Combining Traditional Marketing and Viral Marketing with Amphibious Influence Maximization

WEI CHEN, Microsoft Research, weic@microsoft.com
FU LI[1], Tsinghua University, fuli.theory.research@gmail.com
TIAN LIN[2], Tsinghua University, lint10@mails.tsinghua.edu.cn
AVIAD RUBINSTEIN[3], University of California at Berkeley, aviad@eecs.berkeley.edu

In this paper, we propose the *amphibious influence maximization (AIM)* model that combines traditional marketing via content providers and viral marketing to consumers in social networks in a single framework. In AIM, a set of content providers and consumers form a bipartite network while consumers also form their social network, and influence propagates from the content providers to consumers and among consumers in the social network following the independent cascade model. An advertiser needs to select a subset of seed content providers and a subset of seed consumers, such that the influence from the seed providers passing through the seed consumers could reach a large number of consumers in the social network in expectation.

We prove that the AIM problem is NP-hard to approximate to within any constant factor via a reduction from Feige's k-prover proof system for 3-SAT5. We also give evidence that even when the social network graph is trivial (i.e. has no edges), a polynomial time constant factor approximation for AIM is unlikely. However, when we assume that the weighted bi-adjacency matrix that describes the influence of content providers on consumers is of constant rank, a common assumption often used in recommender systems, we provide a polynomial-time algorithm that achieves approximation ratio of $(1 - 1/\mathrm{e} - \varepsilon)^3$ for any (polynomially small) $\varepsilon > 0$. Our algorithmic results still hold for a more general model where cascades in social network follow a general monotone and submodular function.

Categories and Subject Descriptors: G.2 [**Mathematics of Computing**]: Discrete Mathematics; G.3 [**Mathematics of Computing**]: Probability and Statistics; F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Economics, Theory

Additional Key Words and Phrases: viral marketing, influence maximization, amphibious influence maximization, hardness of approximation, social networks, influence diffusion

---

[1] This work was mostly done while visitng Microsoft Research Asia. This research was supported in part by National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, and by National Natural Science Foundation of China Grant 61033001, 61361136003.
[2] This work was mostly done while interning at Microsoft Research Asia.
[3] This work was mostly done while interning at Microsoft Research Asia, and partly at the Simons Institute for the Theory of Computing. This research was supported by NSF grants CCF0964033 and CCF1408635, and by Templeton Foundation grant 3966.

EC'15, June 15–19, 2015, Portland, OR, USA. ACM 978-1-4503-3410-5/15/06 ...$15.00.

http://dx.doi.org/10.1145/2764468.2764480

1. INTRODUCTION

Marketing is traditionally partitioned into several stages: advertisers pay content providers (e.g. TV networks, radio stations, online news sites, influential bloggers, etc.); content providers recruit audience; and then the audience who are exposed to the advertisements influence their friends. Today, with the development of the Internet and social networks, there is an enormous amount of data that can be used to predict which users will enjoy a specific content, which users are likely to purchase the advertised product, and which users can influence their friends to buy the product as well. More importantly, information is available to track the individuals who participate in each one of those interactions. This suggests a new marketing approach in which advertisers can contact both content providers and the audience at the same time, with the goal of maximizing the overall exposure (through direct exposure as well as propagation via social networks) to the advertisement.

Consider the following example. Suppose a technology company wants to select a subset of regular tech bloggers (content providers) and engage them with marketing activities so that they would cover the company extensively and favorably. However, this alone does not guarantee that these favorable blogs can reach the targeted customers of the company. The company may further select a number of non-bloggers and spend its marketing effort on them (e.g. buying advertising slots to remind them about the blog entries of their selected bloggers) to make them active in subscribing, reading, and propagating the blog entries written by the company's selected bloggers. The objective of the company is to maximize the number of targeted customers who get exposed to the favorable blogs, either directly or indirectly through links forwarded by friends in the social network.

The above proposed marketing strategy can be viewed as a combination of traditional marketing via content providers and viral marketing in social networks. It can be modeled as a controlled diffusion in a joint network consisting of a bipartite graph modeling provider-consumer relationship and a social graph modeling social influence relationship among the consumers. The bipartite graph and its edge weights indicate the influence from content providers to consumers, while the social graph and its edge weights indicate the influence among consumers. An advertiser wants to select a subset of content providers (called *seed providers*) and a subset of consumers (called *seed consumers*) in the social network such that the influence from seed providers could activate enough seed consumers, which in turn could activate more consumers in the social network. Since the two marketing activities involve costs of different types, we enforce separate budgets on provider selection and consumer selection.

In this paper, we model the above combined marketing strategy as the following *amphibious influence maximization (AIM)* problem. We are given (a) a bipartite graph $B = (U, V, M)$ where U represents content providers, V represents consumers, and M is the weighted biadjacency matrix representing the influence probabilities from providers to consumers; and (b) a directed social graph $G = (V, P)$ where V is the same set of consumers as in B and P is the weighted adjacency matrix representing influence probabilities of each consumer over her friends. Given a subset $X \subseteq U$ of seed providers and a subset $Y \subseteq V$ of seed consumers, the influence propagates from X to Y and then to other consumer nodes in V following the independent cascade model [Kempe et al. 2003]. Given budgets b_1 for providers and b_2 for con-

sumers, the AIM problem is to select at most b_1 seed providers and b_2 seed consumers such that the expected number of activated consumer nodes after the diffusion process is maximized.

One important nature of the AIM problem formulation is that seed consumer selection is non-adaptive. That is, we need to select seed providers and seed consumers together before we observe the actual cascades from the seed providers. This is motivated by long-term marketing campaigns, during which repeated cascades may be generated from content providers. For such campaigns it is impractical for advertisers to adaptively select seed consumers for every cascade, and thus non-adaptive seed consumer selection aiming at maximizing the cumulative effect over multiple cascades is desirable.

Our results

We study both the hardness of the AIM problem and its approximation algorithms in the independent cascade (IC) model [Kempe et al. 2003]. In terms of hardness, we warm up (Section 3) with an easy result that finding any constant-factor approximation for AIM (even when the social network graph has no edges at all) is as hard as approximating the densest-k-subgraph problem, for which no polynomial-time algorithm is known. Our main impossibility result (Section 4) is that AIM is also NP-hard to approximate to within any constant factor. The result is proven by a reduction from Feige's k-prover proof system for 3-SAT5 [Feige 1998].

In order to overcome the above strong inapproximability results, we introduce additional assumptions in our model. Both hardness reductions construct a providers-consumers bipartite graph B with a complex and elaborate structure. In practice, even if the true relationship is indeed so intricate in nature, most of the learning techniques that are used to estimate this relationship assume some simple underlying structure - so we can expect the input for our algorithm to be "simple". In particular, for the specific motivation of influence of content providers on consumers, a common assumption in the construction of the influence matrix is that it is (approximately) low-rank (e.g. the "Netflix Problem"; [Koren et al. 2009]). This assumption is typically motivated by modeling the relationship between content and consumers via a small number of (hidden) features. In Section 5 we show that when the weighted bi-adjacency matrix P has constant rank, we can approximate AIM to within a factor of $(1 - 1/\mathrm{e} - \varepsilon)^3$ in polynomial time for any (polynomially small) $\varepsilon > 0$. Our algorithmic result can be generalized to accommodate any diffusion model in the social network that has a monotone, submodular, and polynomial-time computable influence spread function.

1.1. Related work

Influence maximization is first studied as an algorithmic problem with application to viral marketing by Domingos and Richardson [2001]; Richardson and Domingos [2002]. Kempe et al. [2003] first formulate it as a discrete optimization problem. They summarize the independent cascade model and linear threshold model, and apply submodular function maximization to obtain approximation algorithms for influence maximization. Extensive research has been done since to improve the scalability of the algorithm, extending the model to competitive setting, etc. (cf. [Chen et al. 2013]).

Conceptually, amphibious influence maximization combines viral marketing with traditional marketing via content providers, and thus it enriches viral marketing and its technical

formulation of influence maximization to a new level. Technically, AIM also contains influence maximization as a special case: when we have provider budget $b_1 = |U|$ (allowing all providers to be seeds) and bi-adjacency matrix to be all-one matrix (providers would deterministically activate all seed consumers), AIM is reduced to the classical influence maximization problem.

Recently, Seeman and Singer initiated a line of works [Seeman and Singer 2013; Badanidiyuru et al. 2014; Rubinstein et al. 2015] on adaptive seeding in social networks that is closely related to ours. In the adaptive seeding problem, a small subset X of the nodes in a social network is initially available to an advertiser. In the first stage, the advertiser selects (or *seeds*) a subset S of these nodes, who may influence some of the neighbors. In the second stage, a random subset of the neighbors of S becomes available; the advertiser spends the rest of her budget on seeding a subset of the newly available nodes, in hope to maximize their influence in the social network. The most important difference between Seeman and Singer's model and ours is that in the former, the seeding is *adaptive*, i.e. the advertiser waits to see which of the second layer's nodes became available before selecting a subset. Recall in our model, per contra, the advertiser must seed consumers in advance; in particular, there is no guarantee that after the edge percolation, a seed consumer $y \in Y$ will have live edges with seed content providers. As already discussed, non-adaptive seeding is appropriate for marketing campaigns during which repeated influence cascades may occur.

From a technical viewpoint, although we certainly build on ideas from [Badanidiyuru et al. 2014; Rubinstein et al. 2015], adaptivity completely changes the approximability of the problem: all the works above achieve constant-factor approximations in different settings of adaptive seeding, while we show that in the non-adaptive case, constant-factor approximation is impossible[1]. Interestingly, all the above works on adaptive seeding use a non-adaptive relaxation of the adaptive problem. It turns out that unlike the non-adaptive AIM problem, *the non-adaptive relaxation can be approximated efficiently to within a constant factor*. (The precise factor of approximation depends on other parameters of the problem such as IC model vs. a general submodular function.)

The problem of acceptance probability maximization (APM) for active friending studied by Yang et al. [2013] is also related to our work. In APM, a source node needs to select k intermediary nodes situated between the source and the target node in a social network, such that if influence from the source only propagates through intermediaries, the probability of activating the target is maximized. AIM and APM are similar in that both need to select some intermediary nodes between the source and the target and both study the non-adaptive version. However, their assumptions on influence cascade are different: APM assumes that cascades only occur in the sub-network consisting of the source, the selected intermediaries and the target, while AIM assumes that cascades occur from the selected sources to the selected intermediaries but from intermediaries cascades can reach the entire social network. For APM problem, Yang et al. [2013] only provide a heuristic algorithm and do not have hardness of approximation results. In our full report [Chen et al. 2015], we build on the hardness

[1]Note that this is a comparison of the algorithmic limitations within each model, and not a competitive analysis. In particular, whenever seeding adaptively is feasible, it is of course preferable and can perform much better than "non-adaptive seeding". As mentioned earlier, our motivation for studying a non-adaptive model is settings where the time required to estimate long-term influence of a marketing campaign makes adaptive seeding impractical.

of approximation of AIM to prove that it is NP-hard to approximate APM in a general graph to within a near-exponential ($2^{n^{1-\varepsilon}}$) factor.

Finally, our algorithm for AIM with constant rank was inspired by recent works that (approximately) solve the Densest-k-Bi-Subgraph problem in graphs with (approximately) constant rank [Alon et al. 2013; Papailiopoulos et al. 2014].

2. MODEL AND PROBLEM DEFINITION

We consider a (heterogeneous) network consisting of the following two components. The first is a bipartite graph $B = (U, V, M)$, where U represents content providers (e.g. bloggers, TV programs, etc.), V represents consumers, and M is the $|U| \times |V|$ weighted bi-adjacency matrix with $M_{ij} \in [0, 1]$ denoting the probability that $i \in U$ would successfully activate $j \in V$ (e.g. j is influenced by the advertisement associated with i). The second is a directed social graph $G = (V, P)$, where V is the same as the V in the bipartite graph B, and P is the $|V| \times |V|$ weighted adjacency matrix with P_{vw} denoting the influence probability from $v \in V$ to $w \in V$. We denote the set of directed edges of the social graph as $E = \{(v, w) \mid P_{vw} > 0\}$.

After fixing a set of seed providers $X \subseteq U$ and a set of seed consumers $Y \subseteq V$, we model the influence diffusion from X to the nodes in the social graph G as follows. For each edge (i, j) in B we sample it as *live* with probability M_{ij} and *blocked* with probability $1 - M_{ij}$; for each edge $(v, w) \in E$, we sample it as live with probability P_{vw} and blocked with probability $1 - P_{vw}$. We say that a node $v \in V$ is *activated (by the influence of X through Y)* if there is a path $(x, y, v_1, \ldots, v_t = v)$ with $x \in X$ and $y \in Y$, and all edges on the path are live. Given X and Y, we use $\sigma(X, Y)$ to denote the expected number of activated nodes in V (with expectation taken among all samples on all edges), and call it the *influence spread* of X and Y.

Note that the diffusion model can be equivalently described as follows.[2] First, every seed $i \in X$ independently tries to activate every node $j \in Y$ with success probability M_{ij}, and $j \in Y$ is activated as long as some $i \in X$ activates y, and nodes outside Y are not activated by seeds in X. Let $S \subseteq Y$ be the (random) set of nodes activated in Y. Then we treat S as the seed set and apply the independent cascade model [Kempe et al. 2003] to start the influence diffusion from S in the social network G using influence probabilities P: namely at each discrete time step, each newly activated node $v \in V$ has one chance to activate each of its outgoing neighbor $w \in V$ with probability P_{vw}.

Our goal is to find a set X of seed providers of size b_1 and a set Y of seed consumers of size b_2 such that they work together to generate the largest influence spread, which we formally define below.

DEFINITION 1 (AMPHIBIOUS INFLUENCE MAXIMIZATION). *In the* AMPHIBIOUS INFLUENCE MAXIMIZATION *(*AIM*) problem, we are given a bipartite graph $B = (U, V, M)$ and a directed social graph $G = (V, P)$, and budgets b_1 and b_2, and we want to find a subset $X^* \subseteq U$ of size b_1 and a subset $Y^* \subseteq V$ of size b_2 such that the influence spread of X^* and Y^* are maximized, that is, finding X^* and Y^* such that*

$$(X^*, Y^*) = \operatorname*{arg\,max}_{X \subseteq U, |X| = b_1, Y \subseteq V, |Y| = b_2} \sigma(X, Y).$$

[2] Equivalence is in the sense of the distribution of final set of activated nodes in V.

Several remarks are now in order. First, when we set $b_1 = |U|$ and M as an all-one matrix, the AIM problem is reduced to the classical influence maximization problem defined in [Kempe et al. 2003]. Thus, AIM is a generalization of the classical influence maximization problem such that it considers interactions between the provider nodes U and consumer nodes V and they have to work together to spread the influence. Second, it is easy to see that when either fixing set X or Y, $\sigma(X, Y)$ as a set function of the other variable is monotone and submodular.[3] However, the interaction of X and Y makes the AIM problem much harder than the classical influence maximization problem: we need both nodes in X and Y to generate influence and missing either of them will not work. Finally, our results can be generalized to allow diffusion models in the social network to follow any monotone and submodular function, and non-seed consumers to be influenced with background probabilities. To simplify the presentation, we focus on the main problem given in Definition 1 and discuss the generalization in Section 6.

3. HIDDEN-CLIQUE HARDNESS

Before we derive our main hardness result, we briefly describe in this section a much simpler reduction which gives a weaker hardness, "Hidden-clique hardness" (sometimes also "planted-clique"). Another feature of this result is that in the hard instance the social network graph G has no edges at all!

Hidden clique. In an Erdos-Renyi random graph $\mathcal{G}(n, 1/2)$ the largest clique size is approximately $2\log_2 n$, with high probability (e.g. [Alon and Spencer 1992]). We can "plant" a clique of size $t \gg 2\log n$, by choosing t nodes at random, and connecting all the edges between them. The hidden clique problem (e.g. [Alon et al. 2011]) is to distinguish between a graph sampled from $\mathcal{G}(n, 1/2)$ and a graph from $\mathcal{G}(n, 1/2)$ with a planted clique. Alon et al. [Alon et al. 2011] reduce this problem to solving the following gap version of DENSEST k-SUBGRAPH. Although the planted clique problem has been extensively studied, the best known algorithms run in quasi-polynomial time ($n^{O(\log n)}$); in particular, there are no known polynomial-time algorithms for the hidden clique problem.

THEOREM 3.1. *(Theorem 1.3 of [Alon et al. 2011]) If there is no polynomial-time algorithm for the hidden clique problem with a planted clique of size $t = n^{1/3}$, then for any $\delta > 0$, there is no polynomial-time algorithm that given a graph G distinguishes between:*

Completeness. G has a clique of size k; and
Soundness. Every k-subgraph of G has density at most δ.

Hardness for AIM follows as a corollary:

COROLLARY 3.2. *If there is no polynomial-time algorithm for the hidden clique problem with a planted clique of size $t = n^{1/3}$, then* AIM *cannot be approximated to within a constant factor in polynomial time - even in the special case where the social network graph has no edges.*

PROOF. We give a reduction from the DENSEST k-SUBGRAPH problem.

[3] A set function f is monotone if for all $S \subseteq T$, $f(S) \leq f(T)$, and submodular if for all $S \subseteq T$ and $v \notin T$, $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$.

Reduction. Given an instance $G^{DkS} = \left(V^{DkS}, E^{DkS}\right)$ of DENSEST k-SUBGRAPH with gap parameter δ, we construct an AIM weighted bipartite graph $B = (U, V, M)$ between content provider nodes and consumer nodes as follows: We identify both U and V with the original set of vertices V^{DkS} (i.e. our AIM instance has twice as many vertices). For any $u \in U$ and $v \in V$, we set $M_{u,v} = 1/n^2$ if the corresponding vertices in G^{DkS} are distinct and have an edge between them, and $M_{u,v} = 0$ otherwise. We set the budgets to $b_1 = b_2 = k/2$.
Completeness. If G^{DkS} contains a k-clique $C \subseteq V$, then partition C into two subsets of size[4] $k/2$ and label them X' and Y'. Consider their respective copies $X \subseteq U$ and $Y \subseteq V$ in the bipartite graph: it follows from the construction that $X \cup Y$ is a bi-clique. Thus, every consumer in Y has probability $1 - \left(1 - 1/n^2\right)^{k/2} = (1 - o(1)) \cdot k/\left(2n^2\right)$. Summing over all $k/2$ consumers, the expected number of activated nodes is $OPT = (1 - o(1)) \cdot k^2/\left(4n^2\right)$.
Soundness. Let X, Y be an optimal solution of the AIM instance. Let $S \subseteq V^{DkS}$ be the union of the copies of X and Y in G^{DkS}. By the premise, S contains at most $\delta\binom{k}{2}$ edges. Thus there are at most $2\delta\binom{k}{2}$ edges between X and Y, each with weight $1/n^2$ (we may count some edges twice in case the copies of their endpoints belong to both X and Y). Therefore, $\sigma(X, Y) < \delta k^2/n^2 < 5\delta \cdot OPT$.

□

4. NP-HARDNESS OF APPROXIMATION

In this section we prove our main hardness result, namely:

THEOREM 4.1. AIM *is* NP*-hard to approximate to within any constant factor.*

Proof outline. We reduce from Feige's k-prover proof system [Feige 1998]. The provers' answers to questions correspond to the provider nodes in U. The provider nodes are connected to a subset $V_1 \subset V$ of the consumers, on which the verifier can test the provers' answers. Since the edges from U to V_1 appear with low probability, it is significantly more cost-effective to select a few nodes from V_1 with many neighbors in U. Intuitively, this corresponds to a verifier's test which many provers would pass. By Theorem 4.2, if we start from a satisfiable formula, all k provers will agree - versus less than 2 provers that agree for an unsatisfiable formula. The rest of the consumers, $V_2 = V \setminus V_1$, have incoming edges from influential consumers in V_1. They will guarantee that the provers answer (almost) all the verifier's questions.

Notice that our hard instance is a three-layered graph. We henceforth call the provider nodes the *top layer*, the influential consumers V_1 constitute the *middle layer*, whereas the *bottom layer* has the rest of the nodes.

k-prover proof system. Consider k provers trying to prove the satisfiability of some 3-SAT5 formula over n variables. A 3-SAT5 formula is a conjunctive-normal-form (CNF) formula where each variable appears in exactly 5 clauses, and each clause contains exactly 3 variables; notice that there are $5n/3$ clauses. The verifier selects l clauses (with replacement) uniformly and independently at random. For each clause, the verifier selects one of the par-

[4]We assume without loss of generality that k is even. Given a polynomial time algorithm for even k it is easy to extend to an algorithm for $k - 1$; e.g. by adding a dummy vertex that is connected to all vertices in the graph.

ticipating variables uniformly and independently at random; we call those the *distinguished* variables. For each random string $r \in R$, we associate a question $q \in Q$ for each prover $i \in [k]$; we denote this as $(q, i) \in r$. Each question consists of $l/2$ clauses, and the distinguished variables form the remaining $l/2$ clauses. An answer a to question q consists of assignments to the $l/2 + 3l/2 = 2l$ variables in question. We henceforth abuse notation and also use R and Q to denote the corresponding cardinalities $R = n^l \cdot 5^l$ and $Q = n^l \cdot \left(\frac{5}{3}\right)^{l/2}$. Given the k provers' answers, Feige's verifier tests the provers answers by comparing their answers on the l distinguished variables. (We will diverge from Feige's construction at this point and use a stronger test that compares the provers' answers on all $3l$ variables.) For constant l, Feige proves the following theorem.

THEOREM 4.2. *(Lemma 2.3.1 in [Feige 1998]) Given a k-prover system, it is* NP*-hard to distinguish between the following:*

> *Completeness. all the provers pass the verifier's test with probability* 1*; and*
> *Soundness. the probability that any pair of provers pass the verifier's test is at most* 2^{-cl}*, for some constant* $c > 0$*.*

Construction. We construct a directed graph with three layers: U, V_1, V_2. The first layer U is precisely the set of "content providers" in our model. The set of "consumers" populates the middle and bottom layers $V = V_1 \cup V_2$. In terms of the weighted bi-adjacency matrix, the layered structure means that $M_{u,v} = 0$ for all $u \in U$ and $v \in V_2$, and similarly $P_{v_1,v_2} = 0$ unless $v_1 \in V_1$ and $v_2 \in V_2$.

Going back to the k-prover system, the top layer U corresponds to triplets of provers' answers to questions; the middle layer V_1 corresponds to assignments to variables -*distinguished and non-distinguished*- that may appear in the verifier's question to any of the provers; finally, the bottom layer corresponds to the random strings of the verifier. All the edges go from the top to the middle layer, or from the middle to the bottom layer. In particular, the graph is tri-partite.

More specifically, for each triplet (q, a, i) of (question, answer, prover) we have a corresponding node in U. For each pair $(r, \overline{a_r})$ of (verifier's random string, assignment to all $3l$ variables) we have a node in V_1. Notice that this is different from [Feige 1998], where the elements to be covered correspond to (r, a_r, i) with a_r being the assignment only for the distinguished variables. The (q, a, i) node is connected to all the nodes $(r, \overline{a_r})$ such that: $(q, i) \in r$, and when restricting $\overline{a_r}$ to the variables specified by (q, i), it is equal to a. In particular, for each i, each $(r, \overline{a_r})$ corresponds to only one (q, a, i). We set the top-layer budget to be the number of nodes in U that correspond to a single assignment, $b_1 = kQ$; similarly we let $b_2 = R$ represent the number of nodes in V_1 that match the same assignment. Finally, all the edges from U to V_1 have probability $1/k$.

For each random string r, we have η nodes in the bottom layer, V_2. We choose a sufficiently large η to ensure that most of the utility comes from the bottom layer. The nodes corresponding to each r are connected to all the nodes $(r, \overline{a_r})$ in V_1 with probability 1. The role of this layer is to force any good assignment to spread its budget across the different random strings (i.e. make sure that the provers answer all the questions).

See Table I for a summary of notation.

Table I: Summary of notation in main reduction

Notation	Interpretation in k-provers system	Vertices in AIM
(q, a, i)	question, answer, prover	1 vertex in U
(q, i)	question, prover	$2^{3l/2}$ vertices in U
r	random string	$k \cdot 2^{3l/2}$ vertices in U ($\forall\ (q,i) \in r$ and $a \in \{0,1\}^{3l/2}$)
$(r, \overline{a_r})$	random string, assignment to all $3l$ variables	1 vertex in V_1
r	random string	η vertices in V_2
(r, h)	random string, copy	1 vertex in V_2

Completeness. Given a satisfiable assignment to the 3SAT-5 formula, we select in the top layer a subset $S \subset U$ of kQ nodes that correspond to the same assignment. Because they all correspond to the same assignment, for each random string r, all k corresponding nodes in S are connected to the common node $(r, \overline{a_r}^*)$. In the middle layer, we let T be the set of these R nodes (i.e. $(r, \overline{a_r}^*)$ for $r \in R$). Before sampling the edges, each $(r, \overline{a_r}^*)$ has k neighbors in S. After sampling, the probability that there is a path from S to $(r, \overline{a_r}^*)$ is $1 - \left(1 - \frac{1}{k}\right)^k \approx 1 - 1/e$.

Since each node in T has η neighbors in V_2 (with probability 1), the value of this solution is approximately $OPT \approx (1 - 1/e)R\eta$.

Soundness. In an unsatisfiable instance, any two provers agree for at most a $\left(2^{-cl}\right)$-fraction of the random strings. We will show in Lemma 4.3 that there are at most $\left(2 \cdot 2^{-(1/3)cl} \cdot R\right)$ *good* random strings r, which are strings r such that there is a node $(r, \overline{a_r})$ with more than one neighbor in S. Since for each random string r there are only η nodes in V_2, each of the good random strings contributes at most η to the value of the solution. Before sampling the edges, any node that does not correspond to a good random string has at most one neighbors in S. After sampling, the probability that any such node has a neighbor in S is at most $1/k$. Since each node in V_1 has η neighbors in V_2, the total contribution from R nodes that do not correspond to good random strings is bounded by $R\eta/k$. Therefore, the expected number of covered nodes is bounded by the contribution of the middle layer, plus the contributions from the good and bad random strings:

$$R + \left(2 \cdot 2^{-(1/3)cl} \cdot R\right)\eta + R\eta/k = \left(1/k + o\left(1\right)\right) R\eta \approx \left(\frac{e}{e-1} \cdot \frac{1}{k}\right) OPT.$$

Lemma 4.3 below completes of Theorem 4.1.

LEMMA 4.3. *There are at most* $\left(2 \cdot 2^{-(1/3)cl} \cdot R\right)$ *good random strings.*

PROOF. Intuitively, any $(r, \overline{a_r})$ which has more than one neighbors in S corresponds to an agreement of at least two provers - and therefore should be a rare event. In order to turn this intuition into a proof, we must rule out solutions that distribute the budget in an uneven manner that does not correspond to answers of honest provers to verifier's questions.

In expectation, for each (q, i) there is only one $(q, a, i) \in S$. Therefore by Markov's inequality, for at most a $2^{-(1/3)cl}$-fraction of (q, i)'s, more than $2^{(1/3)cl}$ corresponding nodes belong to S; we

call those (q,i)'s *heavy*, and *light* otherwise, i.e.,

$$\Pr_r[\exists i : \ (q,i) \text{ is heavy}] \leq 2^{-(1/3)cl} \cdot k. \tag{1}$$

We henceforth focus on bounding the number of good random strings that correspond only to light (q,i)'s.

Consider only r's whose (q,i)'s are light. For each light (q,i), there are at most $2^{(1/3)cl}$ nodes (q,a,i) in S. In other words, each prover submits at most $2^{(1/3)cl}$ answers to each question. By Theorem 4.2, if each prover submits only one answer to each question, the fraction of random strings for which at least one pair agrees is at most 2^{-cl}; having $2^{(1/3)cl}$ answers, the probability that any pair agrees increases by at most $2^{(2/3)cl}$. Therefore at most a $2^{-(1/3)cl}$-fraction of random strings have at least one pair of agreeing answers. Recall that a random string r is *good* if for some $\overline{a_r}$, the node $(r,\overline{a_r})$ has more than one neighbor (q,a,i) in S.

$$\Pr_r[r \text{ is good and } \forall i : (q,i) \text{ is light}] \leq 2^{-(1/3)cl}. \tag{2}$$

Summing with (1), we have that:

$$\Pr_r[r \text{ is good}] \leq 2 \cdot 2^{-(1/3)cl}. \tag{3}$$

□

5. ALGORITHM FOR CONSTANT RANK WEIGHTED BI-ADJACENCY MATRIX M

The previous sections show that the AIM problem for general bipartite graph B and social graph G is hard to approximate to within any constant factor. In this section, we restrict the (weighted) bi-adjacency matrix M between content provider nodes and consumer nodes to be of constant rank r, and show that for this case we can obtain a constant factor approximation in polynomial time. We denote this restricted problem AIM-r. Our main algorithmic result is:

THEOREM 5.1. *For any constant $r > 0$ and $\delta, \varepsilon > 0$,* AIM-r *can be approximated to within $(1-1/\mathrm{e}-\varepsilon)^3$ with probability $1-\delta$ and in time polynomial in $n, m, \lambda, 1/\varepsilon, \log(1/\delta)$, where $n = |U|$, $m = |V|$ and λ is the maximum number of bits in any entries of matrix M.* [5]

For any fixed X, $\sigma(X,Y)$ is a monotone submodular function of Y. Similarly, for any fixed Y, $\sigma(X,Y)$ is a monotone submodular function of X. Each of those can be (approximately) optimized independently, thus the main algorithmic challenge is due to the interaction between the choice of X and the choice of Y. Intuitively, a constant rank bi-adjacency matrix creates an "information bottleneck" which restricts the complexity of this interaction.

How can we use the restriction on the matrix rank to optimize a non-linear objective? To this end, we introduce in Subsection 5.2 a relaxation of our objective function which conveniently views M as a linear operator acting on X. Because M has constant rank, the resulting subspace has a constant dimension; in Subsection 5.3, we show that we can efficiently (approximately) enumerate over all the points in this subspace. Finally, given the (approximately) optimal choice of X, we can use standard submodular maximization techniques to (approximately) optimize over Y (Subsection 5.4).

[5] The running time is exponential in r. See Section 5.5 for more details.

5.1. Notation

Henceforth, we use the following notational conventions. For vector $\mathbf{x}$, x_i is the i-th element of $\mathbf{x}$. All vectors are column vectors (unless otherwise stated). Let $|U| = n$ and $|V| = m$. When the context is clear, we also use the index set $[n] = \{1, 2, \ldots, n\}$ to represent U and index set $[m] = \{1, 2, \ldots, m\}$ to represent V.

Given the provider seed set $X \subseteq U$ and the consumer seed set $Y \subseteq V$, for convenience we denote $\mathbf{x}, \mathbf{y}$ as the indicator vectors of X and Y, respectively. An indicator vector for a subset X of U is a vector in $\{0, 1\}^n$ such that the entries corresponding to nodes in X are 1's and nodes in $U \setminus X$ are 0's, and indicator vector for subset Y of V is defined similarly.

Given $\mathbf{x}$ and $\mathbf{y}$, we use $f_j(\mathbf{x}, \mathbf{y})$ to denote the *initial activation probability* of each node $j \in V$, which is the probability that some node $i \in X$ activates j based on matrix M, i.e., $f_j(\mathbf{x}, \mathbf{y}) = y_j \left(1 - \prod_{i \in [n]} (1 - x_i M_{ij})\right)$. We denote $\mathbf{f}(\mathbf{x}, \mathbf{y})$ as the vector $(f_1(\mathbf{x}, \mathbf{y}), \ldots, f_m(\mathbf{x}, \mathbf{y}))^\top$. Note that the initial activation of nodes in Y from nodes in X are mutually independent for every node in Y. Moreover, the initial activation probability of node $j \in V$ is not its *final activation probability*, which is the probability that node j is activated by the end of the diffusion process, since j may be later activated by other nodes in V through the diffusion process in the social graph G. In particular, a node $j \in V \setminus Y$ has zero initial activation probability by our model definition, but its final activation probability may be greater than zero.

5.2. A concave relaxation

A key step in our algorithm is to approximate every coordinate $f_j(\mathbf{x}, \mathbf{y})$ via the following concave relaxation[6]

$$F_j(\mathbf{x}, \mathbf{y}) = y_j \left(1 - \mathrm{e}^{-\left(\mathbf{x}^\top M\right)_j}\right).$$

Notice that this relaxation has two important features: (a) it is a function of the linear form $\left(\mathbf{x}^\top M\right)_j$, which allows us to use the constant rank condition; and (b) it is both concave in $\mathbf{x}$ for a fixed $\mathbf{y}$, and concave in $\mathbf{y}$ for a fixed $\mathbf{x}$ — this will make it much easier to maximize efficiently. Now we will show that it is a $(1 - 1/\mathrm{e})$-approximation of $f_j(\mathbf{x}, \mathbf{y})$ by the following lemma.

LEMMA 5.2. *For any* $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$,

$$(1 - 1/\mathrm{e})\, f_j(\mathbf{x}, \mathbf{y}) \leq F_j(\mathbf{x}, \mathbf{y}) \leq f_j(\mathbf{x}, \mathbf{y}), \forall j = 1, \ldots, m.$$

PROOF. For the right inequality, since $\mathrm{e}^{-a} \geq 1 - a$ for any real a, we get

$$F_j(\mathbf{x}, \mathbf{y}) = y_j \left(1 - \mathrm{e}^{-\sum_{i \in [n]} x_i M_{ij}}\right) \leq y_j \left(1 - \prod_{i \in [n]} (1 - x_i M_{ij})\right) = f_j(\mathbf{x}, \mathbf{y}).$$

[6] The relaxation is inspired by [Badanidiyuru et al. 2014]. Essentially the same relaxation was also used before by [Dughmi et al. 2011] in the context of Poisson rounding.

For the left inequality, because $(1-\mathrm{e}^{-1})a \le 1-\mathrm{e}^{-a}$ holds for any $a \in [0,1]$, we have

$$\begin{aligned}
\left(1-\mathrm{e}^{-1}\right)\left(1-\prod_{i=1}^{n}\left(1-x_i M_{ij}\right)\right) &= \sum_{i=1}^{n}\left(\left(1-\mathrm{e}^{-1}\right)x_i M_{ij}\right)\prod_{k=1}^{i-1}\left(1-x_k M_{kj}\right) \\
&\le \sum_{i=1}^{n}\left(1-\mathrm{e}^{-x_i M_{ij}}\right)\cdot\prod_{k=1}^{i-1}\mathrm{e}^{-x_k M_{kj}} \\
&= \sum_{i=1}^{n}\left(\exp\left(-\sum_{k=1}^{i-1}x_k M_{kj}\right)-\exp\left(-\sum_{k=1}^{i}x_k M_{kj}\right)\right) \\
&= 1-\mathrm{e}^{-\sum_{i\in[n]}x_i M_{ij}} \\
&= 1-\mathrm{e}^{-\left(\mathbf{x}^\top M\right)_j}.
\end{aligned}$$

Multiplying y_j on both sides of the above inequality, we get $(1-1/\mathrm{e})\, f_j(\mathbf{x},\mathbf{y}) \le F_j(\mathbf{x},\mathbf{y})$. □

5.3. Approximating initial activation probability $\mathbf{f}(\mathbf{x},\mathbf{y})$ via $(1+\varepsilon)$-net construction

Notice that the value of $F_j(\mathbf{x},\mathbf{y})$ is uniquely determined by $\mathbf{x}^\top M$ and $\mathbf{y}$. We use $\operatorname{Im} M$ to denote the image of M when M is treated as a linear operator from $\{0,1\}^n$ to $\mathbb{R}^m$, i.e. $\operatorname{Im} M$ is the subspace of all vectors $\mathbf{x}^\top M \in \mathbb{R}^m$, $\forall \mathbf{x} \in \{0,1\}^n$. Recall that since M has constant rank r, the dimension of $\operatorname{Im} M$ is also r.

Our next goal is to enumerate over (approximately) all feasible $\mathbf{s} \in \operatorname{Im} M$. For any $\varepsilon > 0$, we say that set $\mathcal{S}_\varepsilon \subseteq \mathbb{R}^m$ is a *multiplicative-$(1+\varepsilon)$-net* for M, if for every $\mathbf{x} \in \{0,1\}^n$, there exists a corresponding point $\mathbf{s} = (s_1, s_2, \cdots, s_m)^\top \in \mathcal{S}_\varepsilon$ such that for each coordinate $j \in [m]$,

$$s_j \le \left(\mathbf{x}^\top M\right)_j \le s_j(1+\varepsilon). \tag{4}$$

Henceforth we drop the multiplicative qualification and simply call such a set a $(1+\varepsilon)$-net.

LEMMA 5.3. *Let $M \in [0,1]^{n\times m}$ be a matrix with constant rank r, and whose entries can be represented with λ bits. Then for any error parameter $\varepsilon > 0$, we can output a polynomial-size $(1+\varepsilon)$-net for M in time* $\mathrm{poly}\left(n, m, 1/\varepsilon, \lambda\right)$.

PROOF. Below, we show how to construct a *weak $(1+\varepsilon)$-net* (in Algorithm 1), which instead of Equation (4) gives the following weaker, two-sided error guarantee:

$$s_j/(1+\varepsilon) \le \left(\mathbf{x}^\top M\right)_j \le s_j(1+\varepsilon). \tag{5}$$

Given such an algorithm, one can construct a $(1+\varepsilon)$-net (with one-sided error) by constructing a weak $\sqrt{1+\varepsilon}$-net, and dividing every entry in the obtained weak $\sqrt{1+\varepsilon}$-net by $\sqrt{1+\varepsilon}$.

Since each entry of $M \in [0,1]^{n\times m}$ has at most λ bits, then for any $\mathbf{x} \in \{0,1\}^n$, every nonzero entry of $\mathbf{x}^\top M$ is bounded in $[2^{-\lambda}, n]$.[7] Thus in each dimension, we lose no more than a $(1+\varepsilon)$-factor by only considering each position in $S_\lambda = \left\{0, 2^{-\lambda}, 2^{-\lambda}\cdot(1+\varepsilon), 2^{-\lambda}\cdot(1+\varepsilon)^2, \ldots, n\right\}$.

[7] This step assumes of course that all the entries in M are positive. While this is a natural for an adjacency matrix of a social network, when taking the low-rank approximation of one, this may no longer be true. Nevertheless, a similar analysis continues to hold even when M has negative entries.

We consider the partitioning $[0,n]^m$ into hyper-rectangles that is induced by $\underbrace{S_\lambda \times \cdots \times S_\lambda}_{m \text{ times}}$. More precisely, the set of hyper-rectangles $\mathcal{H}$ is every possible direct product of intervals, i.e.,

$$\mathcal{H} = \left\{ \begin{array}{c} H = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_m, b_m] : \\ \forall i \in [m], \forall a_i \in S_\lambda \setminus \{n\}, b_i = \min\{\max\{(1+\varepsilon)a_i, 2^{-\lambda}\}, n\} \end{array} \right\}. \tag{6}$$

Observe that the disjoint union of those hyper-rectangles covers $[0,n]^m$; in particular, for any $\mathbf{x} \in \{0,1\}^n$, $\mathbf{x}^\top M$ must belong to a hyper-rectangle $H \in \mathcal{H}$.

Notice that if $\mathbf{x}^\top M$ and $\mathbf{s}$ lie in the same hyper-rectangle H, then they satisfy the two-sided error approximation guarantee of Equation (5). Our weak $(1+\varepsilon)$-net has at least one such $\mathbf{s}$ for every $\mathbf{x}^\top M$.

Consider any hyper-rectangle $H \in \mathcal{H}$ with a non-empty intersection with $\operatorname{Im} M$, and let $I_H = H \cap \operatorname{Im} M$ denote their intersection. Since I_H is an intersection of convex polytopes, it is also a convex polytope, defined by $m - r$ linearly independent equations that define $\operatorname{Im} M$ and $2m$ inequalities that define H. Every vertex $\mathbf{v}$ of this polytope must lie on the intersection of m linearly independent constraints: $m - r$ equations and r inequalities. In other words, $\mathbf{v}$ lies in the intersection of at least r facets of H; i.e. there exist $i_1, \ldots, i_r \in [m]$ such that $v_{i_k} \in \{a_{i_k}, b_{i_k}\} \ \ \forall k \in [r]$. Furthermore, these r coordinates must correspond to linearly independent columns of M.

Therefore, the following polynomial-time procedure (summarized in Algorithm 1) correctly construct a weak $(1+\varepsilon)$-net: First we take a submatrix M' of M with r linearly independent rows, so $\operatorname{Im} M' = \operatorname{Im} M$ (Line 3). Next enumerate over all $\binom{m}{r}$ r-tuples of linearly independent columns of M' (Line 4). For each r-tuple, enumerate over all vectors in $\underbrace{S_\lambda \times \cdots \times S_\lambda}_{r \text{ times}}$ (Line 5). Each such vector uniquely defines a point in $\mathbf{s} \in H \cap \operatorname{Im} M$ (Line 7). Finally, adding it to $\mathcal{S}_\varepsilon$ guarantees that we can approximate any other $\mathbf{x}^\top M \in H$ (Line 7). $\square$

We define

$$\hat{F}_j(\mathbf{s}, \mathbf{y}) = y_j \left(1 - \mathrm{e}^{-s_j}\right), \tag{7}$$

thus $\hat{F}_j(\mathbf{s}, \mathbf{y}) = F_j(\mathbf{x}, \mathbf{y})$ for all $\mathbf{s} = \mathbf{x}^\top M$. Notice that the definition of $\hat{F}_j(\mathbf{s}, \mathbf{y})$ naturally extends also to $\mathbf{s} \notin \operatorname{Im} M$. In the following lemma we relate $\hat{F}_j$ to the original f_j.

LEMMA 5.4. *Let $\varepsilon > 0$, and let $\mathcal{S}_\varepsilon$ be a $(1+\varepsilon)$-net for M. Then for every $\mathbf{x} \in \{0,1\}^n$, there exists an $\mathbf{s} \in \mathcal{S}_\varepsilon$ such that for every $\mathbf{y} \in \{0,1\}^m$ and $j \in [m]$,*

$$(1 - 1/\mathrm{e} - \varepsilon) f_j(\mathbf{x}, \mathbf{y}) \leq \hat{F}_j(\mathbf{s}, \mathbf{y}) \leq f_j(\mathbf{x}, \mathbf{y}).$$

PROOF. By Lemma 5.3, for any $\mathbf{x} \in \{0,1\}^n$, we know that there exists $\mathbf{s} \in \mathcal{S}_\varepsilon$ satisfying Inequality (4). Moreover, for any $\mathbf{y}$, we claim that

$$\hat{F}_j(\mathbf{s}, \mathbf{y}) \geq F_j(\mathbf{x}, \mathbf{y}) / (1+\varepsilon) \geq (1 - 1/\mathrm{e} - \varepsilon) \, f_j(\mathbf{x}, \mathbf{y}), \tag{8}$$

where the second inequality is due to Lemma 5.2 and the fact that $(1-1/\mathrm{e})/(1+\varepsilon) \geq (1-1/\mathrm{e}-\varepsilon)$ for all $\varepsilon > 0$. Recall that $F_j(\mathbf{x}, \mathbf{y}) = 1 - \mathrm{e}^{-y_j \cdot (\mathbf{x}^\top M)_j}$, thus to show the first inequality of Eq. (8),

ALGORITHM 1: Construct a weak $(1+\varepsilon)$-net over the image of M

Input: The matrix M of rank r, and the error factor ε
Output: The multiplicative $(1+\varepsilon)$-net $\mathcal{S}_\varepsilon$.

Let λ be the maximum number of bits in any entry of M, and $S_\lambda = \left\{0, 2^{-\lambda}, 2^{-\lambda}\cdot(1+\varepsilon), 2^{-\lambda}\cdot(1+\varepsilon)^2, \ldots, n\right\}$
$\mathcal{S}_\varepsilon \leftarrow \emptyset$
Let M' be the $r \times m$ submatrix of M with r independent row vectors $\mathbf{v}_1, \ldots, \mathbf{v}_r \in [0,1]^m$
for $\left(\substack{i_1, i_2, \ldots, i_r \in [m] \text{ s.t. } i_1 < i_2 < \cdots < i_r \\ \text{and the corresponding columns of } M' \text{ are independent}}\right)$ **do**
  /* Enumerate every r coordinates according to the grid */
  for $k_1, \ldots, k_r \in S_\lambda$ **do**
    Construct linear system of r equations: $\begin{cases} (\hat{z}_1\mathbf{v}_1 + \cdots + \hat{z}_r\mathbf{v}_r)_{i_1} = & k_1 \\ (\hat{z}_1\mathbf{v}_1 + \cdots + \hat{z}_r\mathbf{v}_r)_{i_2} = & k_2 \\ & \vdots \\ (\hat{z}_1\mathbf{v}_1 + \cdots + \hat{z}_r\mathbf{v}_r)_{i_r} = & k_r \end{cases}$
    /* $\hat{z}_1\mathbf{v}_1 + \cdots + \hat{z}_r\mathbf{v}_r$ is an m-dimensional vector, and denote $(\hat{z}_1\mathbf{v}_1 + \cdots + \hat{z}_r\mathbf{v}_r)_i$ as its i-th coordinate for each $i \in [m]$ */
    Use Gaussian elimination to derive the solution $(\hat{z}'_1, \ldots, \hat{z}'_r)^\top \in \mathbb{R}^r$
    $\mathbf{s} \leftarrow \hat{z}'_1\mathbf{v}_1 + \cdots + \hat{z}'_r\mathbf{v}_r$; $\mathcal{S}_\varepsilon \leftarrow \mathcal{S}_\varepsilon \cup \{\mathbf{s}\}$
  end
end
return $\mathcal{S}_\varepsilon$

we only need to show that,

$$1 - \mathrm{e}^{-s_j y_j} \geq \left(1 - \mathrm{e}^{-y_j \cdot \left(\mathbf{x}^\top M\right)_j}\right) / (1+\varepsilon).$$

For $y_j = 0$, it is trivial. For $y_j = 1$, since $(1+\varepsilon)s_j \geq \left(\mathbf{x}^\top M\right)_j$, we have $\mathrm{e}^{-(1+\varepsilon)s_j} \leq \mathrm{e}^{-\left(\mathbf{x}^\top M\right)_j}$. Therefore, it is enough to show that $(1-\mathrm{e}^{-s_j}) \geq \left(1-\mathrm{e}^{-(1+\varepsilon)s_j}\right)/(1+\varepsilon)$, namely,

$$\frac{\varepsilon + \mathrm{e}^{-(1+\varepsilon)s_j}}{1+\varepsilon} \geq \mathrm{e}^{-s_j}.$$

Note that the above inequality holds due to Weighted AM-GM inequality $\frac{ax+by}{a+b} \geq x^{\frac{a}{a+b}} y^{\frac{b}{a+b}}$, $\forall x, y, a, b \in \mathbb{R}^+$ by letting $a = \varepsilon, x = 1, b = 1, y = \mathrm{e}^{-(1+\varepsilon)s_j}$.

Furthermore, we claim that

$$f_j(\mathbf{x}, \mathbf{y}) \geq F_j(\mathbf{x}, \mathbf{y}) \geq \hat{F}_j(\mathbf{s}, \mathbf{y}),$$

where the first inequality is due to Lemma 5.2, and the second one follows from Equation 4. □

Analogously to $\hat{F}_j(\mathbf{s}, \mathbf{y})$, we can also define $\hat{\sigma}(\mathbf{s}, \mathbf{y})$ to be the expected number of (eventually) activated consumer nodes given that the set of *initially* activated consumer nodes is distributed according to $\hat{\mathbf{F}}(\mathbf{s}, \mathbf{y})$. (I.e. each node $j \in V$ is independently initially activated with probability $\hat{F}_j(\mathbf{s}, \mathbf{y})$.) For any fixed $\mathbf{s}$, by the definition of $\hat{F}_j(\mathbf{s}, \mathbf{y})$ in Eq. (7), we know that $\hat{\sigma}(\mathbf{s}, \mathbf{y})$ is equivalent to the influence spread obtained by selecting seed set Y (indiated by $\mathbf{y}$), each seed

$j \in Y$ being activated with probability $1 - \mathrm{e}^{-s_j}$ and then influence propagated in the social graph G. Then, by the result in [Kempe et al. 2003], it is straightforward to see that $\hat{\sigma}(\mathbf{s}, \mathbf{y})$ is monotone and submodular on $\mathbf{y}$. [8]

Furthermore, our $(1+\varepsilon)$-net continues to (approximately) capture all possible inputs to $\hat{\sigma}$:

LEMMA 5.5. *Let $\varepsilon > 0$, and let $\mathcal{S}_\varepsilon$ be a $(1+\varepsilon)$-net for M. Then for every $\mathbf{x} \in \{0,1\}^n$, there exists an $\mathbf{s} \in \mathcal{S}_\varepsilon$ such that for every $\mathbf{y} \in \{0,1\}^m$,*

$$(1 - 1/\mathrm{e} - \varepsilon)\sigma(\mathbf{x}, \mathbf{y}) \leq \hat{\sigma}(\mathbf{s}, \mathbf{y}) \leq \sigma(\mathbf{x}, \mathbf{y}).$$

PROOF. The right inequality follows immediately from Lemma 5.4.

To prove the left inequality, we unfortunately need to define yet another function. For $\mathbf{z} \in \{0,1\}^m$, let $\rho(\mathbf{z})$ be the expected number of activated nodes given that the set of *initially* activated nodes is $Z \subseteq Y$ indicated by $\mathbf{z}$. Notice that $\rho(\mathbf{z})$ is also submodular [Kempe et al. 2003].

For a fractional $\bar{\mathbf{z}} \in [0,1]^m$, we extend $\bar{\rho}(\bar{\mathbf{z}})$ to be the expectation over integral $\mathbf{z} \in \{0,1\}^m$, where each coordinate is sampled independently with expectation $\bar{z}_j$. (I.e. $\bar{\rho}(\bar{\mathbf{z}}) = \sum_{\mathbf{z} \in \{0,1\}^m} \left[\rho(\mathbf{z}) \cdot \prod \left(\bar{z}_i^{z_i} \cdot (1-\bar{z}_i)^{1-z_i}\right)\right]$.) Thus, $\sigma(\mathbf{x}, \mathbf{y}) = \bar{\rho}\left(\mathbf{f}(\mathbf{x}, \mathbf{y})\right)$ and $\hat{\sigma}(\mathbf{s}, \mathbf{y}) = \bar{\rho}\left(\hat{\mathbf{F}}(\mathbf{s}, \mathbf{y})\right)$.

Finally, since $\bar{\rho}(\mathbf{z})$ is the multilinear extension of a submodular function, we have (e.g. by Lemma 2.2 of [Vondrák 2007]) that

$$(1 - 1/\mathrm{e} - \varepsilon)\bar{\rho}\left(\mathbf{f}(\mathbf{x}, \mathbf{y})\right) \leq \bar{\rho}\Big((1 - 1/\mathrm{e} - \varepsilon) \cdot \mathbf{f}(\mathbf{x}, \mathbf{y})\Big) \leq \bar{\rho}\left(\hat{\mathbf{F}}(\mathbf{s}, \mathbf{y})\right).$$

□

5.4. From $(1+\varepsilon)$-net to approximation algorithm

Armed with our $(1+\varepsilon)$-net, we can use standard submodular maximization techniques to approximately solve AIM-r. The full algorithm is summarized in Algorithm 2, and referred as Sampled Double Greedy (SDG) algorithm.

For every $\mathbf{s} \in \mathcal{S}_\varepsilon$, let $\mathbf{y}_\mathbf{s}$ be a $(1 - 1/\mathrm{e} - \varepsilon)$-approximation to the feasible vector $\mathbf{y}^*_\mathbf{s}$ that maximizes $\hat{\sigma}(\mathbf{s}, \mathbf{y})$. Recall that such an approximation can be found in polynomial time (with high probability) via standard submodular maximization techniques (see e.g. [Kempe et al. 2003] as well as Subsection 5.5). Let $(\mathbf{x}^*, \mathbf{y}^*)$ be the optimal feasible solution to the AIM-r problem. Then for some $\mathbf{s}^*$, we have that for every $\mathbf{y} \in \{0,1\}^m$:

$$(1 - 1/\mathrm{e} - \varepsilon)\sigma(\mathbf{x}^*, \mathbf{y}) \leq \hat{\sigma}(\mathbf{s}^*, \mathbf{y}) \leq \sigma(\mathbf{x}^*, \mathbf{y});$$

and in particular,

$$\hat{\sigma}(\mathbf{s}^*, \mathbf{y}_{\mathbf{s}^*}) \geq (1 - 1/\mathrm{e} - \varepsilon)\hat{\sigma}(\mathbf{s}^*, \mathbf{y}^*) \geq (1 - 1/\mathrm{e} - \varepsilon)^2 \sigma(\mathbf{x}^*, \mathbf{y}^*). \tag{9}$$

For each fixed $\mathbf{y}_\mathbf{s}$, function $\sigma(\mathbf{x}, \mathbf{y}_\mathbf{s})$ is monotone and submodular on $\mathbf{x}$. This is because we can remove all edges from U to nodes not in the set indicated by $\mathbf{y}_\mathbf{s}$, and $\sigma(\mathbf{x}, \mathbf{y}_\mathbf{s})$ is the influence spread of seed set X in the combined bipartite graph and social graph after removing those

[8] For fixed $\mathbf{s}$, when we say that a vector function $\hat{\sigma}(\mathbf{s}, \mathbf{y})$ is monotone and submodular on the indicator vector $\mathbf{y}$, we mean that $\hat{\sigma}(\mathbf{s}, Y) \triangleq \hat{\sigma}(\mathbf{s}, \mathbf{y})$ is monotone and submodular on set Y indicated by $\mathbf{y}$.

ALGORITHM 2: SDG: A constant-factor approximation to AIM-r

Input: Bi-adjacency matrix M, adjacency matrix P, budgets b_1, b_2, accuracy parameter $\varepsilon > 0$
Output: Subsets (X, Y) that approximate the optimal $\sigma(X, Y)$

Construct $(1+\varepsilon)$-net $\mathcal{S}_\varepsilon$ from M by Algorithm 1
for $\mathbf{s} \in \mathcal{S}_\varepsilon$ **do**
  Use greedy algorithm to find solution $\mathbf{y_s}$ on submodular function $\hat{\sigma}(\mathbf{s}, \cdot)$ with budget b_2
  Use greedy algorithm to find solution $\mathbf{x_{y_s}}$ on submodular function $\sigma(\cdot, \mathbf{y_s})$ with budget b_1
end
return $\arg\max_{(\mathbf{x_{y_s}}, \mathbf{y_s}):\mathbf{s}\in\mathcal{S}_\varepsilon} \sigma(\mathbf{x_{y_s}}, \mathbf{y_s})$

edges. Since diffusion from X in this subgraph can be viewed as IC model diffusion, by [Kempe et al. 2003] we know that $\sigma(\mathbf{x}, \mathbf{y_s})$ is monotone and submodular on $\mathbf{x}$. Then we can take $\mathbf{x_s}$ to be a $(1 - 1/\mathrm{e} - \varepsilon)$-approximation to the feasible vector $\mathbf{x}^*_\mathbf{s}$ that maximizes $\sigma(\mathbf{x}, \mathbf{y_s})$. Thus, for $\mathbf{y}_{s^*}$, we have

$$\sigma(\mathbf{x}_{s^*}, \mathbf{y}_{s^*}) \geq (1 - 1/\mathrm{e} - \varepsilon)\sigma(\mathbf{x}^*, \mathbf{y}_{s^*}) \geq (1 - 1/\mathrm{e} - \varepsilon)\hat{\sigma}(\mathbf{s}^*, \mathbf{y}_{s^*}). \tag{10}$$

Taking (9) and (10) together, we have

$$\sigma(\mathbf{x}_{s^*}, \mathbf{y}_{s^*}) \geq (1 - 1/\mathrm{e} - \varepsilon)^3\sigma(\mathbf{x}^*, \mathbf{y}^*).$$

This completes the proof of Theorem 5.1. □

5.5. Running time

We have completely ignored the question of how to compute $\sigma(\mathbf{x}, \mathbf{y})$ and its relaxation $\hat{\sigma}(\mathbf{s}, \mathbf{y})$. In particular, this computation is necessary for the submodular maximization procedures used in Algorithm 2. Although their exact computations are #P-hard [Chen et al. 2010], both can be efficiently approximated with arbitrarily good precision by sampling from the corresponding random processes. In particular, for the independent cascade model, both greedy steps in Algorithm 2 can apply the near-linear time algorithms in [Borgs et al. 2014; Tang et al. 2014].

The total running time of SDG is $O(m^r(\lambda+\log n)^r(b_1+b_2)(n+m+\ell)\varepsilon^{-(r+2)}\log(1/\delta)))$, where $O(m^r(\lambda + \log n)^r\varepsilon^{-r})$ is the size of $(1 + \varepsilon)$-net S_ε, and $O(\varepsilon^{-2}(b_1 + b_2)(n + m + \ell)\log(1/\delta))$ is the time running two greedy steps using the algorithm in [Tang et al. 2014], with ℓ being the total number of edges in B and G. In some situations r could be very small, e.g. $r = 1$ when the influence probability from provider i to consumer j can be approximated as the product of provider i's influence strength and consumer j's susceptibility. Moreover, in practice seed consumers may be selected from a candidate set of size m' (e.g. the fan base of a product) much smaller than the social network size ($m' << m$), then the dominant term m^r would be replaced by the much smaller $(m')^r$. Therefore, SDG would be efficient in these practical situations.

6. GENERAL DIFFUSION MODEL

Our results can be generalized to support any diffusion model in the social network G that has a monotone and submodular influence spread function. More specifically, for any subset $Z \subseteq V$ in the social graph G, let $\rho(Z)$ be the influence spread of Z in G, that is, the expected number of activated nodes in G after the diffusion process when Z is selected as the initial seed set. The diffusion model in the combined bipartite graph B and social graph G with

selected seed providers X and seed consumers Y is as follows: First, all seed providers in X are activated; and then following the probabilities given in the bi-adjacency matrix M, a subset of seed consumers, $Z \subseteq Y$, is activated (each $j \in Y$ is activated independently with probability $\mathbf{f}(\mathbf{x}, \mathbf{y})$ as before). Then the diffusion from Z follows the social graph diffusion model, with expected spread $\rho(Z)$. We still use notation $\sigma(X, Y)$ to represent the influence spread of X and Y in the combined network. Then we have $\sigma(X, Y) = \sum_{Z \subseteq Y} \Pr_X[Z]\rho(Z)$, where $\Pr_X(Z)$ is the probability that Z is the initially activated set in Y by provider seed set X according to the matrix M.

One particular instantiation of interest is that of *background propagation*: each consumer has some initial ("background") probability of being influenced by content providers even without being seeded by the advertiser; if the advertiser seeds the consumer, she has a (higher) *boosted probability* of being influenced by the content providers. To implement this model as a submodular influence function, we can sample the result of the diffusion process from the background probabilities and incorporate the expected outcome into the definition of $\rho(\cdot)$.

In general, we can show that as long as $\rho(\cdot)$ is monotone, submodular, and polynomial-time computable, and matrix M is of constant rank as assumed before, AIM problem is still solvable in polynomial time. The main revision of the proof is to show that in the general model $\sigma(X, Y)$ is still monotone and submodular when we fix either X or Y. See the full report [Chen et al. 2015] for details.

7. CONCLUSION

In this paper we propose the amphibious influence maximization (AIM) model as a proxy framework that combines traditional marketing via content providers together with viral marketing to consumers in social networks. We show that the associated computational problem is NP-hard to approximate to any constant factor, and provide a polynomial-time algorithm with $(1 - 1/\mathrm{e} - \varepsilon)^3$ approximation ratio for any (polynomially small) $\varepsilon > 0$ when we restrict the weighted bi-adjacency matrix M for the provider-consumer network to be of constant rank.

It would be interesting to see to what extent *amphibious marketing* (i.e. targeting individual users via a combination of traditional content providers and social network viral marketing) can be implemented in practice. Beyond the algorithmic challenge of optimizing the sets of seed providers and consumers we discuss in this paper, this notion raises many interesting challenges in terms of learning the influence factors (the adjacency matrices in our model), privacy of the consumers, economic incentives, etc.

From the perspective of theoretical computer science, we view our algorithm for AIM with low rank assumption as part of the ongoing effort in the community to incorporate assumptions that are both reasonable in practice, and allow better algorithmic results. In this context we remark that although our low rank assumption is most natural in the context of content providers-consumers influence matrix, it is also closely related to another important property that has been observed in graphs of social networks: the eigenvalues exhibit a power law [Faloutsos et al. 1999; Mihail and Papadimitriou 2002].

References

ALON, N., ARORA, S., MANOKARAN, R., MOSHKOVITZ, D., AND WEINSTEIN, O. 2011. Inap-

proximabilty of densest k-subgraph from average case hardness. Manuscript.

ALON, N., LEE, T., SHRAIBMAN, A., AND VEMPALA, S. 2013. The approximate rank of a matrix and its algorithmic applications: approximate rank. In *STOC'13*. 675–684.

ALON, N. AND SPENCER, J. 1992. *The Probabilistic Method*. John Wiley.

BADANIDIYURU, A., PAPADIMITRIOU, C., RUBINSTEIN, A., SEEMAN, L., AND SINGER, Y. 2014. A $(1 - 1/e)^2$-approximation for adaptive seeding of monotone submodular functions. Manuscript.

BORGS, C., BRAUTBAR, M., CHAYES, J., AND LUCIER, B. 2014. Maximizing social influence in nearly optimal time. In *SODA'14*.

CHEN, W., LAKSHMANAN, L. V., AND CASTILLO, C. 2013. *Information and Influence Propagation in Social Networks*. Morgan & Claypool.

CHEN, W., LI, F., LIN, T., AND RUBINSTEIN, A. 2015. Combining traditional marketing and viral marketing with amphibious influence maximization. Tech. rep., arXiv. to appear.

CHEN, W., WANG, C., AND WANG, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD'10*. ACM, 1029–1038.

DOMINGOS, P. AND RICHARDSON, M. 2001. Mining the network value of customers. In *KDD'01*. ACM, 57–66.

DUGHMI, S., ROUGHGARDEN, T., AND YAN, Q. 2011. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *STOC'11*. STOC '11. ACM, New York, NY, USA, 149–158.

FALOUTSOS, M., FALOUTSOS, P., AND FALOUTSOS, C. 1999. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev. 29,* 4, 251–262.

FEIGE, U. 1998. A threshold of $\ln n$ for approximating set cover. *J. ACM 45,* 4, 634–652.

KEMPE, D., KLEINBERG, J., AND TARDOS, É. 2003. Maximizing the spread of influence through a social network. In *KDD'03*. ACM, 137–146.

KOREN, Y., BELL, R. M., AND VOLINSKY, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer 42,* 8, 30–37.

MIHAIL, M. AND PAPADIMITRIOU, C. H. 2002. On the eigenvalue power law. In *RANDOM*. 254–262.

PAPAILIOPOULOS, D. S., MITLIAGKAS, I., DIMAKIS, A. G., AND CARAMANIS, C. 2014. Finding dense subgraphs via low-rank bilinear optimization. In *ICML'14*. 1890–1898.

RICHARDSON, M. AND DOMINGOS, P. 2002. Mining knowledge-sharing sites for viral marketing. In *KDD'02*. ACM, 61–70.

RUBINSTEIN, A., SEEMAN, L., AND SINGER, Y. 2015. Approximability and incentives in adaptive seeding. In *EC'15*. To appear.

SEEMAN, L. AND SINGER, Y. 2013. Adaptive seeding in social networks. In *FOCS'13*.

TANG, Y., XIAO, X., AND SHI, Y. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*.

VONDRÁK, J. 2007. Submodularity in combinatorial optimization. Ph.D. thesis, Citeseer.

YANG, D.-N., HUNG, H.-J., LEE, W.-C., AND CHEN, W. 2013. Maximizing acceptance probability for active friending in online social networks. In *KDD'13*.