The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Robust Online Matching with User Arrival Distribution Drift

Yu-Hang Zhou, Chen Liang, Nan Li, Cheng Yang, Shenghuo Zhu, Rong Jin
Alibaba Group, Hangzhou, China
{zyh174606,liangchen.lc,nanli.ln,charis.yangc,shenghuo.zhu,jinrong.jr}@alibaba-inc.com

Abstract

Recently, online matching problems have attracted much attention due to its emerging applications in internet advertising. Most existing online matching methods have adopted either adversarial or stochastic user arrival assumption, while on both of them significant limitation exists. The adversarial model does not exploit existing knowledge of the user sequence, and thus can be pessimistic in practice. On other hands, the stochastic model assumes that users are drawn from a stationary distribution, which may not be true in real applications. In this paper, we consider a novel user arrival model where users are drawn from drifting distribution, which is a hybrid case between the adversarial and stochastic model, and propose a new approach RDLA to deal with such assumption. Instead of maximizing empirical total revenues on the revealed users, RDLA leverages distributionally robust optimization techniques to learn dual variables via a worst-case consideration over an ambiguity set on the underlying user distribution. Experiments on a real-world dataset exhibit the superiority of our approach.

Introduction

Matching is a classic problem with a long history, which is defined on graphs originally (Berge 1957; Gale and Shapley 1962). Given a bipartite graph $G(U, V, E)$, a matching is a set of edges $M \subset E$ such that for every $v \in V$ there is at most one edge in M incident on v. Matching problems are widely relevant to many applications, e.g., resource allocation, stable matching, network routing and so on. For ad allocation problems, the vertices in U and V represent bidders and users respectively, thus each edge can be seen as an allocation from bidder to user with a predefined revenue. Since there are supply and demand constraints on both bidders and users, the goal is to find a matching with maximal revenues satisfying the constraints. However, in real applications, it is usual to see that the matching problem is inherently online (Mehta 2013). In other words, the users often arrive incrementally, and we have to allocate a bidder to the incoming user without complete information about the subsequent users. An optimal online matching algorithm can reach the same objective value as its offline version, which is difficult to achieve because the allocation results are irrevocable while the user arrival information is unknown.

To make the online problem more resoluble, researchers introduce some assumptions regarding the user arrival fashion. In general, there are two common user arrival models, i.e., adversarial model and stochastic model. They differ from each other in how much information the algorithm knows in advance about the user distribution. The adversarial user arrival model assumes that there is an adversary who knows the strategy of the algorithm and generates the worst user sequence for the algorithm, which is usually too pessimistic in practice (Aggarwal et al. 2011; Kalyanasundaram and Pruhs 2000). The stochastic user arrival model, by contrast, assumes that the users are drawn from a stationary distribution or arrive at a random order (Devanur and Hayes 2009; Feldman et al. 2010; Vee, Vassilvitskii, and Shanmugasundaram 2010). However, this assumption can be too optimistic in real applications, because a stationary distribution might not capture the real user arrival fashion well. For example, in an ad allocation task, the user crowd on weekends could be rather different from that on weekdays because the former contains more office workers. When the user distribution is changing, the algorithms with stochastic assumption could suffer severely degenerated performance.

In order to design more practical algorithms, we consider a hybrid case between the adversarial and stochastic assumption in which users are drawn from a drifting distribution, and propose the Robust Dynamic Learning Algorithm (RDLA) to address such issue. Specifically, we assume that there is a drift between the user distributions of adjacent periods, which is much more realistic in real tasks. Rather than maximizing the empirical total revenues on the revealed users, RDLA leverages distributionally robust optimization techniques (Goh and Sim 2010; Wiesemann, Kuhn, and Sim 2014) to learn dual variables via a worst-case approach over an uncertainty set on the underlying user distribution. Furthermore, we present a dynamic learning framework to tackle user arrival distribution drifts by updating dual variables at equal time intervals. Experiments on a real-world dataset exhibit the impressive performance of RDLA.

The rest of this paper is organized as follows. Section 2 introduces related works and Section 3 gives some preliminaries. Section 4 proposes the RDLA approach and Section 5 reports our experiments. Section 6 concludes the paper.

Related Works

Online Matching In the past decades, many efforts have been devoted to online matching problems (Mehta 2013) due to the increasing demand of internet advertising applications. In general, there are two kinds of different assumptions on user arrival models, i.e., adversarial model and stochastic model, thus existing online matching methods fall naturally into two categories accordingly.

The *adversarial user arrival model* assumes that there is an adversary who knows the strategy of the algorithm and generates the worst user sequence for the algorithm, which is reasonable when we have no knowledge of the user sequences. It has been proved that the optimal competitive ratio of algorithms designed for the adversarial setting is $1 - 1/e$ (Karp, Vazirani, and Vazirani 1990), and several algorithms, e.g., Perturbed Greedy (Aggarwal et al. 2011) and Balance Algorithm (Kalyanasundaram and Pruhs 2000) have been proposed to achieve the optimal value. However, in reality we often have some priors about the user sequence, leading to the fact that the adversarial user arrival model is usually too pessimistic. It is shown that even under some mild assumptions such as the random order setting, the competitive ratio of the adversarial algorithms can only be improved to 0.76, which is far from the near-optimal performance (Mirrokni, Gharan, and Zadimoghaddam 2012).

The *stochastic user arrival model* assumes that users are drawn from a stationary distribution or arrive at a random order. The online primal-dual scheme is a classic framework with the stochastic assumption, in which dual variables are learned by solving a fractional online matching problem on the revealed users and used for the allocation of subsequent online users. This approach was first presented for AdWords problem (Buchbinder and Naor 2009; Devanur and Hayes 2009), and then extended to other general matching problems (Feldman et al. 2010; Vee, Vassilvitskii, and Shanmugasundaram 2010). For the stochastic user arrival model, these algorithms can achieve a near-optimal competitive ratio of $1 - \epsilon$. Moreover, under the assumption of random order of arrival users, (Agrawal, Wang, and Ye 2014) proposed a Dynamic Learning Algorithm (DLA) that updates dual variables at geometric time intervals and achieves a competitive ratio of $1 - O(\sqrt{M \log T/B})$ where M and T are the numbers of bidders and users respectively and B is the minimum budget among all bidders.

Though the best algorithms for the stochastic setting are near-optimal when users arrive stochastically as expected, it is notable that they could have a degenerated competitive ratio close to zero when the real users are adversarial selected (Esfandiari, Korula, and Mirrokni 2015). In other words, algorithms designed for stochastic setting are not as robust as that for the adversarial setting. Recently, to capture the presence of user traffic spike caused by unpredictable events such as breaking news, (Esfandiari, Korula, and Mirrokni 2015) proposed a robust online stochastic model to capture the nature of traffic spikes in online advertising, but it is still different from our consideration that the user arrival distribution is drifting all the time.

Distributionally Robust Optimization (DRO) Optimization models are used in statistical learning and other decision making problems, where there are some parameters to be specificated or estimated. It is known that the optimal solution of an optimization model depends heavily on these uncertain parameters. However, due to the slackness of data and other useful information, it is difficult to estimate these parameters precisely. To cope with such issues, *Robust Optimization* approaches have been proposed aiming at finding an optimal solution that is tolerant to the ambiguous parameters (Ben-Tal, El Ghaoui, and Nemirovski 2009; Bertsimas, Brown, and Caramanis 2011).

It is typical to model the ambiguity by using an uncertainty set of parameters and to optimize with the worst case of the parameters in this set (Ben-Tal and Nemirovski 1998; Ghaoui, Oustry, and Lebret 1998). However, such approaches may ignore the stochastic nature of parameters. Therefore, *Distributionally Robust Optimization (DRO)* was proposed, which introduces a distribution P on the parameter set and models the uncertainty by introducing ambiguity set specified by distribution P. Finally, we aim at optimizing the worst case of the distribution in the set (Goh and Sim 2010; Wiesemann, Kuhn, and Sim 2014).

Now the key question is how to choose the ambiguity distribution set. Many approaches were proposed to work with an ambiguity set constructed by the moments of the distribution (Delage and Ye 2010; Goh and Sim 2010). In addition, a statistical estimation of the distribution called *nominal distribution* can be available, containing valuable knowledge of the underlying distribution. We restrict the ambiguity set such that the distribution in the set is within a certain distance from the nominal distribution. Different distances such as KL divergence (Hu and Hong 2013), Wasserstein metric (Esfahani and Kuhn 2018; Gao and Kleywegt 2017) and ϕ-divergence (Namkoong and Duchi 2016; Namkoong and Duchi 2017) has been considered. Recently, (Namkoong and Duchi 2017) builds off of techniques for DRO and empirical risk minimization.

Preliminaries

Online Matching Problem

To make notation more precise, we take the ad allocation task as an example of matching problem. Suppose that there are M bidders and T users to take part in the bidding. When allocating the i-th bidder to the t-th user, the bidder wins a revenue of R_{it} and pays a cost of c_i. The total costs paid by the i-th bidder can not exceed its predetermined budget b_i. The goal is to find an optimal matching with maximal total revenues satisfying all budget constraints, which can be formulated as,

$$\begin{aligned} \max_{X \in \Omega} \ & \sum_{t=1}^{T} \sum_{i=1}^{M} R_{it} X_{it} \\ \text{s.t.} \ & \sum_{t=1}^{T} c_i X_{it} \le b_i, \ i \in [M] \end{aligned} \tag{1}$$

where $\Omega = \{X \in \mathcal{R}_+^{M \times T} : \sum_{i=1}^{M} X_{it} = 1, \forall t \in [T]\}$. $X_{it} \in \{0, 1\}$ is the decision variable indicating whether the algorithm allocates the i-th bidder to the t-th user. Note that $\sum_{i=1}^{M} X_{it} = 1$, meaning that only one bidder can be allocated to the t-th user.

It is easy to see (1) is a special case of linear programming, and can be solved efficiently in polynomial time. However, in reality the users arrive in a sequence, which means that $R_{:t}$ is revealed incrementally. Therefore it is of great importance to develop online matching algorithms making sequential decisions $X_{:t}$'s such that the final total revenues $\sum_{t=1}^{T}\sum_{i=1}^{M} R_{it}X_{it}$ is maximized. Specifically, given the previous decisions $\{X_{:j}\}_{j=1}^{t-1}$ and users $\{R_{:j}\}_{j=1}^{t}$, the t-th variable $X_{:t}$ has to be decided, subject to the budget constraints $\sum_{j=1}^{t} c_i X_{ij} \le b_i, i \in [M]$. We can measure an online matching algorithm by *Competitive Ratio*.

Defination 1. *Competitive Ratio (C.R.)*

$$\textit{C.R.} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{M} R_{it}X_{it}}{\textit{OPT}} \tag{2}$$

where OPT is the optimal objective value for the corresponding offline problem.

Remark 1. For offline algorithms, all users are revealed in advance, thus OPT is an upper bound of the optimal total revenues for online algorithms. When C.R. of an online algorithm approaches to 1, we call it a *near-optimal* algorithm.

The Primal-Dual Framework

The Online Primal-Dual is a classic framework working for the stochastic user arrival model. Suppose the total number of users T is known as a priori and we have collected $S = \epsilon T$ users till now. A fractional matching problem can be formulated as

$$\begin{aligned} \max_{X\in\Omega} \ & \sum_{t=1}^{S}\sum_{i=1}^{M} R_{it}X_{it} \\ \text{s.t.} \ & \sum_{t=1}^{S} c_i X_{it} \le \epsilon b_i, \ i \in [M] \end{aligned} \tag{3}$$

where $\Omega = \{X \in \mathcal{R}_{+}^{M\times S} : \sum_{i=1}^{M} X_{it} = 1, \forall t \in [S]\}$, and the budgets are also rescaled in proportion to the number of revealed users.

Solving (3) directly leads to the optimal allocations, i.e., $\{X_{it}^{*}\}_{i\in[S]}$ for the revealed users. However, this solution has little contribution to future arrivals. To tackle this problem, online primal-dual algorithms were proposed to learn dual variables for each bidder, which can be seen as an allocation strategy for future users. Formally, the dual problem of the primal fractional matching problem (3) can be derived via Lagrange Multiplier as

$$\min_{\boldsymbol{\alpha}\ge\mathbf{0}} \ \sum_{i=1}^{M} \epsilon b_i \alpha_i + \sum_{t=1}^{S} \max_{i\in[M]} [R_{it} - c_i\alpha_i]_{+} \tag{4}$$

Denote $\boldsymbol{\alpha}^{*}$ as the optimal solution of (4), then with the help of K.K.T conditions we can obtain the allocation rule,

$$X_{it}^{*} = \begin{cases} 1, \ i = \arg\max\{R_{it} - c_i\alpha_i^{*}\} \\ 0, \ \text{other} \end{cases} \tag{5}$$

Algorithm 1 One-Time Online Primal-Dual Algorithm

Input: T: length of user sequence; M: number of bidders; ϵ: fraction of users used for training dual variables; $\{c_i, b_i\}_{i=1}^{M}$: costs and budgets of bidders; $\{R_{:t}\}_{t=1}^{T}$: revenues stream.
 Compute $S = \epsilon T$, and initialize $X_{:t} = \mathbf{0}$, for all $t \in [T]$;
 Solve the dual problem (4) and obtain $\hat{\boldsymbol{\alpha}}$;
 for $t = S+1$ **to** T **do**
  $maxi = -1$; $maxv = -$INF;
  for $i = 1$ **to** M **do**
   if $c_i \le b_i - \sum_{j=1}^{t-1} c_i X_{ij}$ **and** $R_{it} - c_i\alpha_i > maxv$ **then**
    $maxi = i$; $maxv = R_{it} - c_i\alpha_i$;
   end if
  end for
  if $maxi <> -1$ **then**
   $X_{maxi,t} = 1$
  end if
 end for
Output: Predicted allocations $X_{:t}$, where $t = S+1, ..., T$

Remark 2. Note that $\boldsymbol{\alpha}^{*}$ has the nature role of price for each bidder, which means that when allocating, we display the bidder with the maximal value of bid minus corresponding price.

Based on the the dual formulation and the allocation rule, an online primal-dual algorithm was presented as in Algorithm 1. Firstly we collect ϵT users, with which a group of optimal dual variables could be learned by solving (4). Then under the budget constraints, we allocate bidders to the unrevealed users according to (5). (Agrawal, Wang, and Ye 2014) showed that under the assumption of random order of arrival, the online primal-dual algorithm is near-optimal. Moreover, since only one group of dual variables is trained for the future allocation in Algorithm 1, we call it *One-Time online Primal-Dual algorithm (OT-PD)*.

The Proposed Algorithm

The online primal-dual framework is based on the stochastic assumption that the coefficient $R_{:t}$ is drawn i.i.d. from an unknown distribution or arrives in a random order. However, the user distribution may not be stationary in real applications. In such cases, the online primal-dual algorithms encounter failure to some extent, because dual variables learned with observed users could suffer bias when applied to the future users from a changing distribution.

In order to design more practical algorithms, we deal with a hybrid case between the adversarial and stochastic assumption: there is a drift between the distributions of adjacent periods. In other words, the distributions of the adjacent periods are similar, but not the same. If we take the drift of distributions into consideration on the learning stage, the learned dual variables can perform better on the subsequent users.

Formulation

Note that the objective (3) tries to train dual variables by maximizing the total revenues on the revealed users, which can be seen as an optimization problem on empirical distribution. When the user distribution is changing, it is feasible to introduce a new distribution on revealed users that is tolerant of the distribution drifts. For this purpose, we borrow the idea from covarite shift learning problem that associates a weight p_t with each user. The user with larger weight is supposed to be more likely to arrive in the unrevealed sequence.

Now the key is how to learn reasonable weights for the revealed users. In this paper, we leverage techniques from distributionally robust optimization. Specifically, we define an ambiguity distribution set with KL-divergence around the empirical uniform distribution, then optimize for the worst case of distributions in the uncertainty set via a max-min objective to win a robust performance. Formally, denote $\mathcal{L} = \{l+1, ..., l+S\}$, consider the following robust matching problem defined on the user set $\mathcal{L}$,

$$\begin{aligned} \max_{X\in\Omega} \min_{\boldsymbol{p}\in\Delta} & \sum_{t\in\mathcal{L}} p_t \sum_{i=1}^{M} R_{it}X_{it} + \lambda \sum_{t\in\mathcal{L}} p_t \mathcal{H}(X_{*t}) \\ \text{s.t.} & \sum_{t\in\mathcal{L}} p_t c_i X_{it} \le \frac{\epsilon b_i}{S}, \ \forall i \in [M] \end{aligned} \tag{6}$$

where $\boldsymbol{p} = \{p_t\}_{t\in\mathcal{L}}$ is the weight associated with each user, $\Delta = \{\boldsymbol{p} \in \mathcal{R}^S : \mathrm{KL}(\boldsymbol{p}||\mathbf{1}/S) \le \delta, \ \sum_{t\in\mathcal{L}} p_t = 1\}$ is the ambiguity distribution set and δ is the maximal uncertainty. Moreover, we introduce an entropy regularizer item $\sum_{t\in\mathcal{L}} \mathcal{H}(X_{*t})$ for controlling the stability of solutions and obtaining benefits of optimizing convenience.

Remark 3. In (6), we consider the ambiguity set near uniform distribution, i.e., the distribution in the set is at most δ KL-divergence far from the uniform distribution. From this ambiguity set, we can see that two distributions from adjacent time intervals change not much and remain similar, which tends to the stochastic assumption. In the meanwhile, the worst-case distribution in the ambiguity set is learned to be optimized via the max-min formulation, which acts as the consideration in adversarial setting. Therefore, (6) deals with the hybrid case between the adversarial and stochastic assumption indeed.

Remark 4. It is critical to set the size of KL divergence radius δ, which is ralated to the number of the revealed users and the assumption about how much the user distribution changes. In practice, we could set this hyper-parameter via cross-validation.

Optimization

The max-min objective in (6) can be solved by alternating optimization. Introducing the dual variable α_i for each constraint, we can obtain

$$\begin{aligned} \min_{\alpha\ge 0} \max_{X\in\Omega} \min_{p\in\Delta} & \sum_{t\in\mathcal{L}} p_t \sum_{i=1}^{M} R_{it}X_{it} + \lambda \sum_{t\in\mathcal{L}} p_t \mathcal{H}(X_{*t}) \\ & + \sum_{i=1}^{M} \alpha_i \Big(\frac{\epsilon b_i}{S} - \sum_{t\in\mathcal{L}} p_t c_i X_{it}\Big) \end{aligned} \tag{7}$$

Since the strong max-min property holds, we can swap the inner min and max, which leads to

$$\begin{aligned} \min_{\alpha\ge 0} \min_{p\in\Delta} \max_{X\in\Omega} & \sum_{t\in\mathcal{L}} p_t \sum_{i=1}^{M} R_{it}X_{it} + \lambda \sum_{t\in\mathcal{L}} p_t \mathcal{H}(X_{*t}) \\ & + \sum_{i=1}^{M} \alpha_i \Big(\frac{\epsilon b_i}{S} - \sum_{t\in\mathcal{L}} p_t c_i X_{it}\Big) \end{aligned} \tag{8}$$

By fixing $\boldsymbol{\alpha}$ and $\boldsymbol{p}$, we consider the max problem, i.e.,

$$\begin{aligned} \max_{X\in\Omega} & \sum_{t\in\mathcal{L}} p_t \sum_{i=1}^{M} R_{it}X_{it} + \lambda \sum_{t\in\mathcal{L}} p_t \mathcal{H}(X_{*t}) \\ & + \sum_{i=1}^{M} \alpha_i \Big(\frac{\epsilon b_i}{S} - \sum_{t\in\mathcal{L}} p_t c_i X_{it}\Big) \end{aligned} \tag{9}$$

Then we set the partial derivations of X in (9) to zero and obtain the solution

$$X_{it} = \frac{1}{Z_t} \exp\Big(\frac{1}{\lambda}(R_{it} - \alpha_i c_i)\Big), \tag{10}$$

where $Z_t = \sum_{i=1}^{M} \exp(\frac{1}{\lambda}(R_{it} - \alpha_i c_i))$ is a normalization factor. By substituting (10) into (9), the dual of (6) can be cast as

$$\min_{\boldsymbol{\alpha}\ge 0, \boldsymbol{p}\in\Delta} \ \lambda \sum_{t=1}^{S} p_t \log\Big(\sum_{i=1}^{M} \exp\Big(\frac{1}{\lambda}(R_{it} - \alpha_i c_i)\Big)\Big) + \sum_{i=1}^{M} \frac{\alpha_i \epsilon b_i}{S} \tag{11}$$

This dual can be solved by alternating optimization.

Fix $\boldsymbol{p}$ and update $\boldsymbol{\alpha}$ The following proposition presents an closed form solution for α.

Proposition 1. *The closed form solution for α in* (11) *can be presented as*

$$\alpha_i = \max\Big(0, \hat{\alpha}_i - \frac{\lambda}{c_i} \log \frac{\epsilon b_i}{S w_i}\Big) \tag{12}$$

Proof. Proposition 1 can be proved with a bound optimization approach. Specifically, by exploiting the first order concavity property of the log-function, we have

$$\log(z) \le \beta z - \log(\beta) - 1, \ \forall \beta > 0 \tag{13}$$

Let $\hat{\alpha}$ be the current solution and $\frac{1}{\beta} = \sum_{i=1}^{M} \exp(\frac{1}{\lambda}(R_{it} - \hat{\alpha}_i))$, then

$$\begin{aligned} & \log\Big(\sum_{i=1}^{M} \exp\Big(\frac{1}{\lambda}(R_{it} - \alpha_i c_i)\Big)\Big) \\ \le & \log\Big(\sum_{i=1}^{M} \exp\Big(\frac{1}{\lambda}(R_{it} - \hat{\alpha}_i c_i)\Big)\Big) + \sum_{i=1}^{M} \hat{X}_{it} \exp\Big(\frac{1}{\lambda}(\hat{\alpha}_i - \alpha_i c_i)\Big) \\ \le & \sum_{i=1}^{M} \hat{X}_{it} \exp\Big(\frac{1}{\lambda}(\hat{\alpha}_i - \alpha_i) c_i\Big) \end{aligned} \tag{14}$$

where $\hat{X}_{it}$ is the primal solution (10) computed with $\hat{\alpha}$. Since p is fixed, according to 14 the optimization problem for α can be approximated by

$$\min_{\alpha\ge 0} \sum_{i=1}^{M} \Big(\lambda w_i \exp\Big(\frac{1}{\lambda}(\hat{\alpha}_i - \alpha_i) c_i\Big) + \frac{\alpha_i \epsilon b_i}{S}\Big) \tag{15}$$

where $w_i = \sum_{t\in\mathcal{L}} p_t \hat{X}_{it}$. It is obvious that (15) can be decomposed into a set of small problems, i.e., for each α_i,

$$\min_{\alpha_i \geq 0} \lambda w_i \exp(\frac{1}{\lambda}(\hat{\alpha}_i - \alpha_i)c_i) + \frac{\alpha_i \epsilon b_i}{S} \tag{16}$$

We can see that (16) is convex optimization, thus α_i can be solved in a closed form solution as

$$\alpha_i = \max\big(0, \hat{\alpha}_i - \frac{\lambda}{c_i}\log\frac{\epsilon b_i}{S w_i}\big) \tag{17}$$

□

Fix $\boldsymbol{\alpha}$ and update $\boldsymbol{p}$ Let $A_t = \log\big(\sum_{i=1}^{M}\exp(\frac{1}{\lambda}(R_{it} - \alpha_i c_i))\big)$ and $C = \epsilon - \log(T)$, the optimization problem of p can be rewritten as

$$\begin{aligned} &\min_{p} \lambda \sum_{t\in\mathcal{L}} p_t A_t \\ &\text{s.t.} \sum_{t\in\mathcal{L}} p_t \log(p_t) \leq C, \ \sum_{t\in\mathcal{L}} p_t = 1 \end{aligned} \tag{18}$$

Then introduce dual variable γ for the inequality constrain, we can obtain the Lagrange function

$$L(p,\gamma) = \lambda \sum_{t\in\mathcal{L}} p_t A_t + \gamma\big(\sum_{t\in\mathcal{L}} p_t \log(p_t) - C\big) \tag{19}$$

Set the partial derivations of p in (19) to zero, we have the solution

$$p_t = \frac{1}{Z}\exp(-\frac{\lambda A_t}{\gamma}) \tag{20}$$

where $Z = \sum_{t=1}^{T}\exp(-\frac{\lambda A_t}{\gamma})$ is a normalization factor. By substituting (20) into (19), the dual of (18) can be formulated as

$$\min_{\gamma>0} \gamma C + \gamma \log(\sum_{t\in\mathcal{L}}\exp(-\frac{\lambda A_t}{\gamma})) \tag{21}$$

The dual variable γ can be updated as

$$\gamma^{k+1} = \gamma^k - \eta\Big(C + \log\big(\sum_{t\in\mathcal{L}}\exp(-\frac{\lambda A_t}{\gamma^k}) + \frac{\lambda}{\gamma^k}\sum_{t\in\mathcal{L}} p_t^k A_t\big)\Big) \tag{22}$$

where η is the step size and p^k can be calculated with γ^k according to (20). Therefore, we alternate between these two steps until convergence.

Algorithm

For the online matching problems with user arrival distribution drifts, we have introduced a robust formulation and presented an efficient optimization method for it. Moreover, given the assumption that there is a drift between the distributions of adjacent periods, it is necessary to maintain a dynamic updating scheme when designing the algorithm framework. In this section, we propose a Robust Dynamic Learning Algorithm (RDLA), where the dual variables are updated with the latest revealed users at equal time intervals. The details are stated in Algorithm 2. As can be seen, RDLA splits the whole users into $\frac{1}{\epsilon}$ time intervals with equal length, and trains a group of dual variables by solving the robust matching objective (6) at each time interval, then use it to determine the sequential decisions in the next period.

Algorithm 2 Robust Dynamic Learning Algorithm

Input: T: length of user sequence; M: number of bidders; ϵ: fraction of users used for training dual variables; $\{c_i, b_i\}_{i=1}^{M}$: costs and budgets of bidders; $\{R_{:t}\}_{t=1}^{T}$: revenues stream.

```
Compute S = εT, and initialize X_{:t} = 0, for all t ∈ [T];
for t = S + 1 to T do
    if mod (t, εT) == 1 then
        Let L = {t − S, ..., t − 1};
        while IS_NOT_CONVERGE do
            Fix p, solve (11) by Proposition 1 and obtain α;
            Compute A_t = log( Σ_{i=1}^{M} exp(1/λ (R_it − α_i c_i)) );

            Fix α, solve (18) by GD and obtain p;
        end while
    end if
    Compute X_{:t} with α according to (10);
    maxi = −1; maxv = −INF;
    for i = 1 to M do
        if c_i ≤ b_i − Σ_{j=1}^{t−1} c_i X_ij and X_it > maxv then
            maxi = i; maxv = X_it;
        end if
    end for
    Set X_{:t} = 0
    if maxi <> −1 then
        X_{maxi,t} = 1
    end if
end for
```

Output: Predicted allocations $X_{:t}$, where $t = 1, ..., T$

The Equal Period Updating Scheme It is important to adopt a dynamic updating scheme when facing distribution drifts. According to (Agrawal, Wang, and Ye 2014), OT-PD achieves near-optimal competitive ratio under the random order assumption. However, when the user arrival distribution is drifting constantly, the dual variables trained some time ago can be outdated such that OT-PD will suffer severely degenerated performance. Therefore, it is necessary to design an algorithm that can update the dual variables dynamically. (Agrawal, Wang, and Ye 2014) proposed a Dynamic Learning Algorithm (DLA) that updates dual variables at geometric time intervals under the assumption of the random order of arrival. The dual variables learned from all revealed users are used to determine the sequential decisions in the current period.

However, the DLA framework has also several disadvantages when facing our consideration. Firstly, since the user arrival distribution is drifting all the time, it is more reasonable to keep the rate of updating stable, while the geometric period updating leads to the lag of model in last half of allocation. On the other hand, when the user arrival distribution is changing, it is unnecessary to involve all the revealed users for updating dual variables because the early revealed users can also be outdated, which may even lead to a poor performance. Based on these considerations, we adopt the equal period updating scheme and train dual variable with only users in the latest period as shown in Algorithm 2.

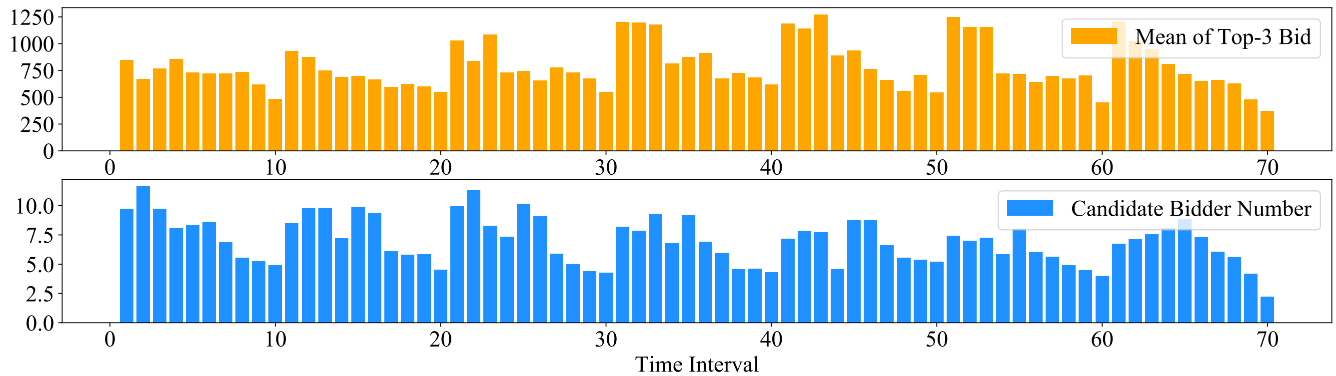

Figure 1: The mean of top-3 bids and count the number of candidating bidders in each query. Each bar shows one of the statistics averaged over users from the same time interval.

Experiments

Settings

In this section, we evaluate the effectiveness of the RDLA approach on a real displaying ads dataset. Online advertisements have been the main revenue source for many internet companies, therefore it is of great importance to improve the performance of displaying ads. The formulation of such problem is the same as (6) setting $c_i = 1, \forall i \in [M]$. To be more precise, there are M bidders each daily displaying capacities b_i and T bid search users arriving incrementally. Based on the relevance of each search keyword, the i-th bidder could expect a revenue value of R_{it} on t-th query. For the t-th query, the allocation system has to choose a vector $\boldsymbol{x_t} = \{x_{it}\}_{i=1}^{M}$, where $x_{it} \in \{0, 1\}$ indicates whether the t-th query is allocated to the i-th bidder. The goal is to maximize the total revenues over all bidders and queries under the displaying capacity constraints.

Dataset The dataset used here consists of 245 bidders and millions of search queries from 7 days. Here we take queries in one day as an offline matching programming, thus 7 displaying ads problems are evaluated. The query numbers of these problems are in order of millions. Due to the consideration of trade secrets, all of the reported information about the dataset has been masked.

In order to see whether the distribution of $X_{:t}$ is changing, we calculate the mean of top-3 bids and count the number of candidating bidders in each $X_{:t}$, which reflect the bidding level and bidding depth respectively. These two statistics could give some expression to the user arrival distribution, and are shown in Figure 1. Each bar shows one of the statistics averaged over users from the same time interval. It can be observed that the distribution of queries are drifting all the time, which is natural because the features of arrival queries and the bidding strategies of bidders on the platform are not stationary. Therefore, we expect an improved performance when our proposed RDLA is adopted in this task.

Baselines We compare the proposed RDLA with the following approaches: i) DLA with geometric period updating (DLA-geometric); ii) a variant of DLA with equal period updating (DLA-equal); iii) OT-PD; iv) Greedy. Parameters ϵ in RDLA, DLA-geometric, DLA-equal, and OT-PD are all fixed as 0.1, and λ in RDLA is set as 10. Furthermore, in order to see the influence of the choice of δ, we evaluate the proposed RDLA method with several δ setting, i.e., $1e-2$, $1e-3$, $1e-4$. Finally, the total revenues are used for evaluating the compared methods.

Results

The results of RDLA and its compared methods are shown in Table 1. As we can see, over all bid allocation problems, DLA-equal performs better than DLA-geometric. This result shows the superiority of the equal period and local updating for cases that distribution drifts. Moreover, despite that both RDLA and DLA-equal adopt the equal period and local updating scheme, the former is better than the latter, which shows the necessity of robustness consideration for online matching problems with drifting user distribution. In particular, it can be calculated that RDLA is able to increase 5.5% more total revenue than the classical OT-PD method.

Besides the dual variables, we also learn a weight for each revealed user by optimizing (6), which can be seen that search queries have different importance when training the dual variables. It is necessary to see the role of the learned weights in improving performance. For this purpose, we analyze the learned weights and some results are shown in Figure 2, where figures are grouped by columns. The red lines of the first row show the sorted ratio of the learned weights of each queries to the uniform distribution, i.e., $\frac{w_t}{1/\epsilon T}$. We note that a few queries received significantly lower weights, while most of the rest hold a bit more weights than $\frac{1}{\epsilon T}$. We further correspondingly plot the mean of top-3 bids and the number of candidating bidders, which are shown as the orange and blue lines of the second row respectively. As we see, all of the queries with significantly lower weights have relatively higher top-3 bids and more candidating bidders, which is an outcome of the max-min formulation of (6).

The contribution of this observation to the performance improvement can be explained from two views. For briefness, we call the queries with significantly higher top-3 bids

Table 1: The results (Total Revenues) of RDLA and its compared methods. Boldface highlights the largest revenues.

$\times 10^4$	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Mean±Std.
Greedy	8053.5	7820.6	9840.1	9759.5	7945.1	8136.7	8783.7	8619.9±798.5
OT-PD	8101.5	7960.6	9894.6	10309.8	9011.6	8260.1	8826.6	8909.3±838.7
DLA-geometric	8551.7	8224.6	10143.9	10621.5	9034.7	8403.8	9021.3	9143.0±841.3
DLA-euqal	8652.4	8467.4	10273.7	10376.0	9179.9	8393.9	9172.1	9216.5±758.0
RDLA ($\delta = 1e-2$)	8825.7	8550.6	10303.5	10626.0	9258.1	8505.6	9395.1	9352.1±772.3
RDLA ($\delta = 1e-3$)	**8902.9**	**8626.5**	**10388.5**	**10683.2**	9221.3	8487.2	**9463.6**	**9396.2±787.2**
RDLA ($\delta = 1e-4$)	8790.5	8552.3	10367.2	10483.0	**9272.8**	**8509.0**	9313.2	9326.9±753.9

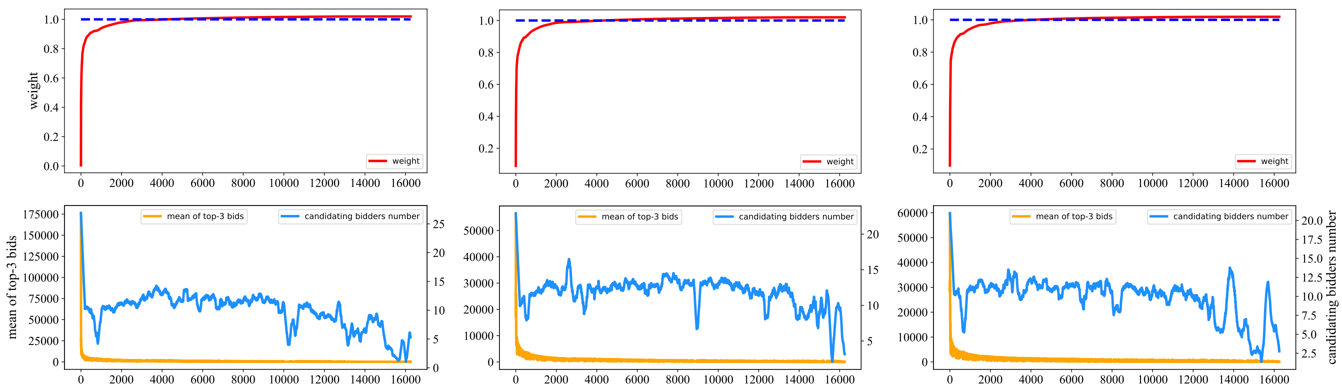

Figure 2: These figures are grouped by columns. The red lines in the first row show the sorted ratio of the learned weights of each queries to the uniform distribution, the orange and blue lines in the second row show the mean of top-3 bids and the number of candidating bidders, correspondingly.

and more candidating bidders *Fall Samples*. From robust optimization aspect, the Fall Sample can be seen as some noise, which could be a large component in the empirical total revenue but actually has no help for improving the dual variables. From distribution drifting aspect, the Fall Samples in previous time interval can be disappeared in the next one with high probability. For both of the above consideration, it can be impactful to reduce the weights of Fall Samples, which has been validated by the empirical results.

Moreover, it is notable that RDLA training with $\delta = 1e-3$ performs better than that with $\delta = 1e-2$ and $\delta = 1e-4$. This phenomenon is reasonable: when δ is set too small, the distribution in the ambiguity set is too close to the nominal distribution, thus there is not enough space for RDLA to capture the user distribution drifts; when δ is set too large, the ambiguity set will contain overmuch distributions that are far from the nominal distribution, leading to a large variance of the algorithm performance. Therefore, the hyper-parameter δ should be chosen carefully such that the learned models can capture the real user arrival fashion. In practice, we could set this hyper-parameter via cross-validation.

Conclusion

Most existing online matching methods have adopted either adversarial or stochastic user arrival assumption, while on both of them significant limitation exists. In this paper, we consider a novel user arrival model where users are drawn from drifting distribution, which is a hybrid case between the adversarial and stochastic model, and propose a new approach RDLA to deal with such assumption. Instead of maximizing empirical total revenues on the revealed users, RDLA leverages distributionally robust optimization techniques to learn dual variables via a worst-case consideration over an ambiguity set on the underlying user distribution. Furthermore, we present a dynamic learning scheme to tackle the drifting distribution by updating dual variables at equal time intervals. Experiments on a real internet advertising task exhibit the impressive performance of RDLA, showing the necessity of the robustness consideration and the superiority of equal period updating scheme.

Acknowledgement

Authors want to thank reviewers for helpful comments and thank Peng Zhao and Yao-Xiang Ding for helpful discussions.

References

Aggarwal, G.; Goel, G.; Karande, C.; and Mehta, A. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, 1253–1264.

Agrawal, S.; Wang, Z.; and Ye, Y. 2014. A dynamic near-optimal algorithm for online linear programming. *Operations Research* 62(4):876–890.

Ben-Tal, A., and Nemirovski, A. 1998. Robust convex optimization. *Mathematics of Operations Research* 23(4):769–805.

Ben-Tal, A.; El Ghaoui, L.; and Nemirovski, A. 2009. *Robust Optimization*, volume 28.

Berge, C. 1957. Two theorems in graph theory. *Proceedings of the National Academy of Sciences* 43(9):842–844.

Bertsimas, D.; Brown, D. B.; and Caramanis, C. 2011. Theory and applications of robust optimization. *SIAM Review* 53(3):464–501.

Buchbinder, N., and Naor, J. 2009. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science* 3(2-3):93–263.

Delage, E., and Ye, Y. 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research* 58(3):595–612.

Devanur, N. R., and Hayes, T. P. 2009. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, 71–78.

Esfahani, P. M., and Kuhn, D. 2018. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming* 171(1-2):115–166.

Esfandiari, H.; Korula, N.; and Mirrokni, V. S. 2015. Online allocation with traffic spikes: Mixing adversarial and stochastic models. In *Proceedings of the 16th ACM Conference on Economics and Computation*, 169–186.

Feldman, J.; Henzinger, M.; Korula, N.; Mirrokni, V. S.; and Stein, C. 2010. Online stochastic packing applied to display ad allocation. In *Proceedings of the 18th Annual European Symposium*, 182–194.

Gale, D., and Shapley, L. S. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69(1):9–15.

Gao, R., and Kleywegt, A. J. 2017. Distributionally robust stochastic optimization with dependence structure. *arXiv preprint arXiv:1701.04200*.

Ghaoui, L. E.; Oustry, F.; and Lebret, H. 1998. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization* 9(1):33–52.

Goh, J., and Sim, M. 2010. Distributionally robust optimization and its tractable approximations. *Operations Research* 58(4-Part-1):902–917.

Hu, Z., and Hong, L. J. 2013. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*.

Kalyanasundaram, B., and Pruhs, K. 2000. An optimal deterministic algorithm for online b-matching. *Theoretical Computer Science* 233(1-2):319–325.

Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 352–358.

Mehta, A. 2013. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science* 8(4):265–368.

Mirrokni, V. S.; Gharan, S. O.; and Zadimoghaddam, M. 2012. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, 1690–1701.

Namkoong, H., and Duchi, J. C. 2016. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems 29*, 2208–2216.

Namkoong, H., and Duchi, J. C. 2017. Variance-based regularization with convex objectives. In *Advances in Neural Information Processing Systems 30*, 2975–2984.

Vee, E.; Vassilvitskii, S.; and Shanmugasundaram, J. 2010. Optimal online assignment with forecasts. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, 109–118.

Wiesemann, W.; Kuhn, D.; and Sim, M. 2014. Distributionally robust convex optimization. *Operations Research* 62(6):1358–1376.