Joint Incentive Optimization of Customer and Merchant in Mobile Payment Marketing

Li Yu*, Zhengwei Wu*, Tianchi Cai, Ziqi Liu,
Zhiqiang Zhang, Lihong Gu, Xiaodong Zeng, Jinjie Gu
Ant Group
{jinli.yl,zejun.wzw,tianchi.ctc,ziqiliu,lingyao.zzq,lihong.glh,xiaodong.zxd,jinjie.gujj}@antgroup.com

Abstract

In the mobile Internet era, *mobile payment service* becomes the foundation of inclusive finance, which brings convenience and security to people. Various marketing strategies are designed to encourage mobile payment activities by allocating incentives such as coupons or commissions to customers or merchants. We summary two significant issues. First, there is a phenomenon of mutual influence between merchants and customers, i.e., *bipartite influence* issue, thus making the independent optimization of customers and merchants non-optimal. Second, the redemptions of coupons are partially observed, as we can only observe that the customer redeems the coupon or not at a specific incentive value, but cannot observe that at other incentive value, i.e., *data censorship* issue. In this paper, we propose a novel joint incentive optimization framework to address the above two issues. We propose to use a graph neural network to represent customers and merchants jointly by modeling the underlying bipartite influences. We then formulate the response model under the hazard regression setting and model the hazard rate with a piecewise nonlinear function to capture the changes of responses to different incentive values. Finally, we propose a linear programming method to allocate approximated optimal incentive values to customers and merchants in real-time. Extensive offline and online experimental results demonstrate the effectiveness of our proposed approach.

Introduction

Inclusive finance, which is first mentioned by the United Nations in the early 2000s, is defined as the availability and equality of opportunities to access financial services. Recently, with the rapid development of *mobile internet*, the idea of inclusive finance is gaining significant ground, helping more than a billion people access financial services with just a mobile phone. In the mobile Internet era, *mobile payment service* becomes the foundation of inclusive finance, since payment is one of the most basic financial services. However, there still exists a large number of people who barely use mobile payments because of unaccustomedness or worries. To encourage mobile payment activities, one effective marketing strategy is to motivate both customers and merchants to accept mobile payment services by offering them incentives (e.g., coupons, commissions, bonus) under specific budget constraints.

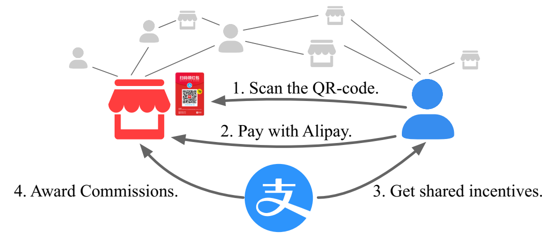


Figure 1: An illustration of the marketing strategy.

Take an example of a marketing strategy at Alipay and Wechat Pay. We illustrate the overall marketing campaign in Figure 1. Each merchant is assigned with a unique incentive QR Code and is encouraged to ask their customers to scan so that the customer can get a certain amount of incentive (coupon). Each customer can redeem the coupon by making proper payment, and the merchant who shares the incentive QR Code is awarded an amount of incentive (commission). With the incentives propagated and redeemed under such a mechanism, the more payments customers can accomplish, the more coupons customers can redeem, and the more commissions merchants can gain.

In practice, people may be insensitive or sensitive to incentives, so that incentive optimization of customers and merchants is the key to maximizing mobile payment activities. For instance, the response (i.e., the probability of redeem the coupon) should remain unchanged for customers with limited interest, no matter their incentives. For merchants with potential enthusiasm or abilities, the more commissions they can get, the more customers they can attract to scan the QR Code.

Previous works have studied various methods (Boutilier and Lu 2016; Beheshti-Kashi et al. 2015) for incentive optimization under a limited budget in the marketing domain. Ito et al. (2017) solved price optimization by first forecasting the relationships between sales and prices. Zhao et al. (2019) proposed a general online budget allocation framework consisting of two components: forecasting models and decision making. However, existing methods ignore the phenomenon of mutual influence between merchants and customers, i.e.,

*First Author and Second Author contribute equally to this work.

bipartite influence. The incentive QR Code scanning and payment behaviors form a large bipartite graph between customers and merchants. When rewarded with more commissions, the merchants would be more willing to ask customers to scan the incentive QR Code. Influenced by the merchants, their neighboring customers in the graph are more likely to redeem the incentives in their shops and thus have higher incentive response, i.e., the probability of redeem the incentive. Influenced by the increasing number of consuming customers, their neighboring merchants would be more willing to encourage them to scan the QR Code. Another issue is that the redemption data on the customer-side are partially observed, i.e., *censored data*. To address this issue, Zhao et al. (2019) propose a response model with a sigmoid function, but ignore the effect of censored data. In reality, when a customer successfully redeems a coupon at a certain value, we cannot observe whether this customer will still redeem at a smaller coupon value, i.e., left-censored data. For any failed redemption at a certain value, we still cannot observe whether the customer will fail the redemption at a larger coupon value, i.e., right-censored data. As such, customers' responses to any given value are partially observed.

In this paper, we propose a novel joint incentive optimization framework to address the above issues. First, we build a joint representation learning task for customers and merchants based on graph neural networks built atop the underlying customer-merchant *bipartite influence* graph, which benefits both representations. Second, we propose a response model layer based on the representations, which estimates the customer's response score, i.e., the probability of customer redeem the coupon. A reasonable estimator should guarantee that the response model's output, i.e., response score, should be cumulative as the incentive value increases. We formulate the response score as a hazard regression problem to address partial observations of *censored data*. Third, we allocate the optimal incentive value to each customer or merchant based on the estimates of the customers and merchants' responses to incentives, under a specific budget in real-time. We formulate the real-time decision making as linear programming and solve it with the Lagrangian method.

We conduct extensive experiments on the Alipay dataset. Moreover, the dataset will be released in the future, which will make the research community further improve. Experimental results show that our proposed framework significantly outperforms other competitive methods. We deploy the proposed framework in the online environment, which can make the budget allocation for thousands of requests per second.

In summary, our contributions in this work are as follows: 1) We propose a graph-based representation learning method to model the bipartite influence between customers and merchants jointly. 2) We develop a response model based on hazard regression with a nonlinear piecewise hazard rate to handle *censored data*. 3) We propose a real-time incentive allocation method for customers and merchants.

Background

Graph Neural Networks. As discussed in Section 1, customers' responses to incentives are correlated with merchants' who interact with each other, and vice versa. Graph neural network is one of the natural ways of modeling latent variables when correlated with each other.

Graph neural networks (GNNs) (Wu et al. 2019) study the learning representation of vertices by aggregating features (Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2017; Liu et al. 2019a; Xu et al. 2018) based on the neighbors defined by the graph, which have demonstrated state-of-the-art performance on learning tasks such as node classification and link prediction, with applications ranging from social networks (Hamilton, Ying, and Leskovec 2017), transaction networks (Liu et al. 2018), gene expression networks (Fout et al. 2017), knowledge graphs (Schlichtkrull et al. 2018), cash-out detection (Hu et al. 2019), and fraud conspiracy detection (Liang et al. 2019).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the graph with nodes $v_i \in \mathcal{V}$, $|\mathcal{V}| = N$, and edges $(v_i, v_j) \in \mathcal{E}$. Let the adjacency matrix denote as $A \in \mathbb{R}^{N\times N}$. Associated with the graph, we also have a feature matrix $X \in \mathbb{R}^{N\times D^{(0)}}$ with x_i denoting the original $D^{(0)}$-dimensional feature for node v_i on the 0-th layer. The simple form of GNNs can be formalized as follows:

$$h^{(l+1)}(v_i) = \sigma\left(\sum_{j=1}^{N} \alpha(v_i, v_j)\, h^{(l)}(v_j)\, W^{(l)}\right) \tag{1}$$

where we define $h^{(l)}(v_i)$ as the hidden feature of the node v_i on the l-th layer, the kernel as $\boldsymbol{\alpha} = (\alpha(v_i, v_j)) \in \mathbb{R}^{N\times N}$, the transform parameter on the l-th layer $W^{(l)} \in \mathbb{R}^{D^{(l)}\times D^{(l+1)}}$, and $\sigma(\cdot)$ as the nonlinear function. The kernel $\boldsymbol{\alpha}$ is essentially defined on the correlation structure among the latent variables. For instance, in our task, a merchant successfully share incentives to a customer, then they are probably correlated with each other on the response to incentives.

Hazard Regression. Hazard regression is commonly used in survival analysis of patients suffering from potentially fatal diseases. There, one aims to estimate the chances of survival of a particular patient with covariates (attributes) x as a function of time to understand the effects of x better. Unfortunately, each patient only has one life and possibly different attributes x. Hence, it is impossible to estimate the fatality rate directly. Instead, one assumes that the hazard rate $\lambda(x, t)$ governs the instantaneous rate of dying of any x at any given time t:

$$\begin{aligned}\lambda(x, t) &= \lim_{dt\to 0} \frac{p(t \le T < t + dt | T \ge t, x)}{dt} \\ &= \lim_{dt\to 0} \frac{p(t \le T < t + dt | x)}{dt} \cdot \frac{1}{p(T \ge t|x)}\end{aligned} \tag{2}$$

That is, the density of dying at time t is given by

$$p(t|x) = \lambda(x, t)\, \underbrace{p(T \ge t|x)}_{F(t|x)}. \tag{3}$$

This leads to a differential equation for the survival probability with solution $F(t|x) = \exp(-\int_0^t \lambda(x, \tau) d\tau)$. Here we assumed, without loss of generality, that time starts at 0.

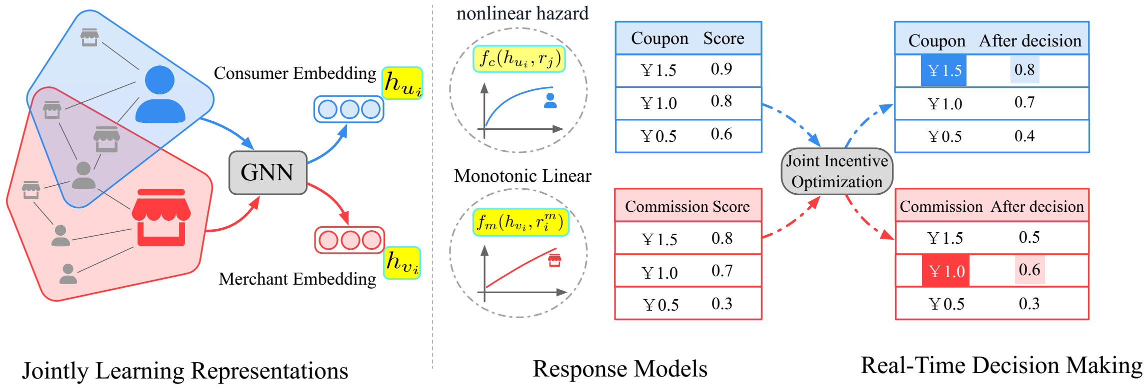


Figure 2: An illustration of the our proposed framework for joint incentive optimization.

In our case, death amounts to the redemption of incentives. We denote an ordered set of incentives $\mathcal{R} = \{r_1, ..., r_{|\mathcal{R}|}\}$ in terms of their values. Based on the evidence of redemptions $\Omega_p = \{(u_i, r_j) | y_{u_i, r_j} = 1\}$, and fails of redemptions $\Omega_n = \{(u_{i'}, r_{j'}) | y_{u_{i'}, r_{j'}} = 0\}$, we have the following likelihood for the observed data:

$$p(\Omega_p \cup \Omega_n | \mathcal{R}) = \prod_{\Omega_p} (1 - F(r_j | u_i)) \prod_{\Omega_n} (F(r_{j'} | u_{i'})), \tag{4}$$

where $1 - F(r_j | u_i)$ denotes the probability user u_i will redeem the incentive at value r_j, i.e., response score, and $F(r_j | u_i)$ denotes that the probability u_i will not redeem the coupon at least if the user has a incentive at value r_j.

Most hazard regression approaches are based on the Cox's propotional hazard model $\lambda(t|x) = \lambda_0(t) \exp(w^\top x)$ (Cox 1972), including parametric models, and nonparametric models with baseline hazard rate $\lambda_0(t)$ unspecified. In this paper, we present a nonlinear parametric hazard regression model inspired by (Liu et al. 2017).

Method

We illustrate our overall framework in Figure 2 which will be discussed in the following sections in detail.

Jointly Learning Representations of Customers and Merchants

Assuming that $\mathcal{R}$ is the finite ordered treatment list, which is defined as $\mathcal{R} = (r_1, ..., r_{|\mathcal{R}|})$ where $r_1 < r_2 ... < r_{|\mathcal{R}|}$. Given customer set $\mathcal{S}_c$ and merchant set $\mathcal{S}_m$, let $r^c \in \mathcal{R}$ be the coupon assigned to customers u and $r^m \in \mathcal{R}$ be the commission assigned to merchants v. Then we can formulate the possibility of customer u to redeem the incentive r^c as:

$$p(r^c | u) = \sum_{v \in \mathcal{S}_m} p(r^c, v | u) \tag{5}$$

where $p(r^c, v|u)$ is the possibility of the customer u to redeem the incentive r^c in the merchant v. Similarly, we can formulate the incentive sharing quantity made by the merchant v under the reward r^m as:

$$q(r^m | v) = \sum_{u \in \mathcal{S}_c} q(r^m, u | v) \tag{6}$$

where $q(r^m, u|v)$ is the incentive sharing quantity made by the merchant v to the customer u.

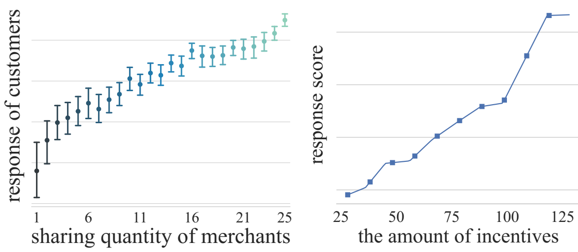
However, it is difficult to apply (5) and (6) directly in an industrial system, since computing each response value requires whole customer or merchant's information. Observing that most customers tend to visit merchants they made payments before, we build a graph $\mathcal{G}$ based on the historical trading data. Therefore we replace the whole merchant set $\mathcal{S}_m$ with customer u's neighbor merchants $\mathcal{N}_u$ in (5) and replace the whole customer set $\mathcal{S}_c$ with the merchant v's neighbor customers $\mathcal{N}_v$ in (6). Thus we could approximate the response value by aggregating neighbors in $\mathcal{G}$. Furthermore, it is natural to use graph neural networks to implement the aggregating process above.

Assuming $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix of graph $\mathcal{G}$ and $X \in \mathbb{R}^{N \times D}$ is the input feature matrix. $h_{u_i} = \mathrm{GNN}(u_i | A, X)$ denotes the embedding of customer u_i node and $h_{v_i} = \mathrm{GNN}(v_i | A, X)$ denotes the embedding of merchant v_i node. In the campaign, merchants will interact with their customers to make both of them get their incentives back from Alipay. We analyze such interactions as follows. We rank all the customers according to their sharing quantities. For each merchant, we calculate the averaged responses of customers who interact with that merchant. We show the relationship between the sharing quantity of merchants and the averaged responses of customers who interact with them in Figure 3a. It demonstrates that high responses of customers tend to interact with merchants who are likely to share incentives. The reason is that intuitively merchants with high responses (large sharing quantity) would influence their customers so that the influenced customers tend to have similar tastes on the values of coupons, and vice versa.

We have training samples $\{(u_i, r_i^c, y_i^c)\}_{i=1}^{|\mathcal{S}_c|}$ denoting whether customer u_i redeems the incentive r_i^c and $\{(v_i, r_i^m, y_i^m)\}_{i=1}^{|\mathcal{S}_m|}$ denoting the incentive sharing quantity of merchant v_i under the reward r_i^m, where $y_i^c \in [0, 1]$ and $y_i^m \in \mathbb{N}$. The loss function can be defined as:

$$\sum_{i=1}^{|\mathcal{S}_c|} l_c\Big(y_i^c, f_c(h_{u_i}, r_i^c)\Big) + \sum_{i=1}^{|\mathcal{S}_m|} l_m\Big(y_i^m, f_m(h_{v_i}, r_i^m)\Big) \tag{7}$$

where $f_c(h_{u_i}, r_i^c)$ and $f_m(h_{v_i}, r_i^m)$ denote the response score of customer u_i and merchant v_i respectively. l_c and



(a) (b)

Figure 3: (a) The correlations between of customers and merchants. (b) customers' response vary with different coupons.

l_m are the losses for the customer and merchant, respectively. For the instantiation of the above losses, please refer to Section 4. Besides, the response scores of customers and merchants are not on the same scale. To optimize customers and merchants jointly, we normalize the merchants' response score by dividing the number of sharing incentives received in the previous days.

Response Models

In this section, assuming that we have the encoding h_{u_i} and h_{v_i} that includes all the knowledge of the user's sensitivities to different values, we design our response models. For merchants, we use the same response models as in (Liu et al. 2019b). We mainly focus on the response model of estimating whether a customer will make an action or not under different amounts of incentives.

We first illustrate the expected response to incentives by averaging over all customers in our random experiments in Figure 3b. We found that the overall response score follows a monotone pattern. It is reasonable that with a higher amount of incentive (coupon), a customer would have a higher probability of redeeming it. Without place this prior in our response model, the model will easily over-fit the data due to the noise introduced from our dataset. However, users' responses to incentives could vary a lot. There will be some noticeable change points at certain incentive value. Thus users' responses to incentives are not smooth across the entire interval. It is not reasonable to model these curves with just a linear representation with a fixed smooth exponential function as in (Zhao et al. 2019). In this part, inspired by traditional hazard regression, we treat the time t in hazard regression as the value of coupons in our setting. As mentioned in Section 2, we aim to formulate a parameterized hazard rate function for our problem.

Given the random experiment, we randomly assign different customers to different buckets with different amount of incentives r_j, all the values in our training dataset are in the ordered set $\mathcal{R} = (r_1, ..., r_{|\mathcal{R}|})$. Instead of modeling the hazard rate function as $\lambda(u_i, r_j) = \lambda_0(r_j)\exp(w^\top h_{u_i})$, we use a nonlinear piecewise form:

$$\begin{aligned} \lambda(u_i, r_j) = \lambda_0(r_j)\exp(w^\top h_{u_i}) = \exp(w_j^\top h_{u_i} + b_j), \\ s.t. \sum_{j=1}^{|\mathcal{R}|-1} |w_{j+1} - w_j| < \zeta \end{aligned} \tag{8}$$

where we parameterize the hazard rate of each piece of segments independently and essentially penalize the hazard rate of each piece with fused lasso (Tibshirani et al. 2014). In our setting, we have the following loss functions:

$$\begin{aligned} \mathcal{L}_c &= -\sum_{(u_i,r_j)\in\Omega_p} \log\left(1 - F(r_j|u_i)\right) \\ &\quad - \sum_{(u_{i'},r_{j'})\in\Omega_n} \log\left(F(r_{j'}|u_{i'})\right) \\ &= -\sum_{(u_i,r_j)\in\Omega_p} \log\left(1 - \exp\left(-\int_0^{r_j} \lambda(u_i,\tau)d\tau\right)\right) \\ &\quad + \sum_{(u_{i'},r_{j'})\in\Omega_n} \int_0^{r_{j'}} \lambda(u_{i'},\tau)d\tau, \end{aligned} \tag{9}$$

and we can infer the response score that user u_i redeem a coupon r_j as:

$$f_c(h_{u_i}, r_j) = 1 - \exp\left(-\int_0^{r_j} \lambda(u_i,\tau)d\tau\right) \tag{10}$$

Joint Incentive Optimization & Real-Time Decision Making

As described in section 3 and 4, we can estimate the response scores of customers $f_c(h_{u_i}, r_j)$ with short form $f^c_{i,j}$. In the same way, we could predict the response score of merchants $f_m(h_{v_i}, r_j)$ with short form $f^m_{i,j}$. Then the *joint incentive optimization* problem can be formulated as a linear programming problem and be solved with the Lagrangian method.

There exist two types of requests: the customer request q^c_i and the merchant request q^m_i. The customer request aims to decide the incentive value received once a customer scans the incentive QR-code. Moreover, the merchant request is to decide the reward commission value received when an incentive shared is redeemed. $Q_c = \{q^c_1, ...\}$ denote the customer request list and $Q_m = \{q^m_1, ...\}$ denote the merchant request list. Our objective is to find an incentive allocation strategy that maximizes the sum of customers' response score (i.e., current redemption rate) and merchants' response score (i.e., future redemption rate) under the given budge B with respect to each user. Formally,

$$\max_{x^c_{i,j}, x^m_{i,j}} \sum_{i=1}^{|Q_c|}\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} f^c_{i,j} + \sum_{i=1}^{|Q_m|}\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j} f^m_{i,j} \tag{11}$$

$$s.t. x^c_{i,j} \in [0,1], \text{ for } i = 1, ..., |Q_c|, j = 1, ..., |\mathcal{R}| \tag{12}$$

$$x^m_{i,j} \in [0,1], \text{ for } i = 1, ..., |Q_m|, j = 1, ..., |\mathcal{R}| \tag{13}$$

$$\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} = 1, \text{ for } i = 1, ..., |Q_c| \tag{14}$$

$$\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j} = 1, \text{ for } i = 1, ..., |Q_m| \tag{15}$$

$$\frac{\sum_{i=1}^{|Q_c|}\sum_{j=1}^{|\mathcal{R}|} r_j x^c_{i,j} f^c_{i,j} + \sum_{i=1}^{|Q_m|}\sum_{j=1}^{|\mathcal{R}|} r_j x^m_{i,j}}{\sum_{i=1}^{|Q_c|}\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} f^c_{i,j} + \sum_{i=1}^{|Q_m|}\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j}} \le B \tag{16}$$

where $x^c_{i,j}$ is the decision variable of whether choosing treatment r_j to customer in the i-th request of Q_c and $x^m_{i,j}$ is the decision variable of whether choosing treatment r_j to merchant in the i-th request of Q_m. Under constraints (14) and (15), $\sum_{j=1}^{|\mathcal{R}|} r_j x^c_{i,j}$ is the incentive value for the customer and $\sum_{j=1}^{|\mathcal{R}|} r_j x^m_{i,j}$ is the incentive value for the merchant. The constraint (16) defines the limited average incentive value B given to each customer or merchant.

By introducing dual variables μ^c_i, μ^m_i, and λ corresponding to the constraints (14) (15) (16), the Lagrange multiplier method is applied to solve the convex optimization problem in objective function (11):

$$
\begin{aligned}
&-\sum_{i=1}^{|Q_c|}\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} f^c_{i,j} - \sum_{i=1}^{|Q_m|}\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j} f^m_{i,j} \\
&+\sum_{i=1}^{|Q_c|}\left(\mu^c_i\Big(\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} - 1\Big)\right) + \sum_{i=1}^{|Q_m|}\left(\mu^m_i\Big(\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j} - 1\Big)\right) \\
&+\lambda\left(\sum_{i=1}^{|Q_c|}\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} f^c_{i,j}(r_j - B) + \sum_{i=1}^{|Q_m|}\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j}(r_j - B)\right)
\end{aligned} \tag{17}
$$

Rearrange the new objective function, we have

$$
\begin{aligned}
&\sum_{i=1}^{|Q_c|}\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j}\left[f^c_{i,j}(\lambda r_j - \lambda B - 1)\right] + \sum_{i=1}^{|Q_m|}\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j}\left[\lambda r_j - \lambda B - f^m_{i,j}\right] \\
&+\sum_{i=1}^{|Q_c|}\left(\mu^c_i\Big(\sum_{j=1}^{|\mathcal{R}|} x^c_{i,j} - 1\Big)\right) + \sum_{i=1}^{|Q_m|}\left(\mu^m_i\Big(\sum_{j=1}^{|\mathcal{R}|} x^m_{i,j} - 1\Big)\right)
\end{aligned} \tag{18}
$$

Both objective function and constraint functions are convex. Applying KKT conditions and L-BFGS (Liu and Nocedal 1989), we could get the optimal λ^*. In a real-time environment, we have to determine the value of incentives allocated to customers and merchants, under a specific budget for each merchant or customer, i.e., *real-time decision making*. Observing that in (18), the different incentive values are related to the first two terms that are independent of the rest terms. Then the approximated solutions can be derived as:

$$
x^c_{i,j} = \begin{cases} 1, & \text{if } j = arg\min_j f^c_{i,j}(\lambda^* r_j - \lambda^* B - 1) \\ 0, & \text{otherwise} \end{cases} \tag{19}
$$

$$
x^m_{i,j} = \begin{cases} 1, & \text{if } j = arg\min_j \lambda^* r_j - \lambda^* B - f^m_{i,j} \\ 0, & \text{otherwise} \end{cases} \tag{20}
$$

In the online environment, for each request, we fetch the graph feature as input of the response model and predict the response scores of the given user for all values in the treatment list. Then based on the value of λ^* and the decision formulas (19) and (20), we get the incentive value. Because the real-time decision making is based on only $|\mathcal{R}|$ times calculations of response scores, the online serving is simple and high-performance.

Experiment

To demonstrate our approach's effectiveness, we first conduct offline experiments based on our dataset collected from the online environment at Alipay. Next, we present our online A/B testing results compared with other methods described in section .

$\vert\mathcal{V}_c\vert$	99.73 x 10^6
$\vert\mathcal{V}_m\vert$	22.58 x 10^6
$\vert\mathcal{V}\vert$	112.59 x 10^6
$\vert\mathcal{E}\vert$	133.64 x 10^6
# node feature dim	33501
# edge feature dim	2468
# labeled records	1.85 x 10^6

Table 1: Experimental summary for dataset and customer-merchant interactions.

Experimental Settings

Dataset [1]

Our data are collected from the online Alipay marketing campaign with more than 1.85 million of samples. Randomized experiments are performed to estimate users' responses to incentives. After randomly select 5% merchants and customers, we partition them into buckets and randomly assign treatment for each bucket. A customer can redeem the shared incentives within the next three days so that we can obtain positive labels if the customer redeems and negative labels if the customers do not redeem. As mentioned in section , the label of merchants is first defined as the 3-days sharing quantity. However, to match the scale of customers' labels, it is normalized by dividing the number of sharing incentives received in the previous 3-days.

We build our model on top of the customer-merchant interactions. These interactions lead to more than 112.59 millions of vertices (including merchants and customers) and 133.64 millions of edges. We summarize our dataset and graph that will be used for offline experiments in Table 1.

Baselines To verify the effectiveness of graph neural networks, we compare it with classic deep neural networks (Schmidhuber 2015) with or without historical customer-merchant interactions. For all models, hazard regression is applied to the estimation of customers' responses to incentives, and the linear mapping layer (Liu et al. 2019b) is utilized for merchants' estimation.

DNN: Deep neural networks with fully connected multi-layer perceptron architectures (Schmidhuber 2015) .

DNN-H: **DNN** with historical customer-merchant interactions as additional features.

To verify the effectiveness of hazard regression, we compare our proposed method with state-of-the-art monotonic methods. All models are trained on top of our graph neural networks.

[1] 1. The data set does not contain any Personal Identifiable Information (PII) 2. The data set is desensitized and encrypted 3. Adequate data protection was carried out during the experiment to prevent the risk of data copy leakage, and the data set was destroyed after the experiment 4. The data set is only used for academic research, it does not represent any real business situation

metric	Our Model	DNN-H	DNN
AUC_c	0.9303†	0.8879	0.8149
MAE_c	0.1934†	0.2880	0.3145
MSE_c	0.0973†	0.1391	0.1555
MAE_m	0.3241†	0.3584	0.3929
MSE_m	0.1486†	0.1698	0.1970

Table 2: Offline performance between the proposed method, DNN-H and DNN

metric	Our Model	SBBM	Cox	DSSM-c
AUC_c	0.9303†	0.9246	0.9221	0.9150
MAE_c	0.1934†	0.1984	0.2106	0.2409
MSE_c	0.0973†	0.1007	0.1113	0.1183
MAE_m	0.3241†	0.3298	0.3319	0.3369
MSE_m	0.1486†	0.1541	0.1567	0.1524

Table 3: Offline performance between different response models.

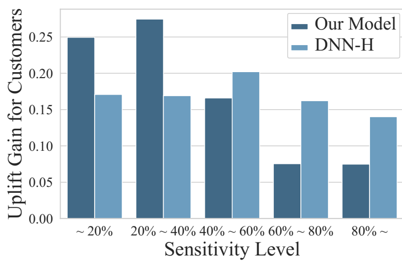


Figure 4: Comparisons of uplift gain for customers vary with different incentive sensitivity levels.

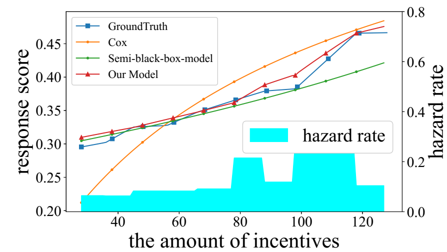


Figure 5: A visualization of different response models.

SBBM: Semi-black-box model extends logit demand curves with the capability of neural networks (Zhao et al. 2019).

DSSM-c: Deep structured semantic models with cost constraint (Huang et al. 2013).

Cox: The classic Cox Proportional model has been extensively used for survival analysis (Cox 1972).

For all experiments, we use Adam optimizer (Kingma and Ba 2014). We apply exponential decay with a learning rate starting at 0.0001 and a decay rate of 0.95. We set the number of hidden units for two fully-connected layers as 256 and 128 with ReLU activation, respectively. We set the dimension of graph embedding in GNN as 64. We set the depth of our graph neural network as 2. The rest of the hyperparameters (such as regularizers) of the comparison methods are tuned via standard grid search. 85% of the merchants and customers are selected for training while the remaining are for validation purposes. In offline experiments, we use the Area Under Curve (AUC_c), Mean Absolute Error (MAE_c) and Mean Square Error (MSE_c) as metrics for customers. We use MAE_m and MSE_m as metrics for the measurement of merchants' performance. The overall system is built on Alipay's graph learning system AGL (Zhang et al. 2020).

Results

Evaluation of Graph Neural Networks We conduct experiments on the real-world dataset collected from the online Alipay marketing campaign, as described above. Table 2 shows that our proposed model obtains the improvement compared to DNN-H, as we leverage information on customer-merchant bipartite influence and learns a better representation of both customers and merchants. Both of these methods outperform DNN, which demonstrates the importance of historical customer-merchant interactions.

To visualize the learned representation between DNN-H and our model, we analyze the estimates of customers' "gradient" (Liu et al. 2019b) which is defined as:

$$f_c(h_{u_i}, r_j) = \text{sigmoid}\big(\underbrace{\text{softplus}(w_g^T h_{u_i})}_{\text{“gradient”}} \cdot r_j + \underbrace{(w_i^T h_{u_i})}_{\text{“intercept”}}\big) \tag{21}$$

where $f_c(h_{u_i}, r_j)$ is the response score and h_{u_i} is the representation of customers. w_g^T and w_i^T are trainable parameters to generate gradient and intercept.

"Gradient" can be used to depict how a customer is sensitive to the incentive. The more sensitive the customer is (i.e., with greater "gradient"), the more uplift gain response model can get. It is reasonable that the group of customers with greater "gradient" may achieve a better uplift gain in terms of commercial objectives than those with less "gradient", under the same treatment of high incentives. We denote $f_c^{r_h}$ as the averaged response score of customers under the treatment of high incentives, and $f_c^{r_l}$ as that under the treatment of low incentives. The uplift gain is defined as $u = f_c^{r_h} - f_c^{r_l}$. The uplift gain of customers who are sensitive to incentives (i.e., with greater "gradient") should be comparatively greater than those with less sensitivity (i.e., smaller "gradient").

To conduct the experiment, all the customers in test data are sorted by the inferred "gradient" in descending order and separated into five groups of different sensitive levels equally. We show the uplift gain of each group in Figure 4. We produce a better uplift gain for the customers of the top sensitive level compared with the DNN-H. For the customers of the bottom sensitive level, the uplift gain produced by our model is relatively smaller compared with the DNN-H, which implies that our model can infer a better representation for those insensitive customers.

Effect of Hazard This set of experiments is conducted to study different response models' effectiveness while using the same GNN layer as representation learning. Table 3 shows the performances of four response models under different evaluation metrics. First, our model outperforms other

Objective	Our Model	DNN-H
Payments	**+2.54%**	0%
	[2.03%, 3.05%]	-
Cost	**-4.04%**	0%
	[-4.84%, -3.24%]	-

Table 4: Relative improvement(%) of our proposed model versus DNN model

Objective	Our Model	SBBM	Cox
Payments	**+2.31%**	+1.47%	0%
	[1.98%, 2.64%]	[1.14%, 1.8%]	-

Table 5: Relative improvement(%) of our proposed method regression with other monotonic methods

response models on customer-side tasks significantly. Our hazard regression can well characterize the flexibility and complexity with a nonlinear function of the response-to-incentive in our setting. Second, our model achieves slightly gains on merchants' results. It indicates that better representation of customers can benefit the task of merchants within the unified optimization. It is in line with our assumption that joint incentive optimization contributes to better customer and merchant representations.

We visualize customers' predicted response scores as we vary different values of incentives in Figure 5. The ground truth is calculated by averaging over the response labels of all customers. The result of each comparison method is calculated by averaging over the predicted response scores. When the amount of incentives increases, Semi-black-box-model fits the ground truth well in small incentive values, but cannot fit well in large values. This is because the samples with small values dominate the sample space, and Semi-black-box-model cannot characterize the data's censoring. Our model fits the ground truth the best under different incentive values. Significantly, there is a noticeable change point in ground truth at incentive value 100 (cent), which means customers are much more sensitive to incentive value larger than 100 than incentive value smaller than 100. Interestingly, the predicted response scores from our model can capture such a change point. Additionally, we show the average hazard rate at each incentive value in the plot. As in our derivation, the response scores are the cumulative function of the hazard rate. The hazard rate plot clearly shows two significant change points at incentive values around 80 percent and 100 percent. The discovery of change points gives us more insights into the design of the values of incentives.

Online A/B Test To show the performance of our approach in real-world scenarios, we conduct two sets of experiments on the mobile payment marketing scenario at Alipay with A/B testing. In the beginning, we conduct A/B testing under 2% traffic lasted for one week. Then we increase the traffic to 30% gradually, and we report the final results observed in the seven days.

The first set of online experiment is conducted to compare our model with DNN-H. We show the results in Table 4. The online experiment shows that our model outperforms DNN-H with a +2.54% relative improvement on payments under 30% traffic flow. We conduct a t-test to show the confidence intervals with 95% confidence level. Besides, our proposed approach significantly saved 4.04% cost while improving the payments. The second set of online experiment is conducted to compare our model with other monotonic methods under the same budget. Comparison results on payments are shown in Table 5. In the online environment, our proposed model outperforms other methods significantly with a +2.31% relative improvement on payments compared to Cox under 30% traffic flow. It shows that the nonlinear hazard rate function in our model can well characterize the flexibility and complexity of the incentives' responses.

Related Work

Marketing has been studied for decades, as it brings significant benefits to improve the efficiency of marketing and many studies focus on this topic. The development of online business realized various data-driven approaches, including forecasting (Beheshti-Kashi et al. 2015) and decision making (Ito and Fujimaki 2017). Geyik et al. (2014) proposed a framework of performance-driven campaign budget allocation across different channels with a specific attribution model as input. Clow (2016) proposed a rule-based method to assign commissions in traditional marketing. Boutilier et al. (2016) formulate the budget allocation as a sequential decision problem and solve it by MDP. Ito et al. (2017) solved price optimization by first forecasting the relationships between sales and prices and then constructing an optimization problem based on those predictive formulas. Staib et al. (2017) study the general budget allocation problem from a robust optimization perspective. Recently Zhao et al. (2019) proposed a general online budget allocation framework consists of two components, including forecasting models and decision making. Our work is related to Ito et al. (2017) and Zhao et al. (2019) that we aim to improve the effectiveness of the response model and solved incentive optimization problems. Our work differs from Ito et al. (2017) and Zhao et al. (2019) in that we apply graph neural networks to build customer-merchant interactions and model the response model of users with hazard regression.

Conclusion

This paper presents our solution for incentive optimization to both customers and merchants in the payment marketing campaign. Our solution consists of three components. First, we learn the representations of customers and merchants based on the intrinsic aggregation operator built inside graph neural networks atop the underlying customer-merchant interactions. Second, to characterize the flexibility and complexity in our setting, we model users' response model with hazard regression using a nonlinear hazard rate function. Third, we formulate the online decision problem as a linear programming problem and derive an approximate online solver. Both offline and online A/B testing results demonstrate its effectiveness with over 2% relative improvement on payments. Our solution has been successfully applied to many scenarios at Alipay.

References

Beheshti-Kashi, S.; Karimi, H. R.; Thoben, K.-D.; Lütjen, M.; and Teucke, M. 2015. A survey on retail sales forecasting and prediction in fashion markets. *Systems Science & Control Engineering* 3(1): 154–161.

Boutilier, C.; and Lu, T. 2016. Budget Allocation using Weakly Coupled, Constrained Markov Decision Processes. In *Conference on Uncertainty in Artificial Intelligence*.

Clow, K. E. 2016. *Integrated advertising, promotion, and marketing communications*. Pearson Education India.

Cox, D. R. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 34(2): 187–202.

Fout, A.; Byrd, J.; Shariat, B.; and Ben-Hur, A. 2017. Protein interface prediction using graph convolutional networks. In *Advances in Neural Information Processing Systems*, 6530–6539.

Geyik, S. C.; Saxena, A.; and Dasdan, A. 2014. Multi-touch attribution based budget allocation in online advertising. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 1–9.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 1024–1034.

Hu, B.; Zhang, Z.; Shi, C.; Zhou, J.; Li, X.; and Qi, Y. 2019. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 946–953.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2333–2338. ACM.

Ito, S.; and Fujimaki, R. 2017. Optimization beyond prediction: Prescriptive price optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1833–1841. ACM.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Liang, C.; Liu, Z.; Liu, B.; Zhou, J.; Li, X.; Yang, S.; and Qi, Y. 2019. Uncovering Insurance Fraud Conspiracy with Network Learning. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1181–1184.

Liu, D. C.; and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45(1-3): 503–528.

Liu, Z.; Chen, C.; Li, L.; Zhou, J.; Li, X.; Song, L.; and Qi, Y. 2019a. Geniepath: Graph neural networks with adaptive receptive paths. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4424–4431.

Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; and Song, L. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2077–2085. ACM.

Liu, Z.; Smola, A.; Soska, K.; Wang, Y.-X.; Zheng, Q.; Zhou, J.; et al. 2017. Attributing hacks with survival trend filtering. *Electronic Journal of Statistics* 11(2): 5311–5341.

Liu, Z.; Wang, D.; Yu, Q.; Zhang, Z.; Shen, Y.; Ma, J.; Zhong, W.; Gu, J.; Zhou, J.; Yang, S.; et al. 2019b. Graph Representation Learning for Merchant Incentive Optimization in Mobile Payment Marketing. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2577–2584.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 593–607. Springer.

Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural networks* 61: 85–117.

Staib, M.; and Jegelka, S. 2017. Robust budget allocation via continuous submodular functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3230–3240. JMLR. org.

Tibshirani, R. J.; et al. 2014. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics* 42(1): 285–323.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* .

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2019. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596* .

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* .

Zhang, D.; Huang, X.; Liu, Z.; Hu, Z.; Song, X.; Ge, Z.; Zhang, Z.; Wang, L.; Zhou, J.; and Qi, Y. 2020. AGL: a Scalable System for Industrial-purpose Graph Machine Learning. *arXiv preprint arXiv:2003.02454* .

Zhao, K.; Hua, J.; Yan, L.; Zhang, Q.; Xu, H.; and Yang, C. 2019. A Unified Framework for Marketing Budget Allocation. *arXiv preprint arXiv:1902.01128* .