An Unbiased Offline Evaluation of Contextual Bandit Algorithms with Generalized Linear Models

Lihong Li LIHONG@YAHOO-INC.COM
Yahoo! Research
Santa Clara, CA, USA

Wei Chu [*] CHU.WEI@MICROSOFT.COM
Microsoft
Bellevue, WA, USA

John Langford JL@YAHOO-INC.COM
Yahoo! Research
New York, CA, USA

Taesup Moon TAESUP@YAHOO-INC.COM
Yahoo! Labs
Sunnyvale, CA, USA

Xuanhui Wang XHWANG@YAHOO-INC.COM
Yahoo! Labs
Santa Clara, CA, USA

Editor: Dorota Głowacka, Louis Dorard, John Shawe-Taylor

Abstract

Contextual bandit algorithms have become popular tools in online recommendation and advertising systems. *Offline* evaluation of the effectiveness of new algorithms in these applications is critical for protecting online user experiences but very challenging due to their "partial-label" nature. A common practice is to create a simulator which simulates the online environment for the problem at hand and then run an algorithm against this simulator. However, creating the simulator itself is often difficult and modeling bias is usually unavoidably introduced.

The purpose of this paper is two-fold. First, we review a recently proposed *offline* evaluation technique. Different from simulator-based approaches, the method is completely data-driven, is easy to adapt to different applications, and more importantly, provides provably unbiased evaluations. We argue for the wide use of this technique as standard practice when comparing bandit algorithms in real-life problems. Second, as an application of this technique, we compare and validate a number of new algorithms based on *generalized* linear models. Experiments using real Yahoo! data suggest substantial improvement over algorithms with linear models when the rewards are binary.

Keywords: Multi-armed bandit, contextual bandit, offline evaluation, generalized linear model, upper confidence bound

[*] The work was done when Wei Chu was with Yahoo! Labs.

© 2012 L. Li, W. Chu, J. Langford, T. Moon & X. Wang.

1. Introduction

Web-based recommendation and advertising services such as the Yahoo! Today Module (at `http://www.yahoo.com`) leverage user activities such as clicks to identify the most attractive contents. One inherent challenge is scoring newly generated content such as breaking news, especially when the news first emerges and little data are available. A personalized service which can tailor contents towards individual users is more desirable and challenging.

A distinct feature of these applications is their "partial-label" nature: we observe user feedback (click or not) for an article *only when* this article is displayed. Such a key challenge, known as the exploration/exploitation tradeoff, is commonly studied in the contextual bandit framework (Langford and Zhang, 2008) that has found successful applications; see, e.g., Agarwal et al. (2009), Graepel et al. (2010), Li et al. (2010), and Moon et al. (2010).

To evaluate a contextual-bandit algorithm reliably, it is ideal to conduct a bucket test, in which we run the algorithm to serve a fraction of live user traffic in the real recommendation system. However, not only is this method expensive, requiring substantial engineering efforts for deployment in the real system, but it can also have a negative impact on user experience. Furthermore, it is not easy to guarantee replicable comparison using bucket tests as online metrics vary significantly over time. *Offline* evaluation of contextual-bandit algorithms thus becomes valuable.

Although benchmark datasets for supervised learning such as the UCI repository have proved valuable for empirical comparison of algorithms, collecting benchmark data towards reliable offline evaluation has been difficult in bandit problems, as explained later in Section 3. The first purpose of the paper is to review a recently proposed evaluation method of Li et al. (2011), which enjoys valuable theoretical guarantees including unbiasedness and accuracy. The effectiveness of the method has also been verified by comparing its evaluation results to online bucket results using a large volume of data recorded from Yahoo! Front Page. Such positive results not only encourage wide use of the proposed method in other Web-based applications, but also suggest a promising solution to create benchmark datasets from real-world applications for bandit algorithms.

As one application, the next focus of the paper is to use this offline evaluation technique to validate a few new bandit algorithms based on generalized linear models or GLMs (McCullagh and Nelder, 1989). We argue that GLMs provide a better way to model average reward when the reward signal is binary, compared to the more widely studied linear models despite their strong theoretical guarantees (Auer, 2002; Chu et al., 2011). Our experiments with real Yahoo! data provide empirical evidence for the effectiveness of these new algorithms, and encourage future work on developing regret bound for them or their variants.

The rest of the paper is organized as follows. After reviewing preliminaries in Section 2, we review the offline evaluation technique in Section 3, including unbiasedness and sample complexity results. Section 4 develops algorithms with GLM-based reward models. These algorithms are inspired by existing ones for linear models, and are empirically validated in Section 5 using real data collected from Yahoo! Front Page. Finally, Section 6 concludes the paper. Since our papers consist of two major components, related work will be discussed in appropriate subsections.

2. Notation

The multi-armed bandit problem is a classic and popular model for studying the exploration-exploitation tradeoff (Berry and Fristedt, 1985). This paper considers the problems with contextual information. Following Langford and Zhang (2008), we call it a *contextual bandit problem*.

Formally, we define by $\mathcal{A} = \{1, 2, \ldots, K\}$ a set of K arms, and a contextual-bandit algorithm A interacts with the *world* in discrete trials $t = 1, 2, 3, \ldots$. In trial t:

1. The world chooses a feature vector $\mathbf{x}_t$ known as the *context*. Associated with each arm a is a real-valued reward $r_{t,a} \in [0, 1]$ that can be related to the context $\mathbf{x}_t$ in an arbitrary way. We denote by $\mathcal{X}$ the (possibly infinite) set of contexts, and $(r_{t,1}, \ldots, r_{t,K})$ the reward vector. Furthermore, we assume $(\mathbf{x}_t, r_{t,1}, \ldots, r_{t,K})$ is drawn i.i.d. from some unknown distribution D.
2. Based on observed rewards in previous trials and the current context $\mathbf{x}_t$, A chooses an arm $a_t \in \mathcal{A}$, and receives reward r_{t,a_t}. It is important to emphasize here that *no* feedback information (namely, the reward $r_{t,a}$) is observed for *unchosen* arms $a \neq a_t$.
3. The algorithm then improves its arm-selection strategy with all information it observes, $(\mathbf{x}_{t,a_t}, a_t, r_{t,a_t})$.

In this process, the *total T-trial reward* of A is defined as

$$G_{\mathsf{A}}(T) \stackrel{\text{def}}{=} \mathbf{E}_D \left[\sum_{t=1}^{T} r_{t,a_t} \right],$$

where the expectation $\mathbf{E}_D[\cdot]$ is defined w.r.t. the i.i.d. generation process of $(\mathbf{x}_t, r_{t,1}, \ldots, r_{t,K})$ according to distribution D (and the algorithm A as well if it is not deterministic). Similarly, given a policy π that maps contexts to actions, $\pi : \mathcal{X} \mapsto \mathcal{A}$, we define its total T-trial reward by

$$G_{\pi}(T) \stackrel{\text{def}}{=} \mathbf{E}_D \left[\sum_{t=1}^{T} r_{t,\pi(\mathbf{x}_t)} \right] = T \cdot \mathbf{E}_D \left[r_{1,\pi(\mathbf{x}_1)} \right],$$

where the second equality is due to our i.i.d. assumption. Given a reference set Π of policies, we define the *optimal expected T-trial reward with respect to* Π as

$$G^*(T) \stackrel{\text{def}}{=} \max_{\pi \in \Pi} G_{\pi}(T).$$

For convenience, we also define the per-trial reward of an algorithm or policy, which is defined, respectively, by

$$\begin{aligned} g_{\mathsf{A}} &\stackrel{\text{def}}{=} \frac{G_{\mathsf{A}}(T)}{T} \\ g_{\pi} &\stackrel{\text{def}}{=} \frac{G_{\pi}(T)}{T} = \mathbf{E}_D \left[r_{1,\pi(\mathbf{x}_1)} \right]. \end{aligned}$$

In the example of news article recommendation, we may view articles in the pool as arms, and for the t-th user visit (trial t), one article (arm) is chosen to serve the user. When the served article is clicked on, a reward of 1 is incurred; otherwise, the reward is 0. With this definition of reward, the expected reward of an article is precisely its *click-through rate (CTR)*, and choosing an article with maximum CTR is equivalent to maximizing the expected number of clicks from users, which in turn is the same as maximizing the total expected reward in our bandit formulation.

3. Unbiased Offline Evaluation

Compared to machine learning in the more standard supervised learning setting, evaluation of methods in a contextual bandit setting is frustratingly difficult. In our application of news article recommendation, for example, each user visit results in the following information stored in the log: user information, the displayed news article, and user feedback (click or not). When using data of this form to evaluate a bandit algorithm offline, we will *not* have user feedback if the algorithm recommends a different news article than the one stored in the log. In other words, data in bandit-style applications only contain user feedback for recommended news articles that were actually displayed to the user, but not undisplayed ones. This "partial-label" nature raises a difficulty that is the key difference between evaluation of bandit algorithms and supervised learning ones.

Common practice for evaluating a bandit algorithm is to create a simulator and then run the algorithm against it. With this approach, we can evaluate any bandit algorithm without having to run it in a real system. Unfortunately, there are two major drawbacks with this approach. First, creating a simulator can be challenging and time-consuming for practical problems. Second, evaluation results based on artificial simulators may not reflect the actual performance since simulators are only rough approximations of real problems and unavoidably contains modeling bias. In fact, building a high-quality simulator can be strictly harder than building a high-quality policy (Strehl et al., 2006).

The goal here is to measure the total reward of a *bandit algorithm* A. Because of the interactive nature of the problem, it would seem that the only way to do this unbiasedly is to actually run the algorithm online on "live" data. However, in practice, this approach is likely to be infeasible due to the serious logistical challenges that it presents. Rather, we may only have *offline* data available that was collected at a previous time using an entirely *different* logging policy. Because rewards are only observed for the arms chosen by the logging policy, which are likely to differ from those chosen by the algorithm A being evaluated, it is not at all clear how to evaluate A based only on such logged data. This evaluation problem may be viewed as a special case of the so-called "off-policy evaluation problem" in the reinforcement-learning literature (Precup et al., 2000).

In this section, we summarize our previous work (Li et al., 2011) on a *sound* technique for carrying out such an evaluation. Interested readers are referred to the original paper for more details. The key assumption of the method is that the individual events are i.i.d., and that the logging policy chose each arm at each time step uniformly at random. Although we omit the details, this latter assumption can be weakened considerably so that any randomized logging policy is allowed and the algorithm can be modified accordingly using rejection sampling, but at the cost of decreased data efficiency. Furthermore, if A is a stationary policy that does not change over trials, data may be used more efficiently via propensity scoring (Langford et al., 2008; Strehl et al., 2011) and related techniques like doubly robust estimation (Dudík et al., 2011).

Formally, algorithm A is a (possibly randomized) mapping for selecting the arm a_t at time t based on the history h_{t-1} of $t-1$ preceding events together with the current context. Algorithms 1 and 2 give two versions of the evaluation technique, one assuming access to a sufficiently long sequence of logged events resulting from the interaction of the logging policy with the world, the other assuming a fixed set of logged interaction. The method takes as input a bandit algorithm A. We then step through the stream of logged events one by one. If, given the current history h_{t-1}, it happens that the policy A chooses the same arm a as the one that was selected by the logging policy, then the event is retained (that is, added to the history), and the total reward $\hat{G}_{\mathsf{A}}$ updated. Otherwise, if the policy A selects a different arm from the one that was taken by the logging policy, then the event is entirely ignored, and the algorithm proceeds to the next event without any change in its state. The process repeats until h_T is reached (Algorithm 1), or until data is exhausted (Algorithm 2).

Note that, because the logging policy chooses each arm uniformly at random, each event is retained by this algorithm with probability exactly $1/K$, independent of everything else. This means that the events which are retained have the same distribution as if they were selected by D. The

Algorithm 1 `Policy_Evaluator` (with infinite data stream).

```
Inputs: $T > 0$; bandit algorithm $\mathsf{A}$; stream of events $S$
$h_0 \leftarrow \emptyset$ {An initially empty history}
$\hat{G}_{\mathsf{A}} \leftarrow 0$ {An initially zero total reward}
for $t = 1, 2, 3, \ldots, T$ do
    repeat
        Get next event $(\mathbf{x}, a, r_a)$ from $S$
    until $\mathsf{A}(h_{t-1}, \mathbf{x}) = a$
    $h_t \leftarrow$ CONCATENATE$(h_{t-1}, (\mathbf{x}, a, r_a))$
    $\hat{G}_{\mathsf{A}} \leftarrow \hat{G}_{\mathsf{A}} + r_a$
end for
Output: $\hat{G}_{\mathsf{A}}/T$
```

Algorithm 2 `Policy_Evaluator` (with finite data stream).

```
bandit algorithm $\mathsf{A}$; stream of events $S$ of length $L$
$h_0 \leftarrow \emptyset$ {An initially empty history}
$\hat{G}_{\mathsf{A}} \leftarrow 0$ {An initially zero total reward}
$T \leftarrow 0$ {An initially zero counter of valid events}
for $t = 1, 2, 3, \ldots, L$ do
    Get the $t$-th event $(\mathbf{x}, a, r_a)$ from $S$
    if $\mathsf{A}(h_{t-1}, \mathbf{x}) = a$ then
        $h_t \leftarrow$ CONCATENATE$(h_{t-1}, (\mathbf{x}, a, r_a))$
        $\hat{G}_{\mathsf{A}} \leftarrow \hat{G}_{\mathsf{A}} + r_a$
        $T \leftarrow T + 1$
    else
        $h_t \leftarrow h_{t-1}$
    end if
end for
Output: $\hat{G}_{\mathsf{A}}/T$
```

unbiasedness guarantee thus follows immediately. Hence, by repeating the evaluation procedure multiple times and then averaging the returned per-trial rewards, we can accurately estimate the total per-trial reward g_{A} of any algorithm A and respective confidence intervals. Furthermore, as the size of data L increases, the estimation error of Algorithm 2 decreases to 0 at the rate of $O(1/\sqrt{L})$. This error bound improves a previous result (Langford et al., 2008) for a similar offline evaluation algorithm and similarly provides a sharpened analysis for the $T = 1$ special case for policy evaluation in reinforcement learning (Kearns et al., 2000). Details and empirical support of the evaluation method are found in our full paper (Li et al., 2011).

In summary, the unbiased offline evaluation technique provides a reliable method for collecting benchmark data so as to evaluate and compare different bandit algorithms, which is not available before. The first such benchmark has been released to the public (Yahoo!, 2011). Moreover, the technique is quite general; it has been successfully applied to domains like ranking (Moon et al., 2010) as well as in bandit problems with multiple objectives (Agarwal et al., 2011).

4. Algorithms based on Generalized Linear Models

As an application of the offline evaluation method, we will compare and validate a number of bandit algorithms based on different reward models. This section describes the models and the corresponding parameter-update and arm-selection rules.

Given context $\mathbf{x}$, we predict the expected reward for arm a using a generalized linear model (McCullagh and Nelder, 1989): $\hat{r}_a(\mathbf{x},\mathbf{w}) = g^{-1}(\mathbf{x}\cdot\mathbf{w}_a)$, where g is a link function. Three instantiations were tried that correspond to the linear, logistic, and probit models, respectively:

$$\hat{r}_a(\mathbf{x},\mathbf{w}_a) = \begin{cases} \mathbf{x}\cdot\mathbf{w}_a & \text{linear} \\ (1+\exp(-\mathbf{x}\cdot\mathbf{w}_a))^{-1} & \text{logistic} \\ \Phi(\mathbf{x}\cdot\mathbf{w}_a) & \text{probit} \end{cases} \tag{1}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard Gaussian distribution.

4.1. Model Fitting

A brief, unified review in a probabilistic framework for fitting these models is given as follows. We specify as the prior a Gaussian distribution, $N(\mu_0,\Sigma_0)$, with mean μ_0 and covariance Σ_0 on the weight vector $\mathbf{w}_a$ of arm a. Each of the three instantiations of GLM in Equation (1) corresponds to a specific likelihood function—the probability of observing a reward upon choosing action a, conditioned on the current context $\mathbf{x}$ and a given value of $\mathbf{w}_a$. At each sample with observed reward, we apply Bayes' theorem to update the posterior distribution of $\mathbf{w}_a$. If the posterior cannot be calculated exactly, a Gaussian approximation is used.

For the linear model, the likelihood function is Gaussian, so the posterior is exactly a Gaussian, denoted by $N(\mathbf{w}_a;\mu_a,\Sigma_a)$. The Gaussian likelihood assumption is clearly a mismatch in applications where the reward signal is binary (for example, 1 for click and 0 for no-click). Such a drawback is remedied by the logistic and probit models, whose likelihood function places probability mass to only two possible outcomes, although the posterior distribution does not have a closed-form solution. To approximate the posterior distributions, we carry out the Laplace approximation for logistic models, and the assumed density filtering or expectation propagation (Lawrence et al., 2002; Minka, 2001) for probit. Details of these approximations are given in Appendices A and B, respectively.

To summarize, in all three cases, we always maintain the posterior of $\mathbf{w}_a$ as represented by a Gaussian distribution, which will serve as the prior distribution for the next update. We next turn to the questions of online exploration/exploitation with these posterior distributions. To simplify notation, the expectation $\mathbf{E}[\cdot]$ and variance $\mathbf{Var}[\cdot]$ are both defined with respect to the posterior.

4.2. Exploitation

Given the (approximate) posterior distributions of all arms, $\{N(\mathbf{w}_a;\mu_a,\Sigma_a)\}_{a\in\mathcal{A}}$, if one is interested in exploitation only, it is natural to choose a greedy arm; namely, the arm with maximum *expected* reward:

$$\arg\max_a \mathbf{E}_{\mathbf{w}_a}[\hat{r}_a(\mathbf{x},\mathbf{w}_a)] \overset{\text{def}}{=} \arg\max_a \int \hat{r}_a(\mathbf{x},\mathbf{w}_a)N(\mathbf{w}_a;\mu_a,\Sigma_a)d\mathbf{w}_a. \tag{2}$$

For linear and probit models, the posterior mean above can be calculated in closed form:

$$\mathbf{E}_{\mathbf{w}_a}[\hat{r}_a(\mathbf{x},\mathbf{w}_a)] \quad = \quad \begin{cases} m & \text{linear} \\ \Phi\left(\frac{m}{\sqrt{1+v}}\right) & \text{probit} \end{cases}$$

where $m \overset{\text{def}}{=} \mathbf{x}\cdot\mu_a$ and $v \overset{\text{def}}{=} \mathbf{x}^\top\Sigma_a\mathbf{x}$ are the mean and variance of the quantity $\mathbf{x}\cdot\mathbf{w}_a$.

For the logistic model, however, approximation is necessary. Using various approximation techniques (see Appendix C), a few candidates are reasonable and will be compared against each other in the next section:

$$\mathbf{E}_{\mathbf{w}_a}[\hat{r}_a(\mathbf{x},\mathbf{w}_a)] \quad \approx \quad \begin{cases} (1+\exp(-m))^{-1} & \text{M0} \\ \left(1+\exp\left(-m/\sqrt{1+\pi v/8}\right)\right)^{-1} & \text{M1} \\ (1+\exp\left(-m-v/2\right))^{-1} & \text{M2} \\ \exp\left(m+v/2\right) & \text{M3} \end{cases}$$

4.3. Balancing Exploration and Exploitation

Choosing arms according to the exploitation rule Equation (2) is desired for maximizing total rewards, but at the same time risky: the lack of exploration may prevent collection of data to correct initial errors in parameter estimation. This section discusses a few candidates for online tradeoffs of exploration and exploitation.

A generic heuristic is ϵ-greedy, in which one chooses a greedy arm (according to Equation (2)) with probability $1-\epsilon$ and a random arm otherwise. This heuristic is simple, completely general, and can be combined with essentially with any reward models. Unfortunately, due to the *unguided*, uniformly random selection of arms for exploration, it is often not the best one can do in practice. For example, the closely related epoch-greedy algorithm (Langford and Zhang, 2008) can only guarantee $\tilde{O}(T^{2/3})$ regret for stochastic bandits, while guided exploration can do significantly better with $\tilde{O}(\sqrt{T})$ even at the presence of an adversary (Auer et al., 2002b; Beygelzimer et al., 2011).

In contrast, UCB-based exploration techniques are explicitly *guided* towards arms with uncertain reward predictions. They are found effective in previous studies (Auer, 2002; Auer et al., 2002a; Dorard et al., 2009; Li et al., 2010; Chu et al., 2011). In the context of present work, calculating the UCB is convenient since we maintain the posterior of each $\mathbf{w}_a$ explicitly. Analogous to previous algorithms for linear models, a UCB exploration rule chooses arms with a maximum upper confidence bound of the expected reward, given the parameter posterior $N(\cdot \mid \mu_a, \Sigma_a)$ and context:

$$\arg\max_a \bar{r}_a(\mathbf{x},\mathbf{w}_a,\alpha) \overset{\text{def}}{=} \arg\max_a \mathbf{E}_{\mathbf{w}_a}[\hat{r}_a(\mathbf{x},\mathbf{w})] + \alpha\sqrt{\mathbf{Var}_{\mathbf{w}_a}\left[\hat{r}_a(\mathbf{x},\mathbf{w}_a)\right]} \tag{3}$$

with a possibly slowing growing parameter $\alpha \in \mathbb{R}_+$. We call this arm selection α-*UCB*.

For linear and probit models, the calculation of $\bar{r}_a$ again can be done in closed form:

$$\bar{r}_a(\mathbf{x},\mathbf{w}_a) \quad = \quad \begin{cases} m+\alpha\sqrt{v} & \text{linear (Dorard et al., 2009; Li et al., 2010)} \\ \Phi\left(m+\alpha\sqrt{v}\right) & \text{probit} \end{cases}$$

where $m \overset{\text{def}}{=} \mathbf{x}\cdot\mu_a$ and $v \overset{\text{def}}{=} \mathbf{x}^\top\Sigma_a\mathbf{x}$ are the mean and variance of the quantity $\mathbf{x}\cdot\mathbf{w}_a$. In the case of probit, since Φ is monotonic, the greedy arm with respect to $\Phi(m+\alpha\sqrt{v})$ is thus the same as the greedy arm with respect to $m + \alpha\sqrt{v}$, which coincides with the linear model. But it should be emphasized that the posterior mean and covariance in these two models are different, since they are updated using different likelihood functions.

For logistic models, calculating $\bar{r}_a$ requires approximation. A few candidates are summarized below; see Appendix C for details:

$$\bar{r}_a(\mathbf{x}, \mathbf{w}_a) \quad = \quad \begin{cases} \left(1+\exp(-m-\alpha\sqrt{v})\right)^{-1} & \text{U0} \\ \left(1+\exp(-m-\alpha\sqrt{v})\right)^{-1} + \alpha\sqrt{\max\{0, V\}} & \text{U1} \\ \mathbf{E}_{\mathbf{w}_a}[\hat{r}_a(\mathbf{x}, \mathbf{w}_a)]\left(1+\alpha\sqrt{\exp(v)-1}\right) & \text{U2} \\ \left(1+\exp(-m)\right)^{-1} + \alpha\sqrt{v} & \text{U3} \end{cases}$$

where $\mathbf{E}_{\mathbf{w}_a}[\hat{r}_a(\mathbf{x}, \mathbf{w}_a)]$ is calculated using any of the approximations in the previous subsection, and

$$V \stackrel{\text{def}}{=} \Phi\left(\frac{\pi m/\sqrt{8}-1}{\sqrt{\pi-1+\pi^2 v/8}}\right) - \left(1+\exp\left(-m/\sqrt{1+\pi v/8}\right)\right)^{-2}$$

5. Experiments

We provide an empirical comparison of the three CTR models described in the previous section. The purpose is to validate the use of the generalized linear models and the respective algorithms although their current theoretical guarantees are not as strong as those for linear models.

5.1. Setup

As required by the evaluation method in Section 3, we collected $34M$ data in the form of $\{(\mathbf{x}, a, r_a)\}$ from a random bucket in the Today Module of Yahoo! Front Page for a one-week period in 2009. On average, each session has about $K = 20$ articles. A similar data set is available through the Webscope program of Yahoo! (2011).

The context was defined as follows. Naturally, a context corresponded to the user in that particular visit. Each user was represented by a raw feature vector of over 1000 categorical components, including demographic/geographic information as well as behavioral categories that summarize the user's consumption history within Yahoo! properties (Li et al., 2010). These features are highly sparse, so direct use of them introduces difficulty in learning and also increases computation complexity. Common approaches are to construct a lower dimensional subspace of features and then work with the new feature representations; see Chu et al. (2009) for an example. Here, we used the standard technique of principal component analysis to reduce features, which is simpler and did not appear to affect online learning performance of bandit algorithms. In particular, we performed a principal component analysis and identified $\mathbf{x}$ as the projection of the raw user feature onto the first 20 principal components, together with a constant feature. We then ended up with context $\mathbf{x}$ of dimension $d = 21$, which was used in the three models. Note that the constant feature serves as the bias term and thus was useful.

Each of the models was combined with two schemes of exploration: ϵ-greedy and upper confidence bound (UCB). Three values of ϵ were tried: $\epsilon \in \{0, 0.02, 0.05, 0.1, 0.15\}$; the case $\epsilon = 0$ corresponded to a purely greedy scheme.[1] The UCB schemes all chose an arm with the largest

[1] We did not try ϵ-greedy with decaying ϵ mainly because the set of arms is dynamic—new articles may be added to the pool while old ones may retire. The change of the arm set is completely asynchronous, so a global decaying

upper-confidence-bound score as defined in Equation (3), where $\alpha \in \{0.2, 0.5, 1, 2, 5\}$ is kept constant.[2] For each algorithm, we subsampled from all data with ratio 50%, and repeated the evaluation process 5 times so that statistics like mean and standard deviation could be obtained.

To evaluate a bandit algorithm, we are interested in two CTRs, following Li et al. (2010). When deploying a bandit algorithm in a large-scale real system like Yahoo! Front Page, one reasonable way is to randomly split all traffic into two buckets (Agarwal et al., 2009): the "learning bucket" usually consists of a small fraction of traffic on which various bandit algorithms are run to learn/estimate article CTRs; the "deployment bucket" is where users are served by articles with highest CTR estimates. The separation of two buckets ensures overall stability and hence is desirable for user experience purposes. Obviously, the learning bucket is where a normal bandit algorithm is run to select arms, so a higher CTR in this bucket implies a better tradeoff between exploration and exploitation; the deployment bucket is where a greedy (or exploit-only) algorithm is run with exploration turned off, so a higher CTR in this bucket indicates a better *greedy* policy can be derived. Related to our two-bucket metric is a model studied by Grünewälder et al. (2010), where the regret of algorithm in a "learning" phrase is ignored, and the algorithm strives to find the optimal policy at the end of the learning phrase. However, our "learning" and "deployment" buckets occur simultaneously.

In practice, completely real-time updates are not possible due to communication delays in large-scale software and network systems. We simulated this delay by updating the CTR models every 5 minutes (based on the recorded timestamp of the random traffic log on Today Module). Finally, to protect business-sensitive information, we have multiplied all absolute CTRs by a constant; the resulting number is called a *normalized* CTR, or *nCTR* for short.

5.2. A Comparison of Three Models

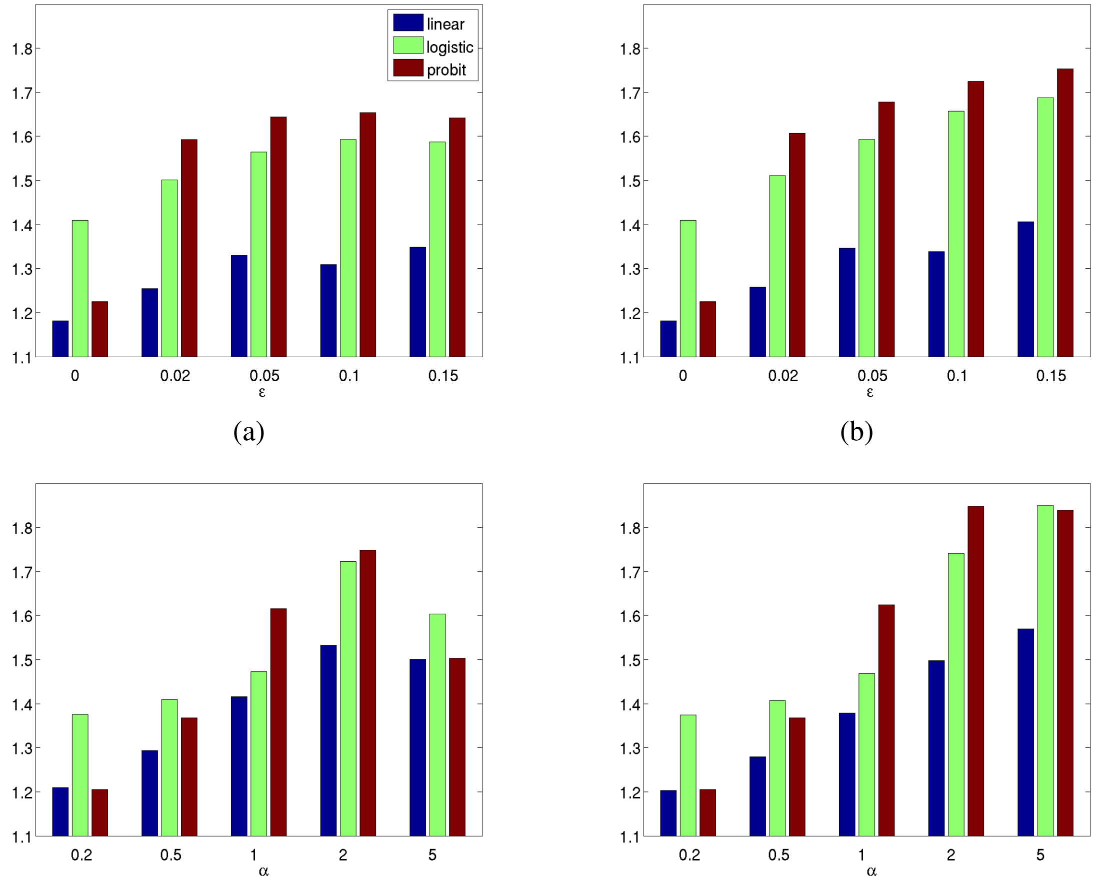
This subsection gives a first comparison of the three generalized linear models, each combined with ϵ-greedy and UCB exploration. We found it helpful to set the prior distribution adaptively: when a new article appears, its weight's prior is set so that μ_0 and Σ_0 are the average and empirical covariance of the posterior means of previously seen articles. Priors set this way can better capture common weight vectors and thus provide better performance. Figure 1 summarizes the normalized CTR in both the learning and deployment buckets. A few interesting observations are in order.

First, the nCTR in the learning bucket clearly demonstrates the need for active exploration. Purely exploiting strategies, those with $\epsilon = 0$ in ϵ-greedy exploration, all suffered significantly lower total rewards. On the other hand, increased amount of exploration (as ensured by higher values of ϵ and α) accelerates model parameter learning, as indicated by the monotonically increasing nCTR in the deployment bucket. However, too much exploration can decrease nCTR in the learning bucket, as seen for UCB exploration with large α values, although it does not necessarily hurt nCTR in the deployment bucket. Intermediate amount of exploration provides best tradeoff.

Second, all three models demonstrated nontrivial performance, suggesting appropriateness of generalized linear models in capturing CTR in Web applications. In contrast, the traditional, *non-contextual* ϵ-greedy and UCB algorithms do not consider user features. On the same data set,

scheme for ϵ is not straightforward. Furthermore, because of the constantly added new arms, the exploration rate cannot decay to 0, so the more complicated rule with decaying ϵ does not seem to differ much from the fixed ϵ one; on the other hand, each arm has a not-too-long life time before retiring, so the asymptotic advantage of adaptive ϵ may not apply.

[2] For similar reasons in Footnote 1, we compared the simpler choice of fixed α, although most theoretical analysis requires a slowly decreasing value of α (e.g., Auer et al. (2002a)).



(a)

(b)

(c)

(d)

Figure 1: A comparison of three generalized linear models with 50% subsamples of data. The plots contain nCTR for the learning bucket (left column) and the deployment bucket (right column) using ϵ-greedy (top row) and UCB (bottom row) combined. Numbers are averaged over 5 runs on random subsamples.

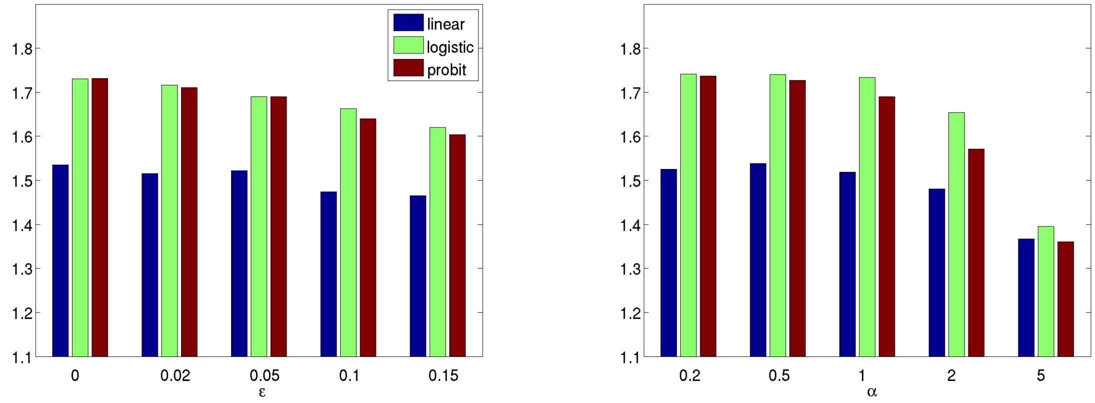
they can, at the best, achieve nCTR of 1.509 and 1.584 in the learning and deployment buckets, respectively. The results were consistent with our previous work (Li et al., 2010) although a different set of features were used.

Third, the logistic and probit models clearly outperform linear models, which is expected as their likelihood models better capture the binary reward signals. Since binary rewards are common in Web-based applications (like clicks, conversions, etc.), we anticipate the logistic and probit model to be more effective in general than linear models. That being said, with a large amount of data, the linear model may still be effective (Li et al., 2010; Moon et al., 2010), and remains a reasonable algorithmic choice given the simplicity of their closed-form update rules.

Fourth, UCB exploration remains effective and works better than ϵ-greedy, despite the lack of *general* theoretical guarantees and the necessity for numerical approximation. It is conjectured that the UCB heuristic is a generally effective exploration technique (when used appropriately), and encourage regret analysis beyond linear models.

Finally, our implementation maintained diagonal posterior covariance matrices of the parameters mainly because of computation reasons—manipulating full covariance matrices is usually too expensive in large-scale serving systems, even if closed-form updates exist for linear models (c.f., Section 4). We also tried full covariance matrix in logistic models, but did not find much improvement compared to the diagonal version. Precise descriptions of our implementation for the logistic and probit models are found in Sections A and B. For the linear model, diagonalized approximation works naturally in a similar way as in the logistic and probit models: when the reward for an arm a is observed in a context $\mathbf{x}$, the posterior variance for $\mathbf{w}_a$ is updated by $\Sigma_a \leftarrow \left(\Sigma_a^{-1} + \mathrm{diag}\left(\mathbf{x}\mathbf{x}^\top\right)\right)^{-1}$; clearly, this approximate update takes $\Theta(\|\mathbf{x}\|_0)$ time and remains efficient.

5.3. On the Effectiveness of Optimistic Initialization



(a)

(b)

Figure 2: A comparison of three generalized linear models using optimistic priors with 50% subsamples of data. The plots contain nCTR for the learning bucket using ϵ-greedy (left) and UCB (right). Numbers are averaged over 5 runs on random subsamples.

In addition to the popular ϵ-greedy and UCB exploration, optimistic initialization (Sutton and Barto, 1998) is a simple alternative that sometimes works well in practice. This subsection demonstrates the simplicity and effectiveness of this heuristic when combined with generalized linear models. Our solution was to use an *optimistic prior* — instead of setting the prior using posterior means of previously observed articles, we set the prior to a fixed one that always led to an over-estimate of the article CTR.

For logistic and probit models, the simple prior $N(\mathbf{0}, \mathbf{I})$ suffices,[3] since the prior CTR of a new article has a lot of probability mass around $g^{-1}(\mathbf{x} \cdot \mathbf{0}) = g^{-1}(0) = 0.5$, which is consistently bigger

[3] The use of $\mathbf{I}$ as the prior covariance is somewhat arbitrary, and may be improved with a more carefully chosen one.

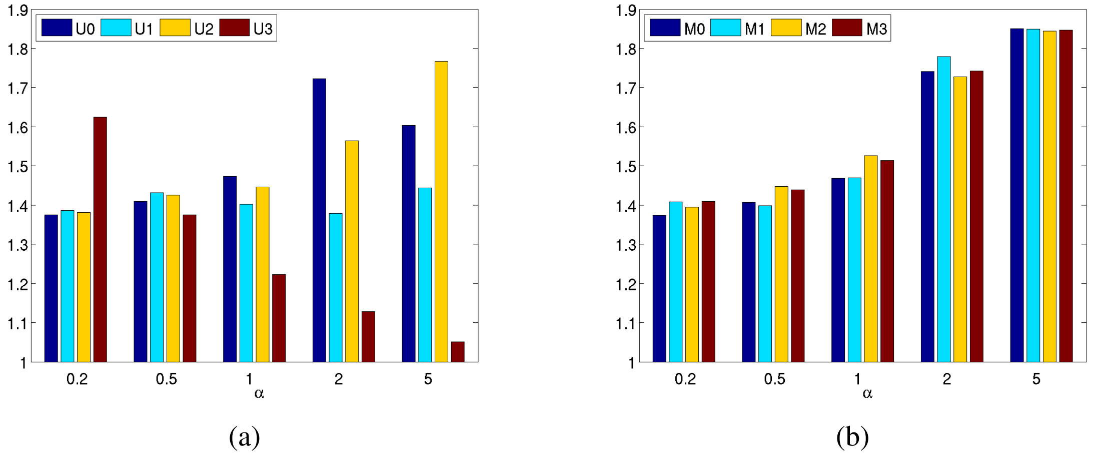


Figure 3: A comparison of different approximations for the posterior mean and posterior UCB in logistic models: (a) learning bucket nCTR with various UCB approximations; (b) deployment bucket nCTR with various poster mean approximations and with U0 for exploration.

than the (unnormalized) CTRs in our problem. But this prior is *not* optimistic for linear models since the probability mass centers around 0. To fix this problem, we changed the prior mean and variance for the constant feature's weight to 0.5 and 0.01, respectively.

Figure 2 plots the nCTR in the learning bucket when the optimistic initialization is used. We omit the deployment bucket's nCTR since they are all similar to the leftmost ones in Figure 1 (b,d). The results show that optimistic initialization alone is effective enough without explicit exploration like ϵ-greedy or UCB. Indeed, the nCTR was highest when such explicit exploration was turned off, that is, when $\epsilon = \alpha = 0$.

In contrast, explicit exploration was still necessary for linear models if the non-optimistic prior $N(\mathbf{0}, \mathbf{I})$ was used: when $\epsilon = \alpha = 0$, mean nCTR in the learning bucket was as low as 1.258 (with standard deviation 0.061), compared to 1.535 (with standard deviation 0.018) in Figure 2. Hence, we conclude that non-optimistic initialization without explicit exploration may result in convergence to suboptimal policies, and that optimistic initialization can indeed be effective in practice.

5.4. Approximations in the Logistic Model

Given the highly promising performance of logistic models, we next investigate how effective the various approximations are. Figure 3 (a) compares the learning bucket's nCTR with different approximations of the UCB score. Here, the formula for exploitation is irrelevant, so we do not show deployment nCTR. The results show that U0, U2, and U3 can be effective with appropriately tuned parameter α, but U1 is not as satisfactory. It also suggests the UCB rule (U3) given by Filippi et al. (2011) is too conservative, and consequently the best α value is very small.

Figure 3 (b) compares the deployment bucket's nCTR with different approximations of the posterior mean; the UCB formula was fixed to U0. Here, the learning bucket performance is irrelevant as we are comparing ways to compute the posterior mean while following the same exploration strategy of U0. No clear pattern is observed; all four approximation seemed to work equally well. An explanation is that the different approximations do not affect the $\arg\max$ result even if their approximation errors differ slightly.

6. Conclusions and Future Work

This paper reviews an offline evaluation method of bandit algorithms that relies on log data directly without building a reward simulator. The only requirement is that the log data is generated i.i.d. with arms chosen by an (ideally uniformly) random policy. The evaluation method gives unbiased estimates of quantities like total rewards. Extensions to non-random data are also possibly, making use of techniques like propensity scoring (Langford et al., 2008; Strehl et al., 2011) and doubly robust estimation (Dudík et al., 2011). The first benchmark is now available through the Webscope program of Yahoo! (2011).

Armed with such reliable evaluation methodology, we provide an extensive empirical comparison of several contextual bandit algorithms using a large volume of real data collected from Yahoo! Front Page. The algorithms are based on generalized linear models that can provide better modeling capability than linear models in many important applications. Our experiments validate this conjecture, illustrate how to combine popular exploration heuristics with GLMs, and suggest the possibility to tighten the state-of-the-art regret analysis of Filippi et al. (2011). These findings also corroborate the usefulness of our evaluation methodology, which we believe should be adopted as standard practice when comparing bandit algorithms with real-life data.

Acknowledgements

We thank Deepak Agarwal, Olivier Chapelle, and Pradheep Elango for helpful discussion regarding numerical approximations and implementations for the logistic model. We are also grateful to an anonymous reviewer for constructive comments that improved the presentation of the paper.

Appendix A. Approximate Inference in Logistic Regression

In logistic regression, the posterior distribution of the weight vector $\mathbf{w}$ is proportional to the product of the logistic likelihood and the Gaussian prior distribution:

$$p(\mathbf{w}) \propto \left(1 + \exp(-y\mathbf{x}^\top \mathbf{w})\right)^{-1} \cdot N(\mathbf{w}; \mu_t, \Sigma_t), \tag{4}$$

where $\mathbf{x}$ denotes a training sample with label $y \in \{\pm 1\}$. In our experiments, a click signal $c \in \{0, 1\}$ has to be converted to a binary label through $2c - 1$.

Instead of obtaining Equation (4) directly, we first approximate the posterior distribution of $\mathbf{w}^\top \mathbf{x}$, then obtain the posterior mean and variance of $\mathbf{w}$. That is, we first consider

$$p(\mathbf{w}^\top \mathbf{x}) \propto \left(1 + \exp(-y\mathbf{x}^\top \mathbf{w})\right)^{-1} \cdot \exp\left(-(\mathbf{w}^\top \mathbf{x} - m)^2/v\right), \tag{5}$$

where $m = \mu_t^\top \mathbf{x}$ and $v = \mathbf{x}^\top \Sigma_t \mathbf{x}$. By changing the variable to $\xi = (\mathbf{w}^\top \mathbf{x} - m)/\sqrt{v}$, and applying Laplace approximation on (5), we obtain the posterior mean and variance of ξ by computing the mode and the Hessian at the mode:

$$\mathbf{E}_{t+1}[\xi] = \hat{\xi}_{t+1} \approx \arg\max_{\xi} \left(1+\exp(-y(\xi\sqrt{v}+m))\right)^{-1} \cdot \exp\left(-\xi^2\right) \tag{6}$$

$$\mathbf{Var}_{t+1}[\xi] = \sigma^2_{\hat{\xi}_{t+1}} \approx \left(v + v^2 \frac{\exp(v\hat{\xi}_{t+1}+\mu)}{(1+\exp(v\hat{\xi}_{t+1}+\mu))^2}\right)^{-1} \tag{7}$$

Once we have the above, we then use the joint Gaussian assumption of $\mathbf{w}$ and ξ and have

$$\mathbf{E}\begin{bmatrix}\mathbf{w}\\ \xi\end{bmatrix} = \begin{bmatrix}\mu_t\\ 0\end{bmatrix} \text{ and } \mathbf{Cov}\begin{bmatrix}\mathbf{w}\\ \xi\end{bmatrix} = \begin{bmatrix}\Sigma_t & \Sigma_t\mathbf{x}\\ \mathbf{x}^\top\Sigma_t & 1\end{bmatrix},$$

where Σ_t is assumed to be diagonal. Then, from the iterated expectation formula, we obtain

$$\mu_{t+1} = \mu_t + \mathbf{x}^\top\Sigma_t\hat{\xi}_{t+1} \tag{8}$$

$$\Sigma_{t+1} = \Sigma_t + \left(\sigma^2_{\hat{\xi}_{t+1}} - \frac{1}{\sigma^2}\right)\cdot\Sigma_t\mathbf{x}\mathbf{x}^\top\Sigma_t \tag{9}$$

For diagonal covariance matrix Σ_t, the above updates can be done efficiently in linear time.

Appendix B. Approximate Inference in Probit Regression

In probit regression, the posterior distribution of the weight vector $\mathbf{w}$ is proportional to the product of the probit likelihood and the Gaussian prior distribution, i.e.

$$p(\mathbf{w}) \propto \Phi(y\mathbf{x}^\top\mathbf{w})N(\mathbf{w};\mu_t,\Sigma_t),$$

where $\mathbf{x}$ denote a training sample with label $y \in \{\pm1\}$. In our experiments, a click signal $c \in \{0,1\}$ has to be converted to a binary label through $2c-1$. We take a variational approach to approximate the posterior distribution $p(\mathbf{w})$ by a Gaussian distribution. The approximate Bayesian inference technique is known as Assumed Density Filter (ADF) or Expectation Propagation (EP) (Lawrence et al., 2002; Minka, 2001). Specifically, let $N(\mathbf{w};\mu_{t+1},\Sigma_{t+1})$ be the target Gaussian, whose parameters $\{\mu_{t+1},\Sigma_{t+1}\}$ are determined by the minimizer of the following Kullback-Leibler divergence:

$$\arg\min_{\mu,\Sigma} \mathbf{KL}\left(\Phi(y\mathbf{x}^\top\mathbf{w})N(\mathbf{w};\mu_t,\Sigma_t)\|N(\mathbf{w};\mu,\Sigma)\right).$$

This optimization problem can be solved analytically by moment matching up to the second order, yielding:

$$\mu_{t+1} = \mu_t + \alpha\,(\Sigma_t\mathbf{x}) \tag{10}$$

$$\Sigma_{t+1} = \Sigma_t - \delta\,(\Sigma_t\mathbf{x})(\Sigma_t\mathbf{x})^\top \tag{11}$$

where

$$\alpha = \frac{y}{\sqrt{\mathbf{x}^\top\Sigma_t\,\mathbf{x}+1}}\frac{N(z)}{\Phi(z)},$$

$$\delta = \frac{1}{\sqrt{\mathbf{x}^\top\Sigma_t\,\mathbf{x}+1}}\frac{N(z)}{\Phi(z)}\left(\frac{N(z)}{\Phi(z)}+z\right),$$

and $z = \frac{y\mathbf{x}^\top \mu_t}{\sqrt{\mathbf{x}^\top \Sigma_t \mathbf{x}+1}}$. Interested readers are referred to Minka (2001) for a detailed derivation. If the dimension of $\mathbf{x}$ is high, the covariance matrix Σ_t can be restricted to be *diagonal*. This restriction also corresponds to the idea of mean-field approximation; see Graepel et al. (2010) for a successful application of this method in a search engine setting. Then, the parameter update above takes only $\mathcal{O}(d)$ time on average, where d is the average number of non-zero features.

Appendix C. Numerical Approximation for the Logistic Model

Since our posterior of the GLM parameter is a Gaussian, $N(\mu_a, \Sigma_a)$, the linear combination of features, $\mathbf{x} \cdot \mathbf{w}_a$, is also a Gaussian with mean $m = \mathbf{x} \cdot \mu_a$ and variance $v = \mathbf{x}^\top \Sigma_a \mathbf{x}$. Denote by $\hat{c}$ the estimate $(1 + \exp(\mathbf{x} \cdot \mathbf{w}_a))^{-1}$.

Normal Approximation. Since the logistic link function is monotonic, finding an upper confidence bound for the CTR estimate can be reduced to finding an upper confidence bound for $\mathbf{x}^\top \mathbf{w}_a$. Hence, we may work with the normally distributed quantity $\mathbf{x}^\top \mathbf{w}_a$. Its UCB is given by

$$\text{U0:} \qquad \left(1 + \exp(-\mu - \alpha\sqrt{v})\right).$$

The posterior mean is approximated by the posterior median:

$$\text{M0:} \qquad \frac{1}{1 + \exp(-\mu)}.$$

Logistic Distribution Approximation Equation (2) is in fact a convolution between a Gaussian density function and the logistic likelihood function. Very effective approximations have been proposed, an example being the one suggested by MacKay (1992):

$$\text{M1:} \qquad \int (1 + \exp(-x))^{-1} N(x|m, v)dx \approx \left(1 + \exp\left(-\frac{m}{\sqrt{1 + \pi v/8}}\right)\right)^{-1}. \qquad (12)$$

The formula above can be derived by approximating standard logistic distribution by a Gaussian distribution with zero mean and variance $8/\pi$. More precisely,

$$\frac{d}{dx}\left(\frac{1}{1 + \exp(-x)}\right) \approx N(x \mid 0, 8/\pi).$$

Taking advantage of this same powerful approximation, we may estimate the second moment of the CTR estimate:

$$\mathbf{E}[\hat{c}^2] = \int (1 + \exp(-x))^{-2} N(x|m, v)dx \approx \Phi\left(\frac{\pi\mu/\sqrt{8} - 1}{\sqrt{\pi - 1 + \pi^2 v/8}}\right).$$

The posterior variance, $\mathbf{E}[\hat{c}^2] - \mathbf{E}[\hat{c}]^2$, can then be estimated immediately:

$$\text{U1:} \qquad \Phi\left(\frac{\pi\mu/\sqrt{8} - 1}{\sqrt{\pi - 1 + \pi^2 v/8}}\right) - \left(1 + \exp\left(-\frac{m}{\sqrt{1 + \pi v/8}}\right)\right)^{-2}.$$

Log-normal Approximation. The logistic model assumption is that $\log \frac{\hat{c}}{1-\hat{c}} = \mathbf{x}^\top \mathbf{w}_a$ is normally distributed. In other words, $\frac{\hat{c}}{1-\hat{c}}$ follows the log-normal distribution, $\ln N(m, v)$. Hence,

$$\mathbf{E}\left[\frac{\hat{c}}{1-\hat{c}}\right] = \exp\left(\mu + v/2\right), \quad \mathbf{Var}\left[\frac{\hat{c}}{1-\hat{c}}\right] = \left(\mathbf{E}\left[\frac{\hat{c}}{1-\hat{c}}\right]\right)^2 \left(\exp(v) - 1\right).$$

If $\hat{c} \ll 1$ (which is the case in most Web applications), we may approximate $\frac{\hat{c}}{1-\hat{c}}$ by $\hat{c}$. The variance can be estimated by

$$\text{U2:} \qquad \mathbf{Var}[\hat{c}] \approx \mathbf{Var}\left[\frac{\hat{c}}{1-\hat{c}}\right] \approx \left(\mathbf{E}\left[\hat{c}\right]\right)^2 \left(\exp(v) - 1\right).$$

Similarly, the posterior mean is approximated by

$$\text{M3:} \qquad \mathbf{E}[\hat{c}] \approx \mathbf{E}\left[\frac{\hat{c}}{1-\hat{c}}\right] = \exp\left(\mu + v/2\right).$$

Approximation M3 may be problematic since $\exp\left(\mu + v/2\right)$ may be larger than 1. It may be useful to correct the mean estimate by applying the logistic function, yielding a slightly different formula:

$$\text{M2:} \qquad \mathbf{E}[\hat{c}] \approx \left(1 + \exp\left(-\mu - v/2\right)\right)^{-1}.$$

Conservative Approximation. Lipschitz continuity of the logistic link function motivates another UCB formula proposed by Filippi et al. (2011):

$$\text{U3:} \qquad \frac{1}{1 + \exp(-\mu)} + \alpha\sigma.$$

References

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, Nitin Motgi, Seung-Taek Park, Raghu Ramakrishnan, Scott Roy, and Joe Zachariah. Online models for content optimization. In *Advances in Neural Information Processing Systems 21*, pages 17–24, 2009.

Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Xuanhui Wang. Click shaping to optimize multiple objectives. In *Proceedings of the Seventeenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-11)*, 2011.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002a.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002b.

Donald A. Berry and Bert Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Monographs on Statistics and Applied Probability. Chapman and Hall, 1985. ISBN 0412248107.

Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, pages 19–26, 2011.

Wei Chu, Seung-Taek Park, Todd Beaupre, Nitin Motgi, Amit Phadke, Seinjuti Chakraborty, and Joe Zachariah. A case study of behavior-driven conjoint analysis on Yahoo!: Front Page Today Module. In *Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1097–1104, 2009.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, pages 208–214, 2011.

Louis Dorard, Dorota Glowacka, and John Shawe-Taylor. Gaussian processes modelling of dependencies in multi-armed bandit problems. In *Proceedings of the Tenth International Symposium on Operational Research (SOR-09)*, pages 721–728, 2009.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning (ICML-11)*, pages 1097–1104, 2011. CoRR abs/1103.4601.

Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23 (NIPS-10)*, pages 586–594, 2011.

Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale Bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning (ICML-10)*, pages 13–20, 2010.

Steffen Grünewälder, Jean-Yves Audibert, Manfred Opper, and John Shawe-Taylor. Regret bounds for Gaussian process bandit problems. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, pages 273–280, 2010.

Michael J. Kearns, Yishay Mansour, and Andrew Y. Ng. Approximate planning in large POMDPs via reusable trajectories. In *Advances in Neural Information Processing Systems 12*, pages 1001–1007, 2000.

John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Advances in Neural Information Processing Systems 20*, pages 1096–1103, 2008.

John Langford, Alexander L. Strehl, and Jennifer Wortman. Exploration scavenging. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08)*, pages 528–535, 2008.

Neil D. Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 609–616, 2002.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the Nineteenth International Conference on World Wide Web (WWW-10)*, pages 661–670, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Fourth International Conference on Web Search and Web Data Mining (WSDM-11)*, pages 297–306, 2011.

David J. C. MacKay. The evidence framework applied to classification networks. *Neural Computation*, 4(5):720–736, 1992.

Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman and Hall, 2nd edition, 1989. ISBN 0412317605.

T. P. Minka. *A family of algorithms for approximate Bayesian inference*. Ph.D. thesis, Massachusetts Institute of Technology, January 2001.

Taesup Moon, Lihong Li, Wei Chu, Ciya Liao, Zhaohui Zheng, and Yi Chang. Online learning for recency search ranking using real-time user feedback. In *Proceedings of the Nineteenth International Conference on Knowledge Management (CIKM-10)*, pages 1501–1504, 2010.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML-00)*, pages 759–766, 2000.

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML-06)*, pages 881–888, 2006.

Alexander L. Strehl, John Langford, Lihong Li, and Sham M. Kakade. Learning from logged implicit exploration data. In *Advances in Neural Information Processing Systems 23*, pages 2217–2225, 2011.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, March 1998. ISBN 0-262-19398-1.

Yahoo! Yahoo! front page today module user click log dataset, version 1.0, 2011. http://webscope.sandbox.yahoo.com.