Sequential Selection of Correlated Ads by POMDPs

Shuai Yuan, Jun Wang
Department of Computer Science, University College London
{s.yuan, j.wang}@cs.ucl.ac.uk

ABSTRACT

Online advertising has become a key source of revenue for both web search engines and online publishers. For them, the ability of allocating *right* ads to *right* webpages is critical because any mismatched ads would not only harm web users' satisfactions but also lower the ad income. In this paper, we study how online publishers could optimally select ads to maximize their ad incomes over time. The conventional offline, content-based matching between webpages and ads is a fine start but cannot solve the problem completely because good matching does not necessarily lead to good payoff. Moreover, with the limited display impressions, we need to balance the need of selecting ads to learn true ad payoffs (exploration) with that of allocating ads to generate high immediate payoffs based on the current belief (exploitation). In this paper, we address the problem by employing Partially observable Markov decision processes (POMDPs) and discuss how to utilize the correlation of ads to improve the efficiency of the exploration and increase ad incomes in a long run. Our mathematical derivation shows that the belief states of correlated ads can be naturally updated using a formula similar to collaborative filtering. To test our model, a real world ad dataset from a major search engine is collected and categorized. Experimenting over the data, we provide an analyse of the effect of the underlying parameters, and demonstrate that our algorithms significantly outperform other strong baselines.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*; I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*

General Terms

Algorithms, Design, Performance

Keywords

computational advertising, revenue optimisation, correlation, POMDPs, value iteration

1. INTRODUCTION

Online advertising has received blooming development in recent years. The volume of the industry has grown from $8.1 billion in 2000 to $26 billion in 2010 with a new record every year [1]. Online advertising is vitally important for both web search engines and online content providers and publishers because it provides them with major sources of revenue. For example, Google search engine was serving 7.2 billion page views per day in 2011 [9] and its revenue is mainly driven by advertising products like AdWords, DoubleClick, and AdMob. Similarly, online content providers such as Yahoo! and news agencies such as New York Times increasingly choose advertising as the smarter alternative to conventional subscription based services.

Online publishers can make profit by selling *impressions* (an instance of an ad being seen on users' monitors). A fundamental problem faced by publishers is that given limited display impressions and ad slots, how to maximize their ad incomes over time. It is a complex problem because in practice publishers usually need to make decisions considering various aspects. First, they have to decide whether to deal private contracts with advertisers or agencies directly, or to participate in public ad networks or exchanges to reach more demands. In [14] a queuing system was proposed to find the optimal policy of selecting advertisers to make private contracts, whereas in [6, 20], the focus is on pricing ads properly in either of the two settings. Second, the payment scheme is different and the publishers need to choose between pay-per-view, pay-per-click, and possibly other models. A balance can be established in a static setting as reported in [21, 13, 19]. Lastly and most importantly, a decision has to be made on which page [8, 33] and which users [31, 28] these ads should be matched with, probably in a real-time fashion using text summaries [3]. Traditionally the matching is done by ad networks and advertisers are allowed to choose which keywords they intend to bid. Now publishers are getting more involved and can actively switch in real-time between different pricing schemes and different networks in order to increase their ad incomes, as demonstrated in the Google DFP Small Business system [15] and some ad exchanges like AdBrite (`http://www.adbrite.com/`).

This essentially allows online publishers to integrate the above decisions all together in a more general framework to optimize their ad revenue. Yet, the following challenges remain unsolved because of its dynamic settings: First, the prior analysis of the matching between content and ads provides a fine start but a good payoff is not necessarily guaranteed by a good matching and need to be verified over time.

With the limited display impressions, we also need to balance the need of selecting ads and their pricing schemes to learn the true payoffs (exploration) with that of allocating ads to generate high immediate payoffs based on the current belief (exploitation). Second, online publishers have more opportunities to explore from different ad networks and pricing schemes provided ads are correlated. A key question is how to make use of the correlations embedded in the data to improve the efficiency of the exploration and increase the ad incomes in a long run. In this paper, we study the above two issues and formulate the sequential ad selection problem by applying Partially observable Markov decision processes (POMDPs). To provide a basic understanding of publisher revenue problem when dealing with multiple ads or ad sources, we do not distinguish between various trade mechanisms and pricing models, but instead consider them essentially the same and focus on their payoffs. Moreover, to make our study focused, we formulate our problem by considering the correlation of ads, while bearing in mind that the same principle and result can be applied to the correlation of webpages and users. Our mathematical derivation shows that the belief states of correlated ads can be naturally updated using a formula similar to collaborative filtering. We examine the model on a carefully collected dataset, and the results show that our models, particularly with the new belief updates, outperform other strong baselines.

The remainder of the paper is organized as follows: in Section 2 we provide a discussion about related work and in Section 3 we present the proposed model and its approximate solutions. The experiments are given in Section 4 and conclusions are presented in Section 5.

2. RELATED WORK

Previous research on publisher revenue optimization has been mainly focused on display ads and their (private) contracts. Traditionally the research question is limited to how to choose advertisers to make (private) contracts with, as elaborated in [12, 14, 26]. In [14] the authors used a queueing system to accommodate advertisers and added constraint that advertisers are impatient and would leave if their ads cannot be displayed right way. A dynamic programming solution was provided in [26] by combining the available ad inventories and dynamically delivery of promised advertising contract to the viewers. Although our model also optimizes publisher revenue over time, the settings are different: we focus on contextual advertising, where ads can be specified using keywords or categories, and study how the matching between webpages and ads can be established w.r.t. payoffs over time. We introduce a simple yet general dynamical payoff model that helps to address the inventory management problem while taking into account the correlation of ads.

An interesting variation of ad inventory management problem is to find the optimal pricing models. The pay-per-view and pay-per-click models were discussed in [17]. In [21], the study concluded that a combination of pay-per-view and per-per-click pricing models may be the most optimal for an advertiser who has to choose between them. The same choice problem has later been studied in [13] and the author concluded that the optimal choice for a price taking publisher is either pay-per-view or per-per-click but not a combination of those two pricing models. Recently similar conclusion was reported in [19], in which the competition between two models was discussed but limited to a static setting: a single period of planning horizon. Similarly in [14] the authors assumed that the publisher could choose ad networks to utilize remnant impressions after fulfilling contracts. Our goal is similar to those papers since we are also targeted to find optimal ad sources. However, our model is focused on real-time impression-based setting; we also utilize the correlation of ads to accelerate the discovery process, and provide an analyse of the optimization problem over time. For instance, different from [13] we do not make any assumption on market elasticity; neither do we constraint the publisher to a contract-first setup [14].

Another aspect of publisher revenue optimization is ad scheduling with the constraint of geometrical features of website, uncertainty of advertisers and available ad inventory (impressions) [18, 26, 30]. If the publisher fails to deliver promised impressions a good-will penalty would be incurred. In [30] the ad scheduling problem was modelled in video games and a dynamic algorithm was developed. When considering pricing models, e.g. pay-per-click, publishers need to understand the webpage content and user preferences to achieve a better scheduling. Because now displaying ads may not earn the publisher anything if ads are not relevant or users are not interested therefore no clicks. Traditionally optimizing matching between webpages and ads belongs to the field of contextual advertising research [8, 24]. In [8] a system utilizing both semantic and syntactic features was proposed to address the problem. Similarly the correlation of user-ad (behaviour targeting) has also been studied previously in [32, 11], focusing on improving advertising effectiveness by modelling attitude and feedback from users. Our study can be consider as a dynamical extension of the above works. The contextual matching can be used as a prior belief in our model and it can be updated and verified using our update formulas sequentially.

In terms of techniques, our work is closely related to multi-arm bandit with dependent arms discussed in [23] as we also face a multi-period selection problem with an exploration and exploitation dilemma. In [23] the dependencies of candidates are modelled by clustering them first; after that, a multi-arm bandit algorithm runs twice: first selects a cluster and then a candidate in that cluster. Our paper is different in that we directly model the dependencies using a covariance matrix. The impact of correlation is well formulated and illustrated by rigorously deriving the belief updates once other correlated ads have been selected. Specifically, the problem is formulated by applying Partially Observable Markov Decision Processes (POMDPs) with discrete action (selection of ads), continuous observations (payoffs), and continuous hidden states (performance of ads). Our model is a special case of continuous POMDPs [10, 29], where two-stage Gaussian generative processes and no transit of hidden states are considered during the planning horizon. To provide an optimal ad selection, similar with [29, 27], we follow the Monte Carlo sampling approach to deal with continuous observations and to approximate a Dynamic Programming solution in the finite planning horizon. As a complement of the approximation approach, we also extend an Upper Confidence Bound algorithm [4] by integrating it with our belief update method.

3. THE PROPOSED MODEL

3.1 The Sequential Payoff Model

In this section we formulate the sequential ad selection problem for a publisher-side system. Suppose there is an online publisher who wants to put ads on their hosting webpages to generate profits. The ads could be obtained from various sources either by making contracts with advertisers directly, by registering with ad networks, or by employing a supply-side-platform [15]. We now formally introduce the sequential payoff model to describe how the publisher revenue is earned from ads. Suppose there are N ads from various ad networks or exchanges available for the publisher. For each display impression, the publisher needs to decide which ad should be selected. Without loss of generality, we consider to make our selection decisions every M impressions and denote the decision times as $t \in [1, ..., T]$. To stay focused, we consider the impression has only one ad slot, while bearing in mind that the scenario of multiple ad slots per impression can be addressed by incorporating user click-through models (for pay-per-click ads) to remove rank bias [25] or consider them from different impressions (for pay-per-view ads).

For each time step $t \in [1, ..., T]$, we define

$$\Psi(t) = \begin{bmatrix} s(1), & x(1) \\ ..., & ... \\ s(t-1), & x(t-1) \end{bmatrix} \tag{1}$$

as the available information up to time t, where $s(t) \in N$ denotes the decision, i.e. the index of ad selected for the time step t. We use the random variable $X(t)$ to denote the payoff gained at time step t and $x(t)$ its realization.

Let π be an arbitrary choosing policy according to the information obtained so far. We define

$$s(t) \equiv \pi(\Psi(t)) \tag{2}$$

To simplify our notation, we use them exchangeably in the rest of the paper. The cumulative payoff over time T with certain policy π is

$$R_\pi(T) \equiv M \sum_{t=1}^{T} X_{s(t)}(t) \tag{3}$$

Our goal is to choose an optimal policy to maximize $R_\pi(T)$. However, we do not observe any future $x_{s(t)}(t)$ (where $t \geq t'$) before making a decision about $s(t)$ (at $t = t'$). For a given webpage, we assume two generative processes to generate the payoffs as shown in Figure 1. First, we consider the matching between ads and the webpage and denote the true but unknown payoffs of ads over a webpage as $\boldsymbol{\mu}$, a N-dimension vector. This vector is generated from a multivariate Gaussian distribution governed by mean vector $\boldsymbol{\theta}$ and covariance matrix $\boldsymbol{\Sigma}$ as the following

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma}) \tag{4}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ are the parameters of the model and can be estimated beforehand from data. Meanwhile, considering the fact that the payoffs are affected by either the visiting users, some unexpected factors, or the uncertainty that has not been well modelled from the Gaussian, the observed payoffs are generated from the true payoffs by

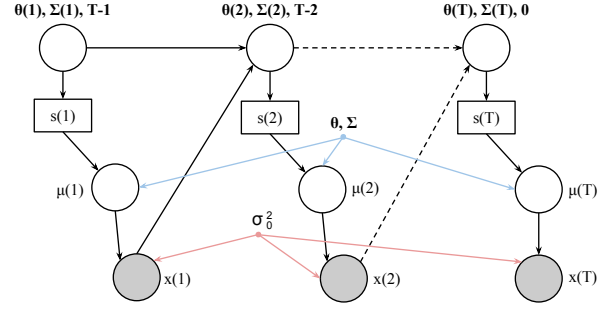$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma_0^2 \cdot \boldsymbol{I}) \tag{5}$$

Figure 1: The payoff model illustrated by an influence diagram representation with generative processes of a finite horizon POMDP. In this diagram the non-filled circular nodes represent variables including belief states; the shaded nodes represent rewards as well as observations of the system; the black point nodes represent non-random values; and the non-filled square rectangular nodes represent actions. The dotted lines indicate indirect dependency and intermediate nodes are not drawn. Note that $s(\cdot)$ also depends on σ_0^2 but lines are not drawn for simplicity.

where $\boldsymbol{X}$ is a N-dimension observed payoff vector and $\boldsymbol{I}$ is the identity matrix. A simple variance σ_0^2 is used to model the noise. Note that our treatment of the uncertainty from the underlying users is basic that we use a universal constant to describe the possible noise. We will show in our experiments that the noise factor, although simple, plays an important role in controlling the sensitivity of our model. For a more advanced treatment about user modelling, we refer to [28]. Given the two-stage process, the objective of the publisher is to find the optimal policy which maximizes the expectation of the overall ad income through the period, i.e.

$$\begin{aligned}
\pi^* &= \arg\max_\pi \mathbb{E}\left[R_\pi(T)\right] \\
&= \arg\max_\pi \mathbb{E}\left[\sum_{t=1}^{T} X_{s(t)}(t)\right] \\
&= \arg\max_\pi \sum_{t=1}^{T} \mathbb{E}\left[X_{s(t)}(t)\right] \\
&= \arg\max_\pi \sum_{t=1}^{T} \int_x x_{s(t)}(t) p(x_{s(t)}(t)|\Psi(t)) dx \\
&= \arg\max_\pi \sum_{t=1}^{T} \theta_{s(t)}(t) \tag{6}
\end{aligned}$$

where we drop M from Equation 3 because the decision is independent of it. Note that $\theta_{s(t)}(t)$ is the shorthand for $\hat{\mu}_{s(t)}|\Psi(t)$, denoting the estimated expectation of $\mu_{s(t)}$ at time step t giving all available information $\Psi(t)$. It presents our belief at time t. Our formulation is in fact a special case of continuous POMDPs [10, 29], where $\boldsymbol{\mu}$ is the hidden state and $\boldsymbol{X}$ is the observation over time and our belief about $\boldsymbol{\mu}$ would be sequentially updated using a posterior probability. Our next task is to provide the estimation $\theta_{s(t)}(t)$ at each time step t after an ad has been selected and its payoff has been observed. We are particularly interested in the up-

date for correlated ads, which will be discussed in the next section.

3.2 Belief Updates

At each time step the publisher makes a decision, observes the payoff, then updates expected payoffs of all ads. In order to calculate the expected payoffs the publisher needs to calculate the value of $\boldsymbol{\theta}(t+1)$ and $\boldsymbol{\Sigma}(t+1)$ according to observation $x_{s(t)}(t)$ and previous belief. In this section we derive the update equation using Bayesian inference. Let us first look at the brief update of the same ads. Suppose the publisher has two ad candidates. The 1st ad was selected at time step t and received a payoff of $x_1(t)$. With Bayes' theorem and marginalizing μ_1 out, we obtain the p.d.f. of X_1 conditioned on the new observation $x_1(t)$ and previous available information $\Psi(t)$:

$$p\left(x_1|x_1(t), \Psi(t)\right)$$
$$= \int p\left(x_1|x_1(t), \Psi(t), \mu_1\right) p\left(\mu_1|x_1(t), \Psi(t)\right) d\mu, \quad (7)$$

where we know

$$p\left(\mu_1|x_1(t), \Psi(t)\right) \propto p(x_1(t)|\mu_1, \Psi(t))p(\mu_1|\Psi(t))$$
$$\propto \exp\left\{-\left(x_1(t)-\mu_1\right)^2 - \left(\mu_1-\theta_1(t)\right)^2\right\} \quad (8)$$

and by inspecting the exponent part, we can find the posterior distribution of μ_1 as

$$\mu_1|x_1(t) \sim \mathcal{N}\left(\theta_1(t+1), \sigma_1^2(t+1)\right) \quad (9)$$
$$\theta_1(t+1) = \frac{\sigma_1^2(t)x_1(t) + \sigma_0^2\theta_1(t)}{\sigma_1^2(t) + \sigma_0^2}$$
$$\sigma_1^2(t+1) = \frac{\sigma_1^2(t)\sigma_0^2}{\sigma_1^2(t) + \sigma_0^2}$$

where similarly we write $\sigma_i^2(t)$ as the shorthand for $\sigma_i^2|\Psi(t)$.

Substituting the posterior of μ_1 into Equation 7 gives the expected payoff of the selected ad as

$$X_1|x_1(t), \Psi(t) \sim \mathcal{N}\left(\theta_1(t+1), \sigma_0^2 + \sigma_1^2(t+1)\right) \quad (10)$$

where recall that we assume the prior noise σ_0^2 is known.

3.2.1 Correlated Ads

In a real world situation the payoffs of ads are correlated. Similar products, using similar creative, or targeting similar potential customers will generate similar payoffs. By considering the correlation of ads the publisher could find a more efficient way of identifying best candidates because not only the beliefs of the selected ads themselves can be updated using Equation 10, but so do the other correlated ads. Again with Bayes' theorem and marginalizing μ_1 out, we obtain the p.d.f. of X_2 conditioned on the observation $x_1(t)$ and previous available information $\Psi(t)$

$$p\left(x_2|x_1(t), \Psi(t)\right)$$
$$= \int p\left(x_2|\mu_2, x_1(t), \Psi(t)\right) p(\mu_2|x_1(t), \Psi(t))d\mu_2 \quad (11)$$

where

$$p(\mu_2|x_1(t), \Psi(t))$$
$$\propto p(x_1(t)|\mu_2, \Psi(t))p(\mu_2|\Psi(t))$$
$$= p(\mu_2|\Psi(t)) \int p(x_1(t)|\mu_1, \Psi(t))p(\mu_1|\mu_2, \Psi(t))d\mu_1 \quad (12)$$

With the covariance known, we have the conditional distribution of μ_1 on μ_2 as

$$\mu_1|\mu_2 \sim \mathcal{N}(\theta_1|\mu_2, \sigma_1^2|\mu_2) \quad (13)$$
$$\theta_1|\mu_2 = \theta_1 + \frac{\sigma_{1,2}}{\sigma_2^2}\left(\mu_2 - \theta_2\right)$$
$$\sigma_1^2|\mu_2 = \sigma_1^2 - \frac{\sigma_{1,2}^2}{\sigma_2^2}$$

where $\sigma_{1,2}$ is the covariance of $\{\mu_1, \mu_2\}$. Substituting them into Equation 12 gives

$$\mu_2|x_1(t) \sim \mathcal{N}(\theta_2(t+1), \sigma_2^2(t+1)) \quad (14)$$
$$\theta_2(t+1) = \theta_2(t) + \sigma_{1,2}\frac{x_1(t) - \theta_1(t)}{\sigma_1^2(t) + \sigma_0^2}$$
$$\sigma_2^2(t+1) = \sigma_2^2(t) - \frac{\sigma_{1,2}^2}{\sigma_1^2(t) + \sigma_0^2}$$

Similarly substituting the posterior of μ_2 from Equation 14 to Equation 11 we obtain the expected payoff of the non-selected ad as

$$X_2|x_1(t), \Psi(t) \sim \mathcal{N}\left(\theta_2(t+1), \sigma_0^2 + \sigma_2^2(t+1)\right) \quad (15)$$

Note the correctness of the equation can be verified by setting the 1st and 2nd ads equal – if we have $\theta_1 = \theta_2$ and $\sigma_1^2 = \sigma_2^2 = \sigma_{1,2}$, then Equation 14 becomes exactly Equation 9. Therefore we take Equation 14 as the unified equation covering both self and correlated updates. The objective function in Equation 6 is now completed with the belief update formulas (constraints), all of which are now summarized together as the following

$$\pi^* = \arg\max_\pi \sum_{t=1}^{T} \theta_{s(t)}(t) \quad (16)$$

subject to

$$\theta_{s(t+1)}(t+1) = \theta_{s(t+1)}(t) + \sigma_{s(t),s(t+1)}\frac{x_{s(t)}(t) - \theta_{s(t)}(t)}{\sigma_{s(t)}^2(t) + \sigma_0^2} \quad (17)$$

$$\sigma_{s(t+1)}^2(t+1) = \sigma_{s(t+1)}^2(t) - \frac{\sigma_{s(t),s(t+1)}^2}{\sigma_{s(t)}^2(t) + \sigma_0^2} \quad (18)$$

It is worth noticing that the update in Equation 17 is closely related to the "word of mouth" heuristic adopted in collaborative filtering [7, 16]. In the collaborative filtering approaches, particularly, the user-based ones, the rating of a target user is estimated by looking at other similar users; The more similar a user is, the more contribution he or she would have to the prediction. Using the heuristic, the final rating prediction is a weighted average across all similar users [16] and the similarity is usually measured by cosine similarity or Pearson's correlation coefficient and user means are used to remove the bias of mean ratings among users [7]. In this paper, using a simple Gaussian model, we naturally derive a collaborative update mechanism across correlated ads. The major difference is that the update is in a sequential way. As seen in Equation 17, the similarity measure here is the correlation normalized by the variance, and $\theta_{s(t+1)}(t)$ and $\theta_{s(t)}(t)$ are used to remove the bias from the mean payoffs between different ads. Moreover, Equation 18 naturally provides the confidence of the predictions.

Optimizing Equation 16 leads to the exploration and exploitation dilemma. The publisher would like to earn more

by selecting the best known $\theta_{s(t)}(t)$ so far. But some ads with higher variances might potentially have higher payoffs and are also required to be selected in order to gather the feedback. Not selecting the local best may result in a loss of the immediate reward, but the loss, however, could be compensated if any better alternatives can be found in later stages. Besides, we can see from the model that the changing dataset problem is dealt with naturally: the new coming ads could be simply assigned high variances to encourage the exploration on them.

3.3 Exact Solution

3.3.1 Value Iteration

Our revenue optimization problem in Equation 16 could be solved exactly following a value iteration approach using Dynamic Programming [5]. Recall we use $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ to denote the belief of N ads true payoffs before $t = 1$. Let $V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, T)$ denote the max possible revenue the publisher could gain in T time steps. We have the following statement and the Bellman equation [5].

LEMMA 1. *For any given priori $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$, there exists an optimal policy π^* to the problem in Equation 16, and $V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, T)$ is achievable. More over $V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, T)$ satisfy the following condition*

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, T)$$
$$= \max_{s(1) \in N} \mathbb{E}\left[X_{s(t)}(1) + V^*\left(\boldsymbol{\theta}|X_{s(t)}(1), \boldsymbol{\Sigma}|X_{s(t)}(1), T-1\right)\right]$$
$$(19)$$

By solving this equation recursively we find the optimal policy. If $T = 1$, we write the optimal revenue as

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 1) = \max_{s(1) \in N} \mathbb{E}\left[X_{s(1)}(1)\right]$$
$$= \max_{s(1) \in N} \theta_{s(1)}(1) \qquad (20)$$

which indicates that the publisher should simply choose the ad with highest expected payoff. This choice is straightforward because no more time steps exist, thus no need of exploration.

For $T = 2$ the optimal revenue is written as

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 2)$$
$$= \max_{s(1) \in N} \mathbb{E}\left[X_{s(1)}(1) + V^*\left(\boldsymbol{\theta}|X_{s(t)}(1), \boldsymbol{\Sigma}|X_{s(t)}(1), 1\right)\right]$$
$$= \max_{s(1) \in N} \int p(x_{s(1)})\left(x_{s(1)} + U^*(\boldsymbol{\theta}|x_{s(t)}(1), \boldsymbol{\Sigma}|x_{s(t)}(1), 1)\right) dx_{s(1)}$$
$$= \max_{s(1) \in N} \int p(x_{s(1)})\left(x_{s(1)} + \max_{s(2) \in N}\left(\theta_{s(2)}(2)\right)\right) dx_{s(1)}$$
$$= \max_{s(1) \in N} \left(\theta_{s(1)}(1) + \int \max_{s(2) \in N} p(x_{s(1)})\theta_{s(2)}(2) dx_{s(1)}\right) \quad (21)$$

The difficulty lies in the last integral as it depends on the max operator. Using Chasles relation, it could be expanded to several regional integrals according to the random vector $\boldsymbol{\theta}(2)$. For instance, if the publisher has to choose from only two ads, and by solving $\theta_1(2) > \theta_2(2)$ the following answer is obtained

$$\theta_1(2) > \theta_2(2) \text{ when } x_i(1) > k \qquad (22)$$

which indicates the publisher should choose the 1st ad when the observation from the 1st time step is bigger than some

value k. Then the last integral of Equation 21 could be broken into two regional integrals with exact solution

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 2) = \max_{i \in 1,2} \left(\theta_1(1) + \int_{-\infty}^{k} p(x_i)\theta_2(2)dx_i \right.$$
$$\left. + \int_{k}^{+\infty} p(x_i)\theta_1(2)dx_i\right) \qquad (23)$$

The above equation could be easily extended to N ads cases provided the solution of N inequalities for vector $\boldsymbol{\theta}$, where the only variable considered as unknown is the observation from the last time step. For simplicity we define that a region is *dominated* by some ad when the ad should be selected if the observation falls in the region. This is similar with [27] where the value function is expressed as a linear combination of α-functions. Formally we denote the region dominated by i-th ad at time step t as $[m_{i,t}, n_{i,t}]$. For a general case of N ads, Equation 21 is written as

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 2) = \max_{s(1) \in N} \left(\theta_{s(1)}(1) + \sum_{s(2)}^{N} \int_{m_{s(2),2}}^{n_{s(2),2}} p(x_i)\theta_{s(2)}(2)dx_i\right)$$
$$(24)$$

where the regional integral could be solved as

$$\int_{m}^{n} xp(x)dx = -\sigma^2\left(p(n) - p(m)\right) + \mu\left(\phi(n) - \phi(m)\right)$$
$$(25)$$

where $\phi(x)$ is the c.d.f. for the Gaussian random variable X. Note that for some ads at some time steps, their dominant regions could be empty, simply indicating the ads would never be selected under such circumstances.

3.3.2 A Toy Example

We give an example to demonstrate the sequential selection mechanism with embedded correlated belief update. Assume a publisher have 2 ads to select from. The Gaussian noise from users is given by

$$\sigma_0^2 = 0.1$$

Random state $\boldsymbol{\mu}$ is defined by a bivariate Gaussian, i.e.

$$\boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\sigma})$$

where

$$\boldsymbol{\theta} = \begin{bmatrix} 1 \\ 0.95 \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} 10 & 0.2 \\ 0.2 & 50 \end{bmatrix}$$

Considering only one time step gives the expected revenue

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 1) = \max_{s(1) \in N} \mathbb{E}\left[X_{s(1)}(1)\right]$$
$$= \max(1, 0.95)$$
$$= 1 \text{ with } s(1) = 1$$

Now consider $T = 2$. Suppose the 1st ad at the 1st time step is selected, which yields the following update for time step 2

$$\text{when } s(1) = 1:$$
$$\theta_1(2) = 1 - \frac{10 \times (1 - x_1(1))}{0.1 + 10}$$
$$\theta_2(2) = 0.95 - \frac{0.2 \times (1 - x_1(1))}{0.1 + 10}$$

which gives

$$\theta_1(2) > \theta_2(2) \text{ when } x_1(1) > 0.95$$

Thus, the optimal reward for $t = 2$ when choosing $s(1) = 1$ could be derived as

$$V_{s(1)=1}^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 2)$$
$$= 1 + \int_{0.95}^{+\infty} p(x_1)\theta_1(2)dx_1 + \int_{-\infty}^{0.95} p(x_1)\theta_2(2)dx_1$$
$$= 3.1993$$

A small defect is that variable x stands for the payoff observed from the 1st time step and it cannot be smaller than zero. However, due to the Gaussian assumption, when calculating V^*, we integrate x over the Real space and sometime result in a negative expected reward. But such policy would vanish in later comparisons due to its low value and has little effect on our decision process.

Similarly selecting the 2nd ad at the 1st time step yields the following update for step 2

$$\text{when } s(1) = 2 :$$
$$\theta_1(2) = 1 - \frac{0.2 \times (0.95 - x_2(1))}{0.1 + 50}$$
$$\theta_2(2) = 0.95 - \frac{50 \times (0.95 - x_2(1))}{0.1 + 50}$$

which gives

$$\theta_1(2) > \theta_2(2) \text{ when } x_2(1) < 1.00$$

Thus we have

$$V_{i=2}^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 2)$$
$$= 0.95 + \int_{-\infty}^{1} p(x_2)\theta_1(2)dx_2 + \int_{1}^{+\infty} p(x_2)\theta_2(2)dx_2$$
$$= 4.7291$$

Finally the maximum expected payoff over two time steps is

$$V^*(\boldsymbol{\theta}, \boldsymbol{\Sigma}, 2) = \max(3.1993, 4.7291) = 4.7291$$

and the corresponding optimal policy is

$$\pi^* = \{s(1) = 2, s(2) = 1 \text{ if } x_1(1) < 1, s(2) = 2 \text{ otherwise}\}$$

From the result, we can see that the optimal policy is different from a myopic one, which tries to maximize the immediate reward only. Following the myopic policy the publisher would choose $s(1) = 1$ and receive a smaller payoff. One of the reasons of selecting the 2nd is because it has a higher variance (taking into account the correlations as well). We will show in the experiments that such high variance is ordinary in real world data.

3.4 Approximate Solution

In order to understand approximations we write the objection function as a combination of immediate reward estimation and exploration function, which utilizes the available information up to the decision time step, denoted by $\xi(\Psi(t), i)$ for the i-th candidate. The decision criteria is to maximize some the objective value function $V_{s(t)}(\Psi(t))$, i.e.

$$s(t) = \underset{s(t) \in N}{\arg \max} V_{s(t)}(\Psi(t))$$
$$= \underset{i \in N}{\arg \max} (\bar{x}_i + \xi(\Psi(t), i)) \quad (26)$$

Algorithm 1 The VI-COR algorithm using value iteration with Monte Carlo sampling.

function VALUEFUNC($\boldsymbol{\theta}, \boldsymbol{\Sigma}, t$)
   array $V \leftarrow 0$ ▷ Expected reward vector.
   loop $i \leftarrow 1$ to N
     $V[i] \leftarrow \theta_i(t)$ ▷ Expected immediate reward.
     if $t < T$ **then**
       for all s in SAMPLE($\boldsymbol{\theta}, \boldsymbol{\Sigma}$) **do**
         $[\boldsymbol{\theta}', \boldsymbol{\Sigma}'] \leftarrow$ UPDATEBELIEF($\boldsymbol{\theta}, \boldsymbol{\Sigma}, s, i$)
             ▷ New belief after selecting i and observing s.
                    ▷ Equations 17 and 18.
         $V[i] \leftarrow V[i] + \frac{1}{M_0}$VALUEFUNCTION($\boldsymbol{\theta}', \boldsymbol{\Sigma}', t+1$)
       end for
     end if
   end loop
   return [MAX(V), MAXINDEX(V)]
end function

Algorithm 2 The UCB1-NORMAL-COR algorithm using multi-armed bandit with correlated update.

function PLAN($\boldsymbol{\theta}, \boldsymbol{\Sigma}, \Psi(t)$)
   array $V \leftarrow 0$
   loop $i \leftarrow 1$ to N
     if $t_i < \lceil 8 \log t \rceil$ **then**
       return i
     end if
   end loop
   $[\boldsymbol{\theta}', \boldsymbol{\Sigma}'] \leftarrow$ UPDATEBELIEF($\boldsymbol{\theta}, \boldsymbol{\Sigma}, \Psi(t)$)
       ▷ New belief of all ads with all available information.
                   ▷ Equations 17 and 18.
   loop $i \leftarrow 1$ to N
     $V[i] \leftarrow \theta_i' + \sqrt{16 \cdot \frac{q_i - t_i \theta_i'^2}{t_i - 1} \cdot \frac{t-1}{t_i}}$
                    ▷ Expected reward.
   end loop
   return [MAX(V), MAXINDEX(V)]
end function

For example, the exploration function in Equation 19 is

$$\xi = \int p(x_{s(1)}) V^* \left(\boldsymbol{\theta}|x_{s(t)}(1), \boldsymbol{\Sigma}|x_{s(t)}(1), T-1\right) dx \quad (27)$$

of which the computation is expensive due to recursive calling and integral. In this section we present two approximate methods.

3.4.1 Value Iteration with Monte Carlo Sampling

For $T \geq 3$ the solution for N inequalities cannot be obtained easily. Instead we use Monte Carlo sampling to deal with the integral and avoid solving the inequalities. The exploration function is written as

$$\xi_{\text{VI-COR}} = \frac{1}{M_0} \sum_{x \in \mathbb{S}} p(x) V^* \left(\boldsymbol{\theta}|x, \boldsymbol{\Sigma}|x, T-1\right) dx \quad (28)$$

where $\mathbb{S}$ is the sample set and M_0 is the normalizing factor from sampling. The algorithm for a general $T \geq 3$ case is represented in Algorithm 1. We refer to this algorithm as VI-COR in our experiments.

3.4.2 The UCB1-Normal-COR Algorithm

The multi-armed bandit is a popular approach dealing with exploration-exploitation dilemma in sequential optimization problems [4]. Similar to the problem discussed

above, in the multi-armed bandit scenario a player must decide which arm to play at each time step to maximize the cumulative reward over the entire planning horizon.

Most multi-armed bandit algorithms reduce the computational cost by approximating the exploration function. In this paper we base on the deterministic policy UCB1-NORMAL [4] and improve its performance by adding the correlated update. The original algorithm assumes the Gaussian distribution of the reward, independence between arms, and underlying mean and variance for reward distribution are unknown but fixed. The exploration function of UCB1-NORMAL is written as

$$\xi_{\text{UCB1-NORMAL}} = \sqrt{16 \cdot \frac{q_i - t_i \theta_i^2(t)}{t_i - 1} \cdot \frac{t-1}{t_i}} \qquad (29)$$

where q_i is the sum of squared reward obtained from i-th arm, and t_i the times i-th has been played so far. We extend the algorithm for our problem by adding the correlated updates. The exploration function would remain the same form, but $\boldsymbol{\theta}(t)$ at each time step is updated according to Equations 17 and 18, instead of only updating the selected candidate. The algorithm, referred to as UCB1-NORMAL-COR in our experiments, is represented in Algorithm 2.

4. EMPIRICAL EXPERIMENTS

4.1 Dataset

We collected our test data set from Google AdWords service [2]. We consider the scenario that advertisers deploy campaigns through an advertiser (demand) side platform, whereas online publishers retrieve ads and earn revenue from a related publisher side platform. Generally, online publishers share the ad revenues with their chosen ad networks or exchanges with a fixed percentage. For instance, with Google AdSense online publishers gain 68% of advertisers' spending, where the ratio has not changed since 2003 [22]. Thus, it is reasonable for us to consider online publishers' ad revenue to be proportional to the corresponding revenue of Google AdWords and also the corresponding cost of the advertisers.

The test data was collected from 12/2011 to 5/2012[1]. The Google AdWords Traffic Estimation service provides real-time data to help advertisers to adjust budgets and select appropriate keywords. When given a keyword, budget, and various targets, the service will return a list of fields on a daily basis including clicks, global and local impressions, average position, average cost-per-click, and total cost. In addition, we also separated the US and UK markets using the geographical targeting option of the service. We have collected keywords across the Google Sponsored Search and Display Networks and do not make a distinction between the types of ads and their specific payment schemes. This allows online publishers to fully concentrate upon the revenue optimisation task. During the collection period 521 different keywords from various categories were collected and 310 have non-zero mean payoffs. As shown in Table 1, we manually categorized the keywords into 8 categories to reflect that fact that it is not necessary for online publishers to try out all the available ads; instead they should specify their target categories (based on hosting webpages content) and

---

[1]The dataset is publicly available at http://www.computational-advertising.org.

Category	Count	Category	Count
Education	34	Person & organization	29
Shopping	86	(Personal) finance	51
Product & service	20	Information	52
Medical	25	Uncategorised	13

Table 1: Categorisation of the collected 310 keywords with non-zero mean payoffs.

make optimal decisions within the targeted categories. In our experiments, ads associated with the collected keywords were considered as candidates and decision making was on a daily basis.

4.2 Baselines and Experiment Setup

The baselines used in our experiment are:
- RANDOM policy, which selects candidates randomly (uniform);
- MYOPIC policy, which selects candidates based on immediate reward estimated so far;
- UCB1 policy, which assumes independent between candidates and is model-free of reward distribution [4]; and
- UCB1-NORMAL policy, which assumes independent between candidates and the reward following Gaussian distribution;

And our algorithms are:
- VI-COR policy, which uses value iteration with Monte Carlo sampling (Algorithm 1); and
- UCB1-NORMAL-COR policy, which consider the dependencies between candidates (Algorithm 2).

For each candidate we have about 150 daily payoff data points (some candidates have less due to later start of collection). For each category (except *Uncategorised*) we select candidates with close mean payoffs to form a dataset, emphasizing the challenge of planning with matching candidates. In order to test the statistical significance of algorithms performance, we divide daily payoff time series into 8 chunks for each dataset (with overlap). For each chunk, we use 20% as the training set to get the prior belief of ad performances, i.e. $\boldsymbol{\theta}(0)$ and $\boldsymbol{\Sigma}(0)$. Then we go over the remaining 80% with each algorithm reporting the cumulative reward at each time step. Besides, we report the averaged results with Wilcoxon signed-rank test for the significance of the best algorithm outperformed the second best within each dataset.

4.3 Results and Discussions

We compared our two algorithms with the baselines over the 10 selected datasets. In order to compare performances across categories we normalize the cumulative revenues against the GOLDEN solution (always picking up the best ads) to give a better representation due to the different scales of means from different categories. The results are summarized in Table 2 and are compared in Figure 2. We can see that, within the 10 different datasets, the proposed VI-COR algorithm performed the best for 8/10 with 5/8 significantly better. The UCB1-NORMAL-COR algorithm performed the best for 1/10 and was significantly better in that trail. With the "Shopping-1" dataset, the UCB1 algorithm performed the best, but the VI-COR had the comparable performance and the difference was not significant.

In Figure 2 we give daily performance comparison on "Education" and "People & Organization" datasets where algo-

Datasets	MYOPIC	RANDOM	UCB1	UCB1-NORMAL	VI-COR	UCB1-NORMAL-COR
Education	21.9	23.0	30.9	30.9	**41.2***	27.6
Finance-1	38.5	27.8	40.9	26.4	**44.5**	27.4
Finance-2	22.1	16.5	30.6	22.8	**38.0***	22.9
Information	14.1	12.9	27.8	15.9	**29.4**	15.9
People & Organization	41.6	30.4	50.5	31.4	**72.9***	63.3
Shopping-1	17.4	10.6	**42.3**	16.1	40.2	16.4
Shopping-2	29.9	14.5	34.3	75.3	52.9	**79.2***
Shopping-3	9.7	4.3	21.9	18.3	**27.3**	19.4
Product & Service	24.7	26.0	47.2	57.1	**67.9***	59.9
Medical	30.5	19.6	52.7	32.2	**58.0***	33.5

Table 2: The overall performance comparison. The cumulative payoffs are averaged on 8 chunks then normalized w.r.t. the GOLDEN policy for a better representation. In each row the one with highest cumulative payoff is in bold and with * if the difference with the second best is significant.

	eminem	justin bieber	michael jackson	selena gomez	wayne rooney
eminem	1.00	-0.43	-0.58	-0.50	-0.73
justin bieber		1.00	0.80	0.74	0.70
michael jackson			1.00	0.97	0.71
selena gomez				1.00	0.63
wayne rooney					1.00

Table 3: The sample correlation matrix for "People & Organization" category. The high correlations made the UCB1 and UCB1-Normal inefficient.

rithm were run on entire payoff data series. We cannot represent all 10 figures due to the space limit; the performances were however consistent across all datasets. We discuss our findings in the following subsections.

4.3.1 The Importance of Exploration

Firstly we study the importance of exploration. Figure 2 compares the daily accumulated revenues over time. As illustrated in Figure 2(a), in the beginning (between day-0 and day-10) the MYOPIC policy achieved excellent result and its cumulative payoff was the best until day-65. This is explained by the fact that there is no exploration involved in this policy and it exploits the current belief directly. However our algorithms with exploration quick outperformed it in the late stage as more profitable ads have been discovered from the exploration from the early stage. In the end, the MYOPIC policy failed to win due to no exploration in the beginning, and later stuck to suboptimal ads. It is similar in Figure 2(b) where the MYOPIC policy outperformed others between day-25 and day-35, but was caught up and passed very soon. These conclude that the exploration is valuable and important in the ad selection task. Note that we only show categories "Education" and "People & Organization" only, while the algorithms behaviours were consistent across all datasets.

4.3.2 The Effect of Correlation

Recall we consider the Gaussian noise constant for all webpages and users. The consequence is that X_1 and X_2 are conditional independent when we know μ_1 and μ_2. We can further derive the relationship between covariances of $X1$ and X_2 and that of μ_1 and μ_2 as,

$$\begin{aligned}
\mathrm{cov}[X_1, X_2] &= \mathbb{E}\big[\,\mathrm{cov}[X_1, X_2|\mu_1, \mu_2]\big] \\
&\quad + \mathrm{cov}\big[\mathbb{E}[X_1|\mu_1, \mu_2], \mathbb{E}[X_2|\mu_1, \mu_2]\big] \\
&= \mathrm{cov}[\mu_1, \mu_2] \qquad\qquad\qquad (30)
\end{aligned}$$

which enables publishers to use either of correlations within the model. In experiments we used $\mathrm{cov}[X_1, X_2]$.

Like most multi-bandit machine approaches, the UCB1 algorithm assumes independence between candidates. Therefore when candidates have relatively low correlations, e.g. in the "Shopping-1" dataset, the algorithm would perform well. This was confirmed by the observation that the UCB1-NORMAL-COR algorithm reported only 0.3% improvement over UCB1-NORMAL in that dataset. But in the situation where the correlation of ads are high such as the "People & Organization" category (a sample correlation matrix is shown in Table 3), our algorithms utilizing the correlation during planning showed better results. For instance the VI-COR algorithm showed 22.4% improvement over the UCB1 and the UCB1-NORMAL-COR showed 31.9% improvement over the UCB1-NORMAL in the "People & Organization" dataset. In Figure 2(b) it was clear that UCB1-NORMAL-COR discovered the better options much quicker than the UCB1-NORMAL did. The same conclusion is obtained by comparing the VI-COR with the MYOPIC solution. From Figure 3, we can see a significant improvement by utilizing correlation of ads. The maximum improvement was obtained with "Product & Service" (43.3%) and on average it was 22.2% across all experiments.

4.3.3 The Impact of the Noise Factor σ_0^2

The introduction of the noise factor σ_0^2 is essential to the success of our model. On one hand, it helps capture the uncertainty which is unforeseeable and has not been properly modelled by the underlying $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ as we discussed be-
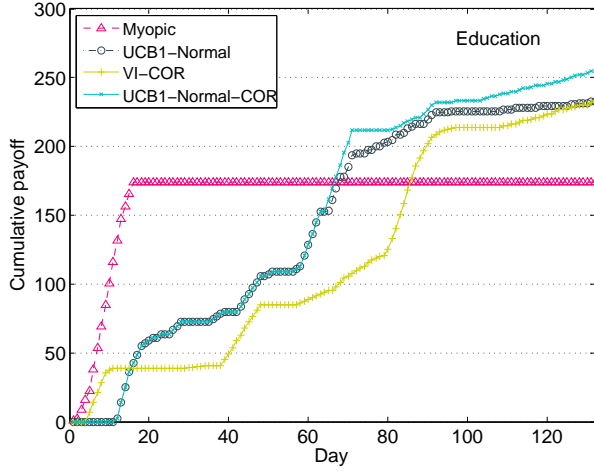
(a) "Education" category, 9 candidates.
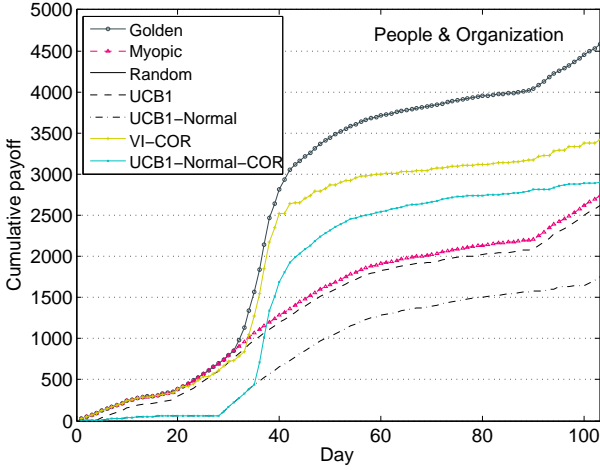
(b) "People & organization" category, 5 candidates.

Figure 2: Comparison on accumulated payoffs.

fore. The assumption about the Gaussian distribution may not be true in practice. This can be seen from Figure 4(a) and Figure 4(b). The noise factor provides us with a certain flexibility of tuning our algorithms towards the specific situations. In our experiments, we obtained the noise factor by tuning using training datasets.

On the other hand, the noise factor is also a control of the sensitivity of the algorithms towards the unexpected daily payoff fluctuation. From Equations 17 and 18 we see that the noise factor is in the denominator and a high noise factor leads to a steady policy while a low one leads to a highly responsive (sensitive) policy. A smaller value, for example $\sigma_0^2 = 0.01$, would make the algorithm switch probably too much, whereas a larger value would not be able to capture the fluctuation of the data. We found from the data that sometimes strong and short bursts may happen and be led by unexpected commercial activities. For instance, in the beginning of May, 2012 the Sumsang Galaxy S III and Nokia 808 PureView were presented for pre-ordering or purchasing, and both claimed to be the 'best' on the market. The competition of commercial campaigns caused the daily payoff of "best phones" became very high between 15/04/2012 and 05/05/2012 (Figure 4(b)). In order to response to such a short and strong abnormal activity a small noise factor is required. From Figure 4(c) we see that with $\sigma_0^2 \leq 40$ the
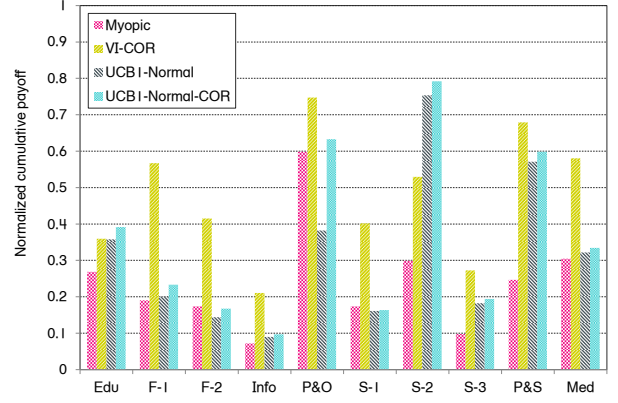

Figure 3: Comparison on accumulated payoffs on the 10 datasets. VI-COR always performed better than MYOPIC and UCB1-NORMAL-COR always performed better than UCB1-NORMAL across all datasets.

VI-COR algorithm was able to identify and switch to "best phones" when the burst happened; but with $\sigma_0^2 \geq 60$ the algorithm was not able to switch, resulting in a loss of payoff.

It is worth noticing that, as illustrated in Figure 4(c), the setting of the noise factor is algorithm-dependent as well. By contrast, the UCB1-NORMAL-COR approach requires a large noise value to deal with the burst – the best cumulative payoff was obtained at $\sigma_0^2 = 40$, and as the value of the noise factor decreased the performance dropped greatly. The different behaviour of two algorithms is due to the different structures of the exploration function. As shown in Equation 29, the exploration function of the UCB1-NORMAL-COR contains the squared expectation of the payoffs in the past, indicating that the candidate with a history of low payoffs would not be favoured especially with a sudden burst. Using a high noise value would increase the chance of selecting and exploring such candidates.

5. CONCLUSIONS

In this paper, we have presented a model of optimally selecting ads in an online setting. Based on POMDPs, we formulated the belief updates by taking the correlation of ads into account. We mathematically showed that the belief update across ads is similar to the "word of month" principle employed in collaborative filtering. Making use of the belief update, two approximate methods were proposed: one was derived from the Value Iteration and Sampling approach, whereas the other was based on the Upper Confidence Bound solution. In empirical experiments we compared our algorithms with various baselines using a collected real world dataset and showed that the Bayesian inference with correlations made the exploration more efficient and significantly improved the revenue optimization.

In the future, we intend to extend the basic model by considering the correlations between webpages as well to further improve the exploration. A further study is also needed in order to incorporate the user click-through model and remove rank bias on the same webpage [25].

6. REFERENCES

[1] The first half '11 internet advertising report. http://goo.gl/NRZ4S, 2011.
[2] ADWORDS, G. Google adwords traffic estimator. http://goo.gl/TYdOQ, 2012.

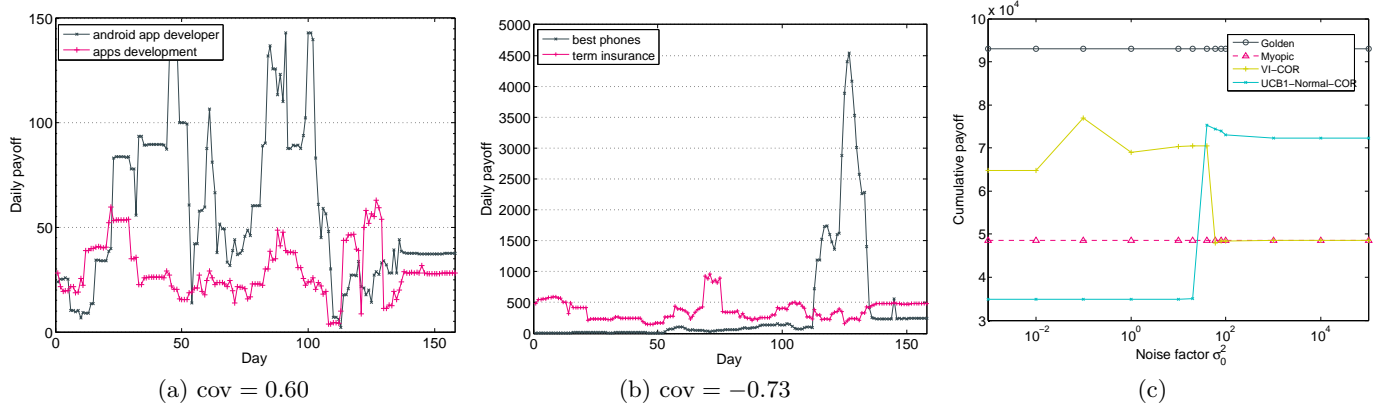(a) cov = 0.60   (b) cov = -0.73 (c)

Figure 4: (a) The daily payoffs of two mobile development related ad candidates. (b) The daily payoff of two candidates with a sudden change. (c) The impact of the noise factor σ_0^2 for the situation in (b).

[3] Anagnostopoulos, A., Broder, A., Gabrilovich, E., Josifovski, V., and Riedel, L. Just-in-time contextual advertising. In *Proceedings of the ACM-CIKM 07'.*

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

[5] Bertsekas, D. P. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995.

[6] Bharadwaj, V., Ma, W., Schwarz, M., Shanmugasundaram, J., Vee, E., Xie, J., and Yang, J. Pricing guaranteed contracts in online display advertising. In *Proceedings of the ACM-CIKM 10'.*

[7] Breese, J. S., Heckerman, D., and Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th conference on Uncertainty in artificial intelligence* (1998), UAI'98, pp. 43–52.

[8] Broder, A., Fontoura, M., Josifovski, V., and Riedel, L. A semantic approach to contextual advertising. In *Proceedings of the ACM-SIGIR 07'.*

[9] Bullas, J. 50 amazing facts and figures about google. http://goo.gl/EfrLS, 2011.

[10] Cassandra, A. A survey of POMDP applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes.*

[11] Chakrabarti, D., Agarwal, D., and Josifovski, V. Contextual advertising by combining relevance with click feedback. In *Proceedings of the WWW 08'.*

[12] Chickering, D., and Heckerman, D. Targeted advertising on the web with inventory management. *Interfaces* (2003), 71–77.

[13] Fjell, K. Online advertising: Pay-per-view versus pay-per-click a comment. *Journal of Revenue & Pricing Management 2*, 3 (2009), 200–206.

[14] Fridgeirsdottir, K., and Asadolahi, S. Revenue management for online advertising: Impatient advertisers. Tech. rep., Working paper, London Business School, 2007.

[15] Google White Paper. Profiting from non-guaranteed advertising: The value of dynamic allocation and auction pricing for online publishers, 2012.

[16] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. An algorithmic framework for performing collaborative filtering. In *Proceedings of the ACM-SIGIR '99.*

[17] Hu, Y., Shin, J., and Tang, Z. Pricing of online advertising: Cost-per-click-through vs. cost-per-action. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on.*

[18] Kumar, S., Jacob, V., and Sriskandarajah, C. Scheduling advertisements on a web page to maximize revenue. *European Journal of Operational Research 173*, 3 (2006), 1067–1089.

[19] Kwon, C. Single period balancing of pay-per-click and pay-per-view online display advertisements. *Journal of Revenue & Pricing Management 10*, 3 (2011), 261–270.

[20] Li, Y., and Jhang-Li, J. Pricing display ads and contextual ads: Competition, acquisition, and investment. *Electronic Commerce Research and Applications 8*, 1 (2009), 16–27.

[21] Mangani, A. Online advertising: Pay-per-view versus pay-per-click. *Journal of Revenue & Pricing Management 2*, 4 (2004), 295–302.

[22] Mohan, N. The adsense revenue share. http://goo.gl/yP4Ln, 2010.

[23] Pandey, S., Chakrabarti, D., and Agarwal, D. Multi-armed bandit problems with dependent arms. In *Proceedings of the ICML 07'.*

[24] Ribeiro-Neto, B., Cristo, M., Golgher, P., and Silva de Moura, E. Impedance coupling in content-targeted advertising. In *Proceedings of the ACM-SIGIR 05'.*

[25] Richardson, M., Dominowska, E., and Ragno, R. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the WWW '07.*

[26] Roels, G., and Fridgeirsdottir, K. Dynamic revenue management for online display advertising. *Journal of Revenue & Pricing Management 8*, 5 (2009), 452–466.

[27] Spaan, M., and Vlassis, N. Perseus: Randomized point-based value iteration for POMDPs. *Journal of artificial intelligence research 24*, 1 (2005), 195–220.

[28] Tang, J., Liu, N., Yan, J., Shen, Y., Guo, S., Gao, B., Yan, S., and Zhang, M. Learning to rank audience for behavioral targeting in display ads. In *Proceedings of the ACM-CIKM 11'.*

[29] Thrun, S. Monte carlo POMDPs. *Advances in neural information processing systems 12* (2000), 1064–1070.

[30] Turner, J., Scheller-Wolf, A., and Tayur, S. Scheduling of dynamic in-game advertising. *Operations research 59*, 1 (2011), 1–16.

[31] Tyler, S., Pandey, S., Gabrilovich, E., and Josifovski, V. Retrieval models for audience selection in display advertising. In *Proceedings of the ACM-CIKM 11'.*

[32] Wang, C., Zhang, P., Choi, R., and Eredita, M. Understanding consumers attitude toward advertising. In *The 8th Americas conference on information systems.*

[33] Wu, Z., Xu, G., Pan, R., Zhang, Y., Hu, Z., and Lu, J. Leveraging wikipedia concept and category information to enhance contextual advertising. In *Proceedings of the ACM-CIKM 11'.*