# Exploitation and Exploration in a Performance based Contextual Advertising System

Wei Li, Xuerui Wang, Ruofei Zhang, Ying Cui, Jianchang Mao
Yahoo! Labs, Sunnyvale, CA, 94089
{lwei,xuerui,rzhang,ycui,jmao}@yahoo-inc.com

Rong Jin
Michigan State University
East Lansing, MI 48824
rongjin@cse.msu.edu

## ABSTRACT

The dynamic marketplace in online advertising calls for ranking systems that are optimized to consistently promote and capitalize better performing ads. The streaming nature of online data inevitably makes an advertising system choose between maximizing its expected revenue according to its current knowledge in short term (exploitation) and trying to learn more about the unknown to improve its knowledge (exploration), since the latter might increase its revenue in the future. The exploitation and exploration (EE) tradeoff has been extensively studied in the reinforcement learning community, however, not been paid much attention in online advertising until recently. In this paper, we develop two novel EE strategies for online advertising. Specifically, our methods can adaptively balance the two aspects of EE by automatically learning the optimal tradeoff and incorporating confidence metrics of historical performance. Within a deliberately designed offline simulation framework we apply our algorithms to an industry leading performance based contextual advertising system and conduct extensive evaluations with real online event log data. The experimental results and detailed analysis reveal several important findings of the EE behaviors in online advertising and demonstrate that our algorithms perform superiorly in terms of ad reach and click-through-rate (CTR).

## Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based services*; I.2.6 [**Artificial Intelligence**]: Learning

## General Terms

Algorithms, Design, Experimentation, Performance

## Keywords

Exploitation and exploration, online advertising, click feedback, click-through-rate


*KDD'10,* July 25–28, 2010, Washington, DC, USA.


## 1. INTRODUCTION

In the advertising industry, an ongoing shift in spending from traditional media to the Internet has been widely observed during the past decade. Since 2006, the year-over-year growth of Internet advertising spending has eclipsed other conventional media such as newspaper. As marketers realize that online spending is more effective at influencing consumers and delivering more measurable results, nowadays online advertising has become a major business. Based on a recent study by Interactive Advertising Bureau, online advertising revenue in the first half of 2009 reached over 10.9 billion US dollars [4].

A significant part of this market consists of ads delivery distributed through three main advertising channels: display ads, sponsored search and contextual advertising. In this paper, we focus on contextual advertising that refers to the placement of ads within the content of a generic Web page for a user, but the ideas illustrated here are generally applicable to other distribution channels as well. In contextual advertising, usually there is a commercial intermediary, called ad-network, in charge of optimizing the ad selection with a three-fold goal of increasing revenue (shared between publishers and ad-network), improving return-on-investment of advertisers, and enhancing user experience.

One of the most prevalent pricing models for contextual advertising is that advertisers pay a certain amount for every click on an ad (cost-per-click or CPC). There are also other models: cost-per-impression where advertisers pay for the number of exposures of an ad, and cost-per-action where advertisers pay only if an ad leads to a sale or a similar transaction. In the CPC model, the revenue of an ad-network is crucially dependent on the click-through-rate (CTR) that can be viewed as the probability that an end-user would click on an ad when shown on a Web page.

From the perspective of ad-networks, it is always preferable to show ads with high expected revenue, which depends on both ad CTR and its bid in the CPC model. Unfortunately, ad CTRs cannot be known a priori, and thus many ad-networks' ranking systems use historical CTRs instead. This approach is usually referred to as performance based or click feedback model. However, this kind of systems are likely to suffer from the *Matthew effect* (the rich get richer and the poor get poorer) in the long run. The ads with large numbers of impressions and high CTRs are usually the top candidates for displaying. Therefore they will get even more impressions (and sometimes clicks) in the future. On the other hand, the ads with low impressions and potentially high CTRs might unluckily have few or no clicks, and thus

are considered inferior performers due to the bogus historical CTRs associated with the sparse data. Their chances for future exposure will be even slimmer regardless of their *true* CTRs. This is especially bad news for a fast changing marketplace where we constantly have new ads joining every hour. In an exploitation only system, these newcomers are not as competent as the established good performers, and it would be difficult for them to catch up. In the long term, most impressions will go to a small set of ads. In general, this trend is not desirable as it could cause serious problems:

1. Keeping showing the same ads repeatedly to the same users easily leads to user fatigue. That is widely considered as very unpleasant user experience and could result in a drop in clicks [7].

2. Either over-exposing ads of the same advertisers in a short period or under-exposing in a long period can severely hurt economic efficiency for both advertisers and ad-network operators. Some advertisers will quickly burn out their budgets while others cannot spend their planned ad dollars due to the lack of exposure.

In order to address these issues, an ideal way is to exploit those established good ads for short-term capitalization, and at the same time explore the unknown (or less known) ad space to discover potentially better ads for the long term. This is a typical tradeoff between exploitation and exploration (EE), frequently studied with the multi-armed bandit problem in reinforcement learning [12, 13]. More specifically, our dilemma can be viewed as a variation of the multi-armed bandit problem, defined as iteratively choosing one out of multiple actions and receiving a reward drawn from a distribution associated with this action. The goal is to find the optimal action sequence that maximizes the accumulated rewards. In the scenario of contextual advertising, we select a small number of ads for each page view and receive a user feedback (click or non-click) as reward.

One classical solution to the multi-armed bandit problem is the $\epsilon$-greedy strategy [12]. With probability 1-$\epsilon$, this algorithm chooses the best action based on current knowledge; and with probability $\epsilon$, it chooses any other action uniformly. The parameter $\epsilon$ essentially controls the tradeoff between exploitation and exploration. One disadvantage of this algorithm is that the optimal $\epsilon$ value is difficult to decide in advance. Moreover, we may need to dynamically adjust the EE tradeoff at different stages of the experiment. This concern can be addressed by a slightly different strategy called $\epsilon$-decreasing [12], where the value of $\epsilon$ decreases as the experiment processes. In other words, this algorithm focuses more on exploration at the beginning and gradually shifts to exploitation at the end.

Compared to the standard multi-armed bandit problem with a fixed set of possible actions, we face a fast changing marketplace in contextual advertising. Old ads may expire and new ads may emerge everyday. Therefore it may not be desirable to monotonically decrease the effort on exploration as the $\epsilon$-decreasing strategy does. Instead, we want to adaptively balance the tradeoff between exploitation and exploration in response to the data change. In this paper we propose two novel approaches that achieve this goal.

In the first method, we extend the $\epsilon$-greedy strategy by updating the value of $\epsilon$ dynamically (not necessarily decreasing). At each iteration, we run a sampling procedure to select a new $\epsilon$ from a finite set of candidates. The probabilities associated with the candidates are initialized uniformly and updated with the Exponentiated Gradient (EG) algorithm [9]. As shown in Section 4, this updating rule increases the probability of being chosen for a candidate if it leads to a user click.

In the second method, we improve a different aspect of the $\epsilon$-greedy algorithm, namely the random exploration strategy. As we mention before, the marketplace in contextual advertising can change rapidly, and different ads have very different histories. Some old ads have already been shown many times while new ones have few or no impressions. As a result, it is not desirable to explore ads in a random way because it will waste opportunities on the established bad performers. Instead, we only want to explore the ads that might lead to higher revenue in the long run, i.e., ads with bogus poor performance due to sparse data or even no history. To address this, we introduce an impression based confidence metric and use it to decide which ads need to be explored. This confidence metric also serves as a dynamic switch between exploitation and exploration. When the confidence increases to a certain level, some exploration opportunities will be automatically shifted to exploitation.

In this paper, we set up an offline simulation framework from real event logs in an industry leading performance based contextual advertising system, and evaluate the short-term and long-term impacts of EE on the system performance. The experiment results demonstrate that our proposed methods can greatly improve the ad reach and empirical CTR. Compared to an exploitation only baseline, both EE algorithms increase the ad reach by more than 25%. While the dynamic $\epsilon$-greedy algorithm with Exponentiated Gradient converges to a slightly lower CTR than its best fix-value counterpart, its fast convergence rate makes it more competent in a real system. Furthermore, the confidence-based algorithm improves CTR almost by half over the baseline and about 20% over other EE algorithms.

## 2. RELATED WORK

The exploitation and exploration tradeoff was first formally studied in reinforcement learning in 1980's, and later flourished in other fields of machine learning [12, 13]. Very frequently used in reinforcement learning to study the EE tradeoff, the multi-armed bandit problem (also sometimes called $K$-armed bandit) was originally described by Robbins [11]. Agarwal et al. recently applied this technique to maximize total clicks on a Web content module [1]. Langford et al. extended this problem to the case where there is some side information that helps to make a better decision [10]. They called the new setting "contextual bandit problem", introduced the epoch-greedy algorithm and derived its regret bound.

The confidence-based approach proposed in this paper is partially inspired by the UCB algorithm [2] that ranks the actions based on the combination of estimated rewards and their confidence intervals. However, given the large number of ads to be selected, the number of impressions received by each ad is usually small, leading to an inaccurate estimation of confidence intervals and consequently the limited performance of the standard UCB algorithm. Thus, unlike the UCB algorithm that is a deterministic strategy in choosing actions, the proposed confidence-based EE algorithm is a hybrid approach that combines UCB with the $\epsilon$-greedy algorithm. By introducing randomness into the UCB algorithm,

we alleviate the trouble in estimating confidence intervals and achieve more reliable performance.

Another closely related area to the discussion in this paper is the click feedback system in online advertising. Click feedback has a relatively long history in the nascent online advertising industry. As early as in 2002, Google Inc. incorporated click feedback into its sponsored search system—*AdWords* [3]. In more recent years, similar ideas are adopted into other distribution channels of online advertising as well. For example, Chakrabarti et al. described a contextually advertising system that combines ad relevance with historical impression and click information through a logistic regression model [6]. The performance-based contextual advertising system in this paper roughly shares the same idea.

# 3. CLICK FEEDBACK MODEL

In this section, we briefly describe a click feedback model (CFB), an approach that predicts future ad CTRs based on their past performance.

The basic idea is to collect performance history of page ad pairs to estimate CTRs in contextual advertising. There are two types of events that we are interested in. The first one is impression, which refers to the showing of an ad on a Web page. Its frequency generally obeys the power-law distribution, where a small set of popular pages and ads account for a large fraction of the traffic while the overwhelming counterparts have very rare frequency individually. Another event is click, which corresponds to the action that a user clicks on an ad. Our estimation of CTR is simply the ratio of the number of clicks to the number of impressions in a certain time period.

In the CFB model, we keep historical counts of impressions and clicks for each page ad pair at multiple levels of granularity. For example, on the page side, the top level is a publisher, followed by domain and page URL. On the ad side, each advertiser account usually consists of multiple ad groups and each ad group is a set of closely related ads. The reason for introducing such hierarchies is because the impressions and clicks at the (page URL, ad) level are very rare for the majority of page ad pairs. Therefore considering aggregations at higher levels reduces the variance and provides more reliable estimation of CTR.

For each page ad aggregation, the historical counts are updated everyday in order to estimate CTRs from the most recent events. Note that if the total number of impressions for a particular aggregation is less than a threshold, it will be considered too sparse to provide reliable estimation. The corresponding CTR is set to a default value instead, such as the overall CTR for all page ad pairs. Finally, we calculate a CFB score by combining the historical CTRs from all aggregations.

As most feedback-based approaches, the CTR estimation in the CFB model is usually more reliable for page ad pairs with large numbers of impressions and less so for the ones with little or no history. For example, consider the following three ads to be shown on a given page. The first one has 100,000 impressions and 100 clicks. The second one also has 100,000 impressions but only 1 click. The third one is relatively new with 10 impressions and no clicks. There is almost no doubt that the first ad should be ranked higher than the second. The question is how to deal with the third one. If we rank the ads strictly based on historical CTRs, the third ad will have the lowest rank. That is obviously not a wise solution considering that the second one is a known bad performer. With an estimation method that compensates CTR based on the actual numbers of impressions and clicks, the third ad may be ranked higher than the second one but not the first one. This strategy may still be cursed by the *Matthew effect* with the drawbacks we discussed in Section 1, i.e., advertiser budgets are not spent efficiently, ad-network loses high capitalization of missing ads, publishers get less yield because not all the ads in the marketplace participate in the auction, and user experience is hurt as well due to user fatigue of seeing the same ads again and again.

# 4. EXPLOITATION AND EXPLORATION

From the discussion in Section 3, we can see that the CTR estimation in the CFB model may not accurately reflect the true performance of a page ad pair, especially when they have a small number of impressions. To resolve this problem, we propose to use a stochastic, rather than deterministic, approach in ad selection. More specifically, given a page and a list of ads ranked by CFB scores, we believe that the ads ranked near the top generally have better performance than ads near the bottom. However, considering the inaccuracy in the CFB model as we elaborate above, we want to give some opportunities to ads with low ranks to be selected for exposing. How to determine an optimal strategy is essentially an exploitation and exploration (EE) problem. In the rest of this section, we present two EE approaches that follow the $\epsilon$-greedy strategy but dynamically control the tradeoff between exploitation and exploration. The first method uses Exponentiated Gradient to learn the optimal value of parameter $\epsilon$, and the second one takes impression history into consideration and only explores ads with few or no impressions.

## 4.1 The $\epsilon$-greedy Algorithm with EG Update

The $\epsilon$-greedy algorithm uses a simple random strategy to overcome the problems identified in the CFB ranking model. In Algorithm 1, we describe a variation of the standard $\epsilon$-greedy method that is more suitable for our application. First we divide the ranking list into two parts. The first part consists of $r$ ads with the highest CFB scores and will be reserved for exploitation. This step is introduced mainly for the business interests as it protects the revenue from dramatically dropping in the short term. For the second part, we further divide it into two lists with high-rank and low-rank ads. In general, we deem the high-rank ads significantly better than the low-rank ads for a given Web page. The parameter $q$ specifies the size of the low-rank list, which consists of the ads that can participate in exploration. These two lists are then merged together with a sampling procedure that follows [8], where the probability of choosing ads from the low-rank list is $\epsilon$.

One problem with the $\epsilon$-greedy algorithm is how to decide the optimal $\epsilon$, a critical tradeoff parameter between exploitation and exploration. Instead of specifying $\epsilon$ manually, we iteratively update this quantity by the Exponentiated Gradient (EG) algorithm. First we assume that we have a finite number of candidate values for $\epsilon$, denoted by $(\epsilon_1, \cdots, \epsilon_T)$. Our goal is to learn the optimal $\epsilon$ from this set. To this end, we introduce $\mathbf{p} = (p_1, \cdots, p_T)$, where $p_i$ stands for the probability of using $\epsilon_i$ in the $\epsilon$-greedy algorithm. These probabilities are initialized to be $1/T$ at the beginning and

then iteratively updated through trials. More specifically, we use a set of weights $\mathbf{w} = (w_1, \cdots, w_T)$ to keep track of the performance of each $\epsilon_i$ and update them using the EG algorithm. The basic idea is to increase $w_i$ if we receive a click from using $\epsilon_i$. Finally, we calculate $\mathbf{p}$ by normalizing $\mathbf{w}$ with smoothing. The detailed algorithm is shown in Algorithm 2. Here $I[z]$ is the indictor function. Parameters $\beta$ and $\tau$ are smoothing factors in weight updating. $\kappa$ is a regularization factor to handle singular $w_i$.

The regret bound of the Exponentiated Gradient $\epsilon$-greedy algorithm is revealed by the following theorem.

THEOREM 1. *With probability $1-\delta$, by running the Exponentiated Gradient $\epsilon$-greedy algorithm in Algorithm 2 for choosing $\epsilon$ against $N$ Web pages using parameters*

$$\beta = \sqrt{\frac{\ln(T/\delta)}{TN}},\ \kappa = \frac{4T\beta}{3+\beta},\ \tau = \frac{\kappa}{2T},$$

*we have the following inequality holds*

$$\sum_{i=1}^{N} c_i - \max_{1\le j\le T} C(\epsilon_j) \le \frac{11}{2}\sqrt{TN\ln(T/\delta)} + \frac{1}{2}\ln T,$$

*where $C(\epsilon_j)$ stands for the number of clicks received by using $\epsilon = \epsilon_j$ and $\sum_{i=1}^{N} c_i$ computes the number of clicks received by running Algorithm 2 to choose $\epsilon$.*

*Proof.* The proof follows directly the proof for Theorem 6.10 in [5].

**Algorithm 1** The $\epsilon$-greedy algorithm for EE in advertising

**Input**:
  $\{a_1, a_2, \cdots, a_n\}$: an ad list ranked by CFB scores
  $r$: number of reserved ads
  $q$: number of ads in the low-rank list
  $\epsilon$: probability to select ads in the low-rank list
**Output**:
  $F$: A list of ads re-ranked by the EE algorithm
$F \Leftarrow \{a_1, \cdots, a_r\}$ {the final ranking list $F$}
$H \Leftarrow \{a_{r+1}, \cdots, a_{n-q}\}$ {the high-rank ad list $H$}
$L \Leftarrow \{a_{n-q+1}, \cdots, a_n\}$ {the low-rank ad list $L$}
**repeat**
  Sample $z$ from Bernoulli($\epsilon$)
  **if** $z = 0$ **then**
    Randomly choose an ad $a_i$ from $L$
    $L \Leftarrow L \backslash \{a_i\}$, $F \Leftarrow F \cup \{a_i\}$
  **else**
    Randomly choose an ad $a_i$ from $H$
    $H \Leftarrow H \backslash \{a_i\}$, $F \Leftarrow F \cup \{a_i\}$
  **end if**
**until** $H$ or $L$ is empty
$F \Leftarrow F \cup H \cup L$

## 4.2 Confidence-based Exploration

In addition to the $\epsilon$-greedy algorithm that treats all ads equally in exploration, we can further introduce a confidence metric on CFB scores to explore those ads with low confidence only. There are many ways to measure the confidence of CFB scores, such as the variance of posterior CTR distribution using Beta-Binomial or Gamma-Poisson models. In this paper, we simply use an impression-based metric for illustration. But the algorithm itself is agnostic to the selection of confidence measurements.

**Algorithm 2** The $\epsilon$-greedy algorithm with the Exponentiated Gradient update

**Input** (in addition to Algorithm 1):
  $\{\epsilon_1, \cdots, \epsilon_T\}$: candidate values for $\epsilon$
  $\beta$, $\tau$ and $\kappa$: parameters for EG
  $N$: number of iterations
$p_k \Leftarrow 1/T$ and $w_k \Leftarrow 1$, $k = 1, \cdots, T$
**for** $i = 1$ to $N$ **do**
  Sample $d$ from Discrete($p_1, \cdots, p_T$)
  Run Algorithm 1 with $\epsilon_d$
  Receive a click feedback $c_i$ from the user
  $w_k \Leftarrow w_k \exp\left(\frac{\tau[c_i I(k=d)+\beta]}{p_k}\right)$, $k = 1, \cdots, T$
  $p_k \Leftarrow (1-\kappa)\frac{w_k}{\sum_{j=1}^{T} w_j} + \frac{\kappa}{T}$, $k = 1, \cdots, T$
**end for**

Given a page ad pair $< p, a >_j$ and their impressions $x_j$, we define the confidence for their CFB score as

$$C(x_j) = \tanh(\frac{x_j}{b}),$$

where *tanh* is the tangent function and $b$ is the shape parameter. With this function, the confidence of CTR estimation is positively correlated with the number of impressions. In a special case where a page ad pair does not meet the impression threshold of the CFB model, the confidence is set to be 0.

The EE algorithm with the confidence metric is outlined in Algorithm 3. In this algorithm, we have multiple knobs to control how to explore the ads. Parameters $r$, $q$ and $\epsilon$ are the same as defined in the $\epsilon$-greedy algorithm. In addition, we have an impression upper-bound $w$, which allows us to stop exploration for ads with too many impressions. Given an ad list ranked by CFB scores, we first divide it into two parts as in the $\epsilon$-greedy algorithm. The top ads are always kept without changing their positions (step 3). For the rest of the ads, we randomly sample some of them into a promotional queue based on a probability distribution defined with the confidence metric (steps 4 to 12). The lower confidence an ad has, the more likely it will be selected. Note that if the number of impressions of an ad reaches the upper-bound $w$, it will not be considered for promotion. Finally we merge the promotional queue with the remaining ads (steps 13 to 21), giving additional chances to increase the ranks of ads in the promotional queue.

Compared to the $\epsilon$-greedy algorithm, we no longer decide whether to promote an ad based on its position in the original ranking list. Instead, we give higher priority to ads with lower confidence in their CFB scores as these scores are more likely to be inaccurate.

# 5. EXPERIMENTS

In this section, we describe our experiment design to evaluate Algorithms 2 and 3 and present the results. We are interested in evaluating the long-term effects of EE algorithms, but it takes time to explore the unknown space and discover new ads with good performance. During such a long time period, CTR may change significantly even if the system remains the same. Therefore we cannot simply evaluate an EE algorithm by comparing the CTRs of the system before and after applying it. Due to the restrictions of setting up an isolated environment in the online system, we

**Algorithm 3** The Confidence-based EE algorithm for advertising

**Input**:
  $A = \{a_1, a_2, \cdots, a_n\}$: an ad list ranked by CFB scores
  $\{x_1, x_2, \cdots, x_n\}$: the impressions of the above ads
  $r$: number of reserved ads
  $q$: number of ads in the promotional queue
  $\epsilon$: probability to select ads in the promotional queue
  $w$: impression upper-bound for exploration
**Output**:
  $F$: A list of ads re-ranked by the EE algorithm
$F \Leftarrow \{a_1, \cdots, a_r\}$ {the final ranking list $F$}
$P \Leftarrow \{\}$ {the promotional queue $P$}
**for** $i = r + 1$ to $n$ **do**
  **if** $x_i < w$ **then**
    $p_i \Leftarrow 1 - \tanh(x_i/b)$
  **else**
    $p_i \Leftarrow 0$
  **end if**
**end for**
Sample $q$ ads from $\{a_{r+1}, \cdots, a_n\}$ without replacement based on (unnormalized) probabilities $\{p_{r+1}, \cdots, p_n\}$ and append the $q$ ads to $P$
**repeat**
  Sample $z$ from Bernoulli($\epsilon$)
  **if** $z = 0$ **then**
    $a = \text{POP}(A)$, $F \Leftarrow F \cup \{a\}$ if $a \notin F$
  **else**
    $a = \text{POP}(P)$, $F \Leftarrow F \cup \{a\}$ if $a \notin F$
  **end if**
**until** $P$ or $A$ is empty
$F \Leftarrow F \cup P \cup A$

instead introduce an offline simulation framework for more accurate evaluation. This system is able to mimic emitting the online events, running the CFB model, as well as setting up controlled experiment buckets to perform apple-to-apple comparisons between a pure exploitation baseline and various EE algorithms.

## 5.1 Simulation Framework Design

For the offline simulation system, we generate a data set from the event logs in an industry leading contextual advertising serving system within a three-month timeframe. First we randomly collect 1,000 Web pages. The only requirement is that their impressions need to be in the middle range. We do not want to use pages that are either too popular or too unpopular. Then we sample 10,000 ads that have been shown on any of these pages. For each page ad pair, we calculate its empirical CTR as the ratio of clicks to impressions and store it in a table as the ground truth. Theoretically there are 10,000,000 (page, ad) pairs in our data set, but most of them have no impressions. Here we only keep the pairs with impressions greater than 100 to be statistically meaningful and obtain a ground truth CTR table with 64,798 entries. This table is used to mimic user actions and online system behavior. On average, each page has about 65 ads with ground truth.

In addition to the ground truth table, we also need to set up the offline CFB model. There are several simplifications we adopt in the simulation. First, we only consider two levels of aggregation: (page URL, ad) and (publisher, ad group), which are common in click feedback models. Secondly, the model is initialized from a snapshot of the online CFB tables, restricting to the pages and ads in our data set. In the (page URL, ad) table, each entry corresponds to a page ad pair and consists of three historical statistics: impression, click and CTR. We use an impression threshold of 100 to decide whether the CTR is calculated by definition or set to a default value. The (publisher, ad group) table is created in a similar way except that each entry corresponds to a publisher and ad group pair.

Before we run the simulation, the (page URL, ad) table contains only 17,814 entries with sufficient impressions. Its coverage is much lower than the ground truth. In other words, the initial CFB model is unclear about the performance of many page ad pairs. However, the (publisher, ad group) table is not so sparse. Out of 35 publishers and 9,184 ad groups, we have 51,847 pairs with enough impressions in the CFB table.

After initialization, the CFB model is updated in the following way. For each page ad pair, we first calculate its CFB score as a linear combination of CTRs from the two aggregations. Based on this score, we generate a ranking list and apply an EE algorithm to it. The top ads from the re-ranked list are then returned for displaying. The numbers of impressions for these (page URL, ad) pairs and the corresponding (publisher, ad group) pairs are increased by one. As for clicks, we experiment with two different methods to simulate user behavior, assuming that each Web page impression can lead to at most one click for each user interaction. The first approach is called random feedback. In this method, when we display a list of ads on a page, the user views them sequentially from top to bottom. For each ad, we simulate the user feedback as a Bernoulli trial. The outcome is either click or non-click, where the probability of click is the same as the ground truth CTR. Repeat this process until we obtain a click or exhaust the ad list. The second approach is called deterministic feedback. Instead of running a random process to decide whether an ad is clicked or not, this method calculates the probability of click for each ad in the display list and awards them with partial clicks based on their ground truth CTRs. Given a page and a list of $m$ ads, the probability of click for the $i$th ad is calculated as

$$P(click_i|page, ad_1, ad_2, \cdots, ad_m) = CTR(page, ad_i) \times \prod_{j=1}^{i-1}(1 - CTR(page, ad_j))$$

where $CTR(page, ad)$ is the ground truth CTR for the page ad pair. While the random feedback method is more realistic to simulate user actions, the deterministic approach allows us to analyze the theoretical upper-bounds for different EE algorithms when true feedbacks are available.

Once receiving the user feedback, we update the numbers of clicks and calculate new CTRs for the affected page ad pairs. As we mention before, a default CTR is used for those whose impressions are below the threshold. Note that when this default value is greater than 0, we are doing implicit exploration to some extent. In this case, ads with insufficient impressions may have better chances to be selected than ads with poor performance on many impressions. Moreover, when we use 1 as the default value, it can be viewed as an EE algorithm where exploration always has higher pri-

ority than exploitation until all the ads in the space receive enough impressions. The simulation procedure for running EE algorithms is summarized in Algorithm 4.

**Algorithm 4** Simulation procedure

```
for i = 1 to N do
  for all pages do
    Calculate a CFB score for each ad with ground truth available
    Retrieve the top n ads ranked by CFB scores
    Apply the EE algorithm to re-rank ads
    Return the top m ads for displaying
    Generate a user feedback
    Update the CFB impression, click and CTR statistics
  end for
end for
```

We use two metrics to evaluate the performance of an EE algorithm. The first one is the ad reach of performance history, measured by the number of entries in the CFB (page URL, ad) table with sufficient impressions. We believe that this metric is positively correlated with CTR estimation accuracy. With increased coverage in the CFB model, we are able to select from a larger pool of potentially good ads for displaying so as to improve user experience. The second metric is the average expected CTR. We could use the actual CTR, but it changes abruptly over iterations for the random feedback method. The expected CTR for a particular iteration is the ratio between the total number of expected clicks and the total number of impressions. The number of expected clicks for each page ad impression is the same as the probability of click defined above. Finally, we calculate the average expected CTR over every 100 iterations.

## 5.2 Experiment results

In all our experiments, we run the simulation until the expected CTR converges or the number of iterations reaches 200,000. The results are recorded at 100 iterations apart. Unless stated otherwise, the number of ads returned by the CFB model for each page ($n$) is 10 and the number of ads selected for display ($m$) is 3. The default way of generating user feedback is the random feedback method. Our experiments start with a single-level CFB model using (page URL, ad) aggregation only. Then we extend it to use two-level CFB tables for comparison. For all the figures in this section, we plot 30 to 40 points for a clear demonstration of the trend. The number of iterations shown for CFB coverage may be different from the average expected CTR since they have different convergence rates.

In the first experiment, we demonstrate the effects of three different values for the default CTR in the CFB model: 0, 1 and the mean of the initial CTRs. As mentioned before, using non-zero default CTRs can be considered as an implicit exploration method. With this experiment, our goal is to analyze their impacts on the CFB model and further decide the appropriate value to use when we compare different EE algorithms.

The evaluation results are presented in Figures 1 and 2. The horizontal axis is the number of iterations and the vertical axis is the performance metric. As we can see, both the CFB coverage and the average expected CTR increase as we use larger default CTRs. With the default CTR being 1, we

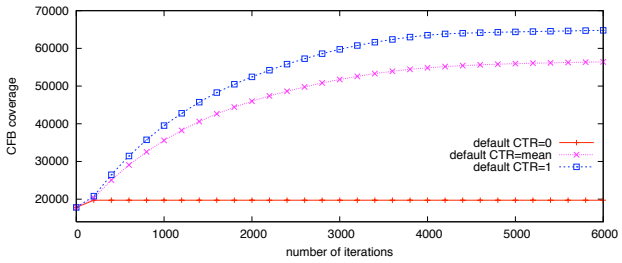


**Figure 1: CFB coverage for three default CTRs in the CFB model.**

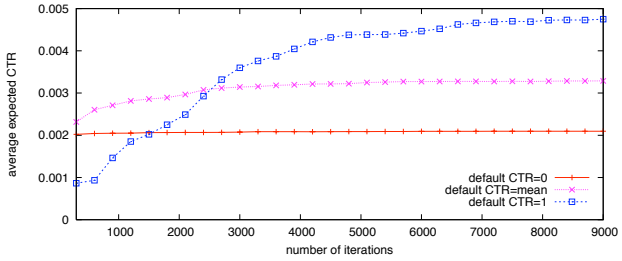


**Figure 2: Average expected CTR for three default CTRs in the CFB model.**

are able to explore all the ads in the ground truth table and reach the upper limit of the CFB coverage. However, we observe a significant drop in the average expected CTR at the early stage. This is expected since we totally ignore the historical performance and focus on exploration only. With more iterations, the CTR gradually converges to a higher level than the other two settings. These results confirm our theory that (even implicit) exploration is helpful to discover more ads with good performance and improve the system in the long run. On the other hand, we need to disable such implicit exploration when comparing the explicit EE algorithms to remove its impact on the performance.

In the second experiment, we evaluate the two EE algorithms described in Section 4. In addition to a pure exploitation baseline, we also compare against the $\epsilon$-greedy and $\epsilon$-decreasing algorithms. As explained above, the default CTR in the CFB model is set to be 0 in this experiment. We try to make the parameters in different algorithms as close as possible. The number of reserved ads ($r$) is set to be 1 and the number of ads in the low-rank list or the promotional queue ($q$) is set to be 4. For the $\epsilon$-greedy algorithm, we experiment with three parameter values: 0.1, 0.5 and 0.9. While it is highly unlikely to use an $\epsilon$ value greater than 0.5 in a real system, we still include the results of $\epsilon = 0.9$ for illustration purpose. For the confidence-based exploration algorithm, we use the same three $\epsilon$ values. Furthermore, its impression upper-bound for exploration ($w$) is 1,000 and the tangent function parameter $b$ is 300.0. For the $\epsilon$-decreasing and Exponentiated Gradient algorithms, they share the same set of candidates $\{\epsilon_i = 0.05 * i + 0.01, i = 1, \cdots, 10\}$. While the EG approach automatically learns the optimal parameter, $\epsilon$-decreasing starts with the largest value and reduces it by 0.05 every 2,000 iterations until $\epsilon$ reaches the smallest value. As we mentioned before, this algorithm is not suitable for a dynamic system; we do not intend to spend much effort tuning the decreasing function but only experiment with a

simple setup. The decreasing rate is determined empirically based on the CTR convergence rates of other EE algorithms.

The CFB coverages are plotted in Figure 3. The notations $x$-greedy and $x$-confidence refer to the $\epsilon$-greedy and confidence-based algorithms with $\epsilon = x$, respectively. Overall the EE algorithms have better performance than the baseline, but for the first 200 iterations, we achieve the fastest increase in the CFB coverage with pure exploitation. This is because for some pages, there may be less than 3 ads with sufficient impressions at the beginning. Therefore some other ads are selected to fill the remaining spots even when their impressions are not high enough. One thing to note is that we apply a stable sorting algorithm when generating the ranked list based on CFB scores. As a result, we always use the same set of fillers until their impressions eventually reach the threshold. Such repeated exposure of a small set of ads is more effective to increase the CFB coverage in the short term than the diffused exploration in EE algorithms. But in the long run, all EE algorithms catch up and improve the coverage by 25.46% at convergence.

We have several interesting findings regarding the behaviors of different EE algorithms. First, they are not able to reach the upper limit of 64,798 entries due to the restriction that they are only applied to the top $n$ ads returned by the CFB model, which is set to be 10 in this experiment. In the initial CFB table, each page has an average of 1.36 ads with non-zero CTRs. So it is fairly safe to assume that most ads outside the top 10 lists have zero CTRs. Because of this limitation and the stable sorting algorithm used in ranking, the top 10 lists returned by the initial CFB model essentially define our ad space for exploration with few exceptions. Therefore the true upper limit for the CFB coverage is approximately 27,814 here, which is the sum of the initial coverage 17,814 and the possible addition of 10,000 new ads (10 for each page). As we can see, the EE algorithms converge to a CFB coverage that is close to this limit.

Another observation is that, for the $\epsilon$-greedy algorithm, there is no obvious correlation between the convergence rate of CFB coverage and the $\epsilon$ value. The fastest one uses an $\epsilon$ of 0.5 and the slowest one uses 0.9. As we understand, a larger $\epsilon$ value corresponds to more opportunities for exploration and thus should lead to faster convergence of CFB coverage. The reason why we do not observe such a trend here is due to data sparseness. As we discuss above, the average number of ads with non-zero CTRs for each page is very small. As a result, the CTR difference between high-rank and low-rank ads is not significant. Most of these ads have zero CTRs in the CFB table. Since the $\epsilon$-greedy algorithm does not consider confidence of CTR estimation when creating the high-rank and low-rank lists, we expect them to have similar distributions over ads with insufficient impressions. In this case, both exploitation and exploration have similar chances to select ads that do not meet the CFB threshold, thus favoring exploration more does not necessarily help CFB coverage increase.

On the other hand, the convergence rate of the confidence-based exploration algorithm is positively correlated with the $\epsilon$ value. This is expected because of the strict distinction between exploitation and exploration in this method. By eliminating the randomness in exploitation and favoring ads with insufficient impressions in exploration, these two phases serve their own purposes more effectively. As a result, different *epsilon* values lead to very different behaviors. Overall, using an $\epsilon$ of 0.9 achieves the fastest convergence rate among all EE algorithms and 0.1 is the slowest.

As for the other two algorithms, $\epsilon$-decreasing is very similar to 0.5-greedy because it starts with 0.51 and converges before 1,000 iterations. The Exponentiated Gradient method effectively learns the optimal $\epsilon$ and its convergence rate is close to the best setting.

The comparison of average expected CTR is shown in Figure 4. Each point corresponds to the average result of the previous 100 iterations. For the $\epsilon$-greedy algorithm, the converged CTR increases as $\epsilon$ decreases (less exploration). This is again because the exploration does not take into account the confidence of CTR estimation. Even after convergence, the 0.9-greedy algorithm still gives 90% of the opportunities to ads with low ranks, which significantly decreases its average expected CTR. While the $\epsilon$-decreasing algorithm converges to a higher CTR than all $\epsilon$-greedy algorithms, its overall performance is not as good as 0.1-greedy. Its CTR drops a lot at the early stage because of more aggressive exploration but does not converge fast enough to compensate the loss. On the other hand, the dynamic $\epsilon$-greedy algorithm with Exponentiated Gradient has the fastest convergence rate and the highest CTR for the first 2,500 iterations. It is only after the CFB coverage stabilizes that 0.1-greedy begins to show its advantage. However, in a real system where the ad space dynamically changes at hourly basis, it is unlikely for the CFB model to converge. Thus the 0.1-greedy algorithm will be less effective in this scenario and the EG method will shine with an adaptively optimized parameter.

With the confidence-based exploration algorithms, we increase CTR by 47.44% over the baseline and 18.20% over all other EE methods. The improvement comes from a dynamic tradeoff between exploitation and exploration, controlled by the confidence of CTR estimation. At the early stage, this algorithm takes full advantage of exploration without wasting opportunities on established bad ads. As the CFB model learns more about the true ad performance, the focus gradually shifts to pure exploitation. As expected, larger $\epsilon$ values (more explorations) correspond to faster convergence. The overall performance of 0.9 and 0.5 are comparable, while 0.1 is slightly worse. For the same reason we mention above, 0.5 is more desirable in a real system. It increases the CTR fast enough to adjust to the changes in a dynamic marketplace.

In addition to the random feedback method, we are also interested in how the EE algorithms perform when true feedbacks are available with the deterministic approach. In this way, we are able to analyze their theoretical upper-bounds. The evaluation results are presented in Figures 5 and 6. As we can see, the trends are almost identical to the previous experiment except that they converge to higher average expected CTRs. Therefore we can conclude from these two experiments that confidence-based EE performs better than others no matter we receive partial or full user feedbacks.

For all the above EE experiments, we disable the implicit exploration by setting the default CTR to be 0. However, if we compare the best EE algorithm and the baseline with a default CTR of 1 (extreme exploration), we find that the latter performs much better in terms of both metrics in the long run. The EE algorithm achieves an CFB coverage of 24,723 entries and an average expected CTR of 0.0031, while the baseline has an CFB coverage of 64,798 entries and a CTR of 0.0048. This does not mean that extreme exploration is more effective than confidence-based exploration. As we

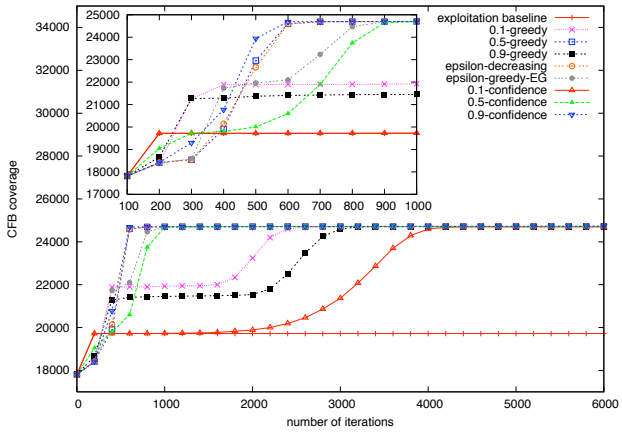


**Figure 3: CFB coverage for different EE algorithms. The small figure at the top left is a magnified plot for the first 1,000 iterations.**

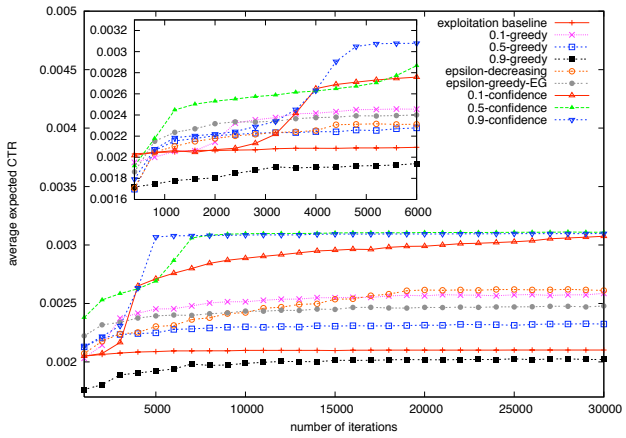


**Figure 4: Average expected CTR for different EE algorithms. The small figure at the top left is a magnified plot for the first 6,000 iterations.**

mention before, the EE algorithm can only explore the top $n$ ($n$=10) ads returned by the CFB model while the extreme exploration does not have such restriction. Therefore the comparison is only fair when we set $n$ to be unlimited. With this setup, we compare the following two algorithms again: baseline with non-zero default CTRs and confidence-based exploration with a zero default CTR. The results are shown in Figures 7 and 8.

In these two figures, the notation baseline-$x$ refers to the baseline algorithm with the default CTR set to $x$ and $x$-confidence is the same as defined before. As we can see, all the EE algorithms converge to the same CFB coverage as baseline-1, which is essentially the ground truth coverage. But in terms of average expected CTR, the EE algorithms perform much better. Note that after 20,000 iterations, 0.9-confidence no longer increases its CFB coverage because it has reached the upper limit. However, its average expected CTR continues to improve and eventually surpasses baseline-1. Because of the randomness in the simulated user feedbacks, the CFB threshold of 100 impressions may not be enough to guarantee reliable estimation of CTR. Therefore,

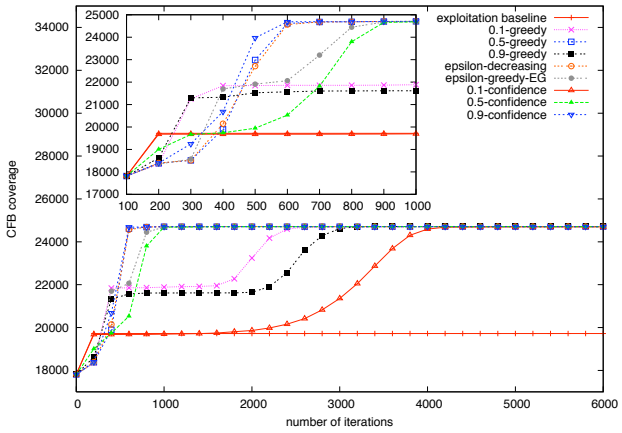


**Figure 5: CFB coverage for different EE algorithms with the deterministic feedback method.**

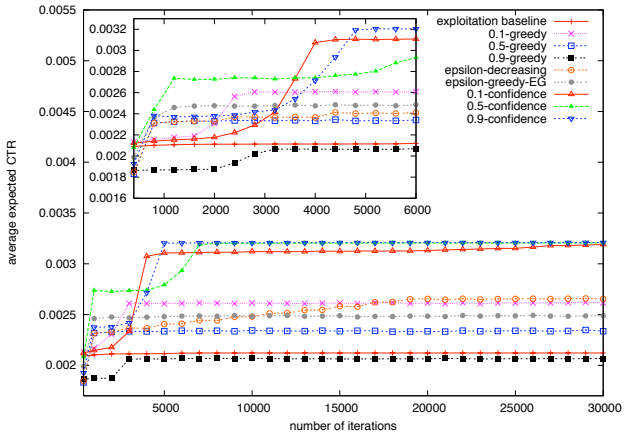


**Figure 6: Average expected CTR for different EE algorithms with the deterministic feedback method.**

even for ads with impressions above this threshold, we can still benefit from exploration to obtain more accurate CFB scores to improve the ranking system. On the other hand, the baseline algorithm stops exploration once the CFB coverage converges and its CTR stabilizes.

Another reason that explicit exploration is more desirable is because the baseline algorithm with a high default CTR is not practical in reality. For example, baseline-1 can only benefit from exploration and improve its CTR after all ads in the space receive enough impressions. However, this scenario will never happen as we have new ads joining the system everyday. Therefore, this algorithm may seem effective in the simulation, but is not practical in a real system.

In our final experiment, we extend the CFB model to incorporate two levels of aggregation. In this case, the CFB score is simply a linear combination of the two CTR estimations, where we use a weight of 0.7 for the (page URL, ad) level and 0.3 for the (publisher, ad group) level. In terms of CFB coverage, all the algorithms exhibit similar trends to the single-level model. The EE algorithms improve the coverage by 23.60% over the baseline. Due to space limitation, we leave out this figure here and only present the CTR comparison in Figure 9.

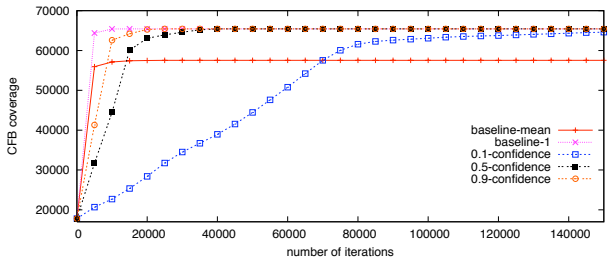


**Figure 7: NCTR coverage for implicit and explicit exploration algorithms.**

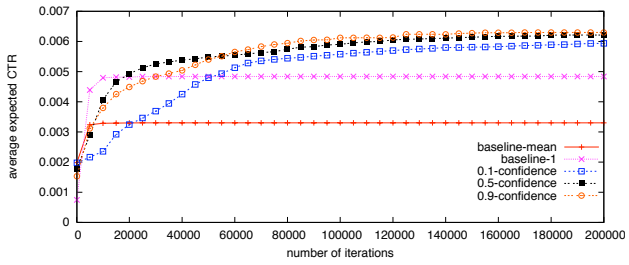


**Figure 8: Average expected CTR for implicit and explicit exploration algorithms.**

As we see, the confidence-based exploration algorithm again outperforms other approaches. We achieve the highest CTRs with parameters 0.5 and 0.9, followed by 0.1. One thing to note is that the absolute values of the converged CTRs are higher than the single-level model. This is because we obtain more reliable CTR estimation with smoothing from the higher level aggregation. In addition, exploration is more effective as every opportunity can potentially benefit multiple pages and ads from the same publisher and ad group.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we study the problem of exploitation and exploration in online advertising and propose two novel approaches that adaptively balance the tradeoff between them. In order to evaluate the performance of the proposed algorithms more accurately, we introduce an offline simulation framework that provides an isolated environment as well as mimicking the online system and user behaviors. Under this framework, we compare our algorithms with a pure exploitation baseline and two other standard EE strategies, using an industry leading performance based contextual advertising system with real online event data. The experimental results demonstrate that our algorithms, especially the confidence-based approach, perform superiorly on the measurement of ad reach and CTR in various configurations. In the future, we are going to experiment with more sophisticated confidence measurements such as the variance of posterior CTR distribution using Beta-Binomial or Gamma-Poisson models. Another direction is to combine the advantages of the confidence-based algorithm and a dynamically learned parameter using the Exponentiated Gradient method.

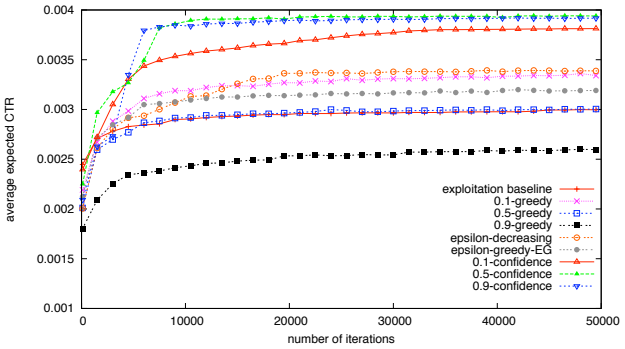


**Figure 9: Average expected CTR for 2-level CFB model.**

## 7. REFERENCES

[1] Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Explore/exploit schemes for web content optimization. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 1–10, 2009.

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

[3] John Battelle. *The search: how Google and its rivals rewrote the rules of business and transformed our culture*. Nicholas Brearley Publishing, 2005.

[4] Interactive Advertising Bureau. Internet ad revenues at $10.9 billion for first half of '09, 2009. http://www.iab.net/media/file/IAB-Ad-Revenue-Six-month-2009.pdf.

[5] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[6] Deepayan Chakrabarti, Deepak Agarwal, and Vanja Josifovski. Contextual advertising by combining relevance with click feedback. In *Proceeding of the 17th International Conference on World Wide Web*, pages 417–426, 2008.

[7] Patrali Chatterjee, Donna Hoffman, and Thomas Novak. Modeling the clickstream: Implications for Web-based advertising efforts. *Marketing Science*, 22(4):520–541, 2003.

[8] Sham Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th International Conference on Machine learning*, pages 440–447, 2008.

[9] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132:1–63, 1997.

[10] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems 20*, pages 817–824, 2008.

[11] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1952.

[12] Richard Sutton and Andrew Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[13] Chris Watkins. *Learning from delayed rewards*. PhD thesis, Cambridge University, 1989.