps *(ii)* the crowdsourcing platform in Baidu, which include *POI data*, *road network data* and *transportation station data* in BEIJING and SHANGHAI. All data are updated daily. We present a detailed data description in Appendix A, available in the online supplemental material.

2.3 Meteorological Data

Meteorological data tend to reflect the temporal dynamics of the situational context when planning trips, and thus may also affect the user preference on transportation modes. For example, the demand for taxis may be higher in the case of snow, rain and severe air pollution. We collect the meteorological data from September 1st to November 30th. Each record of meteorological data consists of an administrative district, a time stamp, the weather, the temperature, the wind strength, the wind direction and the Air Quality Index (AQI). The weather is categorized as sunny, cloudy, rainy and overcast. The AQI is an integer of the air pollution level.

2.4 User Profile Data

User profile attributes reflect individual preference on transportation modes. For instance, subways are more cost-effective than taxis for most urban commuters, and driving is likely to be the first choice for car owners. We collect user profile attributes from multiple Baidu applications including Baidu search, Baidu App and Baidu Maps. The BEIJING dataset contains 1,199,399 distinct user records and the SHANGHAI dataset contains 1,217,140 distinct user records. Each record consists of a user's demographic attributes including the age, the gender, and social attributes such as the industry, the educational level, and whether the user is a car owner. All user profile records are anonymized and cannot be associated with sensitive personal information such as names and phone numbers. Fig. 2f plots the age distribution of BEIJING dataset. Most Baidu Maps users are between 18 and 54 years old.

3 Problem Statement and Framework Overview

In this section, we first present the problem statement, and then overview the architecture of Hydra.

3.1 Problem Statement

Let $\mathcal{M} = \{m_1, m_2, \ldots, m_k\}$ denote k different unary or multi-modal transport modes. Consider a user $u \in \mathcal{U}$, a departure time t, and an OD pair (o, d), where o and d are arbitrary geographical locations represented by a pair of longitude and latitude. Our problem is to recommend the

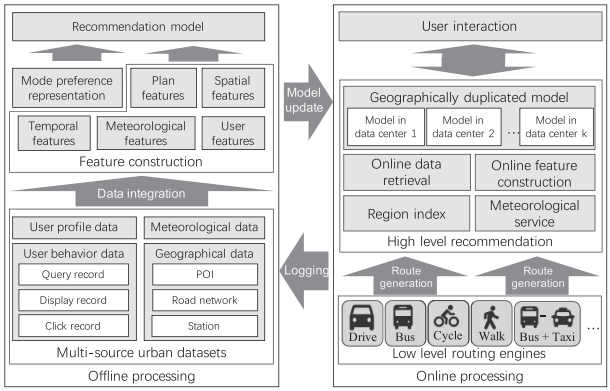


Fig. 3. Hydra Overview.

most appropriate transport mode $m_i \in \mathcal{M}$ for the user u travel between (o, d) at t.

3.2 Framework Overview

Fig. 3 shows an overview of Hydra. It consists of four major components, *Route generation*, *Feature construction*, *Transport mode preference representation* and *Transportation recommendation*. The *Route generation* module leverages existing uni-modal routing engines to generate feasible routes in different transport modes. Thereafter, the *Feature construction* module extracts features from various urban datasets. Meanwhile, the *Transport mode preference representation* module captures high-order user (resp. OD pair) transport mode preference representation through a graph embedding method. Finally, the *Transportation recommendation* module integrates hand-crafted features and embedding features to make recommendation. In this paper, we consider seven transport modes $\{drive, taxi, bus, cycle, walk, taxi\text{-}bus, bus\text{-}cycle\}$. In particular, the first five modes are uni-modal transport modes whereas $taxi\text{-}bus$ and $bus\text{-}cycle$ are multi-modal transport modes. $taxi\text{-}bus$ and $bus\text{-}cycle$ are already well supported in Baidu Maps. Besides, according to log analysis, we find the origin or destination of over 14 percent taxi queries are bus stations, and the origin or destination of over 18 percent cycle queries are bus stations. The above statistics do not mean all such queries are multi-modal trips but indicate a strong multi-modal transportation demand. Note that we treat each uni-modal and multi-modal transport mode as distinct transport modes, which makes our model extendable for other potential transport modes in a straightforward way. We left other multi-modal transport mode recommendation as future work.

4 Route Generation

We adopt existing low level routing engines to generate feasible routes for each transport mode. First, when a query is received, a station binding process is applied to bind origin and destination locations to validate start and endpoints. For example, the location is bound to road segments for drive and taxi, and to transport stations for the bus. After that, we employ a task-parallel paradigm for route candidate generation. Specifically, we initialize multiple individual threads where each thread invokes a different routing engine to generate feasible routes in the corresponding transport mode. For each uni-modal transport mode, a bidirectional shortest-path search [6] is applied to each transportation network. Besides, the contraction hierarchy (CH) [7] is pre-constructed on the transportation network to reduce search latency. A set of valid routes is generated by various criteria, e.g., fastest, distance shortest, and least transfer. For multi-modal transportation, we propose a simple yet effective substitution based heuristic [8]. In particular, to generate multi-modal routes of $taxi\text{-}bus$ and $bus\text{-}cycle$, we first generate a set of feasible bus routes based on the existing bus engine. For each bus route, we enumerate each station in the route and invoke the $taxi$ and $cycle$ engine to derive sub-routes from origin to current station and sub-routes from the current station to the destination, respectively. We concatenate the bus sub-route with the $taxi$ sub-route to generate the $taxi\text{-}bus$ route candidate, and combine the bus sub-route with the $cycle$ sub-route to generate the $bus\text{-}cycle$ route candidate. We restrict the number of modal-transfer less than two to guarantee the utility [9] of the concatenated route. The multi-modal route is added to the candidate set if it satisfies a certain criteria (e.g., faster than bus route, cheaper than the taxi route). We also build a pre-computed cache to prune infeasible mode-transfer stations for each OD pair to speed up the enumeration process. The substitution based heuristic yield about 300 ms multi-modal route generation time in average. The overall route generation process is summarized in Algorithm 1 of Appendix E, available in the online supplemental material. Finally, an internal rule based ranking model is applied in each transport mode to filter out routes with high segment overlap and decide the order of routes. For ease of cross-mode comparison, only one route of each transport mode will appear in the final display. In offline processing, the route is directly retrieved from user behavior data. In the production environment, a query understanding component will be invoked before route generation to bind fuzzy search keywords with concrete POIs. We omit further discussions since they are out of the scope of transportation recommendation.

5 Feature Construction

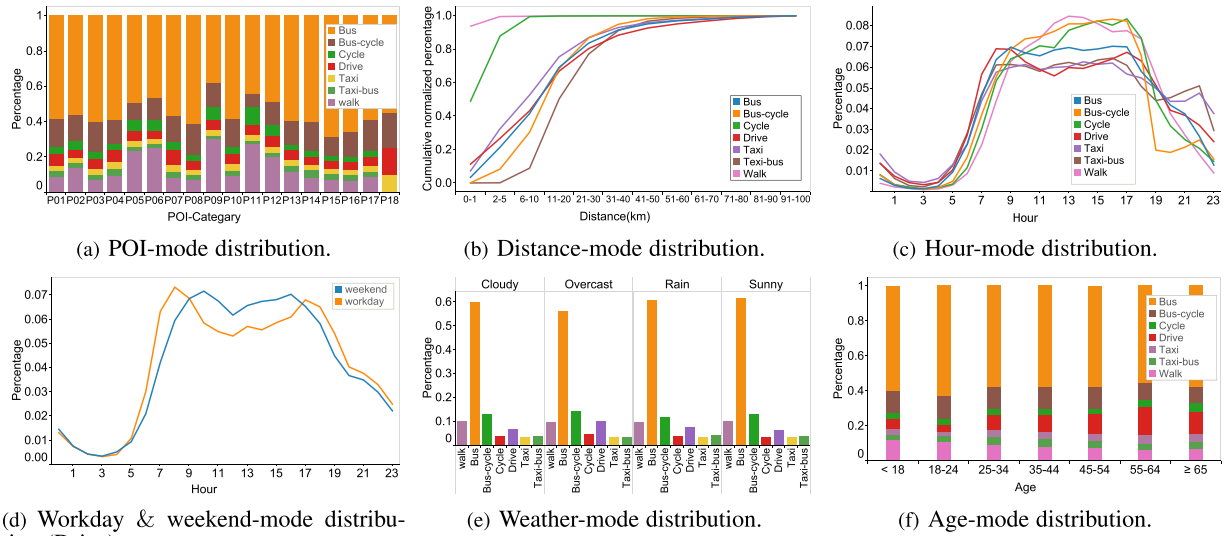
We introduce the process of constructing, transforming and augmenting feature vectors below. Appendix B, available in the online supplemental material, lists features we construct based on each dataset with a detailed description.

5.1 Plan Features

Cost of a plan such as *Price* and *ETA* are part of considerations for user preferences. For each plan, we extract *Road network distance*, *Route distance*, *ETA*, *Price*, *Transfer count*, *Transfer model count* from display records. The *Road network distance* is the real travel distance on the road network. For walking and cycling, *Price* is set to zero.

5.2 Spatial Features

We first extract *District* and *POI category* features of the origins and destinations. As shown in Fig. 4a, the transportation mode choices of different destination POI categories vary. For example, the demand for buses to *Sports* and *Tourist Attraction* POIs is higher than average. In contrast, the demand for buses to *Beauty*, *Life Service* and *Food* POIs is lower than average. The detailed POI categories in Fig. 4a



(a) POI-mode distribution. (b) Distance-mode distribution. (c) Hour-mode distribution.

(d) Workday & weekend-mode distribution (Drive). (e) Weather-mode distribution. (f) Age-mode distribution.

Fig. 4. Feature distributions of the BEIJING dataset.

are listed in Table 2 of Appendix A, available in the online supplemental material. Then we calculate the *Spherical distance* of OD pairs. Fig. 4b shows the relation between trip distance and the percentage of different transport modes. We observe a strong correlation between Spherical distance and transport mode choice. Walk and cycle are the major choices for trips shorter than 5 km whereas bus and drive are the major choices for trips longer than 10 km. The peak of demand for taxi appears when the trip distance is near 5 km. Since the road connectivity and transport stations in a region are fixed, the transport availability of adjacent OD pairs is similar. To incorporate such regional dependency, we partitioned the city into a set of non-overlapping regions through the road network [10]. For each origin region, we further compute the POI count of each POI category as *Regional POI distribution*, transport facility count (i.e., road segment, road intersection, bus station and bus line) as *Regional transport facility distribution* and mode click count as *Regional historical mode distribution*. We also extract similar features for destination regions and OD region pairs.

5.3 Temporal Features

We exploit *Hour*, *Minute*, *Day of week*, *Day of month* and *Workday* as the temporal features. As shown in Fig. 4c, distributions of transportation mode choices differ in different time periods. The demand for walk and cycle is mainly in daytime whereas the demand for taxi and taxi-bus is still high at night. As illustrated in Fig. 4d, the transport mode preferences during different time periods on weekdays and weekends also differ. For drive, there are two peak hours in a day. However, the peak on weekday mornings is earlier than that on weekend mornings and the peak on weekday evenings is later. Conversely, peak hours at weekends are closer and the demand is more evenly distributed in the daytime.

5.4 Meteorological Features

We adopt *Weather*, *Temperature*, *AQI*, *Wind speed* and *Wind direction* as the meteorology features. Fig. 4e depicts the correlation between weather and transport mode preference distributions. The demand for drive is higher on overcast and rainy days whereas the demand for bus on overcast days is lower.

5.5 User Features

We construct user features based on users' *Demographic attribute*, *Social attribute* and *User historical mode distribution*, as shown in Appendix B, available in the online supplemental material. Fig. 4f depicts the correlation between the age of users and the transport mode choices. We observe that

TABLE 2
Overall Recommendation Performance

Dataset	BEIJING				SHANGHAI			
Algorithm	NDCG	PREC	REC	F1	NDCG	PREC	REC	F1
UHP	0.29	0.159	0.207	0.18	0.288	0.162	0.188	0.174
ODHP	0.343	**0.478**	0.229	0.31	0.367	**0.454**	0.253	0.325
LR	0.802	0.255	0.681	0.371	0.789	0.262	0.652	0.374
RF	0.754	0.329	0.448	0.379	0.747	0.336	0.423	0.37
LTR	0.798	0.258	0.673	0.373	0.794	0.265	0.653	0.377
Trans2vec	0.462	0.26	0.282	0.271	0.46	0.266	0.258	0.262
WDL	0.795	0.273	0.725	0.397	0.805	0.271	0.697	0.39
DeepFM	0.781	0.263	0.713	0.389	0.785	0.266	0.695	0.384
Hydra-L (Ours)	0.798	0.271	0.72	0.396	0.804	0.274	0.685	0.391
Hydra-H (Ours)	**0.805**	0.286	**0.748**	**0.414**	**0.817**	0.278	**0.731**	**0.403**

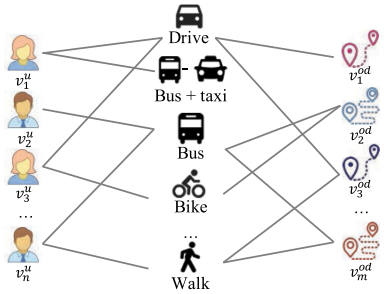


Fig. 5. An illustrative example of the heterogeneous transportation graph. Each edge indicates the frequency of a user v_i^u (resp. OD pair v_j^{od}) clicking on a route of a specific transport mode.

older people have higher demand for drive and taxi, whereas younger people prefer walk and bus more.

5.6 Transport Mode Preference Representation

Transport mode preference representation aims to learn high order collaborative relationship among users, OD pairs, and transport modes. The intuition is, users travelling similar OD pairs via similar transport modes have similar transport mode preference. Inspired by the recent success of embedding methods [11], [12] on preserving local network structures, we construct a heterogeneous graph $G = (\mathcal{V}, \mathcal{E})$ of user nodes $\mathcal{U}$, OD pair nodes $\mathcal{OD}$ and transport mode nodes $\mathcal{M}$ based on the user behavior data (Fig. 5). The target is to project each node $v \in \mathcal{V}$ into a low dimensional vector in the latent space, each of which reflects the neighborhood relationship (a.k.a. the second-order proximity) in G. We analogize the constructed click event as a sentence, where a user u clicked on a route in transport mode m over a specific OD pair od is regarded as a short sentence. We adopt Trans2vec [13] and skip-gram [14] on G. Specifically, given a click event, the latent vectors of $v^u \in \mathcal{U}$, $v^{od} \in \mathcal{OD}$ and $v^m \in \mathcal{M}$, denoted as $\mathbf{u}^u$, $\mathbf{u}^{od}$, and $\mathbf{u}^m$, are learned by maximizing the following conditional log probability:

$$O_t = \sum_{t \in T} \sum_{v_i \in V^t} \sum_{n_j^t \in N_t(v_i)} \log p(n_j^t | v_i), \tag{1}$$

where $T = \{u, od, m\}$ is the type of nodes in G, and $n_j^t \in N^t(v_i)$ is the type aware context node of v_i ever co-occurred in a click event. That is, only heterogeneous neighbour nodes are considered as valid context nodes. For example, for $v_i \in \mathcal{U}$, we have $N^t(v_i) \subseteq \{\mathcal{OD}, \mathcal{M}\}$. $p(n_j^t | v_i)$ is the conditional probability of observing type aware neighborhood $n_j^t \in G$ conditioned on the presence of v_i:

$$p(n_j^t | v_i) = \frac{e^{\mathbf{u'}_j^{\intercal} \cdot \mathbf{u}_i}}{\sum_{k=1}^{|V^t|} e^{\mathbf{u'}_k^{\intercal} \cdot \mathbf{u}_i}}, \tag{2}$$

where $\mathbf{u}'_j$ is the context representation vector of v_j as a context node and $|V^t|$ is the number of nodes with type t in graph G. To reduce the computation complexity, we employ negative sampling [14] for efficient learning. The objective function becomes:

$$O_t = \log \sigma(\mathbf{u'}_j^{\intercal} \cdot \mathbf{u}_i) + \sum_{i=1}^{K} E_{v_n^t \sim U_n(v^t)} [\log \sigma(-\mathbf{u}_n^{t\intercal} \cdot \mathbf{u}_i)], \tag{3}$$

where σ is the sigmoid function. The first term models observed edges in click events whereas the second term draws K negative edges from a uniform distribution. In this way, the distance between the learned user (resp. OD) embedding and each transportation mode embedding reflects the preference of a user (resp. OD) to each transport mode. That is, those users (resp. ODs) having similar transportation mode preference should be close to each other in the latent embedding space.

6 GBDT Based Recommendation

In this section, we introduce the Gradient Boosting Decision Tree (GBDT) based model for multi-modal transportation recommendation. We model the transport mode recommendation as a multi-class classification problem. Once the embedding vectors are learned, the proper transport mode can be derived by calculating the inner product of embedding vectors (as in [13]). In the production environment, however, the embedding method suffers from the cold-start problem. That is, 62.9 percent queries are from new users (i.e., users migrate from other routing engines and new users of Baidu Maps) or target to new OD pairs (i.e., OD pairs which have not been queried by users). To handle such cases, we concatenate the learned embedding vector of the user and the OD pair with the handcrafted features (as in Section 5) into a d dimensional feature vector.

Given a preprocessed dataset of n instances, m transport modes and d feature dimensions, we transform the raw data into a 2D matrix $\mathcal{D} = \{\mathbf{x}_i, y_i\}$ where $|\mathcal{D}| = n$, $\mathbf{x}_i \in \mathcal{R}^d$ is the feature vector and $y_i \in \mathcal{R}^{\mathcal{M}}$ is the ith transport mode. We employ the gradient boosting tree [15] as our recommendation model because gradient boosting tree based algorithms [16] are suited for data mining with sparse and high dimensional features. Specifically, we sequentially generate a set of tree classifiers $\mathcal{F}(\cdot) = \{f_1(\cdot), f_2(\cdot), \ldots, f_k(\cdot)\}$ and ensemble the result of each classifier to generate the overall predictive result.

$$\hat{y}_i = \mathcal{F}(\mathbf{x}_i) = \sum_{j=1}^{k} f_j(\mathbf{x}_i), f_j \in \mathcal{F}, \tag{4}$$

where $\hat{y}_i$ is the estimated transport mode of ith instance, $f(\cdot)$ is a softmax regressor for multi-class classification:

$$f(\mathbf{x}_i) = \frac{e^{\mathbf{w}_q^{\intercal} \mathbf{x}_i}}{\sum_{p=1}^{|\mathcal{M}|} e^{\mathbf{w}_p^{\intercal} \mathbf{x}_i}}, \tag{5}$$

where $\mathbf{w}_q$ is the parameter vector of the qth class. The learning objective is to minimize

$$O = \sum_{i=i}^{n} l(y_i, \hat{y}_i) + \frac{\lambda_1}{2} \sum_{j}^{k} \|\mathbf{w_j}\|_1 + \frac{\lambda_2}{2} \sum_{j}^{k} \|\mathbf{w_j}\|_2, \tag{6}$$

where $l(\cdot)$ is is the cross-entropy loss, λ_1 and λ_2 are hyper-parameters for $L1$ and $L2$ regularizations, respectively.

The gradient of the tree function is derived much harder than traditional optimization tasks. Since we train classifiers sequentially, we approximate the gradient

based on the previous step. The objective at the tth iteration becomes

$$\widetilde{O}_i = \sum_{i=1}^{n}(g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)) + \frac{\lambda_1}{2}\sum_{j}^{k}\|\mathbf{w_j}\|_1 + \frac{\lambda_2}{2}\sum_{j}^{k}\|\mathbf{w_j}\|_2, \tag{7}$$

where $g_i = \partial_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1})$ and $h_i = \partial^2_{\hat{y}_i^{t-1}} l(y_i, \hat{y}_i^{t-1})$ are the first order and second order gradient statistics of $l(\cdot)$. The detailed deduction can be found in [17].

7 MTWDL Based Recommendation

In practice, the GBDT based model provides a light-weight yet effective recommendation service. However, tree-based model is less powerful on extracting high-order feature representations from large-scale data, which limit its expressive power on transport mode recommendation. In this section, we introduce a multi-task wide and deep learning (MTWDL) based model for multi-modal transportation recommendation. Compared with the tree-based model, the advantages of MTWDL are three folds. First, by stacking multiple neural network layers, MTWDL is capable of capture high order feature interactions [18]. Second, except latent representations learned from the transportation mode preference representation module, MTWDL further introduces an embedding layer to learn low dimensional representations of high-dimensional categorical features. Third, MTWDL formulates transport mode recommendation and user click prediction as multiple individual tasks and introduce a multi-task learning framework to further improve the recommendation performance.

7.1 Tasks Definition

In the GBDT based model, the multi-class formulation assigns instances without click to a negative class. However, this formulation doesn't distinguish the difference between user click behavior and specific clicked transport mode. To this end, we define multiple related main tasks $\{T^{m_1}, \ldots, T^{m_k}\}$ as well as an auxiliary task T^a, and learning to optimize them simultaneously. Given the feature matrix $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^n$ and the corresponding labels $\{y_i\}_{i=1}^n$, we first define each main task. For task T^{m_i}, we aim to learn a binary classifier f^{m_i} to predict if transport mode i is preferred for each instance,

$$\hat{y}_i^{m_j} = f^{m_j}(\mathbf{x}_i), \tag{8}$$

where $\hat{y}_i^{m_j}$ is the estimated click likelihood of transport mode j for the ith instance. Similarly, the auxiliary task aims to learn a binary classifier f^a to predict if the user click any transport mode,

$$\hat{y}_i^a = f^a(\mathbf{x}_i), \tag{9}$$

where $\hat{y}_i^a$ is the estimated likelihood if user would click the ith instance.

7.2 Basic Model

We first introduce the deep learning based model for each individual task. We adopt the wide and deep learning model [3], which is widely used in many recommender systems. As shown in Fig. 6, the wide and deep learning consists of two components: the *wide component* for user preference memorization and the *deep component* for high-order feature generalization.

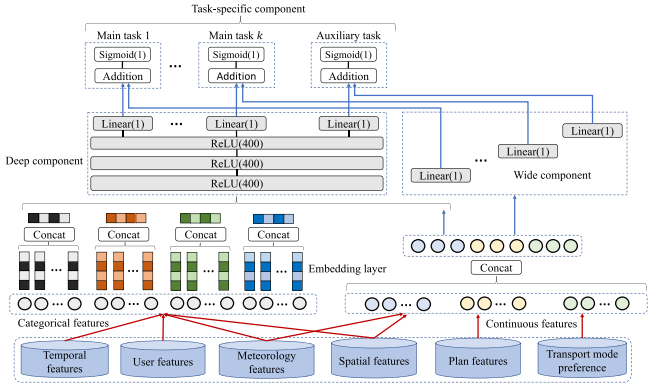


Fig. 6. Architecture of MTWDL. It jointly trains multiple main tasks and an auxiliary task, the embedding layer and deep component are shared among different tasks.

Specifically, the wide component is designed for feature co-occurrence memorization, defined as

$$\hat{y}_j^{m_i} = \mathbf{w}^\top \mathbf{x}_j + b, \tag{10}$$

where $\mathbf{x}_j$ is the input feature vector, $\mathbf{w}$ is the learnable weighted matrix and b is the bias.

The deep component stacks multiple neural network layers to capture higher order feature representations. An embedding layer is first applied to transform categorical features into low dimensional dense vectors. Then we concatenate all dense embedding vectors and continuous features and feed the concatenated vector into several fully connected layers. Each fully connected layer transform input vector as follow

$$\mathbf{z}^{(l+1)} = ReLU(\mathbf{w}^{(l)\top}\mathbf{z}^{(l)} + b^{(l)}), \tag{11}$$

where $\mathbf{z}^{(l)}$ and $\mathbf{z}^{(l+1)}$ are the input and output of lth layer, $\mathbf{w}^{(l)}$ and $b^{(l)}$ are parameters of layer l, and $ReLU$ is the rectified linear units as the activation function. The deep component explores new feature combinations to improve model generalization power.

The final output is a combination of the wide component and the deep component,

$$\hat{y}_i = \sigma(\mathbf{w}_w^\top \mathbf{x}_j + \mathbf{w}_d^\top \mathbf{z}^{(l_f)} + b), \tag{12}$$

where $\hat{y}_i$ is the final output, σ is the activation function, $\mathbf{w}_w$ is the parameter of the wide component, $\mathbf{w}_d$ is the parameter of the final output of the deep component $\mathbf{z}^{(l_f)}$. An Adam [19] optimizer is employed for optimization.

7.3 Multi-Task Wide and Deep Recommendation

In general, there are two paradigms of multi-task learning, the parameter sharing based approach and the constraint based approach [20]. In this paper, MTWDL follow the first paradigm, where lower level parameters in the deep component are shared cross all tasks. The parameter sharing mechanism further improves the generalization power of the model. As shown in Fig. 6, MTWDL consists three

components, the *wide component*, the *deep component*, and the *task-specific component*. Specifically, all tasks share the deep component and each task has an individual wide component and a task-specific component. For each task, the wide component and the deep component are identical with the basic model. Each task has two sets of features, general features and task-specific features. General features such as temporal features and meteorological features are identical for all tasks, whereas task-specific features such as ETA, distance, price are different in different tasks.

The difference between MTWDL and the basic model is there are multiple output layers for different tasks. For example, for task T^{m_i}, the output layer is defined as

$$\hat{y}_i = \sigma(\mathbf{w}_w^{m_i\top}\mathbf{x}_j + \mathbf{w}_d^{m_i\top}\mathbf{z}^{(l_f)} + b), \tag{13}$$

where $\mathbf{w}_w^{m_i}$ and $\mathbf{w}_d^{m_i}$ are the task specific parameters of the wide component and the deep component, respectively.

7.4 Objective

In MTWDL, all learnable parameters are optimized jointly. For the main task T_{m_j}, the objective is defined as

$$L_{m_j} = -\frac{1}{n}\sum_{i=1}^{n} \alpha y_i^{m_j} \log \hat{y}_i^{m_j}, \tag{14}$$

where $y_i^{m_j} \in \{0, 1\}$ indicates if a user click transport mode m_j or not, α is a hyper-parameter to alleviate class imbalance.

For auxiliary task, the objective is defined as

$$L_a = -\frac{1}{n}\sum_{i=1}^{n} \beta y_i^a \log \hat{y}_i^a, \tag{15}$$

where $y_i^a \in \{0, 1\}$ indicates if a user click any transport mode or not, β is a hyper-parameter to trade-off the click ratio and the recommendation coverage ratio.

Beside the objectives for each task, we consider the relationship between main tasks and the auxiliary task. Consider the output of each main task represents the likelihood a user click on a transport mode m_j, denoted by $P_{m_j}(\mathbf{x}_i)$. The estimated probability a user click on any transport mode is

$$Q(\mathbf{x}_i) = 1 - \prod_{m_j \in \mathcal{M}} (1 - P_{m_j}(\mathbf{x}_i)). \tag{16}$$

We aim to minimize the Jensen-Shannon divergence [21] between $Q(\mathbf{x}_i)$ and the probability a user click on the transport mode $P_a(\mathbf{x}_i)$,

$$L_r = \frac{1}{2} D_{KL}(P(\mathbf{x}_i)||Q(\mathbf{x}_i)) + \frac{1}{2} D_{KL}(Q(\mathbf{x}_i)||P(\mathbf{x}_i)), \tag{17}$$

where $P(\mathbf{x}_i)$, $D_{KL}(P(\mathbf{x}_i)||Q(\mathbf{x}_i))$ and $D_{KL}(Q(\mathbf{x}_i)||P(\mathbf{x}_i))$ are defined as

$$P(\mathbf{x}_i) = \frac{P_a(\mathbf{x}_i) + Q(\mathbf{x}_i)}{2} \tag{18}$$

$$D_{KL}(P(\mathbf{x}_i)||Q(\mathbf{x}_i)) = \sum_{i=1}^{n} P(\mathbf{x}_i) \log \frac{P(\mathbf{x}_i)}{Q(\mathbf{x}_i)}, \tag{19}$$

$$D_{KL}(Q(\mathbf{x}_i)||P(\mathbf{x}_i)) = \sum_{i=1}^{n} Q(\mathbf{x}_i) \log \frac{Q(\mathbf{x}_i)}{P(\mathbf{x}_i)}. \tag{20}$$

By considering all task specific objective and the Jensen-Shannon divergence loss, we aims to optimize the following objective function

$$L = \sum_{m_j \in \mathcal{M}} L_{m_j} + L_a + \frac{\lambda}{n} L_r, \tag{21}$$

where λ is a hyper-parameter controls the importance of the Jensen-Shannon divergence loss.

8 Deployment

Hydra has been deployed on Baidu Maps. In this section, we describe the implementation and deployment details. cost of the mapping process is much lower than that of R-tree.

8.1 Offline Processing

Due to the complex data dependency, we propose an automatic pipeline for data integration and feature engineering. We employ Bigflow[2] as the offline data pipeline platform. Bigflow is an open source programming abstraction that allows for programming and processing data on various distributed computing engines (e.g., Hadoop Tez [22] and Spark [23]). In Bigflow, a set of data wrangling operators such as *map*, *filter* and *join* is well supported and the lower level distributed operations are transparent to users.

8.1.1 GBDT Training

We use the XGBoost library[3] to train the GBDT based model. The GBDT based model is updated on daily basis to take new data into consideration. To exclude seasonal changes, we define a three-month sliding time window for training data selection. Once the data pipeline is finished, the model update script is triggered to update the model.

8.1.2 MTWDL Training

We use the PaddlePaddle[4] platform to implement the MTWDL model. PaddlePaddle is an efficient and scalable deep learning platform, which is supporting a variety of AI empowered products at Baidu. In the deep component, the embedding layer first transforms each categorical feature into a 32-dimensional embedding vector and concatenates them with all the continuous features as the input vector. The input vector is then fed into three fully connected layers. Each fully connected layer consists of 400 hidden units and a ReLU activation function. The hyper-parameters α, β, λ, learning rate and dropout rate are set to 1.0, 10.0, 0.5, 0.0001, and 0.5, respectively. The batch size is 512. The output of the last fully connected layer is used as the input features of the multi-task component. To accelerate model training, in daily update, the embedding parameters are initialized from latest model.

8.2 Online Processing

Baidu Maps answers billions of queries in each day. Thus, it is crucial to offer effective and scalable online service to users. To this end, we build efficient region index and

2. http://bigflow.baidu.com
3. https://xgboost.readthedocs.io/en/latest/
4. https://github.com/PaddlePaddle

scalable prediction service to enable low latency and high throughput online service.

8.2.1 Region Index

For online feature processing, a batch of statistical features is required to be mapped from coordinates to regions (e.g., join the origin coordinates of a query with the regional POI distribution). Traditional spatial index, such as R-tree, requires $O(\log n)$ search time, which is time consuming for cities with a large number of regions. We proposed a dedicated region index to speed up such mapping process. Specifically, we divide the city into fine-grained grids based on coordinates with a unique grid id. We then allocate regions to the corresponding grids. Note that each region is an irregular polygon, therefore, a grid may be intersected with one or multiple regions. For example, the minimum bounding rectangle (MBR) $[(116.30, 40.05), (116.31, 40.06)]$ is partitioned to grid g_1, with id 11630_4005. If there are two regions r_1 and r_2 intersect with g_1, the index in the database is stored as a key-value pair $(11630\_4005, [r_1, r_2])$, where the value is a list of regions. Internally, the grid-regions pair is stored as a hash table in Redis. Since the region is partitioned based on the road network, most grids are only associated with one or a few regions. In practice, the average time

8.2.2 GBDT Prediction

We build the web service based on BRPC (https://github.com/brpc/brpc), a scalable Remote Procedure Call (RPC) framework used throughout Baidu. The GBDT model is duplicated in four data centers distributed over China to reduce network latency of the service. Specifically, the online service contains three components. First, retrieve geographical information, meteorological data, user profile data in parallel and integrate them with raw route plans. Second, execute the online feature engineering process by leveraging the metadata generated in the offline data pipeline. Third, feed the processed feature vector into the model, sort each mode by model score and return the transport mode with the highest score to the user. About 6 percent of transport modes with the highest score have no corresponding plan. Instead, we recommend the next transport mode that has a feasible plan.

8.2.3 MTWDL prediction

We provide two modes for MTWDL based prediction, i.e., the server mode and the mobile mode. The server mode runs the model on a server in our data centers, whereas the mobile mode runs the model on mobile phones along with the navigation app. Specifically, the server mode implements a high-performance parallel predictor *AnalysisPredictor* in C++, and adopt the *ZeroCopyTensor* mechanism in PaddlePaddle to avoid redundant data copy operation. The mobile mode implement a lightweight predictor *MobilePredictor* in both C ++ and Java. Similar to the GBDT prediction, both modes need first request meteorological data from existing online service, retrieve geographical data and user profile data from Redis services, and integrate them with raw route plans. In the production environment, we finally choose the server mode because of the following three reasons. 1) Route plans and corresponding raw features are computed and retrieved on the server-side; the network cost of server mode is smaller than the mobile mode. 2) Routing request is a relatively low-frequency service (several times in each day), the model in mobile mode need load in and flush out before and after query, which induces longer prediction time than server mode. 3) Since our model is updated on a daily basis, mobile mode requires frequent app updates.

9 Experiments

9.1 Experimental Setup

We conduct experiments on the datasets described in Section 2. We mainly focus on (1) the overall performance, (2) each feature contribution, (3) parameter sensitivity and (4) the robustness of our approach. We also present the user satisfaction analysis and the efficiency and scalability of our system. We split data from September 1 to November 10 as training set, November 11 to November 20 as validation set, and the remaining as testing set.

Metrics. We adopt the overall NDCG [24], weighted precision, recall and F1 metrics to evaluate the performance. The NDCG metric takes all transport modes into consideration whereas the rest metrics only care about the top-1 recommendation.

Baselines. We compare our approach with eight baselines and two variants of Hydra.

- *UHP* recommends the transportation mode of route using the fraction of user historical preference. The most common transport mode choice of the user will be recommended.
- *ODHP* recommends the transportation mode of route using the fraction of OD historical preference. The most popular transport mode between the OD pair will be recommended.
- *LR* recommends the transportation mode of route via the well-known logistic regression model. The input feature is same with our method as described in Section 5.
- *RF* recommends the transportation mode of route using Random Forest. The input feature is same to our method as described in Section 5.
- *LTR* is a popular LambdaMart [25] learning to rank method, where the pairwise loss is minimized. We use the plan feature described in Section 5 as input.
- *Trans2vec* is the state-of-the-art transportation mode recommendation method [13] based on graph embedding. It makes recommendation based on the inner product of user vector and transportation mode vector and the inner product of OD pair vector and transportation mode vector.
- *WDL* is the original wide and deep learning framework [3]. It integrates a wide linear model and a deep neural network. We use a softmax layer to obtain the model output.
- *DeepFM* is a state-of-the-art recommendation model [26] that combines factorization machine for learning feature interactions and the power of the deep neural network.
- *Hydra-L (ours)* is the light-weight version of Hydra where the high level model is the gradient boosting decision tree based recommendation model.

TABLE 3
Top-10 Features Ranked by Information Gain (Hydra-L)

Rank	Feature name	Relative gain
1	Walk ETA	1
2	Bus-cycle ETA	0.803
3	Bus ETA	0.577
4	Taxi-bus ETA	0.451
5	User walk percentage	0.295
6	Consumption level	0.213
7	Origin station count	0.162
8	Primary POI category	0.096
9	Hour	0.092
10	Spherical distance	0.051

TABLE 4
Top-10 Input Perturbation Feature Importance (Hydra-H)

Rank	Feature name	Relative gain
1	Bus ETA	1
2	Hour	0.669
3	Weather	0.4
4	Walk ETA	0.337
5	Walk distance	0.336
6	Age	0.313
7	Income level	0.274
8	Bicycle distance	0.24
9	Car ETA	0.188
10	Taxi-bus price	0.15

- *Hydra-H (ours)* is the heavy-weight version of Hydra where the high level model is the multi-task wide and deep learning based recommendation model.

9.2 Overall Recommendation Result

Table 2 depicts the overall results of our methods and all the compared baselines with respect to four evaluation metrics. We can make the following observations. (1) Hydra-H achieves better performance than other methods over all metrics except PREC, which indicates the effectiveness of our models. Although ODHP and RF achieves higher PREC score, Hydra-H achieve better balance between PREC and REC, which is evaluated by F1. (2) Hydra-H consistently outperforms Hydra-L in terms of all metrics, demonstrates the effectiveness of the MTWDL framework. Specifically, Hydra-H achieves 4.55 and 3.07 percent improvement over Hydra-L of F1 on BEIJING and SHANGHAI, respectively. The improvement of REC are 3.89 and 6.72 percent on BEIJING and SHANGHAI. Note that although Hydra-L performs worse than Hydra-H, it outperforms other non-deep-learning based model and is more time efficient than Hydra-H. Hydra-H takes about ten times longer training time than Hydra-L. (3) Hydra-L is competitive against WDL and DeepFM, which matches our expectation that situational context information and tailored feature engineering is curial for multi-modal transportation recommendation. (3) WDL and DeepFM outperform six other baselines, illustrate the effectiveness of the deep learning based model. (5) The performance of solely Trans2vec is not well on the dataset with large proportion of cold-start users (resp. OD pairs). Overall, incorporating handcrafted features and high-order embedding features with a multi-task wide and deep learning framework outperforms all other baselines.

9.3 Feature Importance Analysis

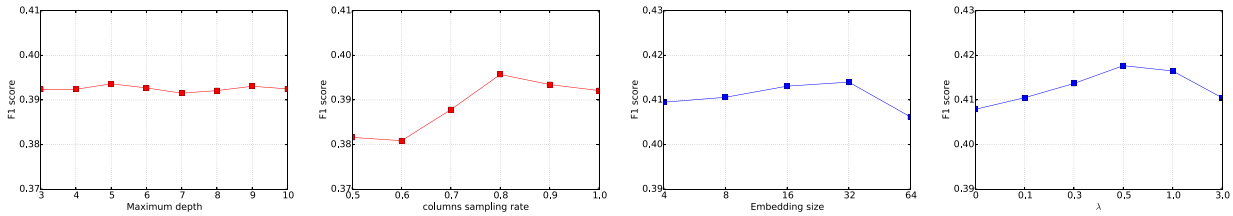
To evaluate the effectiveness of our feature construction, we exam the importance of each feature in our models. For the gradient boosting decision tree based model, we rank features by information gain [27]. The higher information gain indicates higher frequency the feature used to split nodes in each individual tree. Table 3 reports top-10 features and their relative information gain. The top 4 features are all plan *ETA* of corresponding modes, which meets our expectation that travel time is the major consideration in the transport mode choice. Besides, we observe user attributes especially user social attributes such as historical mode preferences (walk preference in rank 5) and consumption level (rank 6) also make significant contributions for transport mode prediction. Features from rank 7 to rank 10 are spatial features and temporal features, which validates our intuition that the spatial and temporal dependency influences the transport mode choice. Since we cannot directly obtain information gain from deep neural network, we apply perturbation feature ranking [28] to evaluate the importance of each feature in MTWDL. The importance of each feature is measured by calculating the performance loss of MTWDL when the corresponding feature column is shuffled. Table 4 reports top-10 features and their relative increase in loss. As can be seen, plan feature *Bus ETA* is the most important feature, but the temporal feature *Hour* becomes the second feature and the meteorological feature *Weather* becomes the third feature. User profile features *Age* and *Income level* are ranked at 6 and 7, respectively. Compared with the GBDT based model, more categorical features contribute more on multi-modal transportation recommendation, which shows the advantage of MTWDL on handling high dimensional categorical features.

9.4 Parameter Sensitivity Analysis

We further study the parameter sensitivity of Hydra-L and Hydra-H. we evaluate the performance of different hyper-parameters on BEIJING dataset, the results on SHANGHAI are similar.

For Hydra-L, we report the influence of maximum depth and columns sample rate, two important parameters in tree-based model. First, we vary the maximum depth from 3 to 10. The results are reported in Fig. 7a. In general, Hydra-L performs stable when the maximum depth changes. Hydra-L achieves best performance when maximum depth is 5. Then, we vary the column sampling rate from 0.5 to 1.0. The results are reported in Fig. 7b. As can be seen, there is a performance improvement when we increase the sampling rate from 0.5 to 0.8, but the performance degrades when we further increase the sampling rate from 0.8 to 1.0. The reason is proper column sampling can avoid overfitting, but too low sampling rate limit the model to capture useful feature combinations.

For Hydra-H, we report the influence of embedding size and the regularization parameter λ. We first vary the embedding size in Hydra-H from 4 to 64. As shown in Fig. 7c, the model achieves the highest F1 score when the embedding size is 32. Set embedding size to 16 or 32 is



(a) Effect of Maximum depth. (b) Effect of column sampling rate. (c) Effect of embedding size. (d) Effect of λ.

Fig. 7. Parameter sensitivities on BEIJING dataset.

enough to represent information in each categorical feature. Fig. 7d depicts the influence of λ. We observe a performance improvement when we improve λ from 0 to 0.5 and a performance degradation when we further improve λ from 0.5 to 3.0. The result validates modeling relationship between each tasks can improve the model performance.

9.5 Robustness Check

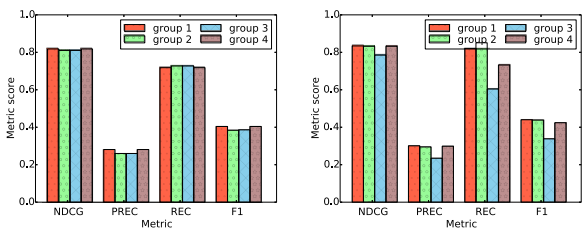
A robust algorithm should perform evenly on different subgroups of queries. We group queries from two perspectives: 1) user profile perspective, and 2) OD profile perspective. For 1), we segment users through gender and age, i.e., women and age lower than 35, men and age lower than 35, women and age older than 35, men and age older than 35. For 2), we segment OD pairs based on region functionality (i.e., we use the POI distribution of corresponding regions), classical K-means is applied to cluster OD pairs into four disjoint groups. Figs. 8 and 9 illustrate the performance of the GBDT based model and the MTWDL based model on different subgroups on BEIJING, respectively. The results on SHANGHAI are similar. For different groups of users, the results are strongly stable on four metrics, which validates the robustness of our method for different users. For different OD pairs, the results of the GBDT based model are also stable on four metrics expect the third group (e.g., for REC, the difference is over 10 percent). This result indicates the variation from the OD profile perspective is more significant. As shown in Fig. 9b, MTWDL based model improves the performance of the third group and mitigates its impact in the overall result.

9.6 User Interview and Online Test

The model has been deployed on Baidu Maps since mid 2018. In past months, the model has answered over a hundred million route planning requests and served over ten million distinct users. To assess the user satisfaction of model recommendations, we published survey questionnaires to frequent Baidu Maps users. Overall, 738 valid questionnaires are collected. In the questionnaire, we set five level satisfaction categories, $G+, G, S, B, B+$, where G stands for good, S stands for same as before, B stands for bad. As shown in Table 5, over 86.7 percent users think the recommendation result is better than before, only 1.6 percent users think the recommendation result becomes worse. That is, our method provides better recommendations in terms of user experience. Moreover, we conduct an online A/B test in the production environment. Hydra-H achieves 2.78 percent click ratio improvement over Hydra-L. The improvement means over 300,000 more queries are satisfied by Hydra-H, which is a significant gain for an online product.

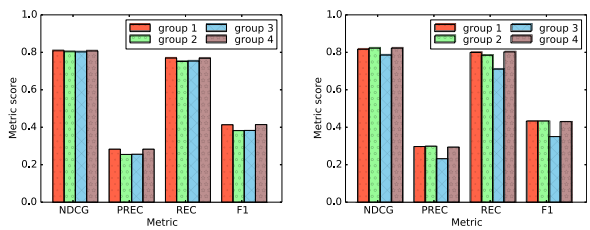
9.7 Efficiency and Scalability

Finally, we evaluate the efficiency and scalability. We randomly test 1,000 queries, and the averaged recommendation time of each baseline and our model are reported in Fig. 10a. As can be seen, learning-based models generally take a longer time than statistical baselines. In particular, deep learning models take longer time than statistical learning models. We further test the query response latency of our framework in the production environment. The query response latency is composed of two parts, low level routing cost and high level recommendation cost. For high level recommendation, we evaluate the latency of GBDT based model in Hydra-L and MTWDL based model in Hydra-H. The results are reported in Fig. 10b. On one hand, when we vary the query per second (QPS) from 1 to 10,000, the low level routing latency increased from 220 ms to 671 ms. On the other hand, the latency of GBDT based model increased from 5 ms to 274 ms whereas the latency of MTWDL based model increased from 73 ms to 566 ms. Above observations demonstrate that although Hydra-H outperforms Hydra-L in terms of recommendation accuracy, it is more time-consuming. Notice that the efficiency gap goes large when the QPS is too small or too large, but the latency gap is relatively small when the QPS is between 100 and 1,000. Since the peak QPS of the online service is less than 1,000, Hydra-



(a) Group by users (b) Group by OD pairs

Fig. 8. Robustness check on the BEIJING dataset (Hydra-L).



(a) Group by users (b) Group by OD pairs

Fig. 9. Robustness check on the BEIJING dataset (Hydra-H).

TABLE 5
User Satisfaction

Level	G+	G	S	B	B+
Percentage	46.3%	40.4%	11.7%	0.9%	0.7%

H is still applicable and the online workload can be well handled by Hydra-H. Overall, the low level routing is the major bottleneck in Hydra-L and we should pay more attention on the recommendation cost in Hydra-H.

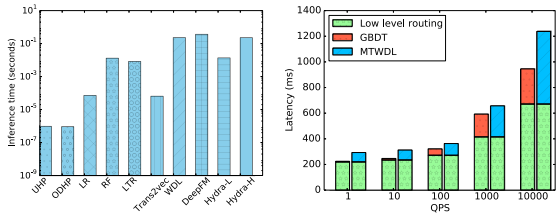
10 Related Work

Route Recommendation. Route recommendation has attracted much attention from both academia (e.g., [29]) and industries (e.g., Google Maps and Baidu Maps). A common routine of route recommendation is to apply the algorithms of shortest distance queries [30] with predefined cost functions [31]. As another important direction, the quality of recommended routes can be improved by leveraging large-scale historical trajectories [32]. Specifically, T-Drive [33] captures the intelligence of taxi drivers via a landmark graph. Dai *et al.* [34] recommends routes by considering personal preference (e.g., time efficiency or fuel efficiency) for each individual driver. Recently, the route recommendation for shared mobility also attracted research interest to improve efficiency [35] and revenue [36]. However, all of them consider uni-modal route recommendations and thus cannot be directly applied for multi-modal route recommendation. Trans2vec [13] considers multi-modal recommendation by learning embedding of users, OD pairs and transport modes. But it suffers from the cold-start problem and requires extra models or strategies to handle new instances.

Urban Computing. With the development of city urbanization, various data generated from GPS, sensors, buildings and humans has been applied to tackle various urban issues. For example, Yu *et al.* [37] predict urban safety by considering multiple spatial and temporal factors. Moreover, Tong *et al.* [38] and Xia *et al.* [39] predicts taxi demands based on multi-sourced urban data. Sun *et al.* [40] mines the urban region-of-interest through map search queries. Motivated by the above studies, we integrate multiple urban datasets to improve the performance of route recommendation among various transport modes. To the best of our knowledge, it is the first work that integrates multiple sources of urban data for route recommendation among various transport modes in a data-driven way at urban-scale.

11 Conclusion

In this paper, we presented Hydra, a personalized and context-aware multi-modal transportation recommendation system. It is a two-level system that adaptively recommends uni-modal and multi-modal transportation routes according to the user preferences and the situational context. We first extracted a rich set of features from user behavior data and several urban data collectged from other sources. Next, we learnt embedding features via the heterogeneous transportation graph to enhance the recommendation performance. Moreover, a gradient boosting tree based model as well as a multi-task wide and deep learning based model was respectively devised for multi-modal transportation recommendation. Finally, we discussed several deployment issues to optimize Hydra to be scalable, including offline data pipelines, high performance spatial index, as well as the construction of web service framework. Extensive evaluations on real-world datasets validate the effectiveness and efficiency of Hydra.



(a) Recommendation time (b) Online ;latency

Fig. 10. Results of efficiency and scalability.

References

[1] H. Liu, Y. Tong, P. Zhang, X. Lu, J. Duan, and H. Xiong, "Hydra: A personalized and context-aware multi-modal transportation recommendation system," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 2314–2324.

[2] N. Tolia, D. G. Andersen, and M. Satyanarayanan, "Quantifying interactive user experience on thin clients," *Computer*, vol. 39, no. 3, pp. 46–52, Mar. 2006.

[3] H.-T. Cheng *et al.*, "Wide & deep learning for recommender systems," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[4] H. Li, C. A. Calder, and N. Cressie, "Beyond moran's I: Testing for spatial dependence based on the spatial autoregressive model," *Geogr. Anal.*, vol. 39, no. 4, pp. 357–375, 2007.

[5] E. Parzen, "On spectral analysis with missing observations and amplitude modulation," *Sankhyā: The Indian J. Statist., Series A*, vol. 25, no. 4, pp. 383–392, 1963.

[6] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A search meets graph theory," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2005, pp. 156–165.

[7] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, "Contraction hierarchies: Faster and simpler hierarchical routing in road networks," in *Proc. Int. Workshop Exp. Efficient Algorithms*, 2008, pp. 319–333.

[8] J. Dibbelt *et al.*, "Engineering algorithms for route planning in multimodal transportation networks," KIT, Karlsruhe, PhD Thesis, 2016.

[9] D. Delling *et al.*, "Computing multimodal journeys in practice," in *Proc. Int. Symp. Exp. Algorithms*, Springer, Berlin, Heidelberg, 2013, pp. 260–271.

[10] N. J. Yuan, Y. Zheng, and X. Xie, "Segmentation of urban areas using road networks," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-2012–65, 2012.

[11] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016 pp. 855–864.

[12] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 701–710.

[13] H. Liu, T. Li, R. Hu, Y. Fu, J. Gu, and H. Xiong, "Joint representation learning for multi-modal transportation recommendation," in *Proc. Assoc. Advancement Artif. Intell.*, 2019, pp. 1036–1043.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[15] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002.

[16] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.

[17] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.
[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[20] Y. Zhang and Y. Qiang, "An overview of multi-task learning," *Nat. Sci. Review*, vol. 5, no. 1, pp. 30–43, 2018.
[21] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, 1991.
[22] B. Saha, H. Shah, S. Seth, G. Vijayaraghavan, A. Murthy, and C. Curino, "Apache tez: A unifying framework for modeling and building data processing applications," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1357–1369.
[23] M. Zaharia *et al.*, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proc. 9th USENIX Conf. Networked Syst. Des. Implementation*, 2012, p. 2.
[24] Y. Wang, L. Wang, Y. Li, and D. He, "A theoretical analysis of NDCG ranking measures," in *Proc. Annu. Conf. Learn. Theory*, 2013, pp. 1–26.
[25] C. J. Burges, "From ranknet to lambdarank to lambdamart: An overview," Microsoft Research, Redmond, WA, Tech. Rep. MSR-TR-2010–82, 2010.
[26] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine based neural network for CTR prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 1725–1731.
[27] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 431–439.
[28] J. F. Jeff Heaton, S. McElwee, and J. Cannady, "Early stabilizing feature importance for tensorflow deep neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, 2017, pp. 4618–4624.
[29] S. Wang, W. Lin, Y. Yang, X. Xiao, and S. Zhou, "Efficient route planning on public transportation networks: A labelling approach," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 967–982.
[30] L. Fu, D. Sun, and L. R. Rilett, "Heuristic shortest path algorithms for transportation applications: state of the art," *Comput. Operations Res.*, vol. 33, no. 11, pp. 3324–3343, 2006.
[31] R. J. Szczerba, P. Galkowski, I. S. Glicktein, and N. Ternullo, "Robust algorithm for real-time route planning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 3, pp. 869–878, Jul. 2000.
[32] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, 2015, Art. no. 29.
[33] J. Yuan *et al.*, "T-drive: Driving directions based on taxi trajectories," in *Proc. 18th SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2010, pp. 99–108.
[34] J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized route recommendation using big trajectory data," in *Proc. IEEE 31st Int. Conf. Data Eng.*, 2015, pp. 543–554.
[35] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, and K. Xu, "A unified approach to route planning for shared mobility," *Proc. VLDB Endowment*, vol. 11, no. 11, pp. 1633–1646, 2018.
[36] Y. Tong, L. Wang, Z. Zhou, L. Chen, B. Du, and J. Ye, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 773–788.
[37] Z. Yu, F. Yi, Q. Lv, and B. Guo, "Identifying on-site users for social events: Mobility, content, and social relationship," *IEEE Trans. Mobile Comput.*, vol. 17, no. 9, pp. 2055–2068, Sep. 2018.
[38] Y. Tong *et al.*, "The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1653–1662.
[39] Y. Xia *et al.*, "Intent-aware audience targeting for ride-hailing service," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2018, pp. 136–151.
[40] Y. Sun, H. Zhu, F. Zhuang, J. Gu, and Q. He, "Exploring the urban region-of-interest through the analysis of online map search queries," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018, pp. 2269–2278.

Hao Liu (Member, IEEE) received the PhD degree from the Hong Kong University of Science and Technology (HKUST), in 2017, and the BSc degree from the South China University of Technology (SCUT), in 2012. He is currently working as a research scientist with the Business Intelligence Lab, Baidu Research. His general research interests include data mining, machine learning, and big data management, with a special focus on mobile analytics, and urban computing.

Yongxin Tong (Member, IEEE) received the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology, in 2014. He is currently a professor with the School of Computer Science and Engineering, Beihang University. His research interests include big spatio-temporal data processing, crowdsourcing, federated learning, and privacy-preserving data analytics.

Jindong Han is currently working toward the master's degree at the Beijing University of Posts and Telecommunications, majoring in information and communication engineering. His research interests include data mining and ubiquitous computing.

Panpan Zhang received the master's degree from Fuzhou University, in 2011. He is currently a staff software engineer at Business Intelligence Lab, Baidu Research. He focus on big data system and machine learning system building.

Xinjiang Lu (Member, IEEE) received the BE degree from Xinjiang University, in 2007, and the MS degree in software engineering from Northwestern Polytechnical University, in 2011, and the PhD degree in computer science from Northwestern Polytechnical University, in 2018. He is currently a researcher at Business Intelligent Lab, Baidu Research. His recent research interests include data mining and mobile intelligence.

Hui Xiong (Fellow, IEEE) received the PhD degree from the University of Minnesota (UMN), Minneapolis, MN. He is currently a full professor with the Rutgers, the State University of New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Deans Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He is a co-editor-in-chief of Encyclopedia of GIS, an associate editor of the *IEEE Transactions on Big Data* (TBD), the *ACM Transactions on Knowledge Discovery from Data* (TKDD), and the *ACM Transactions on Management Information Systems* (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program co-chair for the IEEE 2013 International Conference on Data Mining (ICDM), a general co-chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a program co-chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an ACM Distinguished Scientist.