Targeting Algorithms for Online Social Advertising Markets

Chaolun Xia  
Rutgers University  
xia.chaolun@gmail.com

Saikat Guha  
Microsoft Research, India  
saikat@microsoft.com

S. Muthukrishnan  
Rutgers & MS, India  
muthu@cs.rutgers.edu

Abstract—Advertisers in online social networks (OSNs) like Facebook and LinkedIn have some preferred set of users they wish to reach by showing their ads. OSNs offer fine-grained sets of user characteristics — including their career, wealth, education information, etc — that advertisers can specify for targeting their audience, and each of these characteristics requires different amounts of money for targeting.

The problem we address is what we call the targeting problem, that is, given a set $S_T$ of characteristics of interest to an advertiser (that is, the advertiser wishes to reach users who have these characteristics, i.e. $U(S_T)$) and a budget $b_0$, he wishes to spend, how to split the budget among the user characteristics so that they can reach the most number of users in $U(S_T)$?

- **OSN-perspective.** OSNs have complete knowledge of all the users and their characteristics. In this case, we propose a polynomial time algorithm for the targeting problem and prove that it is an $1−1/e$ approximation to the targeting that gets the optimal number of users. We define the marginal increment ratio and iteratively maximize it.

- **Advertiser-perspective.** No single advertiser has the mapping from users to their characteristics, and hence they cannot use the algorithm from above. We show through empirical analysis that the strategy of targeting subsets $U(S) \subseteq U(S_T)$ is their only feasible approach (in other words, targeting $U(S) \not\subseteq U(S_T)$ will be arbitrarily worse than the direct solution). For evaluation, we crawl and analyze more than one million suggested bids from Facebook and LinkedIn. Further, we propose a polynomial greedy algorithm for the targeting problem based on targeting subsets. In experiments, it increases the number of reached preferred users by nearly 40% over directly targeting $U(S_T)$, and for a moderate budget, it increases the number of reached preferred users by nearly 20%.

I. INTRODUCTION

Online advertising is one of the pillars in the Internet industry. In 2013, the online advertising markets generated 42.8 billion dollars in revenue in the US alone\(^1\). An online advertising market allows advertisers to pay for targeting (by showing ads to) specific audience through its targeting language. Google AdWords, the largest online ad network, for instance, allows advertisers to target users based on search terms from user input, the website (publisher) that the user is browsing, and simple user demographics (gender, age, location). The cost of reaching a user from a specific user set is set by auction mechanisms, e.g. second-price auction [6] or by contracts.

Other online advertising markets, specifically that run by Facebook, LinkedIn and other OSNs, offer much finer targeting controls. Their targeting languages contain detailed information shared directly by users, inferred from user daily activities [3] or purchased from third-part data providers. This includes detailed educational records about the user, past and present employment experience, significant life events like changes in marital status or birth of a baby. LinkedIn, for instance, allows advertisers to target a software engineer in Microsoft, or a college student whose major is nursing. Obviously, the price to advertise users varies with their characteristics [15]. For example, in a certain time period, the cost to target any software engineer in Microsoft could be twice high as the cost to target a nursing student.

This motivates a natural targeting problem. Each user $x$ has a set $C(x)$ of characteristics. An advertiser has set $S_T$ of characteristics of his interest and hence the set $U(S_T)$ of users (for any $x \in U(S_T)$, $S_T \subseteq C(x)$) is the advertisers’ preferred set of users to reach with ads. The advertiser has some budget $b_0$ and can split it reaching users with any combination of characteristics. What is the optimal strategy, that is, a way to split the budget so that the set $T$ of users reached has as much of $U(S_T)$ as possible, i.e., $|T \cap U(S_T)|$ is maximized.

- **OSN-perspective.** An OSN can take in an advertiser’s preferred set and solve his targeting problem. This is called proxy-bidding and OSNs (and other ad platforms like AdWords) provide such a service. In this case, the OSN knows the precise mapping from $x$ to $C(x)$ for any $x$. We present a polynomial algorithm that has $1−1/e$ approximation guarantee. In this algorithm, we first define the marginal increment ratio, and then iteratively allocate budget to the characteristic sets with the largest ratio.

- **Advertiser-perspective.** If an advertiser were to solve the targeting problem on his own with the price estimates provided by the OSNs, then the advertiser does not know $C(x)$ precisely for each $x$ and only knows the number of users, i.e. $|U(S)|$, with a given set $S$ of characteristics. In this case, the advertiser cannot infer the overlap between $U(S_1)$ and $U(S_{2})$ for two different characteristic sets $S_1$ and $S_2$. We focus on subset targeting, that is, given $S_T$ of interest to the advertiser, he splits budget between $S_i$’s such that $U(S_i) \subseteq U(S_T)$.

\(^1\)http://www.statista.com/statistics/275883/online-advertising-revenue-in-the-us-by-half-year/
and $U(S_i)$’s are pairwise disjoint. We propose a fast greedy algorithm using subsets and study its performance empirically using a unique dataset consisting of more than one million suggested bids from Facebook and LinkedIn that we crawl.

The rest of this paper is organized as follows. Section II introduces the background of OSN advertising and formulates the problem. In Section III, we describe the approximation algorithm for OSN-perspective. In Section IV, we first propose two heuristics and based on them we propose the greedy algorithm for Advertiser-perspective. We introduce the non-trivial work of crawling the datasets in Section V, and evaluate the greedy algorithm for Advertiser-perspective in Section VI. We summarize related works in Section VII, and conclude in Section VIII.

II. PRELIMINARY

In OSN advertising markets considered in this paper, e.g. Facebook and LinkedIn, an OSN user $x$ has one or more characteristics, and every characteristic contains exactly one attribute $a \in A$ and one value $v \in D(a)$ of the attribute. For example, if a user has the characteristic (Location:CA), it means that the OSN knows that the user lives (or works) in California. Let $A$ be the set of all attributes, e.g. $A = \{\text{Location}, \text{Age}, \ldots\}$, and $D(a)$ be the domain of the attribute $a$, e.g. $D(\text{Location}) = \{\text{CA, NY, NJ}, \ldots\}$. Let $S$ denote a set of characteristics and $U(S)$ denote the set of all the users who have all the characteristics in $S$. For example, if $c_1 = (\text{Location:CA})$ and $c_2 = (\text{Gender:Female})$ and $S_1 = \{c_1, c_2\}$, we say that $U(S_1)$ is the set of all the users who are female and live in CA. But any user in $U(S)$ could possibly have other characteristics, e.g. $(\text{Age:18-21})$. Let $A(S)$ denote the set of attributes involved in $S$, e.g. $A(S_1) = \{\text{Location, Gender}\}$.

When an advertiser wants to promote his campaign, we assume that he has a limited budget $b_0$ and a single preferred characteristic set $S_T$, which means that the advertiser only wants to target (by showing an ad) to users in $U(S_T)$. For example, a luxury car dealer in CA may only want to target rich people in CA. To be clear, if the characteristic set of a user is a superset of $S_T$, the user is equivalently preferred by the advertiser. But a user who lacks any characteristic in $S_T$ is not preferred. For example, a rich software engineer in CA, a rich banker in CA and a rich woman in CA are assumed to be equivalently preferred by this car dealer, but a software engineer in CA (without knowing s/he is rich) is not preferred by the dealer. For any $S$, the number of its users (i.e. $|U(S)|$) and the price $p(S)$ are public to advertisers. More specifically, $p(S)$ is the unit cost to target any user in $U(S)$.

It is trivial to see that if an advertiser allocates budget $b \leq |U(S)| \cdot p(S)$ to target the user set $U(S)$, the number of targeted users is $\frac{b}{p(S)}$. We assume that these $\frac{b}{p(S)}$ users are uniformly sampled from $U(S)$, i.e. each user $x \in U(S)$ has the equal probability $\frac{1}{|U(S)| \cdot p(S)}$ to be targeted. Now we formulate the targeting as a maximization problem in Eq (1): given the preferred characteristic set $S_T$ and the budget $b_0$ from the advertiser and the price function $p(S)$ for any $S$ from the OSN, we want to find an optimal allocation of budget, i.e. a vector $B = (B_1, \ldots, B_N)$ where $B_i$ is the budget allocated to target the user set that corresponds to the $i$-th characteristic set $U(S_i)$, to

$$\begin{align*}
\text{maximize} & \quad \sum_{x \in U(S_T)} \min\{1, f(x)\} \\
\text{subject to} & \quad 0 \leq B_i, \forall i \in \{1, \ldots, N\} \\
& \quad \sum_{i=1}^N B_i \leq b_0
\end{align*}$$

(1)

$f(x)$ is the expected (total) number of times that the user $x$ is targeted. In other words, if $x$ is preferred (i.e. $x \in U(S_T)$), we say that the advertiser has reached $f(x)$ unique preferred user in expectation (by ignoring others). We use $\min\{1, f(x)\}$ to formulate the constraint that even if any user is targeted more than once, we only count it as one targeted user. Thus, the expected total number of reached preferred users is $\sum_{x \in U(S_T)} \min\{1, f(x)\}$. According to the uniform assumption, $f(x)$ is defined in Eq (2) where $\mathbbm{1}_{x \in U(S_i)}$ is an indicator function, returning 1 if $x \in U(S_i)$ and 0 otherwise.

$$f(x) = \sum_{i=1}^N \mathbbm{1}_{x \in U(S_i)} \frac{B_i}{|U(S_i)| \cdot p(S_i)}$$

(2)

However, in many OSN markets, e.g. Facebook, LinkedIn and Twitter, advertisers are unable to evaluate the indicator function $\mathbbm{1}_{x \in U(S_i)}$ for any $x$ and $S_i$, because they are not allowed to know which specific users have which specific characteristics for privacy concerns. Thus, we consider two settings of the problem. For the first setting where $\mathbbm{1}_{x \in U(S_i)}$ is computable, we name it as the OSN-perspective setting since OSNs has the complete knowledge. For the second setting where $\mathbbm{1}_{x \in U(S_i)}$ is not computable, we name it as the Advertiser-perspective setting. For the OSN-perspective setting, we propose a polynomial algorithm with performance guarantee, and for the Advertiser-perspective setting, we propose a greedy algorithm based on reasonable heuristics.

III. OSN-PERSPECTIVE: AN APPROXIMATION ALGORITHM

To present the algorithm, we first formulate $R_i$ as the ratio of marginal increment (of the objective function in Eq (1)) to a small amount of budget allocated to the $i$-th characteristic set, i.e. $S_i$. We say that a user set $U(S)$ is full iff it is true that $\forall x \in U(S) \cap U(S_T)$, $f(x) \geq 1$. In other words, being full means that every user belonging to the intersection $U(S) \cap U(S_T)$ has already been targeted for at least once in expectation. On the other hand, if $U(S)$ is not full, there always exists at least one user $x$ such that $x \in U(S) \cap U(S_T)$ and $f(x) < 1$. Thus, for any $U(S_i)$ that is not full, we can always find a small $\epsilon_i \in (0, 1]$ such that $f(x) + \epsilon_i \leq 1, \forall x \in U(S_i) \cap U(S_T)$ and $f(x) < 1$. Let $h_i = \max\{f(x) \mid x \in U(S_i) \cap U(S_T), f(x) < 1\}$, then the largest value of $\epsilon_i$ is $1 - h_i$. It

2Assuming that we have an ordered indexing, from 1 to $N$, for every characteristic set, and we will analyze the size of $N$ in Section IV-B.
is straightforward to see that if the advertiser allocates $\epsilon_i \cdot |U(S_i)| \cdot p(S_i)$ dollars to target $U(S_i)$, the objective function becomes

$$R_i = \frac{\sum_{x \in U(S_i) \cap U(S_i')} f(x) < 1}{|U(S_i)| \cdot p(S_i)}$$

(3)

The essential idea of the algorithm is that for each iteration, we first identify the characteristic set $S_i$ with the largest marginal increment, i.e. $i^* = \arg \max_i R_i$, and then allocate budget to $U(S_{i^*})$ till the budget is exhausted or $U(S_{i^*})$ is full.

**Algorithm 1** Approximation Algorithm

**Input** $S_T$ and $b_0$ from the advertiser, pricing and mapping from the OSN.

**Output** $(B_1, ..., B_N)$.

1: $\forall x, f(x) = 0; \forall i \in [N], B_i = 0$
2: repeat
3: $i^* \leftarrow \arg \max_i R_i$
4: $\Delta b \leftarrow \min(b_0, |U(S_{i^*})| \cdot p(S_{i^*})(1 - h_{i^*}))$
5: $B_{i^*} \leftarrow B_{i^*} + \Delta b$
6: for $x \in U(S_{i^*}) \cap U(S_T)$ do
7: $f(x) \leftarrow f(x) + \frac{\Delta b}{|U(S_{i^*})| \cdot p(S_{i^*})}$
8: end for
9: until $b_0 = \sum_{i=1}^N B_i$ or $\forall x \in S_T, f(x) \geq 1$
10: return $B$

**Proposition 1.** The algorithm will stop after at most $m$ iterations. With a proper preprocessing, the running time for each iteration is $O(m + N)$. Thus the overall time complexity is $O(m^2 + mN)$ where $m = |U(S_T)|$ and $N$ is the total number of characteristic sets.

**Proof Sketch:** In each iteration, there is at least one user $x \in U(S_T)$ whose $f(x)$ is increased to 1 (if there is no such a user, it means that the budget runs out and the algorithm will stop immediately). Thus the algorithm will stop after at most $m$ iterations. We can do an $O(mN)$ preprocessing (only once before “repeat” in line 2) to build the mapping between all the users in $U(S_T)$ and all the characteristic sets. Moreover, we can maintain the values of $\sum_{x \in U(S_i)} f(x) < 1$ for all $S_i$ and update them after we update $f(x)$ between line 7 and 8, thus the running time of each iteration becomes $O(m + N)$.

We show that the greedy that maximizes the marginal value in Eq (3) yields $1 - 1/e$ guarantee in Theorem 2. The proof is in the appendix.

**Theorem 2.** Algorithm 1 is a $1 - 1/e$ approximation to the targeting problem in the OSN-perspective setting.

**IV. ADVERTISER-PERSPECTIVE: A PRACTICAL ALGORITHM**

Although the proposed algorithm in Section III has a desirable performance guarantee, it is not practical to individual advertisers due to the two challenges as follows.  

**Huge Search Space.** Algorithm 1 has a polynomial time complexity w.r.t. the number of all the characteristic sets, i.e. $N$, however, in the mainstream OSN advertising markets, $N$ could be very large since the advertiser can compose a characteristic set by arbitrarily choosing compatible characteristics. Even if we only allow all the characteristics in a characteristic set have different attributes (thus they are compatible to each other), there are up to $\prod_{a \in A} |D(a)| + 1$ distinct characteristic sets (in our crawled LinkedIn dataset, $N \approx 2.4 \times 10^{12}$)! This means that any practical algorithm cannot traverse all the characteristic set even for once.

**Incomplete Information.** In the Advertiser-perspective setting, advertisers do not know the value of the indicator function $\mathbb{I}_{x \in U(S_i)}$ for any $x$ and $S_i$. To our best knowledge, in this setting advertisers cannot even compute the objective function in Eq (1) in polynomial time for a given allocation of budget.

**A. Heuristics For Advertisers**

We propose two heuristics to address the two challenges respectively. For the huge search space, we proposed the heuristic of subset targeting and verify it through a dataset of suggested bids crawled from Facebook and LinkedIn. The detailed description of the dataset is in Section V. In short, we have 29,420 distinct characteristic sets from Facebook and 8,056 from LinkedIn. For each $S$, we know $p(S)$ and $|U(S)|$. To present the heuristics, we first define cheap characteristic set in Definition 1.

**Definition 1.** We say a characteristic set $S$ is cheap to $S_T$ iff $|U(S)| - p(S) < |U(S) \cap U(S_T)| - p(S_T)$.

It implies that $|U(S) \cap U(S_T)| > 0$ is a necessary condition for any $S$ to be cheap to $S_T$. If $S$ is cheap to $S_T$, according to the definition, it means that if the advertiser wants to target only $|U(S) \cap U(S_T)|$ preferred users, the cost to target $U(S)$ is less than the cost to directly target $U(S_T)$. Finding out cheap characteristic sets to $S_T$ is the essential idea to address the targeting problem. We present the first heuristic as follows.

**Heuristic 1 (Subset Targeting).** For any preferred characteristic set $S_T$, the algorithm only needs to consider allocating budget to a set $S$ of characteristic sets such that $U(S) \subseteq U(S_T), \forall S \in S$.

Instead of directly targeting $U(S_T)$, in general, the advertiser could consider three types of targeting strategies, namely superset targeting (i.e. targeting some $S$ such that $U(S) \supseteq U(S_T)$), subset targeting (i.e. $U(S) \subseteq U(S_T)$) and overlap targeting (i.e. $U(S) \cap U(S_T) \neq \emptyset$ but neither superset nor subset). It is trivial to see that targeting any $S$ such that $U(S) \cap U(S_T) = \emptyset$ cannot increase the objective function, thus the algorithm will never target disjoint characteristic sets. To be clear, when we mention subset targeting for $S_T$, it means that we find a characteristic set $S$ such that

4For example, the two characteristics (Location:CA) and (Age:18-21) are compatible to each other but (Age:18-21) and (Age:22-24) are not because any user has only one number for her age. Whether any two characteristics are compatible is decided by the OSNs. It is mostly likely safe to say that if any two characteristics with different attributes, they are compatible.
\( U(S) \subseteq U(S_T) \) other than \( S \subseteq S_T \), similarly for superset and overlap targeting.

Hypothesis 1 would be reasonable if for any \( S_T \) and \( S \), when \( U(S) \) is the superset of \( U(S_T) \) or is partially overlapped with \( U(S_T) \), \( S \) has very low probability to be cheap to \( S_T \). If it is true, we lose only a few cheap characteristic sets by ignoring superset and overlap targeting. To verify this, we test the following three hypotheses through data analysis. Note that, for each hypothesis we test it over all the data snapshots and report the average results.

**Hypothesis 1** (Infeasibility of Superset Targeting). For any \( S \) and \( S_T \) such that \( U(S) \supseteq U(S_T) \), it is true that \( p(S_T) \leq p(S) \frac{|U(S)|}{|U(S_T)|} \).

The superset targeting strategy is that, instead of directly targeting \( U(S_T) \) with the unit cost \( p(S_T) \), the advertiser targets a superset \( U(S) \supseteq U(S_T) \). For example, instead of targeting any software engineer in Microsoft, we target any employee in Microsoft. One necessary condition for this strategy to be feasible is that we can reject Hypothesis 1. By examining all the 175392 pairs of characteristic sets (one’s user set is the superset of the other’s user set) from Facebook, there is only 1.3 (\(< 0.01\%\) pair violating the hypothesis. Similarly in LinkedIn, only 17 (\(\approx 0.07\%) \) out of the 24420 pairs violate the hypothesis. These observations support that if \( U(S) \supseteq U(S_T) \), \( S \) is unlikely to be cheap to \( S_T \) for any \( S \) and \( S_T \). Thus the hypothesis is verified.

**Hypothesis 2** (Infeasibility of Overlap Targeting). For any \( S \) and \( S_T \) such that \( U(S_T) \not\subseteq U(S) \), \( U(S) \not\subseteq U(S_T) \) and \( U(S_T) \cap U(S) \not= \emptyset \), it is true that \( p(S_T) \leq p(S) \frac{|U(S)|}{|U(S_T) \cap U(S)|} \).

With this strategy, for instance, instead of targeting software engineers in Microsoft, we target male employees in Microsoft. One necessary condition for this strategy to be feasible is that we can reject Hypothesis 2. However, as we verify all the 326758 pairs of characteristic sets (one’s user set is partially overlapped with the other’s user set) from Facebook, there are only 56 (\(\approx 0.02\%) \) pairs violating the hypothesis. Similarly, out of all the 16112 pairs of characteristic sets from LinkedIn, there are only 15 (\(\approx 0.09\%) \) violating pairs. These observations confirm Hypothesis 2, thus the overlap targeting is infeasible.

**Hypothesis 3** (Infeasibility of Subset Targeting). For any \( S \) and \( S_T \) such that \( U(S) \subseteq U(S_T) \), it is true that \( p(S_T) \leq p(S) \).

With this strategy, the advertiser can alternatively target a subset of \( U(S_T) \). For example, instead of targeting any software engineer in Microsoft, the advertiser only targets entry-level software engineers in Microsoft. One necessary condition for this strategy to be feasible is that we can reject Hypothesis 3. Surprisingly, we find that among all the 175392 pairs (one’s user set is the subset of the other’s user set) of characteristic sets from Facebook, there are 93165 (\(\approx 53.1\%) \) pairs violating the hypothesis. Similarly, out of all the 24420 pairs from LinkedIn, we find 9430 (\(\approx 38.6\%) \) violating pairs. These observations show that, both in Facebook and LinkedIn, subset targeting are potentially feasible strategies to solve the targeting problem.

Through hypotheses testing, we show that for more than 99.9% cases, there is no cheap characteristic set for superset or overlap targeting. Thus we conclude that the Heuristic 1 is reasonable. Note that, any combination of the three targeting strategies still belongs to one of them. For example, assuming \( U(S) \) is a subset of \( U(S') \) and \( U(S') \) is partially overlapped with \( U(S_T) \), it is easy to see that \( U(S) \) must be either a subset or an overlapped set of \( U(S_T) \). This means that although we only verify the three “simple” strategies, the verification indeed covers all possible “paths”, i.e. any combination of the three strategies.

Next, the critical reason for the second challenge is that a user with multiple characteristics appear in multiple (say \( k \)) user sets. If an allocation targets all the \( k \) user sets, we do not find any polynomial method (w.r.t. \( N \)) to compute \( f(x) \) because we do not know the indicator function \( 1_{x \in U(S)} \) for any \( x \) and \( S \). However, if an allocation only targets \( k \) pairwise-disjoint subsets of \( U(S_T) \), this would not be a problem because once we allocate budget \( B_t \) to \( U(S) \), the value of objective function will be increased by \( |U(S_T) \cap U(S)| \cdot \min \{1, \frac{B_t}{p(S_T)} \} \). Thus, we propose Heuristic 2.

**Heuristic 2** (Disjoint Targeting). We only consider to allocate budget to a set of characteristic sets whose corresponding user sets are pairwise disjoint.

**Remark 3.** Applying Heuristic 1 and 2 together, we consider to allocate budget to only a set \( S \) of characteristic sets such that:

- \( \forall S \in S, \ U(S) \subseteq U(S_T) \).
- \( \forall S \neq S' \in S, U(S) \cap U(S') = \emptyset \).

**B. Greedy Algorithm**

Based on the two heuristics, we propose a top-down greedy algorithm presented in 3 subroutines in Procedure 2, 3 and 4. Given \( S_T \) and \( b_0 \), by calling \( \text{FindSubsets}(S_T, b_0) \) it outputs the budget allocation in sparse representation. The result is a set \( L \) of pairs, i.e. \( L = \{(S,b)|b > 0\} \). Each pair \( (S,b) \) stands for the algorithmic decision that the advertiser should allocate \( b \) dollars to target \( U(S) \), and the advertiser is expected to reach unique \( \frac{b}{p(S)} \) users in \( U(S_T) \). By following the allocation \( L \), the advertiser is expected to target \( \sum_{(S,b) \in L} \frac{b}{p(S)} \) users in \( U(S_T) \).

Starting with the initial preferred characteristic set \( S_T \) and initial budget \( b_0 \), the algorithm iteratively selects and adds the locally optimal characteristic to the current preferred characteristic set (denoted by \( S^{(t)} \); note that \( t \), the iteration number, starts from \( 0 \) and \( S^{(0)} = S_T \)) to construct a new preferred characteristic set \( S^{(t+1)} \). It is easy to see that \( U(S^{(t+1)}) \subseteq U(S^{(t)}) \subseteq \ldots \subseteq U(S_T) \) since the algorithm only adds characteristics. If the remaining budget \( b^{(t)} \) (note that \( b^{(0)} = b_0 \)) is not enough to target all the users in \( U(S^{(t+1)}) \), then we recursively call the algorithm to solve another targeting problem with \( S^{(t+1)} \) as the initial preferred characteristic and \( b^{(t)} \) as the total budget. Otherwise, by calling
Procedure 4, the algorithm allocates \( p(S(t+1)) \cdot |U(S(t+1))| \) budget to fully target \( U(S(t+1)) \), and after this, the algorithm will select and add another characteristic to \( S(t) \) to form a new current preferred characteristic set \( S'(t+1) \), and repeats this process until all the users in \( U(S_T) \) are targeted or the budget is exhausted.

**Algorithm 2** Find Subsets

1: procedure FINDSUBSETS(S, b)
2: \( n \leftarrow \min \{ |U(S)|, \frac{|p|}{b} \} \)
3: \( a^* \leftarrow \text{argmax} \text{EVALUATEATTRIBUTE}(S, b, a) \) \( \forall a \in A \setminus \{b\} \)
4: if EVALUATEATTRIBUTE(S, b, a*) > n then
5: return ALLOCATEBUDGET(S, b, a*)
6: else
7: return \( \{(S, b)\} \)
8: end if
9: end procedure

**Algorithm 3** Evaluate Attribute

1: procedure EVALUATEATTRIBUTE(S, b, a)
2: \( D' \leftarrow D(a) \), \( n \leftarrow 0 \), \( V \leftarrow \emptyset \)
3: while \( b > 0 \) and \( D' \neq \emptyset \) do
4: \( u^* \leftarrow \text{argmin} p(S \cup \{(a : v)\}) \) \( \forall v \in V \)
5: \( D' \leftarrow D' - \{u^*\} \)
6: if \( \forall v \in V, u((\{a : v^*\}) \cap u((\{a : v\})) = 0 \) then
7: \( S^* \leftarrow S \cup (\{a : v^*\}) \)
8: \( n \leftarrow \min \{ |U(S^*)|, \frac{|p|}{b} \} \)
9: \( n \leftarrow n + n \)
10: \( b \leftarrow b - p(S^*) \cdot n \)
11: \( V \leftarrow V \cup \{v^*\} \)
12: end if
13: end while
14: return \( n \)
15: end procedure

**Algorithm 4** Allocate Budget

1: procedure ALLOCATEBUDGET(S, b, a)
2: \( D' \leftarrow D(a) \), \( n \leftarrow 0 \), \( V \leftarrow \emptyset \)
3: while \( b > 0 \) and \( D' \neq \emptyset \) do
4: \( u^* \leftarrow \text{argmin} p(S \cup (\{a : v\})) \) \( \forall v \in V \)
5: \( D' \leftarrow D' - \{u^*\} \)
6: if \( \forall v \in V, u((\{a : v^*\}) \cap u((\{a : v\})) = 0 \) then
7: \( S^* \leftarrow S \cup (\{a : v^*\}) \)
8: if \( b \geq |U(S^*)| \cdot p(S^*) \) then
9: \( R \leftarrow R \cup \{(S^*, |U(S^*)| \cdot p(S^*)) \}
10: \( b \leftarrow b - (|U(S^*)| \cdot p(S^*)) \)
11: end if
12: end if
13: end if
14: \( V \leftarrow V + \{v^*\} \)
15: end while
16: return \( R \)
17: end procedure

**Proposition 3.** The time complexity of the algorithm is \( O(|A|^2 + |A| \cdot \sum_{a \in A} D(a)^2 \cdot |b|) \).

Proof Sketch: for any input pair \((S_T, b_0)\), Procedure 2 and 4 will be called at most \(|A| \) times each in total, and Procedure 3 will be called at most \(|A|^2 \) times. The amortized running time of the non-recursive operations in Procedure 2, 3 and 4 are \( O(|A|) \), \( O\left( \frac{1}{|A|} \cdot \sum_{a \in A} D(a)^2 \right) \) and \( O\left( \frac{1}{|A|} \cdot \sum_{a \in A} D(a)^2 \right) \) respectively.

Note that, the time complexity is significantly lower than that of Algorithm 1 because this algorithm does not enumerate all the characteristic sets as Algorithm 1 does, instead, it goes through all the characteristics. Incorporating data-driven heuristics, it is not expected to have performance guarantee. Thus we evaluate its effectiveness through experiments in Section VI.

**V. PRICE DATA ACQUISITION**

In this section, we introduce the dataset we use to test the hypotheses in Section IV-A and evaluate the algorithm in Section IV-B, and the method how we crawl it. We need real price data from OSN advertising markets. However, to our best knowledge, there is no such large dataset available yet.

Fortunately, in order to guide advertisers who face a variety of targeting characteristic sets, both Facebook and LinkedIn advertising markets provide Suggested Bid which is a function that, for each characteristic set \( S \), shows the suggested bid to win an action to show an ad to one user in \( U(S) \) and the number of users in \( U(S) \), i.e. \( |U(S)| \). Moreover, through reverse-engineering, the recent work in [15] finds that in Facebook, the suggested bids are sampled from recent winning bids. This means that the suggested bid of a characteristic set \( S \) is the best estimate, that we are able to get from real OSN advertising markets so far, of the cost, i.e. \( p(S) \), of showing an ad to one user in \( U(S) \). Therefore, for any \( S \), we use the suggested bid \( p(S) \) of \( S \) to estimate \( p(S) \) for all the experiments in this paper. Although there is no literature about how LinkedIn generates suggested bids, we also include them for experiments.

**A. Suggested Bid**

We first introduce suggested bids. In the advertising systems of Facebook and LinkedIn, once an advertiser creates an ad with a characteristic set \( S \) at time \( t \), a suggested bid \( \hat{p}(S) \) and the number of users in \( U(S) \) are provided. It can be formulated as the function \( SB \) in Eq. (4).

\[
SB : (S, t) \rightarrow (\hat{p}(S), |U(S)|)
\]

**Verification.** Besides \( S \) and \( t \), there are many other factors, for example, ad content, advertising history of the advertiser, social activities\(^5\) of the advertiser, profiles of the advertiser, and total amount of budget, that we speculate that they might influence the suggested bid \( \hat{p}(S) \). To test existence of their influence, for each factor we conduct a simple A/B testing. Due to the limited space, we omit the detail and summarize the testing results in Table I. In LinkedIn, if the advertiser’s profile shows that he is not in the US, \( \hat{p}(S) \) will be slightly different. If an advertiser has advertising records, Facebook might slightly adjust the suggested bid shown to him. Since we want to use \( \hat{p}(S) \) to approximate \( p(S) \) as close as possible, when we crawl suggested bids, we try to fix all other factors as follows.

\(^5\)The advertiser account of Facebook (or LinkedIn) is based on the normal Facebook (or LinkedIn) account.
Take A Snapshot. We follow the method in [13] to take a snapshot of suggested bids. Since suggested bids (and true costs) in Facebook are sensitive [15] to time, for a large number of characteristic sets, we have to crawl their suggested bids simultaneously.

New Advertiser Accounts. We follow the setting in [20]. For crawling, we create and use new advertiser accounts without any advertising history or social interaction. Each account’s location is set to US. We choose Feed Ads (text ads shown in Timeline) which is a common ad type in both Facebook and LinkedIn.

We create two crawlers interacting with Facebook [1] and LinkedIn [2] advertising systems respectively. Each interaction consists of two steps. First, a crawler logs in an advertiser account, composes a characteristic set S and sends it to the OSN. The OSN returns the query result \( \hat{p}(S), |U(S)| \) to the crawler. Since the crawlers do not create or run real campaigns, the advertising history of accounts remains empty during the entire crawling period. Besides, the crawling process neither spends money nor participates real auctions in the markets.

B. Crawling Facebook Suggested Bids

We first manually select 6 common targeting attributes and their 87 frequent values listed in Table II from Facebook advertising system. Thus we have 87 characteristics. We enumerate all the characteristic sets with up to 3 distinct characteristics. As a result, we get 29420 characteristic sets, among which there are 1, 87, 2311 and 27021 characteristic sets with exactly 0, 1, 2 and 3 characteristics, respectively. The crawler simultaneously issues the 29420 queries, each of which is for a characteristic set, as a snapshot. We take one snapshot per day from July 2015 to Aug 2015. As a result, we have 35 such snapshots, i.e. 1029700 data in total. We test hypotheses or evaluate algorithms in Sec IV on each snapshot and report the average results over all the 35 snapshots. We use Facebook Ads APIs\(^7\) to build the crawler.

\(^{6}\)For any characteristic set, if we do not specify the location, it is the US by default. Thus, the empty characteristic set corresponds to all the users in the US. It is similar for crawling LinkedIn data.

\(^{7}\)https://developers.facebook.com/docs/graph-api

C. Crawling LinkedIn Bid Suggestion

Similarly for LinkedIn, we consider 8 common attributes and 449 of their frequent values in Table III, thus we have 449 characteristics. We enumerate all the characteristic sets with up to 2 distinct characteristics. As a result, we have 8311 characteristic sets, among which there are 1, 449 and 8056 characteristic sets with exactly 0, 1 and 2 characteristics, respectively. The crawler simultaneously issues the 8311 queries as a snapshot. We harvest 4 snapshots, i.e. 33244 data in total from LinkedIn. Since LinkedIn does not release open APIs for suggested bids, we use Python and Selenium (a web automation package) to mimic real advertisers to harvest data from LinkedIn advertising interface. Fig 1 is an example of the web page showing the suggested bid as 4.58$ and the number of qualifying users as 140950 for the characteristic set \{Location:CA,(Skill:C++)\}.

![Example of the suggested bids provided by LinkedIn.](image)

### VI. Experiments

In this section, we evaluate the greedy algorithm proposed for Advertiser-perspective setting in Sec IV-B with both Facebook and LinkedIn datasets that are introduced respectively in Sec V-B and V-C. In short, we have 4 snapshots of suggested bids from LinkedIn and 35 snapshots from Facebook, i.e. more than 1 million data in total. Since the algorithm follows subset targeting, we do not consider any characteristic set already with 2 characteristics in LinkedIn or with 3 characteristics in Facebook as \(S_T\) (because their user sets have no subset in the crawled data). We also do not consider any characteristic set whose user set size is too small. For each of the remaining characteristic sets, we consider it as \(S_T\) in turn, i.e. the input for Procedure 2 (together with a budget). For each snapshot, we calculate the average results over all \(S_T\), and then report the final average results over all snapshots.
A. Budget Variation

We first evaluate the algorithm for different budgets. We define the measurement budget-increment curve as follows. The baseline algorithm that we consider is to target $U(S_T)$ directly, which means the baseline can reach $\frac{b_0}{p(S_T)}$ preferred users with the budget $b_0$. A single evaluation point $(x, y)$ on the budget-increment curve, shown in Fig 2 and 3, denotes that if the advertiser is given the budget enough to exactly target $x \in (0, 1)$ proportion of preferred users by the baseline, our greedy algorithm can increase the number of reached preferred users by $y$ proportion, i.e. reach $x(1+y)$ proportion. In Facebook, we compute the curve for each of the 2399 characteristic sets (as $S_T$). The overall budget-increment curve is presented by the blue solid line in Fig 2. Similarly, we plot the overall budget-increment curve for 253 characteristic sets from LinkedIn with the red solid line. The green dashed line is for the baseline.

We have the following observations. First, the less budget the advertiser has, the more effective the algorithm is. As shown in Fig 2, when the advertiser has very limited budget, e.g. the budget is enough to target only a small portion $x \in (0, 0.05]$ of the preferred users, by our algorithm, he is able to target 40.8% more users on Facebook and 37.6% more on LinkedIn. When the advertiser has enough budget to target 20% of the total preferred users by the baseline, he is still able to target 24.9% more users in Facebook and 19.3% more in LinkedIn. Second, when the budget is approaching to monopoly the entire user set, the effectiveness of the algorithm is closed to 0, but it is guaranteed that the advertiser can target at least as many users as the baseline does. The reason that the effectiveness appears to be decreasing is that the user supplies of cheap characteristic sets are limited, and as the budget gradually increases, the user supplies of those cheap characteristic sets gradually run out. Then the algorithm has to allocate remaining budget to characteristic sets that are not cheap to $S_T$. Third, although Facebook and LinkedIn run different advertising markets, we find that our algorithm produces budget-increment curves in a similar shape, which shows its generalizability.

B. Price Variation

We then evaluate the algorithm’s effectiveness when the cost of $S_T$, i.e. $p(S_T)$, varies. For LinkedIn, we create 5 price intervals, namely $(0, 4.00], (4.00, 5.00], (5.00, 6.00], (6.00, 7.00]$ and $(7.00, +\infty]$ (all prices are in the US dollar). Then we map all the 253 characteristic sets into one of the 5 price intervals based on their prices. As a result, the 5 intervals get 5.4%, 27.2%, 28.5%, 17.5% and 21.4% of the total characteristic sets respectively. For each price interval, we compute its individual budget-increment curve for each characteristic set assigned to this interval, and then average them to get the interval-average curve. We process 2399 characteristic sets from Facebook in the similar way. The results are shown in Fig 3(a) and 3(b). We find that no matter how large $p(S_T)$ is, the proposed algorithm consistently outperforms the baseline. As shown in Fig 3(b), the area under each budget-increment curve decreases as the interval price decreases. This indicates that the larger $p(S_T)$ is, the larger effectiveness the algorithm has, both on LinkedIn and Facebook.

![Fig. 2: Increment with different amount of budget.](image)

![Fig. 3: Performance with preferred characteristic sets in different prices.](image)

VII. Related Work

The knowledge of the price distributions can help advertisers make better decisions. If prices are not known, advertisers can learn them with potential penalty during exploration [11]. Instead we focus on the case when the prices are known to advertisers. In sponsored search markets, budget optimization (BO) is a well studied problem in [18][10][9][17]. When given the landscapes of keywords, as pointed out by [4][21], the degenerated BO problem is an instance of the Multiple-Choice Knapsack problem [19]. [10] proves that a simple random strategy has the $1 - 1/e$ guarantee.

However, instead of maximizing the profit, our problem has new challenges when the advertisers want to target a specific user segment in OSNs and to maximize the number of reached unique preferred users. [8] proposes algorithms to identify a set of alternative topics [7] that have (approximately) the same audience on a micro-blogging platforms for a targeting topic. We point out that their algorithm is not applicable in our setting for the two following reasons. First, they assume that advertisers have to purchase all the users in a set which is not true in practice. Second, they only consider the setting where the complete user information is available (similar to the OSN-perspective setting in this paper).

Although our problem is related to the Allocation problem [16][12][5], they are essentially different, since our problem aims to benefit individual advertisers rather than the market owners.
VIII. CONCLUSIONS

We study the targeting problem for advertisers. For the OSN’s perspective, we present a polynomial time algorithm and prove that it has $1 - 1/e$ guarantee. For the advertiser’s perspective, we show through data analysis that the strategy of targeting subsets of audience sets is viable and propose a greedy algorithm based on subset targeting. For evaluation, we crawl a large unique dataset which contains more than one million suggested bids from Facebook and LinkedIn, and we show that the proposed algorithm makes advertisers reach more target audience than directly targeting the users.

Our work suggests that smart advertisers can utilize the “arbitrage” to target more users for the given budget, but we do not know how these strategies influence the revenue of OSNs. Therefore, it will be of interest to study pricing mechanisms for OSNs that prevent any potential arbitrage available for advertisers. Moreover, with the large dataset of suggested bids crawled from Facebook and LinkedIn, an open problem is to predict the prices of various characteristic sets.

REFERENCES

[5] E. Even Dar, V. S. Mirrokni, S. Muthukrishnan, Y. Mansour, and A. Moss. The budgeted maximum coverage problem) to prove Theorem 2. W.l.o.g. we assume that by using Algorithm 1, $b_0$ is exhausted after exactly $T > 0$ iterations. Let $B^t$ denote the budget allocation vector after $t$-th iteration and the allocated budget is denoted by $|B^t| = \sum_{i=1}^{N} B_i^t$. Let $e_i$ denote the budget allocated in the $t$-th iteration, i.e. $e_i = |B_i^t| - |B_i^{t-1}|$: $g(B)$ is the value of objective function in Eq (1) given the allocation $B$. It is easy to see that $B^0 = 0$ and $g(B^0) = 0$. We also assume that there exists an optimal allocation $B^*$ (not necessarily unique) which maximizes the objective function (note that the number of iterations to compute the optimal allocation is not necessarily $T$). Let $j(t) \in [N]$ denote the index of the characteristic set chosen by the algorithm at the $t$-th iteration (i.e. $S_{j(t)}$).

**Lemma 4.** After each iteration $t \in [T]$, the following holds:

$$g(B_t) \geq \frac{e_t}{b_0}g(B^*) + (1 - \frac{e_t}{b_0})g(B^{t-1})$$

**Proof.** Let $\hat{B}^{t-1}$ denote an allocation vector such that $\hat{B}^{t-1} = (\max\{B_i^{t-1}, B_i^t\})_{i=1}^{N}$. It is trivial to see that $|B^t| - |B^{t-1}| \leq b_0$. For any $i$ such that $B_i^{t-1} < B_i^t$, we can find $R_i \leq R_i(t)$ since the marginal increment ratio $R_i(t)$ chosen by Algorithm 1 is the largest at iteration $t$. Thus we have $g(B^{t-1}) - g(B^{t-1}) \leq b_0 R_i(t)$. Noting that $g(B^*) \leq g(\hat{B}^{t-1})$ and $R_i(t) = \frac{g(B^*) - g(B^{t-1})}{e_i}$, we prove the lemma.

**Lemma 5.** After each iteration $t \in [T]$, the following holds:

$$g(B_t) \geq \left(1 - \prod_{j=1}^{t} \left(1 - \frac{c_j}{b_0}\right) \right) g(B^*)$$

**Proof.** We prove the lemma by induction on the number of iterations in which the allocation $B_t$, $t = 1, ..., T$ are considered. For $t = 1$, it is true by directly applying Lemma 4. Suppose the statement of the lemma holds for iterations from 1 to $t - 1$, we show that it is also holds for the $t$-th iteration.

$$g(B_t) \geq \frac{c_t}{b_0}g(B^*) + \left(1 - \frac{c_t}{b_0} \right) \left(1 - \prod_{j=1}^{t-1} \left(1 - \frac{c_j}{b_0}\right) \right) g(B^*)$$

$$= \left(1 - \prod_{j=1}^{t} \left(1 - \frac{c_j}{b_0}\right) \right) g(B^*)$$

Note that $\prod_{i=1}^{T} \left(1 - \frac{c_i}{b_0}\right) \leq (1 - \frac{1}{T})^T < \frac{1}{e}$, thus we have $g(B^T) \geq (1 - \frac{1}{e})g(B^*)$, proving the theorem.

APPENDIX

Here we extend the techniques in [14] (for budgeted maximum coverage problem) to prove Theorem 2. W.l.o.g. we assume that by using Algorithm 1, $b_0$ is exhausted after exactly $T > 0$ iterations. Let $B^t$ denote the budget allocation vector after $t$-th iteration and the allocated budget is denoted by $|B^t| = \sum_{i=1}^{N} B_i^t$. Let $e_i$ denote the budget allocated in the $t$-th iteration, i.e. $e_i = |B_i^t| - |B_i^{t-1}|$: $g(B)$ is the value of objective function in Eq (1) given the allocation $B$. It is easy to see that $B^0 = 0$ and $g(B^0) = 0$. We also assume that there exists an optimal allocation $B^*$ (not necessarily unique) which maximizes the objective function (note that the number of iterations to compute the optimal allocation is not necessarily $T$). Let $j(t) \in [N]$ denote the index of the characteristic set chosen by the algorithm at the $t$-th iteration (i.e. $S_{j(t)}$).

**Lemma 4.** After each iteration $t \in [T]$, the following holds:

$$g(B_t) \geq \frac{e_t}{b_0}g(B^*) + (1 - \frac{e_t}{b_0})g(B^{t-1})$$

**Proof.** Let $\hat{B}^{t-1}$ denote an allocation vector such that $\hat{B}^{t-1} = (\max\{B_i^{t-1}, B_i^t\})_{i=1}^{N}$. It is trivial to see that $|B^t| - |B^{t-1}| \leq b_0$. For any $i$ such that $B_i^{t-1} < B_i^t$, we can find $R_i \leq R_i(t)$ since the marginal increment ratio $R_i(t)$ chosen by Algorithm 1 is the largest at iteration $t$. Thus we have $g(B^{t-1}) - g(B^{t-1}) \leq b_0 R_i(t)$. Noting that $g(B^*) \leq g(\hat{B}^{t-1})$ and $R_i(t) = \frac{g(B^*) - g(B^{t-1})}{e_i}$, we prove the lemma.

**Lemma 5.** After each iteration $t \in [T]$, the following holds:

$$g(B_t) \geq \left(1 - \prod_{j=1}^{t} \left(1 - \frac{c_j}{b_0}\right) \right) g(B^*)$$

**Proof.** We prove the lemma by induction on the number of iterations in which the allocation $B_t$, $t = 1, ..., T$ are considered. For $t = 1$, it is true by directly applying Lemma 4. Suppose the statement of the lemma holds for iterations from 1 to $t - 1$, we show that it is also holds for the $t$-th iteration.

$$g(B_t) \geq \frac{c_t}{b_0}g(B^*) + \left(1 - \frac{c_t}{b_0} \right) \left(1 - \prod_{j=1}^{t-1} \left(1 - \frac{c_j}{b_0}\right) \right) g(B^*)$$

$$= \left(1 - \prod_{j=1}^{t} \left(1 - \frac{c_j}{b_0}\right) \right) g(B^*)$$

Note that $\prod_{i=1}^{T} \left(1 - \frac{c_i}{b_0}\right) \leq (1 - \frac{1}{T})^T < \frac{1}{e}$, thus we have $g(B^T) \geq (1 - \frac{1}{e})g(B^*)$, proving the theorem.