Scaling Influence Maximization with Network Abstractions

Mahsa Maghami and Gita Sukthankar

Department of EECS
University of Central Florida
mmaghami@cs.ucf.edu,gitars@eecs.ucf.edu

Abstract. Maximizing product adoption within a customer social network under a constrained advertising budget is an important special case of the general influence maximization problem. Specialized optimization techniques that account for product correlations and community effects can outperform network-based techniques that do not model interactions that arise from marketing multiple products to the same consumer base. However, it can be infeasible to use exact optimization methods that utilize expensive matrix operations on larger networks without parallel computation techniques. In this chapter, we present a hierarchical influence maximization approach for product marketing that constructs an abstraction hierarchy for scaling optimization techniques to larger networks. An exact solution is computed on smaller partitions of the network, and a candidate set of influential nodes is propagated upward to an abstract representation of the original network that maintains distance information. This process of abstraction, solution, and propagation is repeated until the resulting abstract network is small enough to be solved exactly.

Keywords: Marketing, Optimization, Multi-agent social simulations

1 Introduction

Advertising in today's market is no longer viewed as a matter of simply convincing a potential customer to buy the product but of convincing their social network to adopt a lifestyle choice. It is well known that social ties between users play an important role in dictating their behavior. One of the ways this can occur is through social influence where a behavior or idea can propagate between friends. By considering factors such as homophily and possible unobserved confounding variables, it is possible to examine these behavior correlations in a social network statistically [1]. The aim of viral marketing strategies is to leverage these behavior correlations to create information cascades in which a large number of customers imitate a much smaller set of informed people, who are initially convinced by targeting marketing schemes.

Marketing with a limited budget can be viewed as a specialized version of the influence maximization problem in which the aim is to advertise to the optimal set of seed nodes to modify opinion in the network, based on a known influence propagation model. Commonly used propagation models such as LTM (Linear Threshold Model) and ICM (Independent Cascade Model) assume that a node's adoption probability is

conditioned on the opinions of the local network neighborhood [15]. Much of the previous influence maximization work [10, 8, 25] uses these two interaction models. Since the original LT model and IC model, other generalized models have been proposed for different domains and specialized applications. For instance, the decreasing cascade model generalizes models used in the sociology and economics communities where a behavior spreads in a cascading function according to a probabilistic rule, beginning with a set of nodes that adopt the behavior [15]. In contrast with the original IC model, in the decreasing cascade model the probability of influence propagation from an active node is not constant. Similarly, generalized versions of the linear threshold model have been introduced (e.g., [23], [5]). The simplicity of these propagation models facilitates theoretical analysis but does not realistically model specific marketing considerations such as the interactions between advertisements of multiple products and the effects of community membership on product adoption.

To address these problems, in previous work [21], we developed a model of product adoption in social networks that accounts for these factors, along with a convex optimization formulation for calculating the best marketing strategy assuming a limited budget. These social factors can emerge from different independent variables such as ties between friends and neighbors, social status, and the economic circumstance of the agents. Similar properties have been shown to influence people in other domains; for instance, Aral and Walker demonstrated the effect of social status on the influence factor of people on Facebook [3]. We believe that in marketing, all these factors affect the customers' susceptibility to influence and their ability to influence others.

Having a more realistic model is particularly useful for overcoming negative advertisement effects in which the customers refrain from purchasing any products after being bombarded with mildly derogatory advertisement from multiple advertisers trying to push their own products. It is critical to model the propagation of negative influence as well since it propagates and can be stronger and more contagious than positive influence in affecting people's decisions [7].

The main limitation of this and similar types of optimization approaches is that they involve matrix inversion which is slightly less than $O(N^3)$ and is the limiting factor preventing these algorithms from scaling to larger networks. In this chapter, we propose a hierarchical influence maximization approach that advocates "divide and conquer"—-the network is partitioned into multiple smaller networks that can be solved exactly with optimization techniques, assuming a generalized IC model, to identify a candidate set of seed nodes. The candidate nodes are used to create a distance-preserving abstract version of the network that maintains an aggregate influence model between partitions. Here we demonstrate how this abstraction technique can be used to scale influence maximization algorithms to larger product adoption scenarios. Moreover, we present a theorem which shows that the realistic social system model has a fixed-point, validating the strategy of optimizing product adoption at the steady state.

The chapter is organized as follows. Section 2 provides an overview of the related work in influence maximization. Section 3 introduces our proposed method, Hierarchical Influence Maximization (HIM) [22], as well as summarizing the operation of the realistic product adoption model introduced by [21]. We evaluate our method vs. other influence maximization approaches on both real and synthetic networks in Sec-

tion 4. This chapter extends on our earlier work [22] by introducing new preprocessing techniques for large networks and presenting a more comprehensive evaluation of our framework on three larger real-world datasets. We end the chapter with a discussion of future work.

2 Related Work

Influence maximization can be described as the problem of identifying a small set of nodes capable of triggering large behavior cascades that spread through the network. This set of nodes can be discovered using probabilistic approaches (e.g., [2, 17]) or optimization-based techniques. [12, 21] treat influence maximization as a convex optimization problem; this is feasible for influencing small communities but does not scale to larger scale problems. Due to the matrix computation requirements, these approaches fail when the number of agents in the system increases. Our HIM algorithm overcomes this deficiency by using a hierarchical approach to factor the system into smaller matrices.

The HIM model is designed to work on a complex social system where multiple factors affect the propagation of influence. The simpler case, where the network topology alone dictates activation spread, has been examined by multiple research groups, seeking to improve on Kempe's early work on greedy approaches for influence maximization [14]. Examples of possible speedups include innovations such as the use of a shortest-path based influence cascade model [16] or a lazy-forward optimization algorithm [19] to reduce the number of evaluations on the influence spread of nodes. Clever heuristics have been used very successfully to speed computation in both the LT model (e.g., the PMIA algorithm [8]) and also the IC model [25]. In this chapter, instead of using the original cascade models by Kempe et al. we introduce a cascade model that accounts for product interactions and community differences in influence propagation.

Proposed models for investigating how ideas and influence propagate through the network have been applied to many domains, including technology diffusion, strategy adoption in game-theoretic settings, and the admission of new products in the market [14]. For viral marketing, influential nodes can be identified either by following interaction data or probabilistic strategies. For example, Hartline et al. [11] solve a revenue maximization problem to investigate effective marketing strategies. [26] presented a targeted marketing method based on the interaction of subgroups in social network. Similar to this work, Bagherjeiran and Parekh leverage purchasing homophily in social networks [4]. But instead of finding influential nodes, they base their advertising strategy on the profile information of users. Achieving deep market penetration can be an important aspect of marketing; Shakarian and Damon present a viral marketing strategy for selecting the seed nodes that guarantees the spread of the word to the entire network [24]. Our work differs from related work in that our model not only considers social factors but also incorporates the negative effect of competing product advertisements and the correlation between demand for different products. Our optimization approach is largely unaffected by the additional complexity since these factors only impact the long-term expected value and not the actual solution method.

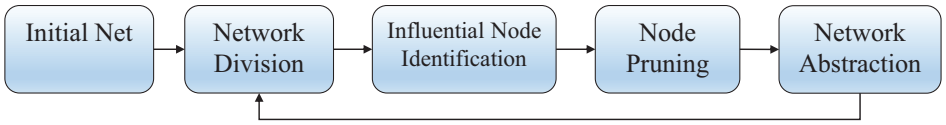

Fig. 1. The flowchart for our algorithm, Hierarchical Influence Maximization (HIM).

Some researchers (e.g., [20, 6]) focus on the adversarial aspect of competing against other advertisers. In this case, the assumption is that the advertiser is unable to unilaterally select nodes. In [5] a natural and mathematically tractable model is presented for the diffusion of multiple innovations in a network. Our work assumes that influential nodes are selected in a central fashion and partitioned between advertisers in an adversarial offline process.

3 Method

Our proposed hierarchical approach operates as follows:

1. Create a local network for each node consisting of its neighbors and neighbors of neighbors;
2. Model the effect of the outside network by assigning a virtual node for each boundary node to abstract activity outside the local partition;
3. Update the interaction parameters to the virtual node based on the model and the network connections;
4. Create a candidate set of influential nodes for each local network using convex optimization to maximize steady state product adoption;
5. Propagate the candidate set upward to a higher-level of abstraction and link the abstract nodes based on their shortest paths in the previous network;
6. Repeat the abstraction process until the resulting network is small enough to be optimized as a single partition; the resulting set of candidate nodes is then targeted for advertisement.

Figure 2 demonstrates the process of the algorithm with three hierarchies. The selected nodes at each local neighborhood, colored in red, are moved to the upper hierarchy and reconnected based on shortest path distances from the lower-level. The same process is repeated at the next hierarchy to select more influential nodes. The procedure terminates at the last hierarchy when the number of influential nodes finally is smaller than the advertising budget. Figure 1 shows a flowchart of the algorithm.

3.1 Market Model

To explore the efficiency of the proposed hierarchical influence maximization (HIM) method in business marketing, we have used the multi-agent system model, presented by [21], to simulate a social system of potential customers. We have slightly changed the definition of some parameters in this model to make a more sensible model with generalized capabilities.

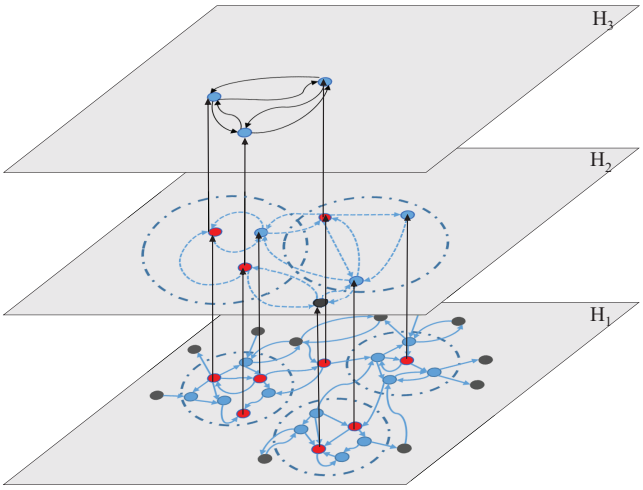

Fig. 2. At each hierarchical level (H_i) local neighborhoods are created and virtual nodes (black) are generated. By using an optimization technique the influential nodes (red) are selected. Nodes that have been selected at least once as an influential node are transferred to the next level of the hierarchy. At the higher levels, the connection between selected nodes is defined using the shortest path distance in the original network. The process is repeated until the final set of influential nodes is smaller than the total advertising budget.

In this model, the population of N agents, represented by the set $A = \{a_1, \ldots, a_N\}$, consists of two types of agents $(A = A_R \cup A_P)$, named *Regular* and *Product* agents respectively. The *Regular* agents are the potential customers in the market who will occasionally change their attitudes on purchasing products based on the influence they receive either from other neighbors or from the *Product* agents who represent salespeople offering one specific product.

Regular agents belong to a connected social network where the directed weighted links in this network possess a history of past interactions among the agents. This social network is modeled by an adjacency matrix, $\mathbf{E}$, where $e_{ij} = 1$ is the weight of a directed edge from agent a_i to agent a_j and the in-node and out-node degree of agent a_i is the sum of all in-node and out-node weights, respectively.

In this model a vector of $\overrightarrow{X_i}$ is assigned to each agent, both *Regular* and *Product* agents, representing the attitude or desire of the agent toward all of the products in the market. Each element of this vector, x_{ip}, is a random variable in the [-1 1] interval that indicates the desire of agent a_i to buy an item or consume a specific product, p.

In the social simulation, each agent interacts with another agent in a pair-wise fashion that is modeled as a Poisson process with rate 1, independent of all other agents. By assuming a Poisson process of interaction, we are claiming that there is at most one interaction at any given time. Here, the probability of interaction between agents a_i and a_j is shown by p_{ij} and is defined as a fraction of the connection weight between these

agents over the total connections that agent i makes with the other agents. Therefore,

$$p_{ij} = \begin{cases} \frac{e_{ij}}{d^i_{out}} & i,j \in A_R \\ \frac{u_{ji}}{Threshold} & i \in A_R, j \in A_P \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where the *Threshold* parameter is the total number of links that *Product* agent can make with *Regular* agents. The bounds on *Threshold* are a natural consequence of the limited budget of companies in advertising their products. The u_{ji} parameter is an indicator marking whether the *Product* agent is connected to the *Regular* agent.

At each interaction there is a chance for agents to influence each other and change their desire vector for purchasing or consuming a product. During these interactions the *Product* agents never change their attitude and maintain a fixed desire vector of 1 toward themselves and -1 toward the other advertising companies. The probability that agent i is susceptible to agent j is denoted as α_{ij} and calculated as:

$$\alpha_{ij} = \begin{cases} \frac{e_{ji}}{d^i_{in}} & i,j \in A_R \\ cte & i \in A_R, j \in A_P \end{cases} \tag{2}$$

The other important parameter in the agent influence process is ε_{ij}, which determines how much agent j will influence agent i. This parameter indicates the role of social factors in decision making of agents. In contrast to previous work, we did not restrict this parameter to a specific distribution to provide more flexibility to the model. Moreover, in real life there is a correlation between the user demand for different products in the market. The desire of customers for a specific product is related to his/her desire toward other similar products. Matrix $\mathbf{M}$ models this correlation, and we consider its effect in our formulation. The ultimate goal of our marketing problem is to recognize the influential agents in the graph and define a set of connections between the A_P agents and A_R agents, in such a way to maximize the long term desire of the agents for the products. Note that the links between *Product* agents and *Regular* agents are directed links from products to agents and not in the opposite direction.

3.2 Generalized ICM

We use a generalized version of ICM similar to [21, 13]. The dynamics of the model at each iteration k proceed as follows:

1. Agent i initiates the interaction according to a uniform probability distribution over all agents. Then agent i selects another agent among its neighbors with probability p_{ij}. Note that the desire dynamic can occur with probability $\frac{1}{N}(p_{ij} + p_{ji})$ as agent i's attitude can change whether it initiates the interaction or is selected by agent j.
2. Conditioned on the interaction of i and j:
   – With propagability α_{ij}, agent i will change its desire:

$$\begin{cases} \overrightarrow{X_i}(k+1) = \varepsilon_{ij}\, \mathbf{M}\overrightarrow{X_i}(k) + (1-\varepsilon_{ij})\, \mathbf{M}\overrightarrow{X_j}(k) \\ \overrightarrow{X_j}(k+1) = \overrightarrow{X_j}(k) \end{cases} \tag{3}$$

Recall that $\mathbf{M}$ is the pre-defined matrix indicating the correlation between the demands of different products.

– With probability of $(1 - \alpha_{ij})$, agent i is not influenced by the other agent:

$$\begin{cases} \overrightarrow{X_i}(k+1) = \overrightarrow{X_i}(k) \\ \overrightarrow{X_j}(k+1) = \overrightarrow{X_j}(k) \end{cases} \tag{4}$$

It is worthwhile to note that the above interaction model can be degraded to the IC model, if we set $\varepsilon_{ij} = 0$, $M = \mathbb{I}$, and restrict $p_{ij}s$ to be equal to 1 right after activation of any node and equal to 0 the rest of the time. Also since the values of the desire vector range from $[-1\ 1]$, the $x_{ip}s \in [0\ 1]$ and $x_{ip}s \in [-1\ 0]$ can be quantized to 1 and 0 respectively to match the IC model representation of activation and deactivation.

Table 1. HIM Algorithm

HIM (*Agent*, E, P, A, A_R, H_{max}, r)
$H = 0$
$\mathbf{E}^H = \mathbf{E}$
$N^H = |A_R|$
While *stopCriteria* do
  $H = H + 1$
  infList = NULL
  for i = 1 to N^H do
    neighborList = FindNeighborList (i, r, $\mathbf{E}^H$)
    $\mathbf{E}_i^H$ = Subgraph (neighborList, $\mathbf{E}^H$)
    $\mathbf{E}_i^H$ = AddOutsideWorld ($\mathbf{E}^H$, $\mathbf{E}_i^H$)
    ($\mathbf{P}_i$, $\mathbf{A}_i$) = UpdateMat ($\mathbf{E}^{\mathbf{H}}$, $\mathbf{P}$, $\mathbf{A}$, neighborList)
    L = Optimize (*Agent*, $\mathbf{E}_i^H$, $\mathbf{P}_i$, $\mathbf{A}_i$)
    infList = infList $\bigcup$ L
    Agent = UpdateAgent (infList)
  end for
  $N^H = |\text{infList}|$
  $\mathbf{U}$ = MakeU (Agent)
  stopCriteria = UpdateCriteria (infList, H)
  $\mathbf{E}^H$ = UpdateHierarchy (infList)
end while
return $\mathbf{U}$

3.3 HIM Algorithm

Using these assumptions about customer product adoption dynamics, we devised a new scalable optimization technique, Hierarchical Influence Maximization (HIM). The pseudocode of our proposed HIM algorithm is presented in Table 1. Here, matrix $\mathbf{E}$ represents the connection matrix among *Regular* agents, and matrices $\mathbf{P}$ and $\mathbf{A}$ contain all

the p_{ij}'s and α_{ij}'s of the market model, respectively. In other words, all the interactions and influence probabilities between two pairs of *Regular* agents, (A_R), are embedded in the elements of these matrices. *Agent* contains all the information about *Regular* and *Product* agent characteristics including desire vectors, ($\overrightarrow{X_i}$'s), and influence tag vectors, $\overrightarrow{I_i}$'s with size P, where I_{ip} indicates the number of times that agent i has been selected as an influential node for product p. The algorithm receives as input all the available data on the agents and the model, and the output of the algorithm is the U matrix that contains the assignments of u_{ji}'s and shows the final connection matrix between all the products and influential seed nodes.

The level of the hierarchy is indicated by parameter H which increments until the stopping criteria are satisfied. At each hierarchy (H), we iterate over all the nodes (is) in the network of that hierarchy, (E^H), and list the neighboring agents around each node. The radius of the neighborhood, denoted with parameter r, indicates the granularity of analysis. Based on radius r, we partition the network into subsections, (E_i^H), and update the probability matrices, $\mathbf{P}_i$ and $\mathbf{A}_i$ for that subsection. HIM selects the influential agents in that local network, E_i^H, using an optimization technique and tags them for future use. The process of node selection is described in detail in 3.3. Then we add these influential nodes to the set of influential nodes that have been identified in other neighborhoods in the same hierarchy.

Outside World Effect When a local neighborhood is detached from the complete network, there exist boundary nodes that are connected to nodes outside the neighborhood. These connections that fall outside of the neighborhood can potentially affect the desire vector of agents within the neighborhood. One possible approach is to ignore these effects and only consider the nodes inside the partition. In this chapter we account for these effects by allocating a virtual node to each boundary node. This virtual node is the representative of all nodes outside the neighborhood that are connected to the boundary node. Figure 3 illustrates the abstraction of outside world effect and shows how the model's parameters are calculated between each boundary and virtual node.

Node Selection The process of selecting influential nodes is repeated at each hierarchy and at each local neighborhood surrounding node i. Following previous works [12, 13, 21], we model the desire dynamic of all agents as a Markov chain where the state of the local neighborhood is a matrix of all existing agents' desire vectors at a particular iteration k and the state transitions are calculated probabilistically from the pairwise interaction between agents connected in a network. The state of the local network around agent i at the k^{th} iteration is a vector of random variables, denoted as $\mathbf{X_i}(k) \in \mathbb{R}^{N_{H_i}P\times 1}$ (created through a concatenation of N_i^H vectors of size P) and expressed as:

$$\mathbf{X_i}(k) = \begin{pmatrix} [\overrightarrow{X_1}(k)] \\ \vdots \\ [\overrightarrow{X_{N_i^H}}(k)] \end{pmatrix}$$

We calculate the expected long-term desire of the agents in each local network around agent i and this calculation results in the following formulation:

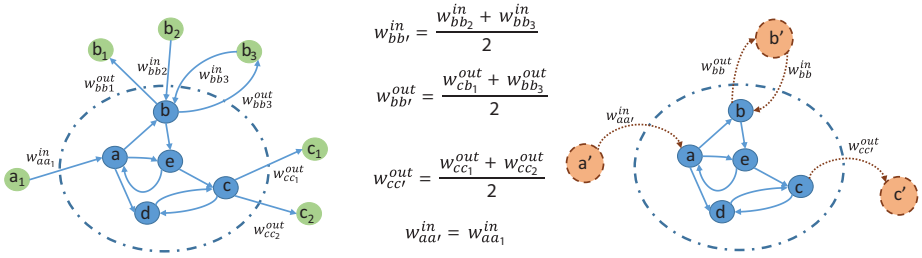

Fig. 3. The network on the left is an example of a neighborhood around node e; the network on the right is the equivalent network with virtual nodes representing the outside world effect. Here w can be any interaction parameter such as link's weight, α, or ϵ. The direction of the interaction with the virtual node is based on the type of links the boundary node has with the nodes outside the neighborhood. The value of the parameter is the average over all similar types of interactions with outside world.

$$E[\mathbf{X_i}(k+1)] = E[\mathbf{X_i}(k)] + \mathbf{Q_i}\ E[\mathbf{X_i}(k)]. \tag{5}$$

In order to solve this system of equations efficiently, we decompose the matrices:

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{pmatrix} \text{ and } \overrightarrow{\mu}_{\mathbf{X}}(\infty) = \begin{pmatrix} \overrightarrow{\mu}_{\mathbf{R}} \\ \overrightarrow{\mu}_{\mathbf{P}} \end{pmatrix} \tag{6}$$

Here $\mathbf{A} \in \mathbb{R}^{RP \times RP}$ is the sub-matrix representing the expected interactions among *Regular* agents while $\mathbf{B} \in \mathbb{R}^{RP \times P^2}$ represents the the expected interactions between *Regular* agents and *Product* agents. Figure 4 shows the breakdown of matrix $\mathbf{Q}$.

Moreover, $\overrightarrow{\mu}_{\mathbf{R}}$ and $\overrightarrow{\mu}_{\mathbf{P}}$ are vectors representing the expected long-term desire of *Regular* agents and *Product* agents, respectively, at iteration $k \to \infty$. Note that vector $\overrightarrow{\mu}_{\mathbf{P}}$ is known since the *Product* agents, the advertisers, are the immutable agents, who never change their desire. Solving for $\overrightarrow{\mu}_{\mathbf{R}}$ yields the vector of expected long-term desire for all regular agents, for a given set of influence probabilities on a deterministic social network.

$$\mathbf{A}\ \overrightarrow{\mu}_{\mathbf{R}} + \mathbf{B}\ \overrightarrow{\mu}_{\mathbf{P}} = 0 \Rightarrow \overrightarrow{\mu}_{\mathbf{R}} = \mathbf{A}^{-1}(-\mathbf{B}\ \overrightarrow{\mu}_{\mathbf{P}}) \tag{7}$$

Thus, we can identify the influential nodes in the network and connect the products to those agents in a way that maximizes the long-term desire of the agents in the social system. We define the objective function as the maximization of the weighted average of the expected long-term desire of all the *Regular* agents in the network toward all the products as:

$$\max_{u} \sum_{1 \le k \le P} \sum_{i \in A_R} (\rho_i . \overrightarrow{\mu}_{\mathbf{R},i}) \tag{8}$$

$\overrightarrow{\mu}_{\mathbf{R},i}$ is the part of $\overrightarrow{\mu}_{\mathbf{R}}$ that belongs to agent i, and ρ_i parameter is simply a weight we can assign to agents based on their importance in the network. In the case of equivalent $\rho_i = 1$ for all the agents, the above function reduces to the arithmetic mean of the expected long-term desire vectors for all agents.

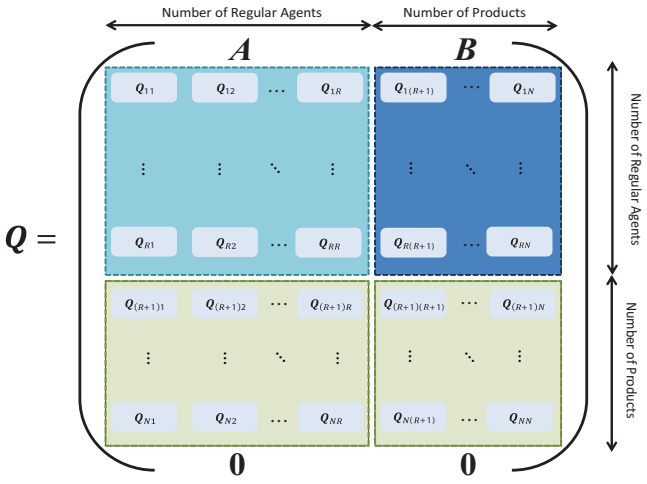

Fig. 4. Q matrix is a block matrix with size $N \times N$ where N is the total number of agents $(R + P)$ and each block has the size of $P \times P$. Matrices A and B are the non-zero part of this matrix which represent the interactions among *Regular* agents and interactions between *Regular* agents and *Products*, respectively.

Convergence Using the Brouwer fixed-point theorem [18], we prove that each local neighborhood has a fixed-point, hence solving Equation 5 at steady state is a valid choice. The theorem states that:

Theorem 1. *Every continuous function from a closed ball of a Euclidean space to itself has a fixed point.*

According to the calculation of Equation 5, $E[\mathbf{X_i}(k+1)]$ is a continuous function as it is the sum of two continuous ones. Also since $\overrightarrow{X_i}(k+1)$ in Equation 3 is a bounded function in $[-1\ 1]$, its expectation ($E[\mathbf{X_i}(k+1)]$) will be bounded as well. As a result we have a bounded, continuous function which is guaranteed a fixed point by the Brouwer fixed-point theorem. This allows us to solve our problem with the proposed optimization algorithm to find the assignment of $u_j$is in a way to maximize the long-term expected desire vector of agents toward all the products in the market.

Update Hierarchy When we proceed from one hierarchy to the next one, the selected nodes which are propagated to the upper hierarchy are not necessarily adjacent. Therefore, we need to define the interaction model between them based on their position in the real network. The $UpdateHierarchy$ function is responsible for building the proper network connection and interaction model for the next hierarchy based on the selected influential nodes in current hierarchy. These nodes were propagated to the higher hierarchy by being selected as influential nodes in at least one local neighborhood. It is possible for a node to be present in multiple partitions and be selected more than once.

Note that the selected nodes are unlikely to be adjacent nodes in the actual network E. Therefore we need to find a way to form their connections to construct E_H. To do so,

we look at the shortest path between these nodes in network E and use that to calculate the weight of the edges in E^H. In the E^H network the weight of the link between two selected nodes is the product of the weights of the shortest path between these two nodes in the previous hierarchy. Also the probabilities of interaction and influence between two influential nodes is set to be the product of the probabilities along the shortest path between them.

Termination Criteria To terminate the loop, we establish two different criteria in the $UpdateCriteria$ function. This function checks the stopping criteria based on the level of the hierarchy and the list of influential nodes. One criterion is based on the maximum number of levels in the hierarchy and the other is based on the ratio of the selected influential nodes and the advertising budget. According to the $stopCriteria$ output, the algorithm decides whether to proceed to a higher hierarchy or to stop the search, returning the current $\mathbf{U}$ matrix to be used as the advertising assignment.

Optimization Procedure The best assignment of *Product* agents to *Regular* agents is obtained through solving the following optimization problem:

$$\begin{aligned} \underset{\widehat{u}}{\text{maximize}} \quad & \|\mathbf{A}^{-1} Vec(\mathbf{M}\, \widehat{\mu}_{\mathbf{P}}\, \widehat{u})\|_1 \\ \text{subject to} \quad & x_{ip} \in [-1\ 1],\ \forall i \in A_R, \\ & \sum_{j \in A_R} u_{ij} = cte. \end{aligned} \tag{9}$$

Here, we are looking for a set of u_{ji}s which minimizes our cost or, in another words, maximizes the desire value of agents. Since u_{ji}s indicate the existence or lack of connection between *Product* and *Regular* agents, they are binary variables and can be identified using mixed integer programming. To solve our optimization problem, we used the GLPK (GNU Linear Programming Kit) package, which is designed for solving large-scale linear programming (LP) and mixed integer programming (MIP) problems. GLPK is a set of routines written in ANSI C and organized in the form of a callable library which is free to download from `http://www.gnu.org/software/glpk`.

4 Evaluation

4.1 Experimental Setup

We conducted a set of simulation experiments to evaluate the effectiveness of our proposed node selection method on marketing items in a simulated social system with a static network. The parameters of the interaction model for all runs are summarized in Table 2(a). All results are computed over an average of 100 runs which represent ten different simulations on each of ten network structures.

In the *Regular* and *Product* agent interactions, parameters α and ε are fixed for a given interaction and are presented in Table 2(a). We assume that these parameters can be calculated by advertising companies based on user modeling. The p_{ij} values for

this type of interaction are calculated using Equation 1 and are parametric. Table 2(b) provides the parameters for our HIM algorithm (neighborhood radius and the maximum hierarchy level). The remaining part of the social system setup is given by matrix $\mathbf{M}$, which models the correlation between the demand for different products. This matrix is generated uniformly with random numbers between $[0\ 1]$ and, as it has a probabilistic interpretation, the sum of the values in each row, showing the total demand for an item, is equal to one.

Table 2. Parameter settings

(a) Market Model Parameters

Parameter	Value	Descriptions
$Threshold$	2	Number of links between P and R agents
ε	0.4	Influence factor between P and R agents
α	0.8	Probability of influence between P and R agents
R	Variable	Number of *Regular* agents
P	10	Number of *Product* agents
$N_{Iterations}$	60,000	Number of iterations
N_{Run}	10	Number of runs
N_{Net}	10	Number of different networks

(b) HIM Parameters

Parameter	Value	Description
r	3	Neighborhood radius
H_{max}	5	Max level of hierarchy

4.2 Benchmarks

We compared our hierarchical algorithm with the non-hierarchical version, Optimized Influence Maximization (OIM) described in [21] and a set of centrality-based measures commonly used in social network analysis for identifying influential nodes based on network structure [14].

- **OIM:** The Optimized Influence Maximization method finds the influential nodes globally using our optimization method on the original network.
- **Degree:** Assuming that high-degree nodes are influential nodes in the network, we calculated the probability of advertising to a *Regular* agent based on the out-degree of the agents and linked the *Product* agents according to a preferential attachment model. Therefore, nodes with higher degree had an increased chance of being selected as an advertising target.
- **Betweenness:** This centrality metric measures the number of times a node appears on the geodesics connecting all the other nodes in the network. Nodes with the highest value of betweenness had the greatest chance of being selected as an influential node.
- **PageRank:** On the assumption that the nodes with the greatest PageRank score have a higher chance of influencing the other nodes, we based the probability of node selection on its PageRank value.
- **Random:** In this baseline, we simply select the nodes uniformly at random.

To evaluate these methods, we started the simulation with an initial desire vector set to 0 for all agents, and simulated 60000 iterations of agent interactions. The entire

process of interaction and influence is governed by Equations 3 and 4 (Section 3.2). At each iteration, we calculated the average of the expected desire value of the agents toward all products. This average is calculated over 100 runs (10 simulations on 10 different network structures) for the synthetic dataset and 100 runs on the real-world datasets. Note that the desire vector of *Product* agents remain fixed for all products; in our simulation it was set to 1 for the product itself and -0.1 for all other products (e.g., $\mu_1 = [1 \ -0.1 \ -0.1 \ldots \ -0.1]$).

4.3 Synthetic Dataset

For the synthetic dataset, we used the same network generation technique described in [21] for generating customer networks. To compare the performance of these methods, the average expected desire value of the agents in a network with 150 agents has been shown over time in Figure 5. Here we selected 150 agents as an optimal number of agents to compare all the algorithms together. With fewer agents, having ten simultaneously marketed products saturates the network while with a larger number of agents OIM suffers from scalability issues.

Marketing Effectiveness In Figure 5, by using the marketing-specific optimization methods for allocating the advertising budget, the desire value of the agents toward all products increases the most, resulting in the largest number of sales. Although HIM sacrificed some performance in favor of scalability, it clearly outperforms the centrality measurement methods. The locally-optimal selection approach of HIM results in a slightly lower performance compared to globally optimal OIM.

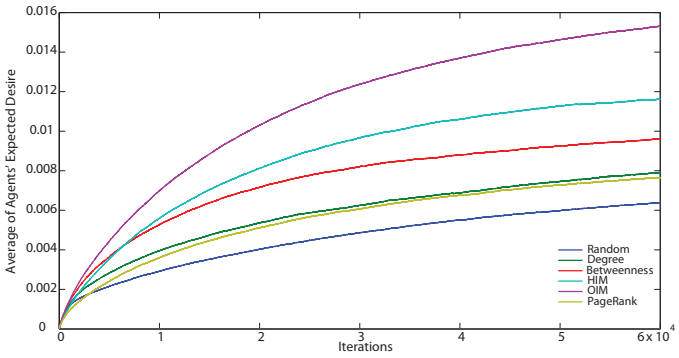

Fig. 5. The average of agents' expected desire vs. number of iterations, calculated across all products and over 100 runs (10 different runs on 10 different networks). The optimization methods have the highest average in comparison to the centrality measurement heuristics. As HIM is a sub-optimal method, it is unsurprising that its performance is worse than the global optimization method, OIM.

Figure 6 shows the final average value of the expected desire of agents in the last iteration for different number of *Regular* agents. Although OIM with global optimization

method outperforms HIM and other centrality measurement methods, it is incapable of scaling up to 300 and more agents in the network due to near singular interaction matrix. HIM, with its ability to scale up linearly, provides a sub-optimal and yet practical solution in selecting the influential nodes in large networks.

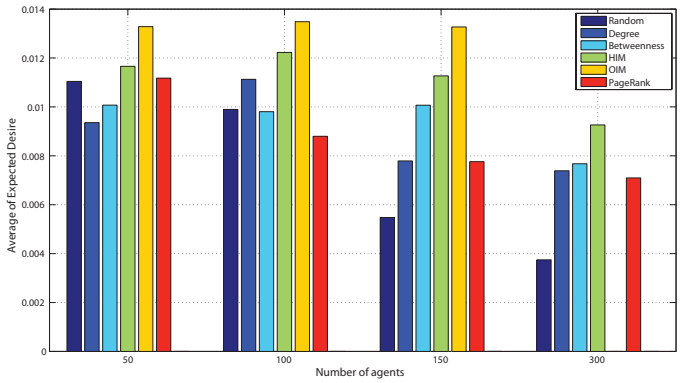

Fig. 6. The average of the final expected desire vectors for different numbers of $Regular$ agents and 10 $Product$ agents. The optimization based methods (OIM and HIM) outperform the other methods in selecting the seed nodes. While OIM is more successful than HIM in selecting the influential nodes, it is unable to scale-up to networks with 300 agents and higher.

Run-time Table 3 shows a runtime comparison between the two optimization methods, HIM (proposed) and OIM (original). In small networks the runtime of the global optimization method is less than the hierarchical but as the size of network grows, its run time increases exponentially while the run time of the HIM increases at a slower rate. The long runtime of OIM for the networks larger than 200 nodes makes the algorithm impractical for finding influential nodes in very large networks.

Table 3. Runtime comparison between OIM and HIM

Number of agents	OIM	HIM
50	10.67s	74.09s
100	94.76s	160.80s
150	290.67s	208.97s
200	897.51s	354.35s

Jaccard Similarity To analyze the differences between the algorithms' selection of influential nodes, we use the Jaccard similarity measurement. This measurement is cal-

culated by dividing the intersection of two selected sets by the union of these sets. Figure 7 shows this measurement for all pairs of algorithms. The OIM and HIM algorithms have the highest similarity compared to the other methods with a similarity value of 0.47. The other pairs of methods have very low similarities, resulting in dark squares in the figure. Not surprisingly, Random has the least similar node selection to other methods. This shows that HIM finds many of the same nodes as the original OIM algorithm, with a much lower runtime cost.

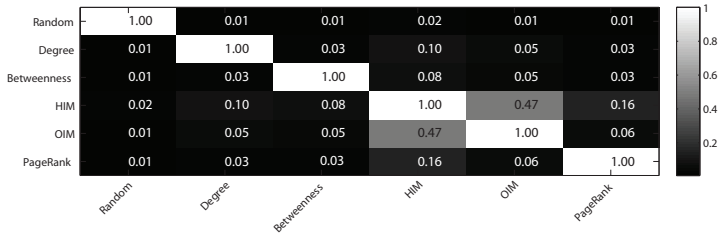

Fig. 7. The average Jaccard similarity measurements between different methods, calculated over 100 runs (10 runs on 10 different networks). Lighter squares denote greater similarity between a pair of algorithms. Note that HIM's selection of nodes is fairly close to OIM's optimal selection.

4.4 Real-world Datasets

We also evaluated the performance and scalability of our proposed algorithm on real-world directed networks from the Stanford Network Analysis Project (`http://snap.stanford.edu/`).

- **WikiVote** The network contains all the Wikipedia voting data from the inception of Wikipedia until January 2008. Nodes in the network represent Wikipedia users, and a directed edge from node i to node j indicates that user i voted on user j.
- **SlashDots** is a technology-related news website known for its user community. The website features user-submitted technology-oriented news. In 2002 Slashdot introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes. This network contains friend/foe links between Slashdot users, obtained in February 2009.
- **Epinions** This is a network extracted from the consumer review site Epinions.com. Nodes are members of the site who have reviewed products. A directed edge from i to j indicates j trusts i's reviews (and thus i has influence over j).

In all the experiments on real-world social media, we have preprocessed the networks to eliminate isolated nodes and boundary nodes (nodes with a degree of one). Tables 4(a) and 4(b) summarize the statistics of these real-world networks before and after the preprocessing stages, respectively. We used the same experimental parameters (presented in Section 4.1). The only differences are the number of products and the advertising budget which are equal to 10 and 50, respectively.

Table 4. Statistics of the Real-world Networks

	(a) Before Pre-processing			(b) After Pre-processing		
Dataset	WikiVote	SlashDot	Epinion	WikiVote	SlashDot	Epinion
$\#Nodes$	7K	82K	76K	2k	72K	20K
$\#Edges$	100K	950K	509K	38K	840K	3700
$Average\ \ Degree$	14.6	13.4	6.7	31.1	10.5	28.9
$Maximal\ \ Degree$	1167	3079	3079	714	5059	256
$Diameter$	7	11	14	7	13	12

We benchmarked our optimization methods against two state of the art influence maximization methods, Prefix-excluding Maximum Influence Arborescence (PMIA) [25] and DegreeDiscount [9], in addition to the centrality measures.

- **PMIA:** This heuristic algorithm, [25], examines the local neighborhood of each node to find the influence pattern in each local arborescence in order to estimate the influence propagation across the network. To our knowledge, the PMIA algorithm is the best scalable solution to the influence maximization problem under the Independent Cascade Model.
- **DegreeDiscount:** This heuristic algorithm presented by Chen et al. [9], refined the degree method by discounting the degree of nodes whenever a neighbor has already been selected as an influential node.

Although using a hierarchical approach reduces the problem of dealing with huge interaction matrices, it is still possible for network partitions to be quite large if they are centered on a high degree node that is connected to a large portion of the network. In addition to creating huge interaction matrices, these nodes will create star-shape subgraphs which result in an infeasible solution for the optimization process. There are a couple of solutions for dealing with these very high degree nodes: 1) ignore them when we partition the network and assume that their high connectivity guarantees that they will appear within the network neighborhood of other nodes or 2) ignore some of the low-degree neighbors of the node. In the following experiments, we adopted the first approach in dealing with these large partitions. Therefore, in all networks we only centered partitions around nodes with a degree less than 100. Examining the average degree of nodes in all datasets presented in Table 4(b) shows that this selection not only prevents huge matrices and star-shaped subgraphs but still gives us a high percentage of nodes to process. The following results have been generated for the WikiVote and Epinion datasets.

Marketing Effectiveness Figure 8 gives the average expected desire value for all the agents over time for $300K$ iterations of the simulated market. In this result, the OIM algorithm has the highest value while HIM algorithm follows it closely. The performance of the HIM algorithm approaches the global optimization method (OIM). The performance of the DegreeDiscount heuristic, PMIA, and PageRank algorithms are very close to each other with no significant differences.

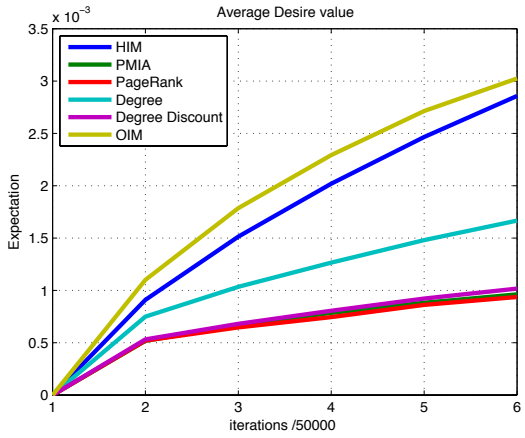

Fig. 8. The average of agents' expected desire vs. number of iterations for the WikiVote dataset, calculated across all products over 100 runs. The dataset was preprocessed by eliminating isolated and boundary nodes, yielding 2K nodes, and the simulation was run for 300K iterations. The optimization methods have the highest average in comparison to the rest of benchmarks. As the HIM algorithm is a sub-optimal method, its performance is less than the global optimization method.

While our algorithms outperform the other benchmarks on the WikiVote dataset, on the Epinion dataset the degree-based algorithms perform better. Figure 9 shows the results for all the benchmarks and the HIM algorithm. Although the HIM performance is better than PMIA and PageRank, it does not beat the degree-based algorithms, Degree and DegreeDiscount.

Figure 10 summarizes the final expected desire value of agents for different algorithms and for different datasets. The low value of desire vector is a consequence of having a low number of advertisers within huge networks; during influence propagation, the agent's desire vectors are repeatedly multiplied by ϵ and α.

Analysis of Dataset Degree Distributions To understand the poor performance of HIM on the Epinion dataset, we examined the network structure to see how the networks different from one another. Table 5 shows the quantile analysis of the node degree for the pre-processed datasets. Based on this analysis we see that the WikiVote network is a very small network compared to other two datasets, yet the max degree of the lower quartiles is higher the other networks. This indicates that the WikiVote network has a more uniform degree distribution, where node degree is not likely to be a highly discriminating feature of influence propagation potential.

This can be verified by looking at the degree distributions of the datasets (Figures 11, 12, and 13). In the Epinion and SlashDot datasets we have a small number of nodes with very high degrees while most of the nodes in the network possess a degree less than 10. In these networks, a few nodes serve as hubs and are highly connected, whereas the other nodes have few connections that, in the worst case, aren't even connected to the

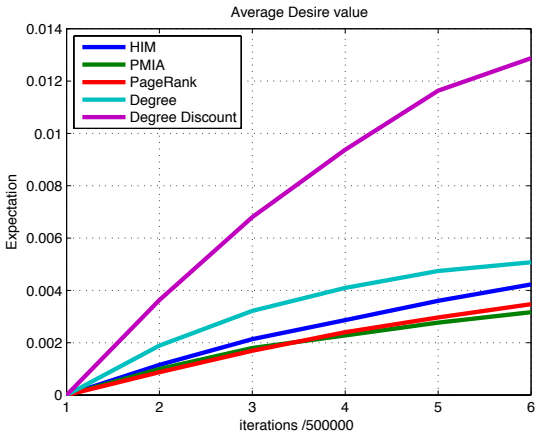

Fig. 9. The average of agents' expected desire vs. number of iterations for the Epinion dataset, calculated across all products, over 100 runs. The dataset was preprocessed by eliminating isolated and boundary nodes, yielding 20K nodes, and the simulation was run for 300K iterations. HIM outperforms PMIA and PageRank, but it beaten by the degree-based algorithms, Degree and DegreeDiscount. The OIM algorithm could not be run on this dataset, due to the size of the network.

Table 5. Quantile Analysis of Node Degree in Preprocessed Datasets

Dataset	0%	25%	50%	75%	100%
WikiVote	3	25	44	79.25	714
Epinion	0	6	11	33	2684
SlashDot	3	4	7	17	5061

high degree node. Hence our heuristic of not centering the partitions on high degree nodes sabotages the performance of HIM's optimization procedure. On the other hand the degree-based algorithms can effectively target these high degree nodes. In contrast, in the networks such as WikiVote or the synthetic networks where the node degree is more uniform, HIM works well as the nodes in the middle bins are more numerous and better connected to the entire network. In this case, the degree-based algorithms perform poorly since degree is not as discriminative.

Optimization with Degree-based Heuristic Based on these results, we modified our preprocessing procedure to use a degree-based heuristic to select the nodes considered by our optimization technique. Here, we selected the top 5% of high degree nodes in the Epinion dataset and created a single-level abstracted network based on the shortest path among these nodes. Then we ran our optimization technique (OIM) on the single network. Figure 14 shows the result of OIM and other benchmarks on this preprocessed

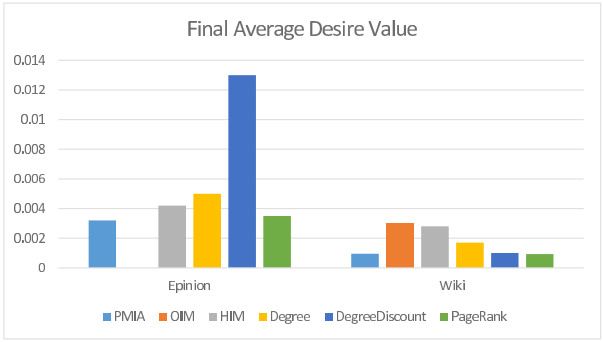

Fig. 10. The final expected desire value of the agents at the end of the simulation for the different methods and datasets. The OIM algorithm could not be run on the Epinion dataset, due to the size of the network.

network. The result shows that applying optimization to the abstracted network conclusively outperforms the other benchmarks.

5 Conclusion and Future Work

In this chapter, we address the problem of influence maximization in social networks for the purpose of advertising. In an advertising domain, our goal is to identify the influential nodes in a social network as advertiser targets based on the network structure, the interactions among the agents in the network, and the limited advertising budget. We adopted agent-based modeling to model such a social system as it is a powerful tool for the study of phenomena that are difficult to study within the confines of the laboratory. We also attempted to model the market, the interactions and propagation of influence, and the product adoption more realistically by incorporating factors such as product correlation and group membership of agents.

Here we present a general hierarchical approach for applying optimization techniques to influence maximization. The advantage our method has over network-only seed selection techniques is that it can account for item correlations and community effects on the product adoption rate. Our method comes close to the optimal node selection, at substantially lower runtime costs. However, prior analysis of the network degree distribution of the network is essential for identifying the correct preprocessing and abstraction procedure. The HIM algorithm can be used to improve the scalability of influence maximization on networks with a semi-uniform degree distribution. In networks with a high centralization, we recommend applying our optimization technique to an abstracted version of the network created from the high degree nodes. In this chapter, we have proposed one approach to partitioning the network into overlapping sections and performing influence maximization on the partitions. Another alternative would be to leverage preexisting network divisions computed with community detection algorithms for the first level of the hierarchy. Furthermore, working with dynamic

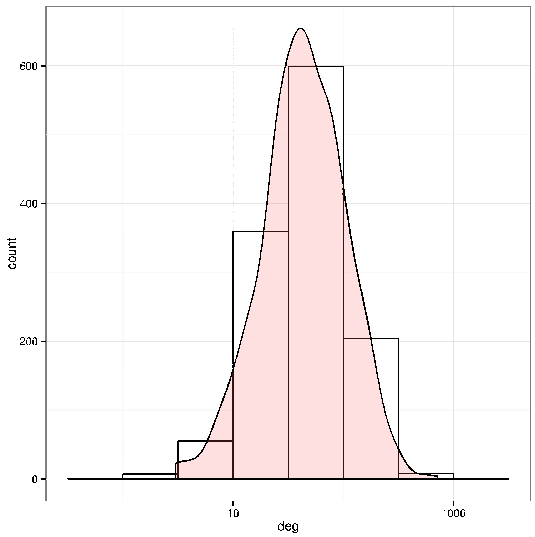

Fig. 11. The degree histogram of the WikiVote dataset. The x-axis shows the logarithmic scale of degree, and the curve shows the kernel density estimation. In this dataset the majority of nodes lie in the middle range and have a degree between 50 to 100.

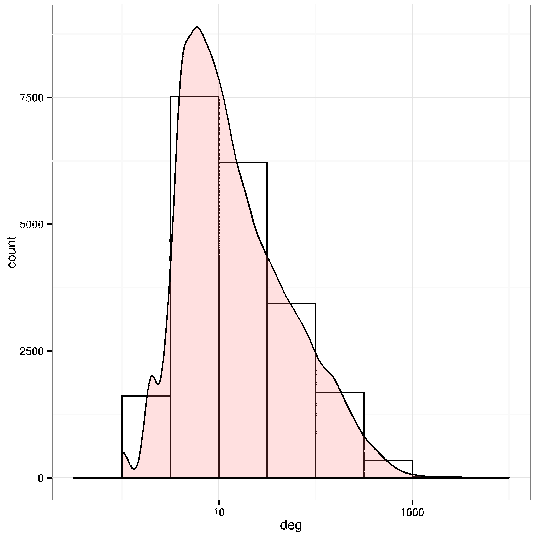

Fig. 12. The degree histogram of the Epinion dataset. The x-axis shows the logarithmic scale of degree, and the curve shows the kernel density estimation. In this dataset the network has a sparse structure, with the majority of nodes possessing a degree less than 10.

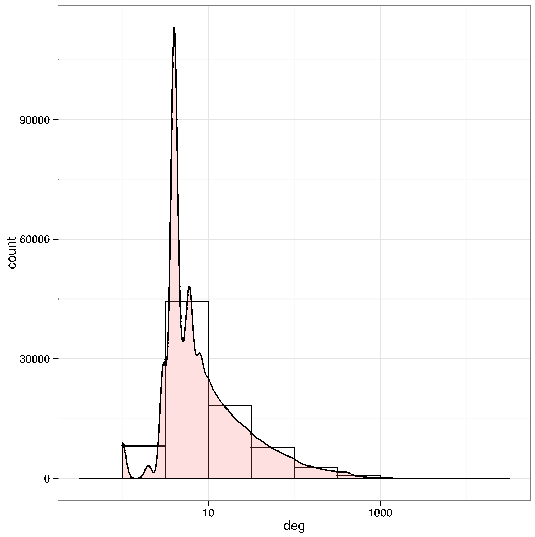

Fig. 13. The degree histogram of the SlashDot dataset. The x-axis shows the logarithmic scale of degree, and the curve shows the kernel density estimation. In this dataset, the same as Epinion dataset, the network has a sparse structure, with the majority of nodes possessing a degree less than 10.

networks where the agents can enter and leave the network would be useful for practical applications in which the pool of customers is constantly changing.

An important potential extension of this work would be to generalize the market simulation to explicitly model the adversarial effects between competing advertisers as a Stackelberg competition, in which one advertiser places ads and subsequent competitors have knowledge of existing ad placement. In this chapter we assumed that the probability of interaction and influence between two agents is small, compared to the size of the network, which results in the agents sticking to a decision for a reasonable period of time. However if the network is smaller or the probability of interaction increases, there can be large fluctuations in the agents' desire vector. Applying a parameter to the model which forces the agents to retain their decisions for a minimum period, regardless of external interactions, would ameliorate this issue [20]. A more general framework for modeling and simulating customer product adoption within social networks would be of great practical importance; our model represents initial steps towards this ambitious goal.

Acknowledgments

This research was supported in part by NSF IIS-08451.

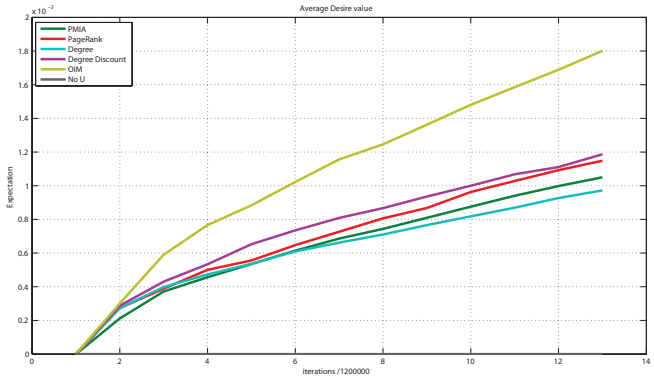

Fig. 14. The average of agents' expected desire vs. number of iterations for the Epinion dataset, calculated across all products and over 10 different runs, for 300K iterations. The dataset was preprocessed by selecting the 1% top degree nodes and building a subgraph based on the shortest path between these nodes, rendering the graph small enough to be directly processed with OIM. OIM outperforms the degree-based methods.

References

1. Anagnostopoulos, A., Kumar, R., Mahdian, M.: Influence and correlation in social networks. In: Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 7–15 (2008)
2. Apolloni, A., Channakeshava, K., Durbeck, L., Khan, M., Kuhlman, C., Lewis, B., Swarup, S.: A study of information diffusion over a realistic social network model. In: Proceedings of the International Conference on Computational Science and Engineering. pp. 675–682 (2009)
3. Aral, S., Walker, D.: Identifying influential and susceptible members of social networks. Science 337(6092), 337–341 (2012)
4. Bagherjeiran, A., Parekh, R.: Combining behavioral and social network data for online advertising. In: IEEE International Conference on Data Mining Workshops (ICDMW). pp. 837–846 (2008)
5. Bharathi, S., Kempe, D., Salek, M.: Competitive influence maximization in social networks. Internet and Network Economics pp. 306–311 (2007)
6. Borodin, A., Filmus, Y., Oren, J.: Threshold models for competitive influence in social networks. Internet and Network Economics pp. 539–550 (2010)
7. Chen, W., Collins, A., Cummings, R., Ke, T., et. al: Influence maximization in social networks when negative opinions may emerge and propagate. In: Proceedings of the SIAM International Conference on Data Mining (2011)
8. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1029–1038 (2010)
9. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 199–208 (2009)
10. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the IEEE International Conference on Data Mining (ICDM). pp. 88–97 (2010)

11. Hartline, J., Mirrokni, V., Sundararajan, M.: Optimal marketing strategies over social networks. In: Proceeding of the International Conference on World Wide Web. pp. 189–198. ACM (2008)
12. Hung, B.: Optimization-based selection of influential agents in a rural Afghan social network. Master's Thesis, Massachusetts Institute of Technology (2010)
13. Hung, B., Kolitz, S., Ozdaglar, A.: Optimization-based influencing of village social networks in a counterinsurgency. In: Proceedings of the International Conference on Social Computing, Behavioral-cultural Modeling and Prediction. pp. 10–17 (2011)
14. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 137–146. ACM (2003)
15. Kempe, D., Kleinberg, J., Tardos, É.: Influential nodes in a diffusion model for social networks. Automata, Languages and Programming pp. 1127–1138 (2005)
16. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. Knowledge Discovery in Databases (PKDD) pp. 259–271 (2006)
17. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. Social Computing and Behavioral Modeling pp. 1–8 (2009)
18. Leborgne, D.: Calcul différentiel et géometrie. Presses universitaires de France (1982)
19. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 420–429 (2007)
20. Liow, L., Cheng, S., Lau, H.: Niche-seeking in influence maximization with adversary. In: Proceedings of the Annual International Conference on Electronic Commerce. pp. 107–112. ACM (2012)
21. Maghami, M., Sukthankar, G.: Identifying influential agents for advertising in multi-agent markets. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems. pp. 687–694 (2012)
22. Maghami, M., Sukthankar, G.: Hierarchical influence maximization for advertising in multi-agent markets. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 21–27. Niagara Falls, Canada (August 2013)
23. Pathak, N., Banerjee, A., Srivastava, J.: A generalized linear threshold model for multiple cascades. In: International Conference on Data Mining (ICDM). pp. 965–970 (2010)
24. Shakarian, P., Paulo, D.: Large social networks can be targeted for viral marketing with small seed sets. In: Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 1–8 (2012)
25. Wang, C., Chen, W., Wang, Y.: Scalable influence maximization for independent cascade model in large-scale social networks. Data Mining and Knowledge Discovery pp. 1–32 (2012)
26. Yang, W., Dia, J., Cheng, H., Lin, H.: Mining social networks for targeted advertising. In: Proceedings of the Annual Hawaii International Conference on System Sciences. IEEE Computer Society (2006)