Submitted to *Marketing Science*
manuscript MKSC-22-0238

Recommending for a Three-Sided Food Delivery Marketplace: A Multi-Objective Hierarchical Approach

(Authors' names blinded for peer review)

Online food delivery platforms are essentially a three-sided marketplace consisting of consumers, restaurant partners and delivery partners. A recommendation system for these platforms faces two main challenges. First, all sides of the marketplace have different and potentially conflicting utilities. Recommending in these contexts therefore entails jointly optimizing multiple conflicting objectives. Second, most food delivery platforms present their food recommendations hierarchically where a recommendation item can be either a single restaurant or a group of restaurants. Off-the-shelf recommendation algorithms are not applicable in these settings, as they focus on ranking items of the same type as a one-dimensional list. We propose MOHR, a multi-objective hierarchical recommender that tackles these challenges. MOHR combines machine learning with scalable multi-objective optimization for multi-sided recommendation, and incorporates a probabilistic model for hierarchical recommendation which accounts for consumers' browsing patterns. The hierarchical approach ensures consistent consumer experience across different levels of aggregations of the recommended items, and provides transparency to the restaurant partners. We further develop an efficient optimization solution for serving MOHR in large-scale online platforms in real time. We implement MOHR at one of the largest industrial food delivery platforms in the world serving millions of consumers, and experiment with objectives including consumer happiness, marketplace fairness and partner earnings. Online experiments show a significant increase in consumer conversion, retention and gross bookings, which combined translate to $1.5 million weekly gain in revenue. Our work has been deployed globally by the industrial food delivery platform as the recommendation algorithm for its homepage.

Key words: recommender systems; multi-sided marketplace; multi-objective optimization; heterogeneous and hierarchical contents; food delivery platforms

1. Introduction

Over the past years, recommender systems are becoming increasingly ubiquitous in retail (Xiao and Benbasat 2007, Pathak et al. 2010, Zhang et al. 2011, Smith and Linden 2017), media (Miller et al. 2003, Bennett et al. 2007, Covington et al. 2016), travel (Ghose et al. 2012, Ursu 2018, Chen and Yao 2017, Noulas et al. 2012), news (Prawesh and Padmanabhan 2014, Dhillon and Aral 2021) and social platforms (Backstrom and Leskovec 2011, Li et al. 2017, Xie 2010). On one hand, they help the consumers by facilitating information acquisition and decision-making; On the other hand, they help the content providers and e-commerce sellers to efficiently target prospective consumers. Recommender systems have become the key drivers for consumer growth in many personalization platforms today. Netflix reported that 80% of what people watch on its platform is from personalized recommendations (Gomez-Uribe and Hunt 2015). YouTube, the world's largest video sharing platform, reported that 70% of watch time is driven by recommendations (Solsman 2018).

While the research on recommender systems has largely focused on optimizing consumer-side utilities (Adomavicius and Tuzhilin 2005), many recommender systems today are operating in a multi-sided marketplace consisting of consumers, merchants or retailers, and potentially other partners as well. In this paper, we consider food delivery platforms such as Uber Eats, DoorDash and Grubhub, which offer personalized restaurant recommendations and deliveries to the consumers. They create a three-sided marketplace consisting of consumers, restaurant partners and delivery partners. Consumers discover and order food through the platform. Restaurant-partners use the platform as a sales channel to identify and target customers. And delivery-partners earn income by picking up food from restaurants and delivering it to the consumers. All sides of this marketplace are crucial to ensure a seamless consumer experience and the success of the business, which calls for a recommender system built for the marketplace that is optimized for all sides in it.

A challenge for recommending for a multi-sided food delivery marketplace is that each side has different and potentially conflicting utilities, and only optimizing for one may hurt other sides and eventually backfire. A recommender system for the food delivery platform which only optimizes consumers' conversion rate (probability of ordering on the platform) may overly recommend well-established and popular restaurants. Such recommendation strategies will cause several issues. First, new restaurants will not get enough exposure on the platform, which hurts marketplace fairness and discourages new restaurants from signing up. The lack of consumer interaction and feedback collected on these restaurants as a result of low exposure further impedes the recommender system's capability to make accurate predictions about consumer preferences on them, which eventually hurts the consumers' experience due to lack of selection and poor personalization. Second, orders concentrating on a few popular restaurants may hurt supply and demand efficiency and service reliability. The restaurant may not be able to fulfill a large quantity of orders in a short period of time, and there might not be enough delivery partners nearby to pick up all the food orders and deliver them in time. This can lead to order delays, cancellations and unfulfillment, in addition to unbalanced job assignments among delivery partners. Third, optimizing consumer's short-term conversion rate may lead to recommending restaurants and dishes that are visually appealing but not of good quality and taste, which hurts consumer happiness and retention. In addition, overly recommending restaurants that are known to be of interest to the consumers provides little extra value from the recommender itself and may cause user fatigue (Ma et al. 2016) and churn. All of these issues hurt one or more sides of the marketplace and eventually hurt the business. Therefore, a delicate balance of the multiple objectives from different sides is required in order to maintain a healthy ecosystem and ensure the success of the business in the long run. While the limits of user-centric recommendation and the need for doing multi-objective recommendation has been recognized (Abdollahpouri et al. 2020), a systematic and principled approach for building and efficiently serving a recommender system in large-scale three-sided food-delivery platforms is still lacking today. The first part of our work aims at filling this gap.

Another challenge for building a recommender system for a food delivery platform comes from the heterogeneity of the recommendation items. A recommendation item can be either a single restaurant or a collection of them. Most food delivery platforms today adopt this hierarchical presentation. For example, Figure 6 in Appendix A shows the homepage of Uber Eats, DoorDash and Grubhub respectively, which are three of the top food delivery platforms in the US. In addition to single restaurants, we see collections of restaurants such as "Fastest near you" or "National Brand" appearing as a recommendation item, which are presented as rows or *carousels* that can be scrolled through horizontally. The homepage is therefore a two-dimensional grid and the recommendation algorithm needs to decide the following: what and how many carousels to show to the consumers, and how to rank restaurants together with the carousels. Off-the-shelf recommendation algorithms are not directly applicable in these settings, as they focus on ranking items of the same type in a one-dimensional list. In addition, rule-based and curated ranking algorithms, such as pining a particular type of content at a particular position in the feed, are not favorable in these cases either, as they do not provide transparency to the restaurant partners as to why a certain item or certain groups of item is ranked on top of another. Moreover, consumers have limited patience in browsing the app and they may give up at any point if no relevant items are found. The second part of our work involves the design of a probabilistic consumer browsing model for hierarchical recommendation that addresses these challenges, with the output as a set of calibrated and interpretable ranking scores while explicitly accounting consumers' browsing patterns.

In our work, we propose a framework that simultaneously tackles these two challenges for a three-sided food delivery platform. We develop a three-step model-based framework, Multi-Objective Hierarchical Recommender (MOHR), that combines machine learning, hierarchical modeling and multi-objective optimization for ranking heterogeneous items for the multi-sided marketplace. In the first step (MO), we build machine learning models for predicting multiple objectives for the individual items given the hierarchy information. In the second step (H), we obtain predictions of the multiple objectives on aggregation level (carousel-level), by developing a probabilistic hierarchical model for consumers' browsing patterns. In the final step (R), we construct the whole homepage using the multi-objective ranking scores by solving a constrained optimization problem.

We implement MOHR at one of the largest food delivery platforms in the world, with multiple objectives for the three-sided marketplace including consumer conversion and retention, gross bookings and marketplace fairness. We conduct live experiments on global consumers and compare MOHR with the latest production system. Online A/B experiments show that MOHR pushes forward the Pareto frontier of the top line business metrics, leading to a significant improvement in consumer conversion, retention and gross bookings, which translate to $1.5 million weekly gain in gross bookings. We are also able to achieve a significant performance gain for the new restaurants on the platform, without significantly impacting consumer-side metrics. Because of the significant business impact, MOHR has been deployed globally as the recommendation algorithm for the company's homepage.

Our research contributes to the marketing community both methodologically and managerially. Methodology-wise, we propose a model-based recommendation framework combining machine learning, hierarchical modeling and multi-objective optimization to address the two prominent challenges in a three-sided food delivery marketplace, namely multi-sided trade-off and hierarchical recommendation. Managerially, our work provides insights on the trade-off among the utilities from different sides in a multi-sided marketplace. In particular, too much emphasis on a particular objective will hurt overall consumer experience and backfire. We show that it is beneficial and necessary for the business to explicitly model and optimize for the conflicting utilities from different sides in order to maintain a healthy ecosystem and be successful in the long term. Our proposed framework is general, flexible and can be readily applied to other recommendation applications within and outside the food delivery industry.

2. Related Work

2.1. Recommender Systems

There are three types of recommender systems, which are built on content-based filtering, collaborative filtering, and hybrid approaches respectively (Adomavicius and Tuzhilin 2005, Ricci et al. 2015, Dhillon and Aral 2021). Content-based recommender systems are based on a description of the item and a profile of the consumer's preferences and recommend items that are similar to items that the consumer has enjoyed in the past (Aggarwal et al. 2016, Brusilovsky 2007, Mooney and Roy 2000). Collaborative filtering approaches are based on the assumption that consumers who liked similar items in the past will like similar kinds of items in the future (Breese et al. 1998, Billsus et al. 1998), and are further classified as memory-based (Adomavicius and Tuzhilin 2005, Delgado and Ishii 1999) and model-based (Billsus et al. 1998, Breese et al. 1998). Most recommender systems today adopt a hybrid approach combining collaborative filtering and content-based filtering (Balabanović and Shoham 1997, Adomavicius and Tuzhilin 2005, Tso-Sutter et al. 2008, Sahoo et al. 2012), which empirically performs better than pure approaches (Adomavicius and Tuzhilin 2005). In this work, we leverage both the content features for the restaurants (e.g. cuisine type, location) and model-based collaborative filtering features based on the interaction history between the consumers and restaurants, contributing to the literature of hybrid recommender systems.

2.2. The Effects of Recommendations

A large number of studies in marketing, information systems and computer science have developed understanding on the effects of recommender systems on consumer decision making. Xiao and Benbasat (2007) provide theoretical perspectives of the effects of recommender systems on consumer decision making processes, outcomes, and consumers' evaluations of the recommender systems. Recommender systems affect consumers' consumption patterns from various aspects, including diversity (Fleder and Hosanagar 2009, Anderson et al. 2020), exploration (Datta et al. 2018, Chen et al. 2021), homogeneity

(Chaney et al. 2018) and fragmentation (Hosanagar et al. 2014). In e-commerce, it has been demonstrated that recommendation and ranking positions have significant impact on the consumers, including search (Narayanan and Kalyanam 2015, Ursu 2018), willingness to pay (Carare 2012, Adomavicius et al. 2018) and even consumption preferences (Adomavicius et al. 2013). It has also been shown that recommender systems affect other parties of the e-commerce marketplace, through impacting demand levels (Oestreicher-Singer and Sundararajan 2012), seller profits (Chen et al. 2008, Das et al. 2009, Azaria et al. 2013) and overall welfare (Zhang et al. 2021, Aridor and Gonçalves 2021). In our work, we show that recommender systems can have positive or negative impact for different sides in a three-sided food delivery marketplace, and propose a recommendation framework that addresses the trade-off between the utilities of different sides of the marketplace in a principled way.

2.3. Multi-Objective Recommendation for Multi-sided Marketplace

Past work on recommender systems focused on optimizing consumer-side objectives (Adomavicius and Tuzhilin 2005). Research evolved from optimizing a single aspect of consumer feedback such as ratings or click-through rates (Adomavicius and Tuzhilin 2005, Hu et al. 2008), to utility-based recommender systems that capture multidimensional preferences of consumer utilities (Ghose et al. 2012, Li et al. 2017, Carbonell and Goldstein 1998).

However, a lot of the personalization platforms today are a multi-sided marketplace (Rochet and Tirole 2003, Evans et al. 2011) consisting of multiple stakeholders including consumers, providers and the system itself. These multi-sided platforms create value by bringing buyers and sellers together, reducing search and transaction costs (Evans and Schmalensee 2016). A key to the success of a multi-sided platform is being able to attract and retain participants from all sides of the business. In recent years, with the increasing awareness of the limitations of user-centric recommendation systems, there has been an increasing number of literature on recommender systems for multi-sided marketplace (Abdollahpouri et al. 2020). As an example, researchers have considered seller earnings and platform profits by explicitly incorporating revenue or profit as objectives for the recommender system (Chen et al. 2008, Das et al. 2009, Hosanagar et al. 2008, Azaria et al. 2013) and further maximize the total welfare of the system (Aridor and Gonçalves 2021, Zhang et al. 2021). Multi-sided recommender systems have wide applications in e-commerce (Li et al. 2018), education (Zheng et al. 2019), loan (Lee et al. 2014), travel (Krasnodebski and Dines 2016), news (Tintarev et al. 2018), and content-sharing platforms (Zhao et al.

2019) etc. However, most of the extant applications for multi-sided recommenders are for two-sided marketplaces consisting of consumers and providers. To our knowledge, we are the first to consider a unique and challenging setting of a three-sided marketplace in the context of food delivery platforms.

Multi-sided recommendation usually entails optimizing multiple objectives that are potentially conflicting with each other. For example, Hosanagar et al. (2008) looked into the trade-offs between the relevance to consumers and the margin for the firm, and between short-term and long-term profits, and showed that a profit-sensitive strategy led to an increased revenue without a significant loss in consumer satisfaction. On the other hand, Zhang et al. (2021) showed that maximizing profit can actually hurt consumer surplus. On the fairness front, Wang et al. (2021) showed that optimizing the Pareto frontier of multiple tasks' accuracy may hurt the fairness of some tasks.

Building a recommender system for a multi-sided marketplace is essentially a multi-objective optimization problem (Sawaragi et al. 1985). Methods from the multi-objective optimization literature have been adapted for multi-sided recommender systems. Examples include constrained optimization (Rodriguez et al. 2012, Agarwal et al. 2015, 2011, 2012), learning-to-rerank (Nguyen et al. 2017) and multiple-gradient descent (Milojkovic et al. 2019). The multi-objective optimization component of our framework is related to the line of work on constrained optimization.

2.4. Consumer Behavior Modeling and Hierarchical Recommendation

Consumers' browsing behavior on recommender systems is related to consumer search for decision making. Weitzman (1979) was among the first to model sequential search behavior. Built on this, Ursu (2018) proposed a sequential search model for understanding the effect of rankings on consumer online choices in the hotel industry. Shi and Trusov (2021) developed an empirical model for consumers' scroll behavior in search engine marketing (SEM) based on laboratory eye-tracking data. Dhillon and Aral (2021) proposed a neural matrix factorization approach to model consumers' dynamic interest over time. These works focused on building structural or temporal models for understanding consumer behavior, but did not leverage the model output or the understanding to improve the ranking or build a new recommender system. Closer to our work is Liebman et al. (2019), which leveraged consumers' in-session sequential behavior for online adaptation to listeners' music preferences. However, the proposed ranking model does not apply to the heterogeneous and hierarchical recommendation items in our case.

Hierarchical recommendation is proposed to recommend items of different levels of aggregations. Methodologies based on hierarchical clustering (Zheng et al. 2013, King and Imbrasaitė 2015) and hierarchical reinforcement learning (Xie et al. 2021) are used to recommend aggregations of items. Oestreicher-Singer and Sundararajan (2012) studied the performance of a single-item recommendation in the context of a group of recommended items. Song et al. (2019) proposed a cascade model for consumers' sequential scrolling and decision process, and a multicategory utility model for recommending items on category levels. To the best of our knowledge, existing works in this area either did not explicitly model and account for consumers' browsing behavior (Zheng et al. 2013, Xie et al. 2021), or the recommendation output is a homogeneous one-dimensional list although the consumer decision process is modeled in a hierarchical way (Song et al. 2019, Oestreicher-Singer and Sundararajan 2012, Agrawal et al. 2009). Our work bridges this gap by developing a probabilistic hierarchical model as a component of the proposed MOHR framework, which ranks items with different hierarchical aggregations. In addition, we point out that another critical consideration for hierarchical recommendation is how consumers navigate the page and interact with items at different horizontal and vertical positions (Alvino and Basilico 2015). In the SEM context, it has been shown that consumers rarely looked at lower ranking results (Guan and Cutrell 2007) and their dominant browsing pattern looks like the letter F or a "golden triangle" (Nielsen 2006, Sherman 2005). Ursu (2018) shows that the click-through rate decreases with lower ranking positions on a hotel recommendation website. The consumer browsing model in our framework explicitly models consumers' browsing patterns with the hierarchical recommendation items.

In sum, our work draws on the two strands of literature on multi-objective recommendation and sequential consumer behavior modeling on personalization platforms. It adds to the literature by proposing a model-based framework combining machine learning, hierarchical modeling and multi-objective optimization. The proposed framework ranks heterogeneous and hierarchical contents in a principled and calibrated way while optimizing for the multiple sides with conflicting utilities, which addresses two of the most prominent challenges for a three-sided food delivery marketplace.

3. Data and Institutional Background

3.1. Three-Sided Food Delivery Marketplace

There has been an emerging wave of food delivery platforms in the past decade. During the coronavirus (COVID-19) pandemic, the use of online food delivery services increased 67% globally between 2019 and 2020 (Muangmee et al. 2021). These services create a three-sided marketplace consisting of consumers, restaurant partners and delivery partners. Consumers place orders on food from the restaurants on the platform. Delivery partners pick up the food from the restaurants and deliver it to the consumers, when consumers have the option to add a tip. The platform charges a fixed portion of the consumers' payment as the commission fee and pays the rest to the restaurant partners. The delivery partners earn income from consumers' tips and the platform's payment[1].

Each side in the food delivery marketplace has different and potentially conflicting utilities, and only optimizing for one may hurt others. As an example, optimizing consumers' conversion can lead to overly recommending popular restaurants. This causes several issues. First, new restaurants will not get enough exposure on the platform. The lack of consumer feedback collected on these restaurants further impedes the recommender system's capability to make accurate predictions about consumer preferences. Second, orders concentrating on a few popular restaurants hurts supply efficiency and service reliability. Restaurants may not be able to fulfill a large quantity of orders in a short time, and there might not be enough delivery partners nearby. Lastly, overly recommending restaurants that are known to be of interest to the consumers provides little extra value and may cause consumer fatigue (Ma et al. 2016) and churn. All of these issues hurt one or more sides of the marketplace and eventually the business. Therefore, a delicate balance of the multiple objectives from different sides is needed to maintain a healthy ecosystem and ensure the success of the business in the long term.

3.2. Hierarchical and Heterogeneous Recommendation Items

As shown in Figure 6 in Appendix A, recommendations on food delivery platforms usually appear as heterogeneous and hierarchical. The advantages for recommending a collection of restaurants as *carousels* in the homepage are threefold: First, carousels can be viewed as nudges tailored to the different modes of the consumers (e.g. in a hurry, looking for something healthy) and help them efficiently navigate through the contents. Second, the title of the carousels provide extra information about the restaurants (e.g. cuisine type) that may be critical to consumers' decision making, but are not clear to the consumers otherwise. Finally, they alleviate cold-start challenges for new restaurants and new consumers. For example, "New Restaurants" and "Popular Near You" are non-personalized carousels based on only content information, which is a popular approach for solving cold-start problems in recommender systems (Schein et al. 2002).

While carousels are appealing in some contexts, at other times a single restaurant is preferred as recommendation. For example, an consumer may order from the same restaurant repeatedly. An ideal recommendation setup is therefore a combination of carousels and single restaurants. This is the strategy adopted by major food delivery platforms today (Zhu 2021). It is worth mentioning that hierarchical recommendation is also common outside the food delivery industry. The homepage of Netflix is a series of rows where each row is a coherent group of videos. This setting is found to be favorable to a large, unorganized collection of relevant videos (Gomez-Uribe and Hunt 2015).

3.3. Data

The company we work with is one of the largest food delivery platforms in the world serving millions of consumers every day. Consumers' interactions in the app are logged and processed through Apache Hive (Thusoo et al. 2009) for data extraction, transformation and loading (ETL). Specifically, an impression event is logged when the consumer scrolls through an item. An order event is logged when the consumer places an order. Contextual information is logged together with the event, including time of day, day of week and geolocation etc. The data we obtain from the company are randomly sampled from the global user logs, consisting of about 600 million impressions and 11 million orders between May 15, 2019 and May 28, 2019.

4. Model

In this section, we describe the three interconnected components of the MOHR framework. In Section 4.1, we build machine learning models for restaurant-level objectives (MO-step). In Section 4.2, we develop a probabilistic hierarchical model for aggregating restaurant-level objectives to carousel-level objectives (H-step). In Section 4.3, we solve a constrained optimization problem and obtain a multi-objective ranking score (R-step).

4.1. MO-Step: Machine Learning Models for Restaurant-Level Objectives

Objective values might vary given different sources of the restaurant. A restaurant appearing in "Fast near you" carousel is more appealing to a consumer who is in a hurry than the same restaurant appearing in "Popular near you" or as a single recommendation. Therefore, we model the objectives on the $(consumer, restaurant, source)$ level, where $source$ is the hierarchy information of the restaurant (e.g. belongs to the "Italian food" carousel[2]). Table 1 summarizes the notations used for our MOHR framework. Table 2 summarizes the objectives in our experiments. It's worth pointing out that although we only discussed the estimation of four objectives in this section, the MO-step is general and can incorporate any number of objectives that are of interest to the platform.

In particular, we model the following four objectives: (1) **Consumer conversion**: whether the consumer places an order; (2) **Consumer retention**: whether the consumer returns to the platform and orders again within the next 14 days[3], if the consumer orders in the current session[4]; (3) **Basket value**: dollar amount of the order, if the consumer orders in the current session; (4) **Marketplace fairness**: the exposure opportunities that new restaurants receive on the platform. Consumer conversion captures consumers' immediate responses and short-term engagement, which is a common objective used by consumer-centric recommender systems (Zhang et al. 2019, Covington et al. 2016). Consumer retention, on the other hand, captures consumers' long-term engagement, which has attracted increasing attention for practitioners in recent years (Wu et al. 2017, Zou et al. 2019). Basket value directly contributes to the revenue of the business. Marketplace fairness ensures that every restaurant on the platform have fair opportunity of being shown. Details for each objective are described below.

Consumer Conversion, Consumer Retention and Basket Value Objectives. Using the notations in Table 1, we build machine learning models for consumer conversion, consumer retention and basket value as:

$$\begin{aligned}
c(i,j,k) &= \mathbf{E}[O(i,j,k,z)=1] = f_c(x_i, x_j, x_k, x_{ij}, x_{ik}, x_{jk}, x_{ijk}, z),\\
r(i,j,k) &= \mathbf{E}[R(i,j,k,z)|O(i,j,k,z)=1] = f_r(x_i, x_j, x_k, x_{ij}, x_{ik}, x_{jk}, x_{ijk}, z),\\
b(i,j,k) &= \mathbf{E}[B(i,j,k,z)|O(i,j,k,z)=1] = f_b(x_i, x_j, x_k, x_{ij}, x_{ik}, x_{jk}, x_{ijk}, z),
\end{aligned} \tag{1}$$

where we drop the dependency on context z for ease of notation. Here x_i represents the set of consumer-level features, x_{ij} represents the interaction history between consumer i

Notation	Definition and comments
i	Index for consumers
j	Index for restaurants
k	Source of the restaurant, e.g. "Popular near you", or "Single" if appears as a single restaurant
q	Index for a recommendation item, which can be either a single restaurant within the carousel or a whole carousel
z	Context features such as time of day, day of week, meal period, country, geolocation
$O(i,j,k,z)$	(Restaurant-level) Binary random variable taking value 1 if consumer i orders from restaurant j from source k under context z, 0 otherwise
$R(i,j,k,z)$	(Restaurant-level) Binary random variable taking value 1 if consumer i returns to the platform and orders within 28 days of ordering from restaurant j from source k under context z, 0 otherwise
$B(i,j,k,z)$	(Restaurant-level) Continuous random variable taking value as the dollar amount of the basket value if consumer i orders from restaurant j from source k under context z, 0 otherwise
$O(i,k,z)$	(Carousel-level) Binary random variable taking value 1 if consumer i orders from source k under context z, 0 otherwise
$R(i,k,z)$	(Carousel-level) Binary random variable taking value 1 if consumer i returns to the platform and orders within 28 days of ordering from source k under context z, 0 otherwise
$B(i,k,z)$	(Carousel-level) Continuous random variable taking value as the dollar amount of the basket value if consumer i orders from any restaurant from source k under context z, 0 otherwise
N_j	Number of impressions from restaurant j
N_j^1	Number of orders from restaurant j
I	Number of consumers
Q	Number of recommendation items (restaurants or carousels)
$\boldsymbol{x} = \{x_{iq}\}$	Ranking plan, where x_{iq} is the probability of serving item q to consumer i
$\boldsymbol{u} = \{u_{iq}\}$	Uniform ranking plan, where $u_{iq} \equiv \frac{1}{Q}$
$c_{iq}, r_{iq}, b_{iq}, f_{iq}$	Compact forms for the consumer conversion, consumer retention, basket value and fairness objectives for consumer i and item q

Table 1 Summary of notations.

Objective	Relevant sides	Level	Notation	Definition
Consumer conversion	Consumers, restaurant partners	Restaurant	$c(i,j,k)$	$\mathbf{P}[O(i,j,k,z)=1]$
		Carousel	$c(i,k)$	$\mathbf{P}[O(i,k,z)=1]$
Consumer retention	Consumers, restaurant partners	Restaurant	$r(i,j,k)$	$\mathbf{E}[R(i,j,k,z)\|O(i,j,k,z)=1]$
		Carousel	$r(i,k)$	$\mathbf{E}[R(i,k,z)\|O(i,k,z)=1]$
Basket value	Restaurant partners, delivery partners	Restaurant	$b(i,j,k)$	$\mathbf{E}[B(i,j,k,z)\|O(i,j,k,z)=1]$
		Carousel	$b(i,k)$	$\mathbf{E}[B(i,k,z)\|O(i,k,z)=1]$
Marketplace fairness	Restaurant partners	Restaurant	$f_r(j)$	$\sqrt{\mathrm{Var}(c_j\|\{O_{jm}\}_{m=1}^{N_j})}$
		Carousel	$f_k(k)$	$\sum_{l=1}^{n} f_r(j_l) \prod_{l'=1}^{l} (1-c(i,j_{l'-1},k)) \cdot p_{l'-1,l'}$

Table 2 Summary of objectives.

and restaurant j, while x_{ijk} represents the interaction history between consumer i and restaurant j conditional on j appears in source k, etc. We also include collaborative filtering

(Breese et al. 1998) features using matrix factorization to capture the similarities among consumers and items. Details are provided in Appendix B.1.1.

For the consumer conversion and retention objective which are binary classification problems, we adopt gradient boosting decision trees (Friedman 2001, 2002) as the nonlinear prediction function f_c and f_r, which are a family of ensemble methods that combines individual weak CART (Breiman et al. 2017) classifiers. For the basket value objective, we use gradient boosting regression tree as f_b with squared loss as the loss function (Hastie et al. 2009), and truncate the predictions to be non-negative[5]. They achieve a nice balance between predictive power and interpretability. Model training is done using the gradient boosting machine (GBM) on H2O (Click et al. 2017, Hastie et al. 2009), which is a popular distributed in-memory machine learning platform. A list of the features and the parameters of the machine learning models can be found in Appendix B.1.2.

Marketplace Fairness Objective. Restaurant recommendation can be viewed as a contextual multi-armed bandit problem (Langford and Zhang 2007, Auer et al. 2002, Katehakis and Veinott Jr 1987) where every arm is a restaurant. The agent for a multi-armed bandit problem balances *exploration*, which acquires new knowledge, and *exploitation*, which optimizes current decisions based on existing knowledge. For food delivery platforms, the recommender system needs to balance between exploiting well-established restaurants and exploring new restaurants to maintain a healthy ecosystem and ensure marketplace fairness. We adopt the multi-armed bandit framework with Bayesian modeling for the marketplace fairness objective as it fits nicely with our use case. A well-adopted approach for the contextual bandit problem is the upper-confidence bound (UCB) algorithm, where the optimal action chosen at each step is given by

$$j^* = \arg\max_j [Q(j) + \kappa\sigma(j)]. \tag{2}$$

Here $Q(j)$ is the estimated value of action j, $\sigma(j)$ is the estimated standard deviation of the value of j, and $\kappa > 0$ controls the level of exploration. In our case, $Q(j)$ can be the estimated conversion rate for restaurant j, i.e. $c(i,j,k)$ from the previous section. $\sigma(j)$ measures the uncertainty of $c(i,j,k)$, which intuitively is high for new restaurants. Therefore, we define $\sigma(j)$ to be the fairness objective, and we would like to maximize the exposure for the restaurants with high $\sigma(j)$.

To estimate the fairness objective $\sigma(j)$, we leverage Bayesian modeling to estimate $\sigma(j)$ as the posterior standard deviation for $c(i,j,k)$. Appendix B.1.3 discusses the prior distribution and the derivation for the posterior distribution for $c(i,j,k)$, which yields the restaurant-level *marketplace fairness objective* $f_r(j)$ for restaurant j as

$$f_r(j) = \sigma(j) = \sqrt{\frac{(\alpha_j + N_j^1)(\beta_j + N_j - N_j^1)}{(\alpha_j + \beta_j + N_j)^2(\alpha_j + \beta_j + N_j + 1)}}, \tag{3}$$

where α_j and β_j are the parameters for the prior Beta distribution $\mathcal{B}(\alpha_j, \beta_j)$ for $c(i,j,k)$, and there are N_j impressions on restaurant j, out of which N_j^1 lead to orders. As a sanity check, given α_j and β_j, lower values of N_j lead to *higher* posterior variance of c_j. In other words, the fewer impressions a restaurant receives, the more uncertain the system is about the estimation of the restaurant's conversion rate, hence the higher value for the fairness objective. We discuss the choice for the prior parameters α_j and β_j in Appendix B.1.4. A trailing window of 120 days[6] is chosen for the impression counts N_j and order counts N_j^1.

The benefits of the marketplace fairness objective $f_r(j)$ are threefold. First, it offers new restaurants more exposure on the platform. Second, it helps the learning of other objectives by introducing more training data on the new and low-volume restaurants. Third, by restricting to a trailing window for counting N_j, it provides a mechanism for adaptively boost new and low-volume restaurants over time: A restaurant will receive a high boost when first entering the platform, with Eq.(2) dominated by the second term ("**exploration**"); As it accumulates enough exposure, the boosting effects dies down and Eq.(2) is dominated by the point estimate $Q(j)$ ("**exploitation**"); Later on, when the restaurant is not performing well in a certain time period by having a low $Q(j)$, it will lose exposure (i.e. having low N_j). As a result $f_r(j)$ will go back up due to increasing values of $\sigma(j)$, therefore offering a "second chance" to the restaurant ("**resurrection**").

4.2. H-Step: Probabilistic Hierarchical Model for Carousel-Level Objectives

In the H-step, we aggregate restaurant-level objectives from the MO-step into carousel-level objectives, using a probabilistic hierarchical model. An important component in this step is the consumer browsing model, which is described below.

Consumers have limited patience when scrolling through the recommendations. Following the sequential search framework in Weitzman (1979), we assume that the consumer examines the restaurants in the carousel one-by-one. At each position, she performs one of the following three options: order from the current restaurant, continue browsing the next restaurant, or abandon the whole carousel[7]. Specifically, we propose a *consumer browsing model*, which outputs a set of scrolling probabilities that a consumer will scroll to the next position in a carousel:

$$p_{l,l+1} = \mathbf{P}(\text{the consumer scrolls to position } l+1 \mid \text{currently at position } l). \tag{4}$$

Figure 1 illustrates the consumer browsing model, where the position index starts at 1 and $p_{0,1} = 1$, meaning that consumers always browse the first restaurant in each carousel[8].

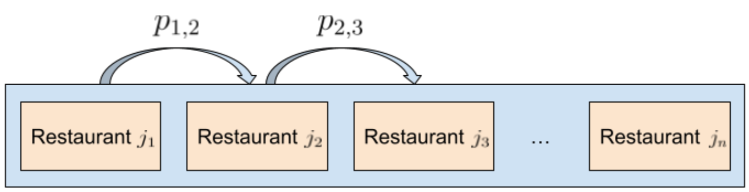


Figure 1 An illustration of the consumer browsing model and the scrolling factors for a carousel.

Consumer Conversion Objective for a Carousel. With the consumer browsing model, the carousel-level objectives can be derived as follows. We first compute carousel-level conversion objective $c(i,k)$[9], the probability that the consumer orders from any restaurant[10] in the carousel k under context z. Assuming the restaurant at position l inside the carousel is indexed by j_l, we have

$$\begin{aligned} c(i,k) &= \sum_{l=1}^{n} \Big[\mathbf{P}(\text{consumer } i \text{ orders from restaurant } j_l \text{ at position } l \mid \text{scrolls to position } l) \\ &\qquad \times \mathbf{P}(\text{scrolls to position } l) \Big] \\ &= \sum_{l=1}^{n} \Big[c(i,j_l,k) \prod_{l'=1}^{l} \mathbf{P}(\text{consumer } i \text{ didn't order at position } l'-1, \text{ and scrolls to position } l') \Big] \\ &= \sum_{l=1}^{n} \Big[c(i,j_l,k) \prod_{l'=1}^{l} (1 - c(i, j_{l'-1}, k)) \cdot p_{l'-1,l'} \Big], \end{aligned} \tag{5}$$

where we define $c(i,j_0,k) = 0$. Equation (5) is intuitive when viewing each term in the summation one by one: The first term, $c(i,j_1,k)$, is the probability that the consumer orders from the first restaurant in the carousel; The second term, $(1 - c(i,j_1,k)) \cdot p_{1,2} \cdot c(i,j_2,k)$, is the probability that the consumer abandons the first restaurant but scrolls to the second position and orders, etc[11].

Basket Value Objective for a Carousel. By law of total expectation, the expected basket value of a carousel can be decomposed as the sum of the basket value at each position:

$$\begin{aligned}\mathbf{E}[B(i,k,z)] &= \sum_{l=1}^{n} \mathbf{E}[B(i,j_l,k,z)|O(i,j_l,k,z)=1]P[O(i,j_l,k,z)=1] \\ &= \sum_{l=1}^{n} b(i,j_l,k)P[O(i,j_l,k,z)=1],\end{aligned} \tag{6}$$

which is a weighted combination of the basket value objective of each individual restaurants $b(i,j_l,k)$ inside the carousel, with the weights being the conversion probability at that position.

The basket value objective of the carousel is therefore

$$\begin{aligned}b(i,k) &= \mathbf{E}[B(i,k,z)|O(i,k,z)=1] = \mathbf{E}[B(i,k,z)]/\mathbf{P}[O(i,k,z)=1] \\ &= \sum_{l=1}^{n} \frac{P[O(i,j_l,k,z)=1]}{\sum_{l=1}^{n} P[O(i,j_l,k,z)=1]} b(i,j_l,k),\end{aligned} \tag{7}$$

where $P[O(i,j_l,k,z)=1] = c(i,j_l,k)\prod_{l'=1}^{l}(1-c(i,j_{l'-1},k)) \cdot p_{l'-1,l'}$ is the probability that the consumer scrolls to position l inside the carousel and orders from the restaurant j_l, as computed in Eq.(5). Therefore, the carousel-level basket value is effectively a *weighted average* of the expected basket values of individual restaurants inside the carousel, with the weights proportional to their predicted conversion at each position while accounting for the consumer's scrolling behavior.

Consumer Retention Objective for a Carousel. Following the same derivation above, the carousel-level consumer retention objective can be computed as

$$r(i,k) = \sum_{l=1}^{n} \frac{P[O_l(i,k,z)=1]}{\sum_{l=1}^{n} P[O_l(i,k,z)=1]} r(i,j_l,k). \tag{8}$$

Marketplace Fairness Objective for a Carousel. The carousel-level marketplace fairness objective $f_c(k)$ is slightly different as it is not conditioned on the consumer placing an order. By the same law of total expectation as in Eq.(5), we have

$$f_c(k) = \sum_{l=1}^{n} f_r(j_l)\mathbf{P}(\text{consumer } i \text{ scrolls to position } l) = \sum_{l=1}^{n} f_r(j_l) \prod_{l'=1}^{l}(1-c(i,j_{l'-1},k)) \cdot p_{l'-1,l'}, \tag{9}$$

which is a weighted sum of restaurant-level fairness objectives $f_r(j_l)$ at each position inside the carousel, with the weights being the probability that the consumer scrolls to that position.

To sum up, the H-step provides an hierarchical modeling approach for estimating each objective on the carousel level, as an aggregation of restaurant-level objectives inside the carousel. The predictions generated by the H-step are interpretable and provides levels of transparency to the consumers and restaurant partners.

4.3. R-Step: Constrained Optimization for Multi-Objective Ranking

4.3.1. Formulation. The final step is to construct the homepage using the output from the MO-step and H-step while accounting for the multi-sided trade-off. We formulate the trade-offs as a constrained optimization problem, with the constraints being the amount of sacrifice that the business is willing to make in some objectives while optimizing for others.

To describe the holistic ranking framework for both within-carousel ranking and across-carousel ranking, we introduce q as the index of a recommendation item, which can be either a restaurant within a carousel, or a whole carousel. We also introduce subscripts for more compact notations. For example, c_{iq} denotes the conversion rate for consumer i on item q, and b_{iq}, r_{iq}, f_{iq} are defined similarly. A ranking algorithm can be expressed as a set of personalized scores, with x_{iq} being the probability of recommending item q to consumer i. Let $\boldsymbol{x} = \{x_{iq} : i = 1, ..., I, q = 1, ..., Q\}$ denote the ranking scores for all consumers and all items, which is the optimization variable in the R-step.

For any ranking plan $\boldsymbol{x}$, its expected total numbers of orders, total gross bookings, total consumer retention and total fairness can be computed as

$$\begin{aligned} C(\boldsymbol{x}) &= \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}c_{iq}, \qquad B(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}c_{iq}b_{iq}, \\ R(\boldsymbol{x}) &= \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}c_{iq}r_{iq}, \qquad F(\boldsymbol{x}) = \sum_{i=1}^{I}\sum_{q=1}^{Q} x_{iq}f_{iq}. \end{aligned} \tag{10}$$

Let

$$C^* = \max_{x\in\mathcal{E}} C(\boldsymbol{x}), \quad B^* = \max_{x\in\mathcal{E}} B(\boldsymbol{x}), R^* = \max_{x\in\mathcal{E}} R(\boldsymbol{x}), \quad F^* = \max_{x\in\mathcal{E}} F(\boldsymbol{x}), \tag{11}$$

be the optimal values for the objectives, where $\mathcal{E} = \{\boldsymbol{x} : x_{iq} \geq 0, \sum_q x_{iq} = 1, \forall i\}$ is the feasible region for $\boldsymbol{x}$. We formulate the multi-objective ranking problem as a *constrained optimization problem*:

$$\max_{\boldsymbol{x}\in\mathcal{E}} C(\boldsymbol{x}) \quad \text{s.t.} \quad B(\boldsymbol{x}) \geq \alpha_b B^*,\ R(\boldsymbol{x}) \geq \alpha_r R^*,\ F(\boldsymbol{x}) \geq \alpha_f F^*, \tag{12}$$

where $0 < \alpha_b, \alpha_r, \alpha_f < 1$ specifies the amount of *tolerable trade-off* for $B(\boldsymbol{x})$, $R(\boldsymbol{x})$ and $F(\boldsymbol{x})$ when optimizing for $C(\boldsymbol{x})$[12]. The linear programming problem in Eq.(12) can be viewed as a multi-objective optimization problem (Sawaragi et al. 1985). In Appendix B.2.1, we prove that the Pareto frontier[13] between any two objectives is concave, so that a small sacrifice in one objective can potentially lead to big improvement in the other.

4.3.2. Solution. Eq.(12) has $I * Q$ number of variables, which can be huge given millions of consumers (I) and thousands of items (Q). This causes scalability issues for solving and serving the solutions online for large-scale food delivery platforms. To tackle this challenge, we adopt the trick in Agarwal et al. (2012) and add a quadratic penalty term to the objective function which leads to analytical solutions[14] for $\boldsymbol{x}$. By KKT conditions, The final ranking function is reduced to

$$\tilde{x}_{iq} = c_{iq} + \lambda_b c_{iq} b_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_f f_{iq}. \tag{13}$$

Here $\lambda_b, \lambda_r, \lambda_f > 0$ are the slack variables for the constraints on $B(\boldsymbol{x}), R(\boldsymbol{x})$ and $F(\boldsymbol{x})$ respectively, and are functions of α_b, α_r and α_f. The detailed optimization procedure and solution are provided in Appendix B.2.2.

Taking a closer look at Eq.(13), the ranking function is essentially a *weighted linear combination* of the multiple objectives. The basket value objective b_{iq} and retention objective r_{iq} is multiplied by the conversion objective c_{iq} while the marketplace fairness objective f_{iq} is not. This is again because the basket value and retention objective is a counterfactual estimation conditioning on the consumer placing an order in the current session, while the marketplace fairness objective is not (Table 2). While we can solve λ_b, λ_r, and λ_f as functions of α_b, α_r and α_f by solving a linear system as shown in Appendix B.2.2, it can be expensive due the large scale. In practice, we treat λ_b, λ_r and λ_f as tuning parameters directly to reduce computation. In addition, λ_b, λ_r and λ_f can also be viewed as the weights controlling the relative importance of the different objectives.

4.3.3. Usage. Figure 2 shows a diagram for the full MOHR framework. The ranking function in Eq.(13) is first used for (horizontal) within-carousel ranking, where c_{iq}, b_{iq}, r_{iq} and f_{iq} are plugged in as the restaurant-level objectives from the MO-step. It is then used for (vertical) across-carousel ranking, where c_{iq}, b_{iq}, r_{iq} and c_{iq} are plugged in as the carousel-level objectives from the H-step.

One of the biggest advantages for MOHR is the ability to dynamically rank heterogeneous and hierarchical contents in a holistic way, as the ranking scores are calibrated and comparable across different levels of aggregation. It is also worth noting that the MOHR framework is readily generalizable to higher orders of aggregated contents as well. For example, a recommendation item can be a *meta-carousel* which is an aggregation of carousels (that is, a carousel of carousels). The objective values for a meta-carousel can be obtained in the same way as described in the H-step, with a set of *inter-carousel* scrolling factors estimated from the consumer browsing model as input.

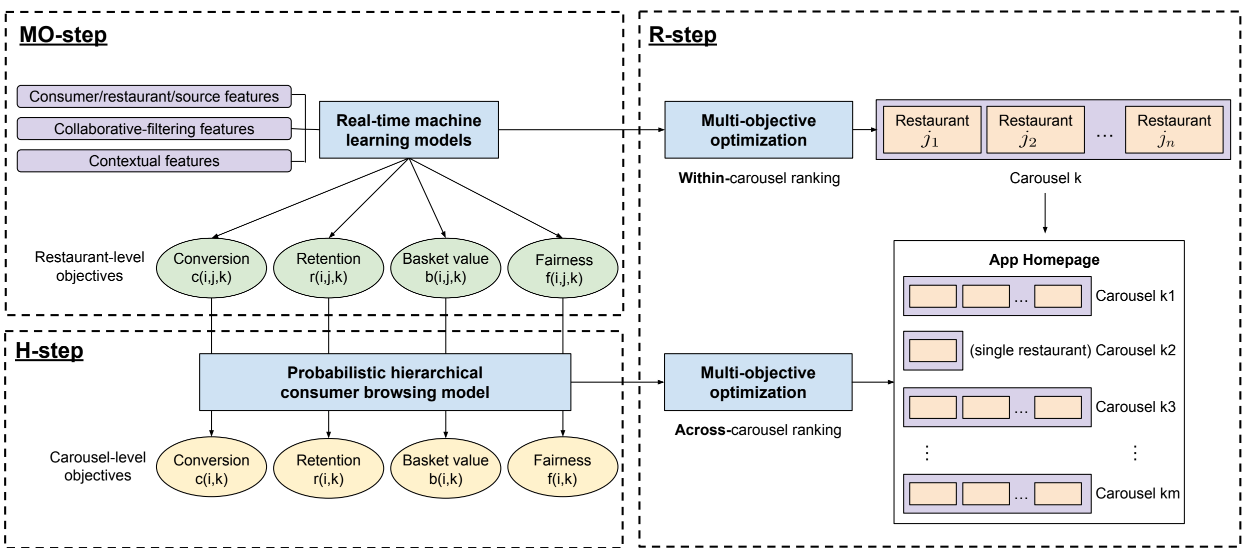


Figure 2 **(Color online) An overview of MOHR.**

5. Results

5.1. Experiment Setup and Performance Measures

5.1.1. Experiment Setup. To the best of our knowledge, we are the first to propose a hierarchical recommender that ranks contents of various levels of aggregation using a single holistic framework. The closest baseline is the latest production recommender system at the company. It decomposes the hierarchical recommendation problem into two parts, where carousels and individual restaurants are ranked separately using disjoint state-of-art hybrid machine learning recommendation algorithms, with consumer conversion as the single objective. All carousels are ranked above all single restaurants with another machine learning model determining how many carousels to display. See Appendix C.1 for a detailed description.

Our experiments were conducted over 28 days[15] in June 2019 on 2% of the company's global consumers[16]. Every consumer in the experiment traffic is assigned with a unique consumer identifier[17] which is randomly hashed into the treatment group or control group. The treatment group information is logged together with consumers' activities on the platform during the experiment period.

5.1.2. Performance Measures and Statistical Hypothesis Testing. Table 3 summarize a list of metrics for the online experiments, which correspond to the multiple objectives for the multiple sides in the three-sided marketplace that are critical to the business.

A consumer may visit the app multiple times during the experiment period. Let S_i be the number of sessions that consumer i generates during the experiment period, and O_{is}, B_{is} be the binary indicator for whether consumer i orders from session s, and the basket value for the session (0 if there is no order) respectively. To measure consumer retention, we let R_{is} be the binary indicator of whether consumer i returns to the app and places another order in the next 14 days following the current session s. Note that basket value and retention are measured only on ordered sessions. For marketplace fairness, we measure the performance of the new restaurants on the platform, which are those joining the platform within 21 days of the experiment start date. Specifically, we measure the percentage of the overall impressions and orders from the platform that are on those newly onboarded restaurants.

Note that the sessions generated by the same consumer are correlated with each other as they reflect the behavior of the same consumer. Therefore, ratio metrics such as conversion rate, basket value per order and retention in Table 3 are *not* from i.i.d. samples. We explicitly account for this intra-consumer correlation when computing the variance for those test statistics in hypothesis testing for the online experiments. The resulting p-values are larger than when the examples are treated as i.i.d., therefore our tests are more rigorous and conservative and less likely to claim the treatment as effective. See Appendix C.2 for details.

Measure	Definition / Explanation	Relevant sides
Conversion rate	$\frac{\sum_i \sum_s O_{is}}{\sum_i S_i}$	Consumers, restaurant partners
Basket value per order	$\frac{\sum_i \sum_s O_{is} B_{is}}{\sum_i \sum_s O_{is}}$	Restaurant/delivery partners
Retention rate	$\frac{\sum_i \sum_s O_{is} R_{is}}{\sum_i \sum_s O_{is}}$	Consumers, restaurant partners
Orders per consumer	$\frac{1}{I} \sum_i \sum_s O_{is}$	Consumers, restaurant/delivery partners
New restaurant impression ratio	% of impressions on new restaurants	Restaurant partners
New restaurant order ratio	% of orders on new restaurants	Restaurant partners

Table 3 List of Measurements.

5.2. Offline Analysis

5.2.1. Hyperparameter Selection with Offline Replay. The ranking function in Eq.(13) contains three hyperparameters λ_b, λ_r and λ_f controlling the relative importance of the objectives. It is costly to run online experiments to select the optimal values for these hyperparameters. It is also risky to serve a new framework in production with arbitrary hyperparameters before we have an understanding of their effects on the platform. Therefore, it is necessary to develop an offline evaluation procedure to pick hyperparameter values for MOHR to be experimented online.

The data for offline evaluation is critical for the quality of the evaluation, as we are faced with the typical challenge of position bias (Ursu 2018) and off-policy evaluation (Strehl et al. 2010, Schnabel et al. 2016). To understand position bias, the company has set aside a small percentage of random sessions for random ranking, where the vertical list of restaurants are ranked completely at random. Figure 3 confirms position bias on the number of impressions, number of orders and conversion rate on the random ranking data, showing that the same restaurant at different positions may appeal very differently to the consumers. Position bias causes challenges for performing off-policy evaluation. For example, if MOHR framework predicts to rank restaurant j_0 at a certain position for a consumer, but the existing production system has never presented restaurant j_0 at that position to her, then it is hard to predict whether the consumer would have ordered from that restaurant.

We adopt the *offline replay* method proposed by Li et al. (2011), which utilizes random data for off-policy evaluation, but in the context of bandit algorithms. We adapt their method to the ranking scenario. Specifically, if it happens that the new algorithm chooses the same restaurant to be ranked on the top position as in the random ranking data[18], then that event is retained and will be used for estimating the performance of the new algorithm. In other words, the replay method is essentially looking for events in the random ranking data that can serve as "replaying" the ranking under the new algorithm to be evaluated. The replay method is proven to provide unbiased offline evaluation (Li et al. 2011) without running online experiments.

Figure 4 shows the offline Pareto frontiers from the offline replay analysis[19]. As expected, larger values of λ_b and λ_f result in better basket value and marketplace fairness at the cost of the conversion objective[20]. As consumer conversion is the top-tier business metric for the company, for online experiments we pick the values for λ_b and λ_f such that the drop in conversion is minimal.

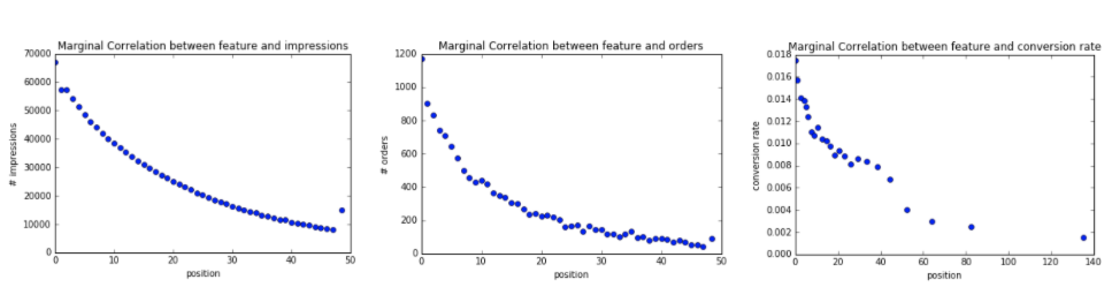


Figure 3 **Impressions (left) , orders (middle) and conversion (right) vs. position on random ranking data.**

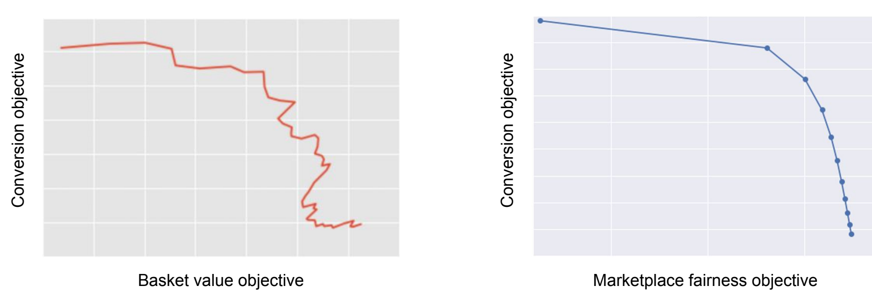


Figure 4 **Offline Pareto frontiers generated by the offline replay analysis. Left: conversion objective (y-axis) vs. basket value objective (x-axis); Right: conversion objective (y-axis) vs. marketplace fairness objective objective (x-axis); We omit the tick values for business compliance reasons. The upper and right directions correspond to better objective values.**

5.2.2. Feature Importance and Model Performance. See Appendix C.3 for the model performance and feature importance of the machine learning-based objectives in the MO-step.

5.3. Online Experiment Results

We present the results from the large-scale online controlled experiment (i.e. A/B testing) at the company. Compared with the latest production system at the company, there are two major changes from the MOHR framework: recommending for the three-sided marketplace using multi-objective optimization (MO-step and R-step), and recommending hierarchical items using hierarchical modeling (H-step). To understand the contributions of the two changes separately, we conduct three sets of experiments: (1) **multi-objective recommendation ("MOR")**: the MOHR framework *without* the hierarchical hierarchical modeling component (H-step), where the restaurants are ranked using the multi-objective ranking score in Eq.(13); (2) **hierarchical single-objective recommendation ("H")**: the MOHR framework *without* the multi-objective optimization component (MO-step and R-step), where the contents are ranked together holistically by the H-step, with conversion as the single objective; (3) **multi-objective hierarchical recommendation ("MOHR")**: the full MOHR framework combining (1) and (2).

5.3.1. Results on Multi-Objective Recommendation ("MOR"). We experiment with adding basket value, consumer retention and marketplace fairness objectives to the production system which uses conversion as the single objective. Without the H-step, the MOR framework is not applicable to rank the carousels (vertically) together with the restaurants. Therefore, we keep the production system's ranking for the carousels, while use the MOR framework for restaurant-level rankings, namely within-carousel (horizontal) ranking and vertical single restaurant ranking.

Constrained by the number of online experiments we can run on live consumer traffic, we adopt a greedy approach in understanding the effect of incorporating each new objective into the system. Specifically, we sequentially add more additive terms in the ranking function in Eq.(13).

Table 4 reports the metrics defined in Table 3 with statistically significant[21] differences between each treatment and control group. We see that by carefully picking the weights for each objective, we are able to achieve Pareto improvements for the three-sided marketplace without hurting consumer conversion. With the basket value objective, we observe a 0.5% relative increase in average basket value per order. In particular, the average basket value of the *top* recommended restaurant has increased by 4.5%, confirming the position effect of the treatments. With the retention objective, we observe a 0.7% relative increase in consumer 14-day retention, indicating that the consumers are coming back to the platform and ordering more often, which also leads to a 0.8% increase in orders per consumer. Lastly with the marketplace fairness objective, the number of impressions and orders on the new restaurants are more than doubled, increasing by 150% and 108% respectively, without a significant drop in the performance of the well-established restaurants on the platform. The fact that introducing the marketplace fairness objective by boosting new restaurants does not significantly hurt consumer conversion is an interesting result to us. This is explained by two observations. First, it has been shown in Appendix B.2.1 that the Pareto frontier for the constrained optimization problem is concave, in that a small sacrifice in one objective can lead to large improvements in others. In this case, the Pareto frontier is concave enough, so that we are able to achieve a significant improvement in marketplace fairness without hurting other objectives significantly. Second, this can be explained as the benefit of consumer exploration (Chen et al. 2021), where boosting new contents helps the consumers discover new interests, and arguably does not hurt consumer experience – sometimes even improving it.

	Basket value	Consumer retention	Marketplace fairness	Combined
Conversion rate	-	-	-	-
Basket value per order	+0.5%	-	-	+0.5%
Retention rate	-	+0.7%	-	+0.7%
Orders per consumer	-	+0.8%	-	+0.8%
New restaurants impression ratio	-	-	+150%	+150%
New restaurants order ratio	-	-	+108%	+108%

Table 4 Results on multi-objective recommendation ("MOR"). Metric differences that are statistically significant at 95% confidence interval are reported, in the form of relative changes over the control group.

The combined impact for MOR by including all the objectives is summarized as the last column in Table 4. Note that only the relative changes of the metrics are reported, as we are not allowed to reveal the actual values of the key business metrics for compliance reasons. Although the relative changes in the key metrics are small (less than 1%), they translate to considerable business impact given the large scale and consumer base of the company's global platform. Specifically, the MOR framework has led to $1.3 million weekly gain in revenue.

5.3.2. Results on Hierarchical Single-Objective Recommendation ("H"). A key input to the H-step is the consumer browsing model, which outputs the scrolling factors $p_{l,l+1}$ at each position l as defined in Eq.(4). We adopt a global estimation procedure that estimates a set of non-personalized scrolling factors for the consumer browsing model. Specifically, at each position l inside the carousel, we compute the ratio of the impressions happening at position l that are followed by an impression event at position $l+1$ as the estimate for $p_{l,l+1}$:

$$\hat{p}_{l,l+1} = \frac{\text{number of impressions happened at position } l+1}{\text{number of impressions happened at position } l}. \tag{14}$$

Appendix C.4 reports the estimated consumer scrolling factors. In practice, we find the global estimation works well. In addition to consumer conversion, we monitor two other metrics that are related to consumers' conversion behavior and the quality of the recommendations: *average vertical order position* and *search rate*. The former measures the average vertical position of an order in the homepage, and the latter measures the percentage of the sessions where the consumers go to the search tab, which is a signal that the recommendations on the homepage are not relevant or interesting to them.

Metric	Conversion rate	Average vertical order position	Search rate
Relative change over control	+1.5%**	-5.7%***	-0.9%***

Table 5 Results on hierarchical single-objective recommendation ("H"). Metrics are reported as relative changes over the control group. $^{***}p < 0.01$, $^{**}p < 0.05$.

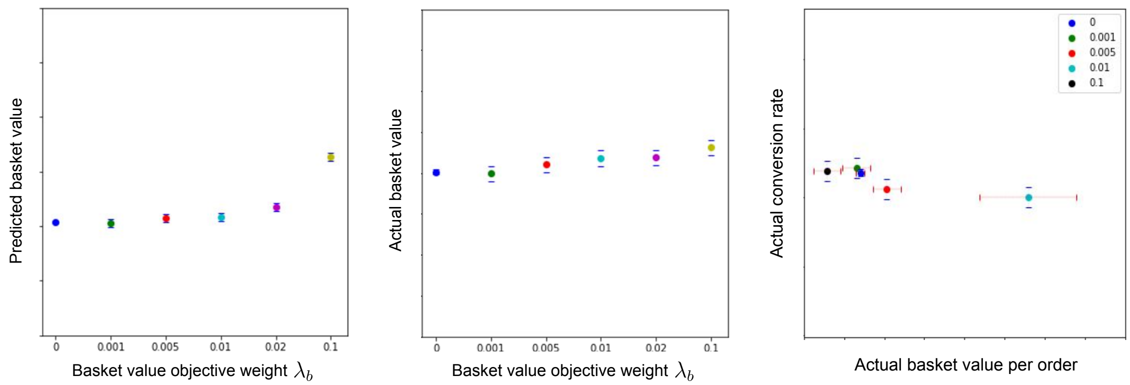
From Table 5, the hierarchical single-objective recommender improves conversion rate by 1.5%, which translates to $1.1 million weekly gain in revenue. There is a significant 5.7% reduction in average vertical order position and 0.9% reduction in search rate, indicating that the recommendations on the homepage are of higher relevance so the consumers don't need to scroll as much[22] or go to the search page to find what they want.

5.3.3. Results on the Full MOHR. Table 6 summarizes the results on the full MOHR as illustrated in Fig.2. We observe Pareto improvements in all key metrics, which together translates to $1.5 million weekly gain in revenue. Note that the improvements in conversion rate (+0.5%), average vertical order position (-3.2%) and search rate (-0.8%) are smaller compared with H-step only (Table 5). This is an expected result of the trade-off between the additional objectives (basket value, retention and marketplace fairness) and the original conversion objective, which also explains the fact that the revenue gain from the MOHR framework ($1.5 million weekly) is less than the sum of that from MOR ($1.3 million weekly) and H ($1.1 million weekly) treatment groups. Nevertheless, we would like to emphasize that compared with the latest production recommender system, our MOHR framework is able to deliver *Pareto improvements* on all key business metrics at no cost to any of the objectives or any sides in the marketplace.

Metric	Conversion rate	Basket value per order	Retention rate
Relative change	+0.5%**	+0.5%***	+0.7%***
Metric	Orders per consumer	New restaurants impression ratio	New restaurants order ratio
Relative change	0.9%***	+150%***	+108%***
Metric	Average vertical order position	Search rate	
Relative change	-3.2%***	-0.8%**	

Table 6 Results on the full multi-objective hierarchical recommender ("MOHR"). Metrics are reported as relative changes over the control group. $^{***}p < 0.01$, $^{**}p < 0.05$.

While MOHR effectively pushes forward the Pareto frontier for the three-sided marketplace, trade-offs still exist as they are the nature of multi-objective optimization. To better understand how the objective weights in the MOHR framework moderate the trade-offs among the online metrics, we conduct additional experiments on the basket value objective with varying λ_b. There are three findings as shown in Fig.5. First, with larger λ_b, the average *predicted* basket values of the recommended carousels and restaurants are also larger (Fig.5a) as expected. As a result, the *actual* basket size per order is also increased (Fig.5b). This confirms the effectiveness of the basket value objective and its weight λ_b as the tuning parameter. Note that the increasing trend in the actual basket value (Fig.5b) is less that in the predicted basket value (Fig.5a). This is expected as the recommendation algorithm has full control on what can be shown (predicted basket value), but only partial influence on what the consumers will order (actual basket value). Second, within a reasonable range of λ_b values, we see a trade-off between conversion rate and basket value (Fig.5c) in the online metrics, corroborating the findings from the offline analysis. Finally and most interestingly, when the weight λ_b is huge so that the ranking function in Eq.(13) is dominated by the basket value objective[23], we observe a significant +2.1% increase in search rate, and a 2.7% drop in retention of new consumers. This suggests two consequences when more expensive restaurants and carousels appear in the homepage recommendation: Consumers are more likely to abandon the homepage recommendation and search to order instead; New consumers (who are not yet familiar with the platform) are left with an impression that the selections on the platform are beyond their affordability, hurting their willingness to come back in the future. In other words, aggressive boosting of the objectives may backfire and hurt both the long-term consumer experience and the business.



(a) Average **predicted** basket value of the recommendation.
(b) Average **actual** basket value of the recommendation.
(c) Actual trade-off between consumer conversion and basket value.

Figure 5 Additional experiments on the basket value objective. Tick values are omitted for business compliance.

Because of its significant business impact, the MOHR framework has been deployed globally by the company and is currently serving as its recommender system for the homepage. It was one of the company's most successful launches over the past few years.

5.4. Robustness Checks

5.4.1. Randomization Check. To check if the random assignment for the online experiments truly holds, we inspect the treatment and control consumers *before* the experiment start date, when they were all receiving recommendations from the same production recommender system. This is called A/A test in the industry. Specifically, we collect 472 metrics related to different aspects of consumer behaviors[24] including the key business metrics in the results above, and conduct hypothesis testing on whether the differences between treatment and control groups are statistically significant before the experiment start date. We also perform Kolmogorov-Smirnov (KS) test on the empirical distribution of p-values for the 472 metric differences, and could not reject the null that they follow a uniform distribution on [0,1], suggesting that our randomization holds true. See Appendix C.5 for details.

5.4.2. Ablation Study on Consumer Browsing Model. We experiment with ablating the consumer browsing model in the H-step of MOHR. Specifically, we let $p_{0,1} = 1$ and $p_{l,l+1} = 0$ for $l > 0$, instead of using $\hat{p}_{l,l+1}$ in Eq.(14) as the consumer scrolling factors. Using the production recommender system as the baseline, the ablated MOHR framework decreases the average order position by 5.5%, which confirms the benefits of recommending carousels and single restaurants intelligently together, but does not change any other business metrics significantly. Compared with the results in Table 5 where the H-step increases conversion rate, retention of new consumers and decreases search rate, the result suggests that the hierarchical consumer browsing model is critical for the improvements from the H-step.

5.4.3. Eliminating the Novelty Effects. With the MOHR framework, carousels and single restaurants are mixed together and the homepage appears as a heterogeneous arrangement of items. One might argue that the new display may introduce a novelty effect (Koch et al. 2018) and consumers' engagement levels with the platform might be higher in the beginning than when they become familiar with the new design. We identified three pieces of evidence to counter this argument. First, the first three days of experiment data is discarded in computing the metrics reported in the previous sections, which eliminates part of the novelty effect. Second, there is an improvement in long-term consumer retention (+0.7%) which by definition measures consumers' future engagement with the platform after finishing the current session. This means that the treatment effect of MOHR persists for at least 28 days. Lastly, we measured the metrics for the new consumers during the experiment, whose *first* interaction with the platform is either always under the current production system (control consumers) or always under the new display under the MOHR framework (treatment consumers). Therefore, there is no novelty effect at play for those consumers. We observe a significant 5.5% increase in the retention of new consumers which is even larger than the overall retention increase. This suggests that the novelty effect, if it exists, actually impacts the effects of the MOHR framework negatively as it introduces a "shock" to the existing consumers with a new display, so that the positive effects on them are actually smaller than the new consumers who have no prior experience with the platform.

6. Discussion

6.1. Research Contributions

This paper proposes a general recommendation framework that addresses two of the most prominent challenges in a three-sided food delivery marketplace, namely multi-sided trade-off and hierarchical recommendation. We propose MOHR, which is a model-based three-step recommendation framework combining machine learning, hierarchical modeling and multi-objective optimization for recommending restaurants and aggregation of restaurants in the homepage of the food-delivery platform. In the first step (MO-step), we develop machine learning models for real-time personalized predictions of the multiple objectives at individual restaurant level, with content-based, collaborative-filtering based and real-time contextual features as input. In the second step (H-step), we adopt a probabilistic structural model for the predictions of the multiple objectives for aggregations of restaurants, or carousels. Specifically, carousel-level objectives are modeled as an aggregation of individual restaurant-level objectives, using a consumer browsing model which captures consumers' browsing patterns on the homepage. The aggregated predictions are calibrated against those for the single restaurants, which ensures consistent consumer experience across different levels of aggregation of the recommended items, and provides levels of transparency to the restaurant partners. In the final step (R-step), we formulate the multi-objective recommendation problem as a constrained multi-objective optimization problem, taking as input the predictions for the multiple objectives at different levels of aggregations from the previous step. The variables are the probabilities of serving each hierarchical item to each consumer, and the constraints specify the amount of tolerable trade-off among the multiple objectives. With a quadratic penalty term added to the objective function, the solution becomes a combination of the multiple objectives with an analytical form. Each objective is associated with a weight, which we treat as tuning parameters controlling the trade-off across multiple objectives. The output of the framework is a hierarchical ranking function that accounts for consumers' browsing patterns and combines multiple objectives, providing recommendations on heterogeneous contents that are optimized for the three-sided marketplace.

Methodologically, the benefits of the proposed MOHR framework are three-fold. First, it provides a general and mathematically principled way to model and optimize for the multiple sides in the marketplace, all of which are crucial to the success of the business.

The weights associated with each objective can be treated as tuning parameters, which offers practitioners full control over the trade-off across multiple objectives. The offline Pareto frontiers generated by the replay analysis further facilitates the understanding and decision-making under the multi-sided trade-off when online experiments are expensive. Second, the hierarchical modeling approach guarantees the interpretability of the ranking function, and that the predictions for the aggregation of items are calibrated against those for the single items. This ensures consistent consumer experience across different levels of item aggregation, and provides transparency to the restaurant partners. Lastly, the analytical solution from the R-step provides a fast and efficient way to do hierarchical recommendation without the need to solve huge linear programming problems online, making it possible to serve the MOHR framework in any large-scale online systems in real time.

6.2. Managerial Implications

Our proposed MOHR framework is general, flexible and can be readily applied to other recommendation applications within and outside the food delivery industry. Doordash and Grubhub as examples of other food delivery platforms, YouTube as a video streaming platform, and Airbnb as a peer home-sharing platform, are all operating in multi-sided marketplaces and the recommendation contents can be heterogeneous and hierarchical. The MOHR framework illustrated in Fig.2 is readily applicable to these platforms, with any number of objectives. The holistic framework also reduces the burden of maintaining separate machine learning systems for ranking and recommending contents of different levels of aggregations.

Components of the MOHR framework can be applied in a modularized fashion. Section 5.3 demonstrates that a subset of its components, namely MOR and H, can act as a complete framework to address a particular challenge. Therefore, if a platform is concerned with multi-objective recommendation in a multi-sided marketplace but the recommendation contents are not hierarchical, it can adopt the MOHR framework without the H-step (i.e "MOR" in Section 5.3.1); if a platform is concerned with hierarchical recommendation but do not need to optimize for more than one objective, it can adopt only the H-step (i.e "H" in Section 5.3.2).

Results and analysis under the MOHR framework provide insights on the trade-offs among multiple objectives in a multi-sided marketplace. On one of the world's largest food delivery platform, we experimented with objectives including conversion and retention for the consumers, marketplace fairness for the restaurant partners, and earnings for the delivery partners. Compared with the latest production system, the MOHR framework is able to achieve Pareto improvements for all objectives. In particular, it improves long-term consumer experience (retention), marketplace fairness and partner earnings without significantly impacting consumers' short-term engagement (conversion). Within the MOHR framework, trade-offs exist as a natural outcome for optimizing multiple conflicting objectives. As it is expensive to generate the full Pareto frontiers in online experiments, we propose to adopt offline replay analysis to generate Pareto frontiers using offline data, to help understand the trade-off between multiple objectives for the three-sided marketplace. We also observe that if the weight for a particular objective is too large, it will hurt overall consumer experience and backfire. For example, an aggressive boost of the basket value objective leads the new consumers on the platform to believe that the selections on the platform are expensive, hurting the long-term experience of those with low price elasticity. Insights like these help inform better managerial decision-making on multi-sided platforms.

Lastly, we would like to call out the connection between the proposed marketplace fairness objective and the cold-start problem as a well-known challenge for recommender systems. For new items or items with low exposure on the platform (i.e. cold-started items), the marketplace fairness objective assigns a high value to them, leading to an increased exposure. This also helps the machine learning models generate more accurate predictions for the new restaurants. Over time, as the new restaurants accumulate more exposure, the marketplace fairness objective assigns a lower value to them, leading to a "graduation" from the cold-start phase. As a result, the items will be recommended mainly based on the values of the other objectives. Therefore, the proposed marketplace fairness objective addresses the cold-start challenge on the restaurant side in a dynamic, adaptive and data-dependent fashion. In addition, from our experiments the marketplace fairness objective does not necessarily hurt consumer experience, which can be explained as the benefit of consumer exploration (Chen et al. 2021), where boosting new contents helps the consumers discover new interests and potentially improves long-term consumer experience.

6.3. Challenges, Limitations and Future Research

A challenge and limitation of the MOHR framework is its scalability with a large number of objectives. With an increasing number of objectives added, it could become unscalable to tune the weights for each objective in an A/B testing framework with a combinatorial number of candidates for the weight vector. Multi-armed bandit experiments (Burtini et al. 2015) are more efficient experiment designs than A/B testing, where the experiment traffic is dynamically allocated to different treatment groups based on their short-term performance metrics. However, they are not feasible for long-term objectives such as consumer retention in our application, which requires the consumer to consistently receive the treatment for an extended period of time. Another alternative is to learn the optimal weight combination offline using more sophisticated methods such as Bayesian optimization. However, we found those methods suffer from training-serving skew due to its off-policy nature, which introduces additional challenges for off-policy learning in addition to the reward design. In practice, we adopt a greedy approach for adding new objectives, where each objective is added and tuned sequentially. This reduces the tuning complexity from exponential to linear in the number of objectives.

The consumer browsing model has two limitations. First, in our application the model is a global static estimate based on a snapshot of consumer behavior logs. It is not personalized and could become outdated after the model is launched to global traffic. In addition, different consumers have different browsing patterns, and even the same consumer could have different browsing patterns under different contexts. A future research direction is to build a personalized and real-time consumer browsing model, which takes as input the consumer's history, current in-session behavior and real-time contextual features, and generates a real-time prediction of the probability that the consumer will continue scrolling. The whole MOHR framework still holds in this case, but with $p_{l,l+1}$ in Eq.(4) plugged in as the output from a personalized real-time machine learning model instead. Second, the consumer browsing model assumes a linear browsing pattern (i.e. consumers inspect one item at a time without going back and forth), following the sequential search framework proposed by Weitzman (1979). This assumption can be relaxed by assuming that the consumers first inspect a set of items and then choose one from the set, which calls for a choice modeling component with position bias taken into account.

Lastly, the objectives in the MOHR framework are estimated by separate machine learning models. However, different objectives may be related to each other in addition to the conflicts, and one may leverage the relatedness for better predictive power. For example, consumers' short-term engagement might be indicative of their long-term happiness.

Multi-task deep learning models (Ruder 2017) are well-suited in this case to jointly and efficiently learn multiple related and conflicting objectives. The multiple machine learning models in the MO-step can be replaced with a single large multi-task deep learning model, with the other components of MOHR unchanged.

Endnotes

[1]Therefore, earnings for the restaurant partners, delivery partners and the platform are all positively correlated with consumers' payments, or gross bookings.

[2]If the restaurant appears as a single recommendation time, we say it belongs to a single-restaurant carousel.

[3]The 14 day time window is aligned with the key business metric for the company.

[4]The condition is counterfactual, meaning that the machine learning model will have a prediction for this objective regardless of whether the consumer orders in the current session.

[5]Theoretically it is possible for gradient boosting regression trees to generate negative predictions even if all training labels are positive, although we didn't observe this from our models.

[6]We also experimented with 90 days and 180 days as the time window. The results were not statistically different.

[7]When the consumer abandons the current carousel, she can either go to the next carousel that's immediately below the current one, or abandon the session completely. It is easy to show that the vertical browsing behavior does not affect the pointwise ranking algorithms, so we did not explicitly model them in the MOHR framework.

[8]This is empirically guaranteed to be true by the design of the homepage of the app.

[9]We drop the dependency on z in $c(i,j,k), c(i,k), b(i,j,k), b(i,k), r(i,j,k), r(i,k)$ for ease of notation, but we would like to emphasize that these estimates all take contextual features z as input.

[10]From our empirical data, in more than 99.98% of the sessions are with zero or one order. Therefore it is reasonable to assume that the consumer places at most one order in the current session.

[11]When $n = 1$, Eq.(5) computes the conversion rate of a single-restaurant carousel, which equals the conversion rate of the only restaurant inside it.

[12]The formulation is equivalent to having $B(\boldsymbol{x})$, $R(\boldsymbol{x})$ or $F(\boldsymbol{x})$ as the objective while constraining on others. This is because the primal problem in Eq.(12) is feasible and bounded, so strong duality holds.

[13]Solution $\boldsymbol{x}_1$ is said to dominate solution $\boldsymbol{x}_2$ if $(C(\boldsymbol{x}_1), B(\boldsymbol{x}_1), R(\boldsymbol{x}_1), F(\boldsymbol{x}_1)) \geq (C(\boldsymbol{x}_2), B(\boldsymbol{x}_2), R(\boldsymbol{x}_2), F(\boldsymbol{x}_2))$ element-wisely, and at least one of the inequalities is strict. A solution $\boldsymbol{x}$ is called *Pareto optimal* if there is no solution $\boldsymbol{x}' \neq \boldsymbol{x}$ such that $\boldsymbol{x}'$ dominates $\boldsymbol{x}$. *Pareto frontier* is the set of all Pareto optimal solutions.

[14]The benefit of having analytical solutions is that we don't need to solve the large-scale linear programming problem online, and only need to plug in the values for the analytical form instead.

[15]Note that a 28-day experiment is considered to be a long-term experiment at the company, and the long-term consumer retention metric the company monitors is also defined using a 28-day window.

[16]We are not allowed to disclose the number of consumers, sessions and orders from the experiments due to business compliance reasons. But we would like to point out that given the large scale of the business, the data gives us more than enough statistical power to conduct hypothesis testing on the performance metrics defined in the next section.

[17]We only look at signed-in consumers as sign-in is required to place an order on the app.

[18]The random ranking data provided by the company is restaurant-level random ranking and we unfortunately don't have carousel-level random ranking data from the company. Nevertheless, the selected parameters from the replay analysis using the restaurant-level random ranking data perform reasonably well in the online experiments.

[19]We unfortunately could not generate the Pareto frontier for the consumer retention objective using the replay method. The reason is that it requires at least 28 days of random ranking data to observe consumer retention, but the random ranking data we have from the company is only one week.

[20]Note that the Pareto frontier for the basket value objective is noisy, while the Pareto frontier for the marketplace fairness objective is much smoother. This is expected as the marketplace fairness objective is measured by the number of *impressions* a restaurant receives, which the recommender system has (almost) full control by determining which restaurants to put on top. On the other hand, the basket value objective depends on the consumer *placing an order* on the restaurant, which is a stochastic event that the recommender system only has partial influence on. In other words, the basket value objective incorporates one extra layer of randomness, leading to a noisier Pareto frontier.

[21]The statistical significance is measured under 0.95 confidence level.

[22]Note that the control group ranks all carousels on top of all single restaurants. So the MOHR framework actually presents *fewer* contents in the top positions, yet it's still able to reduce the average vertical order position by 5.7% compared with control. This further confirms the increased quality of the homepage.

[23]The basket value objective, which is measured in dollar amounts, is about 2 orders of magnitude larger than the other three objectives. Therefore $\lambda_b = 0.1$ means that the term for the basket value objective, $\lambda_b c_{iq} b_{iq}$, is roughly 10 times the value of the other terms, making the ranking function in Eq.(13) dominated by the basket value objective.

[24]We unfortunately don't have access to consumers' personal attributes such as age, gender and demographic information.

7. Declarations

Funding and Competing Interests. All authors have received research support from Uber. Uber has a right to remove its intellectual property or trade secrets from the paper.

References

Abdollahpouri H, Adomavicius G, Burke R, Guy I, Jannach D, Kamishima T, Krasnodebski J, Pizzato L (2020) Multistakeholder recommendation: Survey and research directions. *User Modeling and User-Adapted Interaction* 30(1):127–158.

Adomavicius G, Bockstedt JC, Curley SP, Zhang J (2013) Do recommender systems manipulate consumer preferences? a study of anchoring effects. *Information Systems Research* 24(4):956–975.

Adomavicius G, Bockstedt JC, Curley SP, Zhang J (2018) Effects of online recommendations on consumers' willingness to pay. *Information Systems Research* 29(1):84–102.

Adomavicius G, Tuzhilin A (2005) Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering* 17(6):734–749.

Agarwal D, Chatterjee S, Yang Y, Zhang L (2015) Constrained optimization for homepage relevance. *Proceedings of the 24th International Conference on World Wide Web*, 375–384.

Agarwal D, Chen BC, Elango P, Wang X (2011) Click shaping to optimize multiple objectives. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 132–140.

Agarwal D, Chen BC, Elango P, Wang X (2012) Personalized click shaping through lagrangian duality for online recommendation. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, 485–494.

Aggarwal CC, et al. (2016) *Recommender systems*, volume 1 (Springer).

Agrawal R, Gollapudi S, Halverson A, Ieong S (2009) Diversifying search results. *Proceedings of the second ACM international conference on web search and data mining*, 5–14.

Alvino C, Basilico J (2015) Learning a personalized homepage. *The Netflix Tech Blog* 9.

Anderson A, Maystre L, Anderson I, Mehrotra R, Lalmas M (2020) Algorithmic effects on the diversity of consumption on spotify. *Proceedings of The Web Conference 2020*, 2155–2165.

Aridor G, Gonçalves D (2021) Recommenders' originals: The welfare effects of the dual role of platforms as producers and recommender systems. *Available at SSRN 3928005* .

Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47(2):235–256.

Azaria A, Hassidim A, Kraus S, Eshkol A, Weintraub O, Netanely I (2013) Movie recommender system for profit maximization. *Proceedings of the 7th ACM conference on Recommender systems*, 121–128.

Backstrom L, Leskovec J (2011) Supervised random walks: predicting and recommending links in social networks. *Proceedings of the fourth ACM international conference on Web search and data mining*, 635–644.

Balabanović M, Shoham Y (1997) Fab: content-based, collaborative recommendation. *Communications of the ACM* 40(3):66–72.

Bennett J, Lanning S, et al. (2007) The netflix prize. *Proceedings of KDD cup and workshop*, volume 2007, 35 (New York, NY, USA.).

Billsus D, Pazzani MJ, et al. (1998) Learning collaborative information filters. *Icml*, volume 98, 46–54.

Breese JS, Heckerman D, Kadie C (1998) Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 43–52.

Breiman L, Friedman JH, Olshen RA, Stone CJ (2017) *Classification and regression trees* (Routledge).

Brusilovsky P (2007) Adaptive navigation support. *The adaptive web*, 263–290 (Springer).

Burke R (2002) Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction* 12(4):331–370.

Burtini G, Loeppky J, Lawrence R (2015) A survey of online experiment design with the stochastic multi-armed bandit. *arXiv preprint arXiv:1510.00757* .

Carare O (2012) The impact of bestseller rank on demand: Evidence from the app market. *International Economic Review* 53(3):717–742.

Carbonell J, Goldstein J (1998) The use of mmr, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335–336.

Chaney AJ, Stewart BM, Engelhardt BE (2018) How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. *Proceedings of the 12th ACM Conference on Recommender Systems*, 224–232.

Chen LS, Hsu FH, Chen MC, Hsu YC (2008) Developing recommender systems with the consideration of product profitability for sellers. *Information Sciences* 178(4):1032–1048.

Chen M, Wang Y, Xu C, Le Y, Sharma M, Richardson L, Wu SL, Chi E (2021) Values of user exploration in recommender systems. *Fifteenth ACM Conference on Recommender Systems*, 85–95.

Chen Y, Yao S (2017) Sequential search with refinement: Model and application with click-stream data. *Management Science* 63(12):4345–4365.

Click C, Malohlava M, Candel A, Roark H, Parmar V (2017) Gradient boosting machine with h2o. *H2O.ai* .

Covington P, Adams J, Sargin E (2016) Deep neural networks for youtube recommendations. *Recsys' 16*, 191–198.

Das A, Mathieu C, Ricketts D (2009) Maximizing profit using recommender systems. *arXiv:0908.3633* .

Datta H, Knox G, Bronnenberg BJ (2018) Changing their tune: How consumers' adoption of online streaming affects music consumption and discovery. *Marketing Science* 37(1):5–21.

Delgado J, Ishii N (1999) Memory-based weighted majority prediction. *SIGIR Workshop Recomm. Syst. Citeseer*, 85 (Citeseer).

Dhillon PS, Aral S (2021) Modeling dynamic user interests: A neural matrix factorization approach. *Marketing Science* 40(6):1059–1080.

Evans DS, Schmalensee R (2016) *Matchmakers: The new economics of multisided platforms* (Harvard Business Review Press).

Evans DS, Schmalensee R, Noel MD, Chang HH, Garcia-Swartz DD (2011) Platform economics: Essays on multi-sided businesses. *Platform Economics: Essays on Multi-Sided Businesses, David S. Evans, Ed., Competition Policy International* .

Fleder D, Hosanagar K (2009) Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. *Management science* 55(5):697–712.

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. *Annals of statistics* 1189–1232.

Friedman JH (2002) Stochastic gradient boosting. *Computational statistics & data analysis* 38(4):367–378.

Ghose A, Ipeirotis PG, Li B (2012) Designing ranking systems for hotels on travel search engines by mining user-generated and crowdsourced content. *Marketing Science* 31(3):493–520.

Gomez-Uribe CA, Hunt N (2015) The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)* 6(4):1–19.

Guan Z, Cutrell E (2007) An eye tracking study of the effect of target rank on web search. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 417–420.

Hastie T, Tibshirani R, Friedman JH, Friedman JH (2009) *The elements of statistical learning: data mining, inference, and prediction*, volume 2 (Springer).

Hosanagar K, Fleder D, Lee D, Buja A (2014) Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation. *Management Science* 60(4):805–823.

Hosanagar K, Krishnan R, Ma L (2008) Recomended for you: The impact of profit incentives on the relevance of online recommendations. *ICIS 2008 Proceedings* 31.

Hu Y, Koren Y, Volinsky C (2008) Collaborative filtering for implicit feedback datasets. *2008 Eighth IEEE international conference on data mining*, 263–272 (Ieee).

Katehakis MN, Veinott Jr AF (1987) The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research* 12(2):262–268.

King J, Imbrasaitė V (2015) Generating music playlists with hierarchical clustering and q-learning. *European Conference on Information Retrieval*, 315–326 (Springer).

Koch M, von Luck K, Schwarzer J, Draheim S (2018) The novelty effect in large display deployments–experiences and lessons-learned for evaluating prototypes. *Proceedings of 16th European conference on computer-supported cooperative work-exploratory papers* (European Society for Socially Embedded Technologies (EUSSET)).

Koren Y (2009) Collaborative filtering with temporal dynamics. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 447–456.

Krasnodebski J, Dines J (2016) Considering supplier relations and monetization in designing recommendation systems. *Proceedings of the 10th ACM Conference on Recommender Systems*, 381–382.

Langford J, Zhang T (2007) The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in neural information processing systems* 20(1):96–1.

Lee EL, Lou JK, Chen WM, Chen YC, Lin SD, Chiang YS, Chen KT (2014) Fairness-aware loan recommendation for microfinance services. *Proceedings of the 2014 international conference on social computing*, 1–4.

Li L, Chen J, Raghunathan S (2018) Recommender system rethink: Implications for an electronic marketplace with competing manufacturers. *Information Systems Research* 29(4):1003–1023.

Li L, Chu W, Langford J, Wang X (2011) Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. *Proceedings of the fourth ACM international conference on Web search and data mining*, 297–306.

Li Z, Fang X, Bai X, Sheng ORL (2017) Utility-based link recommendation for online social networks. *Management Science* 63(6):1938–1952.

Liebman E, Saar-Tsechansky M, Stone P (2019) The right music at the right time: Adaptive personalized playlists based on sequence modeling. *MIS Quarterly* 43(3).

Ma H, Liu X, Shen Z (2016) User fatigue in online news recommendation. *Proceedings of the 25th International Conference on World Wide Web*, 1363–1372.

Miller BN, Albert I, Lam SK, Konstan JA, Riedl J (2003) Movielens unplugged: experiences with an occasionally connected recommender system. *Proceedings of the 8th international conference on Intelligent user interfaces*, 263–266.

Milojkovic N, Antognini D, Bergamin G, Faltings B, Musat C (2019) Multi-gradient descent for multi-objective recommender systems. *arXiv preprint arXiv:2001.00846* .

Mooney RJ, Roy L (2000) Content-based book recommending using learning for text categorization. *Proceedings of the fifth ACM conference on Digital libraries*, 195–204.

Muangmee C, Kot S, Meekaewkunchorn N, Kassakorn N, Khalid B (2021) Factors determining the behavioral intention of using food delivery apps during covid-19 pandemics. *Journal of theoretical and applied electronic commerce research* 16(5):1297–1310.

Narayanan S, Kalyanam K (2015) Position effects in search advertising and their moderators: A regression discontinuity approach. *Marketing Science* 34(3):388–407.

Nguyen P, Dines J, Krasnodebski J (2017) A multi-objective learning to re-rank approach to optimize online marketplaces for multiple stakeholders. *arXiv preprint arXiv:1708.00651* .

Nielsen J (2006) " f-shaped pattern for reading web content," jakob nielsen's alertbox. *http://www. useit. com/alertbox/reading_pattern. html* .

Noulas A, Scellato S, Lathia N, Mascolo C (2012) A random walk around the city: New venue recommendation in location-based social networks. *2012 International Conference on Privacy, Security, Risk and Trust (PASSAT)*, 144–153 (IEEE Computer Society).

Oestreicher-Singer G, Sundararajan A (2012) The visible hand? demand effects of recommendation networks in electronic markets. *Management science* 58(11):1963–1981.

Pathak B, Garfinkel R, Gopal RD, Venkatesan R, Yin F (2010) Empirical analysis of the impact of recommender systems on sales. *Journal of Management Information Systems* 27(2):159–188.

Prawesh S, Padmanabhan B (2014) The "most popular news" recommender: Count amplification and manipulation resistance. *Information Systems Research* 25(3):569–589.

Ricci F, Rokach L, Shapira B (2015) Recommender systems: introduction and challenges. *Recommender systems handbook*, 1–34 (Springer).

Rochet JC, Tirole J (2003) Platform competition in two-sided markets. *Journal of the european economic association* 1(4):990–1029.

Rodriguez M, Posse C, Zhang E (2012) Multiple objective optimization in recommender systems. *Proceedings of the sixth ACM conference on Recommender systems*, 11–18.

Ruder S (2017) An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* .

Sahoo N, Singh PV, Mukhopadhyay T (2012) A hidden markov model for collaborative filtering. *MIS quarterly* 1329–1356.

Sawaragi Y, Nakayama H, Tanino T (1985) *Theory of multiobjective optimization* (Elsevier).

Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, 253–260.

Schnabel T, Swaminathan A, Singh A, Chandak N, Joachims T (2016) Recommendations as treatments: Debiasing learning and evaluation. *international conference on machine learning*, 1670–1679 (PMLR).

Sherman C (2005) A new f-word for google search results. *Search Engine Watch* (March 7).

Shi SW, Trusov M (2021) The path to click: Are you on it? *Marketing Science* 40(2):344–365.

Smith B, Linden G (2017) Two decades of recommender systems at amazon. com. *Ieee internet computing* 21(3):12–18.

Solsman JE (2018) Youtube's ai is the puppet master over most of what you watch. *CNET, January* 10.

Song Y, Sahoo N, Ofek E (2019) When and how to diversify—a multicategory utility model for personalized content recommendation. *Management Science* 65(8):3737–3757.

Strehl A, Langford J, Li L, Kakade SM (2010) Learning from logged implicit exploration data. *Advances in neural information processing systems* 23.

Thusoo A, Sarma JS, Jain N, Shao Z, Chakka P, Anthony S, Liu H, Wyckoff P, Murthy R (2009) Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment* 2(2):1626–1629.

Tintarev N, Sullivan E, Guldin D, Qiu S, Odjik D (2018) Same, same, but different: algorithmic diversification of viewpoints in news. *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*, 7–13.

Tso-Sutter KH, Marinho LB, Schmidt-Thieme L (2008) Tag-aware recommender systems by fusion of collaborative filtering algorithms. *Proceedings of the 2008 ACM symposium on Applied computing*, 1995–1999.

Ursu RM (2018) The power of rankings: Quantifying the effect of rankings on online consumer search and purchase decisions. *Marketing Science* 37(4):530–552.

Wang Y, Wang X, Beutel A, Prost F, Chen J, Chi EH (2021) Understanding and improving fairness-accuracy trade-offs in multi-task learning. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1748–1757.

Weitzman ML (1979) Optimal search for the best alternative. *Econometrica: Journal of the Econometric Society* 641–654.

Wu Q, Wang H, Hong L, Shi Y (2017) Returning is believing: Optimizing long-term user engagement in recommender systems. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1927–1936.

Xiao B, Benbasat I (2007) E-commerce product recommendation agents: Use, characteristics, and impact. *MIS quarterly* 137–209.

Xie R, Zhang S, Wang R, Xia F, Lin L (2021) Hierarchical reinforcement learning for integrated recommendation. *Proceedings of AAAI.*

Xie X (2010) Potential friend recommendation in online social network. *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, 831–835 (IEEE).

Zhang S, Yao L, Sun A, Tay Y (2019) Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52(1):1–38.

Zhang TC, Agarwal R, Lucas Jr HC (2011) The value of it-enabled retailer learning: personalized product recommendations and customer store loyalty in electronic markets. *Mis Quarterly* 859–881.

Zhang X, Ferreira P, Godinho de Matos M, Belo R (2021) Welfare properties of profit maximizing recommender systems: Theory and results from a randomized experiment. *MIS Quarterly* 45(1).

Zhao Z, Hong L, Wei L, Chen J, Nath A, Andrews S, Kumthekar A, Sathiamoorthy M, Yi X, Chi E (2019) Recommending what video to watch next: a multitask ranking system. *Proceedings of the 13th ACM Conference on Recommender Systems*, 43–51.

Zheng L, Li L, Hong W, Li T (2013) Penetrate: Personalized news recommendation using ensemble hierarchical clustering. *Expert Systems with Applications* 40(6):2127–2136.

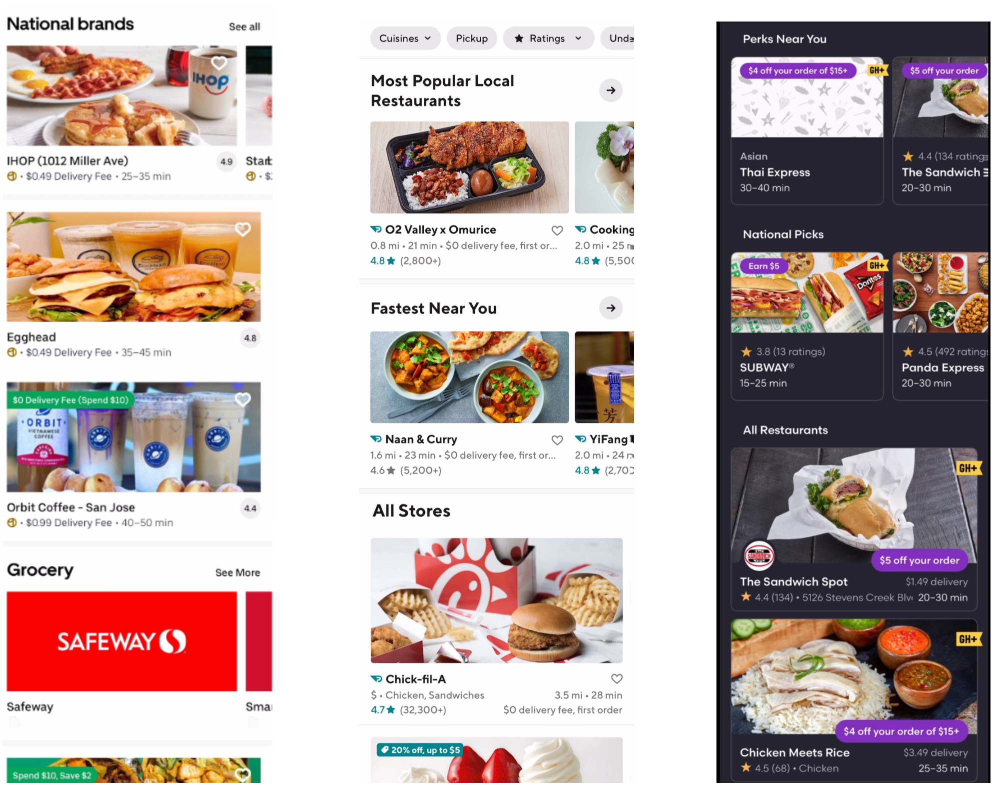
Zheng Y, Ghane N, Sabouri M (2019) Personalized educational learning with multi-stakeholder optimizations. *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, 283–289.

Zhu J (2021) Leveraging the pipeline design pattern to modularize recommendation services. URL `https://doordash.engineering/blog/`.

Zou L, Xia L, Ding Z, Song J, Liu W, Yin D (2019) Reinforcement learning to optimize long-term user engagement in recommender systems. *KDD' 19*, 2810–2818.

Appendix. Recommending for a Three-Sided Food Delivery Marketplace: A Multi-Objective Hierarchical Approach

A. Illustration of hierarchical recommendation items



(a) Uber Eats. (b) DoorDash. (c) Grubhub.

Figure 6 **Screenshots of the homepage of three major food delivery apps.**

B. Additional Technical Details for MOHR

B.1. MO-Step

B.1.1. Collaborative filtering features based on matrix factorization. To leverage the idea of collaborative filtering that similar consumers enjoy similar contents, we build matrix factorization models to learn a latent vector representation for every consumer, restaurant and source. Suppose there are I consumers and J restaurants in total, we learn their latent representations by

$$\{u_i\}_{i=1}^I, \{v_j\}_{j=1}^J = \underset{\{u_i\}_{i=1}^I, \{v_j\}_{j=1}^J}{\arg\min} \sum_{i=1}^I \sum_{j=1}^J (u_i^T v_j - r_{ij})^2 + \lambda_u \sum_{i=1}^I \|u_i\|_2 + \lambda_v \sum_{j=1}^J \|v_j\|_2, \tag{15}$$

where r_{ij} is the number of orders between consumer i and restaurant j, λ_u and λ_v are positive penalization coefficients preventing the optimization from learning wild values. This optimization problem also has a Bayesian interpretation with Gaussian prior on the representations, in which case λ_u and λ_v are determined by the variance parameter of the priors. See Section 4.2 in Dhillon and Aral (2021) as an example.

Eq.(15) is a biconvex problem and can be solved efficiently using alternating least squares (ALS) (Koren 2009). The output of the optimization problem, u_i's and v_j's, are used as latent representations for the consumers and restaurants. We build another matrix factorization model on ($consumer, source$) level similar to Equation (15) but changed v_j to source representation $\tilde{w_k}$ and r_{ij} to r_{ik}, the order counts between consumer i and source k:

$$\{\tilde{u_i}\}_{i=1}^{I}, \{\tilde{w_k}\}_{k=1}^{K} = \underset{\{\tilde{u_i}\}_{i=1}^{I}, \{\tilde{w_k}\}_{k=1}^{K}}{\arg\min} \sum_{i=1}^{I}\sum_{k=1}^{K}(\tilde{u_i}^T\tilde{w_k} - r_{ik})^2 + \lambda_u \sum_{i=1}^{I}\|\tilde{u_i}\|_2 + \lambda_w \sum_{k=1}^{K}\|\tilde{w_k}\|_2, \tag{16}$$

and obtain another set of representations, which are $\tilde{u}_i$ for consumers and $\tilde{w}_k$ for sources. For the individual machine learning models defined in Eq.(1), u_i and $\tilde{u}_i$ are included as part of consumer-level features x_i, v_j is part of restaurant-level features x_j, and $\tilde{w}_k$ is part of source-level features x_k.

B.1.2. Details for the machine learning models. Table 7 summarizes the features used for predicting consumer conversion, consumer retention and basket value. Note that for the count-based features such as the number of impressions/views/orders, we include both the raw count and the *normalized count* as features, where the normalized count are divided by the average impression/view/order count at the position of the event, in order to correct for the position bias as illustrated in Fig.3. The name/id of the source (e.g. i.e. the name of the carousel or "single" if the training instance is a single restaurant recommendation) is explicitly used as a feature.

For the gradient boosted trees as the predictive machine learning model, we use learning rate of 0.1, and maximum depth of the tree as 8, which is the same model architecture and capacity as the latest production system at the company. For the conversion rate model, the training data is unbalanced with a positive sample ratio as low as 1.8%, which could potentially cause challenges to the binary classification models. We therefore experimented with down-sampling the negative examples. However, we didn't find a performance boost on the test data, which could be explained as fact that the training data is big enough (around 600 million). In the final version of MOHR, the individual machine learning models are all trained without data reweighting or resampling.

B.1.3. Bayesian modeling for the marketplace fairness objective. We now describe the Bayesian modeling procedure to estimate $\sigma(j)$, the posterior variance for $c(i,j,k)$ as the value for the fairness objective. The order event $O(i,j,k,z)$ is a Bernoulli random variable with parameter $c(i,j,k)$. Therefore, we choose Beta distribution as the prior for $c(i,j,k)$. Proposition 1 below states the posterior for $c(i,j,k)$.

Proposition 1 *Suppose the prior distribution for $c(i,j,k)$ is $\mathcal{B}(\alpha_j,\beta_j)$, and that there are N_j impressions on restaurant j, out of which N_j^1 lead to orders. Then the posterior distribution for $c(i,j,k)$ is $\mathcal{B}(\alpha_j+N_j^1, \beta_j+N_j-N_j^1)$, and its posterior variance is $\sigma(j)^2 = \frac{(\alpha_j+N_j^1)(\beta_j+N_j-N_j^1)}{(\alpha_j+\beta_j+N_j)^2(\alpha_j+\beta_j+N_j+1)}$.*

Proof of Proposition 1. For ease of notation we drop the dependency on i, k for now and denote $c(i,j,k)$ as c_j for restaurant j. Suppose there are N_j impressions on restaurant j, $O_{j1},...,O_{jN_j}$ are random variables represents the corresponding conversion events where $O_{jm}=1$ means the m-th impression on restaurant j

Feature	Definition
x_i	(Normalized) Impression/view/order count/ratio from consumer i in the past X days Average basket values from consumer i in the past X days Consumer embedding u_i and $\tilde{u_i}$ from matrix factorization
x_j	(Normalized) Impression/view/order count/ratio on restaurant j in the past X days Average basket values from restaurant j in the past X days Percentage of consumers churned after ordering from restaurant j in the past X days Restaurant embedding v_j from matrix factorization Delivery radius of restaurant j
x_k	Source name (name of the carousel or "single" if appearing as single restaurant recommendation) (Normalized) Impression/view/order count/ratio from source k in the past X days Average basket values from source k in the past X days Source embedding $\tilde{w_k}$ from matrix factorization
x_{ij}	(Normalized) Impression/view/order count/ratio between consumer i and restaurant j in past X days Haversine distance between restaurant j and consumer's delivery location Estimated delivery time range Delivery fee, busy area fee, service fee $u_i^T v_j$, i.e. dot product of consumer embedding and restaurant embedding from matrix factorization $\frac{cos(u_i,v_j)}{\|\|u_i\|\|\|\|v_j\|\|}$, i.e. cosine similarity between consumer embedding and restaurant embedding
x_{ik}	(Normalized) Impression/view/order count/ratio from consumer i in source k in the past X days Average basket values from consumer i in source k in the past X days $\tilde{u_i}^T \tilde{w_k}$, i.e. dot product of consumer embedding and source embedding from matrix factorization $\frac{cos(\tilde{u_i},\tilde{w_k})}{\|\|\tilde{u_i}\|\|\|\|\tilde{w_k}\|\|}$, i.e. cosine similarity between consumer embedding and source embedding
x_{jk}	(Normalized) Impression/view/order count/ratio from restaurant j in source k in the past X days Average basket values from restaurant j in source k in the past X days
x_{ijk}	(Normalized) Impression/view/order count/ratio from consumer i, restaurant j, source k in the past X days Average basket values from consumer i, restaurant j, source k in past X days
z	Source k (name of the carousel or "single restaurant") Vertical position of the recommendation City, geolocation, language, device Temporal features including day of week, local hour of day, meal period

Table 7 List of features for the machine learning models in the MO-step. X=7,14,30,60,120.

leads to an order, and 0 otherwise. In other words, $O_{j1},...,O_{jN_j} \overset{i.i.d.}{\sim} \text{Bernoulli}(c_j)$. The conjugate prior for Bernoulli distribution is the beta distribution with parameters (α_j, β_j):

$$p(c_j) = \frac{1}{B(\alpha_j,\beta_j)} c_j^{\alpha-1}(1-c_j)^{\beta-1}, \tag{17}$$

where $B(\alpha_j,\beta_j)$ is the Beta function acting as a normalizing constant. The likelihood for $O_{j1},...,O_{jN_j}$ is

$$p(\{O_{jm}\}_{m=1}^{N_j}|c_j) = \binom{N_j}{N_j^1} c_j^{N_j^1}(1-c_j)^{N_j - N_j^1}, \tag{18}$$

where $N_j^1 = \sum_{l=1}^{N_j} O_{jm}$ is the number of orders (i.e. impressions that lead to orders) on restaurant j, and $\binom{N_j}{N_j^1}$ is the combination number. Multiplying Eq.(17) and (18), we get the posterior for c_j as

$$\begin{aligned} p(c_j|\{O_{jm}\}_{m=1}^{N_j}) &= \frac{p(\{O_{jm}\}_{m=1}^{N_j}|c_j)p(c_j)}{\int_{c_j=0}^{1} p(\{O_{jm}\}_{m=1}^{N_j}|c_j)p(c_j)dc_j} \\ &= \frac{1}{B(\alpha_j + N_j^1, \beta_j + N_j - N_j^1)} c_j^{\alpha+N_j^1-1}(1-c_j)^{\beta+N_j-N_j^1-1}, \end{aligned} \tag{19}$$

which is another Beta distribution with parameters $(\alpha_j + N_j^1, \beta_j + N_j - N_j^1)$.

Plugging in the formula for the variance of Beta distribution, we get the posterior variance for c_j as

$$\sigma(j)^2 = \frac{(\alpha_j + N_j^1)(\beta_j + N_j - N_j^1)}{(\alpha_j + \beta_j + N_j)^2(\alpha_j + \beta_j + N_j + 1)} \tag{20}$$

which concludes the proof. □

B.1.4. Choice of prior parameters for the marketplace fairness objective. To reduce the number of parameters, we let $\alpha_j = \alpha, \beta_j = \beta, \forall j$, that is, all restaurants follow the same prior distribution for its conversion rate. This is a reasonable assumption to ensure fairness across all restaurants (i.e. no prior bias for any of the restaurants). There are three considerations for picking the values for α and β for the prior distribution. First, the prior mean should not be too far from the actual point estimate for the conversion rate, which is around 2% in our training data. Second, it is preferable to have the posterior relatively stable and robust to bot attacks such as a huge amount of fake view and orders from a new restaurant. Third, the posterior variance in Eq.(19) should be able to differentiate new restaurants with few impressions and orders from the well-established restaurants. The first condition implies the mean of a Beta distribution $\mathcal{B}(\alpha, \beta)$, $\frac{\alpha}{\alpha+\beta}$ should be close to 2%. The second condition implies that α and β should be large enough to guard the posterior against noisy data, while the third condition implies that α and β should be small enough so that the numerator and denominator in Eq.(19) is not dominated by them. Given these considerations, we set $\alpha = 2$ and $\beta = 98$ and find them to work well empirically.

B.2. R-Step

B.2.1. Proof of the concavity of the Pareto frontier.

Proof of the concavity of the Pareto frontier. We prove the case for the trade-off between the conversion objective $C(\boldsymbol{x})$ and the basket value objective $B(\boldsymbol{x})$. The cases for other objectives readily follow.

Given B^* is a fixed constant independent of $\boldsymbol{x}$, we let $\lambda := \alpha_b B^*$ and rewrite the optimization problem as

$$\begin{aligned} &\max_{\boldsymbol{x} \in \mathcal{F}} C(\boldsymbol{x}) \\ &\text{s.t. } B(\boldsymbol{x}) \geq \lambda \end{aligned} \tag{21}$$

where $\mathcal{F} = \{\boldsymbol{x} \in \mathcal{E} : R(\boldsymbol{x}) \geq \alpha_r R^*,\ F(\boldsymbol{x}) \geq \alpha_f F^*\}$ is the feasible region for $\boldsymbol{x}$. Let $\boldsymbol{x}_\lambda^*$ be the solution to Eq.(21) which is a function of λ. We would like to show that $C(\boldsymbol{x}_\lambda^*)$ is concave in $B(\boldsymbol{x}_\lambda^*)$ as a function of λ. We decompose the proof into two steps, which are the two claims below.

Claim 1: $z(\lambda) := C(\boldsymbol{x}_\lambda^*)$ is a concave function of λ.

Proof. Define the Lagrangian function as

$$L_\lambda(\boldsymbol{x}, \mu) = C(\boldsymbol{x}) + \mu(B(\boldsymbol{x}) - \lambda), \quad \mu \geq 0. \tag{22}$$

Therefore the dual problem for Eq.(21) can be written as

$$\begin{aligned} D_\lambda(\mu) &= \max_{\boldsymbol{x}\in\mathcal{F}} L_\lambda(\boldsymbol{x},\mu) = -\mu\lambda + \max_{\boldsymbol{x}\in\mathcal{F}} (C(\boldsymbol{x}) + \mu B(\boldsymbol{x})) \\ &:= -\mu\lambda + \kappa(\mu), \end{aligned} \tag{23}$$

where $\kappa(\mu) := \max_{\boldsymbol{x}\in\mathcal{F}} (C(\boldsymbol{x}) + \mu B(\boldsymbol{x}))$. Because Eq.(21) is a feasible linear optimization problem, strong duality holds, i.e.

$$z(\lambda) = \max_{\boldsymbol{x}\in\mathcal{G}_\lambda} C(\boldsymbol{x}) = \min_{\mu\geq 0} D_\lambda(\mu), \tag{24}$$

where $\mathcal{G}_\lambda = \{\boldsymbol{x} \in \mathcal{F} : B(\boldsymbol{x}) \geq \lambda\}$ is the feasible region for Eq.(21). For any positive λ_1, λ_2 and $t \in [0,1]$, we have

$$\begin{aligned} z(t\lambda_1 + (1-t)\lambda_2) &= \min_{\mu\geq 0} (-\mu(t\lambda_1 + (1-t)\lambda_2) + \kappa(\mu)) \\ &\geq t \min_{\mu\geq 0}(-\mu\lambda_1 + \kappa(\mu)) + (1-t)\min_{\mu\geq 0}(-\mu\lambda_2 + \kappa(\mu)) \\ &= tD_{\lambda_1}(\mu) + (1-t)D_{\lambda_2}(\mu) \\ &= tz(\lambda_1) + (1-t)z(\lambda_2), \ \forall t \in [0,1]. \end{aligned} \tag{25}$$

Therefore by definition of concavity, $z(\lambda)$ is concave in λ.

Claim 2: $B(\boldsymbol{x}^*_\lambda)$ is a piecewise linear function of λ. Specifically, $B(\boldsymbol{x}^*_\lambda) = \lambda_0$ for $\lambda \leq \lambda_0$, $B(\boldsymbol{x}^*_\lambda) = \lambda$ for $\lambda > \lambda_0$.

Proof. Let

$$\boldsymbol{x}_0 = \arg\max_{\boldsymbol{x}\in\mathcal{F}} C(\boldsymbol{x}) \tag{26}$$

be the solution to a modified version of Eq.(21) that relaxes the feasible region from $\mathcal{G}$ to $\mathcal{F}$. If there is more than one solution to Eq.(26), pick x_0 to be the one such that $B(\boldsymbol{x})$ is *maximized*. Let $\lambda_0 = B(\boldsymbol{x}_0)$. λ_0 can be bigger or smaller than λ. We discuss the two cases separately below.

If $\lambda \leq \lambda_0$, then x_0 is also the solution to the original optimization problem in Eq.(21). Therefore $B(\boldsymbol{x}^*_\lambda) = \lambda_0$.

Otherwise, if $\lambda > \lambda_0$, next we show that $B(\boldsymbol{x}^*_\lambda) = \lambda$. Because $\mathcal{G} \subseteq \mathcal{F}$ and $\boldsymbol{x}^*_\lambda = \arg\max_{\boldsymbol{x}\in\mathcal{G}} C(\boldsymbol{x})$, we have

$$C(\boldsymbol{x}^*_\lambda) \leq C(\boldsymbol{x}_0). \tag{27}$$

We know that

$$B(\boldsymbol{x}_0) = \lambda_0 < \lambda \leq B(\boldsymbol{x}^*_\lambda). \tag{28}$$

If $C(\boldsymbol{x}^*_\lambda) = C(\boldsymbol{x}_0)$, by Eq.(28) it contradicts with the assumption that $\boldsymbol{x}_0$ is picked among the optimal solutions such that $B(\boldsymbol{x})$ is maximized. Therefore the inequality in Eq.(27) is strict, i.e.

$$C(\boldsymbol{x}^*_\lambda) < C(\boldsymbol{x}_0). \tag{29}$$

Note that if $B(\boldsymbol{x}^*_\lambda) > \lambda$, we have $B(\boldsymbol{x}^*_\lambda) > \lambda > B(\boldsymbol{x}_0)$. By linearity of $B(\cdot)$, we have that there exists a $\boldsymbol{x}' = t'\boldsymbol{x}_0 + (1-t')\boldsymbol{x}^*_\lambda$ such that $t' \in (0,1)$ and $B(\boldsymbol{x}') = \lambda$. Then by linearity of $C(\cdot)$ and Eq.(29), we have

$$\begin{aligned} C(\boldsymbol{x}') &= t'C(\boldsymbol{x}_0) + (1-t')C(\boldsymbol{x}^*_\lambda) \\ &> t'C(\boldsymbol{x}^*_\lambda) + (1-t')C(\boldsymbol{x}^*_\lambda) \\ &= C(\boldsymbol{x}^*_\lambda). \end{aligned} \tag{30}$$

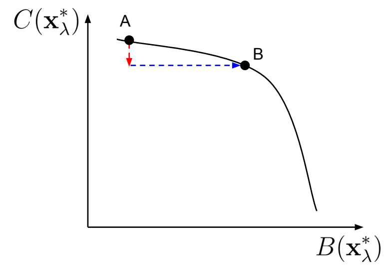


Figure 7 **An illustration of a concave trade-off curve. A small sacrifice in one of the objectives can lead to a big improvement in the other.**

Because the feasible region $\mathcal{G}$ is convex and $\boldsymbol{x}'$ is a linear combination of two points within $\mathcal{G}$, we have $\boldsymbol{x}' \in \mathcal{G}$ but $C(\boldsymbol{x}') > C(\boldsymbol{x}_\lambda^*)$. This contradicts the fact that $\boldsymbol{x}_\lambda^*$ is the optimal solution for Eq.(21). So we must have $B(\boldsymbol{x}_\lambda^*) = \lambda$ for $\lambda > \lambda_0$. So $B(\boldsymbol{x}_\lambda^*)$ is a piecewise linear function in λ.

Finally, combining Claim 1 and 2, we arrive at the conclusion that $C(\boldsymbol{x}_\lambda^*)$ is concave in $B(\boldsymbol{x}_\lambda^*)$. In other words, the trade-off curve between $C(\boldsymbol{x}_\lambda^*)$ and $B(\boldsymbol{x}_\lambda^*)$ with varying λ is a concave curve. The benefit of a concave trade-off curve is illustrated in Fig.7. Comparing with point A on the trade-off curve, point B achieves a big boost in $B(\boldsymbol{x}_\lambda^*)$ with only a small sacrifice in $C(\boldsymbol{x}_\lambda^*)$. $\square$

B.2.2. Formulation and solution for the constrained optimization problem in R-step. We adopt the trick in Agarwal et al. (2012) and add a quadratic penalty term to the objective function in Eq.(12) for an efficient and scalable solution that can be readily served in large-scale online systems. Specifically, we penalize the squared Frobeniums norm between $\boldsymbol{x}$ and a uniform ranking plan $\boldsymbol{u} = \{u_{iq} = \frac{1}{Q}, \forall i, q\}$ that assigns equal probability to all items for all consumers[25]:

$$\begin{aligned} &\max_{\boldsymbol{x} \in \mathcal{E}} \; C(\boldsymbol{x}) - \frac{\gamma}{2} \|\boldsymbol{x} - \boldsymbol{u}\|_F^2 \\ &\text{s.t. } B(\boldsymbol{x}) \geq \alpha_b B^*, \; R(\boldsymbol{x}) \geq \alpha_r R^*, \; F(\boldsymbol{x}) \geq \alpha_f F^*, \end{aligned} \tag{31}$$

Proposition 2 below provides the solutions to Eq.(31). Propsition 3 provides guidance on serving the solution for online systems.

Proposition 2 *The solution to Eq.(31) is*

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq} b_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_f f_{iq} - \mu_i) + \frac{1}{Q}, \tag{32}$$

for any $x_{iq} > 0$. Here $\lambda_b, \lambda_r, \lambda_f$ are the slack variables for the constraints on $B(\boldsymbol{x}), R(\boldsymbol{x})$ and $F(\boldsymbol{x})$ respectively, and are functions of α_b, α_r and α_f. μ_i is the slack variable for the constraint $\sum_q x_{iq} = 1$.

Proof of Proposition 2. First, we write out the element-wise form of Eq.(31):

$$\begin{aligned}
\max_{\{x_{iq}\}\in\mathcal{E}} \ & \sum_{i,q}(x_{iq}c_{iq}-\frac{\gamma}{2}(x_{iq}-\frac{1}{Q})^2) \\
\text{s.t.} \ & \sum_{i,q} x_{iq}c_{iq}b_{iq} \geq \alpha_b B^*, \\
& \sum_{i,q} x_{iq}c_{iq}r_{iq} \geq \alpha_r R^*, \\
& \sum_{i,q} x_{iq}f_{iq} \geq \alpha_f F^*, \\
& x_{iq} \geq 0, \ i=1,...,I, q=1,...,Q, \\
& \sum_{q} x_{iq} = 1, \ i=1,...,I,
\end{aligned} \tag{33}$$

where c_iq, r_iq, b_iq, and f_iq are the values for the consumer conversion objective, consumer retention objective, basket value objective and fairness objective between consumer i and item q, respectively. The objective for the maximization problem in Eq.(33) is concave, the inequality are all affine functions. Therefore, the KKT conditions are *necessary and sufficient* conditions for optimality. We use them to solve Eq.(33).

Let $\lambda_b, \lambda_r, \lambda_f, \delta_{iq}$ and μ_i be the non-negative slack variables for the five sets of constraints in Eq.(33), which are used to define the Lagrangian:

$$\begin{aligned}
L(\{x_{iq}\},\lambda_b,\lambda_r,\lambda_f,\{\delta_{iq}\},\{\mu_i\}) = & \sum_{i,q}(x_{iq}c_{iq}-\frac{\gamma}{2}(x_{iq}-\frac{1}{Q})^2) - \lambda_b(\sum_{i,q} x_{iq}c_{iq}b_{iq} - \alpha_b B^*) \\
& - \lambda_r(\sum_{i,q} x_{iq}c_{iq}r_{iq} - \alpha_r R^*) - \lambda_f(\sum_{i,q} x_{iq}f_{iq} - \alpha_f F^*) \\
& - \delta_{iq}x_{iq} + \mu_i(\sum_{q} x_{iq} - 1).
\end{aligned} \tag{34}$$

By *stationarity* from the KKT conditions, we have

$$-c_{iq} + \gamma(x_{iq}-\frac{1}{Q}) - \lambda_b c_{iq}b_{iq} - \lambda_r c_{iq}r_{iq} - \lambda_f f_{iq} - \delta_{iq} + \mu_i = 0, \tag{35}$$

which yields

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq} + \delta_{iq} - \mu_i) + \frac{1}{Q}. \tag{36}$$

By *complementary slackness* from the KKT conditions, $x_{iq} > 0$ only when $\delta_{iq} = 0$. Therefore

$$x_{iq} = \frac{1}{\gamma}(c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq} - \mu_i) + \frac{1}{Q} \tag{37}$$

for any $x_{iq} > 0$. □

Proposition 3 *Ranking according to x_{iq} in Eq.(32) is equivalent to ranking according to*

$$\tilde{x}_{iq} = c_{iq} + \lambda_b c_{iq}b_{iq} + \lambda_r c_{iq}r_{iq} + \lambda_f f_{iq}. \tag{38}$$

Proof of Proposition 3. When serving the ranking plan $\boldsymbol{x}$ for consumer i, only the relative ordering of x_{iq} matters. Therefore the intercept $\frac{1}{Q}$, the multiplier $\frac{1}{\gamma}$ and μ_i do not have affect the final ranking results. □

We now show that λ_b, λ_r, and λ_f can be solved as functions of α_b, α_r and α_f in addition to the other inputs. By *primal feasibility* from the KKT conditions, we have $\sum_q x_{iq} = 1, \forall i$. Plugging in Eq.(36) and solve for μ_i we have

$$\mu_i = \frac{1}{Q}\sum_q (c_i q + \lambda_b c_{iq} b_{iq} + \lambda_r c_{iq} r_{iq} + \lambda_f f_{iq} + \delta_{iq}),\ \ i = 1, ..., I, \tag{39}$$

Which are I linear equations involving the unknown variables $\mu_i (i = 1, ..., I)$, λ_b, λ_r and λ_f.

By complementary slackness from the KKT conditions, we have

$$\begin{aligned} \lambda_b(\sum_{i,q} x_{iq} c_{iq} b_{iq} - \alpha_b B^*) = 0, \\ \lambda_r(\sum_{i,q} x_{iq} c_{iq} r_{iq} - \alpha_r R^*) = 0, \\ \lambda_f(\sum_{i,q} x_{iq} f_{iq} - \alpha_f F^*) = 0. \end{aligned} \tag{40}$$

The first equation in Eq.(40) implies either $\lambda_b = 0$, or $(\sum_{i,q} x_{iq} c_{iq} b_{iq} - \alpha_b B^*) = 0$, which is another linear equation for $\mu_i (i = 1, ..., I)$, λ_b, λ_r and λ_f after plugging in Eq.(36). Similar observations hold for the other two equations in Eq.(40). Therefore, combining Eq.(39) and Eq.(40), we have a linear system with $I + 3$ unknowns and $I + 3$ equations, which can be solved using any linear equation solver.

In practice, I is the number of consumers, so solving the linear system directly can be expensive. We propose instead of solving λ_b, λ_r and λ_f as a function of α_b, α_r and α_f which are treated as tuning parameters, we propose to treat λ_b, λ_r and λ_f as tuning parameters directly to reduce computation. In addition, λ_b, λ_r and λ_f can also be viewed as the weights controlling the relative importance of the different objectives.

C. Additional Experiment Details

C.1. Latest Production Recommender System at the Company

The latest production recommender system at the company is a framework using three disjoint machine learning (ML) models to rank carousels and single restaurants in the homepage, based on conversion rate as the single objective: (1) A $(consumer, restaurant)$-level model predicting the conversion objective on restaurant level, i.e. the probability that the consumer will order from the restaurant in the current session, which is used to determine the ranking among the single restaurants and within each carousel (**ML Model A**); (2) A $(consumer, carousel)$-level model predicting the conversion objective on carousel level, i.e. the probability that the consumer will order from *any* restaurant inside the carousel in the current session, which is used to determine the ranking among the carousels (**ML Model B**); (3) A $(consumer, number\ of\ carousels)$-level model predicting the conversion rate under different number of carousels recommended, which is used to determine how many carousels to display in the current session (**ML Model C**). Figure 8 shows an overview of the production recommender system.

All of the models are real-time personalized machine learning models, using the state-of-art hybrid recommender systems (Burke 2002) based on gradient boosting decision trees with the features and hyperparameters same as those in Appendix B.1.2. For fair comparison, we adopt the same model architecture and model size for estimating the individual objectives in the MO-step of the MOHR framework for the experiments at the company.

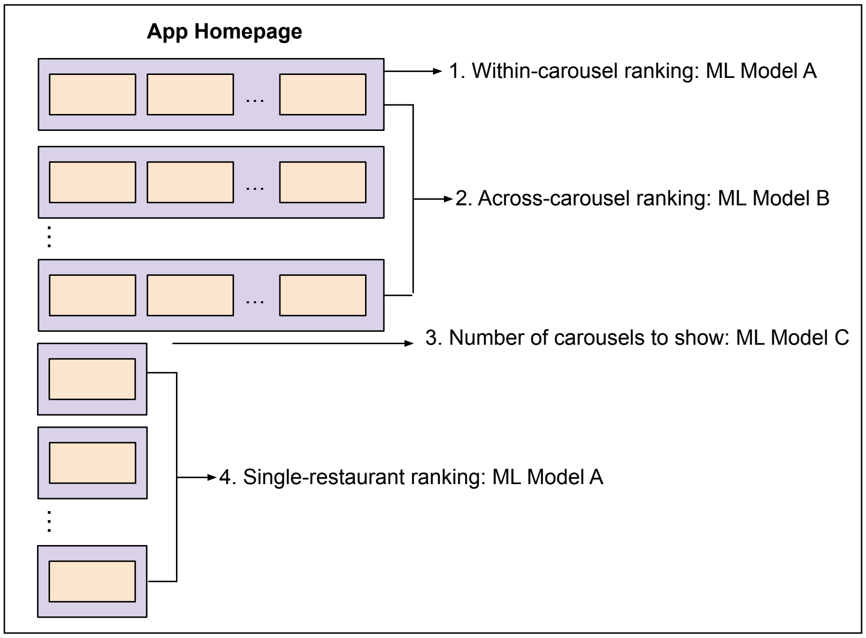


Figure 8 An overview of the latest production recommender system at the company.

Because the framework is unable to generate calibrated ranking scores across carousels and single restaurants, all of the carousels are ranked above all of the single restaurants in the production recommender system.

C.2. Variance Correction For Ratio Metrics with Intra-consumer Correlation

Different sessions from the same consumer during the experiment period could be correlated with each other. To explicitly account for this intra-consumer correlation, we derive the corrected variance calculation for the three ratio metrics in Table 3 in the hypothesis testing procedure. Without loss of generality, we present the derivation for the conversion rate metric below. The derivation for the basket value per order and retention rate readily follows.

Following the notation in Table 3, let

$$\bar{O} = \frac{1}{I}\sum_i \sum_s O_{is}, \;\; \bar{S} = \frac{1}{I}\sum_i S_i \tag{41}$$

be the average number of orders O and average number of sessions S *per consumer*. Therefore, the conversion rate metric $C = \bar{O}/\bar{S}$ is the ratio of the two. We assume that the observations within each consumer could be correlated, but the observations across different consumers are independent. By multivariate central limit theorem, we have

$$\begin{pmatrix}\bar{O}\\ \bar{S}\end{pmatrix} \overset{I\to\infty}{\sim} N\left[\begin{pmatrix}\mu_O\\ \mu_S\end{pmatrix}, \begin{pmatrix}\sigma_O^2/I & Cov(O,S)/I\\ Cov(O,S)/I & \sigma_S^2/I\end{pmatrix}\right] \tag{42}$$

where μ_O and σ_O^2 are the mean and variance of the random variable O (number of orders from each consumer), μ_S and σ_S^2 are the variance of the random variable S (number of sessions from each consumer), and $Cov(O,S)$ is the covariance between O and S. By multivariate delta method, we have the conversion rate

$$C = \bar{O}/\bar{S} \sim N(\mu_O/\mu_S, \sigma_C^2), \tag{43}$$

where

$$\begin{aligned}
\sigma_C^2 &= Var(\frac{\bar{O}}{\bar{S}}) \\
&= \begin{pmatrix} \frac{\partial}{\partial \bar{O}}(\frac{\bar{O}}{\bar{S}}) & \frac{\partial}{\partial \bar{S}}(\frac{\bar{O}}{\bar{S}}) \end{pmatrix} \begin{pmatrix} \sigma_O^2/I & Cov(O,S)/I \\ Cov(O,S)/I & \sigma_S^2/I \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \bar{O}}(\frac{\bar{O}}{\bar{S}}) \\ \frac{\partial}{\partial \bar{S}}(\frac{\bar{O}}{\bar{S}}) \end{pmatrix} \\
&= \begin{pmatrix} \frac{1}{\bar{S}} & -\frac{\bar{O}}{\bar{S}^2} \end{pmatrix} \begin{pmatrix} \sigma_O^2/I & Cov(O,S)/I \\ Cov(O,S)/I & \sigma_S^2/I \end{pmatrix} \begin{pmatrix} \frac{1}{\bar{S}} \\ -\frac{\bar{O}}{\bar{S}^2} \end{pmatrix} \\
&= \frac{1}{I}\left[\frac{\sigma_O^2}{\bar{S}} + \frac{\bar{O}^2}{\bar{S}^4}\sigma_S^2 - \frac{2\bar{O}}{\bar{S}^3}Cov(O,S)\right].
\end{aligned} \tag{44}$$

When computing the p-values for C, σ_O^2, σ_S^2 and $Cov(O,S)$ can be plugged in as the sample variance and covariance estimated from the data. Generally speaking, the estimated variance is *larger* when considering the intra-consumer correlation compared with treating all sessions to be i.i.d.. So the variance correction in Eq.(44) yields a p-value that's larger than if treating all sessions as i.i.d., making the hypothesis testing more rigorous and conservative.

C.3. Results on the MO-Step

Table 8 summarizes the model performance and top important features for the machine learning-based objectives, namely consumer conversion, consumer retention and basket value.

C.4. Results on the H-step

Table 9 and Fig.9 presents the estimated scrolling factors in the experiment. Note that there are at most 6 restaurants presented in every carousel. To see more restaurants within the carousel, there is a "see all" button at the top right corner of every carousel. The H-step is only applied to the top 6 positions within each carousel.

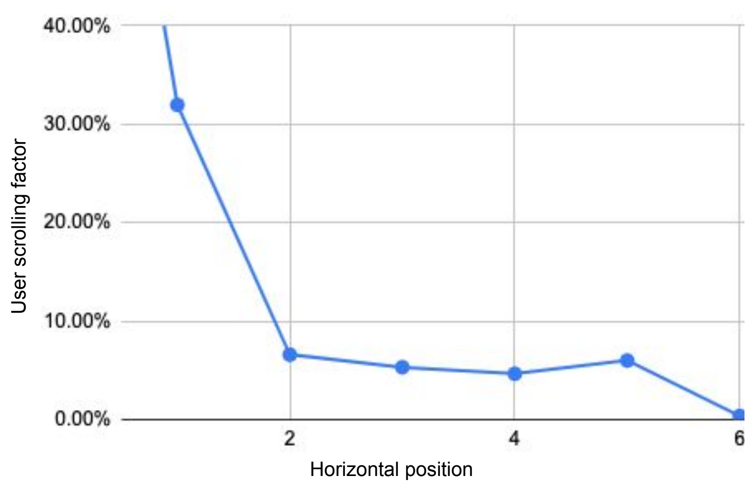


Figure 9 **Scrolling factors from the consumer browsing model.**

Model name	**Model performance**	**Top 10 important features**
Consumer conversion	Test AUC = 0.8797	Normalized (consumer, restaurant) order count
		Local hour of day
		Consumer view count
		Normalized (consumer, restaurant) impression count
		Normalized (consumer, restaurant) click count
		$u_i^T v_j$, i.e. dot product of consumer embedding and restaurant embedding
		Consumer order-to-impression ratio
		Restaurant delivery time
		Meal period
		(restaurant, source) order-to-impression ratio
Consumer retention	Test AUC = 0.7847	Consumer order counts in the past 120 days
		Restaurant average basket value
		Consumer order counts in the past 14 days
		Consumer order counts in the past 7 days
		Delivery radius
		Consumer ride count
		City
		% of consumers churned after ordering from restaurant j in past 60 days
		% of consumers churned after ordering from restaurant j in past 30 days
		% of consumers churned after ordering from restaurant j in past 120 days
Basket value	Test rMSE = 0.1135	(consumer, restaurant) average basket value in the past 120 days
		Consumer average basket value in the past 120 days
		$u_i^T v_j$, i.e. dot product of consumer embedding and restaurant embedding
		Local hour of day
		(restaurant, source) average basket value in the past 120 days
		Source name
		Restaurant average basket value in the past 120 days
		$\frac{cos(u_i,v_j)}{\|\|u_i\|\|\|\|v_j\|\|}$, i.e. cosine similarity between consumer and restaurant
		(consumer, restaurant) average basket value in the past 60 days
		Consumer average basket value in the past 60 days

Table 8 **List of features for the machine learning models in the MO-step.**

Horizontal position	**Consumer scrolling factor $\hat{p}_{l,l+1}$**
0	100.00%
1	31.96%
2	6.58%
3	5.32%
4	4.65%
5	5.99%
6	0.36%

Table 9 **Estimated values for the scrolling factors from the consumer browsing model.**

C.5. Randomization Check

We compute the p-values for the metric differences between treatment and control group 28 days *before* the experiment start date, when both treatment and control consumers are expected to receive recommendations generated by the same algorithm. Table 10 shows that the A/A testing p-values are all greater than 0.05 (or 0.10 depending on the significance level of choice), suggesting that there is no significant difference in the treatment and control group in terms of the key business metrics, before the experiment start date.

Metric	Conversion rate	Basket value per order	Retention rate	Orders per consumer	Search rate
A/A testing p-value	0.326	0.452	0.947	0.853	0.286

Table 10 p-values for the A/A testing on key business metrics.

We further collected a comprehensive set of 472 metrics capturing various aspects of consumer behavior on the platform and across different surfaces, and computed the p-values for the 472 metric differences. Under the null hypothesis that treatment and control group consumers are not statistically different, the p-values should follow a uniform distribution. We conduct the Kolmogorov-Smirnov (KS) test on the empirical distribution of those 472 p-values, and could not reject the null that they follow a uniform distribution on [0,1] (Fig.10), suggesting that our randomization holds true.

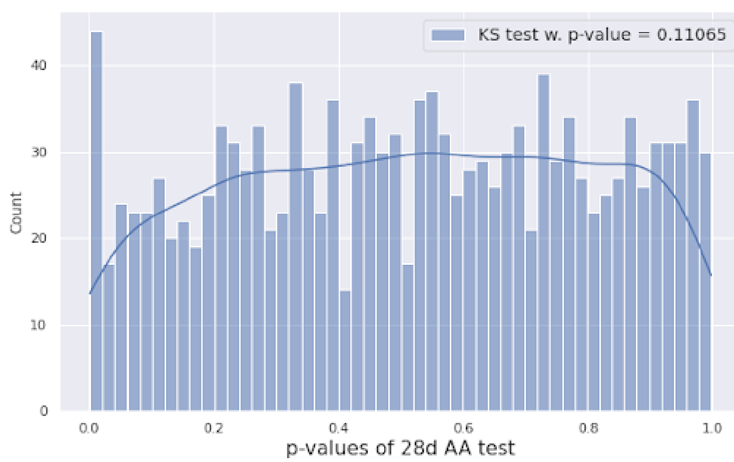


Figure 10 Histogram of the p-values for the 472 metric differences for A/A test. Kolmogorov-Smirnov (KS) test which compares the empirical distribution of the p-values against the uniform distribution on [0,1] has p-value of 0.11, which fails to reject the null hypotheses that these metrics are not statistically significantly different during the A/A testing period.