Pay-per-action model for online advertising

Mohammad Mahdian* Kerem Tomak†

Abstract

The online advertising industry is currently based on two dominant business models: the pay-per-impression model and the pay-per-click model. With the growth of sponsored search during the last few years, there has been a move toward the pay-per-click model as it decreases the risk to small advertisers. An alternative model, discussed but not widely used in the advertising industry, is pay-per-conversion, or more generally, pay-per-action. In this paper, we discuss various challenges involved in designing mechanisms for the pay-per-action model, and approaches to tackle some of them.

1 Introduction

Online advertising is one of the fastest growing segments in the marketing industry [2]. Currently, there are two main commodities traded in the online advertising market. These are impressions for brand awareness and clicks for traffic. Generally, advertisers is willing to pay for impressions if the aim of the advertising campaign is to increase brand awareness. However, they are more inclined to pay for clicks if the goal is to generate traffic which in turn increases the probability of a sale. In the former case, advertisers pay per impression (PPM) while in the latter they pay per click (PPC). The PPC model currently is based on a rank-by-revenue mechanism in which ads are sorted by their bid per click times click-through-rate (CTR).

With the growth of sponsored search in companies such as Google, Yahoo!, and MSN, the trend in the online advertising market has been to shift more and more of the advertising budgets toward the PPC model. This is mainly due to the fact that the PPC model reduces the risk to advertise to consumers not in the target audience of an advertiser. If this risk is high, advertisers (in particular small advertisers who are more risk averse) tend to choose the PPC model over the PPM model.

As a next step in this direction, the pay per conversion/action (PPA) model links payments to events such as sales, phone calls, or online order directly. An advertiser states his/her willingness to pay for an "action," which can encapsulate anything beyond a click. This includes pay-per-conversion, but also other things. The important distinction is that an "action" needs to be reported by the advertiser, whereas clicks are counted by the ad publisher. For many of the same reasons the PPC model has taken the market over the PPM model, we expect the online advertising market to evolve toward the PPA model in the future. Early signs of such an evolution is evident in Google's announcement of the PPA model in their AdSense platform.

In this paper, we discuss the PPA model, the advantages it offers, and issues that need to be resolved to successfully apply this model. Our focus is on theoretical questions regarding incentive issues facing the advertisers in reporting the true action data. We show how the mechanisms proposed for the PPC model can be adapted to cope with challenges specific to the PPA model.

---

*Yahoo! Research, 2821 Mission College Blvd., Santa Clara, CA 95054, USA. Email: mahdian@alum.mit.edu.

†Yahoo! Inc., 4401 Great America Parkway, Santa Clara, CA 95054, USA. Email: kerem@yahoo-inc.com.

2 The pay-per-action model

The interaction of a user with an ad publisher like Yahoo! or Google starts with the user requesting a page from the publisher that contains ads. This results in an ad *impression* for the ads displayed on the page. The user might then click on an ad, resulting in a *click-through*. Beyond this point, the user leaves the domain of the publisher and enters the advertiser's web site. In her interaction with this web site, the user might perform certain *actions* that are valuable to the advertiser, such as filling out a form, signing up at the web site, calling a phone number listed on the web site, or purchasing a merchandise. In the PPA model, the advertiser can make payments contingent on not only impressions and click-throughs, but also actions. For example, the advertiser can offer to pay 0.1 cents for every impression of their ad, 10 cents every time their ad is clicked on, plus $40 every time the user fills out a credit card application on their web site. The auction mechanism extracts all such bids from the advertisers, decides which ads to show, and how much each advertiser should be charged, depending on whether the ad is clicked on, and whether the advertiser reports that the ad has resulted in an action. A natural generalization of the common *rank-by-revenue* mechanism for the PPC model is to rank the advertisers based on their bid per impression, plus their bid per click times their click-through rate (CTR), plus their bid per action times their action rate.[1]

The major factor that distinguishes the PPA model from the PPC or PPM model is that an action takes place outside the scope of control of the publisher. Therefore, the publisher needs to rely on the advertiser to report the actions that take place (perhaps through an automatic software agent supplied by the ad publisher), whereas click-throughs are counted by the ad publisher. In fact, even the definition of an action can be different from one advertiser to another.

Another distinction between the PPA model and PPC or PPM models is in the *timing* of events. An impression takes place instantaneously after the user requests a page and the publisher decides which ads to display on the page. Also, a click-through often happens shortly after (if at all). However, an action such as buying a merchandise might take place days or even weeks after the user sees the ad. This makes the job of linking a particular action to an ad difficult. There are methods, such as using post-purchase surveys, or using the cookie technology to link the two, but the data obtained this way is inherently more noisy than the click-through or impression data.

To discuss the advantages of the PPA model, we need to understand what generates value for an advertiser. In general, there are two factors that the advertisers value:

- *Attention.* Many advertisers, particularly brand advertisers, mainly seek attention from the users. As attention is difficult to measure, various other measures such as impressions, click-throughs, or other actions can be used as proxies for attention.
- *Conversion.* A conversion is defined as any action that directly brings in some revenue. This often means buying a product from the advertiser's web site. However, there can be many other types of conversion, depending on the type of the advertiser. For example, for an ad portal, a click on one of the ads listed on the page can be considered a conversion.

There is a spectrum of advertisers, from purely attention-seeking (such as big brands, e.g., auto manufacturers) to purely conversion-seeking (e.g., small online shops).

Advantages of the PPA model. In the following, we list several advantages the PPA model offers over the more restrictive PPC and PPM models.

- *Trust requirement.* In the PPM model, as in traditional magazine advertisements [3], the advertiser needs to trust the publisher to count the number of impressions of their ad. The situation is better for the PPC model, but still various technical difficulties produce discrepancies between the click statistics on the publisher side and on the advertiser side [5]. In the PPA model, this issue is completely eliminated, as it is the advertiser who counts the number of actions.

---

[1] To completely specify the mechanism, we need to specify a payment scheme as well. The payment scheme we will consider is similar to the generalized second price auction *in expectation.* The details of this issue will be discussed in later sections.

- *Expressiveness.* Clearly, the PPA model is a more expressive bidding language than the PPC model. It is not hard to construct examples to show that if an advertiser cannot change her bid too frequently (which is often the case, either because the burden of frequently updating bids is too high for the advertiser, or because of the limits imposed by the publisher), this expressiveness can result in a higher utility for the advertiser. This is explained in a simple toy model in Appendix A.
- *Reducing risk.* In addition to increasing the advertisers' utility, the PPA model can reduce the risk to (some) advertisers, as the example in Appendix A illustrates.
- *Click fraud.* Click fraud is a phenomenon that has plagued the pay-per-click model for selling online advertisement [6, 7]. By definition, a fraudulent click is one that is done without the intention of buying a product. Therefore, an obvious remedy for the click fraud problem (for conversion-seeking advertisers) is to ask the advertisers to report clicks that lead to a conversion, and charge the advertiser only based on those clicks. Given the data about which clicks lead to a conversion, publishers such as Google or Yahoo! can not only eliminate click fraud for the involved advertisers, but also find partner web-sites that are frequent targets of click fraud (perhaps because the fraud is committed by their owners), and discount their value for other advertisers as well.

Challenges of the pay-per-action model. The PPA model assumes that the advertisers voluntarily provide the action data to the publisher. However, there are three main reasons for advertisers not to provide a truthful report of the action data to the publisher:

- *Strategic reasons:* Advertisers might be able to increase their utility by misreporting the actions. For example, if the advertiser is charged a fixed amount per action, she might benefit from not reporting some of the actions.
- *Cost of gathering data:* It might be costly to gather data about which clicks lead to action, especially because an action has a different meaning for each advertiser, many advertisers do not have the software means to track all the actions of their users, and the data is inherently noisy.
- *Cost of disclosing data:* many big advertisers treat the conversion data as confidential information that is valuable to them and their competitors, and therefore might not be willing to share this data with a publisher like Yahoo! or Google.

In the next section, we discuss the strategic factor, and show that in a simple model based on the click-fraud-resistant learning algorithms introduced by Immorlica et al. [4] combined with a participation fee, advertisers cannot gain any significant amount by misreporting the actions.

3 The incentive problem

In this section, we discuss the problem of mechanism design in the PPA model with the aim of providing incentive for advertisers to reveal the action data truthfully to the auctioneer (the publisher). A major step toward this goal was taken in the paper of Immorlica et al. [4] on click fraud. We start by briefly explaining their result and its implication for the PPA model, and then move on to a few specific problems in the PPA model.

3.1 Click-Fraud Resistant CTR Learning

Immorlica et al. [4] study a setting with one advertiser and one ad slot. The advertiser is interested in displaying an ad in the ad slot. In a PPM model with an auction mechanism such as generalized second price, the advertiser wins the slot if his bid per impression is more than a threshold p, and pays p per impression. The value of p is often the bid of the next advertiser or the reserve price. The details of how the mechanism computes p is irrelevant to our discussion; all we need to know is that p is independent of the bid or other characteristics of the advertiser. Similarly, in a PPC model, if the advertiser has a bid b_c per click, and our estimate of the click-through-rate of the advertiser (the probability that an impression of the ad leads to a click) is CTR, then the ad will be shown if $b_c \times CTR \geq p$, and the advertiser will be charged an amount equal to p/CTR if a click occurs.[2] Intuitively, this means that assuming that the estimate CTR is accurate, the advertiser pays an expected amount of p per impression. If this is the case, then fraudulent clicks should not be able to increase the average cost per impression to the advertiser. This, however, assumes that the estimate CTR is accurate, which is not a reasonable assumption, especially in a scenario where an adversary injects fraudulent clicks. The main result of Immorlica et al. [4] is that if the algorithm used to learn CTR is from a class of algorithms termed *click-based algorithms*, then the conclusion is indeed true: fraudulent clicks cannot increase the *average* cost per impression to the advertiser by more than a negligible amount.

Immorlica et al. [4] also observe that their result applies to *self-inflicted* fraud as well, i.e., if an advertiser creates fraudulent impressions that lead or not lead to clicks, he cannot change his average cost per impression by any non-negligible amount. This, taken in the context of a PPA model (replacing clicks in the argument by actions), implies that in a PPA model with a payment rule similar to the one for the PPC model and an action-rate learning algorithm from a suitable class of algorithms, the advertiser cannot change his average cost per impression by any non-negligible amount. There are, however, three issues that are left unanswered by this result:

- The payment rule when payments are associated with more than one type of event (impression, click, action): The model studied by Immorlica et al. [4] assumes that the advertiser has a bid b_a per action, AR is the estimated action rate for this advertiser, and p is the price-per-impression of the ad slot. In this setting, the mechanism displays the ad if $b_a \times AR \geq p$, and charges the advertiser an amount equal to p/AR per action. However, in general the advertiser might want to specify a bid per impression b_m, a bid per click b_c, and a bid per action b_a (or perhaps even different bids for different types of action). In this case, the mechanism must display the ad if $b_m + b_c \times CTR + b_a \times AR \geq p$, and following the logic of a generalized second price auction [1], it should charge the advertiser in such a way that in expectation, the advertiser pays p. More formally, if p_m is the price charged per impression, p_c is the price per click, and p_a is the price per action, these values must be set in such a way that $p_m + p_c \times CTR + p_a \times AR = p$. However, there are many ways this charging scheme can be designed, for example, the bids per impression, click and actions can be discounted by the same factor, or by different factors to make the expected payment equal to p. It is not clear for which, if any, of these rules the result of Immorlica et al. [4] works.
- False-name bidding: The result of Immorlica et al. [4] is asymptotic, in the sense that it shows that if an advertiser stays in the system for long enough the per-impression gain he can derive from misreporting the actions tends to zero. However, one plausible strategy for an advertiser is to stay in the system for a short time and gain from misreporting the actions, and then leave and re-enter the system with a different name.
- Timing of events: In the model studied in [4], the mechanism learns whether an impression has led to an action or not immediately after the impression, and can use this information to update the estimate of the action-rate that will be used for allocating and pricing the next impression. While this is a reasonable model for the PPC model (since most clicks take place almost immediately after the impression), it is far from being realistic in the PPA model, where an action can take place weeks after the impression.

In the following, we discuss how the above issues can be resolved. Before doing so, we need a formal definition of the setting.

---

[2] This mechanism is called a *rank-by-revenue* mechanism since $b_c \times CTR$ is the revenue that auctioneer expects from the impression, and p is the opportunity cost of this impression.

3.2 The model and the mechanism

We study a pay-per-action model similar to the pay-per-click studied in [4]. Consider a publisher who wishes to sell the right to display an ad on a web page. Here we focus on only one ad space. Each advertiser submits an ad, a bid-per-impression, and a bid-per-action[3]. Based on these values and an *action rate* that the publisher estimates for each advertiser, the publisher selects one ad, and the corresponding advertiser is charged upon an impression and when they report an action resulting from the ad. The rule the publisher uses to select this advertiser is *rank-by-revenue*: for each advertiser, an expected revenue per impression (also known as eCPM) is calculated by adding their bid per impression to the product of their action rate and their bid per action. The advertiser which has the maximum expected revenue per impression is selected as the winner. From now on, we focus on this advertiser over a period of time where she is the winner of the auction, and hold everything else in the system (e.g., the bids and action rates of other advertisers) constant. Let b_m denote the bid-per-impression of this advertiser, b_a denote her bid-per-action, and AR denote her estimated action rate. Also, let p denote the expected revenue of the next bidder. In other words, p is the opportunity cost of an impression in this ad space. Therefore, as long as $b_m + AR \times b_a \geq p$, the slot is allocated to the advertiser. Furthermore, the publisher must decide on a price p_m that the advertiser is charged per impression, and a price p_a that she is charged when she reports an action. These prices can change over time, but the properties we seek to satisfy are the following:

(a) Price per impression should always be less than bid per impression ($p_m \leq b_m$).

(b) Price per action should always be less than bid per action ($p_a \leq b_a$).

(c) The average amount the advertiser is charged per impression over a long enough period of time should be asymptotically p, *no matter how the actions are reported.*

The payment rule. If we ignore strategic reporting of actions by the advertiser, and assume that the action rate AR is constant over time, there are many possible payment schemes that satisfy the above conditions. Perhaps the most natural is the *proportional payment rule*, which proportionally discounts the prices per impression and per action to make the expected revenue equal to p. In other words, proportional payment rule charges

$$p_m = \frac{b_m p}{b_m + AR \times b_a} \qquad \text{and} \qquad p_a = \frac{b_a p}{b_m + AR \times b_a}. \tag{1}$$

Unfortunately, the above payment rule does not satisfy the property (c) above if the advertiser reports actions strategically, or if the action rate changes over time. We define an alternative payment scheme, which we call the *action-discounted* payment rule, for which we can prove that the above properties hold. This payment rule is as follows:

$$p_m = \min(b_m, p) \qquad \text{and} \qquad p_a = (p - p_m)/AR. \tag{2}$$

Intuitively, this payment rule first discounts the actions, and then the impression to the point where the expected payment per impression becomes equal to p.

Learning action rates. There are many plausible methods for learning the action rates. For our result, we consider the following general class of learning algorithms, which we call *action-based* algorithms. The algorithm estimates the AR of the ad for the current impression as follows: Label the previous impressions, starting with the most recent, by $1, 2, \ldots$. Let r_i be the number of impressions among impressions 1 through $i-1$ that lead to an action. The learning algorithms we are interested in are parameterized by a discounting

---

[3] For simplicity, we have restricted ourselves to the case where advertisers bid only per impression and per action. Our results generalize in a straightforward manner to the more general setting where advertisers bid per click as well as per impression and action.

function $\delta(r_i)$. We assume that δ is a *decreasing* function, which means that the learning algorithm emphasizes recent history over more distant history. Furthermore, for technical reasons, later on we will assume that δ is so that $\sum_{i=0}^{\infty}(i+1)\delta(i)$ is finite. Let h_i be an indicator variable for the event that the i'th impression lead to an action. The learning algorithm then computes

$$AR = \frac{\sum_{i=1}^{\infty} h_i \delta(r_i)}{\sum_{i=1}^{\infty} \delta(r_i)}. \tag{3}$$

In other words, if $y_i(h)$ denotes the number of impressions in the history h after which there has been exacly i impressions that have lead to an action, the above estimate can be written as follows:

$$AR = \frac{\sum_{i=0}^{\infty} \delta(i)}{\sum_{i=0}^{\infty} \delta(i) y_i(h)} \tag{4}$$

The main point about this class of AR learning algorithms is that the discount factor δ only depends on the number of actions r_i since the impression, and not on other factors such as i or the actual time spent since that impression. Also, in the above expressions, the summations are for every i from 1 to ∞. This is ambiguious, since the advertiser has not been always present in the system. To remove this ambiguity, the algorithm assumes a *default* infinite history for every advertiser that enters the system. We denote this default history by h^*. This default sequence could be a sequence of impressions all leading to actions, indicating that the newly arrived advertiser is initialized with an AR equal to one, or it could be a sequence indicating a system-wide default initial AR for new advertisers, or even a sequence that depends on a one-time premium the advertiser might pay upon arrival (as we will discuss later). For most common learning algorithms, the discount factor becomes zero or very small for far distant history, and hence the choice of the default sequence only affects the estimate of the AR at the arrival of a new advertiser.

False-name bidding. We model the possibility of false name bidding by allowing the advertiser to quit and re-enter the system under a different name at any step she pleases, thereby resetting her action history to the default history h^*. Note that we do not consider the possibility that the advertiser remains in the system and also re-enters under a different name, since in our setting it can only hurt the advertiser to place two bids for the same ad space.

Our approach for tackling this problem is to charge each advertiser a fixed premium for entering the system, and stop displaying ads whose action rate drops below a certain minimum rate $\lambda_{\min}$. Intuitively, the premium is set at a level so that an advertiser cannot gain by entering the system and not reporting any action, until her action-rate drops below the threshold. The premium is a small one-time fee, so it does not affect honest advertisers who stay in the system for long. Also, all or part of the premium can be refunded to the advertiser upon leaving the system, depending on the advertiser's action rate upon leaving.[4]

The exact value of the premium depends on the minimum allowable action rate $\lambda_{\min}$, the discounting function δ used by the action-rate learning algorithm, and the initial history h^*. As an example, in the case where the action-rate learning algorithm estimates the action rate by averaging over the last k actions (i.e., $\delta(i)$ is one for $0 \leq i < k$ and zero otherwise) with an initial history corresponding to an AR of 1, it is enough to charge a premium of

$$kp(\frac{1}{\lambda_{\min}} - 1).$$

Note that the amount of premium increases if the threshold $\lambda_{\min}$ is decreased, or if the value of k increases, which intuitively corresponds to increasing the robustness of the learning algorithm.

Timing. A simple fix to the timing problem is to use any of the action-based algorithms for learning the action rate using the data available at the moment the estimate is needed, and *re-adjust* previous payments

---

[4] In other words, the premium can be thought of as the fee for "buying" the default history h^*. Upon leaving the system, the advertiser can sell her current history back to the auctioneer.

Input: bid-per-impression b_m and bid-per-action b_a of the winner;
opportunity cost p for each impression of the ad space

Parameters: decreasing discounting function δ; default history h^*; minimum action rate $\lambda_{\min}$

Upon the arrival of the advertiser:
Let the history be $h := h^*$;
Calculate AR using (4);
Charge a one-time fee equal to

$$Premium := \frac{\max(0, p - b_m)}{\delta(0)\lambda_{\min}} \sum_{i=0}^{\infty}(i+1)\delta(i). \tag{5}$$

On each impression:
If $b_m + AR \times b_a < p$ or $AR < \lambda_{\min}$, the advertiser is no longer the winner;
Otherwise, allocate the impression to the advertiser, and charge her $p_m := \min(p, b_m)$;
Update h by appending a zero at the end;
Recalculate AR using (4).

When the advertiser reports an action:
Update h by setting h_i to 1, where i is the impression that has lead to this action;
Recalculate the cost of actions after impression i using the updated history;
Let Δ be the total decrease in the cost of actions after impression i;
Let AR_i denote the action rate computed using (4) with the history up to impression i;
Charge the advertiser $(p - p_m)/AR_i - \Delta$;
Recalculate AR using (4).

Figure 1: The PPA Mechanism

every time a new action is reported (e.g., by refunding part of the charge for a previous action, if the new information reduces the payment for that action). We explain this with the following simple example: assume we use the learning algorithm that estimates the action rate as 1 divided by the number of impressions since the last impression that lead to an action. Also, assume the advertiser has only specified a bid on actions (i.e., no bid on impressions or clicks). With this learning algorithm, every time an action corresponding to an impression i is reported, if i is the latest impression for which an action is reported, then the advertiser will be charged an amount equal to p/AR, where AR is the estimate of the action-rate at the time of impression i. In other words, the charge corresponding to this action will be equal to $p \times k$, where k is the number of impressions before impression i and after the last impression previously reported to lead to an action. If i is not the latest impression for which an action is reported, then this impression should be charged using a similar formula, but in addition, the charge corresponding to the first impression after i for which an action is previously reported should be adjusted. Doing the calculations, it is easy to see that for this particular learning algorithm, this adjustment cancels out the charge for i; in other words, in this case action i will not be charged, since all the charge corresponding to this action are previously paid.

3.3 Main Result

The techniques discussed in the previous section for addressing the incentive problem can be summarized in the mechanism sketched in Figure 1. Note that the statement of this mechanism is focused on one advertiser, presumed to be the winner (i.e., having an action rate of at least $\lambda_{\min}$ and an eCPM of at least p). Our result shows that for a long enough period of time when this advertiser is the winner, if the opportunity cost p is constant, the way the advertiser reports the actions does not significantly change her average cost per impression, and therefore the advertiser does not have any significant incentive to report the actions strategically.

The following theorem summarizes this argument.

Theorem 1. *Consider a rank-by-revenue system as sketched in Figure 1, and assume the discounting function* δ *and* h^* *are so that the summations* $\sum_{i=0}^{\infty}(i+1)\delta(i)$ *and* $\sum_{i=0}^{\infty}\sum_{j=i}^{\infty} y_i(h^*)\delta(j)$ *are finite.*[5] *Consider a sequence of* n *impressions during which an advertiser* A *wins the auction (possibly with quitting and re-entering the system), and assume that the opportunity cost* p *does not change during this time period. Then no matter how the advertiser reports the actions, the total amount charged to the advertiser in this time period divided by* n *is at least* p*. Furthermore, if the advertiser does not quit and re-enter, this value tends to* p *as* n *tends to infinity.*

Proof. First, assume that the advertiser does not quit and re-enter the system during the n impressions. We prove that the average cost per impression is at least p. Similarly to the proof in [4], we prove the result using a "charging" argument. We distribute the payment for each action over the impressions preceding it, and then bound the expected total charge to any single impression due to the actions after it. First, note that when the advertiser reports an action resulting from an impression, the mechanism corrects the prices of all actions after this impression. Therefore, the ordering of the reports does not matter, and we can assume without loss of generality, that the advertiser reports all the actions at the end of the n impressions, and is charged for all of them at that point.

Using Equations (2) and (4), the amount the advertiser is charged for an action on impression i can be written as

$$(p - p_m) \times \frac{\sum_{j=0}^{\infty} \delta(j) y_j(h^i)}{\sum_{l=0}^{\infty} \delta(l)},$$

where h^i denotes the history up to the impression i. We distribute this charge among the impressions before i as follows: each impression i' before i gets a charge equal to

$$(p - p_m) \times \frac{\delta(j)}{\sum_{l=0}^{\infty} \delta(l)},$$

where j is the number of impressions between i' and i that lead to an action. Clearly, summing over these charges yields the correct value for the cost of the action on impression i. Therefore, the total cost of actions is precisely equal to the total charge that all impressions (including the "imaginary" impressions in the default history h^*) receive.

We start by bounding the total amount charged to the imaginary impressions in the default history h^*. Any such impression i is charged a total value of at most

$$(p - p_m) \times \frac{\sum_{j=r_i(h^*)}^{\infty} \delta(j)}{\sum_{l=0}^{\infty} \delta(l)},$$

where $r_i(h^*)$ is the number of impressions after i in h^* that lead to actions. Summing this over all impressions in h^* and using the notation $y_i(h^*)$, the total sum charged to impressions in h^* is bounded by

$$(p - p_m) \times \frac{\sum_{i=0}^{\infty}\sum_{j=i}^{\infty} y_i(h^*)\delta(j)}{\sum_{l=0}^{\infty} \delta(l)}.$$

This sum is bounded by our assumption. Therefore, the total value charged to impressions in h^* is bounded by a constant (independent of n).

---

[5] It is implicitly shown in [4] that when h^* is a random sequence, if the first summation is finite, so is the second one.

We now bound payments charged to the "real" impressions (i.e., impressions after h^*). A fixed impression i in this set is charged a total amount of

$$(p - p_m) \times \frac{\sum_{j=0}^{r_i(h)-1} \delta(j)}{\sum_{l=0}^{\infty} \delta(l)} = (p - p_m)\left(1 - \frac{\sum_{j=r_i(h)}^{\infty} \delta(j)}{\sum_{l=0}^{\infty} \delta(l)}\right),$$

where $r_i(h)$ is the number of impressions after i in h that lead to actions. Summing over all n impressions, we obtain that the total charge to these impressions is $n(p - p_m)$ minus a term L which is at most

$$(p - p_m)\frac{\sum_{i=0}^{k} y_i(h) \sum_{j=i}^{\infty} \delta(j)}{\sum_{l=0}^{\infty} \delta(l)},$$

where k is the number of impressions among the n impressions that lead to actions. To bound this term, we use the fact that our mechanism does not show an ad if its action rate drops below $\lambda_{\min}$. Therefore, the action rate just before each of the k impressions that lead to actions must be at least $\lambda_{\min}$. This, using the formula (4) for computing the action rates, implies that $y_i(h) \leq \frac{\sum_{l=0}^{\infty} \delta(l)}{\lambda_{\min}\delta(0)}$ for every $i = 0, \ldots, k$. This gives the following bound on L:

$$(p - p_m)\frac{\sum_{i=0}^{k} y_i(h) \sum_{j=i}^{\infty} \delta(j)}{\sum_{l=0}^{\infty} \delta(l)} \leq (p - p_m)\frac{1}{\lambda_{\min}\delta(0)} \sum_{i=0}^{k} \sum_{j=i}^{\infty} \delta(j) < (p - p_m)\frac{\sum_{i=0}^{\infty}(i+1)\delta(i)}{\lambda_{\min}\delta(0)}.$$

By (5) the right-hand side of the above inequality is precisely the premium the mechanism charges at the start. Therefore, the total amount the advertiser is charged for the actions plus her initial premium is at least $n(p - p_m)$ and at most $n(p - p_m)$ plus a constant independent of n. This, together with the fact that the advertiser is charged p_m for each impression, shows that the total payment of the advertiser divided by n is at least p and at most p plus a positive term that tends to zero as n tends to infinity.

We now get back to the assumption that the advertiser does not quit and re-enter the system. If an advertiser quits and re-enters the system, we can still use the above argument for each consecutive segment where the advertiser is using the same identity. This implies that the total payment of the advertiser in each such segment with n_i impressions is at least pn_i. Summing up, this shows that the total payment for the n impressions must be at least np. This completes the proof of the theorem. $\square$

4 Conclusion

The contribution of this paper is two-fold: to discuss important theoretical questions in the design of incentive-compatible pay-per-action mechanisms for selling online advertisements, and to provide an answer to some of these questions. There are still many directions that remain unexplored. A few particular problems that we would like to emphasize are the following:

Cost of collecting action data. As mentioned earlier, one of the barriers in using the PPA model for selling online advertisements is the difficulty of gathering action (or conversion) data. It would be interesting to model this factor, and design mechanisms where the auctioneer can provide incentive for the advertiser to spend the cost for collecting the data. Notice that collecting the PPA data benefits not only the advertiser, but also the auctioneer, as the auctioneer can detect sources of fraud using this data, and avoid paying any commission to partner web sites that commit fraud.

Cost of disclosing data. Larger advertisers usually have the tools to track and collect the action data, but might not be willing to share this potentially valuable information with the auctioneer. It would be interesting to explore the potential of using privacy-enhancing technologies to reduce this disincentive to use the PPA model.

Using action data to improve the PPC model. One of the main reasons click fraud is an issue in online advertising is the obvious incentive of *partner web-sites* (i.e., web sites that are not owned by publishers like Google or Yahoo! but allow these publishers to display ads in return for a commission) to commit click fraud to increase their commission. For this reason, fraud is usually targeted at particular partner web-sites, and not on particular advertisers. This means that even if some percentage of the advertisers use the PPA model to buy ads, the publisher can use the action data that they provide to detect partner web sites that are targets of fraud, and alleviate the fraud problem by discounting the value of a click on such web sites. However, this creates an obvious incentive problem, as the data an advertiser provides is used to change not only his effective bid, but also the effective bid of other advertisers. It would be interesting to explore this tradeoff between incentive compatibility in reporting the action data, and the potential use of the action data in calculating discount rates for partner web sites.

Robustness vs. adaptivity tradeoff. There is a tradeoff between how robust the estimate of the learning algorithm is toward random noises in the data (affected by the length of history the learning algorithm looks at) and how quickly can the algorithm adapt to changes in the action rate caused by changes in the market. The optimal point in this tradeoff should depend on parameters such as the volatility of the market and the amount of noise. Furthermore, by the discussion in the previous section, two other parameters, namely the action-rate threshold below which the ad is dropped and the amount of premium that needs to be charged, also enter this tradeoff. A theoretical analysis of this tradeoff remains open.

References

[1] B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 2007.

[2] Forrester Research, *US Online Marketing Forecast: 2005 to 2010*, May 2, 2005.

[3] D. McCollam, *Bad Circulation: How often do newspapers and magazines goose their numbers?*, Columbia Journalism Review Publication, May 1, 2004.

[4] N. Immorlica, K. Jain, M. Mahdian, and K. Talwar, *Click Fraud Resistant Methods for Learning Click-Through Rates*, Proceedings of the First InternationalWorkshop on Internet and Network Economics (WINE), Lecture Notes in Computer Science 3828, 34–45, 2005.

[5] J. Lockhorn, *Cache Busting: Busted?*, The ClickZ Network, July 11, 2001.

[6] A. Penenberg. Click fraud threatens web. *Wired News*, October 13, 2004.

[7] B. Stone. When mice attack: Internet scammers steal money with 'click fraud'. *Newsweek*, January 24, 2005.

A An example: expressiveness of the PPA model

In this appendix, we give an example that illustrates how the added expressiveness of the PPA model can lead to higher revenue and reduced risk for the advertiser.

Assume an advertiser wants to bid for an ad slot with a reserve price of p per impression, i.e., in a PPM model, the advertiser needs to bid at least p per impression to get the slot at a price of p per impression, or in a PPA model, if the advertiser bids x per action and her estimated action-rate is α, then the ad will be shown only if $\alpha x \geq p$, and the amount the ad will be charged if an action occurs is p/α.

Assume each impression of the ad leads to a conversion with probability AR. The advertiser does not know the exact value of AR, but based on previous estimates, she knows that AR is a uniform random value in $[c_1, c_2]$. Each conversion brings a revenue of R for the advertiser, and the probability that the advertiser can detect that the conversion is resulted from an ad impression is a known value $1 - \varepsilon$. The advertiser needs to decide how much to bid for the next N auctions (i.e., the bid will not change during these N auctions).

In the above setting, it is easy to see that in the PPM model, the best strategy for the advertiser is to bid her expected value $(c_1 + c_2)R/2$. This results in an expected revenue of $\max(0, (c_1 + c_2)R/2 - p)N$. However, in the PPA model, the advertiser can submit a bid of $R/(1 - \varepsilon)$ per action, and report every action that she detects. Assume the auction mechanism uses an algorithm for learning the action rate AR that quickly converges to an estimate in $(1 \pm \delta)(1 - \varepsilon)AR$ for a small δ, and stays in this range for all N auctions[6]. This means that if the realized value of AR is such that $R \cdot AR \cdot (1 - \delta) \geq p$, then the advertiser wins the slot in all N auctions. If $R \cdot AR \cdot (1 + \delta) < p$, then the ad will quickly drop below the reserve price and disappear, and hence the utility of the advertiser in this case will be $o(N)$. Also, the probability that $R \cdot AR \cdot (1 - \delta) < p \leq R \cdot AR \cdot (1 + \delta)$ is small, and therefore the contribution of this case to the expected utility of the advertiser is small (details of this calculation is omitted here). Thus, assuming $\frac{p}{R(1-\delta)} \in [c_1, c_2]$, the overall expected revenue of the advertiser is

$$\begin{aligned}
\text{Rev} &= \Pr[AR \geq \frac{p}{R \cdot (1-\delta)}] \times \left(R \cdot \mathrm{E}[AR | AR \geq \frac{p}{R \cdot (1-\delta)}] - p \right) N + o(N) \\
&= \frac{c_2 - \frac{p}{R(1-\delta)}}{c_2 - c_1} \times \left(R \times (\frac{c_2 + \frac{p}{R(1-\delta)}}{2}) - p \right) N + o(N) \\
&\geq \frac{(c_2 - p/R)^2 N}{c_2 - c_1} - O(\delta N)
\end{aligned}$$

It is easy to see that for every $p/R \in [c_1, c_2]$, $\frac{(c_2 - p/R)^2 N}{c_2 - c_1}$ is greater than $\max(0, (c_1 + c_2)R/2 - p)N$. This means that (not surprisingly), in cases where the estimates of the advertiser of the action rate is not very accurate, the added expressiveness of the PPA model can result in higher a utility for the advertiser. Note that the PPA model is not more expressive if both advertiser and publisher have the same amount of information and can change the bid at the same rate, since in this case the advertiser can simulate a PPA mechanism and change her bid for every auction. However, often the advertisers can/do not update their bids very frequently[7]. Also, the more volatile a market is, the harder it will be to have an accurate estimate of the action rate, and hence the gain from a PPA model in such markets is higher.

Finally, the above example also illustrates how the PPA model can reduce the risk to the advertiser, as in cases where $p/R \in [(c_1 + c_2)/2, c_2]$, in a PPM model the advertiser sometimes suffers a loss (when $AR < p/R$) while in the PPA model the potential loss is negligible.

---

[6] We will discuss robustness of the learning algorithm later in the paper. For the purpose of our toy model it is enough to consider a learning algorithm that takes average over the recent history of size $O(\log N)$. For the first $O(\log N)$ steps, the algorithm can overestimate the value of AR as 1.

[7] In fact, often publishers have limits on the frequency of bid updates, or charge a fee for updating the bids frequently.