Optimizing Display Advertising in Online Social Networks

Zeinab Abbassi
Dept. of Computer Science
Columbia University
zeinab@cs.columbia.edu

Aditya Bhaskara
Google Research
New York
bhaskaraaditya@google.com

Vishal Misra
Dept. of Computer Science
Columbia University
misra@cs.columbia.edu

ABSTRACT

Advertising is a significant source of revenue for most online social networks. Conventional online advertising methods need to be customized for online social networks in order to address their distinct characteristics. Recent experimental studies have shown that providing *social cues* along with ads, e.g. information about friends liking the ad or clicking on an ad, leads to higher click rates. In other words, the probability of a user clicking an ad is a function of the set of friends that have clicked the ad. In this work, we propose formal probabilistic models to capture this phenomenon, and study the algorithmic problem that then arises. Our work is in the context of *display advertising* where a contract is signed to show an ad to a pre-determined number of users. The problem we study is the following: given a certain number of impressions, what is the optimal *display strategy*, i.e. the optimal order and the subset of users to show the ad to, so as to maximize the expected number of clicks?

Unlike previous models of influence maximization, we show that this optimization problem is hard to approximate in general, and that it is related to finding dense subgraphs of a given size. In light of the hardness result, we propose several heuristic algorithms including a two-stage algorithm inspired by influence-and-exploit strategies in viral marketing. We evaluate the performance of these heuristics on real data sets, and observe that our two-stage heuristic significantly outperforms the natural baselines.

1. INTRODUCTION

Advertising is one of the main sources – if not the main source – of revenue for most online social networks; e.g., online advertising comprised about 91% of Facebook's revenue in 2013. A significant portion of this revenue comes from display ads where certain business rules from traditional advertising apply. Conventional display ads (and more generally traditional online advertising methods) have not been as successful in generating revenue in social networks compared to other types of online advertising such as sponsored search advertising. To increase their revenue, online social networks have utilized a new paradigm called *social advertising*, which aims to leverage the social influence of a user on their friends. In this paper, we develop a formal model to study social advertising in display ads, and formalize the algorithmic problem faced by a social network platform. We then present new theoretical and empirical results for this problem.

Let us start with a brief review of display advertising: An online social network has multiple pages where it displays ads in the form of images, video or text. Once a user visits a page, she views an ad. Her exposure to the ad is called an "impression." Advertisers buy blocks of impressions ahead of time via contracts, choosing blocks carefully to target a particular market segment. Once the contract is agreed upon, the advertiser expects a specified number of impressions to be delivered by the social network platform over an agreed-upon time period. In addition to delivering a predetermined number of impressions, the advertiser may choose to optimize other objectives like clicks or conversions [4, 3, 2].

It has been observed, however, that conventional methods of ad allocation are not very successful in the context of online social networks [6, 24], and the paradigm of *social advertising* was developed to address this shortcoming. The goal of social advertising is to leverage social influence among users. The impact of social influence among users has been confirmed in sociological studies, statistical models, and in online randomized experiments (see [6] and references therein). A social ad has been defined as "an ad that incorporates user interactions that the consumer has agreed to display and be shared. The resulting ad displays these interactions along with the user's persona (picture and/or name along with the ad content)" [20]. In other words, an ad being shown to a person can incorporate information about others who have clicked the ad in the past (assuming their consent).

The goal of our work is to understand how social ads can affect click-through rates in social networks. Recent field studies [24, 6] show that advertising using social cues are more effective than conventional demographic and behavioral targeting methods.

Our Contributions and Techniques. The main contributions of our paper are as follows:

- We propose a formal model for social ads in the context of display advertising. In our model, ads are shown to users one after the other. The probability of a user u clicking an ad depends on the users who have clicked (or taken a certain action on) this ad so far. This information is presented to u as a *social cue* (which could be of different kinds, as we see later), thus the click probability is a function of this cue.
- We introduce the *social display optimization* problem: suppose an advertiser has a contract with a publisher for showing some number (say B) impressions of an ad. What *strategy* should the publisher use to show these ads so as to maximize the expected number of clicks?[1]

[1]We will discuss publishers' motivations for this in Section 2.1.

- We show that this optimization problem is APX hard.[2] In fact, under a complexity assumption known as the *planted dense subgraph* conjecture (widely believed to be true, see [10]), we prove that it is impossible to find a strategy that approximates the best one to a factor better than $n^{1/8-\epsilon}$, where n is the total number of users, and $\epsilon > 0$ is any constant.

- In light of the general hardness results, we develop heuristic algorithms and compare them to natural *baseline* ones. Inspired by influence-and-exploit strategies studied in [21, 18], we propose a two-stage algorithm: we first show the ad to αB users of highest influence, and then show it to users most likely to click (given the ones that clicked so far).

- Finally, we evaluate the performance of this as well as other heuristics on two real datasets with influence probabilities from Flixster.com and GoodReads.com.

Our model shares some common elements with previous work on influence maximization and viral marketing, but our aim is quite different: we care only about the users we show the ad to (which we are allowed to pick), and the *strategy* used for doing so (as opposed to creating a cascade over the *entire* graph). The strong inapproximability results mentioned above are also in contrast to constant-factor approximations known for influence maximization models.

Formally, a display *strategy* will consist of a set of users and a specificied order of showing ads, both of which are adaptive, in the sense they depend on which users have clicked the ad so far (thus it can be viewed as a decision tree). Indeed the number of such *strategy trees* can be doubly exponential in the size of the graph. Proving hardness results thus involves reasoning about arbitrary adaptive strategies. This is our main theoretical contribution that might find other applications.

This is also the reason proving worst approximation guarantees is difficult – because any approximation algorithm must (implicitly) certify that there is no strategy that could obtain a value much larger than that output by the algorithm. While this may suggest that the problem is hopeless, we will see in the experiments that our two-stage algorithm (outlined above) outperforms the baseline heuristics by a large margin – typically by about 11% to 100%.

1.1 Related Work

There is a rich and diverse body of work in the intersection of display advertising, social networks, marketing and influence maximization. For purposes of exposition, we will discuss these results in three categories. The most relevant to us is work on social network advertising, which is what we intend to formalize in the contex of display advertising.

Social Network Advertising: Works on social advertising have looked at the impact of displaying social signals (cues) to users. In other words, they measure the increase in the likelihood of a user clicking an add, given she knows that her friends have already taken an action on that ad. In particular, in [6], the authors run an experiment on Facebook to measure this impact. They find that showing social cues increases the probability of clicks on fan pages. Tucker [24] studies the same problem on a different network and makes similar observations. In a recent paper [7], the authors argue that viral marketing would be more effective if a large number of ordinary users are picked as influencers. None of the above work, however, looks at how one could optimize the number of clicks, likes or conversions in display ads by leveraging these social cues.

Recently, a social ad model considering user influence, called AdHeat, has been explored [8]. In this model, the advertising platform diffuses hint words of influential users to others and then matches ads for each user with aggregated hints. They perform experiments on a real-world data set, and show that AdHeat outperforms the traditional relevance models by a large factor. Although this study shows the effectiveness of using social network information in online advertising, they do not consider active propagation of ads by the users of the social network.

Viral Marketing and Influence Maximization: The problem of influence maximization in social networks has received a lot of attention in the past decade or so, with applications to viral marketing, studying the spread of diseases, and a variety of other settings. Introduced in the seminal work of Kempe, Kleinberg and Tardos [21], the goal is to pick a small set of vertices to *influence*, with the goal of maximizing the expected number of nodes that this influence *cascades* to. We do not get into the formal definitions, but note that these works [12, 21, 18, 23] give formal ways to model the probability of a user buying a product based on her friends buying the product.

This is very similar to the way in which our work models the click probability, and our model is indeed inspired by this literature. However, as we stated earlier, our goal is to find good display strategies, which is quite different from finding good nodes from which to start a cascade. Thus it seems the algorithmic tools developed there do not apply to our setting (indeed our hardness results imply that we *cannot* obtain constant factor approximations, as in the case of influence maximization).

Another related work is the revenue maximization model [18, 22], in which a person's decision to buy a product is influenced by the set of other people who own the product, as well as the price at which the item is offered. The results in this line of work have a more economic focus, and thus have a very different flavor compared to ours.

Online display ad allocation: The problem of optimal allocation for display ads has been recently studied as an online optimization problem [16, 15, 25]. The display ad optimization model considered in these papers is similar to our model in that there is a goal of delivering the ad to a predetermined number of impressions while trying to maximize the expected number of clicks or conversions. Incorporating social influence into these settings is partly the motivation for our work.

2. MODEL

We will first give a brief background on display advertising, and thus motivate our main question of study. This will help set up the notation for describing the model formally.

2.1 Display Advertising

There are three major pricing models for online ads on the Internet: Cost-per-mille/impression (CPM), Cost-per-click (CPC), and Cost-per-action (CPA). In these models, the advertisers pay the platform (publisher) for the number of impressions, clicks, or actions[3] respectively. Even though a majority (in terms of revenue) of search advertising operates on the CPC and CPA models, a significant portion (roughly 33 percent, as of 2013 [1]) of display ads are sold based on the CPM model.

The CPM model is simple to describe: an advertiser enters in a contract with a web publisher for its ads to be shown to a fixed number of site's visitors. The advertiser may specify a segment of the market or some demographic criteria to target the ads to. The

[2] Hard to approximate to some constant factor unless $P = NP$.

[3] Actions, or conversions correspond to a specific action by the online user, e.g., purchases of a product or signs up for newsletters on a website.

contract requires the publisher to show this number of impressions. Thus the publisher can choose which users to show the ad to (and on which pages). How should he make this choice?

Note that the more clicks an ad gets, the higher the chances that the advertiser would come back for another advertising campaign. Thus it is in the publisher's interest to show the ads to users more likely to click. This is a well-recognized objective – in fact, most of the advanced display-ad platforms offer tools to optimize metrics like clicks, conversions, or even return-on-investment (ROI) while delivering a predetermined number of ads [4, 3, 2]. This motivates the question of optimizing the expected number of clicks (or conversions) subject to displaying a specified number of ads. We study this question formally in the setting of social networks.

2.2 Display ads in Social Networks

Let us consider the situation in which the publisher is a social network. The contracts now require the publisher to show an ad (from an advertiser) to a fixed number (say B) users of the social network. As we saw above, the publisher wishes to maximize the number of clicks or 'likes' the ad would receive. This, in turn, could be used for pricing the contract, or improving the customer-loyalty.

We will develop a way to model how users react to *social cues*, and use the model to optimize the number of *clicks* an ad is expected to get. Social cues can be of different kinds. In its most general form, the publisher could display to a user the entire set of friends who have clicked the ad so far. This has many problems – the first is the privacy of the users who have clicked the ad. A fix for this is (as in the experiments of [6]) is to only show users who have given consent. Even so, displaying a list is cumbersome; a realistic way is to show a small subset (say, ones with closest ties) of friends, or the fraction of friends, who clicked the ad (and consented to spreading the information). Additional information, such as the number or demographics of other people (non-friends) who clicked the ad may also be provided. We would like to have a model that is general enough to capture the probability of a user's click probability in all of these settings.

We now formally describe such a model. Let B denote the *budget*, i.e. the number of ads that are to be displayed. We show ads one by one to users. At some point of time, suppose S is the set of users who have clicked the ad. Then the probability that a user u clicks the ad is given by an *influence* function $p_u(S)$. We assume that $p_u(S)$ is increasing with S, i.e., the probability that a user clicks an ad only increases if more people click the ad (e.g. $p_u(S)$ may be a linear function or a submodular function). Given this setting, the objective of the publisher (the social network) is to find the optimal set of users and the optimal *display strategy* (described below) in order to maximize the number of clicks the ad receives.

What kind of functions $p_u(S)$ are possible? From our discussion above, the user u does not see all of S, but only a *social cue*, which is something derived from S. Thus $p_u(S)$ is only a function of the social cue that u receives. In Section 2.2.1, we will see a list of reasonable candidates for $p_u(S)$ and the corresponding social cues.

Display strategy.

Formally a display strategy is a binary decision tree of depth B, the total number of impressions. The vertex at the root is the first user to be shown the ad. If the user clicks the ad, we follow the display strategy in the left subtree, else the right subtree, and so on. Note that even if the same user appears in the tree at the same depth, the probability of him/her clicking the ad will depend on the set of users who clicked so far (which is captured by the path from the root). Given a strategy, we can define the *expected clicks*, which is the expected number of users who click the ad if the publisher follows this strategy. This can be computed by a bottom-up computation in the tree.

The caveat here is that a strategy tree typically has exponential size (since it is a binary tree of depth B), thus computing the expected clicks is non-trivial. All the algorithms we consider will have a 'succinct description' of the tree (at each step, it will be a simple computation to pick the next root), however it is still not clear how to compute the expected clicks. We will compute this quantity using Monte-Carlo simulation. This can be done efficiently, because the variance is at most k^2 (in practice it is much smaller), and thus we get a good estimate of the expectation with only a few samples. The vast number of strategies is one reason it is difficult to reason about the optimal strategy (one with the largest expected clicks).

Adaptivity vs. non-adaptivity.

We have allowed our display strategy to be adaptive (the publisher can decide who to show the ad to based on which users clicked it so far). This is reasonable in most realistic cases. There are instances in which adaptivity gives a huge advantage (B vs. B^{ε}, for small ε). We do not get into them due to space constraints.

2.2.1 Influence Functions

In social advertising, the probability of a user u clicking or liking an ad could increase depending on the knowledge that certain other users have clicked the ad in the past, due to an inherent trust in the taste or judgement of those users. This is what we capture using *influence* functions, as we defined earlier. Below we will list out some influence functions $p_u(S)$ we consider.

- *Linear influence*: Here for each pair (u, v) of users, we have a weight $w(u, v)$ (not necessarily symmetric), and $p_u(S) = c_u + \sum_{v \in S} w(v, u)$.[4] The constant term c_u could be zero for certain vertices. An interesting special case is one in which the weights $w(u, v)$ are all in $\{0, p\}$, i.e., given a graph $G(V, E)$ over users, $w(u, v) = p$ if $(u, v) \in E(G)$ and 0 otherwise. In this case, $p_u(S) = c_u + p \cdot |S \cap N_u|$. This special case is particularly interesting because it is very easy to communicate $p_u(S)$ via social cues – we can simply tell a user the number of friends who clicked the ad so far.

- *Independent Cascade Model*: This is discussed and motivated in [21]. Here as above, we have influences $p(u, v)$ for pairs of users, and $p_u(S) = 1 - \Pi_S(1 - p(u, v))$. In our context, we need to allow certain vertices u to have $p_u(S) = c_u$ for constants c_u (otherwise no one would click the ad to start with). We note that when $p(u, v)$ are all small, $\Pi_S(1 - p(u, v))$ is roughly $1 - \sum_S p(u, v)$, in which case this is a special case of linear influence.

- *Concave influence*: We have weights $w(u, v)$ as before, and have a concave function $g : R \to R$ such that $p_u(S) = g(\sum_{v \in S} w(v, u))$. Interesting examples of such functions are $g(x) = x^d$ for $d < 1$, and $g(x) = \log x$.

The linear and concave functions for influence are inspired by similar models considered in [18]. We could also have another *threshold based* functions $p_u(S)$, again inspired by [21].

- *Deterministic threshold function*: We have weights $w(u, v)$, and thresholds T_u. We have $p_u(S) = 1$ if $\sum_{v \in S} w(v, u) \geq T_u$, and 0 otherwise. We also need to have some vertices with $p_u(S) = c_u$ as explained earlier.

[4]We cap probabilities at 1, though we do not explicitly write this.

Allowing thresholds makes the problem extremely hard to approximate (and possibly unrealistic), thus we do not study algorithmic results for it.

Let us now formally define our problem.

DEFINITION 1 ((SOCIAL DISPLAY OPTIMIZATION)). *Given a tuple (U, B, p) of a set of users U, a bound B on the number of users to show an ad to, and influence functions $p_u(\cdot)$, the goal is to find a (possibly adaptive) strategy for showing the ad to B users so as to maximize the expected clicks. Sometimes we will simply refer to the problem as* display optimization.

3. HARDNESS RESULTS

In this section, we will examine the complexity of the Display-Optimization problem (in terms of approximating the objective, which is the maximum expected clicks). We show that the problem is NP-hard to approximate up to a factor $(1+\varepsilon)$ for some small constant $\varepsilon > 0$.[5] Such a result is also called APX-hardness. Then, under a stronger hardness assumption, called the planted dense subgraph conjecture, we will show that we cannot approximate the optimal display strategy problem to a factor roughly $n^{1/8}$, where n is the number of users.

We first present the latter result — strong inapproximability under planted dense subgraph assumption — because it seems to highlight the crux of the problem, which is the following: if we wish to influence k users in a network, and we wish to take advantage of the graph structure, we should be able to find a set of k users who are well connected to each other, and this is hard in general. The reasoning below will make this rough intuition formal, and also illustrate how to argue about adaptive algorithms.

3.1 Strong inapproximability

We prove that for any $\varepsilon > 0$, the Display-Optimization problem cannot be approximated to a factor better than $n^{1/8-\varepsilon}$, unless we can approximate the random-planted version of the densest k-subgraph (DkS) problem to a factor better than $n^{1/4-\varepsilon}$ (conjectured to be hard [10]).

Let B be the budget, and suppose the probability that u clicks given S is the set of vertices that have clicked before, is given by

$$p_u(S) = \min\{1,\ p_0 + c \cdot |S \cap N(u)|\},$$

where p_0 and c will be picked appropriately. So a user has an 'independent' probability p_0 of clicking,[6] and there is an increase depending on the number of friends who clicked the ad. The aim, of course, is to maximize the expected number of clicks.

The planted DkS problem is the following: let $\varepsilon > 0$ be any constant; define two distributions over graphs as follows

$\mathcal{D}_1$: pick a graph from $G(n, n^{-1/2})$ (thus the expected degree is $n^{1/2}$).

$\mathcal{D}_2$: pick a graph from $G(n, n^{-1/2})$, and a random subset P of size $n^{1/2}$. Replace the induced subgraph on P by a graph from $G(n^{1/2}, n^{-(1/4+\varepsilon)})$.

To see that $\mathcal{D}_1$ and $\mathcal{D}_2$ are statistically different, we note that:

1. For a graph in $\mathcal{D}_1$, every induced subgraph on $n^{1/2}$ vertices has average degree $\leq O(\log n)$ with probability $1 - \exp(-n^{1/2})$. (Proof follows from Lemma 3.)

2. For a graph in $\mathcal{D}_2$, there exists an induced subgraph on $n^{1/2}$ vertices and average degree $\Omega(n^{1/4-\varepsilon})$ with probability $1 - \exp(-n^{1/2})$.

CONJECTURE 1 (PLANTED DENSE SUBGRAPH CONJECTURE). *Given a graph G, it is not possible in polynomial time to tell (with probability $> 2/3$) if $G \sim \mathcal{D}_1$ or $G \sim \mathcal{D}_2$. [10]*

Our theorem is now the following

THEOREM 2. *Assuming Conjecture 1 (for some $\varepsilon \in (0, 1/16)$), it is not possible to approximate the Display-Optimization problem to a factor better than $n^{1/8-\varepsilon}$ in polynomial time.*

PROOF. The reduction uses the same graph G (drawn from either $\mathcal{D}_1$ or $\mathcal{D}_2$), with parameters as follows:

$$B = n^{1/2}\ ;\ p_0 = \frac{1}{n^{1/8-\varepsilon/2}}\ ;\ c = \frac{1}{n^{1/8-\varepsilon/2}}.$$

We show that $\max \mathbb{E}[\#\text{clicks}]$ is (a) $O(\log^2 n) \cdot n^{3/8+\varepsilon/2}$ if $G \sim \mathcal{D}_1$, and is (b) $\Omega(n^{1/2})$ if $G \sim \mathcal{D}_2$, with high probability. These two claims easily imply the theorem.

It is easier to see (b). Suppose we are given a graph $G \sim \mathcal{D}_2$. Suppose P is the planted set of vertices, and suppose we show ads to the vertices in P (in a random order). Consider the situation after we show the ads to half the vertices in P. Of these vertices, $(B/2)p_0 = (1/2) \cdot n^{3/8+\varepsilon/2}$ vertices will have clicked the ad in expectation (and with very high probability, at least half this number). This means that for every remaining vertex in P, at least $\Omega(n^{1/8-\varepsilon/2})$ of its *neighbors* will have clicked the ad w.h.p. (Here we are using the fact that the planted subgraph is random and has degree $n^{1/4-\varepsilon}$). Thus by the choice of c, each subsequent vertex will click the ad with probability $\Omega(1)$, thus the expected number is $\Omega(n^{1/2})$ with high probability.

Now consider $G \sim \mathcal{D}_1$, and let $v_1, v_2, \ldots, v_B$ be any sequence of B users. Now suppose a display strategy shows the ad to these users in this order. Let S_i be the subset of $\{v_1, v_2, \ldots, v_{i-1}\}$ that clicked. We can upper bound $p_{v_i}(S_i)$ as:

$$p_{v_i}(S_i) = p_0 + c \cdot |S_i \cap N(v_i)| \leq p_0 + c \cdot |\{v_1, \ldots, v_{i-1}\} \cap N(v_i)|.$$

Now for $j \geq 1$, define vertex v_i to be in *level* j if $|\{v_1, \ldots, v_{i-1}\} \cap N(v_i)|$ lies in the interval $[2^{j-1}, 2^j)$.[7] Then, Lemma 3 (proved below) shows that with high probability ($\geq 1 - 1/n^2$, say), for *every* sequence $v_1, v_2, \ldots, v_B$, the number of v_i in level j is at most $\frac{O(\log n) n^{1/2}}{2^j}$.

Thus consider any (adaptive) display strategy. Suppose it shows the ad to users $v_1, v_2, \ldots, v_B$. Now divide these users into levels as above, and consider some level j. By the above, there are at most $O(\log n) n^{1/2}/2^j$ users in level j. Thus the expected number of these who click on the ad is

$$\frac{O(\log n) n^{1/2}}{2^j} \cdot \left(p_0 + \frac{2^j}{n^{1/8-\varepsilon/2}}\right) \leq O(\log n) \cdot n^{3/8+\varepsilon/2}.$$

By Chernoff bounds, the probability that the number who click is twice the expectation is $< 1/n^4$ in this case. Thus the total number of clicks is at most $O(\log^2 n) n^{3/8+\varepsilon/2}$ with probability at least $1 - 1/n^4$. This then implies that the expected number of clicks is at most $O(\log^2 n) n^{3/8+\varepsilon/2}$.

Note that the proof holds for every display strategy, thus concluding the proof. □

[5] Formally, it means that unless $P = NP$, it is impossible to tell if the optimal display strategy has expected clicks equal to M or expected clicks $\leq M/(1+\varepsilon)$ for some parameter M.

[6] This is necessary for the clicking to *kick off*. We can simulate this by adding a new user who clicks with probability 1, and is connected to everyone with an edge of weight p_0.

[7] Include vertices with no edges to their predecessors into level 1.

It only remains to show Lemma 3.

LEMMA 3. *Let $G = (V, E) \sim \mathcal{G}(n, n^{-1/2})$, and define $M = n^{1/2}$. Then with probability $(1 - 1/n^2)$, we have that for every $v_1, v_2, \ldots, v_M$, the number of edges in the induced subgraph is at most $(2 \log n)M$. Consequently, the number of v_i with $> t$ neighbors among $\{v_1, \ldots, v_M\}$ is at most $(4 \log n)M/t$.*

PROOF. Consider some $v_1, \ldots, v_M$. The probability that there are $\geq k$ edges is essentially

$$\binom{M^2/2}{k} p^k < \left(\frac{M^2 pe}{2k}\right)^k.$$

Now for $k = 2M \log n$, since $Mp = 1$, we have the probability above to be less than $e^{-2M \log n}$. Thus we can take a union bound over all choices of $v_1, \ldots, v_M$ (there are only $n^M = e^{M \log n}$ of them), completing the proof. □

3.2 APX hardness

The theorem here is the following. Though the factor is much weaker, it is based on a much more standard assumption (NP hardness). This is the reason we include the result.

THEOREM 4. *There is an absolute constant $\epsilon > 0$ such that it is NP-hard to approximate the Display Optimization problem to a factor $(1 + \epsilon)$.*

We apply a result about the complexity of the k-uniform set cover problem (set cover in which all sets have size precisely k) and present a reduction from this. Formally, an instance of this problem consists of a family $\mathcal{S}$ of m subsets $S_1, \ldots, S_m$, each of size k, over a ground set of elements $[n] := \{1, 2, \ldots, n\}$. The goal is to find a subfamily of $\mathcal{S}$ of minimum size that covers all of $[n]$. The following hardness result is known:

PROPOSITION 5. *[14] For every choice of constants $s_0 > 0$ and $\varepsilon > 0$, there exists a k (depending on ε) and instances of $k-$uniform regular set cover with n elements on which it is NP-hard to distinguish between the case in which all elements can be covered by $t = \frac{n}{k}$ disjoint sets (called YES instances), and the case in which every $s \leq s_0 t$ sets cover at most a fraction of $1 - (1 - \frac{1}{t})^s + \varepsilon$ of the elements (called NO instances).*

Reduction. We now give a reduction from k-uniform set cover to Display Optimization. Let V be the set of elements and $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ be a family of sets over V each of size k. The instance of Display-Optimization we construct is as follows: for each element $v \in V$, we have one user. For each $S_i \in \mathcal{S}$, we have a set U_i of $4 \log n/p$ users, where $p = 1/n^{1/4}$. The edges are as follows. We place an edge between $v \in V$ and $u \in U_i$ iff $u \in S_i$ (thus such a v has an edge to all the users in the 'group' U_i). Now the influence functions $p_u()$ are defined as follows. For users $u \in U_i$, $p_u(S) = p$ (i.e., these users click the ad with probability p independent of the rest). For users $v \in V$, we have

$$p_v(S) = \begin{cases} 1 \text{ if } |S \cap \Gamma(v)| \geq 1, \\ 0 \text{ otherwise.} \end{cases}$$

The number of users to show the ad to, i.e., the parameter B, is picked to be $(4t/p) + n$. (Recall $t = n/k$, defined above.)

We now show that if we started with a YES instance for set cover, there is a strategy which has expected clicks $\geq (1 - \delta)n + 4t$, and if we started with a NO instance, then *any* strategy has an expected clicks at most $(1-\delta')n+4t$, for some constants $\delta' > \delta > 0$. Recall that $t = n/k$ (k constant), so this will establish APX-hardness.

YES case. In this case there exists a sub-family of t sets – say $S_1, S_2, \ldots, S_t$ which cover all the elements V. Now consider the following display strategy:

1. Show the ad to precisely $4t/p$ users, according to the following algorithm: first show the ad to users in U_1 until either one of them clicks the ad, or we have exhausted all of U_1, then do the same with U_2, and so on, until the ad is either shown to $4t/p$ users, or we have one click in each each U_i for $i = 1, \ldots, t$. In the latter case, if we have not shown the ad to $4t/p$ users in total, show it to arbitrary (other) users in $\cup_j U_j$ so that the total is $4t/p$.

2. Then show the item to all the n users in V in any order.

LEMMA 6. *With probability $1 - 1/n^2$, step 1 of the algorithm ends up with at least one click in each of $U_1, \ldots, U_t$.*

PROOF. The full proof is technical, so we only give an outline in this version. The intuition is that in each of the U_i, by showing the ad to $1/p$ users, there is a probability roughly $1/2$ of some user clicking. Thus in roughly $t/2$ of the groups U_i, the algorithm will show the ad only to $1/p$ users in the group. Now a similar argument will show that in roughly $1/2$ of the rest, we require showing the ad to $2/p$ users, and in general, roughly $t/2^j$ of the groups will require showing the ad to j/p users, for $j = 1, 2, \ldots, \log n$. Thus the total number of users to show the ad to, will be

$$\sum_{j=1}^{\log n} \frac{j}{p} \cdot \frac{t}{2^j} < 2t/p.$$

By allowing some slack in each bound, we can get high-concentration versions of these, which completes the proof. □

Note that if we have one click in each U_i, $i \leq t$, then in step 2, we get n clicks. Further, the expected number of clicks in step 1 will be precisely $p \cdot (4t/p) = 4t$, and by Chernoff bounds, it will be $\geq 4t - \sqrt{40t \log n}$ with probability at least $(1 - 1/n^2)$. Thus with probability $\geq 1 - 2/n^2$, we have that in the YES case, the algorithm above gets $n + 4t - \sqrt{40t \log n}$ clicks. Thus the expectation is $\geq (1 - \delta)n + 4t$, for any constant $\delta > 0$ (for large enough n, since $t = n/k < n$). This completes the analysis.

NO case. Here we need to show that no display strategy can have expected clicks $> (1 - \delta')n + 4t$, for some absolute constant δ'. The key is to observe that an optimal strategy will (w.l.o.g.) first show the ad to users in U and then to users in V (this is because of the structure of our instance), and among the users in U, the order does not matter – the only thing that matters is the number of users in each U_i that are shown the ad.

LEMMA 7. *In any strategy, the number of users in U who click the ad is at most $Bp + \sqrt{10Bp \log n}$ with probability $\geq 1 - 1/n^2$.*

PROOF. Any strategy shows the ad to at most B users in U (B is in fact the total number of users it shows the ad to). For this lemma, it does not matter which users are shown the ad, because each user likes with probability p independent of all others. Thus the lemma follows by standard Chernoff bounds. □

Note that the bound above is $Bp = 4t + np + O(\sqrt{t \log n})$, for our choice of p. We introduce a bit of notation: we will call a group U_i "good" if at least one of the users in U_i clicks the ad.

LEMMA 8. *In any strategy, at most $40t$ of the groups are good with probability at least $1 - 1/n^2$.*

PROOF. From the way n, t, p are related, we have $B < 5t/p$, thus any strategy shows the ad only to $5t/p$ users in U. This means that the number of groups in which the ad was shown to $> 1/(2p)$ users is at most $10t$ (else we would get a contradiction).

Thus if $40t$ groups are good in total, it means that in at least $30t$ groups, the strategy shows the ad to at most $1/(2p)$ users, and it manages to have one of the users like the ad. We show that this is very unlikely – can happen with probability at most $1/n^2$. Let us now perform a finer division. For each j, let n_j be the number of groups in which the ad is shown to $1/jp$ users, for $j = 2, \ldots, (1/p)$.[8] Define these groups to be in *level* j. By a calculation similar to the above, we have that

$$\sum_{j=2}^{1/p} n_j \cdot \frac{1}{jp} < \frac{5t}{p}.$$

For convenience, denote the quantity n_j/j by C_j. Then the above inequality becomes $\sum_j C_j < 5t$. We claim that for each j, the probability that there are $> 4(C_j + \log n)$ good groups in level j, is at most $1/n^2$.

The probability that some group in level j is good, is at most $1 - (1-p)^{1/jp} \approx 1 - \exp(-1/j) < 2/j$, for $j \geq 2$. Thus the expected number of good groups in level j is at most $2n_j/j = 2C_j$. Simple concentration bounds then give the claim above (because groups being good are independent events).

This then implies that with prob. $1 - 1/n^2$, the total number of good groups is at most $\sum_j 4(C_j + \log n) < 20t + t\log^2 n/p < 30t$. This completes the proof. $\square$

Once we have Lemmas 7, 8, it is easy to see that with probability $\geq 1 - 1/n^2$, the number of users who clicked the ad in total is at most (because we are in the NO instance of set cover)

$$4t + np + O(\sqrt{t \log n}) + \left\{ \left(1 - 1/t\right)^{40t} + \varepsilon \right\} \cdot n < (1-\delta')n + 4t,$$

for some absolute constant δ' (since ε can be picked small enough). This completes the proof of APX hardness.

Deterministic thresolds.

The proof above can be modified to show that if the influence function is allowed to be hard threshold, then there is no hope of approximating. More precisely we can show:

THEOREM 9. *The Display-Optimization problem when some users are allowed to have a deterministic threshold function is NP-hard to approximate up to any polynomial (n^c) approximation factor.*

PROOF. (Sketch) We modify the reduction above, noting that in the YES case, all n vertices of V would click the ad (in our algorithm, w.h.p.), while in the NO case, at most $(1-\delta')$ fraction of V could click the ad w.h.p. (the same argument gives $1/n^{2c}$ instead of $1/n^2$ with minor changes). Now connect $M = n^{c+1}$ new users to all the users of V, and suppose these new users click the ad only if all n of V click the ad (threshold). Then in the YES case we get $M + O(n)$ clicks, while in the NO case we only get $O(n)$ clicks w.h.p. This shows inapproximability to an n^c factor. $\square$

4. ALGORITHMS

In light of the above hardness results, we cannot hope to obtain algorithms with provably good approximation ratios. However, we will describe heuristics which perform much better than the natural baseline algorithms on real life data sets. The baseline algorithms are similar to those used in the context of influence maximization (and are known to give good algorithms for special cases).

[8] Formally, we need to have the interval $(1/(j+1)p, 1/jp]$.

Most Influential Greedy Heuristic

Input: (U, B, p_u).

Output: A sequence $(a_1, a_2, \ldots, a_B)$, and set S.

Goal: Maximize $E[|S|]$.

1. **Initialize:** $S = \emptyset, A = \emptyset$
2. **For** $i := 1$ **to** B **do**
3. **Let** $a_i \in U \backslash A$ be the user maximizing top-infl(u)
4. **Let** $A := A \cup \{a_i\}$.
5. With probability $p_{a_i}(S)$, **let** $S = S \cup \{a_i\}$.

Figure 1: Most Influential Greedy Heuristic

4.1 Baseline algorithms

We present two natural heuristics for the problem, which we then use to compare the performance of our algorithms.

4.1.1 Largest probability greedy

One simple algorithm is to pick users that are most likely to click on the ad at each time step, i.e., given an input (U, B, p_u), pick users $\{a_1, a_2, \ldots, a_B\}$ as follows: at the i'th step, let $a_i \in U \backslash \{a_1, a_2, \ldots, a_{i-1}\}$ be the user maximizing $p_{a_i}(S_{i-1})$ where $S_{i-1} \subset \{a_1, a_2, \ldots, a_{i-1}\}$ is the set of users who have clicked on the ad so far. Ties are broken arbitrarily.

The problem with this heuristic is, intuitively, that it ignores the *future*. While picking users most likely to click, we may have picked ones that do not influence others (we will see examples).

4.1.2 Most influential greedy

The heuristic above ignores the influence of the chosen users on others. We can consider the other extreme: algorithms that pick the most influential users at each step (and ignore the click probability).

Given an input (U, B, p_u) for an instance of social display optimization, we define a total influence infl(u) for each user u as $\sum_{v \in U} p_v(\{u\})$. The algorithm simply picks the top B users in the non-increasing order of infl(u). One issue with the above is that it considers the total influence of a vertex on all other nodes as opposed to the influence on at most B nodes. To deal with this, we define the *total top influence* top-infl(u) for each user u as $\sum_{v \in J(u)} p_v(\{u\})$, where $J(u) \subset U$ is a subset of users v with the B largest values of $p_v(\{u\})$. The greedy algorithm is to pick the top B users in the non-increasing order of top-infl(u).

This algorithm performs poorly in instances where the click probability and the influence are negatively correlated, but is reasonable on instances in which such correlations do not occur. We thus view the algorithm as a relevant baseline for evaluating better heuristics. The algorithm is formally stated in Figure 4.1.2.

4.1.3 Worst-case examples greedy heuristics

Let us describe concrete instances in which the baseline heuristics perform badly. These will inspire the more sophisticated algorithms.

For completeness, we also note that picking B vertices at random will perform very poorly. A simple example is a path with B vertices, in which the first vertex has a click probability 1 (always clicks), and each vertex has a click probability 1 if at least one neighbor has clicked, and 0 otherwise. It is easy to see that a random order does very badly, while the linear order obtains a value B. (The largest-probability greedy recovers this.)

Next we see that even in simple linear influence models with $B = n$, there are bad examples for the two greedy algorithms

Adaptive Hybrid Heuristic

Input: (U, B, p_u).

Output: A sequence $(a_1, a_2, \ldots, a_B)$, and set S.

Goal: Maximize $E[|S|]$.

1. **Initialize:** $S = \emptyset, A = \emptyset$
2. **For** $i := 1$ **to** B **do**
3. **Let** $a_i \in U \backslash A$ be the user maximizing $p_{a_i}(S) \times$ top-infl(a_i)
4. **Let** $A := A \cup \{a_i\}$.
5. With probability $p_{a_i}(S)$, **let** $S = S \cup \{a_i\}$.

Figure 2: Adaptive Hybrid Heuristic.

above. For the largest-probability greedy algorithm, consider a path as above, but with asymmetric weights: $w(i, i+1) = 1$ whereas $w(i+1, i) = 0$ (i influences the neighbor to the right, but not the one on the left). Suppose the probability $p_u(S) = \min\{1, \alpha + i\varepsilon + |S \cap N_i|\}$, where $\alpha = n^{\delta - 1}$ for small δ, and $\varepsilon = 1/n^2$. The greedy algorithm picks vertices in the order $n, n-1, \ldots$ (because of the $i\varepsilon$ terms), and the expected number of clicks is only n^δ. However if shown in the order $1, \ldots, n$, then in roughly $1/\alpha$ steps, we see at least one click, and all vertices following that will definitely click, thus the expected value is $\Omega(n)$ for this strategy.

We can also construct easy counter examples for most-influential greedy heuristic, with two sets of vertices, one with slightly higher influence but low click probabilities, which fool the greedy strategy. We do not get into the details.

4.2 Better heuristics

4.2.1 Adaptive Hybrid Heuristic

This heuristic is based on the simplest way to take into account both the influence and the click probability – the product of the two. More specifically, given an input (U, B, p_u) for an instance of the social display optimization, the algorithm greedily picks users $\{a_1, \ldots, a_B\}$ as follows: in the i'th step, let $a_i \in U \backslash \{a_1, \ldots, a_{i-1}\}$ be the user maximizing $p_{a_i}(S_{i-1}) \times$ top-infl(a_i) where S is the set of users who clicked so far, and top-infl$(a_i) = \sum_{v \in J(u)} p_v(\{u\})$, where $J(u)$ consists of users v who have not yet been shown the ad, and who have the $(B - i)$ highest values of $p_v(\{a_i\})$. The algorithm is shown in Fig. 4.2.1.

4.2.2 Two-stage heuristic

Inspired by the idea of the the influence-and-exploit strategies in viral marketing [21, 18], and the greedy algorithm for Densest k-subgraph [10], we propose the following two-stage heuristic: follow the most-influential greedy heuristics for the first stage of the algorithm, and then switch to the largest-probability greedy algorithm in the second stage. More specifically, we can run the adaptive most-influential algorithm for the first αB steps, and then follow the naive greedy largest-probability heuristic in the last $(1 - \alpha)B$ steps. Our motivation for this greedy algorithm is to follow the intuition behind the greedy algorithm for influence maximization [21]. Although this algorithm does not provide a guaranteed approximation algorithm for this problem, we hope that this technique works well in practice, since the greedy heuristic has been very effective for influence maximization [21] [**?**, **?**, **?**]. The optimal value of α certainly depends on the influence functions and the structure of the influence among users. Such an optimal value of α can be computed by trying a range of values for α and estimating the expected number of clicks via simulations. As part of our empirical study, we report a number of insights for the optimal choice of α for various settings.

Two-stage Heuristic

Input: (U, B, p_u).

Output: A sequence $(a_1, a_2, \ldots, a_B)$, and set S.

Goal: Maximize $E[|S|]$.

1. **Initialize:** $S = \emptyset, A = \emptyset$
2. **For** $i := 1$ **to** αB **do**
3. **Let** $a_i \in U \backslash A$ be the user maximizing top-infl(u)
4. **Let** $A := A \cup \{a_i\}$.
5. With probability $p_{a_i}(S)$, **let** $S = S \cup \{a_i\}$.
6. **For** $i := \alpha B + 1$ **to** B do **do**
7. **Let** $a_i \in U \backslash A$ be the user maximizing $p_{a_i}(S)$
8. **Let** $A := A \cup \{a_i\}$.
9. With probability $p_{a_i}(S)$, **let** $S = S \cup \{a_i\}$.

Figure 3: The two-stage heuristic for specific α; the overall algorithm tries different α and picks the best

Bad examples. Note that our bad example for largest-probability greedy (path with asymmetric influence) can be modified easily to give an $n^{1-\delta}$ gap for both the heuristics above. Influence of every vertex in that example is precisely 1 – so the adaptive hybrid works exactly like largest-probability; if ties are broken badly (which is possible), this example is also bad for the two stage heuristic. However, the example is based on a *chain* of highly asymmetric influence, which is unlikely in real instances. We believe that this why our algorithms seem to perform quite well in real instances.

5. EMPIRICAL EVALUATION

In this section, we evaluate variants of four heuristics discussed in the previous section on two families of instances taken from real-world datasets. After elaborating on our datasets, we report the improvement of the two-stage and adaptive hybrid algorithms over the two baseline algorithms.

5.1 Datasets

It is important to explain how we obtain the instances of Display-Optimization (in particular the influences $w(u, v)$) from real data. There is no a priori "correct" way.

Flixster[9]**:** Flixster is a social network for rating movies. We obtained the Flixster dataset from Goyal et al's work [17, 9]. This dataset contains 13,000 users with 192,400 directed edges between them. There are 1.84 million ratings done by these users. These statistics are presented in Table 1. The influence probabilities are learned by looking at the log of user ratings with time: $< u, i, t, r >$ (meaning user u rated item i at time t with rating r). We estimate the influence probability of user u on user v as the fraction of times user v rated an item after user u had rated that item. This fraction is then normalized over all neighbors of user v to make the sum of influence probabilities equal to 1.[10]

Goodreads[11]**:** Goodreads is a social book cataloging website where users can register books to create personal bookshelves and also form friendships with each other. The dataset contains 4,654 users

[9] www.flixster.com

[10] This typically overestimates causality; for the next dataset we consider a different way to estimate influences.

[11] www.GoodReads.com

# Users	13,000
# Friendship links	192,400
# Ratings	1.84M

Table 1: Summary of Flixster Data Statistics

with 445,947 edges between them [5] (statistics in Table 2). This time, we produce the influence probabilities according to the so-called *voter model*. This was introduced by [11] and [19] to model probabilistic influence. The model explains the diffusion of opinions in a social network as follows: in each step, each node changes her opinion by choosing one of her neighbors at random and adopting that neighbor's opinion. In [13], the authors show that degree is a good predictor of influence probabilities.

# Users	592,081
# Friendship links	2,045,177
# Books	248,252

Table 2: Summary of Goodreads: owner-book information data

5.2 Experimental Setup

Here, we report the performance of our algorithms on the Flixster and GoodReads data sets. For each data set, we study the performance of these algorithms with $B = \beta n$ for four different values β, $0.02, 0.05, 0.1$ and 0.15. I.e., we set the the goal of showing the ad to 2%, 5%, 10% or 15% of the whole population, and report the results for each value of B. The way we compute the edge weights (probabilities) is described in Secton 5.1. As for the choice of the influence function, we examine the independent cascade model, and linear and concave influences. The concave functions we examined are $g(x) = \sqrt{x}$ and $g(x) = \log x$.

Finally, for each node u, the individual click probability $p_u(\emptyset)$ is drawn from a log-normal distribution with a large mean (between 0.1 and 0.45). We chose a large mean for these distributions to make sure that in the final click probability, the individual click probability is not dominated by the incremental probability due to influence.[12] The performance of the algorithms under the independent cascade model were almost the same as linear influence (as we noted, this is not surprising), thus we report the plots for the independent cascade model, and concave influences with $\sqrt{x}$ and $\log x$. The reason we report the results for both of these is to illustrate that our empirical observations are similar for seemingly very different influence functions.

As we discussed, we compare four algorithms (including two baseline heuristics). For the two-stage algorithm, we try different values of α and choose the α with maximum expected value.

5.3 Observations

The empirical results for both data sets and for all propagation cases that we ran can be found in Figures 4, 5, 6, 7, 8, and 9. In these plots, the X axis changes β where $B = \beta n$. The Y axis is the expected number of clicks during the simulation. Here we summarize our main observations in these plots:

[12]This probability could be much smaller for certain ad types. The goal of our empirical study is mainly to compare different heuristic methods. We observe that the magnitude of the click-through-rate numbers is not important for this comparison, and we expect to get similar relative performance if we scale all click-through-rates by the same factor. It is, however, important to choose the individual probability factors in such a way that their magnitude dominates that of the probabilities due to influence.

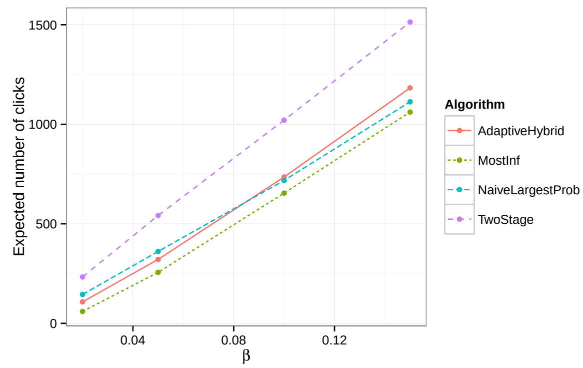

Figure 4: Performance of the heuristics on the Flixster dataset for the independent cascade model. The X axis shows β, where $B = \beta n$

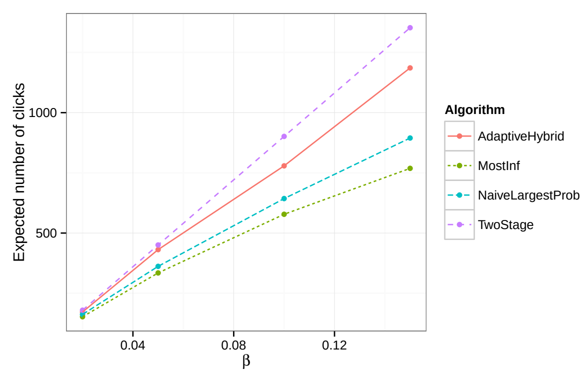

Figure 5: Performance of the heuristics on the Flixster dataset with a concave influence function, specifically $\log(x)$

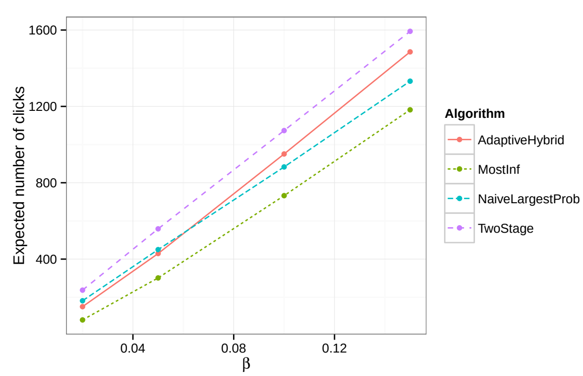

Figure 6: Performance of the heuristics on the Flixster dataset with a concave influence function, specifically $\sqrt{x}$

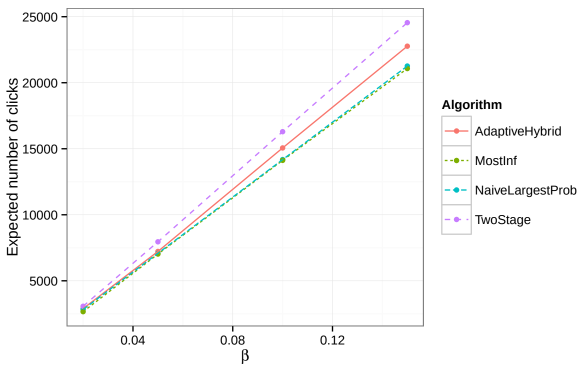

Figure 7: Performance of the heuristics on the GoodReads dataset for the independent cascade model

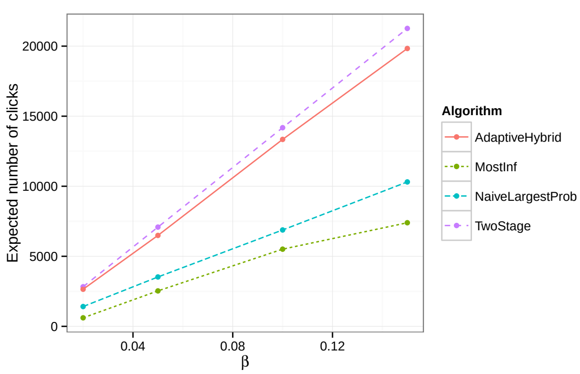

Figure 8: Performance of the heuristics on the GoodReads dataset with a concave influence function, specifically $\log(x)$

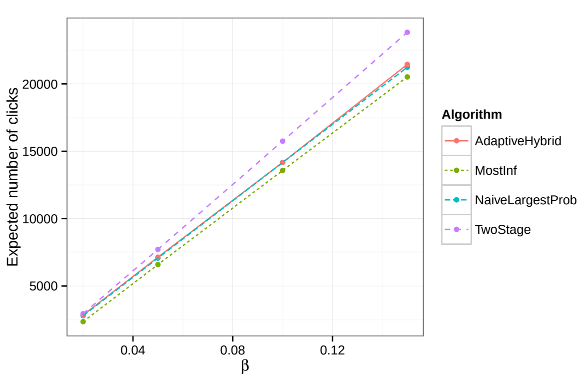

Figure 9: Performance of the heuristics on the GoodReads dataset with a concave influence function, specifically $\sqrt{x}$

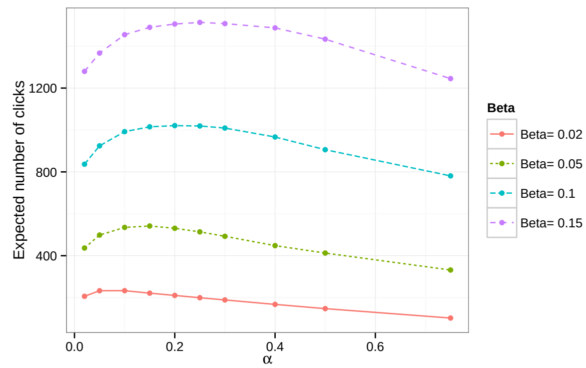

Figure 10: Two-stage heuristic for different values of α. Plot is for the independent cascade model on the Flixter dataset.

- Most notably, we observe that the two-stage heuristic algorithm consistently outperforms all the other heuristics. Across all instances, the gap between the performance of the two-stage heuristic and other algorithms increases as the the budget B increases. For example, for $\beta = 0.15$, for the Flixster dataset, the percentage increases for the two-stage heuristic from the best of other algorithms (i.e., adaptive hybrid) is around 25%, 14%, and 6% for the three different influence propagation models. For the GoodReads dataset, the percentage increases are around 7%, 5%, and 12%. The percentage increases from the output of the largest-probability heuristic (that ignores the influence function) to the output of the two-stage are 26%, 62%, and 23%. The same percentage increases for the GoodReads dataset are 11%, 100%, and 12%. The interesting parameter in two-stage algorithms is the α at which the best performance is achieved. See below for a discussion on how this behaves.

- Even our first heuristic, the adaptive hybrid greedy algorithm, outperforms the baselines for all values of β except $\beta = 0.02$ where the largest-probability heuristic is slightly better for two instances, and $\beta = 0.05$ where the largest-probability heuristic is slightly better for one instance. Again the performance increase from largest-probability heuristic to the adaptive hybrid heuristic increases as the budget increases.

- Finally, the most-influential greedy algorithm performs the worst. This was expected since it only focuses on picking the most influential users and not on their click probability.

Optimal α for the two-stage heuristic: As we discussed, in order to find the optimal α for the two-stage algorithm, we tried several values and chose the best one. The expected number of clicks for each value of α and for different budgets is plotted in Figure 11. The optimal choice of α for each instance may be related to the optimal way of influencing the network through ads. We observe that the optimal α for different instances vary from 0.05 to 0.2. It is worth noting the following points in these plots:

- The optimal choice of α increases as β increase from 2% to 15%. This suggests that with a higher budget, we can afford to spend a bit more fraction of time on 'exploration' (influencing) and attain higher expected clicks.

- The optimal choice of α decrease as the exponent of the concave influence function decreases. This is reasonable, be-

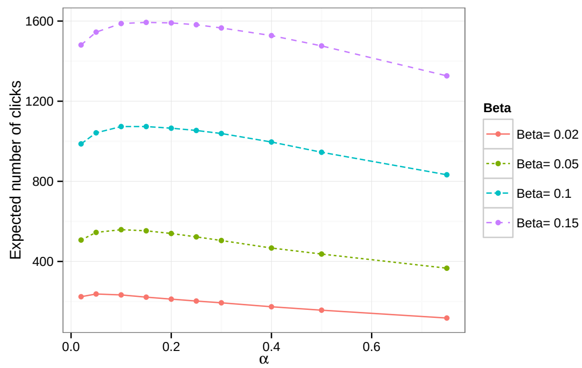

Figure 11: Two-stage heuristic for different values of α. Plot is for $\sqrt{x}$ influence function on the Flixter dataset.

cause the lesser the influence among users, the lesser the importance of the influencing steps.

- Fixing an instance (i.e., fixing β and an influence propagation model), we observe that as α increases, the expected number of clicks from the two-stage heuristic is first monotonically increasing, and then it decreases monotonically (i.e., it is unimodal). This is a curious fact which would be nice to prove in restricted models. Of course, it also implies that the optimal α can be found by trying only a logarithmic number of values of α (by essentially binary search).

6. CONCLUSION

Social advertising has emerged as a promising alternative to conventional online advertising methods. In the setting of display advertising, relying on the notion that leveraging social cues can increase clicks in online social networks, we proposed a formal model for social display ad optimization, and initiated the study of the problem of optimally allocating display ads by modeling the impact of social influence on users' decisions. We showed that the social display ad optimization problem is APX-hard and is unlikely to be approximable within a factor much better than $n^{1/8}$. On the algorithms side, we proposed new algorithms which seem to perform significantly better than the baseline heuristics on datasets from real social networks. E.g., our two-stage algorithm achieved a 11% to 100% improvement over the output of baseline greedy algorithms. We also examined the question of the optimal "influence/exploit trade-off" for the two-stage heuristic under different choices of influence functions.

As a first step towards better display advertising in social networks, our work raises many interesting questions: can we develop algorithms with provable guarantees for synthetic graphs, such as the well-studied models for social networks? What other restricted influence models can we study for which we can prove provable approximation algorithms? What are good ways to learn influence weights from real data? Can we extend the model to incorporate multiple advertisers (each having a certain number of impressions)?

Our work suggests that the structure of the (induced) graph between the target users is a crucial parameter in this model of 'social display advertising'. It suggests that the graph structure, used well, could significantly increase the expected click rate. Running experiments to test this on real advertising platforms is an interesting direction for future research.

7. REFERENCES

[1] IABinternet advertising revenue report. `http://www.iab.net/media/file/IABInternetAdvertisingRevenueReportHY2013FINALdoc.pdf` (2013).

[2] Facebook doubleclick for publishers (dfp) optimization website. `https://www.facebook.com/business/a/online-sales/ad-optimization-measurement` (2014).

[3] Google doubleclick bid manager website. `http://www.thinkwithgoogle.com/products/doubleclick-bid-manager.html` (2014).

[4] Google doubleclick for publishers (dfp) optimization website. `http://static.googleusercontent.com/media/www.google.com/en/us/doubleclick/pdfs/optimization.pdf` (2014).

[5] ABBASSI, Z., ASSADI, S., AND TAHMASBI, M. Predicting ratings in online social networks: Friends or the crowd? *Working paper*.

[6] BAKSHY, E., ECKLES, D., YAN, R., AND ROSENN, I. Social influence in social advertising: evidence from field experiments. In *ACM Conference on Electronic Commerce* (2012), pp. 146–161.

[7] BAKSHY, E., HOFMAN, J. M., MASON, W. A., AND WATTS, D. J. Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining* (2011), ACM, pp. 65–74.

[8] BAO, H., AND CHANG, E. Y. Adheat: an influence-based diffusion model for propagating hints to match ads. In *Proceedings of the 19th international conference on World wide web* (2010), ACM, pp. 71–80.

[9] BHAGAT, S., GOYAL, A., AND LAKSHMANAN, L. V. Maximizing product adoption in social networks. In *Proceedings of the fifth ACM international conference on Web search and data mining* (2012), ACM, pp. 603–612.

[10] BHASKARA, A., CHARIKAR, M., CHLAMTAC, E., FEIGE, U., AND VIJAYARAGHAVAN, A. Detecting high log-densities: An o($n^{1/4}$) approximation for densest k-subgraph. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing* (New York, NY, USA, 2010), STOC '10, ACM, pp. 201–210.

[11] CLIFFORD, P., AND SUDBURY, A. A model for spatial conflict. *Biometrika 60*, 3 (1973), 581–588.

[12] DOMINGOS, P., AND RICHARDSON, M. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (2001), ACM, pp. 57–66.

[13] EVEN-DAR, E., AND SHAPIRA, A. A note on maximizing the spread of influence in social networks. *Internet and Network Economics* (2007), 281–286.

[14] FEIGE, U., LOVÁSZ, L., AND TETALI, P. Approximating min sum set cover. *Algorithmica 40*, 4 (2004), 219–234.

[15] FELDMAN, J., HENZINGER, M., KORULA, N., MIRROKNI, V., AND STEIN, C. Online stochastic packing applied to display ad allocation. In *ESA* (2010).

[16] FELDMAN, J., KORULA, N., MIRROKNI, V., MUTHUKRISHNAN, S., AND PAL, M. Online ad assignment with free disposal. In *WINE* (2009).

[17] GOYAL, A., BONCHI, F., AND LAKSHMANAN, L. V. A data-based approach to social influence maximization.

Proceedings of the VLDB Endowment 5, 1 (2011), 73–84.

[18] HARTLINE, J., MIRROKNI, V., AND SUNDARARAJAN, M. Optimal marketing strategies over social networks. In *Proceedings of the 17th international conference on World Wide Web* (2008), ACM, pp. 189–198.

[19] HOLLEY, R. A., AND LIGGETT, T. M. Ergodic theorems for weakly interacting infinite systems and the voter model. *The annals of probability* (1975), 643–663.

[20] INTERACTIVE-ADVERTISING-BUREAU:. social advertising best practices. *http://www.iab.net/media/file/Social-Advertising-Best-Practices-0509.pdf* (2009).

[21] KEMPE, D., KLEINBERG, J., AND TARDOS, É. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), ACM, pp. 137–146.

[22] MIRROKNI, V. S., ROCH, S., AND SUNDARARAJAN, M. On fixed-price marketing for goods with positive network externalities. In *WINE* (2012), pp. 532–538.

[23] MOSSEL, E., AND ROCH, S. Submodularity of influence in social networks: From local to global. *SIAM J. Comput. 39*, 6 (2010), 2176–2188.

[24] TUCKER, C. Social advertising. *Available at SSRN: http://ssrn.com/abstract=1975897 or http://dx.doi.org/10.2139/ssrn.1975897* (2012).

[25] VEE, E., VASSILVITSKII, S., AND SHANMUGASUNDARAM, J. Optimal online assignment with forecasts. In *ACM EC* (2010).