Latent Dirichlet Allocation for Internet Price War *

‡,♯Chenchen Li,† ‡,♯Xiang Yan,† ♮Xiaotie Deng, ♯Yuan Qi, ♯Wei Chu,
♯Le Song, ♯Junlong Qiao, ♯Jianshan He, ♯Junwu Xiong
‡Shanghai Jiao Tong University ‡, ♯AI Department, Ant Financial Services Group
‡,♮School of Electronics Engineering and Computer Science, Peking University
lcc1992@sjtu.edu.cn, xyansjtu@163.com, xiaotie@pku.edu.cn,
{yuan.qi, weichu.cw, le.song, junlong.qjl, yebai.hjs, junwu.xjw}@antfin.com

Abstract

Current Internet market makers are facing an intense competitive environment, where personalized price reductions or discounted coupons are provided by their peers to attract more customers. Much investment is spent to catch up with each other's competitors but participants in such a price cut war are often incapable of winning due to their lack of information about others' strategies or customers' preference. We formalize the problem as a stochastic game with imperfect and incomplete information and develop a variant of Latent Dirichlet Allocation (LDA) to infer latent variables under the current market environment, which represents preferences of customers and strategies of competitors. Tests on simulated experiments and an open dataset for real data show that, by subsuming all available market information of the market maker's competitors, our model exhibits a significant improvement for understanding the market environment and finding the best response strategies in the Internet price war. Our work marks the first successful learning method to infer latent information in the environment of price war by the LDA modeling, and sets an example for related competitive applications to follow.

Introduction

Price war commonly refers to a scenario where companies reduce the prices of their products, sometimes even below cost, to attract more customers in market competitions (Krämer, Jung, and Burgartz 2016). It is a classic tactic for traditional entrepreneurs where competitors in the market sell similar products. It has recently become overwhelmingly popular for Internet platform competitions, especially after Uber's great success of conquering the world's ride-hailing market through price reduction in different cities one by one (Ellen Huet 2015).

With the help of big data, Internet companies are able to provide personalized price reductions, for example, they decide bonuses of different values for different cities, or different types of awards (discount coupons, cash back, or free gifts) according to customers' preference. Internet market makers, especially for single product or service commonly referred to as unicorns, are willing to compete in an Internet price war to recruit participants, boost up membership, venture into new frontiers, and ultimately, eliminate competitors. For the sake of convenience, we make it the objective of each of them to maximize the total number of customers who would take the offer and enjoy the goods and services provided, under their given fixed budgets.

In recent years, we have witnessed a significant number of price wars in both traditional industries such as the airlines, retails, crude oil, and emerging markets such as p2p financing, ride-hailing, bicycle sharing, online (and offline) cash back shopping , there has been few studies on how to design the optimal strategy to win any price war. From one company's perspective, simply setting lower prices or providing worthier coupons to customers may not always lead to more consumptions. This is because customers have limited demand for consumption and may have inherent preference for specific company's products. And its opponents may also increase their investments at the same time, resulting in an equal attraction for customers. This means on behalf of a company, providing coupons seems to have no effect on attracting customers, but in fact it will loss some customers if providing nothing to them.

Indeed, entrepreneurs' fighting in an Internet price war can be viewed as playing an imperfect and incomplete information game, to gain an advantage over opponents via financial investment. One may distinguish between games with imperfect and perfect information through whether the opponents' potential strategies are accessible to a player. For example, in chess or go game, each player knows all possible plays his opponent can do at any step, meaning it is a **perfection information game**. On contrast, in card games each player's cards are often hidden from others, thus it is an **imperfect information game**, and so is the Internet price war since participant companies have no information about how their competitors provide personalized price reduction. On the other hand, the win/loss outcome of chess, go game and card games, formally the structure of these games is known to all players after their plays, which means they are **complete information games**. However, in an Internet price war, companies do not know how customers make their choice after receiving awards, thus not able to calculate their util-

*This work was partially supported by the National Nature Science Foundation of China (No. 61632017, 61761146005), and by Ant Financial. Thanks Zhengyang Liu and Kai Li for the advices.

†Equal contributions.

‡Both are the first institutions.

ities accurately even if they know other companies' strategies. Such a lack of customers' preference means it is also an **incomplete information game**.

If we are able to reveal these kinds of missing information, we can find the best strategy for playing such a game, and also obtain a better understanding of the price war. Latent Dirichlet Allocation (LDA) is a powerful tool to learn the latent variables, which have been applied in a lot of fields, such as text processing (Blei, Ng, and Jordan 2003), causal inference (Lauritzen 2001), image classification (Chong, Blei, and Li 2009) and so on. Thus we also consider the LDA model for this scenario. It characterizes the interactions using the observable information about consumptions in one's own company as a variable dependent on customers' preferences, which is in turn also dependent on both its strategy and its competitors' strategies of providing price reduction. Aided by the LDA, we can infer the latent variables to approximately characterize the environment and further seek better strategies through other decision-making algorithms like Deep Reinforcement Learning (DRL). The combined method forms a complete framework to deal with imperfect information scenario, inferring latent variables through LDA first and find better strategies based on transferred perfect information environment.

To show that the inferred information is useful in the part of decision making, we conducted experiments on simulated Internet price war game, using our framework to play against baseline methods. And we also applied our LDA on an open dataset from real business and evaluate the results by comparing prediction likelihood with baselines and distribution distance to the real distribution. All these experiments justify our framework's effectiveness.

Related Works

Price war as a competitive business environment was modeled as an imperfect information game (Ferrero, Rivera, and Shahidehpour 1998). Some researchers focused on how to avoid the war (Krämer, Jung, and Burgartz 2016), while others considered strategies for setting prices for whole market in competitions (Feng, Li, and Li 2014; Wang, Chen, and Wu 2017). None studies micro operating strategies when a price war is inevitable.

In recent years, reinforcement learning (Sutton and Barto 1998) is commonly believed to be useful in exploring strategies in game scenarios with opponents. For example, (He et al. 2016) suggested an opponent modeling method adding to the action set of deep reinforcement learning. And another famous application for imperfect information game is by (Heinrich and Silver 2016), who propose an approach named NFSP to solve the approximated Nash Equilibrium through DRL with fictitious self-play. Their work seeks strategies under partial observed information directly but has no understanding for that unknown information.

On the other hand, exploring hidden information from observed data have been commonly desired in applications of recommend systems (Luo, Shang, and Li 2016), information retrieval (Xuan et al. 2015), statistical natural language processing (Li et al. 2015) and so on. Among them, probabilistic graphical models are widely used since its huge success in classifying topics from contexts (Blei, Ng, and Jordan 2003). Similar to our work, graphical models have been applied on inferring users' preference from user-generated data, such as (Giri et al. 2014) understanding the preference of mobile device user and (Yu et al. 2014) finding buyers' preference on e-commerce search results. But in these works, latent variables are never under the competitive environment, and as far as we know, there is no application that models one's competitor's strategies as a latent variable before this work.

Game Characterization for Internet Price War

In this section, we formalize the Internet price war through a game theoretical characterization. It is the first time, as far as we know, that such an important marketing phenomenon is formalized in a combined form of both macroscopic competition and microcosmic strategies.

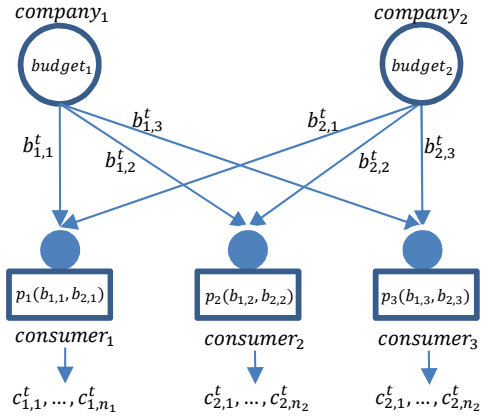


Figure 1: The Internet Price War Game.

Problem Definition

As shown in Fig.1, in an Internet price war, each company (indexed by $i = 1, \ldots, M$) informs each customer in the market (indexed by $j = 1, \ldots, m$) that a personalized award b^t_{ij} ($b^t_{ij} \in \{0, \ldots, B_i\}$) is provided during time period $(t, t+1)$ if the customer purchases its products. W.l.o.g $b^t_{ij} = 0$ means no award. Customer j consumes n^t_j times during the period, for example one week, and let $c^t_{j,k} = i$ if he chooses company i for his k-th consumption. He makes these choices according to his preference function, represented by the probability $p^t_j(\vec{b}^t_j, i)$ he chooses company i for each consumption with respect to received awards $\vec{b}^t_j = (b^t_{1,j}, b^t_{2,j}, \ldots, b^t_{M,j})$.

The objective for each company i is to find the best strategy on providing awards b^t_{ij} to maximize its market share after R time periods, formally

$$\max_{b^t_{ij}} \sum_t \sum_{j,k} \mathbb{I}\{c^t_{j,k} = i\} / \sum_j n^t_j \qquad (1)$$

under budget constraint $\sum_j cost(b^t_{i,j}) \leq budget^t_i$, for $1 \leq t \leq R$. Here $\mathbb{I}\{\cdot\}$ is the 0/1 indicator function.

Corresponding to real Internet price war, each company i only has its own transaction data, i.e. records of customer j who received award b_{ij}^t and purchased company i's products during time period $(t, t+1)$, formally $\{(t, j, b_{ij}^t, c_j^t) | c_{j,k}^t = i\}$. This means (1) company i does not know how its opponents choose awards $b_{i'j}^t$ for $i' \neq i$, so it is playing an imperfect information game; (2) company i does not know how customers decide their consumptions $c_{j,k}^t$, so it is playing an incomplete information game.

Basic Assumptions

In a price war, participants are willing to provide awards for customers mainly because of two important assumptions on customers' behavior patterns.

Assumption 1: In each short time period, say $(t, t+1)$, customers have higher probability to choose one specific company if it offers award of higher value, that is $v_i(b_{i,j}^t) > v_i(b_{i,j}^{'t})$ implies

$$p_j^t(b_{i,j}^t, b_{-i,j}^t, i) > p_j^t(b_{i,j}^{'t}, b_{-i,j}^t, i).$$

Assumption 2: After each time period, say $(t, t+1)$, the preference of customer j on choosing company i without any award tend to his usage rate $u_i = \sum_k \mathbb{I}\{c_{j,k}^t = i\}/n_j^t$ of it, that is

$$(p_j^{t+1}(\vec{0}, i) - p_j^t(\vec{0}, i)) * (u_i - p_j^t(\vec{0}, i)) \geq 0.$$

Such an evolution of customers' preference, and further evolution of related outcome function for all players in the game, make it a **stochastic game**. For the sake of analysis, we assume customers make their decisions $c_{j,k}^t$ at any time t only depend on the award $b_{i,j}^t$ each company offers, but are unrelated to the total number of his consumptions n_j^t in the period $(t, t+1)$, nor to other buyers' choice.

And for companies, since we are considering this problem from one company's perspective, all his competitors can be regarded as one opponent. Meanwhile, as modern marketing always does, companies cluster customers into several groups, each of which contains customers of similar behavior.

Now the process of the Internet price war can be precisely described by Alg. 1.

Algorithm 1: The Process of the Internet Price War

Input: R, m, $budget_i, i \in \{1, 2\}$
Output: The market share of each company, s_1, s_2

Initialize company $i, i \in \{1, 2\}$ and customer j with their private v_j and p_j^0, $j \in \{1, \ldots, m\}$
for $t \leftarrow 1$ **to** R **do**
  for $j \leftarrow 1$ **to** m **do**
    $b_{1,j}^t \leftarrow$ company 1 choose an award for customer j
    $budget_1 \leftarrow budget_1 - cost(b_{1,j}^t)$
    $b_{2,j}^t \leftarrow$ company 2 choose an award for customer j
    $budget_2 \leftarrow budget_2 - cost(b_{2,j}^t)$
    for $k \leftarrow 1$ **to** n_j **do**
      $c_{j,k}^t \leftarrow p_j^t(b_{1,j}^t, b_{2,j}^t)$
    $p_j^{t+1} \leftarrow$ update p_j^t according to c_j^t
$s_1 = \sum_{j,k,c_{j,k}^t=1} 1 / \sum_j n_j^t$
$s_2 = \sum_{j,k,c_{j,k}^t=2} 1 / \sum_j n_j^t$

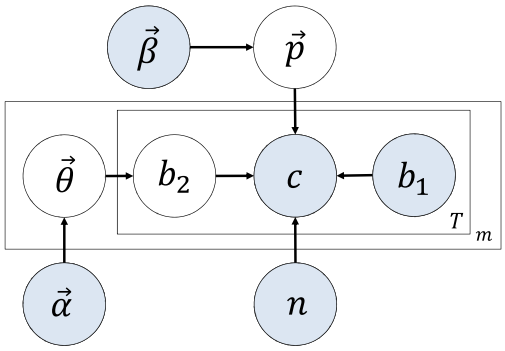


Figure 2: The Internet Price War PGM.

Latent Dirichlet Allocation of Price War Game

We model the process of each customer choosing company 1 to consume, called the Internet Price War LDA, as shown in Fig. 2. We omitted the superscripts about time and subscripts about customers for expressions of all variables.

Price War LDA

In this subsection, we first show the generative process of observed data in the game of price war, then we introduce the details.

- Choose a preference distribution $\vec{p} \sim Dir(\beta)$
- For the each customer j, choose a strategy distribution $\vec{\theta} \sim Dir(\alpha)$
- Customer j decides to consume n_j times
- Company 2 chooses an award $b_2 \sim \vec{\theta}$
- Company 1's choice of the award b_1 is known
- For each consumption $c_{k,j}$, $k = 1, 2, \ldots, n_j$,
  - customer j chooses the company $c_{k,j} \sim \vec{p}(b_1, b_2)$
- Company 1 observes that customer j has c consumptions, where $c = \sum_{k, c_{k,j}=1} 1$

At the beginning of each time period, company 1 decides to provide customer j with award b_1, while his opponent company 2 provide b_2. Company 2 decide b_2 according to some strategy $\vec{\theta}$, representing the probabilistic distribution of all possible awards, but the exact award b_2 and the distribution $\vec{\theta}$ is unknown to company 1. Meanwhile, customer j's preference function is simplified as $\vec{p} = p(b_1, b_2)$ as the probabilistic distribution on choosing company 1 to consume with respect to all possible awards pair (b_1, b_2). And in this period, the customer plans to consume n times in total,

which is subject to a distribution of $\vec{n}$. For each consumption, he chooses one specific company according to the preference distribution $\vec{p}$ along with actually received awards (b_1, b_2), thus company 1's observed records c consist of his all consumptions on it in the period. Since we focus on the probability that customer j will choose company 1, we consider the $\frac{c}{n}$ as observed data. For $\frac{c}{n}$ is in [0, 1], we define a function $g(\cdot) = \llcorner \frac{c}{n} * acc \lrcorner$ to discretize their value into a new range according to required accuracy acc. It is noticeable that we figure out the the distribution of $\vec{n}$ by statistics in advance, rather than inferring it by LDA. When we infer the latent variables, we sample n till $c \leq n$ in order to avoid that $n < c$.

Without loss of generality, we assume the hidden variables $\vec{b_2}$ and $\vec{p}$ is from two Dirichlet distribution $Dir(\vec{\alpha})$ and $Dir(\vec{\beta})$. We define $\vec{p}$ as the multinomial distribution on the $\{0, acc-1\}$ with size of $|B_1| * |B_2|$, where $|B_1|$ is the number of awards company 1 provides and $|B_2|$ is the number of awards we assume the opponent offers. And we define $\vec{b_2}$ as the multinomial distribution on the $\{0, \ldots, |B_2|\}$. And on behalf of company 1, we assume that company 2 is using the same strategy $\vec{\theta}$ for a specific customer in recent several periods of time, say T. Meanwhile, we assume company 1 has clustered customers into groups, so that customers in each group have the same preference functions. Thus company 1 could use records for each group of customers in the normalized form $(j, t, b_1, g(\frac{c}{n}))$, where $j \in \{1, 2, \ldots, m\}$ and $t \in \{1, 2, \ldots, T\}$. Then it is able to get approximations for the distribution of his opponent's strategy $\vec{\theta}$ for each customer j and the preference function $\vec{p}$ for these group of customers by solving the Price War LDA.

Inference

We use the Gibbs Sampling method to solve our LDA. The joint probability of the opponent's bonus $\vec{b_2}$ and count $\vec{c}$ can be factored into the following:

$$Pr(\vec{c}, \vec{b_2} | \vec{b_1}, \vec{n}, \alpha, \beta) = Pr(\vec{c} | \vec{b_1}, \vec{b_2}, \vec{n}, \beta) Pr(\vec{b_2} | \alpha)$$

Gibbs sampling will sequentially sample each variable of interest from the distribution over that variable given the current values of other variables and the data.

According to Gibbs Sampling, and letting the subscript $-i$ denote the statistic value for an variable without the i-th sample, the conditional posterior for p_i and $b_{2,i}$ is

$$\begin{aligned} &Pr(b_{2,j,i} = k | \vec{b}_{2,-i}, \vec{p}_{-i}, \vec{c}_i, \alpha, \beta, b_1, n) \\ &\propto \frac{\{N^{(h)}_{b_1,k}\}_{-i} + \beta}{\{N_{b_1,k}\}_{-i} + acc\beta)} * \frac{\{N^{(k)}_j\}_{-i} + \alpha}{\{N_j\}_{-i} + |B_2|\alpha)} \end{aligned} \quad (2)$$

Here $N^{(h)}_{b_1,k}$ is the number of times $g(\frac{c}{n}) = h$ when given (b_1, k) and $N_{b_1,k}$ is the total number of records when given (b_1, k). $N^{(k)}_j$ is the number of times customer j receives k from company 2 and N_j is the total number of consumptions of customer j.

Post-processing

It is worth noting when we get $\vec{p}_{j_1}(b_{1,j_1}, b_{2,j_1})$ and $\vec{p}_{j_2}(b_{1,j_2}, b_{2,j_2})$ via different records of customer j_1 and customer j_2, they do not represent the distributions of the same pair of awards if $(b_{1,j_1}, b_{2,j_1}) = (b_{1,j_2}, b_{2,j_2})$. The reason is that for each customer we do not assign exact awards offered by opponents when inferring, but identification numbers (ids) to represent them. This means the ids may indicate different awards for different customers. In order to avoid the situation, we assume that the opponent has $|B_2|$ kinds of awards, where $v_j(x_1) < v_j(x_2)$ if $x_1, x_2 \in B_2$ and $x_1 < x_2$. According to Assumption 2 in Section 2.2, the expected consumptions of customer j on company 1 when $b_2 = x_1$ should be larger than the one when $b_2 = x_2$ if $x_1 < x_2$. Thus we sort the inferred preference distribution $\vec{p}$ accordingly when b_2 is fixed, to get all $\vec{p}$ in the same order.

Simulation Experiments

In this section, we introduce the experiments on the simulation framework to show that the distributions learned from our LDA is useful for awards decision to achieve more market share.

Firstly, we build an simulation environment for the Internet Price War, where two companies compete for a plenty of customers by providing awards. Such a game is played repeatedly for finite or infinite time periods while customers' preference functions are evolving along with time according to Assumption 1&2 in Section 2.2. This simulation framework is summarized in Algorithm. 1.

Then we introduce some methods able to utilize inferred information from our LDA, such as Deep Reinforcement Learning (DRL) and Dynamic Programming (DP). In our experiments, Company 1 in the simulation environment uses these methods to play against Company 2 using other baseline methods.

Finally the results of market share for Company 1 in these experiments show the significance of our LDA model.

Framework of Simulation Environment

We design an framework to simulate how customers act after receiving awards from two companies in an Internet war, motivated by Sethi and Somanathan (Sethi and Somanathan 2001). The simulation environment mainly consists of two parts, the form of customers' preference functions at each time period, and how preference evolves along with time.

Preference Function: Here we focus on the situation when customer j receiving awards b^t_1 and b^t_2 from company 1 and company 2 respectively. At time $t = 0$ a customer has an initial preference distribution $p^0_j(b_1, b_2, 1)$ on choosing company 1, dependent on the difference $d = v_j(b_1) - v_j(b_2)$ between the value of awards he receives from both companies. The preference for choosing company 2 is naturally $1 - p^0_j(b_1, b_2, 1)$ and we do not mention it specifically in the followings. We define $v_j(x) = cost(x) = x$ in our simulated experiments, and the notation for $p^t_j(b_1, b_2, 1)$ can be simplified as $p^t_j(d)$. The preference distribution takes the same form as a Sigmoid function except its mean value modified to customer j's inherent preference for choosing com-

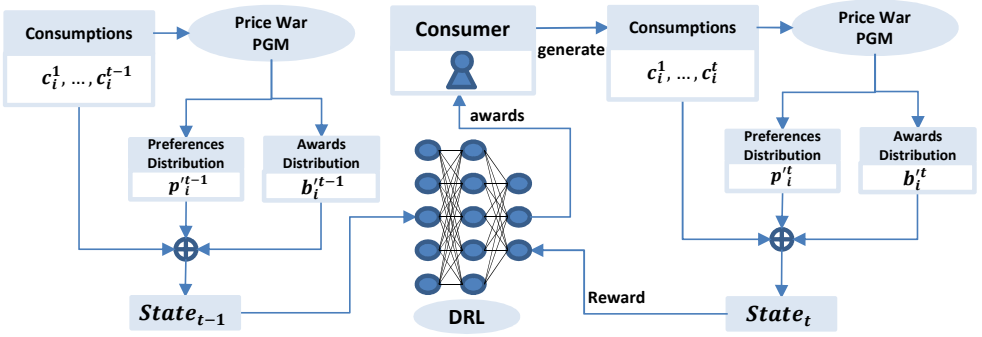


Figure 3: The DRL framework for Price War with the information learned from the Price War LDA

pany 1 when no award is provided. That is letting $p_j^0(d) = Sig(d, \sigma)$ for $\forall d$ where

$$Sig(d,\sigma) = \begin{cases} \frac{\sigma}{0.5} \times (\frac{1}{1+e^{-d}} - 0.5) + \sigma & if\ d < 0 \\ \frac{1-\sigma}{0.5} \times (\frac{1}{1+e^{-d}} - 0.5) + \sigma & if\ d > 0 \end{cases} \tag{3}$$

and $\sigma = p_j^t(0)$. And whenever σ is determined, the whole function can be determined. We choose the preference function in this form because (1) it increases monotonously as the value difference between awards from two companies increases, corresponding to Assumption 1 in Section 2.2; (2) it accords with the property of diminishing marginal returns.

Updating Process: During the period $(t, t+1)$, customer j consumes for n_j^t times, each of which is independently subject to the preference distribution $p_j^t(d^t)$, where $d^t = v_j(b_1^t) - v_j(b_2^t)$ is the value difference of his actual received awards. After that, we can calculate the usage rate $u_{i,j}^t$. According to Assumption 2 in Section 2.2, we let the updating formula to be:

$$p_j^{t+1}(0) = (u_{i,j}^t - p_j^t(0)) * \gamma + p_j^t(0) \tag{4}$$

, where γ is a parameter reflecting how sensitive the customer is to the awards, called updating rate. Then the whole preference distribution can be calculated accordingly as $p_j^{t+1}(d) = Sig(d, p_j^{t+1}(0))$ for $\forall d$.

Some Methods Can Utilize the Information

Deep Reinforcement Learning (DRL) Deep Reinforcement Learning is a flexible framework for Markov Decision Process. The input of DRL only requires a fixed-length vector, which usually represents the state of the observed environment. Thus we directly combine the preference distributions and strategy distributions with the raw features vectors. DRL also pays attention to model the transitions between different states, which may be a good model for the evolution of customers' preferences and the transformation of the opponents' strategies. It is also a framework of optimization, thus we do not need other extra operations. Thus, we design a DRL framework to utilize the information of LDA:

State: s_j^t contains three parts, consumptions history h_j^t of customer j before time t, preference distribution $p_j'^{t-1}$ and award distribution $b_j'^{t-1}$ of the opponents learning from h_j^t, which are the approximation of p_j^{t-1} and b_j^{t-1}. As the preference and award of opponent may change little in a short period, i.e, $(t-1, t)$, we can consider the $p_j^{t-1} \approx p_j^t$ and $b_j^{t-1} \approx b_j^t$. Therefore, we add the preference and opponents' award of time $t-1$ into state s_j^t. In this paper, we simply concatenate three parts, that

$$s_j^t = \begin{pmatrix} h_j^t \\ p_j'^{t-1} \\ b_j'^{t-1} \end{pmatrix} \tag{5}$$

The transition is from s_j^{t-1} to s_j^t for each state.

Action: a is naturally the award b_j^t we choose for customer j at time t is in $\{0, .., |B_1|\}$, where B_1 is the set of actions predefined. In our deep reinforcement learning, the *Action* only consists of all the possible value of awards in an interval pre-announced by a company. And for the convenience of experiments, we further discretize those value.

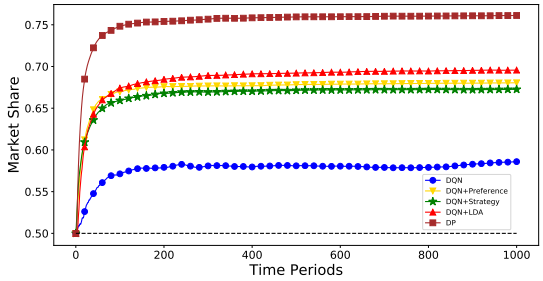
Reward: In a price war, when a company provide the award b_j^t to a customer represented by s_j^t, the number of consumptions he chooses the company is a nature reward $R(s_j^t, b_j^t)$. But in real marketing, such feedback should also include a factor of cost as a negative part, since companies have limit budgets. As a result, $R(s_j^t, b_j^t) = c - \xi * cost(b_j^t)$, where c represents the number of consumptions and ξ is the parameter to control the weight of two parts. The reason why a company's remainder budget are not included in *State* is because the company cannot be sure how many customers it will capture immediately after providing the award. On contrast, the average cost of attracting a customer matters more than the total money spends in the end.

Framework: Fig. 3 show the overall framework. We adopt the Deep-Q-Network (Mnih et al. 2015) as the version of DRL. The inputs of DQN are itemized above. The optimization process can be defined as

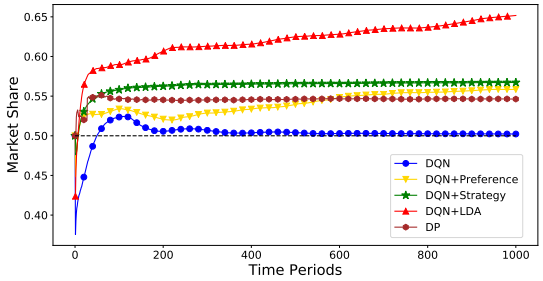
$$Q^t(s_j^t, a) = (1-\alpha)Q^t(s_j^t, a) + \alpha(R(s_j^t, a) + \lambda \max_a Q(s_j^{t+1}, a)) \tag{6}$$

where α is the learning rate and λ is the discount factor.

Dynamic Programming (DP) Since we learn the preference distributions and strategy distributions, we can do optimizations directly according to these kinds of information.



(a) Market share curves when playing against Random Strategy.



(b) Market share curves when playing against DQN.

Figure 4: Market share curves in simulation experiments.

In precise, we define $\mathrm{f}(i,k)$ as the maximum market shares we can get when we finish offering awards to the first i customers costing k budgets. Then we take advantage of Dynamic Programming (DP) to learn the optimal result of $\mathrm{f}(m, budget)$ in every single round. Finally, we choose the awards corresponding to the optimal solution for each customer as our policy.

Formally, the transition equation for solving $\mathrm{f}(i,k)$ is

$$\mathrm{f}(i,k) = \max_{1 \le j \le |B|} (\mathrm{f}(i-1, k-j) + \psi(i,j)) \tag{7}$$

$\psi(i,j)$ is expected benefit from customer i if we offer award j to him, which is calculated by

$$\psi(i,l) = \sum_{l=0}^{|B|} b_i(j) p_i(j,l) \tag{8}$$

where $b_i(j)$ is the probability that the opponent choose awards j for customer i, and $p_i(j,h)$ is the probability that the customer i choose our company if it received j from us and l from the opponent. And we choose the award j^* that maximizes Eqn. (7) for the i-th customer.

Other Baseline Methods

To evaluate our model, we conduct a series of simulation experiments. In the experiments, company 1 uses the DRL or DP as introduced before, to play against company 2 using one of following baseline methods:

- **Random Strategy (Random)** randomly chooses one of possible awards for each customer with equal probability.
- **Deep-Q-Network(DQN)** (Mnih et al. 2015) is a version of DRL. Note that the settings of this DQN are exactly the same as the ones mentioned in previous subsection except its state does not include features about the customer's preference and opponent's strategy.

Other Experimental Settings

- **Simulated Environment**: In the simulated environment, there are 10 kinds of customers at all, each of which has 1000 persons. The initial σ = 0.5, updating rate γ = 0.5. $n_j^t \in [1, 100]$ for $\forall j, t$ There are two companies in the markets at all. Each company has 5 kinds of awards, $B_1 = B_2 = \{0, 1, 2, 3, 4\}$, with the same amount of budgets, $budget_1 = budget_2 = 20000$.
- **Learning Methods**: The network of adopted DQN has 3 layers, the sizes of which are N_{input}, 512, 5, where N_{input} is the size of input features. The reward function is $R(s_j^t, b_j^t) = c_j^t - 0.5 * cost(b_j^t)$ and its decay rate is 0.9. The learning rate is 0.01 and memory size is 200000.
- **Variants**: Here since the approximation solution to LDA are two sets of variables, representing customers' preference and opponent's strategy, we do experiments of adding these two features to DQN's states separately and together, and they are referred as "DQN + P", "DQN + S" and "DQN + LDA" respectively. And the DP introduced before requires both features, it is simply referred as 'DP'.

Results

We list final market shares of company 1 after 1000 rounds in Table 1. It uses variants of our methods (DQN, DQN+P, DQN+S, DQN+LDA and DP), playing against company 2 using Random Strategy or DQN. The market share is the average value taken from 10 repeated experiments.

	DQN	DQN + P	DQN + S	DQN + LDA	DP
Random	58,59%	68.05%	67.26%	69.57%	**76.12%**
DQN	50.22%	55.84%	56.72%	**65.16%**	54.63%

Table 1: Comparison of market share. The number in the i-th row is the market share of company 1 when company 1 uses the i-th method in the first row playing against company 2 using the j-th method in the first column.

Generally speaking, our methods get market shares over 50% when competing with Random Strategy and DQN, which do not include specific information about customers' preference and opponent's strategy. This means that the inferred latent variables from the Price War LDA, either separately or joint together, are helpful to characterize the environment of an Internet price war.

Meanwhile, DP shows the best result when playing against Random Strategy, while DQN + LDA performs best against DQN. This coincides with common sense as Random Strategy is not evolving along with time, which means

DP can learn the optimal solution with respect to known information. When the opponent is using a complicated method like DQN, DQN + LDA is the most effective method because it models both the transition of the evolving environment and inferred information.

And Fig.4 shows the average market share of company 1 after t time periods, when using different strategies competing against company 2 using baseline methods. We can find that the convergence procedures in the Fig. 4 (a) is faster and more stable than the ones in the Fig. 4 (b). The reason is that Random Strategy can be considered as the static environment, while DQN is evolving along with the rounds. This is in line with the intuition.

Real-World Dataset Analysis

In this section, we apply our model on a real-world open dataset and conduct a series of experiments, to prove that the model can indeed infer our desired latent information.

Coupon Usage Data for O2O

Coupon Usage Data for O2O, referred to *O2O Dataset* in following description, is an open dataset from the Tianchi contest (Aliyun.com 2018). O2O represents "online to offline", while a typical example of "O2O marketing" is that merchants in a shopping mall send coupons to potential customers through emails or short messages in their own APPs. Merchants want to attract customers to their offline shops and decide these personalized discount rate of coupons based on a large amount of users' behavior and location information recorded by various APPs.

In our experiment, we make use of the offline training data from *O2O Dataset*, where the coupon promotion is conducted by 7737 retail stores from Jan. 1st 2016 to June 30th 2016. Merchant with id 3381 who has the most records in the dataset is chosen to be Company 1 in our model. All other merchants are considered together as its opponent, in other word, Company 2 in our model. There are 74823 records related to company 1, among which three groups of coupons, namely coupons of level low, middle and high according to their discount rate (denoted by 1,2,3 respectively), are provided to 64152 users.

These users are clustered into 4 different preference groups based on features only related to the merchant and users themselves, meaning each group has similar preference distribution. And users in each preference group are further clustered into 10 different subgroups, named strategy groups, based on features only related to themselves, representing that the opponent adopts the same strategy distribution for users in each strategy group. Then we can infer the preference distribution and strategy distribution for each preference group accordingly, by treating records of each strategy group as records of one specific customer as we do in LDA model.

Evaluation

Our model is shown to be effective from two aspects. And in the experiments, original dataset is randomly split into training dataset and testing dataset with ratio 9:1.

Behavior Prediction We first train our model on training dataset, then use inferred distributions to predict behaviors of users in testing dataset. Our model is then evaluated by measuring Negative Log Likelihood of the predictions, compared with baselines.

Negative Log likelihood is a common metric for evaluating the performance of probabilistic models (Blei, Ng, and Jordan 2003), and can also evaluate the performance of supervised model with probabilistic outputs. Mathematically, negative log likelihood is defined as

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} log(p(y_i|\theta, x_i))$$

Here θ is the model to be evaluate, N is the number of samples, x_i is the features, y_i is the ground truth of sample i, and $p(y_i|\theta, x_i)$ is the output probability of y_i from model θ when given x_i. The smaller the likelihood of prediction is, the better the corresponding model is.

We consider 5 common probabilistic prediction models as baselines: Naive Bayesian (NB), Logistic Regression (LR), Support Vector Machine (SVM), Gradient Boosted Decision Tree (GBDT), and Neural Network (NN). All of the above baselines are implemented by sklearn (Pedregosa et al. 2011). They take the same extracted features used by LDA including the cluster of each customer and coupon ids as their features, and whether customers use the corresponding coupons as labels. And we take the average results from 5 fold cross-validation for our model as well the baselines.

Model	NB	LR	SVM	GBDT	NN	LDA
Result	494.93	580.26	1085.40	597.93	509.26	**401.97**

Table 2: Comparison of Negative Log Likelihood

As shown in Tab.2, our model get the smallest negative log likelihood in prediction, meaning that it provides the best modeling for the real-world data.

Distance of Strategy Distributions We also evaluate the distribution distance between our strategy distribution and the real strategy distribution. The real strategy distribution for each strategy group is calculated by the number of coupons that all other merchants in the whole dataset provide to users in the group. Similar to the preprocessing, these coupons are also divided into three groups as level low, middle and high according to their discount rate. We adopt the Wasserstein Distance (Ramdas, Trillos, and Cuturi 2017) to measure the distance of two distribution, which is defined as

$$W_1(\vec{p}, \vec{q}) = \inf_{\pi \in \Gamma(\vec{p}, \vec{q})} \int_{\mathbb{R}\times\mathbb{R}} |x - y| \, \mathrm{d}\, \pi(x, y)$$

, where $\Gamma(\vec{p}, \vec{q})$ denotes the collection of all joint distributions on $\mathbb{R} \times \mathbb{R}$ whose marginals are $\vec{p}$ and $\vec{q}$ on the first and second factors respectively.

We consider two distributions as our baselines.

- The overall distribution of received coupons. We count the total number of each kind of coupons that all other merchants in the whole dataset provide to all users of company 1 as the baseline distribution. It can be regarded as

the average strategy distribution for all strategy groups, thus we denote it 'Average'.

- Uniform distribution: $p = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, which is what a single merchant may assume for its opponent without knowing further information, denoted by 'Uniform'.

Table 3 shows the average Wasserstein Distance between the real strategy distributions and our inferred strategy distributions (denoted by 'LDA') among all strategy groups of all preference groups, comparing with the distance between 'Uniform' or 'Average' distribution and the real distributions. As we can see, the distance between our inferred distribution and the real distribution is the closest.

Model	Uniform	Average	LDA
Result	0.18794	0.13105	**0.12303**

Table 3: Comparison of Distribution Distance

Conclusions

In this paper, we formalize the Internet price war as an imperfect and incomplete information game. We design an LDA model to explore unknown variables from one participant's perspective. The inferred information is shown to help decision making method, like DRL and DP, for finding better strategies in simulated experiments. And the model also exhibits better characterization for an open dataset from a practical business. It is the first time that LDA is used in a game scenario and makes efforts in the competitive business environment. This design not only makes a major contribution towards achieving better market sharing in an Internet price war but also inspire a novel technique for dealing with incomplete and imperfect information games.

References

Aliyun.com. 2018. Coupon usage data for o2o. https://tianchi.alibabacloud.com/datalab/dataSet.html?dataId=59. [Online; accessed 31-May-2018].

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022.

Chong, W.; Blei, D.; and Li, F.-F. 2009. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1903–1910. IEEE.

Ellen Huet, L. C. 2015. World war uber: Why the ride-hailing giant can't conquer the planet (yet). https://www.forbes.com.

Feng, Y.; Li, B.; and Li, B. 2014. Price competition in an oligopoly market with multiple iaas cloud providers. *IEEE Transactions on Computers* 63(1):59–73.

Ferrero, R.; Rivera, J.; and Shahidehpour, S. 1998. Application of games with incomplete information for pricing electricity in deregulated power pools. *IEEE Transactions on Power Systems* 13(1):184–189.

Giri, R.; Choi, H.; Hoo, K. S.; and Rao, B. D. 2014. User behavior modeling in a cellular network using latent dirichlet allocation. In *International Conference on Intelligent Data Engineering and Automated Learning*, 36–44. Springer.

He, H.; Boyd-Graber, J.; Kwok, K.; and Daumé III, H. 2016. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, 1804–1813.

Heinrich, J., and Silver, D. 2016. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.

Krämer, A.; Jung, M.; and Burgartz, T. 2016. A small step from price competition to price war: understanding causes, effects and possible countermeasures. *International Business Research* 9(3):1.

Lauritzen, S. L. 2001. Causal inference from graphical models. *Complex stochastic systems* 63–107.

Li, H.; Lin, R.; Hong, R.; and Ge, Y. 2015. Generative models for mining latent aspects and their ratings from short reviews. In *Data Mining (ICDM), 2015 IEEE International Conference on*, 241–250. IEEE.

Luo, X.; Shang, M.; and Li, S. 2016. Efficient extraction of non-negative latent factors from high-dimensional and sparse matrices in industrial applications. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, 311–319. IEEE.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research* 12(Oct):2825–2830.

Ramdas, A.; Trillos, N. G.; and Cuturi, M. 2017. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy* 19(2):47.

Sethi, R., and Somanathan, E. 2001. Preference evolution and reciprocity. *Journal of economic theory* 97(2):273–297.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press Cambridge.

Wang, S.; Chen, H.; and Wu, D. 2017. Regulating platform competition in two-sided markets under the o2o era. *International Journal of Production Economics*.

Xuan, J.; Lu, J.; Zhang, G.; Da Xu, R. Y.; and Luo, X. 2015. Infinite author topic model based on mixed gamma-negative binomial process. In *Data Mining (ICDM), 2015 IEEE International Conference on*, 489–498. IEEE.

Yu, J.; Mohan, S.; Putthividhya, D. P.; and Wong, W.-K. 2014. Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Proceedings of the 7th ACM international conference on Web search and data mining*, 463–472. ACM.