# A Predict-Then-Optimize Customer Allocation Framework for Online Fund Recommendation[*]

Xing Tang[1], Yunpeng Weng[1], Fuyuan Lyu[2], Dugang Liu[4], and
Xiuqiang He ✉[3]

FiT, Tencent, Shenzhen, China
School of Computer Science, McGill University, Montreal, Canada
School of Big Data and Internet,Shenzhen Technology University, Shenzhen, China
Shenzhen University, Shenzhen, China
{shawntang,edwinweng}@tencent.com
fuyuan.lyu@mail.mcgill.ca; {dugang.ldg,he.xiuqiang}@gmail.com

**Abstract.** With the rapid growth of online investment platforms, funds can be distributed to individual customers online. The central issue is to match funds with potential customers under constraints. Most mainstream platforms adopt the recommendation formulation to tackle the problem. However, the traditional recommendation regime has its inherent drawbacks when applying the fund-matching problem with multiple constraints. In this paper, we model the fund matching under the allocation formulation. We design **PTOFA**, a **P**redict-**T**hen-**O**ptimize **F**und **A**llocation framework. This data-driven framework consists of two stages, i.e., prediction and optimization, which aim to predict expected revenue based on customer behavior and optimize the impression allocation to achieve the maximum revenue under the necessary constraints, respectively. Extensive experiments on real-world datasets from an industrial online investment platform validate the effectiveness and efficiency of our solution. Additionally, the online A/B tests demonstrate PTOFA's effectiveness in the real-world fund recommendation scenario.

**Keywords:** Fund Matching, Predict-then-Optimize, Allocation

## 1 Introduction

In recent years, selling funds via online platforms, such as Ant Fortune[1], Tencent LiCaiTong[2] and Robinhood[3], has gained immense popularity due to their convenience of financial services. The core problem is achieving an optimal match between funds and customers, thereby fulfilling customers' interests and fund managers' requirements. Based on extensive investment records, these online financial platforms adopt data-driven approaches to improve their business and service quality as other online services [6, 15, 19, 24].

---

[*] The first three authors contribute equally.
[1] https://www.antgroup.com/en/
[2] https://www.tencentwm.com
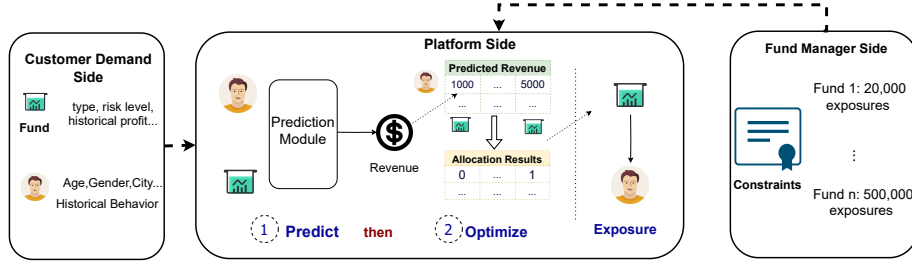[3] https://robinhood.com

Fig. 1: An illustration of fund matching process on an online investment platform.

The matching problem between funds and customers on the platform is complex. As shown in Fig. 1, the platform must consider the requirements of three stakeholders: customers' interests, fund managers' requirements, and platforms' risk regulation. To the end of customers, clicks and conversions don't fully reveal their interest in funds. Platforms should better predict customer revenue to gauge potential interest, as they tend to invest more in funds satisfying their needs. To the end of fund managers, funds should be exposed to a certain amount of customers to guarantee their interest. To the end of platforms, platforms must match risk levels between funds and customers, ensuring each fund only targets customers with higher risk tolerance. Under these constraints, platforms intend to maximize the revenue between customers and fund managers.

One possible solution is to formulate the problem as a top-$k$ recommendation [9, 8]. The system will predict the relevance score measuring the customer-fund pair based on the customer's preference and the fund's properties. Top-$k$ funds will then be exposed to customers. Such a formulation aims to increase the conversion rate and the total number of transactions. However, we observe that this formulation may cause several problems: (i) *Matthew effect* [5]. Exposure to customers is mainly occupied by a few top funds, leaving others few opportunities to be exposed. Meanwhile, some high-quality funds fail to be exposed to enough customers. Correspondingly, the majority of revenue also goes to these top funds. Such a phenomenon could harm the platform in the long run. (ii) *Objectives mismatch*. The traditional recommender system aims to maximize the conversion of customers to items. We argue that this may result in a sub-optimal total revenue since they treat every conversion equally. However, in the fund matching problem, the revenue of the same customer may differ significantly on various funds. Both problems of the recommendation formulation motivate us to solve the fund matching problem by adopting an allocation [22, 27] formulation.

Instead of recommending suitable funds to each customer from a local perspective, we formulate the fund-matching problem as an allocation format, which can better encode constraints and maximize total revenue from a global perspective. We introduce the predict-then-optimize framework [4, 27, 25] to allocate funds to each customer, as illustrated in Fig. 1. However, we still have to face the following main challenges. *How to predict expected revenue accurately?* No-

tice that the distribution of expected revenue is long-tailed because both the investment intention and available capital vary greatly regarding customer-fund pairs. Despite that mean squared error (MSE) has become the de facto loss for regression tasks, it will underestimate on imbalanced label [16]. Moreover, *How to effectively allocate funds to customers under constraints?* In practical industrial applications, the delivery systems depend on manual operations [11] or reinforcement learning [27]. As to manual operations, the allocation strategy determines the priority of $K$ products according to historical sales and operation specialists' experience. After that, it selects enough customers for each product to deliver by the order of priority. Therefore, the manual allocation strategy relies highly on human experience, which makes them sub-optimal. As to reinforcement learning, it is too complex to deploy in our scenario, which requires the model to converge to the optimal state quickly due to the low maintenance costs of funds. Also, the solution is hard to be easily extended to large-scale settings.

In this paper, we design a two-stage Predict-Then-Optimize Fund Allocation (PTOFA) framework. During the first stage, PTOFA predicts the revenue brought by customer investments for each fund. In the second stage, PTOFA allocates the funds with a solver according to the predicted revenue in the first stage. Specifically, we first introduce the entire sample space for customer intention and expected revenue prediction to predict the revenue accurately. Then, we adopt a counterfactual multi-task learning module to predict the expected revenue in the entire sample space. Moreover, our framework takes all potential customer's predicted revenues for each fund as coefficients and employs an efficient optimization module to get the near-optimal allocation solution. In summary, the main contributions of our work are as follows:

- We highlight the inherent drawbacks of adopting recommendation formulation for the fund-matching problem. Instead, we propose to formulate it as the allocation problem. A predict-then-optimize fund allocation framework, namely PTOFA, is designed to match funds and customers on investment platforms under various constraints.
- In PTOFA, we propose a counterfactual entire space multi-task module to predict the expected revenue. Note that it is the first attempt to predict the expected revenue of funds. We then design an efficient optimization algorithm based on the predicted revenue to get the solution.
- Both offline and online experiments are conducted on LiCaiTong, one of the largest online investment platforms in China. Experimental results demonstrate the effectiveness and efficiency of our proposed PTOFA framework.

## 2 Formulation of Fund Allocation

As we discussed in earlier sections, the core idea of fund allocation is to efficiently allocate various funds $f \in \mathbf{F}$ to different customers $u \in \mathbf{U}$ given customers' preferences and various constraints. We adopt a matrix $\mathbf{X} \in \{0,1\}^{|\mathbf{U}| \times |\mathbf{F}|}$ to

denote the allocation result, with $\mathbf{X}_{u,f} = 1$ indicating fund $f$ is allocated to customer $u$ and vice versa. The goal of the fund allocation problem is to maximize the total expected revenue income, and the objective function is formulated as follows:

$$Obj = \sum_{u \in \mathbf{U}, f \in \mathbf{F}} \mathbf{X}_{u,f} \cdot \mathbb{E}_{u,f}, \tag{1}$$

where $\mathbb{E}_{u,f}$ is the revenue expectation of customer $u$ given fund $f$. The revenue expectation is predicted given the profile and historical record of customer $u$, as well as various properties of the fund $f$.

Such an allocation process is also subject to certain constraints. Similar to recommendation systems, only limited slots are available for exposing funds to customers. This can be defined as:

**Constraint 1 (Customer Top-K)** *Each customer $u$ is only exposed to $K$ recommended funds:* $\sum_{f \in \mathbf{F}} \mathbf{X}_{u,f} = K, \ \forall u \in \mathbf{U}$.

Considering the long-term interest, the impression of customers should be distributed fairly among funds. This is defined as:

**Constraint 2 (Fund Exposure)** *Each fund should be allocated to exactly $d_f$ customers, formulated as:* $\sum_{u \in \mathbf{U}} \mathbf{X}_{u,f} = d_f, \ \forall f \in \mathbf{F}$.

Finally, exposure to funds needs to be regulated due to the risky nature of the investment. Only funds with risk levels lower than customers' explicitly self-declared risk tolerance can be recommended to specific customers. This is defined as:

**Constraint 3 (Platform Risk)** $\mathbf{X}_{u,f}$ *can only be 1 if the customer risk tolerance $t_u$ is no less than the risk level $r_f$.* $\mathbf{X}_{u,f} = 1 \ \rightarrow \ t_u \geq r_f, \forall u \in \mathbf{U}, \forall f \in \mathbf{F}$.

Hence, the goal of fund allocation is to maximize Eq. 1 under the above-mentioned constraints. However, both the $\mathbb{E}_{u,f}$ and $\mathbf{X}_{u,f}$ are not static during the process in Eq. 1. An inaccurate prediction of the revenue expectation could mislead the fund allocation process. To solve this problem, we adopt a commonly used predict-then-optimize framework as a solution. As its name explains, it consists of two tasks: prediction and optimization.

**Prediction.** Accurately predicting the expected revenue requires the model to address two main issues: i) predict whether the customer will invest the delivered fund, and ii) if the conversion occurs, predict how much money the customer is willing to invest. Thus, we formulate the expected revenue as

$$\mathbb{E} = \mathbf{P}(\mathrm{C}) \times \mathbb{E}(R|C), \tag{2}$$

where $C \in \{0,1\}$ denotes whether the customer is converted, $R$ denotes the revenue brought by the customer given the conversion, and $\mathbb{E}(R|C)$ is its expectation given that the customer converts.
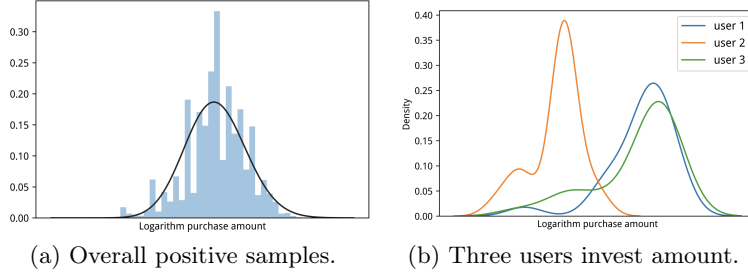
(a) Overall positive samples.          (b) Three users invest amount.

Fig. 2: Distribution of logarithmic transaction value.

**Optimization.** After the prediction stage, for each customer $u$ and fund $f$, the $\mathbb{E}_{u,f}$ is regarded as the constant. The fund allocation can be viewed as a combinatorial optimization problem:

$$\arg\max_{\mathbf{X}} \sum_{u \in \mathbf{U}, f \in \mathbf{F}} \mathbf{X}_{u,f} \cdot \mathbb{E}_{u,f},$$

$$\text{s.t. Constraint 1, 2, \& 3.}$$

(3)

## 3    Methodology

**Revenue Prediction Over the Entire Space.** In the prediction stage, the prediction module needs to predict the conversion probability and the revenue brought by this conversion according to Eq. 2 simultaneously. However, here are some unique challenges in the prediction task. First, imbalanced labels exist among converted customers. Specifically, customer revenue follows a long-tail distribution in real-world scenarios. Fig. 2a shows the distribution of the logarithmic values of the revenue (purchase amount) brought by converted customers. Instead of a conventional regressor optimized by MSE loss, we adopt a parametric manner to model revenue distribution motivated by previous work [21, 26]. Meanwhile, we also randomly present the historical investment amounts of three customers in Fig. 2b, which follow log-normal distributions but with different parameters. Hence, the parameters of the distribution, i.e. the mean $\mu$ and the standard deviation $\sigma$, are estimated in the neural network with customer features and fund features. We can obtain the probability density function of converted samples' revenue $\mathbf{P}(R|C = 1, \mu, \sigma) = 1/(R\sqrt{2\pi}\sigma) \cdot \exp(-(\log R - \mu)^2/2\sigma^2)$.

Second, previous works [21, 29] train multi-task learning model with conversion rate prediction task training over the entire sample space $\mathcal{D}$, while the regression task for revenue estimation is only trained over the positive sample space $\mathcal{S}^+$, as shown in Fig. 3a. This will lead to the **sample selection bias** [13, 23, 31] between training and inference, as the inference space is the entire sample space. Therefore, we utilize a counterfactual mechanism to debias the sample selection bias. As shown in Fig. 3, we utilize the do-calculus[31] $do(C = 1)$ to provide the counterfactual sample for regression task training over negative sample space $\mathcal{S}^-$. The $do$ indicates that we suppose the unconverted sample is converted

(a) Training the regression task over positive sample space.

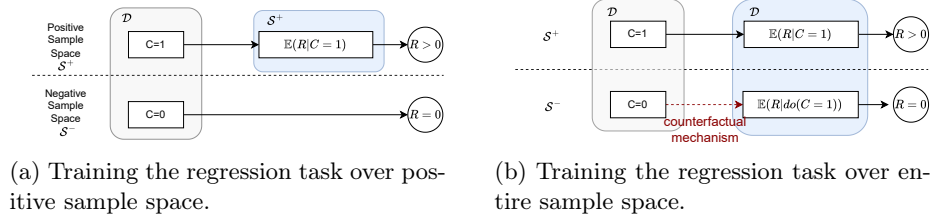(b) Training the regression task over entire sample space.

Fig. 3: The existing solution and our idea.

and predicts its expected revenue. Thus, we can train the prediction model in the entire space. To implement training in the counterfactual space, we need to set counterfactual labels for negative samples, i.e., the revenue $R$ if they convert. Since most customers have $R = 0$ after exposure and the counterfactual samples are all from negative sample space, we assume that the counterfactual revenue is a small value $\epsilon$.

Furthermore, we can formulate the probability of an observed positive sample with a customer-fund pair $(u, f)$:

$$
\begin{aligned}
\mathbf{P}(R|\boldsymbol{x}_u, \boldsymbol{x}_f) &= \mathbf{P}(C = 1|\boldsymbol{x}_u, \boldsymbol{x}_f) \times \mathbf{P}(R|\boldsymbol{x}_u, \boldsymbol{x}_f, C = 1) \\
&= \frac{\mathrm{P}_c}{R \cdot \sqrt{2\pi}\sigma} \exp(-\frac{(\log(R) - \mu)^2}{2\sigma^2}),
\end{aligned}
\tag{4}
$$

where $P_c = \mathbf{P}(C = 1|\boldsymbol{x}_u, \boldsymbol{x}_f)$ denotes the predicted conversion rate. The probability of an observed negative sample $(R = 0)$ can be formulated as:

$$
\begin{aligned}
\mathbf{P}(R = 0|\boldsymbol{x}_u, \boldsymbol{x}_f) &= 1 - P_c + P_c \cdot \mathbf{P}(\epsilon|\boldsymbol{x}_u, \boldsymbol{x}_f, C = 1) \\
&= (1 - P_c) + \frac{P_c}{\epsilon \cdot \sqrt{2\pi}\sigma} \exp(-\frac{(\log(\epsilon) - \mu)^2}{2\sigma^2}),
\end{aligned}
\tag{5}
$$

We can thereby derive the following form of the entire space multi-task joint loss function from the negative log-likelihood:

$$
\begin{aligned}
\mathcal{L} = -\frac{1}{M} &\left( \sum_{(u,f)\in\mathbf{Y}_{\mathrm{pos}}} \left( \log P_c + \log \frac{1}{R \cdot \sqrt{2\pi}\sigma} - \frac{(\log(R) - \mu)^2}{2\sigma^2} \right) \right. \\
&+ \left. \sum_{(u,f)\in\mathbf{Y}_{\mathrm{neg}}} \left( \log(1 - P_c + \frac{P_c}{\epsilon \cdot \sqrt{2\pi}\sigma} \exp(-\frac{(\log(\epsilon) - \mu)^2}{2\sigma^2}))) \right) \right).
\end{aligned}
\tag{6}
$$

Here $\mathbf{Y}_{\mathrm{neg}} \equiv \{(u, f)|R = 0\}$ and $\mathbf{Y}_{\mathrm{pos}} \equiv \{(u, f)|R > 0\}$ denote the negative and positive sets respectively. Throughout the proposed loss, the model has a joint probability modeling of the classification and regression tasks on the entire sample space. With the predicted $P_c$, parameters of the log-normal distribution $\mu$ and $\sigma$, we can estimated the expected revenue for each $< u, f >$ pair: $\mathbb{E} = P_c \times (\exp(\mu + \sigma^2/2))$.

---

**Algorithm 1:** Optimization Algorithm *HA* of PTOFA

---

**Input:** fund set $\mathbf{F}$; matrix:$\{\mathbb{E}_{uf}|\forall u \in \mathbf{U}, \forall f \in \mathbf{F}\}$; customer risk tolerance: T
$= \{t_u|u \in \mathbf{U}\}$; fund risk level:R $=\{r_f\}$; exposure demend: D $= \{d_f\}$;
K in top-K

**Output: X**

**1** For all $< u, f >$ pair, if $r_f > t_u$, $\mathbb{E}_{u,f} = -\infty$;
**2** $\forall f \in F$, $\alpha_f = \sum_{u \in \mathbf{U}} \mathbb{E}_{uf}/d_f$
**3 for** $u \in \mathbf{U}$ **do**
**4** $\quad$ Sort $\forall f \in \mathbf{F}$ in descending order based on $\mathbb{E}_{uf}$; Compute the heuristic
$\quad\quad$ score $h_u$ according to Eq. 7.
**5 end**
**6** Sort $\forall u \in \mathbf{U}$ in descending order based on $\{h_u\}$;
**7 while** $u \in |\mathbf{U}|$ **do**
**8** $\quad$ Select the top-K funds $f$ with highest $\mathbb{E}_{u,f}$ in terms of user $u$, $\mathbf{X}_{uf} = 1$
$\quad\quad$ ,$\mathbf{U} = \mathbf{U}\backslash u$ , $d_f = d_f - 1$ ;
**9** $\quad$ **if** $d_f = 0$ **then**
**10** $\quad\quad$ $\forall u, \mathbb{E}_{u,f} = -\infty, \mathbf{F} = \mathbf{F}\backslash f$ ; Execute from line 2-6;
**11** $\quad$ **end**
**12 end**

---

**Large Scale Fund Allocation Module**. The optimization stage solves Eq. 3 with $\mathbb{E}_{u,f}$ obtained in the prediction stage as constants. We proposed a specially designed heuristic algorithm (HA) in Algorithm 1 to tackle this problem under complex constraints. Specifically, a heuristic score is introduced to measure the potential loss of platform revenue. For the customer $u$, we sort all funds in descending order of coefficient $\mathbb{E}$ according to the revenue matrix, getting the candidate fund list with ranking $\{f_1, f_2, ..., f_{|F|}\}$ for the specific user and respectively calculate the predicted revenue difference between two adjacent funds. The final heuristic score is defined as follows:

$$h_u = \sum_{j \in \{2,3,...,|F|\}} \alpha_{f_{j-1}} \times (\mathbb{E}_{u,f_{j-1}} - \mathbb{E}_{u,f_j}), \tag{7}$$

where $\alpha_{f_i} = \sum_{u \in \mathbf{U}} \mathbb{E}_{u,f_i}/d_{f_i}$ is introduced to estimate the consumption speed of the fund. A large $\alpha_f$ indicates that the fund $f$ may reach the exposure limit sooner. We set $\mathbb{E}_{u,f} = -\infty$ if the risk tolerance level of customer $u$ is lower than the risk level of fund $f$ to satisfy the *Platform Risk Constraint*.

Our heuristic algorithm (HA) in Algorithm 1 begins by allocating the allocated top-K funds $f$ for the user $u$ with the highest heuristic score and subsequently updates $d_f = d_f - 1$ for these top-K funds. The allocation process then proceeds to the user with the second-highest score and so forth. Once a specific fund $f$ achieves allocation limited under the *Fund Exposure Constraint*, we set $\mathbb{E}_{u,f} = -\infty$ for all remaining customers, ensuring that $f$ will not be allocated further. This process is repeated until all funds have been completely allocated.

Table 1: Comparison of prediction models on Two real-world datasets.

| Model | LCT-A | | | LCT-B | | |
|---|---|---|---|---|---|---|
| | AUC↑ | MSE↓ | MAE↓ | AUC↑ | MSE↓ | MAE↓ |
| ESMM | 0.8303 | - | - | 0.9270 | - | - |
| DNN-$C$ | 0.8237 | - | - | 0.9040 | - | - |
| DNN-$R$ | - | 0.0386 | 0.0413 | - | 0.4248 | 0.1369 |
| Shared-Bottom | 0.8120 | 0.0367 | 0.0323 | 0.9208 | 0.4265 | 0.1404 |
| MMoE | 0.8205 | 0.0381 | 0.0392 | 0.9192 | 0.4236 | 0.1436 |
| ZILN | 0.8287 | 0.0386 | 0.0426 | 0.9228 | 1.7885 | 0.5909 |
| MTL-MSE | 0.8276 | 0.0382 | 0.0353 | 0.9213 | 0.4262 | 0.1446 |
| **Ours** | **0.8327** | **0.0362** | **0.0227** | **0.9313** | **0.4203** | **0.1326** |

## 4    Experiments

In this section, we conduct thorough experiments on the proposed method's effectiveness and efficiency. Because our method is a two-stage framework, predict-then-optimize, we compare some baselines on real-world offline datasets to verify that our prediction module is effective. Moreover, we compare different allocation optimizing strategies to show the superiority of the optimization module with the prediction module being determined. Finally, online experiments regarding the combination of various components are conducted.

### 4.1    Offline Experiments

**Prediction Module Evaluation**: We use two real-world datasets from a large-scale platform to compare revenue prediction performance: **LCT-A** contains training(90%) and validation (10%) samples from historical promotional campaigns of three months, involving 8 promotional funds and 4.75 million users. The test set has 3.45 million users sampled from the promotional delivery of the subsequent week. **LCT-B** contains training (90%) and validation (10%) samples from recommendation scenario, involving 4 fund types and 4.3 million users. The testing set includes 0.69 million users.

We compare our approach with several widely used methods: Shared-Bottom ("Shared" for short) [2], MMoE [12], ESMM [13], ZILN [21]. Additionally, two DNN models are trained for conversion rate prediction and revenue estimation, denoted as *DNN-C* and *DNN-R* respectively. The feature embedding sizes are set to 24 and the batch size to 512 for all cases. We conduct the grid search for learning rate in the range of {0.01, 0.02, 0.04, 0.06} and dropout in {0.1, 0.3, 0.5} for each model, respectively. All models are optimized with Adam [10]. AUC is used for conversion prediction while Mean Absolute Error (MAE) and Mean Squared Error (MSE) are used to evaluate revenue estimation [30, 28].

Results reported in Table 1 illustrate that our model performs best in both conversion rate and revenue prediction tasks.

**Optimization Algorithm *HA* Evaluation** :

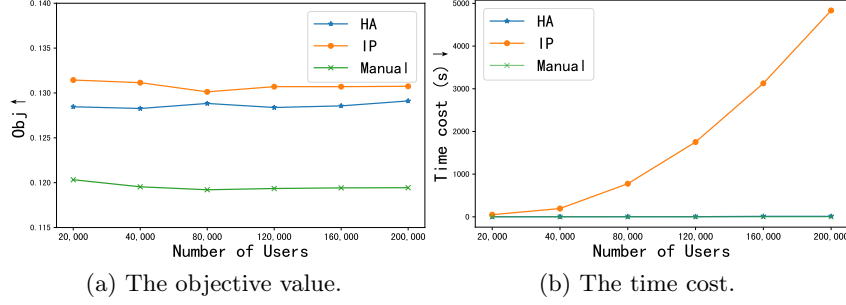(a) The objective value.          (b) The time cost.

Fig. 4: Comparison of different allocation strategies.

Following the prior work [9], we randomly sample datasets of varying user scales from the **LCT-A** dataset. For each sampled dataset with scale $S$, we linearly scale down the fund exposure constraint $d_f$ according to the sampling ratio and simulate the exposure allocation. Two baselines are introduced as a comparison. (1)*manual* allocation strategy. The priority of the funds is manually defined, given the funds' average historical conversion rate and the operation specialists' experience. Specifically, for the fund $f$ with the highest priority, we iteratively expose it to $d_f$ customers with the highest predicted revenue score until the allocation is completed. This is the baseline optimization algorithm, which is more similar to the traditional recommendation. (2)*integer programming (IP)* solver, an off-the-shelf method for combinatorial optimization[14].Fig. 4a shows that the IP achieves the best objective. Our proposed HA can obtain a similar performance as IP and can obtain significant improvement compared to manual allocation. For example, in the dataset with 200,000 users, HA achieves the **98.75%** objective score of IP. Fig. 4b shows the time cost of different strategies. The time cost of HA and Manual is close and continues to be significantly lower than IP in different data scales. In the dataset with 200,000 users, the speed-up ratio achieves **416**. In our business, the allocation problem involves over 10 million users, and the solution time for traditional IP methods is unacceptable.

## 4.2  Result from online A/B testings

We have deployed our PTOFA on a large-scale online financial platform and conducted online A/B testing experiments on promotional campaigns. There are 10 million customers for each campaign, which is further divided into 5 million in the control and experimental groups, respectively. In each campaign, there are eight distinct candidate funds available for exposure, and each user will only be exposed to one of these funds (K=1). For online A/B testing, we focus on two metrics: i) Conversion Per Mille Exposures (CPME) and ii) Revenue Per Mille Exposures (RPME), which are formulated as follows:

$$\mathbf{CPME} = \frac{\# \ \mathbf{conversion}}{\# \ \mathbf{exposures}} \times 1000, \quad \mathbf{RPME} = \frac{\sum \mathbf{revenue}}{\# \ \mathbf{exposures}} \times 1000. \quad (8)$$

Table 2: Online A/B testing results of different frameworks.

| Period | Predict | Optimize | CPME | RPME | %CPME Lift | %RPME Lift |
|---|---|---|---|---|---|---|
| P1 | ESMM | Manual | 2.69 | 14,901 | - | - |
|  | ESMM | HA | 2.88 | 13,492 | 7.06% | -9.46% |
| P2 | ESMM | HA | 2.05 | 7,359 | - | - |
|  | Ours | HA | 2.55 | 10,048 | 24.39% | 36.54% |
| P3 | ESMM | Manual | 3.39 | 16,608 | - | - |
|  | Ours | HA | 4.29 | 25,558 | 26.54% | 53.89% |

As the Table. 2 illustrated, our proposed PTOFA framework is superior to the current online baseline framework, which uses ESMM to predict the conversion rate after the fund $f$ exposed to user $u$ as $\mathbb{E}_{u,f}$ and employs the manual allocation strategy. Notice that the ESMM+Manual campaign recommends items in a traditional manner, which further indicates the significance of our proposed PTOFA framework.

## 5    Related Work

### 5.1    Generalized Recommendation and Fund Recommendation

Generalized recommendations have been adopted in various online services, such as video distribution [11], real estate [22], and financial product [18, 8]. Top-$k$ recommendation is a widely researched topic, where the goal is to recommend a list of items to users that users may be interested in [9]. It is able to enlarge the engagement of users, and is more practical in accords to real-world recommendation scenarios. Starting from matrix factorization techniques [17] to learn latent representation of users and items, increasing research focus on deep neural network-based methods. NeuMF [7] and NGCF [20] utilize multi-layer perception and graph neural networks to boost performance. This also motivates research to deploy top-$k$ recommendations on fund distribution. Graphical deep collaborative filtering has been used in fund recommendation [8, 3, 19]. However, we argue that generalized recommendation leads to some problems for fund matching on online investment platforms, motivating a shift from recommendation to allocation.

### 5.2    Predict-then-Optimize Problem

Predict-then-Optimize(PTO) is summarized as predicting unknown parameters based on historical data and generating decisions by solving the corresponding optimization problem via the predicted parameters. SPO [4] is a general framework for solving this problem by proposing an SPO loss to measure the error in predicting the cost vector of the optimization problem. Many applications involve PTO problems, e.g. couriers allocation [27], portfolio investment [1]. PTOC [27]

is proposed to solve the couriers allocation for emergency last-mile logistics. It predicts user demand with a variational graph GRU encoder and optimizes the courier task allocation problem with multi-agent graph reinforcement learning. As far as we know, we are the first to adopt PTO in fund allocation.

## 6    Conclusion

In this paper, we propose a Predict-Then-Optimize Fund Allocation framework that transforms the fund-matching problem from recommendation to allocation. The PTOFA consists of two stages. In the prediction stage, we propose a revenue prediction module that deals with how to predict revenue over the entire space. In the optimization stage, we design an efficient heuristic algorithm to allocate the funds under the necessary constraints, i.e., *customer top-k*, *fund exposure*, and *platform risk*. Extensive online and offline experiments on a real investment platform confirm the effectiveness and efficiency of our proposed framework.

## References

1. Ban, G., Karoui, N.E., Lim, A.E.B.: Machine learning and portfolio optimization. Manag. Sci. (2018)
2. Caruana, R.: Multitask learning. Machine learning **28**, 41–75 (1997)
3. Chou, Y., Chen, C., Huang, S.: Modeling behavior sequence for personalized fund recommendation with graphical deep collaborative filtering. Expert Syst. Appl. **192**, 116311 (2022)
4. Elmachtoub, A.N., Grigas, P.: Smart "predict, then optimize". Manag. Sci. (2022)
5. Gao, C., Huang, K., Chen, J., Zhang, Y., Li, B., Jiang, P., Wang, S., Zhang, Z., He, X.: Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In: SIGIR (2023)
6. Ghiye, A., Barreau, B., Carlier, L., Vazirgiannis, M.: Adaptive collaborative filtering with personalized time decay functions for financial product recommendation. In: Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023 (2023)
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: WWW (2017)
8. Hsu, P., Chen, C., Chou, C., Huang, S.: Explainable mutual fund recommendation system developed based on knowledge graph embeddings. Appl. Intell. (2022)
9. Ji, H., Zhu, J., Shi, C., Wang, X., Wang, B., Zhang, C., Zhu, Z., Zhang, F., Li, Y.: Large-scale comb-k recommendation. In: WWW (2021)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Lei, H., Zhao, Y., Cai, L.: Multi-objective optimization for guaranteed delivery in video service platform. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '20 (2020)
12. Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In: KDD (2018)
13. Ma, X., Zhao, L., Huang, G., Wang, Z., Hu, Z., Zhu, X., Gai, K.: Entire space multi-task model: An effective approach for estimating post-click conversion rate. In: SIGIR (2018)

14. Optmizor, G.: https://developers.google.com/optimization
15. Pei, C., Yang, X., Cui, Q., Lin, X., Sun, F., Jiang, P., Ou, W., Zhang, Y.: Value-aware recommendation based on reinforcement profit maximization. In: WWW (2019)
16. Ren, J., Zhang, M., Yu, C., Liu, Z.: Balanced MSE for imbalanced visual regression. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022. pp. 7916–7925. IEEE (2022)
17. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: bayesian personalized ranking from implicit feedback. In: UAI (2009)
18. Tang, X., Qiao, Y., Lyu, F., Liu, D., He, X.: Touch the core: Exploring task dependence among hybrid targets for recommendation. In: Proceedings of the 18th ACM Conference on Recommender Systems. RecSys '24 (2024)
19. Tu, K., Qu, W., Wu, Z., Zhang, Z., Liu, Z., Zhao, Y., Wu, L., Zhou, J., Zhang, G.: Disentangled interest importance aware knowledge graph neural network for fund recommendation. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. CIKM '23
20. Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: SIGIR (2019)
21. Wang, X., Liu, T., Miao, J.: A deep probabilistic model for customer lifetime value prediction. arXiv preprint arXiv:1912.07753 (2019)
22. Wei, S., Tong, Y., Zhou, Z., Liu, Q., Zhang, L., Zeng, Y., Ye, J.: Towards capacity-aware broker matching: From recommendation to assignment. In: ICDE (2023)
23. Wen, H., Zhang, J., Wang, Y., Lv, F., Bao, W., Lin, Q., Yang, K.: Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '20 (2020)
24. Weng, Y., Tang, X., Chen, L., He, X.: Curriculum modeling the dependence among targets with multi-task learning for financial marketing. In: SIGIR (2023)
25. Weng, Y., Tang, X., Chen, L., Liu, D., He, X.: Expected transaction value optimization for precise marketing in fintech platforms. arXiv preprint arXiv:2401.01525 (2024)
26. Weng, Y., Tang, X., Xu, Z., Lyu, F., Liu, D., Sun, Z., He, X.: Optdist: Learning optimal distribution for customer lifetime value prediction. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. p. 2523–2533. CIKM '24 (2024)
27. Xia, K., Lin, L., Wang, S., Wang, H., Zhang, D., He, T.: A predict-then-optimize couriers allocation framework for emergency last-mile logistics. In: KDD (2023)
28. Xing, M., Bian, S., Zhao, W.X., Xiao, Z., Luo, X., Yin, C., Cai, J., He, Y.: Learning reliable user representations from volatile and sparse data to accurately predict customer lifetime value. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. p. 3806–3816. KDD '21 (2021)
29. Yang, X., Jia, B., Wang, S., Zhang, S.: Feature missing-aware routing-and-fusion network for customer lifetime value prediction in advertising. In: WSDM 16(2023). pp. 1030–1038 (2023)
30. Zhang, S., Yan, X., Yang, X., Jia, B., Wang, S.: Out of the box thinking: Improving customer lifetime value modelling via expert routing and game whale detection. In: CIKM '23. p. 3206–3215 (2023)
31. Zhu, F., Zhong, M., Yang, X., Li, L., Yu, L., Zhang, T., Zhou, J., Chen, C., Wu, F., Liu, G., et al.: Dcmt: A direct entire-space causal multi-task framework for post-click conversion estimation. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). pp. 3113–3125. IEEE (2023)