Learning Personalized Utility Functions for Drivers in Ride-hailing Systems Using Ensemble Hypernetworks

Jie Gao[a], Weiming Mai[a], and Oded Cats[a]

[a] Department of Transportation and Planning, TU Delft, Delft, Netherlands
j.gao-1@tudelft.nl,
w.m.mai@tudelft.nl
o.cats@tudelft.nl

Extended abstract submitted for presentation at the Conference in Emerging Technologies in Transportation Systems (TRC-30)
September 02-03, 2024, Crete, Greece

April 30, 2024

Keywords: Personalized utility functions; driver behavior; ride-hailing systems; explainable machine learning

1 Introduction

In ride-hailing systems, drivers' decisions to accept or reject ride requests depend on various factors including order characteristics, traffic conditions, and personal preferences [1]. Traditional models such as the Random Utility Maximization (RUM) approach assume linear correlations among these attributes to predict drivers' decisions. However, these models often inaccurately represent driver behaviors as they do not account for the non-linear interactions and personalized preferences of individual drivers. For example, a driver with a higher expected income may prioritize the fare more significantly than a driver with a lower expected income. The linear nature of the utility function in traditional models fails to address such complex, non-linear relationships, which can lead to suboptimal prediction accuracy.

To address these challenges, we introduce a model leveraging hypernetwork architectures and ensemble learning to more effectively learn drivers' personalized utility functions. Hypernetworks dynamically generate weights for a linear utility function based on trip request data and driver profiles, capturing the non-linear relationships. Additionally, employing an ensemble of hypernetworks trained on diverse data segments enhances model adaptability and generalization, minimizing the risk of overfitting through controlled randomness. Our model, validated with real-world driver behavior data, not only predicts utility functions accurately but also offers significant insights into drivers' decision-making processes through improved explainability and uncertainty quantification.

2 Methodology

The concept of employing a deep neural network as a hypernetwork to generate hyperparameters for another predictive neural network is well-established [2, 4]. A hypernetwork is essentially a neural model designed to output the parameters for another network, typically enhancing interpretability and adaptability. In pioneering work by Kadra et al. [6], hypernetworks were first used to create interpretable linear models, proving highly effective for both tabular and image data. Given a dataset $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_n \in \mathbb{R}^F$ is an F-dimensional data point

and y_n is the corresponding label, the optimization objective function for the parameters of the hypernetwork can be described by the following equations:

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{n=1}^{N} \left[\mathcal{L}(y_n, w(\mathbf{x}_n; \theta)) + \lambda \left\| w(\mathbf{x}_n; \theta) \right\|_1 \right]. \tag{1}$$

Where θ represents the optimal parameters for the hypernetwork, and $w(\cdot) : \mathbb{R}^F \rightarrow \mathbb{R}^{F+1}$ represents the output weights of the features and bias term. The loss function $\mathcal{L}$ in our application domain is set as the Binary Cross Entropy loss (BCE), since the driver utility prediction can be treated as a binary classification problem of accept or reject. The second part of the objective function is an L1-regularized term for maintaining the sparsity of the output weights.

To obtain accurate uncertainty quantification, an ensemble learning technique [7] is used to train multiple hypernetworks to predict the weights in the utility function. The final weights are caculated by averaging the outputs of the hypernetworks:

$$\bar{w}(\mathbf{x}_n) = \frac{1}{M} \sum_{m}^{M} w_m(\mathbf{x}_n; \theta_m). \tag{2}$$

In the equation above, θ_m represents the parameters of the m-th hypernetwork in the ensemble. Previous studies [5, 7] have shown that employing the ensemble models can significantly enhance both the robustness and accuracy of the uncertainty quantification. Then, the predicted probability from one hypernetwork can be obtained using the sigmoid function: $f_m(y|\mathbf{x}; \theta_m) = \frac{1}{1+e^{-(w(\mathbf{x};\theta_m)^T \mathbf{x} + b(\mathbf{x};\theta_m))}}$. Here, $b(\mathbf{x}; \theta_m)$ represents the predicted bias for the utility function, which mimics the error term in the random utility function. The final predicted probability is the average of the predicted probability of each sub-hypernetwork: $f(y|\mathbf{x}) = M^{-1} \sum_m^M f_m(y|\mathbf{x}; \theta_m)$.

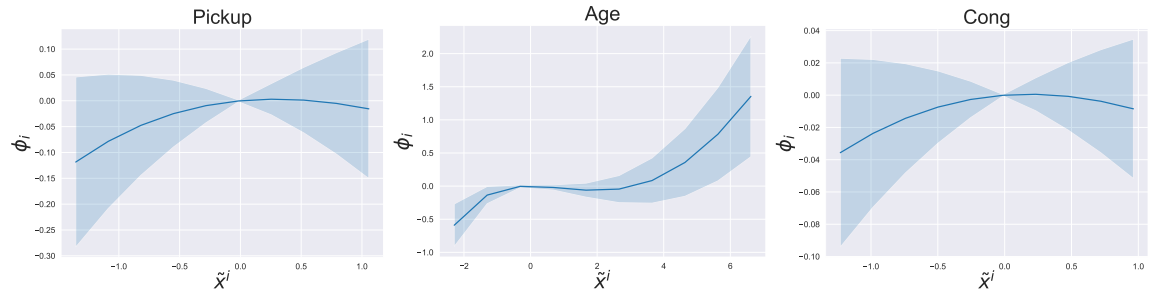
3 Results

3.1 Dataset and model performance

We used a dataset from a cross-sectional stated preference (SP) survey by Ashkrof et al. [1] collected in 2021. Initially containing 46 features, we applied SHAP (SHapley Additive exPlanations) to assess feature importance and retained 32 relevant features. All continuous features were standardized using Z-Score normalization. The dataset exhibited an imbalance between rejected and accepted data; therefore, in addition to considering the overall prediction accuracy (ACC), we also employed Area Under the Curve (AUC) and Area Under the Precision-Recall Curve (AUCPR) as the other two metrics. The Expected Calibrated Error (ECE), Brier Score (BS), and Negative Log Likelihood (NLL) [3] are used to measure the quality of predictive uncertainty. The dataset was divided into five train-test sets using different random seeds. Table 1 summarizes the average and standard deviation of model performance across these sets. We

Table 1 – *Models Performance on the driver acceptance behavior dataset. The dataset is split into 5 train-test sets with different random seeds. The results in the table are the average values tested on the train-test sets with the standard deviation.*

Models	ACC (↑)	AUC (↑)	AUCPR (↑)	ECE (↓)	BS (↓)	NLL (↓)
Logistic Regression	0.786 ± 0.002	0.633 ± 0.017	0.334 ± 0.016	0.566 ± 0.005	0.163 ± 0.002	1.447 ± 0.038
Decision Tree	0.764 ± 0.016	0.665 ± 0.025	0.360 ± 0.020	0.420 ± 0.029	0.188 ± 0.012	5.185 ± 1.086
Xgboost	0.794 ± 0.012	0.745 ± 0.022	0.492 ± 0.020	0.654 ± 0.007	0.157 ± 0.006	1.722 ± 0.152
TabResNet	0.779 ± 0.005	0.699 ± 0.024	0.434 ± 0.030	0.219 ± 0.003	0.158 ± 0.005	1.459 ± 0.082
HyperNetwork (M=1)	0.794 ± 0.004	0.704 ± 0.016	0.444 ± 0.030	0.565 ± 0.007	0.155 ± 0.003	1.316 ± 0.116
Ens-Hyper (M=5)	0.802 ± 0.006	0.720 ± 0.008	0.463 ± 0.012	0.571 ± 0.005	0.149 ± 0.001	1.269 ± 0.059



(a) (b) (c)

Figure 1 – *The feature contribution of Pickup (pick up time), Age (age of the driver) and Cong (delay time due to congestion). The shawdow area represents the standard deviation of* ϕ_i.

evaluate two interpretable machine learning models, logistic regression and decision tree, along with a commonly used ensemble method, Xgboost. For deep learning, we use Tabular ResNet as a baseline and the backbone for our ensemble hypernetworks. The results indicate that while ensemble hypernetworks not achieve the highest accuracy, they provided superior uncertainty quantification compared to the baseline models. Ensemble hypernetworks also demonstrated a balanced performance in terms of explainability, accuracy, and uncertainty estimation. In contrast, a single hypernetwork model (M=1) performed worse than the ensemble models (M=5) across most metrics, except for ECE.

3.2 Determinants for General Driver Preference

To understand the impact of attributes that influence rejections, we quantify the contribution ϕ_i of the i-th variable by multiplying the feature value and the expectation of the weights on the dataset:

$$\phi_i = \tilde{x}^i \mathbb{E}_{\tilde{\mathbf{x}}} \left[\bar{w}(\tilde{\mathbf{x}})_i \right], \tag{3}$$

where $\tilde{x}^i \in [\tilde{x}^i_{min}, \tilde{x}^i_{max}]$ is the observed value of the i-th attribute of a ride request data $\tilde{\mathbf{x}}$, bounded by $\tilde{x}^i_{min}$ and $\tilde{x}^i_{max}$. We then adjust the value of $\tilde{x}^i$ from $\tilde{x}^i_{min}$ to $\tilde{x}^i_{max}$ to observe how ϕ_i changes. Figure. 1 demonstrates that pickup time and congestion delay impact rejections non-linearity. Features exceeding mean values do not consistently lead to rejections, suggesting compensatory effects from monetary factors. Age impacts acceptance rates, with younger drivers generally accepting more rides. This analysis shows how attributes globally affect decisions, with significant variability influenced by specific driver profiles and request details.

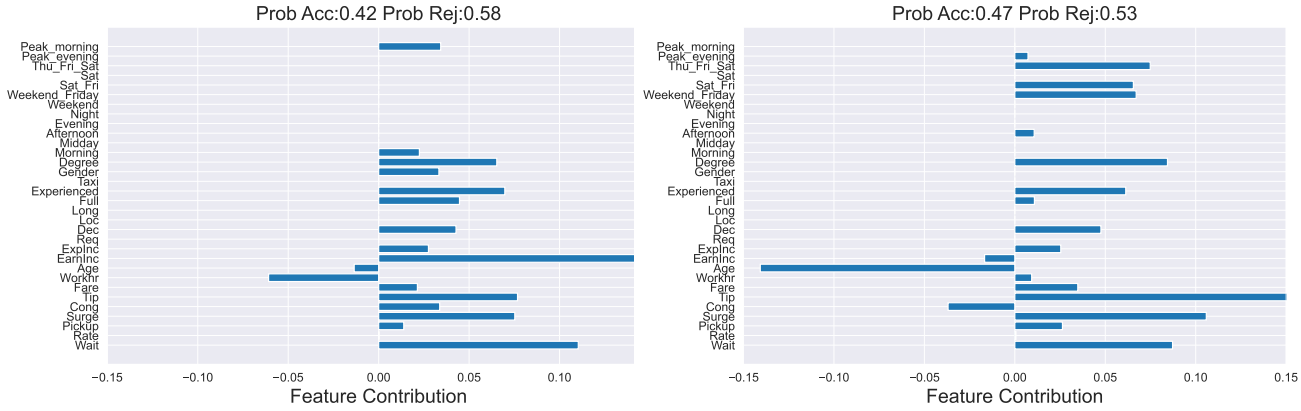
3.3 Personalized Driver Preference Analysis

One insightful application of the personalized utility function is that we can use it to analyze the preferences of an individual driver and also counterfactual analysis. We choose two true positive (the model correctly predicts the driver will decline the request) instances for illustrating the differences between individuals. Partial information of the drivers is presented in Table 2, and the predicted features contributions of these two drivers are shown in Figure. 2. As we can see in the table, the attribute *Age* plays a crucial role in the decision of the drivers. The model suggests that the older driver (ID=68), is more likely to accept the request compared to the younger driver (ID=133). Interestingly, although all the monetary variables are the same in these two request data, their contribution to driver 133's decision is higher than that of driver 68. This can imply that the younger driver is more sensitive to higher monetary rewards than the older one.

Table 2 – *Partial information of the two ride request data.*

ID	Rate	Pickup (mins)	Surge (price)	Cong (mins)	Tip (price)	Fare (price)	Workhr (hours)	Age	EarnInc (price)	ExpInc (price)
68	4	15	\$0	30	\$0	\$8	35	40	\$450	\$250
133	4	15	\$0	30	\$0	\$8	25	20	\$50	\$50

Remark: *Rate*: the rate of the rider; *Surge*: surge price for the request; *Tip*: guaranteed tip for the request; *Workhr*: working hours of the driver; *EarnInc*: earned income of the driver; *ExpInc*: driver's expected income.



(a) *Driver 68* (b) *Driver 133*

Figure 2 – *Personalized feature contribution of two drivers.*

4 Discussion

Our proposed model effectively learns personalized utility functions for ride-hailing drivers, achieving a balance between explainability, prediction accuracy, and uncertainty quantification. This approach allows practitioners to explore and adjust the utility function's weights, providing deep insights into individual drivers' decision-making processes. Additionally, the model supports the implementation of dynamic pricing strategies, such as adjustments in surge pricing, which can influence a driver's decision to accept or reject rides. This capability enables service operators to design personalized pricing strategies that improve the matching rate within ride-hailing systems.

References

[1] Ashkrof, Peyman, de Almeida Correia, Gonçalo Homem, Cats, Oded, & van Arem, Bart. 2022. Ride acceptance behaviour of ride-sourcing drivers. *Transportation Research Part C: Emerging Technologies*, **142**(9).

[2] Chauhan, Vinod Kumar, Zhou, Jiandong, Lu, Ping, Molaei, Soheila, & Clifton, David A. 2023. A brief review of hypernetworks in deep learning. *arXiv preprint arXiv:2306.06955*.

[3] Gneiting, Tilmann, & Raftery, Adrian E. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, **102**(477), 359–378.

[4] Ha, David, Dai, Andrew M., & Le, Quoc V. 2017. HyperNetworks. *In: ICLR (Poster)*. OpenReview.net.

[5] Havasi, Marton, Jenatton, Rodolphe, Fort, Stanislav, Liu, Jeremiah Zhe, Snoek, Jasper, Lakshminarayanan, Balaji, Dai, Andrew Mingbo, & Tran, Dustin. 2021. Training independent subnetworks for robust prediction. *In: ICLR*. OpenReview.net.

[6] Kadra, Arlind, Pineda-Arango, Sebastian, & Grabocka, Josif. 2023. Breaking the Paradox of Explainable Deep Learning. *CoRR*, **abs/2305.13072**.

[7] Lakshminarayanan, Balaji, Pritzel, Alexander, & Blundell, Charles. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, **30**.