Optimizing Peer Referrals for Public Awareness using Contextual Bandits

Ramaravind Kommiya Mothilal
Microsoft Research India
Bangalore, India
t-rakom@microsoft.com

Amulya Yadav*
University of Southern California
Los Angeles, USA
amulyaya@usc.edu

Amit Sharma
Microsoft Research India
Bangalore, India
amshar@microsoft.com

ABSTRACT

Programs that reward people for referring their friends are increasingly being used to raise awareness about important topics. With a fixed budget for referral incentives, a natural goal for such referral programs is to maximize the number of people reached. Unlike a typical influence maximization problem, however, the social network of potential adopters is unknown apriori. Further, people's response to a referral incentive can depend on various factors such as their preference for the content, size of their social network, and their estimated value for sharing. Therefore, we introduce an incentive-aware variant of the influence maximization problem and formalize it under an online learning setting. Given the lack of initial information about the social network or how people respond to referral incentives, we use an explore-exploit strategy and present a contextual bandit agent *CoBBI* that optimizes the incentives for each user by learning from the results of its past actions. We demonstrate the effectiveness of CoBBI on data from a real-world referral program for raising land rights' awareness among farmers. Compared to a wide range of baselines, we find that CoBBI is consistently more cost-effective, across a wide range of influence probabilities and people's response to incentives.

CCS CONCEPTS

• **Human-centered computing** → **Social content sharing**; • **Information systems** → *Incentive schemes*; • **Computing methodologies** → *Online learning settings.*

KEYWORDS

peer-to-peer referrals, social influence maximization, contextual bandit

ACM Reference Format:
Ramaravind Kommiya Mothilal, Amulya Yadav, and Amit Sharma. 2019. Optimizing Peer Referrals for Public Awareness using Contextual Bandits. In *ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS) (COMPASS '19), July 3–5, 2019, Accra, Ghana.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3314344.3332497

1 INTRODUCTION

Raising awareness about a topic or an intervention is important for almost all global health and public welfare projects. In some projects, the goal is to promote a beneficial practice, such as encouraging regular washing of hands to prevent disease [1]. For others, the goal is to draw attention to an available intervention, such as a vaccination or campaign that people may benefit from [15]. While not sufficient for behavioral change, delivering correct and timely information to people is a necessary first step. However, reaching out to people can be expensive and time-intensive especially when the target recipients are in low-resource or remote communities. During the Ebola outbreak in Liberia, for example, the per-person expenditure to spread awareness about Ebola was more than the daily wage of most people in the country [9, 23].

Motivated by the importance of spreading awareness and the relative inefficiency of doing it through manual outreach, phone-based solutions have been proposed that can deliver the desired information to people instantly and often at low cost. To alleviate the need for knowing every person's phone number beforehand, many of these solutions have a peer-to-peer referral component that allows a message to be spread from a small set of initial people to many hundreds. For example, one solution is "Learn2Earn" [30] that awards mobile phone talktime to any person who listens to the message and answers questions correctly, and subsequently a referral bonus talktime for every person whom they refer to do the same. As in word-of-mouth marketing for commercial products [5], peer referrals can enable a wide reach. In a deployment in India, the Learn2Earn referral system spread

*This author's work was done during an internship at Microsoft Research.

COMPASS '19, July 3–5, 2019, Accra, Ghana

ACM ISBN 978-1-4503-6714-1/19/07…$15.00
https://doi.org/10.1145/3314344.3332497

awareness about land rights to 17000 farmers in 45 days, starting from an initial set of just a few hundred farmers [30].

To repeat the success of such peer-based awareness campaigns, a natural question is how to set incentives for peer referral. Since awareness programs typically have a limited budget for referral incentives, we consider the following question: how to design incentives to maximize reach of a particular message given a fixed budget? An intuitive answer is to reward each referral with a fixed incentive, as in the Learn2Earn system. However, all referrals are not equal. Referring one's first friend is different from referring one's *n*th friend, and referring a friend in a close-knit community is different from referring someone from a hard-to-reach community. These differences indicate the value of customizing referral incentives based on context.

Deciding a custom incentive, however, is difficult in public awareness campaigns for two main reasons. First, unlike much of research in computer science on social influence maximization [4, 14, 31], the underlying social network of people is unknown. Second, unlike work in economics on social referrals, people's response to referrals is typically unknown apriori.

In this paper, we address these practical constraints in referral programs by introducing the Incentive-aware Influence Maximization problem. This formulation assumes that different people have different *response functions* to incentives, and that the social network of referrers is unknown apriori, rather it will be incrementally discovered as referrals are made. We decide referral incentives using an explore-exploit paradigm where our proposed algorithm learns an optimal incentive payout for a referral based on the results of its past actions. That is, our algorithm adapts to people's response functions by offering initial incentives randomly and then awarding future incentives based on how people responded to past incentives. Specifically we present a contextual bandit-based [2, 6, 19] algorithm where past history of each referrer and the state of the discovered social network is modeled as context. Given the current context, the algorithm outputs whether to offer an incentive to a potential referrer. We call our proposed solution CoBBI (**Co**ntextual **B**andit **B**ased **I**ncentives).

We evaluate CoBBI using data from a deployment of the Learn2Earn system. We compare it to six baseline policies, ranging from simple policies that assign fixed payments to naive versions of an explore-exploit policy. First, we show that simple strategies to customize incentives work for some scenarios, but fail when we consider different ways in which people can respond to incentives. Since CoBBI learns people's responses by exploring, it adapts well to different response functions and influence probabilities. Second, CoBBI is cost-effective: among the policies that reach the maximum number of people, it has the highest ratio of number of people influenced to money spent on referrals. Finally, we show how one can tune the tradeoff between reach and cost in CoBBI by changing its bandit reward function.

2 RELATED WORK

Peer-to-peer referral programs are an important mechanism for spreading awareness and driving product adoption [22, 28]. For example, companies like Dropbox, Uber, and Verizon have used such referrals to grow the adoption of their services. Dropbox, for instance, offers an additional 500 MB of storage for every friend a user refers [21]. Referral programs are also useful for non-profit organizations in public welfare, e.g., to enable timely dissemination of critical information to relevant communities [11, 15, 37]. When a person refers someone, we say that the person *influences* them and call the overall process the *influence process* or simply *diffusion*.

A common feature of such referral programs is that people are incentivized, often monetarily, for each referral that they make. Given the scale of these programs, an important question is to decide an *incentive payout* for each referral, in order to maximize the number of people reached within a given budget. This goal, however, is tricky to achieve because very little is known about the potential referrers and referees before a campaign is started. Each individual may have a different marginal cost for taking the effort to make a referral, and these costs are hard to estimate in advance. Thus, different people may respond in different, unknown ways to the same incentive. Further, little information about the target population's social network is known in advance, making this problem more challenging than the traditional influence maximization problems studied over social networks. To tackle this problem, our work builds upon two streams of work: optimizing the set of initial seed nodes and designing optimal referral incentives.

Social network-based influence maximization. Given a social network, Kempe et al. [14] provided a constant-ratio approximation algorithm to find initial "seed" sets of nodes to optimally spread influence in a social network. This was followed by many speed-up techniques over several years [4, 31, 36]. Subsequently, Golovin et. al. [10] extended the approach to include settings where seed nodes can be selected adaptively after observing results of previous selections. These algorithms, however, make the simplifying assumption that all individuals have the same response to a given incentive.

A related question, then is to find an optimal incentive payout strategy when people's response to incentives—their *response functions*—can be heterogeneous. When response functions are assumed to be step-functions based on a cost threshold, Lobel et al. [21] and Singer et al. [29] provide a

characterization of the problem in a game theoretic setup. Lobel et al. describe a game between people and the referral program where optimal incentives can be computed if individuals' cost to refer their friends are known. In practice, however, it is unclear how to estimate the cost for each referral, which may differ for individuals. As a solution, Singer et al. consider asking the referrers directly about their cost to refer their friends and design a mechanism to elicit truthful disclosure of incentive thresholds. However, asking people may not be practical in most referral programs.

Influence maximization over an unknown network. The above algorithms depend on full knowledge of the underlying social network. Recent work relaxes this assumption by assuming dynamic networks that can change over time [32, 40] or uncertainty in the edges of a given network [35]. Still, these approaches do not address the fundamental problem in referral programs where we have zero knowledge of the underlying social network.

Thus, while incentives matter in diffusion processes regardless of the social network [18], deciding a good incentive mechanism becomes critical in settings where the network is unknown. Through field experiments, Vasistha et al. [33] show how static incentives lead to more referrals than a standard lottery system, and Pickard et al. [24] show how recursive incentives can lead to successful mobilization for a task. In general, however, the effectiveness of an incentive program depends on a number of factors, including tie strength [27], referee's value for the content [39], and sensitivity of content being referred [17]. As a result, optimizing incentive payouts for arbitrary response functions and partially known social networks remains an open problem.

Given the lack of any apriori information, we will consider an online learning solution to this problem. That is, rather than making any assumptions about the nature of the social network or people's response functions, we will learn these from the observed data as referrals are made. To account for the various factors that can affect people's response functions, we will associate each referral with a set of features called the *context*. In the next section, we formally define our learning problem. Then in Section 4, we present *CoBBI*, our contextual bandit-based algorithm.

3 BACKGROUND AND PROBLEM FORMULATION

Let us define the incentive-aware influence maximization (IAIM) problem. We motivate our formulation through the Learn2Earn system and then use behavioral considerations to make it practical.

Problem Definition

Consider the problem of spreading mass awareness in low-resource communities. We assume that an informational message has been chosen and converted to a digital form (e.g., a voice recording). Given a fixed budget, we would like to use peer referrals to spread the message to the maximum number of people. As a specific example, the Learn2Earn system [30] uses voice recording of a message and wraps it around an Interactive Voice Response (IVR) system. Users call a toll-free number to access the IVR system, which then plays out the message and asks people to answer a few multiple-choice questions using their keypad. These questions are designed to verify that the user paid attention to the message. Finally, a user can share the Learn2Earn phone number to their friends by using a unique referral code. For each friend who calls the Learn2Earn system and enters their referral code, the user receives an INR 10 talktime on their phone.

Our insight is that optimizing the referral incentive involves maximizing the *causal influence* of an incentive in encouraging a person to share the message with others. That is, we would like to pay incentive only for the referrals that would not have happened without the incentive. For referrals that would have happened anyway, incentives are less useful, except perhaps to accelerate the process of referral. For instance, if people in a community are already sharing the message at high rates, it may not be useful to offer additional incentives. Instead, one may offer incentives to parts of the community where it is the hardest to spread the message. Similarly, one may want to pay more to a person who refers someone from a remote community, than when they refer someone who is connected to many other people and thus likely to find out about the system through other means. To capture a user v's *context*, we define a feature vector $\Phi(v, t)$, where t refers to time. This user context can include details of past referrals by v, number of previous payment offers made to v, and so on.

In addition to the base rate of influence spread, causal influence of referral incentives also depends on how people respond to the incentives. Different people can respond differently to a monetary incentive. Some may be motivated to share more, some may be unaffected, while some others may be even discouraged to share. The latter is plausible when people spread information as a part of building and sharing their social capital, and the introduction of monetary incentives may dissuade them, acting as a "repugnant" transaction [26]. For example, in an altruistic setting such as a referral program for a medically effective drug, people may be willing to refer their friends without any monetary incentive. However, being offered money might decrease the chances of referral (i.e., they might not want their friends to know that they are being paid for it). Thus, understanding how people respond to incentives is important. We define the (unobserved) response function of an individual v to an incentive by $\mathcal{R}_v = g_v(I)$, which characterizes how the individual v's sharing activity responds to the incentive I.

Formally, let $G(V, E)$ be the underlying social network with nodes $v \in V$ and edges $e \in E$. At any point, an influence maximizing agent can only access the observed part of the social network due to the referrals that have been made. For each new referral, the agent decides how much incentive should it award. More generally, the agent learns a policy $\mathcal{P}$ for awarding incentives based on the current context $\Phi(v, t)$ of a referrer v. Under such a setting, the goal is to devise an incentive payout policy $\mathcal{P}$ that maximizes the spread of content under a fixed budget.

We define the effectiveness of a payout policy $F^T(\mathcal{P})$ as the average number of people successfully influenced per unit of monetary incentive by following $\mathcal{P}$ for T time steps in the diffusion process. To benchmark the effectiveness of a payout scheme, we consider an oracle that knows both the social network G and the response functions $\mathcal{R}_v \ \forall v \in V$. Let $\mathcal{P}^*$ be the optimal payout scheme that this oracle outputs. Define the regret of any payout scheme $\mathcal{P}$ at time T as $R_P^T = F^T(\mathcal{P}^*) - F^T(\mathcal{P})$. Then, the IAIM Problem can be defined as:

Problem 1. ***IAIM Problem.*** *Given as input time T, and context $\Phi(v, t)$ for every potential referrer, the goal of the IAIM problem is to learn an optimal payout scheme function $\mathcal{P}$ which minimizes regret against an oracle scheme $\mathcal{P}^*$ after T time steps in the diffusion process, i.e.,* $\text{argmin}_{\mathcal{P}} F^T(\mathcal{P}^*) - F^T(\mathcal{P})$.

Considerations for Incentive Design

In practice, implementing an influence maximization system using referral incentives involves three additional questions: how much to pay, when to pay, and whom to pay.

How much to pay. Current systems such as Learn2Earn pay out a fixed monetary incentive for each referral. As we discussed above, it can be beneficial to customize the incentive based on the observed social network and response functions of individuals. However, having variable incentives can be confusing for users of the system and introduce concerns of differential payouts for the same referral task. It also adds additional complexity to program management since program officials need to know the exact dynamic incentive payouts to address users' queries on (failed) transaction payouts.

Therefore, even if guidelines can be fixed, a complex incentive structure will be difficult to communicate with all users. While it is mathematically appealing, for behavioral and program management reasons, we focus our attention to cases where the incentive amount is kept fixed.

When to pay. If the incentive amount is fixed, how would we then customize incentives? One way of customizing incentive payments is to change how often people are paid the same incentive. Paying some people more frequently than others can be shown mathematically to be equivalent to changing people's incentive payouts, within a multiplicative constant. In addition, based on behavioral research on loss aversion [13], we expect that changing frequency of payment, rather than the payment amount itself, can be a more acceptable way to customize incentives. It has the added advantage of being simple in its implementation.

A related problem to "when to pay" is in deciding when to disclose the incentive: before or just after a referral is made? Disclosing and paying the incentive just after is simpler and expects that the payout will motivate the user to refer more in the future. The alternative is to disclose an "offer to pay" first and communicate to the user a time period for which the payout offer is valid, and then pay conditional on whether the user referred another person in that time period. For a policy that pays an incentive for every referral, this distinction does not matter since users know apriori that they will receive a fixed incentive per referral. However, if we are choosing a specific subset of referrals for incentive payout, then this distinction can be important. Announcing the offer earlier sets expectations for the user and possibly also motivates them to refer someone. In comparison, when users are notified and paid after a particular referral, they do not know whether they will receive the incentive payout for any of their next referrals. Assuming that knowledge of an incentive payout in advance motivates people to refer, we focus on the strategy of *offering to pay* before a referral and then conditionally paying based on the referral made.

Whom to pay. Finally, we restrict our attention to the setting where the referrer is paid an incentive. It is possible that both the referrer and referee receive an incentive (i.e., double-sided incentives), or that only the referee receives the incentive. We leave these alternatives for future work.

Based on the above considerations, we define the *incentive policy* as an "offer to pay" scheme: a function $\mathcal{P}\Big(v, \Phi(v, t)\Big) \rightarrow \{0, 1\}$ that is called for each known user at time t and decides whether to offer an incentive payout or not. Note that the payout decision need not be the same for everyone; it can vary based on the user's context features $\Phi(v, t)$. We now propose a *practical* version of the IAIM problem.

Problem 2. **Practical *IAIM Problem.*** *Given as input time T, and context $\Phi(v, t)$ for every potential referrer, the goal of the IAIM problem is to learn an optimal "offer to pay" scheme function $\mathcal{P}$ that chooses the referrers to which a fixed payout will be offered, and that minimizes regret against an oracle scheme $\mathcal{P}^*$ after T time steps in the diffusion process, i.e.,* $\text{argmin}_{\mathcal{P}} F^T(\mathcal{P}^*) - F^T(\mathcal{P})$.

4 COBBI: CONTEXTUAL BANDIT OPTIMIZER

Without oracle access to the underlying social network or people's response functions to incentives, there is little information to decide whom to offer an incentive payout. Initially,

one might imagine choosing referrers at random since no better information is available. If people respond identically to incentives, it should be possible to learn whether offering to pay an incentive has a positive effect on the number of referrals, compared to the alternative of no incentives. This simple exploration strategy (similar to an A/B test) can be generalized using the multi-armed bandit problem [34]. That is, we can use a continous intermixing of exploration and exploitation: we employ the best known incentive policy but every once in a while, select people randomly to learn if those people respond more favorably to incentives. This strategy, known as an ϵ-greedy multi-armed bandit, works well for a variety of decision optimization problems [16].

However, a key complication in referral programs is that people's response to incentives is not a universal function; but rather a mixture of many diverse responses. Therefore, instead of thinking about a single optimal decision for everyone, it is more suitable to think about optimal decisions for different kinds of people who might be in different stages of the referral process. In other words, the right incentive depends on the *context* of each referrer, which includes factors such as the number of people already referred, the expected number of friends of a user, their value for incentives, and so on. Such a setting requires an extended version of a multi-armed bandit that can decide an optimal incentive for each referrer's context. We summarize the workings of a *contextual bandit* below and then describe how we use it to develop CoBBI, our online learning agent for the IAIM problem.

Contextual Bandits

In the contextual bandit problem [2, 6, 19], an agent repeatedly takes one of K actions in response to an observed context, and obtains a reward for the chosen action. Specifically, the agent collects rewards for actions taken over a sequence of *rounds*; in each round, the agent chooses an action on the basis of (i) *context* (or features) for the current round, and (ii) *feedback*, in the form of rewards obtained in previous rounds. Contextual bandit problems are found in many applications such as online recommendation and clinical trials [3, 20, 25]. Note that the feedback is incomplete: in any given round, the agent observes the reward only for the chosen action; it does not observe the reward for other actions. In our IAIM setting, we can observe the result of offering to pay to a person or not, but never both. Thus, random exploration in providing a payment offer can help in discovering the causal influence of offering to pay on future referrals.

Specifically, a contextual bandit tries to learn the distribution of rewards for each context-action pair; however, instead of learning a separate reward distribution for each context-action pair, it tries to generalize the reward distribution over the space of context vectors. Let A be a finite set of K actions, X be a space of possible contexts (e.g., a feature space). Let $\mathbb{R}_+^A := \{r \in \mathbb{R}^A : r(a) \geqslant 0 \; \forall a \in A\}$ be the set of non negative reward vectors. A contextual bandit *policy* π learns a function that outputs a decision given any context vector as input. Whenever the CB policy makes a decision, it receives a reward from the environment, signalling whether the action was effective. Over multiple rounds of actions, these rewards help the policy to optimize its decisions for achieving the maximum reward.

In the i.i.d. contextual bandit setting, the context/reward pairs $(x_t, r_t) \in X \times [0,1]^A$ over all rounds $t = 1, 2, ..$ are randomly drawn independently from a distribution $\mathcal{D}$. In any round t, the agent first observes the context x_t, then chooses an action $a_t \in A$, and finally receives the reward $r_t(a_t) \in [0,1]$ for the chosen action. The observable record of interaction resulting from round t is the tuple $(x_t, a_t, r_t(a_t)) \in X \times A \times [0,1]$. Let $\mathcal{R}(\pi) := \mathbb{E}_{(x,r)\sim\mathcal{D}}[r(\pi, x)]$ denote the expected instantaneous reward of a policy $\pi \in \Pi$, and let $\pi_* := argmax_{\pi\in\Pi}\mathcal{R}(\pi)$ be a policy that maximizes the expected reward (the optimal policy). Let $Reg(\pi) := R(\pi_*) - R(\pi)$ denote the expected (instantaneous) regret of a policy $\pi \in \Pi$ relative to the optimal policy. Then the (empirical cumulative) regret of an agent after T rounds is defined as follows: $\sum_{t=1}^{T} \Big(r_t(\pi_*(x_t)) - r_t(a_t)\Big)$.

Our goal is to find an algorithm whose regret with respect to the optimal policy is minimized. In the simplest contextual bandit algorithm, *ϵ-greedy*, the agent chooses to take the current best action with probability $1 - \epsilon$, and chooses to explore a random action with probability ϵ. We refer the reader to Langford and Zhang [19] for theoretical guarantees of the contextual bandit algorithm. Next, we explain how we map the IAIM problem into a contextual bandit.

CoBBI

CoBBI, our online learning agent consists of two different components: a *context-generation engine*, and the *IAIM bandit*. The context generation engine interacts with the world to maintain the best possible belief about the current state of the social network and the ongoing diffusion process in that network. The context-generation engin then parses this information into a context vector, which is then given to the IAIM bandit. The IAIM bandit looks at this context vector and decides whether to offer an incentive payout.

Context-Generation Engine. First, we describe the context space X for the IAIM problem. The context space X is composed of two disjoint spaces, $X = X_{net} \cup X_{diff}$, where X_{net} consists of context information related to the current knowledge of the ground-truth social network and X_{diff} consists of context information related to the current knowledge about the diffusion process. Note that in the beginning, the social network is completely unknown, and hence $X_{net} = \phi$

in the first round. As more and more referrals occur, the context generation engine builds up its understanding of the social network structure through the observed referral network, and uses this understanding to create the context sub-vector X_{net}. Possible features in X_{net} can be the the number of people referred by the user, the number of second-degree referral connections, etc. Similarly, possible features in X_{diff} can be the age, sex or location of the referrer, the number of times the referrer has been offered payment, etc.

The IAIM Bandit. The practical IAIM problem can be modeled as a contextual bandit. The proposed IAIM bandit takes a context vector X for a given individual at any time T and outputs whether he/she should be offered an incentive payout. In other words, the IAIM bandit serves as a dynamically updating payout agent, with a binary decision on whether to offer payment. For simplicity, we use the ϵ-greedy algorithm for learning the decision policy.

A critical choice for optimizing the bandit policy is the reward function, that maps <context, action> pairs to a reward between 0 and 1. A simple way is to assign a reward of 1 if the agent offered to pay a person and the person referred, and 0 if they did not. However, we would also need to consider the case when the agent decided not to offer to pay. If the person still refers someone, what should the agent's reward be? At first, it might seem as if the agent made a wrong prediction. However, going back to the causal influence definition from Section 3, if the person shares when the agent did not offer them any incentive, then that indicates that they would have shared anyways and thus the agent made the correct decision of not offering any incentive. Additionally, when the agent does not offer any incentive and the person does not share, we consider it as a neutral outcome. Thus, the correct reward ordering between the four different outcomes is: agent does not offer and person refers, agent offers and person refers, agent does not offer and person does not refer, agent offers and person does not refer.

However, when we consider rewards for multiple referrals, things get complicated. If a person refers once without an incentive offer, maybe they could have referred multiple times if they were given an incentive offer? As a possible solution, we propose separate reward functions for the cases when an incentive is offered or not ($\mu = 0.1$ is a parameter).

$$r_{offered} = 0.5^x; \quad r_{notoffered} = (0.5 - \mu)^x \tag{1}$$

These reward function ensure the ordering above when the number of referrals (x) by a person is 1, and then generalize the relationship for any number of referrals.

CoBBI's Workflow. We assume that an awareness program has initiated a social referral process by selecting and influencing some seed individuals from the underlying social network. First, CoBBI is invoked at each round for each influenced person. CoBBI uses its context-generation engine to create a context vector for an influenced person v. Then, CoBBI uses its IAIM bandit to decide whether to offer an incentive payout to the person v. After this step, v tries to refer the awareness message to their network connections in response to the incentive offer. If v is successful in referring and the IAIM bandit had offered an incentive, then they receive a payout for each referral. Based on this, the bandit's reward is updated as per Equation 1 and the process repeats.

Since the context consists of information about the inferred state of the social network, subsequent contexts in the IAIM problem are not necessarily i.i.d., and hence, CoBBI's policies are not guaranteed to be optimal. Still, as we show in the next sections, CoBBI learns effective payout policies for the practical settings encountered in referral programs.

5 EVALUATION SETUP

The ideal evaluation will be a randomized controlled trial where we test incentives based on CoBBI and compare with baseline policies. However, running such an experiment is non-trivial since it involves interventions on a social network and thus has spillover effects [8]. To partially control for the spillover, one could run different policies at different times on the same community (which may have exposure bias), or run different policies in different communities at the same time (which may have selection bias). Before running such a complex experiment, we would like to estimate how well CoBBI may perform. Thus, in this section, we utilize data from a Learn2Earn deployment and diffusion simulations to evaluate CoBBI against a wide range of baseline policies.

Configurations: Social influence model

We use data from a Learn2Earn awareness campaign on farmers' land rights conducted in a rural community in India [30]. We obtain a referral network over 3116 users that led to a total of 2826 referrals. We repurpose this referral network as a proxy for a new underlying, unobserved social network over which a different referral program can occur. While we acknowledge that the observed referral network is a subset of the true social network of the farmers' community, it still corresponds to real-world connections and thus provides a realistic network to conduct our simulations. Therefore, for our evaluation, we will treat this network as the maximum *realizable* network and the IAIM goal is to initiate a new referral program and reach as many people as possible under a given budget for incentive payouts.

To model transfer of content, we use a variant of the independent cascade model [14]. We first assume that an initial set of seed nodes are provided from which the referrals start (e.g., they could be randomly sampled). Subsequently, the social influence process proceeds in discrete time periods or

rounds, which can be thought of as *days* in the real-world setting. In each round, each node tries to refer their network connections with probability p, known as the influence probability. An influence probability of $p(v, w) = 0.5$ on an edge (v, w) denotes that if node v is already exposed to the message, it refers node w with $p = 0.5$. In the standard independent cascade model, all nodes that are influenced at round t get a *single* chance to influence their un-influenced neighbors at time $t+1$. If they fail to spread influence in this single chance, they don't spread influence to their neighbors in future rounds. Our model is different in that we assume that nodes get *multiple* chances to influence their un-influenced neighbors. If they succeed in influencing a neighbor at a given time step t', they stop influencing that neighbor for all future time steps. Otherwise, if they fail in step t', they try to influence again in the next round. This variant of independent cascade has been shown to empirically provide a better approximation to real influence spread [7, 37, 38]. Further, we assume that nodes that are influenced at a certain time step remain influenced for all future time steps, which is well-suited for referral programs aimed for mass awareness or product adoption. Based on the above setup, we construct the following influence model configurations.

- **Constant-Prob.** The influence probability is constant for all individuals in the social network. That is, each person is assumed to exert the same effort to spread a given message. We choose a conservative value, $p = 0.1$.
- **Random-Prob.** Influence probabilities of individuals are chosen uniformly at random. To ensure the same mean as above, probabilities are chosen randomly from (0, 0.2].
- **Friends-Prob.** Influence probability is a monotonically increasing function of the number of friends (n) of a person. We use $p = sigmoid(\frac{log(n)-\mu_{log(n)}}{\sigma_{log(n)}})$, where $\log(n)$ is standardized using its mean (μ) and standard deviation (σ). Like the probabilities above, p is bounded by 0.2.

Configurations: People's response functions

As described in Section 3, we assume that the influence probabilities can change due to an incentive offer. This corresponds to a situation where an incentive payout may motivate an individual to change their efforts at sharing, and thus the resultant influence probability. Specifically, each person $v \in V$ has an influence probability p_v and we define a parameter η that controls how much p_v is affected by an offer to pay incentive. Thus, η controls the *response function* of an individual. We consider three different response functions motivated by behavioral assumptions from Section 3.

- **Increasing response functions.** Much of past work on influence maximization assumes an increasing response function, where an individual's influence probability increases when offered an incentive payout. Thus, whenever a person v is offered an incentive, their influence probability to their friends becomes $p_v + \eta$.
- **Decreasing response functions.** Next, we consider monotonically decreasing response functions that model people's behavior in altruistic scenarios where an incentive offer may decrease an individual's referral efforts, as discussed in Section 3. Here, the influence probability becomes $p_v - \eta$ when v is offered an incentive for referral.
- **Idiosyncratic response functions.** Finally, we evaluate CoBBI's performance on idiosyncratic response functions. This class of response functions models people's behavior as completely unpredictable, which has recently received support with regards to social media [12]. To sample such idiosyncratic responses, we sample influence probabilities uniformly at random in the range $[p_v - \eta, p_v + \eta]$ for each individual who has been offered an incentive to refer.

Policies: Baseline methods

We compare our proposed contextual bandit-based agent, CoBBI with the following baseline methods.

- **NoPay**: Never offer to pay an incentive.
- **AlwaysPay**: Always offer to pay an incentive for a referral.
- **RandomPay**: Offer to pay individuals who are randomly selected according to some probability (default=0.5).
- **Pay>=5**: Offer to pay individuals once they have shared the message to atleast five of their friends.
- **Pay<=5**: Offer to pay individuals each time they share the message, but upto a maximum of five of their referrals.
- **PayMultipleOf5**: Offer to pay individuals once they have shared the message to {5, 10, 15…} people.
- **PayFriendsLen**: Offer to pay depending on the number of friends an individual has. This policy has a partial knowledge of the network, and hence not realistic. However, we include this method to compare CoBBI's performance to a method that has additional knowledge of the network.

These methods range from the simple to complicated, and are motivated by intuitive strategies to award incentives.

Context Feature Specification for CoBBI

As explained in Section 4, CoBBI relies on two sets of context features—X_{net} relating to the network, and X_{diff} relating to diffusion characteristics. For our Learn2Earn network, X_{net} is the number of friends influenced so far, and X_{diff} is the number of times an offer to pay is made by a policy. Note that in practice, available context features may exceed the basic ones listed above, and can be easily included. For example, features like age, sex or location of node v_1 may play an important role in determining his/her optimal incentive payout and should be included in X_{diff}.

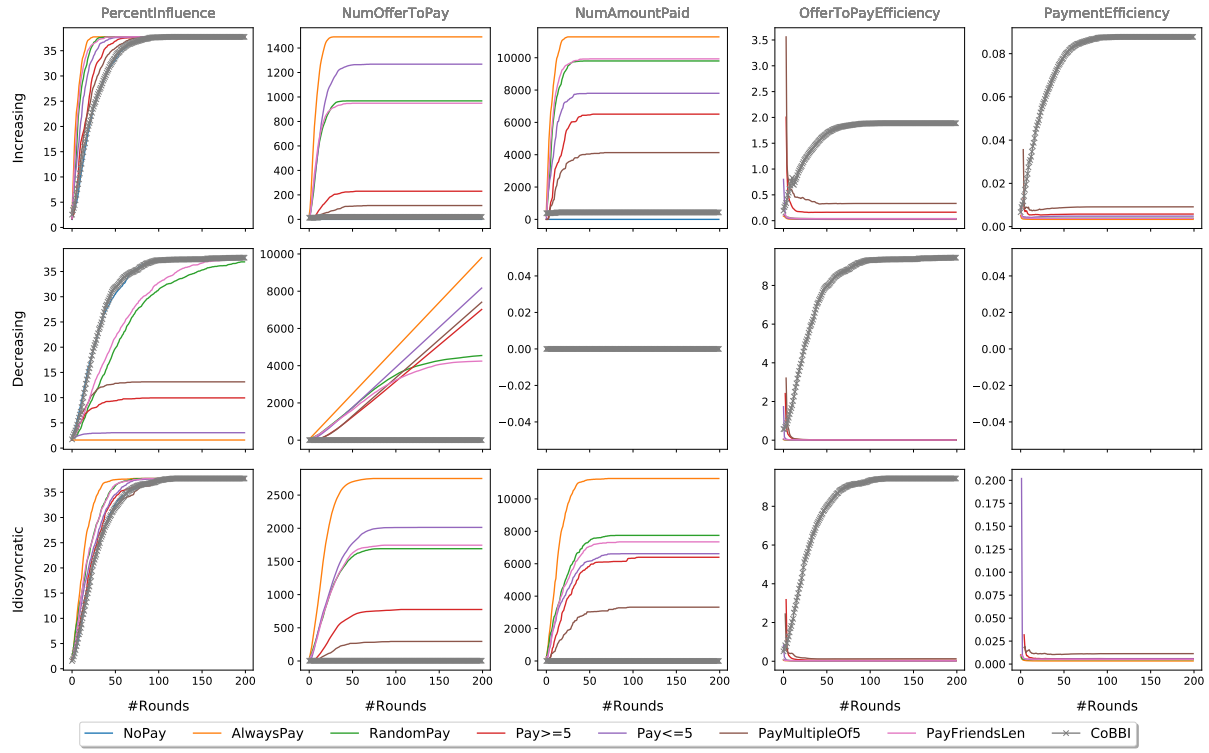

Figure 1: Comparing CoBBI to baseline policies under the Constant-Prob configuration. CoBBI achieves highest efficiency on giving out incentive offers and the total amount paid across increasing, decreasing and idiosyncratic response functions. For the decreasing response function and the *NoPay* policy, no payment was made and thus PaymentEfficiency is undefined.

Evaluation Metrics

To summarize the evaluation setup for a single simulated referral campaign, a fixed number (50) of seed nodes are chosen at random and provided as input. Next, influence starts spreading in the social network according to our influence model. At each round, for each person v that has been exposed to the message, the incentive offer policy decides whether to offer an incentive to v. If v has been offered an incentive, the influence probabilities $p(e)$ on all edges e adjacent to node v are updated dynamically (in accordance with $\mathcal{R}_v$, the response function of v). Further, if v is successful at referring his network connections, he receives an incentive payout per referral. The process continues until all nodes are influenced or the budget is exceeded. We set the budget as 6232 payouts in total, double the number of nodes in the network for an average of 2 payouts per node.

Based on the evaluation setup above, we deploy CoBBI and baseline policies on the Learn2Earn network data. All results reported are averages for 10 simulated referral campaigns for each <configuration, policy> pair. We set the response parameter ($\eta = 0.1$) and the bandit reward ($\mu = 0.1$) as defaults. The influence process for all campaigns saturated by 50 rounds; for completeness we show results for 200 rounds. We evaluate incentive policies on the following metrics:

- **PercentInfluence:** Percentage of nodes in the network that received the message.
- **NumOfferToPay**: Total number of times an offer to referral payment was made.
- **NumAmountPaid**: Total amount of money spent.
- **OfferToPayEfficiency**: *PercentInfluence/NumOfferToPay.*
- **PaymentEfficiency**: *PercentInfluence/NumAmountPaid.*

6 SIMULATION RESULTS

We now report on the effectiveness of CoBBI on three configurations from the last section, and discuss its sensitivity to people's response functions and its own reward function.

CoBBI obtains high efficiency

Figure 1 shows summary results for the simulated awareness campaigns on the Constant-Prob configuration. First, let us consider the increasing response function, shown in the top panel. The X-axis shows increasing rounds of the diffusion process (our notion of *time*) and the Y-axis shows evaluation metrics: percentage of nodes reached, number of times a policy offered to pay, total amount paid, offer-to-pay efficiency and payment efficiency. We find that CoBBI matches the maximum number of nodes reached by any of the baseline policies. Further, the CoBBI incurs substantially lower

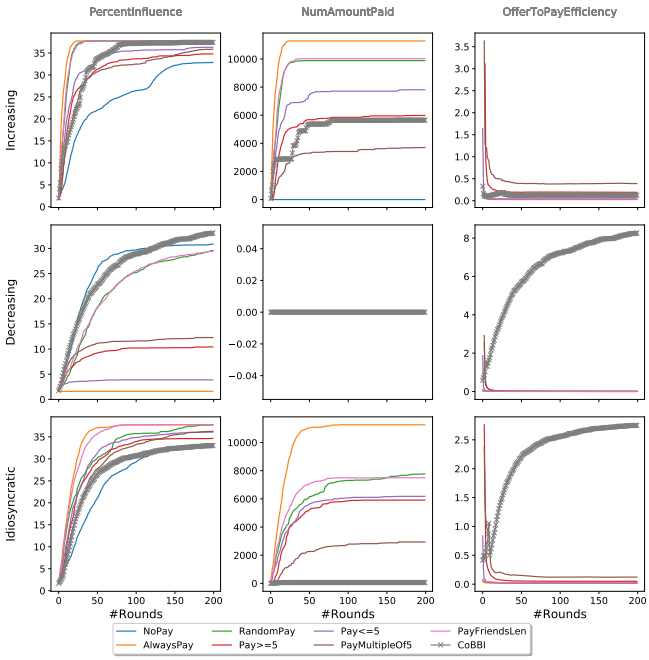

Figure 2: Evaluation metrics for comparing CoBBI and baseline policies under the Random-Prob configuration.

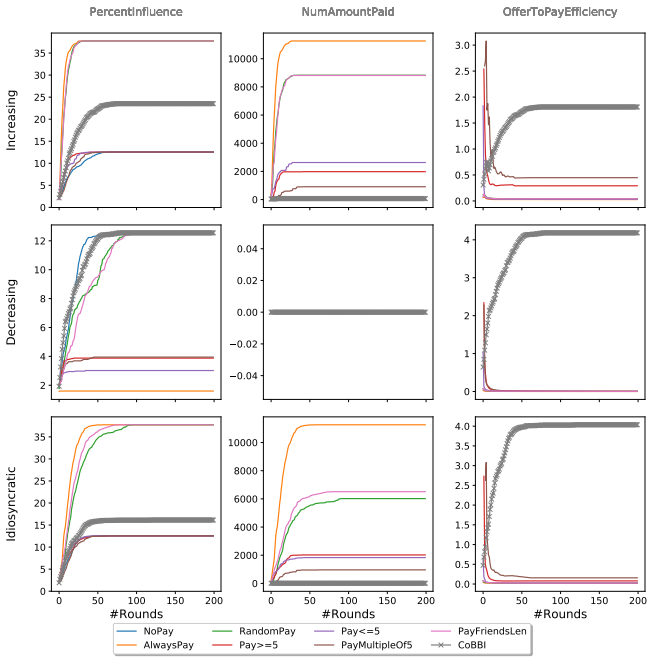

Figure 3: Evaluation metrics for comparing CoBBI and baseline policies under the Friends-Prob configuration.

cost among all the baselines that pay incentives: number of incentives offers is 6 times lower than the lowest baseline, and incentive amount paid is 10 times lower. Offer-to-Pay efficiency of CoBBI is nearly 2, indicating that its incentive offers lead to nearly 2 referrals on average, whereas the same for other baselines is less than 0.5. Similarly, CoBBI's payment efficiency is over 8 times the nearest baseline. Among the baselines, policies that pay rarely (*NoPay* and *PayMultipleOf5*) perform the best. Under this configuration, CoBBI learns that payments may not always be required to encourage referrals, even with an increasing response function.

When we look at the second panel of Figure 1 on decreasing response functions, CoBBI outperforms all other baselines that pay. The best strategy under a decreasing response function is *NoPay*, to not pay at all. *RandomPay* and *PayFriendsLen* also reach the maximum number of people influenced, but at a cost of making unnecessary incentive offers. Offer-to-Pay efficiency for CoBBI is 9, compared to 0.01 for*RandomPay* and *PayFriendsLen* respectively. We do not report PaymentEfficiency since all methods incur zero payment (no referrals happen when people are provided an incentive offer). The bottom panel of Figure 1 shows the setting with idiosyncratic response functions. Here we find similar results to the positive response function case: CoBBI matches baseline algorithms in reaching the maximum percentage of people, but does so slower than the paying baselines. However, it leads to a substantially higher efficiency (nearly 10 times for Offer-to-Pay) than those baselines.

Next, we consider the Random-Prob configuration where each person has a different, random propagation probability (Figure 2). We find a similar story: incentive offers by CoBBI can match the highest number of people reached by other policies, but do so at a slower rate than those policies. Especially for decreasing response functions, in terms of efficiency of offer-to-pay decisions and total amount spent, CoBBI outperforms these policies. Similarly, in the case of idiosyncratic response functions, while CoBBI reaches slightly fewer people than other policies (5%), its Offer-to-Pay efficiency is much higher (25 times) than other baselines.

The above two configurations show that CoBBI is able to adapt based on people's response functions. Even when people have different propagation probabilities, CoBBI uses bandit exploration to estimate the number of incentive offers needed to reach the maximum number of people, and thus spends the least amount of money among the policies that reach those many people. That said, we find that CoBBI can be too conservative for some settings. Figure 3 shows the third configuration where a person's propagation probabilities are set proportional to their number of friends. Since most (86%) of the people have only one friend in the network, this implies that the propagation probability is low (P=0.01) for most people, and only a few outliers have high probability (0.2). CoBBI is unable to model the effect of these outliers and can reach only two-thirds of the people reached by *AlwaysPay*, *RandomPay* and *PayFriendsLen*. It still has the highest efficiency (1.5-4 times) among these baselines as

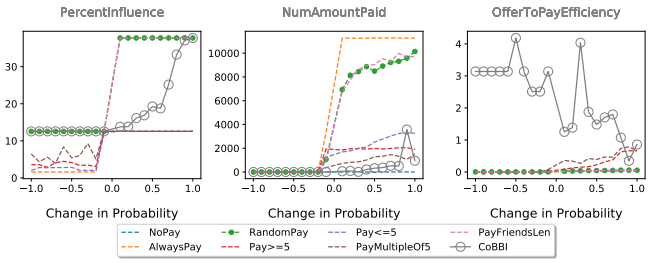

Figure 4: Sensitivity of CoBBI and baseline policies to η, the change in influence probability due to an incentive offer under the Friends-Prob configuration.

shown in Figure 3, but loses out due to being conservative in offering incentives.

Sensitivity to people's response functions

To investigate further for the Friends-Prob configuration, we now look at the sensitivity of CoBBI's performance to different values of η, the strength of the response function (Figure 4). For all policies, number of people reached decreases when the response function to incentives is negative, and increases when the response function has a positive relationship with incentive offers. When η is negative, CoBBI matches the best policy, *RandomPay*, in the number of people reached. When η is positive, *RandomPay* reaches more people. In all cases, as we saw before, CoBBI is the most cost-effective: it spends the least money and achieves the highest offer-to-pay efficiency (2-4 times more effective).

Impact of the CoBBI's reward function

While CoBBI generalizes well to different response functions, it does make a tradeoff between reducing incentive expenses and reaching the maximum number of people. This tradeoff is controlled by the reward function for the contextual bandit from Equation 1. The default parameter ($\mu = 0.1$) is set so that no referral without an incentive offer is slightly worse than a referral under an incentive offer, but can be tweaked. Figure 5 (right panel) shows how the efficiency of CoBBI changes as we change this parameter: higher μ leads to higher efficiency and lower number of people reached, and vice-versa for lower μ. This effect is consistent for different strengths of the response function, ranging from $\eta = \{-0.9, -0.5, -0.1, 0.1, 0.5, 0.9\}$. Interestingly, we find a similar tradeoff in other baseline policies, such as *RandomPay*, which can be parameterized by the probability of offering an incentive. When we change this probability from its default of 0.5 to a lower value (Figure 5, left panel), we see that its efficiency increases.

To summarize, we learned that CoBBI is more cost-effective than other baseline approaches under a wide set of propagation probability configurations and response functions.

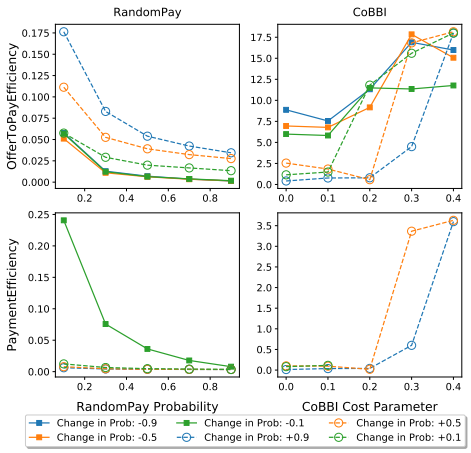

Figure 5: Impact of CoBBI's reward (cost) parameter μ on the tradeoff between increasing reach and minimizing the amount paid, for the Random-Prob configuration.

Moreover, a key advantage of CoBBI is that it is able to adapt its payout to different response settings, as we saw for the decreasing and idiosyncratic response functions where policies such as *RandomPay* reach fewer number of people. While we saw that some baselines can reach more people in the *Friends-Prob* configuration, CoBBI still achieves a higher cost efficiency; its tradeoff can be tuned in practice by changing the reward function.

7 DISCUSSION & FUTURE WORK

Even as referral programs have become increasingly popular for spreading information, prior models for influence maximization do not consider heterogeneity in people's response to incentives and partial knowledge about the structure of social networks. Therefore, we introduced the Influence-Aware Influence Maximization (IAIM) problem that captures these realities and presented a contextual bandit-based algorithm for finding an optimal incentive payout scheme.

While we modeled our solution to be faithful to constraints in a real-world referral program, there are limitations to our work. First, we assumed that a person's response to incentives depends only on the most recent incentive offer which ignores the effect of past incentives. Second, properties of successive referrals depend upon each other and thus the referral contexts received by CoBBI are not i.i.d., thereby violating the conditions for algorithmic guarantees for contextual bandits. Considering more stateful response functions and algorithms, and doing field experiments with referral programs will be useful future work.

ACKNOWLEDGMENTS

We thank William Thies for providing access to the Learn2Earn data and for his guidance throughout this work. We also thank Colin Scott and Devansh Mehta for their inputs.

REFERENCES

[1] Frances E Aboud and Daisy R Singla. 2012. Challenges to changing health behaviours in developing countries: a critical overview. *Social science & medicine* 75, 4 (2012), 589–594.

[2] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. 2014. Taming the monster: A fast and simple algorithm for contextual bandits. In *International Conference on Machine Learning*. 1638–1646.

[3] Ashwinkumar Badanidiyuru, John Langford, and Aleksandrs Slivkins. 2014. Resourceful contextual bandits. In *Conference on Learning Theory*. 1109–1134.

[4] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing Social Influence in Nearly Optimal Time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '14)*. SIAM, 946–957.

[5] Francis A Buttle. 1998. Word of mouth: understanding and managing referral marketing. *Journal of strategic marketing* 6, 3 (1998), 241–254.

[6] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*.

[7] Jean-Philippe Cointet and Camille Roth. 2007. How Realistic Should Knowledge Diffusion Models Be? *Journal of Artificial Societies and Social Simulation* 10, 3 (2007), 5.

[8] Dean Eckles, Brian Karrer, and Johan Ugander. 2017. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference* 5, 1 (2017).

[9] Amaya M Gillespie, Rafael Obregon, Rania El Asawi, Catherine Richey, Erma Manoncourt, Kshiitij Joshi, Savita Naqvi, Ade Pouye, Naqibullah Safi, Ketan Chitnis, et al. 2016. Social mobilization and community engagement Central to the Ebola Response in West Africa: lessons for future public health emergencies. *Global Health: Science and Practice* 4, 4 (2016), 626–646.

[10] Daniel Golovin and Andreas Krause. 2011. Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization. *Journal of Artificial Intelligence Research* 42 (2011), 427–486.

[11] Sonya Grier and Carol A Bryant. 2005. Social marketing in public health. *Annu. Rev. Public Health* 26 (2005), 319–339.

[12] Tianran Hu, Eric Bigelow, Jiebo Luo, and Henry Kautz. 2017. Tales of Two Cities: Using Social Media to Understand Idiosyncratic Lifestyles in Distinctive Metropolitan Areas. *IEEE Transactions on Big Data* 3, 1 (2017), 55–66.

[13] Daniel Kahneman, Jack L Knetsch, and Richard H Thaler. 1991. Anomalies: The endowment effect, loss aversion, and status quo bias. *Journal of Economic perspectives* 5, 1 (1991), 193–206.

[14] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 137–146.

[15] David A Kim, Alison R Hwong, Derek Stafford, D Alex Hughes, A James O'Malley, James H Fowler, and Nicholas A Christakis. 2015. Social network targeting to maximise population behaviour change: a cluster randomised controlled trial. *The Lancet* 386, 9989 (2015), 145–153.

[16] Levente Kocsis and Csaba Szepesvári. 2006. Bandit based Monte-Carlo Planning. In *Machine Learning: ECML 2006*. Springer, 282–293.

[17] Laura J Kornish and Qiuping Li. 2010. Optimal referral bonuses with asymmetric information: Firm-offered and interpersonal incentives. *Marketing Science* 29, 1 (2010), 108–121.

[18] Gabriel E Kreindler and H Peyton Young. 2014. Rapid innovation diffusion in social networks. *Proceedings of the National Academy of Sciences* 111, Supplement 3 (2014), 10881–10888.

[19] John Langford and Tong Zhang. 2008. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*. 817–824.

[20] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[21] Ilan Lobel, Evan Sadler, and Lav R Varshney. 2016. Customer referral incentives and social media. *Management Science* (2016).

[22] Victor Naroditskiy, Sebastian Stein, Mirco Tonin, Long Tran-Thanh, Michael Vlassopoulos, and Nicholas R Jennings. 2014. Referral incentives in crowdfunding. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

[23] United Nations OCHA. 2014. Ebola virus disease outbreak: overview of needs and requirements. *UN* (2014). https://www.unocha.org/sites/dms/CAP/Ebola_outbreak_Sep_2014.pdf

[24] Galen Pickard, Wei Pan, Iyad Rahwan, Manuel Cebrian, Riley Crane, Anmol Madan, and Alex Pentland. 2011. Time-critical social mobilization. *Science* 334, 6055 (2011), 509–512.

[25] Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. 2014. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*. SIAM, 461–469.

[26] Alvin E Roth. 2008. What have we learned from market design? *Innovations: Technology, Governance, Globalization* 3, 1 (2008).

[27] Gangseog Ryu and Lawrence Feick. 2007. A penny for your thoughts: Referral reward programs and referral likelihood. *Journal of Marketing* 71, 1 (2007), 84–94.

[28] Philipp Schmitt, Bernd Skiera, and Christophe Van den Bulte. 2011. Referral programs and customer value. *Journal of Marketing* 75, 1 (2011).

[29] Yaron Singer. 2012. How to win friends and influence people, truthfully: influence maximization mechanisms for social networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 733–742.

[30] Saiganesh Swaminathan, Indrani Medhi Thies, Devansh Mehta, Edward Cutrell, Amit Sharma, and William Thies. 2019. Learn2Earn: Using Mobile Airtime Incentives to Bolster Public Awareness Campaigns. *Technical Report* (2019). http://billthies.net/learn2earn-paper.pdf

[31] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-Optimal Time Complexity meets Practical Efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 75–86.

[32] Guangmo Tong, Weili Wu, Shaojie Tang, and Ding-Zhu Du. 2017. Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transactions on Networking (TON)* 25, 1 (2017), 112–125.

[33] Aditya Vashistha, Edward Cutrell, and William Thies. 2015. Increasing the reach of snowball sampling: The impact of fixed versus lottery incentives. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 1359–1363.

[34] Joannes Vermorel and Mehryar Mohri. 2005. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*. Springer, 437–448.

[35] Amulya Yadav, Hau Chan, Albert Xin Jiang, Haifeng Xu, Eric Rice, and Milind Tambe. 2016. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. 740–748.

[36] A Yadav, L Marcolino, E Rice, R Petering, H Winetrobe, H Rhoades, M Tambe, and H Carmichael. 2015. Preventing HIV Spread in Homeless Populations Using PSINET. In *Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence (IAAI-15)*.

[37] Amulya Yadav, Bryan Wilder, Eric Rice, Robin Petering, Jaih Craddock, Amanda Yoshioka-Maxwell, Mary Hemler, Laura Onasch-Vera, Milind Tambe, and Darlene Woo. 2017. Influence maximization in the field: The arduous journey from emerging to deployed application. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 150–158.

[38] Qiuling Yan, Shaosong Guo, and Dongqing Yang. 2011. Influence Maximizing and Local Influenced Community Detection based on Multiple Spread Model. In *Advanced Data Mining and Applications*. Springer, 82–95.

[39] Dan Zhou and Zhong Yao. 2015. Optimal Referral Reward Considering Customer's Budget Constraint. *Future Internet* 7, 4 (2015), 516–529.

[40] Honglei Zhuang, Yihan Sun, Jie Tang, Jialin Zhang, and Xiaoming Sun. 2013. Influence maximization in dynamic social networks. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE.