Pay-per-action model for online advertising

Mohammad Mahdian[1] and Kerem Tomak[1]

Yahoo! Research. Email: {mahdian,kerem}@yahoo-inc.com

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 [1]. 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.

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.

[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.

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 [2], 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 [4]. In the PPA model, this issue is completely eliminated, as it is the advertiser who counts the number of actions.
- *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.
- *Reducing risk.* In addition to increasing the advertisers' utility, the PPA model can reduce the risk to (some) advertisers.
- *Click fraud.* Click fraud is a phenomenon that has plagued the pay-per-click model for selling online advertisement [5, 6]. 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. [3] 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. [3] 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.

The model focuses on 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. [3] 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. [3] 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. [3] 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 $b_m + b_c \times CTR + b_a \times AR \geq p$, but it is not clear that in the event of a click or an action, how much the advertiser should be charged. There

[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.

are many ways this charging scheme can be designed to yield an expected price per impression of p (for example, the bids per impression, click and actions can be discounted by the same 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. [3] works.

- False-name bidding: The result of Immorlica et al. [3] 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 [3], 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 briefly sketch how the above issues can be resolved. The details of the proofs are omitted.

The payment rule. We consider the problem in the case where the advertiser can specify a bid b_m for impressions and a bid b_a for actions. In this case, the ad slot is allocated to the advertiser if $b_m + AR \times b_a \geq p$, where AR is the current estimate of the action rate. Consider the following charging scheme: we define

$$p_m = \min(b_m, p) \qquad \text{and} \qquad p_a = (p - p_m)/AR. \tag{1}$$

The advertiser is charged p_m for every impression, and p_a every time an action occurs. We can show that with this payment scheme, the result of Immorlica et al. [3] holds with a very similar proof. To explain the intuition, we show this fact for the case that the learning algorithm simply estimates the action rate as 1 divided by the number of impressions since the last impression that lead to an action (this simple algorithm is a canonical case of an *action-based* learning algorithm).

Consider the sequence of impressions, and two consecutive impressions i_1, i_2 in this sequence that have lead to an action (i.e., these two impressions have lead to an action, but none of the impressions between them has). Let $k-1$ denote the number of impressions between impressions i_1 and i_2. Therefore, at the time of impression i_2, the estimate for AR is equal to $1/k$. Therefore, the advertiser pays a price of $(p-p_m)k$ for the action corresponding to impression i_2. We can re-assign this payment to impressions between i_1 and i_2 (including i_2) by assigning an amount equal to $p-p_m$ to each of those impressions. In addition, each impression is charged an amount equal to p_m. Therefore, with this reassignment of charges, the cost corresponding to each impression will be precisely p.

The above argument can be generalized to more general learning algorithm that are *action-based* (defined analogously to the click-based algorithms of [3]). It is worth noting that other payment rules such as the proportional rule

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

do not yield the same result. Therefore, for the purpose of designing a mechanism that incentivizes the advertisers to reveal the correct action data, it is important to use this particular payment scheme.

False-name bidding. 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 rate. Intuitively, the premium is set at a level so that an advertiser cannot gain by entering the system and not reporting any action, until his 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.[3]

The exact value of the premium depends on the threshold for the minimum allowable action rate, the action-rate learning algorithm, and how it initializes the action rate when an advertising campaign starts. For example, in the case where the action-rate learning algorithm estimates the action rate by the average over the last k actions (i.e., AR is equal to k divided by the number of impressions it took to get the last k actions), initializing the AR to 1 (i.e., prepending the history by k impressions all leading to an action), the amount of premium can be calculated to be

$$kp(\frac{1}{\delta} - 1),$$

where δ is the threshold for minimum allowable action rate. Note that (not surprisingly), the amount of premium increases if the threshold δ 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 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

[3] In other words, the premium can be thought of as the fee for "buying" an initial high action rate. Upon leaving the system, the advertiser can sell the value of their current action rate back to the auctioneer.

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.

A potential problem with the learning algorithm used in the above example is that it consistently under-estimates the action rate. Even though this is not a problem for charging scheme, it can cause problems in the allocation step (i.e., if the action rate gets too low, the advertiser might not even get the slot). To resolve this problem, the learning algorithm should use some of the older data for which most of the actions have already occurred (e.g., it is safe to assume that if no action corresponding to a month-old impression has taken place, the impression will not lead to an action). However, one must be careful not to include the element of time in the formula used for learning, since doing so renders the algorithm *non-action-based*, which causes the fraud-resistance result of Immorlica et al. [3] to fail and opens the door for gaming the system by strategically timing the reports.[4]

The above discussions can be summarized in the following result. The proof of this result is based on the ideas mentioned above, but the details are omitted here.

Theorem 1. *Consider a rank-by-revenue system with a payment rule according to (1) and a payment re-adjustment scheme as above that charges a large enough premium at the sign up (where the value of premium is calculated as described above) and does not show ads whose action rate has drops below a certain threshold. In this system, an advertiser cannot gain more than* $o(1)$ *per impression by misreporting the actions and/or re-entering the system under other names.*

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.

[4] Still, the element of time can be used to compute the initial estimate of the action-rate, but the payments should eventually be adjusted according to an action-based algorithm.

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. Forrester Research, *US Online Marketing Forecast: 2005 to 2010*, May 2, 2005.
2. D. McCollam, *Bad Circulation: How often do newspapers and magazines goose their numbers?*, Columbia Journalism Review Publication, May 1, 2004.
3. 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.
4. J. Lockhorn, *Cache Busting: Busted?*, The ClickZ Network, July 11, 2001.
5. A. Penenberg. Click fraud threatens web. *Wired News*, October 13, 2004.
6. B. Stone. When mice attack: Internet scammers steal money with ‘click fraud’. *Newsweek*, January 24, 2005.