2014

Identifying Influential Agents In Social Systems

Mahsa Maghami
University of Central Florida

IDENTIFYING INFLUENTIAL AGENTS IN SOCIAL SYSTEMS

by

MAHSA MAGHAMI
B.S. at K.N. Toosi University of Technology, 2005
M.S. at University of Tehran, 2007

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2014

Major Professor: Gita R. Sukthankar

ABSTRACT

This dissertation addresses the problem of influence maximization in social networks. Influence maximization is applicable to many types of real-world problems, including modeling contagion, technology adoption, and viral marketing. Here we examine an advertisement domain in which the overarching goal is to find the influential nodes in a social network, based on the network structure and the interactions, as targets of advertisement. The assumption is that advertisement budget limits prevent us from sending the advertisement to everybody in the network. Therefore, a wise selection of the people can be beneficial in increasing the product adoption. To model these social systems, agent-based modeling, a powerful tool for the study of phenomena that are difficult to observe within the confines of the laboratory, is used.

To analyze marketing scenarios, this dissertation proposes a new method for propagating information through a social system and demonstrates how it can be used to develop a product advertisement strategy in a simulated market. We consider the desire of agents toward purchasing an item as a random variable and solve the influence maximization problem in steady state using an optimization method to assign the advertisement of available products to appropriate messenger agents. Our market simulation 1) accounts for the effects of group membership on agent attitudes 2) has a network structure that is similar to realistic human systems 3) models inter-product preference correlations that can be learned from market data. The results on synthetic data show that this method is significantly better than network analysis methods based on centrality measures.

The optimized influence maximization (OIM) described above, has some limitations. For instance, it relies on a global estimation of the interaction among agents in the network, rendering it incapable of handling large networks. Although OIM is capable of finding the influential nodes in the social network in an optimized way and targeting them for advertising, in large networks, performing the matrix operations required to find the optimized solution is intractable.

To overcome this limitation, we then propose a hierarchical influence maximization (HIM)

algorithm for scaling influence maximization to larger networks. In the hierarchical method 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. The budget limitation for the advertising dictates the algorithm's stopping point. On synthetic datasets, we show that our method comes close to the optimal node selection, at substantially lower runtime costs.

We present results from applying the HIM algorithm to real-world datasets collected from social media sites with large numbers of users (Epinions, SlashDot, and WikiVote) and compare it with two benchmarks, PMIA and DegreeDiscount, to examine the scalability and performance. Our experimental results reveal that HIM scales to larger networks but is outperformed by degree-based algorithms in highly-connected networks. However, HIM performs well in modular networks where the communities are clearly separable with small number of cross-community edges. This finding suggests that for practical applications it is useful to account for network properties when selecting an influence maximization method.

To Tara,

my beloved daughter, all my inspiration for life,

$\sim$

To Ramin,

my beloved husband, my very best friend,

$\sim$

To my dear parents

my perfect support team throughout my life.

ACKNOWLEDGMENTS

I would like to thank Dr. Gita Sukthankar for her guidance, advice, and fully support. She was an amazing adviser, a wise mentor, and a caring friend through all these years. Under her guidance,not only I learnt how to be a good researcher but also a better human being. I am grateful to Xi Wang, Bulent Tastan, Rahmatollah Beheshti, Erfan Davami, and Fahad Shah for being amazing lab-mates and for their inspiration. I am also grateful to Dr. Annie Wu, Dr. Damla Turgut, Dr. Lotzi Boloni, and Dr. Ivan Garibay for serving on my committee and their helpful advice. My especial thanks to my dear husband, Ramin Mehran, my lovely parents, and my dear sister, Katayoon, for all their support, encouragement, and help, all these years.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Overview and Motivation

The gift of persuasion is a powerful and highly-sought after skill, as evidenced by the fact that individual self-help books in this area, the most famous being *How to Win Friends and Influence People* published in 1936, remain popular. The rise of social media outlets and click-through advertisement opened the door for relatively small groups to influence large numbers of people. Combined with modern data analysis techniques, it is possible to create a detailed social simulation of the population of interest, but the problem of whom to influence remains as an open research question. Particularly in advertisement, indiscriminate mass marketing techniques can lead to negative information cascades about product quality, even if cost efficiency is not an issue. This problem can be framed as a network influence propagation problem; previous work in this area has looked at diverse domains such as information propagation in the Flickr social network [16] and identifying important blogs for marketing [6].

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 [3]. 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 [54]. Much of the previous influence maximization work [21, 18, 100] 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 [54]. 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., [79], [11]). 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, first 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. We believe that in marketing, all these factors affect the customers' susceptibility to influence and their ability to influence others. As an example, [5] analyzes the effect of social status on the influence factor of people on Facebook. 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 [17].

In this thesis, we use social simulation to facilitate the study of phenomena that are difficult to study within the confines of the laboratory. Although all simulations need to be validated with other types of experimental results, agent-based simulations are one tool for studying effects that occur on a long time scale over large groups of people. In this thesis, we present a paradigm for studying the impact of social factors, on task-oriented groups and on influence propagation.

Social simulations have been used to address many types of questions including how social ties and connections influence the propagation of information [37], the spread of epidemics [78] and the emergence of social conventions [26]. Here in one section of the work, we examine the impact of social phenomena such as stereotype on the structure of the network. The social system is simulated using an adaptive network that modifies its structure based on the agents' experiences. In our experiments, we quantify how the network structure affects group formation and task accomplishment of agent teams. In contrast to previous work [71] that investigated the impact of group membership on stereotype formation here we focus on the impact of stereotype bias on link creation and, consequently, group formation.

Group membership influences many aspects of our lives, including our self-identities, activities, and associates; it affects not only what we do and who we do it with, but what we think of ourselves and the people around us. It can also give rise to stereotypical thinking in which group differences are magnified and the importance of individual variations are discounted. Thinking categorically about people simplifies and streamlines the person perception process [67], facilitating information processing by allowing the perceiver to rely on previously stored knowledge in place of incoming information [45]. Stereotypes based on relatively enduring characteristics, such as race, religion, and gender, have an enormous potential for error [45] and can give rise to performance impairments [71].

We hypothesize that stereotypes formed independently of real group differences can result in negative effects for the collective system and therefore affect the propagation of influence as well. However, studying the long-term effects of stereotypes can be difficult, especially to quantify the effects over a population rather than an individual. Here we describe an agent-based simulation for evaluating the impact of stereotypes on the performance of task-oriented groups. Understanding the role that stereotypes play in group formation can refine existing theory while providing insight into the efficacy of methods for combating the adverse effects of stereotypes on behavior and decision-making.

To examine the effect of stereotypes on the social interaction and the network structure, we base our simulation on a model of multi-agent team formation [35] since task-oriented groups share many characteristics with teams, although lacking in shared training experience. In our simulation, the population of agents is connected by a social network that is locally updated over time by unoccupied agents depending on their preferences. Stereotypes are represented as an acquired preference model based on prior experience and observable agent features. In multi-agent environments, stereotypes have been used to provide faster modeling of other agents [27, 28] and also to bootstrap the trust evaluation of unknown agents [13]. In contrast, we examine the case of non-informative stereotypes; stereotypes affect the agents' preferences for forming social attachments but do not affect the agents' willingness or ability to cooperate with other agents.

Moreover, in investigating the influential nodes in the social network, we incorporated the stereotype model in our social simulation model to present a more realistic model of the interaction between people. We assumed that these group formations not only affect the structure of the network to alter the pattern of influence propagation but also that they play an important role in affecting the decision making of people in adopting a specific behavior or selecting a specific product in the market.

1.2 Contributions

Our main focus in this thesis is to investigate the influence propagation in a social network and identify the influential people in a connected social network as targets for advertising.

In our first contribution, we present a mathematical analysis of how influence propagation occurs over time and propose a new optimization technique for identifying effective messenger agents in the network that outperforms other network analysis methods while accounting for realistic factors such as group membership and product preference correlation. Following the work of Hung et al. [47, 48], optimization is used along with an analysis of the expected long-term system behavior to assign the advertisement of the available products to appropriate agents in the network. In contrast with previous work on identifying influential nodes for marketing purposes (e.g., [42] and [8]), in this thesis we model the effects of realistic social factors such as group membership on product adoption. In the analysis presented in [47, 48] for counterinsurgency messaging tactics, there exists a single random variable representing the attitude of agents toward counterinsurgency, but in our work, we use a vector of random variables which represents the desire of each agent toward any single product. This consideration combined with product demand correlations in the market make the analysis and optimization more complicated, but ultimately our approach has the promise of being applicable to a wider variety of social systems.

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. As a result, in our second contribution, 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 create a scalable algorithm Hierarchical Influence Maximization (HIM) for maximizing steady-state product adoption by customers connected by a social network. 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. Since social factors play an important role in the propagation of influence among connected people, we investigated the effects of one of the most common social factors, stereotype bias. This investigation prompted the use of a more complex interaction model in the influence maximization problem.

1.3 Organization of the thesis

This document is organized as follows. Chapter 2 provides an overview of the related work in social simulation models, agent-based models, and influence maximization in social networks. Chapter 3 presents our proposed model for stereotypes in multi-agent systems and the impact of stereotypes on group formation. Chapter 4 introduces our influence maximization techniques, including the optimization based method and a hierarchical extension, as well as summarizing the operation of the realistic product adoption model. The evaluation of our proposed methods vs. other centrality based network analysis techniques can be found in Sections 4.4.3 and 4.5.6. Chapter 6 concludes the document.

CHAPTER 2: RELATED WORK

In this chapter, we introduce the context of this research by covering the most relevant recent research results. First we provide an overview about social systems modeling which focuses on the two main approaches, agent-based modeling and variable-based modeling. Next we present the related work on stereotype modeling. In addition, we present some of the works in network structures and group formation. Finally, we target the literature on the influence maximization problem and we present some of the prominent works on this topic.

2.1 Modeling Social Systems

In this section, we review the two main methodologies, agent-based modeling (ABM) and variable-based modeling (VBM), commonly used to model social systems. Our research utilizes an agent-based model of human communities to examine the group effects on task completion and product adoption. We discuss the strengths and weaknesses of both modeling methodologies to illustrate why ABM is well suited for this particular problem.

2.1.1 Agent-based Modeling

Agent-based modeling (ABM), with its focus on representing biological agents and their interaction [74], provides a powerful way to study the behavior of heterogeneous agents in a dynamic environment over an extended period of time [30, 10]. An ABM is a simulated multi-agent system capable of capturing key theoretical elements of some social or psychological process. [1]In an ABM, each agent usually represents a simplified, abstract version of a human being, that acts according to a set of theoretically postulated behavioral rules. These rules may involve simple heuristics or more complicated mechanisms that include learning, constructing internal represen-

[1]See [43] for a review of simulation approaches in social psychology

tations of the world, or other computational models of decision-making [91]. In this work we use this approach to analyze a phenomenon, stereotype bias, which is difficult to study accurately in real world. Using an agent-based model allows us evaluate the effect of different system parameters on network structure, team formation, and the global performance of the agents.

The advantages of this approach can be listed as [91]:

- We are able to envision the large-scale consequences of theoretical assumptions when the behaviors are performed in the context of many other agents and iterated dynamically over an extended period of time.

- We are capable of bridging between the micro level of individual agent behaviors and interactions to the macro level of the overall patterns that result in population-level effects.

- In contract to real-world, we have the capability of setting the values of parameters in our multi-agent model to arbitrary values.

- We have the flexibility of testing our theories in the real world with a much better vision of what we are looking for and how to interpret our findings.

Pioneering models presented by Schelling [85, 86] and Kalick and Hamilton [50] were among the first examples of the use of agent-based simulations for social modeling. Since then, agent-based modeling has been utilized in many different fields including economics [96, 97, 98], psychology [7], ecology [40], sociology [68, 83, 76, 31], and biology [94].

2.1.2 Variable-Based Modeling

In the traditional modeling approach employed by social psychologists, variable-based modeling (VBM), the focus is on relations among variables, not on interactions among agents, in contrast to the agent-oriented ABM. With the theoretical analysis of VBM, it is difficult to model dynamic networks of agents, agent learning or/and evolution, or non-linear interactions between agents [10]. Especially, in analyzing social and psychological phenomena, where the result of repeated interactions between multiple individuals over time matters, VBM is not able to model and capture the types of complex, dynamic, and interactive processes [91, 31]. Also, in contrast to ABM which offers an applied statistical approach, VBM offers the generative or mechanistic explanation [91].

For this research, we opted to use ABM over VBM in order to study two aspects of stereotype bias: 1) the effect of repeated interactions on network structure and 2) the impact of network structure on group formation. These two phenomena are not easily quantified and modeled using VBM, making ABM the better approach. For a general conceptual introduction to ABM and its uses in social psychology, please see [31, 103, 30, 81, 7].

2.2 Stereotypes

The related work on stereotype modeling spans diverse areas including human decision-making [32], intelligent tutoring systems [51], trust and reputation [64, 63, 14, 15], and general multi-agent systems [13, 27, 28]. Although stereotypes exist across cultures, the actual stereotypic beliefs can differ significantly [22].

The judgments we make about one person's behavior are more likely to influence judgments of the same or a different person who performs the same behavior at a later time, even when the traits were only associated with particular actors but not attributed to them [95]. These judgments could be simply based on the ethnicity, personal appearance or attributes of other people [49]. Here, in this work we are concerned about modeling judgments which simply rely on visible attributes.

Not only do stereotypes affect our own perceptions and judgments, but also they are propagated from person to person in a social network [66] where these stereotypes persist over time [84].

This propagation of stereotypical information in a network, followed by the expectancies it engenders about what a specific group as a whole is like [41], motivated us to model stereotype bias as affecting the entire social network. Our agent-based simulation considers the impact the stereotype has on the network and accounts for that as well as the effect on individual agent decision making. By updating the connections of the social network among the agents based on the experience of the agent's neighbors, we capture the concept of propagation of stereotypical information in a network and its remaining within the community.

Recently, there has been interest in incorporating stereotype modeling into multi-agent systems. [13] used stereotypes to bootstrap their evaluations of new and unknown partners in open, dynamic multi-agent systems. In their model, similar to this work, the agents interact in ad-hoc groups and use the stereotypical information as an additional source of information to evaluate the trustworthiness of the other unknown agents. But unlike in our work, the stereotypical information directly affects the judgment of agents in selecting partners, whereas here it has a long-term effect on the social network but is not utilized directly in the group formation mechanism.

Denzinger and Hamdan [27, 28] enhanced the prediction of other agents' behaviors by applying stereotype models. They tested their model on a toy problem and the results showed substantial benefits in using stereotypical knowledge. The common element of these works is the use of similar or frequent patterns of agents to build stereotypical knowledge for other unknown agents or behaviors in the system. But, as mentioned earlier, there is no assurance about the beneficial or destructive effects of the stereotypes on the structure of the social network and consequent effects of these structural changes on other social activities like group formation and teamwork.

In the area of trust and reputation, Casare et al. [14, 15] introduced a new type of reputation called "stereotype reputation" which is based on the social prejudices and computed with no direct interaction among agents. The StereoTrust computational model, presented in [63] and [64], uses real-life stereotypes and the biases people perceive from past experiences to build a trust model in online environments about risky transactions. In contrast, our model does not directly affect the agent's decision-making but has the subtler effect of modifying the pool of neighboring agents.

To model social stereotyping, some of the previous work has utilized connectionist networks. Connectionism spread from cognitive psychology to social psychology [72, 73] as social psychologists found that connectionist models of neural concepts are directly relevant to social constructs [93]. Smith and DeCoster [93, 92] proposed a recurrent connectionist network model to simulate phenomena related to person perception and group-based stereotyping. They demonstrated that a connectionist memory could learn a group stereotype when presented with a number of input patterns representing individual group members; in our agent-based simulation, stereotypic value judgments are learned using linear regression, based on an initial set of training experiences and held fixed for the remainder of the simulation. Queller et al. [80] proposed a distributed connectionist network model to examine the effects of of stereotype change and development. Finally, Van Rooy [99] created a connectionist agent-based model to simulate stereotype effects in a social network. They model the effects of social influence by accounting for variables on both the individual and aggregate level of social systems. In contrast with this work, they do not consider the the dynamic nature of the social network.

2.3 Network Structure and Group Formation

In synthetic systems, individual robots or agents, will often need to form coalitions to accomplish complicated tasks which they are not able to accomplish alone [39]. The execution of complex tasks may require cooperation among agents and efficient grouping strategies to accomplish the task successfully [89]. To achieve efficiency in its performance, an agent system should employ a reasonable organizational design [46]. As the organization of a multi-agent system is the collection of roles, relationships, and authority structures which govern its behavior [46], our research focuses on the effect of stereotype on this organization and network structure and its consequences on task-oriented group formation. The existence of dynamic environments, such as in [33], and partial observation of agents in the system, (e.g., [1]), makes group formation more challenging and vulnerable to social forces such as stereotype bias and prejudice.

An important aspect of social systems is information propagation, which is significantly affected by network structure, and in turn affects group formation and the emergence of social conventions. [26] analyzes the effect of network structure in the emergence of social conventions in multi-agent systems. The results show that complex graphs make the system much more efficient than regular graphs with the same average number of links per node and that scale-free networks make the system as efficient as fully connected graphs. Also, Glinton et al. [37, 38], analyzed local belief sharing of agents in a peer-to peer network and its impact on dynamics of information propagation in large heterogeneous teams. Their work shows how the dynamics of information propagation is vulnerable to small amounts of anomalous information maliciously injected in the system. In our work, the stereotypical judgments of agents propagate through the network while the agents adapt their connections based on their neighbors connections.

2.4 Influence Maximization Problem in Marketing

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. In [3], the authors examine the statistical correlation between the actions of friends in a social network by considering factors such as homophily and possible unobserved confounding variables. Hence it follows that it is not only important to advertise to your customer but also to your potential customer's friends.

Influence maximization was first studied as an algorithmic problem by Domingos et al. [29] who viewed the market as a social network and modeled the system as a Markov random field. Later, Kempe et al. [53] formulated influence maximization as a discrete optimization problem and proved that a greedy node selection approach obtains a solution within 63% of optimal for this

NP-hard problem. In [54], the behavior spreads in a cascading fashion according to a probabilistic rule, beginning with a set of initially active nodes. To identify influential agents, they select a set of individuals to target for initial activation, such that the cascade beginning with this active set is as large as possible in expectation. [57] find influential nodes in a complex social network by formulating the likelihood for information diffusion data, the activation time sequence data over all nodes; they propose an iterative method to search for the probabilities that maximize this likelihood. Although this was an important theoretical result, their proposed greedy algorithm was neither fast nor particularly scalable to larger networks.

This motivated work on potential speedups; examples of this line of research include innovations such as the use of a shortest-path based influence cascade model [56] or a lazy-forward optimization algorithm [61], in order to reduce the number of evaluations on the influence spread of nodes. [20] made improvements upon existing greedy algorithms to further reduce run-time and also proposed new degree discount heuristics that improve influence spread. Clever heuristics have been used very successfully to speed computation in both the LT model (e.g., the PMIA algorithm [18]) and also the IC model [100]. In this dissertation, 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.

As an alternative to greedy algorithms that reach approximate solutions using graph theory (e.g., [58, 55]), Dayama et al. [25] formulate the problem as a continuous-time deterministic optimal control problem and uses a mean-field approach. More commonly, the problem is framed as identifying a set of initial nodes that can trigger large behavior cascades that spread through the network. This set of nodes can then be identified either using probabilistic approaches [4, 57] or optimization-based techniques. For instance, [47, 69] 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.

Apolloni et al. [4] examine the spread of information through personal conversations by proposing a probabilistic model to determine whether two people will converse about a particular topic based on their similarity and familiarity. On the other hand, [82] propose a learning method for ranking influential nodes and perform behavioral analysis of topic propagation; they compare the results with conventional heuristics that do not consider diffusion phenomena. Ghanem et al. [36] investigate the difference in the relative time people allocate to their friends versus that which their friends allocate to them, and propose a measure for this difference in time allocation. The distribution of this measure is used to identify classes of social agents through agglomerative hierarchical clustering. They demonstrate their approach on two large social networks obtained from Facebook. The characteristics of these datasets are presented in [104].

First, we present one approach for framing and solving the optimization problem using convex programming. The optimization problem can also be solved using greedy algorithms (e.g., [58, 55]) that find approximate solutions using graph theory. [53] also utilized greedy algorithms to identify the influential nodes. Intelligent heuristics can be used to improve the scalability of influence maximization [19]. [20] made improvements upon existing greedy algorithms to further reduce run-time and proposed new degree discount heuristics that improve influence spread. In [24], authors take a mean-field approach and formulate the problem as a continuous-time deterministic optimal control problem.

The effects of network topology on influence propagation have been studied in several domains, including technology diffusion, strategy adaption in game-theoretic settings, and the admission of new products in the market [53]. It has been demonstrated that the way information spreads is affected by the topology of the interaction network [105] and also that there exists a relationship between a person's social group and his/her personal behavior [90].

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 [53]. For viral marketing, influential nodes can be identified either by following interaction data or probabilistic strategies. For example, Hartline et al. [42] solve a revenue maximization problem to investigate effective marketing strategies. [106] 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 [8]. 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 [88]. 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.

Outside of social network marketing approaches, there exist many marketing methods based on personalization techniques for delivering advertisements [52] or news [6].

Some researchers (e.g., [62, 12]) 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 [11] a natural and mathematically tractable model is presented for the diffusion of multiple innovations in a network. Our work assumes that influential nodes are partitioned between advertisers in an adversarial offline process.

CHAPTER 3: STEREOTYPE EFFECT ON GROUP FORMATION

Agent-based simulations can be an important tool for modeling social systems, enabling researchers to examine phenomena that are difficult to study empirically. In this dissertation, we introduce an agent-based simulation for investigating the impact of social factors on the formation and evolution of task-oriented groups. Task-oriented groups are created explicitly to perform a task, and all members derive benefits from task completion. However, even in cases when all group members act in a way that is locally optimal for task completion, social forces that have mild effects on choice of associates can have a measurable impact on task completion performance. In this dissertation, we show how our simulation can be used to model the impact of stereotypes on group formation. The effects of stereotype bias on a social system are notoriously difficult to study due to problems with subject self-reporting and creating experimental manipulations. In our model, stereotypes are based on observable features, learned from prior experience, and only affect an agent's link formation preferences. Even without assuming stereotypes affect the agents' willingness or ability to complete tasks, the long-term modifications that stereotypes have on the agents' social network impair the agents' ability to form groups with sufficient diversity of skills, as compared to agents who form links randomly. An interesting finding is that this effect holds even in cases where stereotype preference and skill existence are completely uncorrelated. When stereotype affects the formation of social networks and network structure modifies the outcome of group formation, stereotype bias can have long-lasting consequences on a populations' ability to form effective groups.

3.1 Problem Statement

To explore the impact of stereotype on group formation and network evolution, we have selected a simple multi-agent system model first introduced by Gaston and desJardins [35] and used in [33, 39] to govern team formation. Since task-oriented groups are similar to teams, this is a reasonable method for modeling the task performance of group behavior on shared utility tasks in absence or existence of stereotypes. Also since this model assumes an adaptive network, it is well suited for analyzing longer term effects of stereotype bias.

In this model, there is a population of N agents represented by the set $A = \{a_1, \ldots, a_N\}$. Each agent can be considered as a unique node in the social network and the connection between the agents is modeled by an adjacency matrix E, where $e_{ij} = 1$ indicates an undirected edge between agent a_i and a_j, and the degree of agent a_i is defined as $d_i = \sum_{a_j \subseteq A} e_{ij}$. Each agent is assigned randomly a single uniformly selected skill given by $\sigma_i \in [1, \sigma]$ where σ is the total number of available skills. Accomplishing each task requires a coalition of agents with the appropriate skills. Tasks are globally advertised for γ time steps at fixed intervals μ. If a coalition of satisfactory agents does not form for a task in designated γ steps, the task will disappear from the environment and marked as unaccomplished. The parameter μ in the model indicates the urgency of task accomplishment. When this parameter's value is low, new tasks in the environment are advertised more frequently and thus need to be accomplished faster.

Each task, T_k, has a size, $|T_k|$, that denotes the number of skills required to accomplish the task and a $|T_k|$-dimensional vector of required skills, R_{T_k}, which are selected uniformly from $[1, \sigma]$. Also, $M_k \subset A$ indicates the set of team associated with T_k. When a coalition has formed with the full complement of skills required for the task, α time steps are required for the group to complete the task. After α time steps the task is marked as accomplished and the agents on the task will be released into the environment to look for new tasks.

During the team formation process, each agent, a_i, can be in one of three states, s_i, defined as: UNCOMMITTED, COMMITTED, or ACTIVE. UNCOMMITTED denotes the state where an agent has not been assigned to a task and is seeking a new task. An agent in the COMMITTED state has been assigned to a task but is still waiting for enough agents with the right skills to join the group. Finally, an ACTIVE agent is currently working on a task with a complete group possessing the right complement of skills. All members of a complete group at a specific task, e.g. T_j, will remain α time steps in ACTIVE state to complete the task.

On each iteration, agents are updated in random order to avoid any bias toward task assignment. UNCOMMITTED agents have the opportunity to adapt their local connectivity (with probability of P_i) or can attempt to join one of existing incomplete groups. Figure 3.1 shows the block diagram of the overall updating process for UNCOMMITTED agents.

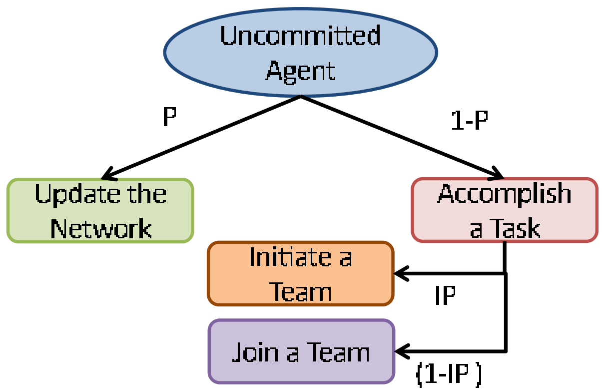

Figure 3.1: Complete updating process for each UNCOMMITTED agent

3.1.1 Group Formation

To implement the group formation process, we simply follow the group formation algorithm used to allocate agents to teams in [33]. In this work, the group formation algorithm is identical in both cases of having or not having the stereotypical judgment among the agents. The difference between these two cases lies in the network updating algorithm which will be discussed in the following section.

According to Figure 3.1, when an agent decides to form a group instead of updating its network, it either chooses to initiate a new group and be the first committed member of the group or to join one of the existing groups and assist the completion of the group. Selecting between these two cases is dependent on the probability IP_i for agent a_i.

Probability IP_i is proportional to the number of immediate UNCOMMITTED neighbors defined as follows:

$$IP_i = \frac{\sum_{a_j \subseteq A} e_{ij} I(s_i, \text{UNCOMMITTED})}{\sum_{a_j \subseteq A} e_{ij}}, \tag{3.1}$$

where $I(x, y) = 1$ when $x = y$ and 0 otherwise.

In this case, when the agent has more neighbors in UNCOMMITTED status, there is a higher chance for it to initiate a team by itself. Agents are only eligible to join task-oriented groups in their local neighborhood, where there is at least one link between the agent and the group members. This eligibility criterion makes the definition of Equation 3.1 more meaningful as the higher number of UNCOMMITTED agents is equivalent to a reduced opportunity to be admitted into an existing group. The algorithm used by an agent to initiate or join a group is presented in Algorithm 1.

Algorithm 1 Group formation algorithm

```
for all T_K ⊆ T do
  if |M_k| = 0 and s_i = UNCOMMITTED then
    r ← UniformRandom([0, 1])
    if r ≤ IP_i then
      if ∃r ∈ R_{T_k} : r = σ_i then
        M_k ← M_k ∪ {a_i}
        s_i ← COMMITTED
      end if
    end if
  else if ∃a_j : e_ij = 1, a_j ∈ M_k and s_i = UNCOMMITTED then
    if ∃r ∈ R_{T_k} : r = σ_i and r is unfilled then
      M_k ← M_k ∪ {a_i}
      s_i ← COMMITTED
    end if
  end if
end for
```

3.1.2 Network Adaptation

In the scenario with no stereotype bias, to adapt the network structure, the agents modify their local connectivity based on the notion of preferential attachment [2]. Therefore, the probability of connecting to a given node is proportional to that node's degree. As mentioned before, at each iteration the agent can opt to adapt its connectivity, with probability P_i. Modifying its local connectivity does not increase the degree of the initiating agent since the agent severs one of its existing connections at random and forms a new connection.

To form a new connection, an agent considers the set of its neighbors' neighbors designated as $N_i^2 = \{a_m : e_{ij} = 1, e_{jm} = 1, e_{im} = 0, m \neq i\}$. The adapting agent, a_i, selects a target agent, $a_j \subseteq N_i^2$, to link to based on the following probability distribution:

$$P(a_i \longrightarrow a_j) = \frac{d_j}{\sum_{a_l \subseteq N_i^2} d_l} \tag{3.2}$$

where d is the degree of agents.

The results in [33] and [39] show that this simple algorithm can be used to adapt a wide variety of random network topologies to produce networks that are efficient at information propagation and result in scale-free networks similar to those observed in human societies. Our model uses this same method for updating the network for group formation in the baseline (non stereotype bias).

3.2 Learning the Stereotype Model

As noted in a review of the stereotype literature [45], stereotypes are beliefs about the members of a group according to their attributes and features. It has been shown that the stereotypes operate as a source of expectancies about what a group as a whole is like as well as what attributes individual group members are likely to possess [41]. Stereotype influences can be viewed as a judgment about the members of a specific group based on relatively enduring characteristics rather than their real characteristics.

Here, we represent a stereotype as a function $\mathcal{F} : \overrightarrow{\mathrm{V}} \longrightarrow S$, mapping a feature vector of agents, $\overrightarrow{\mathrm{V}}$, to a stereotypical impression of agents in forming friendships, S, which we will designate as the stereotype value judgment. This value represents the agents' judgments on other groups and is only based on observable features rather than skills or prior task performance.

In most contexts, humans possess two types of information about others: 1) information about the individual's attributes and 2) the person's long-term membership in stereotyped groups [41]. Therefore, to learn the stereotype model, the simulation offers these two sources of information, $\overrightarrow{\mathrm{V}}$ and its corresponding S which are related to the agents' group membership, for a specific period of time. In our simulation, this initial learning period lasts for I time steps and helps the collaborating agents gain experience about the attributes of different groups of agents. Note that membership in these groups is permanent and not related to the agent's history of participation in short-term task-oriented groups.

During the initial period, the whole process is the same as the rest of simulation with the difference that there exist no network updating. Therefore, according to Figure 3.1, an uncommitted agent with probability P_i either decides to do nothing or accomplish a task. Here, in any collaboration, agents will be provided by the feature vector of their team members and their corresponding stereotype value judgment. These feature vectors and stereotype value judgments are derived from the group membership of agents which was set at the beginning of the simulation. Hence, at the end of the initial period each collaborated agent has a stack of feature vectors and their corresponding stereotype value judgments which we call the "experience" of that agent. It is clear that the size of this stack is different from agent to agent and it is related to the number of collaborations they had.

In our work, we propose that each agent, a_i, can use linear regression to build its own judgmental function, $\mathcal{F}_i$ based on its own experience, and consequently to estimate the stereotype value of another agent, a_j, according to the observable features of that agent, $\overrightarrow{\mathrm{V}_j}$. Note that after initial learning period, each agent builds its own linear function which is only based on its collaboration experience and is different from others. Therefore, after the initial learning period, I time steps, the estimated stereotype value of agent a_j by agent a_i will be uniquely calculated as $\hat{S_{ij}} = \mathcal{F}_i(\overrightarrow{\mathrm{V}_j})$.

In our model, this stereotype value judgment affects the connection of agents during the network adaptation phase, as we will describe in the following section.

3.2.1 Network Adaptation with Stereotype Value Judgments

In the stereotype case, the group formation algorithm is the same as described in Algorithm 1 but the network adaptation is based on the learned stereotype. If an agent decides to adapt its local network, again with probability P_i , it will do so based on its own stereotype model. To adapt the local connectivity network, each agent uses its learned model to make stereotype value judgment on other neighboring agents. This network adaptation process consists of selecting a link to sever and forming a new link.

Specifically, the agent a_i first searches for its immediate neighbor that has the lowest stereotype value judgment, a_j, and severs that link. The agent then searches for a new agent as a target for link formation. To form this link, it searches its immediate neighbors and the neighbors of neighbors. First the agent selects the neighbor with the highest stereotype value judgment, a_m, for a referral as this agent is likely to be a popular agent in its neighborhood. Then the adapting agent, a_i, will establish a new connection with a_n, one of the most popular neighbors of a_m, assuming that it is not already connected.

$$a_n = \arg_{a_k \in N_i^2, e_{ik}=0} \max \hat{S_{ik}}.$$

Note that all of these selections are the result of the stereotype value judgment model that agent a_i has about the other agents in its neighborhood.

3.2.2 Experimental Setup

We conducted a set of simulation experiments to evaluate the effects of stereotype value judgments on the interaction network structure and consequently on group formation in a simulated society of agents. Although there exist several specialized programming languages and tool kits for agent-based simulations such as NetLogo [102], Repast [77], MASON [65], Swarm [75], we opted to use Matlab to design and model our system due to the ease of implementing the learning aspect of the system. While in [34] the claim that network structure has significant impact on team formation in networked multi-agent systems, our experiments were designed to reveal the potential impact of stereotype bias on task-oriented group formation within social systems. Note that stereotype bias only affects network structure and not group formation; the agents always join available groups formed by their network neighbors whenever their skills are needed.

The parameters of the group formation model for all the runs are summarized in Table 4.4(a). In task generation, each task is created with a random number of components less than or equal to σ and a vector of uniformly-distributed skill requirements with the same size. To generate the agent society, each agent is assigned a specific skill, a feature vector, and a class label. The agents' skills are randomly generated from available skills. Inspired by [13], four different long-lasting groups with different feature vector distributions are used as the basis for stereotype value judgments. Agents are assigned a six-dimensional feature vector, with each dimension representing an observable attribute, and a hidden stereotype value judgment drawn from Gaussian distribution assigned to the group. Table 5.1(b), shows the mean and standard deviations of the Gaussian distributions and the observable feature vector assigned to each group. The binary observable feature vectors are slightly noisy. To indicate the existence of an attribute, a random number is selected from distribution $N(0.9, 0.05)$ to be close to 1 and to indicate the lack of an attribute this number is selected from distribution $N(0.1, 0.05)$ to be close to zero. During the initial training period, here $I = 2000$ iterations, agents are allowed to observe the hidden stereotype value judgment of other agents to learn the classifier that will be used for the rest of the agent's lifetime. During the remainder of the simulation (5000 iterations), the agent uses the learned classifier to make its own stereotype value judgments about others.

In these experiments all the runs start with a random geometric graph (RGG) as the initial network topology among the agents. A RGG is generated by randomly distributing all the agents in a unit square and connecting two agents if their distance is less than or equal to a specified threshold, d [23]. The random network we generated is a modified version of the RGG, proposed by [33]. In this version d is selected as a minimal distance among the agents to guarantee that all the agents have at least one link to other agents.

When the initial network is generated, the group formation is allowed for an initial period with no adaptation $(I = 2000)$. During these initial training steps, the agents can form groups and participate in task completion to gain experiences about working with other agents. Therefore, the network topology remains static during the $I = 2000$ iterations and after this training period the agents start updating their interaction network as described in 3.1.2 and 3.2.1 in two cases of having and not having stereotypical judgment among the agents, respectively.

In this set of experiments our main focus is on the effect of two control parameters, μ and σ, on the team formation and task performance when the stereotypical judgment exists among the agents. Simulation parameter μ, which indicates the task interval, controls the frequency of task injection in the environment and the load of task accomplishment while parameter σ controls the complexity of tasks in case of number of required skills. The results are conducted in a way to show how the effect of stereotypical judgment can vary in different situations such as having more complicated tasks in the environment or having more tasks to accomplish.

All experiments are based on the average of 10 different runs with a different initial network for each run.

Table 3.1: Parameter settings

(a) Experimental parameters

Parameter	Value	Descriptions
N	120	Total number of agents
σ	6, 10	Total number of skills
γ	10	Time steps for task advertisement
α	4	Agents' active time
μ	2, 10	Task interval
$\lvert T\rvert$	max 10	Number of skills required for a task
$N_{Iterations}$	5000	Number of iterations
$N_{Initial}$	2000	Number of learning iterations

(b) Stereotype groups and feature vectors

Group	Mean Value	StDev	f_1	f_2	f_3	f_4	f_5	f_6
G1	0.9	0.05	X					X
G2	0.6	0.15		X		X		
G3	0.4	0.15			X	X		
G4	0.3	0.1		X	X		X	

3.2.3 Results

3.2.3.1 Global Performance

The global performance of the system, like [33], is calculated as follows:

$$Performance = \frac{T_{SuccessfullyDone}}{T_{Total}}, \tag{3.3}$$

which is the proportion of successfully accomplished tasks divided by the total number of introduced tasks in the environment. Figure 4.4 shows the global performance of the system with stereotypes and without stereotypes (named **Plain**) by iteration. For the stereotype condition we tested the performance of the social system once with learned stereotypes (**StLin**), where the agents based their stereotypical judgments on their learned model, and once with no learning (**StNL**), where the agents had perfect knowledge about the assigned judgment value of other agents. The results of these three different algorithms are shown and compared for only two different values of μ. To select values of μ, we set this parameter to even numbers in the interval of $[216]$ and calculated the performance. As there exists no significant difference between the performance value in high values of μ and also no significant difference in low values of μ, therefore we picked values 2 and 10 as the representative of the performance result at low and high values of task interval, respectively. Also we did the same process for parameter σ but we only show the results for $\sigma = 10$ as it is representing moderate complex tasks; not too complex to prevent the agents to have successful accomplishment and not too simple to be done easily.

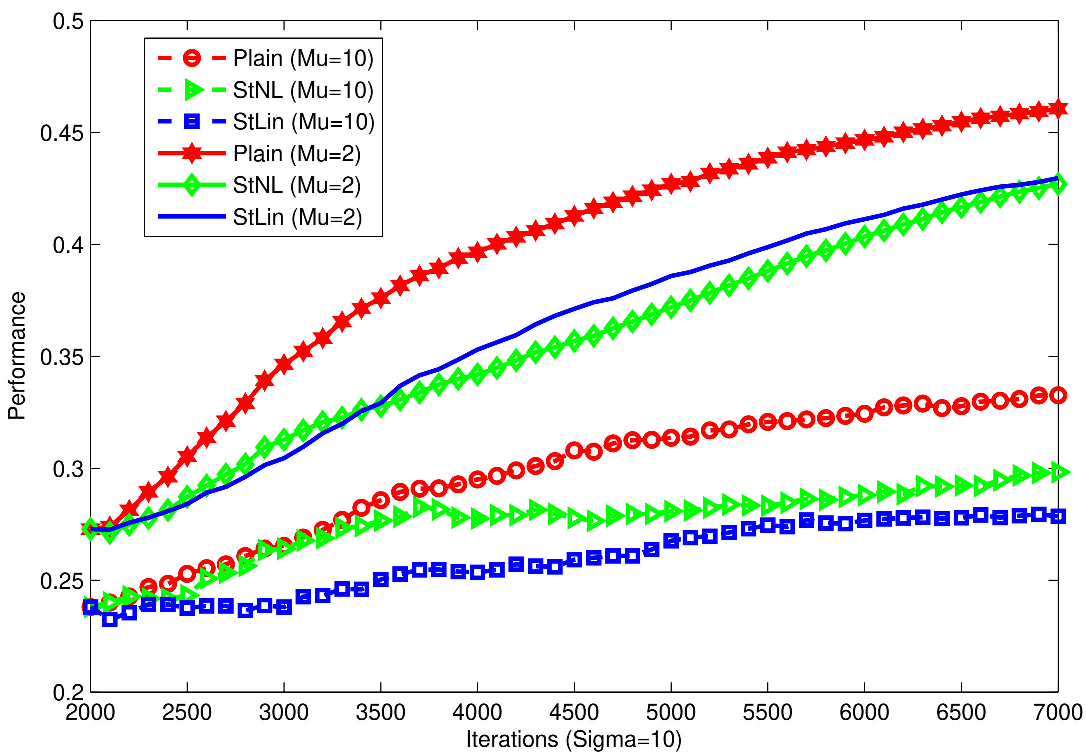

Figure 3.2: The performance of task-oriented groups (with and without stereotypes) vs. iterations shown for two different values of μ and a fixed value of $\sigma = 10$. The performance is significantly lower in both stereotype conditions and drops dramatically when μ is increased.

As it is shown, the performance of the system in the **Plain** condition is noticeably higher than the two stereotype bias conditions. The significance of the difference between the **Plain** and two conditions with stereotypes was measured with the student's T test and was found to be statistically significant at the $\alpha = 0.05$ significance level. There was no significant difference between learned stereotypes or those based on perfect knowledge. The same pattern of results occurred with $\mu = 10$ but with a dramatic drop in the task performance, resulting from the less frequent injection of tasks into the system.

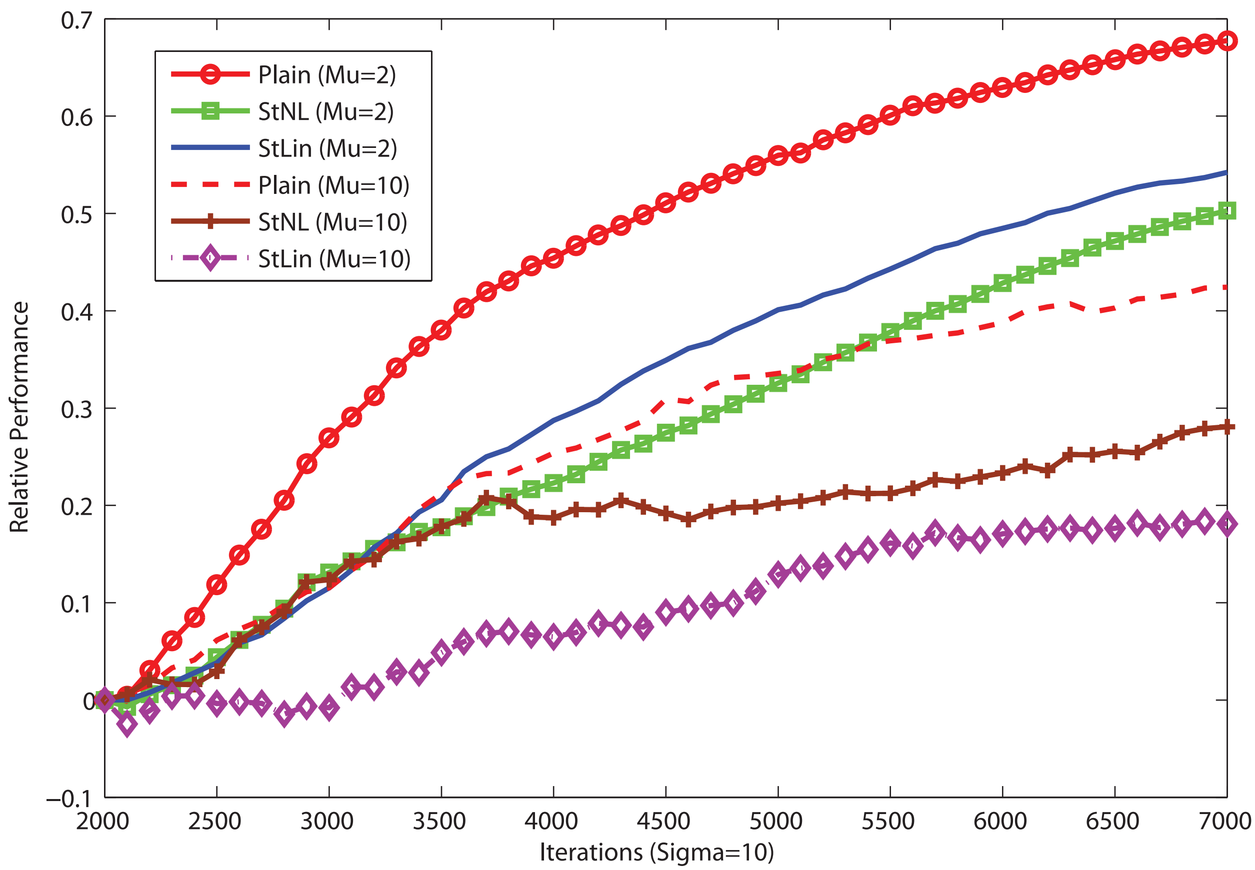

Figure 3.3: The relative performance of task-oriented groups (with and without stereotypes) vs. the iterations for two different values of μ and a fixed value of $\sigma = 10$. In the case with no stereotype bias, the performance of the overall social system experiences a higher rate of increase with more iterations as compared to the cases with stereotype.

In addition to general performance, we calculated the relative performance as well. The relative performance is the comparison between the global performance at any iteration with the measurement at the end of the initial period to evaluate the improvement of the agents collaboration over the time compared to the starting point.

Figure 3.3 shows this evaluation for different conditions. The same as Figure 4.4, the value of σ is fixed to 10, and results are shown for two different values of μ but other values of μ followed a similar pattern. The experiments show that for $\mu = 2$, in the **Plain** condition the performance of the system increases almost 70% in comparison with the initial performance after the learning period. In the stereotype condition this improvement is only around 50%. The main effect of the stereotype is to adapt the network toward a sparse network structure with a dramatic increase in isolate nodes. This drop in performance is even more pronounced with fewer total agents. Also the increase in the μ value drops performance as the number of advertised tasks decreases dramatically. Also we conclude that the task injection or in another words, the load of the tasks in the system is independent of the stereotype effect as changing this value keeps the pattern of systems's performance the same in difference algorithms.

3.2.3.2 Local Performance

Equation 3.3 can be used to compute a global performance evaluation of the social system but sometimes it is instructive to also examine individual performances or local performance. According to [33], the local performance can be calculated using the successful rate of agents (SR) defined as:

$$SR = \frac{N_{SuccessfulJoined}}{N_{Joined}}, \tag{3.4}$$

where $N_{SuccessfulJoined}$ is the number of successful teams joined by an agent divided by the total teams joined N_{Joined}. Here the N_{Joined} value is calculated as the total number of teams that agent initiated by itself summed to the ones it joined. Figure 3.4 shows the average of successful rate value (SR) of all agents for different values of μ and σ for all the conditions.

The results show that by freezing the parameter μ to value 2 and changing σ (figure on right), the successful rate value decreases dramatically as σ increases. This pattern occurs in all three cases but in the stereotype condition this value suffers more from the increase of σ. As the number of skills required to accomplish the tasks increases, finding the right collaboration of agents becomes more critical and ignoring agents due to stereotype bias becomes more destructive. The other values of μ (not shown) almost follow the same trend and it shows that changing the task injection and load of the work does not significantly effect the successful rate of the agents on average.

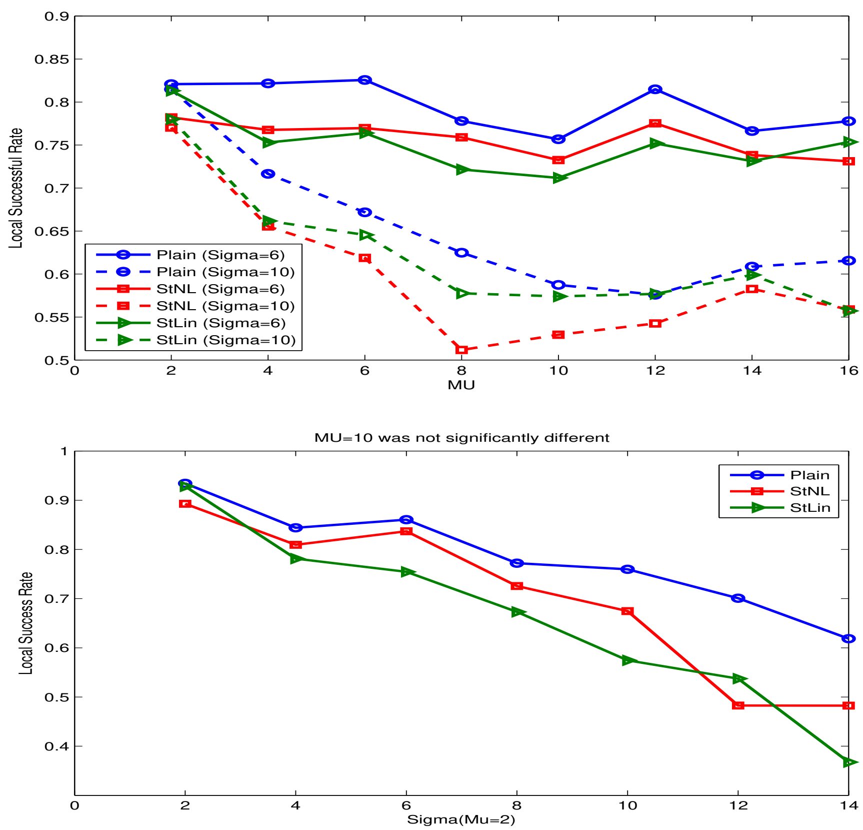

Figure 3.4: The effect of parameters μ and σ on successful rate of agents in the environment. The figure on top fixed $\mu = 2$ and varied the σ for all three approaches with and without stereotypes. The figure on the bottom, shows the variation of μ and fixed values of $\sigma = 6$ and $\sigma = 10$

Moreover, when we freeze the value of σ and change the parameter μ (figure on left) we can see that for low number of required skills ($\sigma = 6$) the successful rate is not really dependent on the frequency of task advertising. But when the σ increases to 10, the successful rate decreases slightly. In all results the successful rate of the stereotype conditions is lower than the non stereotype condition. Here, the same as the performance result, we can conclude that the load work of the system has not significant effect on the team formation. This is reasonable as during the team formation and making decision to join a group, the agents do not consider other remaining tasks in the environment. What plays a significant role is their match skill and their connection with any current group members at the task therefore, when the number of required skill increases, fulfilling all these requirements gets harder and harder and consequently makes the ratio of unsuccessful tasks higher.

3.2.3.3 Linear Regression Learning

To evaluate the performance of the applied linear regression method at learning stereotype value judgments, we calculate the Mean Square Error (MSE) between the estimation of learning model (StLin) and the model with ideal knowledge (StNL). The result is shown in Figures 3.5 and 3.6 for different values of μ and σ parameters, respectively.

In Figure 3.5, we can see that increasing μ increases the error in estimating the true stereotype value of the agents; fewer tasks and collaborations reduces the amount of training data accumulated, resulting in a less accurate model. In these results when σ is fixed to 10, the difference between the error in different parameter setting of μ becomes less significantly different. In other words, when the number of required skills increases, the agents have a reduced chance of group formation. This case is magnified in the stereotype condition and not offset by the increased frequency of tasks.

In Figure 3.6, the MSE result has been shown for two different values of μ while the σ parameter is modified. These results indicate that with a higher value of μ, the error is increased in conditions where σ is equal to 2, 4, or 6 but when σ is set to $\sigma \geq 8$ there is no difference created by the frequency of task advertisement.

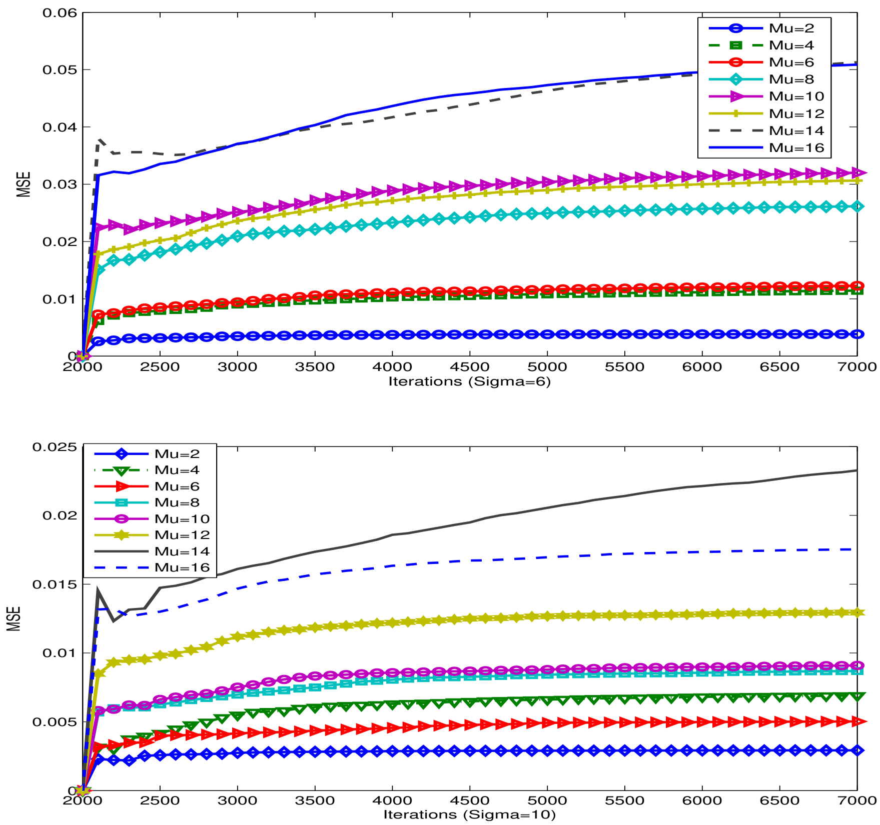

Figure 3.5: Mean Square Error of the stereotypical value judgments of agents with and without learning based on changing the μ parameter ($N = 120$). The result is shown for two different values of σ ($\sigma = 6$ on top and $\sigma = 10$ on the bottom) with varying parameter μ.

3.2.3.4 Network Structure

Here, we examine the network structure to determine the evolution of the agent society. Figure 3.7 shows the Fiedler-embedding [44] of networks in the final connectivity network of $N = 200$ agents with and without stereotype value judgments. The color and shape differences show the profile of agents. As it is clear in the **Plain** scenario the number of isolated nodes is less than the scenario with stereotype knowledge. Also in the **Plain** scenario there is no difference between the profiles, therefore we can see all type of profiles in the isolated nodes and nodes with high degree. On the other hand in the stereotype condition the agents in group 3 and 4 were more likely to become isolated and fail to use their capability to accomplish more tasks.

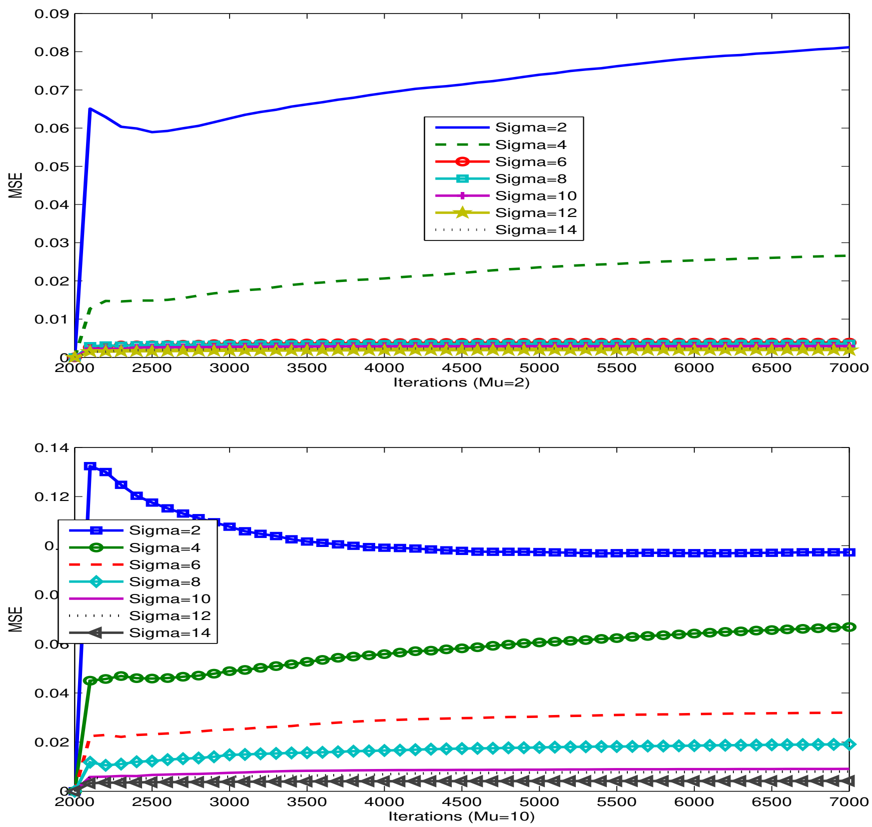

Figure 3.6: Mean Square Error of the stereotype value estimation of agents based on changing the σ parameter ($N = 120$). The result is shown for two different values of μ ($\mu = 2$ on top and $\mu = 10$ on the bottom) with varying parameter σ.

The degree-based strategy moves the structure toward being similar to a scale-free network whereas with stereotype value judgments the network becomes progressively more star-shaped.

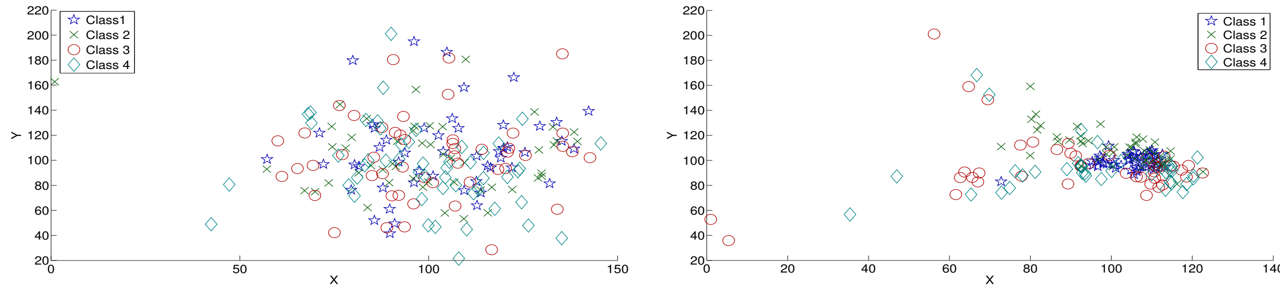

Figure 3.7: Fiedler embedding of the final network structures in non-stereotype (left) and stereotype (right) based network evolution ($N = 200$). There are more isolated nodes in Class 3 and Class 4 when we have stereotypical judgments.

3.2.3.5 Effects of Rapid Attachment Modification

Here we examine the effects of modifying the parameter P, the probability of updating the network, on the performance of the system, both with and without stereotypes. We varied this parameter from 0.1 to 0.9 with a step size 0.2. Figure 3.8 shows the performance during 5000 iterations in both strategies. As shown in the figure, the performance does not change significantly with P values before a certain threshold. After that threshold, the performance drops dramatically, as the agents spend more time updating the network than accomplishing tasks. This threshold is dependent on the total agents and number of skills required in the environment. In both conditions the task performance drops by $P = 0.7$ but in the stereotype conditions the system performance falls at an earlier iteration, after the information transmission efficiency of the network has been sabotaged by the network adaptations caused by stereotype-value judgments.

Cumulatively, these experiments illustrate that stereotype bias can negatively impact the ability of a community to effectively form task-oriented groups, if the agents make long-term network modifications based on stereotype value judgments. These long-term network modifications can be seen as representing the cumulative result of many subtle changes in people's daily routines, based on stereotype bias. Our agent-based model illustrates how the manifestation of these network changes can appear later in a group formation and task accomplishment, even if they have imperceptible effects in situations that do not require coordination. These network structure changes have more pronounced effects when the tasks become more complicated (requiring a larger pool of skills) and efficient group work is more critical. Whether these judgments are learned (based on previous experience) or are directly based on an observable value does not seem to have a significant impact in our agent-based simulation.

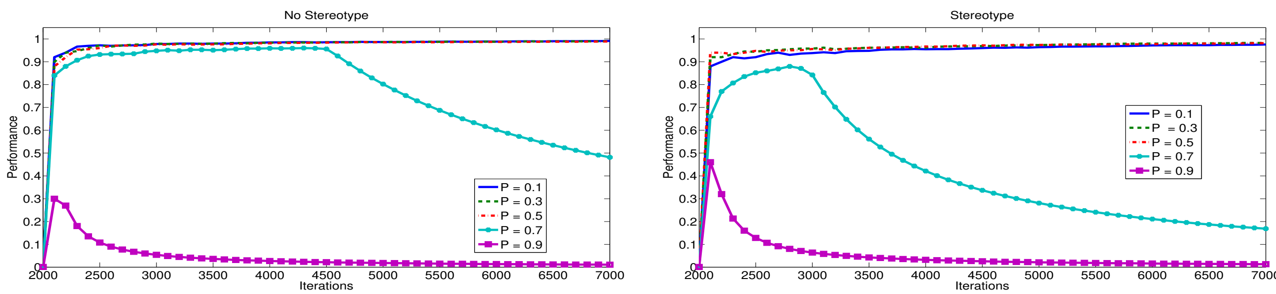

Figure 3.8: The effect of the network adaptation probability, P_i

3.3 Summary

In this chapter we introduced an agent-based simulation for examining the effects of stereotypes on task-oriented group formation and network evolution. We demonstrate that stereotype value judgments can have a negative impact on task performance, even in the mild case when the agents' willingness and ability to cooperate is not impaired. By modifying the social network from which groups are formed in a systematically suboptimal way, the stereotype-driven agents eliminate the skill diversity required for successful groups by driving the network toward specific topological configurations that are ill-suited for the task. The results show that making connections with agents solely based on group membership yields a sparser network with many isolated nodes.

Due to the technical challenges of investigating the long-term effects of stereotype across populations, we suggest our agent-based simulation method is a useful tool for investigating these research questions.

CHAPTER 4: INFLUENCE MAXIMIZATION TECHNIQUES FOR ADVERTISING

The question of how to influence people in a large social system is a perennial problem in marketing, politics, and publishing. It differs from more personal inter-agent interactions that occur in negotiation and argumentation since network structure and group membership often pay a more significant role than the content of what is being said, making the messenger more important than the message. In this part of the thesis, we propose a new method for propagating information through a social system and demonstrate how it can be used to develop a product advertisement strategy in a simulated market. In the following sections we will describe our market model, our interaction model, and the synthetic data has been generated for evaluation.

4.1 Market Model

To explore the efficiency of the proposed marketing method, we have extended a multi-agent system model, inspired by [47] and [48], to simulate a social system of potential customers. In this model, there is a population of N agents, represented by the set $A = \{a_1, \ldots, a_N\}$, that consists of two types of agents $(A = A_R \cup A_P)$. The first type of agent, defined as: $A_R = \{a_r \mid a_r \textit{ is Mutable } and \; 1 \leq r \leq R\}$, are the *Regular* agents, who are the potential customers. These agents have a changing attitude on purchasing products and can be influenced by the *Product* agents who represent salespeople offering one specific product. These agents have an immutable attitude toward a specific product and are defined as: $A_P = \{a_p \mid a_i \textit{ is Immutable } and \; 1 \leq p \leq P\}$. Figure 4.1 provides an illustration of the market model.

Each *Regular* agent can be considered as a unique node in the social network, connected by directed weighted links based on the underlying interactions with other agents. The connection between the *Regular* agents 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. The in-node and out-node degrees of agent a_i are the sum of all in-node and out-node weights, respectively ($d^i_{in} = \sum_{a_j \subseteq A_R} e_{ji}$ and $d^i_{out} = \sum_{a_j \subseteq A_R} e_{ij}$). This network is assumed to follow a power law degree distribution like many human networks, and is generated synthetically as we will explaine in Section 4.2.

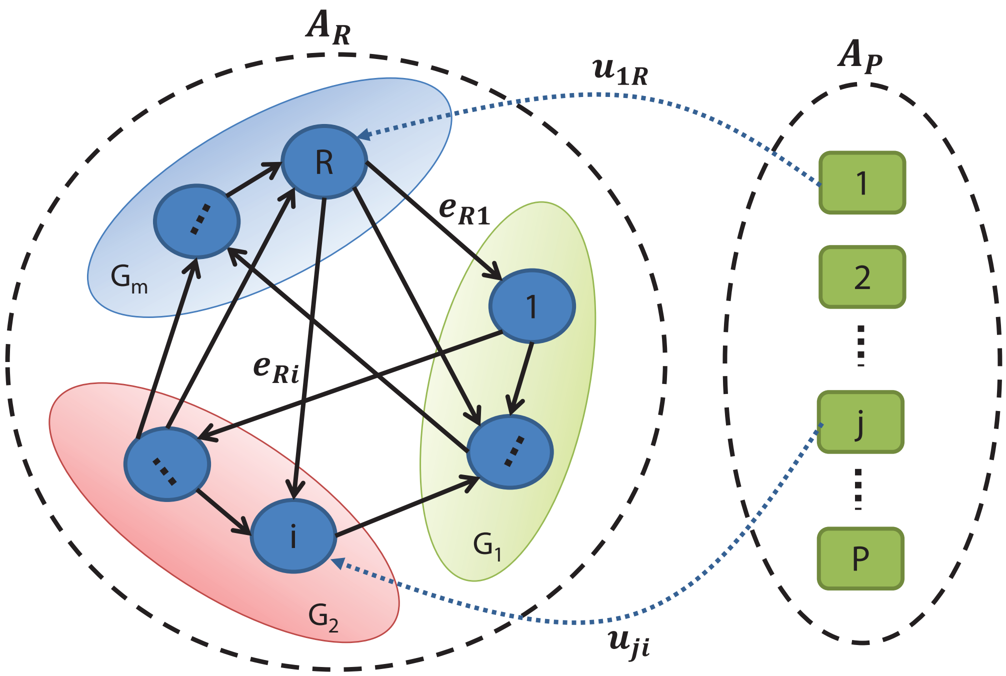

Figure 4.1: The model of the social system. There exist two types of agents, *Regular* agents (A_R) and *Product* agents (A_P). A static network exists among *Regular* agents, and our problem is to find effective connections between the *Product* (sellers) and *Regular* agents (customers) in order to influence the customers to buy products. *Regular* agents also can belong to different groups in their society (G_m), which modifies the local influence propagation properties.

We model the desire of an agent, a_i, to buy an item or consume a specific product, p, as a random variable denoted by $x_{ip} \in [-1\ 1]$. As there exist P items in the environment, each agent is assigned a vector of random variables, $\overrightarrow{X_i}$, representing the attitude or desire of the agent toward all of the products in the market.

Within the social network there are different groups of *Regular* agents; these groups could represent demographic groups or other types of subcultures. Agents from the same group are more effective at influencing each other. To model this, the social system contains m different long-lasting groups, $G_1, \ldots, G_m$, and each agent i is designated with a group membership, G_i.

Here, we do not attempt to capture a rich social-cultural behavior model of these interactions, but rather view the model simply as a function $\mathcal{F} : \mathrm{G_i} \longrightarrow S_i$, mapping the group label of agents, $\mathrm{G_i}$, to a social impression, S_i, that affects link formation and influence propagation, which we designate as the group value judgment. This value represents the agents' judgments on other groups and is based on observable group label of the agent rather than real characteristics of the person. We assume that the impression of different groups has been learnt by agents beforehand therefore each agent has a unique vector of judgment values, noted as $\overrightarrow{S_i} = S_1, S_2, \ldots, S_m$, to indicate the judgment of each agent on different groups in the simulated society.

Moreover, in real life there is a correlation between the user demand of different products in the market. The desire of customers for a specific product is related to his/her desire toward other similar products. To model this correlation and consider its effect in our formulation, we designate a matrix $\mathbf{M}$ that identifies the relationship between demands among advertised items and can be shown as:

$$\mathbf{M} = \begin{pmatrix} m_{11} & \ldots & m_{1P} \\ \vdots & \ddots & \vdots \\ m_{P1} & \ldots & m_{PP} \end{pmatrix}$$

where m_{ij} indicates the probability of having desire toward item j assuming the agent already has a desire for item i. We assume that this matrix is known beforehand and has been modeled by the advertisement companies by tracking the users and applying user modeling.

In the market, the companies are trying to select 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. We define a simple decision variable u_{ji}, where

$$u_{ji} = \begin{cases} 1 & \textit{Product } j \text{ connects to } \textit{Regular} \text{ agent } i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.1}$$

Note that the links between *Product* agents and *Regular* agents are directed links from products to agents and not in the opposite direction, and that *Product* agents will never connect to other *Product* agents. 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,

$$p_{ij} = \begin{cases} \frac{e_{ij}}{d_{out}^{i}} & i, j \in A_R \\ \frac{u_{ji}}{Threshold} & i \in A_R, j \in A_P \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

where d_{out}^{i} is the out-node degree of a *Regular* agent i and 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.

At each interaction there is a chance for agents to influence each other and change their desire vector for purchasing or consuming a product. In all these interactions *Product* agents, the immutable agents, are the only agents who do not change their attitude and have a fixed desire vector. The probability that agent j influences agent i is denoted as α_{ij} and is calculated based on the out-node degree of agent j as:

$$\alpha_{ij} = \begin{cases} \frac{e_{ji}}{d^j_{out}} & i, j \in A_R \\ cte & i \in A_R, j \in A_P \end{cases} \tag{4.3}$$

Figure 4.2 shows a simple example of how to calculate p_{ij} and α_{ij}.

The other important parameter in the agent influence process is ε_{ij}, which determines how much agent j will influence agent i. This parameter is derived from a Gaussian distribution assigned to the membership group of agent j based on the experience of agent i with this group. Therefore, this value can easily be extracted from the previously defined vector $\overrightarrow{S_i}$.

As a final note, in this model the agents can access the following information:

1. the links connecting agents that possess a history of past interactions. Each agent is aware of its connections with neighbors and their weights;

2. the group membership of neighboring agents and other select members of the community.

The ultimate goal of our marketing problem is to recognize the influential agents in the graph and define $u_{ji}s$ in a way to get the maximum benefit of the product advertising.

4.2 Synthetic Data

To evaluate the performance of proposed methods on identifying influential agents in a variety of networks, we simulate the creation of agent networks formed by the combined forces of homophily and group membership. Since social communities often form a scale-free network, whose degree distribution follows a power law [9], we model our agent networks using the network generation method described in [101]. Note that this network only connects the regular agents $(a_i \in A_R)$. The connection between the *Product* and *Regular* agents is identified later in a way to optimize the efficiency of the product marketing.

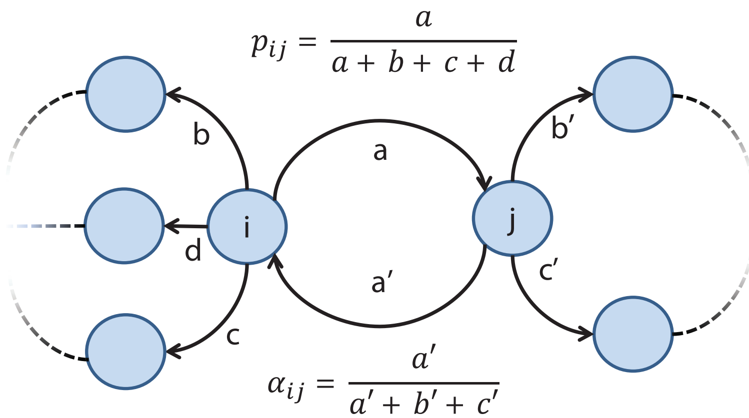

Figure 4.2: An illustration of how the probability of interaction (p) and the probability of influencing others (α) is calculated between the *Regular* agents.

Following the network data generation method in [87], we control the link density of the network using a parameter, *ld*, and value homophily between agents using a parameter, *dh*. The effects of value homophily are simulated as follows:

1. At each step, a link is either added between two existing nodes or a new node is created based on the link density parameter (*ld*). In general, linking existing nodes results in a higher average degree than adding a new node.

2. To add a new link, first, we randomly select a node as the source node, a_i, and a sink node, a_j $(a_i, a_j \in A_R)$, based on the homophily value (*dh*), which governs the propensity of nodes with similar group memberships to link. Node a_j is selected among all the candidate nodes in the correct group, based on the degree of the node. Nodes with higher degree have a higher chance to be selected.

3. If a prior link exists between agent a_i and a_j, selecting them for link formation will increase the weight of their link by one.

Table 4.1: Agent Network Generator

```
Agent Network Generator (numNodes, numLabels, ld, dh)
i = 0
E = NULL
while i < numNodes do
   sample r from uniform distribution U(0,1)
   if r ≤ ld then
      connectNode(E,numLabels,dh)
   else
      addNodes(E,numLabels,dh)
      i = i + 1
   end if
end while
return E
```

Group membership also governs the process of reciprocal link formation. Once the link generation process starts and the source and sink nodes have been selected, we add a directed link from node a_i to node a_j by default, under the assumption that the first selected agent initiated the interaction. The group value judgment of the second node governs whether a reciprocal link is formed or not. We use an evaluation function $F_a(S)$ to map an observed group value S to a binary evaluation of interaction (positive or negative). We assume that all agents use the same evaluation function, which is:

$$F_a(S) = \begin{cases} 1: & S \geq 0.5 \\ -1: & S < 0.5 \end{cases}$$

The result of this process is to create clusters of agents with the same group labels within the network, since group membership affects both the probability of the initial interaction (through the homophily parameter) and also the reciprocal link formation.

To generate a new node, we first select a group label based on a uniform group distribution and assign that group label to the node. Then we add links between the new node and one of the existing nodes as we described above. The algorithm for generating the static network is outlined in Table 4.1.

4.3 Dynamics in the Market

As explained in Section 4.1, the agent i's desire toward product p, is modeled as a random variable that assumes a scalar value after each interaction $(x_{ip} \in [-1\ 1])$. Therefore, since there exist P different products, each agent has a vector of random variables, $\overrightarrow{X_i}$, which indicates the desire of the agent toward all the available products in market. Following Hung et al. [47, 48], we model the desire dynamic of all agents as a Markov chain where the state of the system is a matrix of all agents' desire vectors at a particular iteration k and the state transitions are calculated probabilistically from the pair-wise interaction between agents connected in a network. The state of the system at the k^{th} iteration is a vector of random variables, denoted as $\mathbf{X}(k) \in \mathbb{R}^{NP\times 1}$ (created through a concatenation of N vectors of size P) and expressed as:

$$\mathbf{X}(k) = \begin{pmatrix} [\overrightarrow{X_1}(k)] \\ \vdots \\ [\overrightarrow{X_N}(k)] \end{pmatrix}$$

4.3.1 Generalized ICM

The independent cascade model presented by Kempe et. al [53] defines the interaction between agents as a cascade process which at each step the recently activated nodes have a chance to activate their neighbors independently. Although this model has been successfully used in many domains, it has the following limitations in the marketing domain:

1. In ICM the probability of interaction between agents is either equal to 1 or 0 depending on what group of agents get activated at each time step. When a node gets activated the probability of interaction between that node and its neighbors switches to 1 in the next time step, while it is equal to 0 at any other time step. This condition cannot simulate the latency

in real-world interactions in which an agent purchases a product and then after some time influences its friends' perception of the product.

2. In the IC model, in the case of interactions between an activated node i and its neighbors, the probability of influencing or activating a neighbor is a binary situation as well. Either the neighbor is completely persuaded and becomes activated or denies any influence and remains deactivated. This is not true in real world interactions where partial influence is more common.

3. The influence propagation in IC model assumes progressive activation–once an agent gets activated or influenced, it cannot change its mind or switch to another state. Therefore, it remains activated for the rest of the simulation. This assumption implies that a costumer is unable to change his mind after choosing a product in the market [62]. Again this assumption does not match with the real world situation where consumers can change their mind at any time and switch back to their previous decision repeatedly. Hence, the IC model cannot represent the situation in real business market accurately.

As a result, in this section a generalized version of ICM is used to have a more realistic interaction model based on the model introduced in [69, 48]. The dynamics of the model at each iteration k proceed as described in [69, 48]:

1. Agent i initiates the interaction on 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:

$$\begin{cases} \overrightarrow{X_i}(k+1) = \varepsilon_{ij}\,\mathbf{M}\overrightarrow{X_i}(k) + (1-\varepsilon_{ij})\,\mathbf{M}\overrightarrow{X_j}(k) \\ \overrightarrow{X_j}(k+1) = \overrightarrow{X_j}(k) \end{cases} \tag{4.4}$$

  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:

$$\begin{cases} \overrightarrow{X_i}(k+1) = \overrightarrow{X_i}(k) \\ \overrightarrow{X_j}(k+1) = \overrightarrow{X_j}(k) \end{cases} \tag{4.5}$$

It is worth to note that in above interaction 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, the model can be degraded to IC model. Also as the values of desire vector belongs to $[-1\ 1]$, the $x_{ip}s \in [0\ 1]$ and $x_{ip}s \in [-1\ 0]$ should be quantized to 1 and 0 respectively to have the similar representation of activation and deactivation in IC model.

4.3.2 Interaction and Influence

In this work, we define interactions as any kind of information or belief sharing between two agents about the available products in the market. During these interactions, there is a possibility for one agent to influence the desire of the other one. As explained in Section 4.1, this possibility is modeled by parameter α_{ij} when agent i initiates the interaction with agent j. Also, in this interaction, we assume that the influenced agent will retain some fraction of its existing desire. This fraction is different for any single agent i while interacting with agent j, but remains fixed, and is denoted as $\varepsilon_{ij} \in [0\ 1]$. The dynamics of the model at each iteration k proceed as follows:

1. Agent i initiates the interaction on 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:

   $$\begin{cases} \overrightarrow{X_i}(k+1) = \varepsilon_{ij}\,\mathbf{M}\overrightarrow{X_i}(k) + (1-\varepsilon_{ij})\,\mathbf{M}\overrightarrow{X_j}(k) \\ \overrightarrow{X_j}(k+1) = \overrightarrow{X_j}(k) \end{cases} \tag{4.6}$$

     Recall that 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:

   $$\begin{cases} \overrightarrow{X_i}(k+1) = \overrightarrow{X_i}(k) \\ \overrightarrow{X_j}(k+1) = \overrightarrow{X_j}(k) \end{cases} \tag{4.7}$$

To analyze Equation 4.6 in detail, we rewrite the matrix calculation for agent i as follows:

$$\overrightarrow{X_i}(k+1) = \begin{pmatrix} \sum_{f=1}^{P} m_{1f}\left(\varepsilon_{ij}x_{if} + (1-\varepsilon_{ij})\,x_{jf}\right) \\ \vdots \\ \sum_{f=1}^{P} m_{Pf}\left(\varepsilon_{ij}x_{if} + (1-\varepsilon_{ij})\,x_{jf}\right) \end{pmatrix} \tag{4.8}$$

A closer look at each row of $(\overrightarrow{X_i}(k+1))$ reveals that the desire of agent i toward a product depends on own previous desire, a fraction of the other agent's desire toward that product, and the desire of both agents toward other available products in the market. This is an interesting result showing how our proposed model can express the complexity of real-world markets and capture the dependency of demand for different products [60].

4.4 Optimization Technique for IM

4.4.1 Expected Long-term Desire

In this work, we determine the long-term desire of the agents for products in the system to find the optimized connection between the *Product* agents and *Regular* agents. In other words, we hypothesize that by examining the expected value of the steady state system $(\mathbf{X}(k))$, we are able to optimize the marketing strategy and identify the most influential nodes in the network. Therefore our goal in this section is to calculate the expectation vector of the system state since it captures all the interactions and the dependencies between the demand of the products.

The conditional expected value of the desire vector of agent i in a single pair-wise interaction between agents i and j, when the current state of the system is observed:

$$\begin{aligned} E[\overrightarrow{X_i}(k+1)|\mathbf{X}(k),j] &= (1-\alpha_{ij})\overrightarrow{X_i}(k) + \alpha_{ij}\left[\varepsilon_{ij}\mathbf{M}\overrightarrow{X_i}(k) + (1-\varepsilon_{ij})\mathbf{M}\overrightarrow{X_j}(k)\right] \\ &= (1-\alpha_{ij})\overrightarrow{X_i}(k) + \alpha_{ij}\varepsilon_{ij}\mathbf{M}\,\overrightarrow{X_i}(k) + \alpha_{ij}(1-\varepsilon_{ij})\mathbf{M}\,\overrightarrow{X_j}(k) \\ &= [\alpha_{ij}\varepsilon_{ij}\mathbf{M} + (1-\alpha_{ij})\mathbf{I}]\;\overrightarrow{X_i}(k) + \alpha_{ij}(1-\varepsilon_{ij})\mathbf{M}\,\overrightarrow{X_j}(k) \end{aligned} \tag{4.9}$$

By defining matrix $\mathbf{W}(i,j) = \alpha_{ij}(1-\varepsilon_{ij})\,\mathbf{M}$, we rewrite Equation 4.9 in the form of:

$$E[\overrightarrow{X_i}(k+1)|\mathbf{X}(k),j] = \overrightarrow{X_i}(k) + \mathbf{W}(i,j)\overrightarrow{X_j}(k) - [\mathbf{W}(i,j) + \alpha_{ij}(\mathbf{I}-\mathbf{M})]\,\overrightarrow{X_i}(k) \tag{4.10}$$

Therefore, based on the probability of interaction between two agents ($\frac{1}{N}(p_{ij}+p_{ji})$), the desire of *Regular* agents dynamically changes as specified in Equation 4.9. It is worthwhile to mention that matrix $\mathbf{W}$ is a factor of matrix $\mathbf{M}$, and it has the same dimensions of $P \times P$. Rewriting the dynamics of $\overrightarrow{X_i}$ in this way indicates that the desire vector of agent i at iteration $(k+1)$ is equivalent to its own desire plus the weighted desire of agent j at iteration k, minus its own weighted desire at that iteration. This finding shows that, in spite of having the extra matrix $\mathbf{M}$, extracted from the marketing situation, and a complicated notion of the agents' desire vector, the computation model simply follows [47], although the optimization approach must account for multiple product interactions.

We substitute $\mathbf{W}(i,j) + \alpha_{ij}(\mathbf{I} - \mathbf{M}) = \mathbf{S}(i,j)$, where $\mathbf{S}(i,j)$ again is dimension $P \times P$. Then, Equation 4.10 can be simplified as follows:

$$E[\overrightarrow{X_i}(k+1)|\mathbf{X}(k), j] = \overrightarrow{X_i}(k) - \mathbf{S}(i,j)\overrightarrow{X_i}(k) + \mathbf{W}(i,j)\overrightarrow{X_j}(k) \tag{4.11}$$

Next, we write the expected value of agent i's desire vector at iteration $(k+1)$ over all the possible interactions it initiates or is subject to by other agents' actions, conditioned on the state of the system at k. Recall that the interaction between i and j occurs with probability $\frac{1}{N}(p_{ij} + p_{ji})$.

$$\begin{aligned} E[\overrightarrow{X_i}(k+1)|\mathbf{X}(k)] = \overrightarrow{X_i}(k) &- \sum_j \frac{1}{N}(p_{ij} + p_{ji})\, \mathbf{S}(i,j)\overrightarrow{X_i}(k) \\ &+ \sum_j \frac{1}{N}(p_{ij} + p_{ji})\, \mathbf{W}(i,j)\overrightarrow{X_j}(k) \end{aligned} \tag{4.12}$$

Now, we want to express the expected desire of all agents at iteration $(k+1)$ conditioned on all agents' previous desire. This step relies on both the laws of interacting expectations and linearity of expectations. Assembling a vector of all entries for each i results in:

$$E[\mathbf{X}(k+1)|\mathbf{X}(k)] = \mathbf{X}(k) + \mathbf{Q}\mathbf{X}(k) \tag{4.13}$$

where $\mathbf{Q}$ is a block matrix and each component of $\mathbf{Q} \in \mathbb{R}^{N \times N}$, considering Equation 4.12, is:

$$Q_{ij} = \begin{cases} \frac{1}{N}(p_{ij} + p_{ji})\mathbf{W}(i,j) & i \in A_R, j \in A \text{ and } i \neq j \\ \\ -\frac{1}{N}\sum_j (p_{ij} + p_{ji})\mathbf{S}(i,j) & i \in A_R, j \in A \text{ and } i = j \\ +\frac{1}{N}(p_{ij} + p_{ji})\mathbf{W}(i,j) & \\ \\ 0 & i \in A_P, j \in A \end{cases} \tag{4.14}$$

Finally, by calculating the expected value of Equation 4.13 and using the linearity of expectations, we have:

$$E[E[\mathbf{X}(k+1)|\mathbf{X}(k)]] = E[\mathbf{X}(k+1)] = E[\mathbf{X}(k)] + \mathbf{Q}\, E[\mathbf{X}(k)] \tag{4.15}$$

We define $\overrightarrow{\mu}_{\mathbf{X}}(k) \in \mathbb{R}^{NP \times 1}$ as the expected value vector of $\mathbf{X}(k)$. Therefore, the above equation is simplified as:

$$\overrightarrow{\mu}_{\mathbf{X}}(k+1) = \overrightarrow{\mu}_{\mathbf{X}}(k) + \mathbf{Q}\, \overrightarrow{\mu}_{\mathbf{X}}(k) \tag{4.16}$$

Since we are seeking the expected value of $\mathbf{X}(k)$ at steady state, the above equation when $k \to \infty$ reduces to:

$$\overrightarrow{\mu}_{\mathbf{X}}(\infty) = \overrightarrow{\mu}_{\mathbf{X}}(\infty) + \mathbf{Q}\, \overrightarrow{\mu}_{\mathbf{X}}(\infty) \Rightarrow \mathbf{Q}\, \overrightarrow{\mu}_{\mathbf{X}}(\infty) = 0 \tag{4.17}$$

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

$$\mathbf{Q} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ 0 & 0 \end{pmatrix} \text{ and } \overrightarrow{\mu}_{\mathbf{X}}(\infty) = \begin{pmatrix} \overrightarrow{\mu}_{\mathbf{R}} \\ \overrightarrow{\mu}_{\mathbf{P}} \end{pmatrix} \tag{4.18}$$

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 4.3 shows the breakdown of matrix $\mathbf{Q}$.

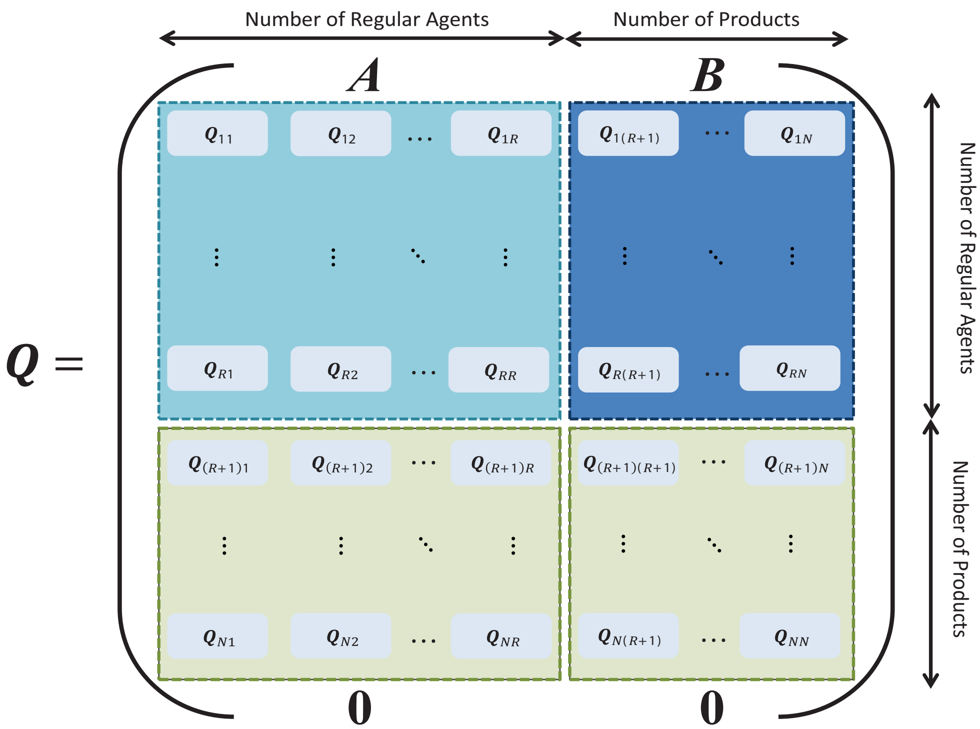

Figure 4.3: Q matrix is a block matrix with size $N \times N$ where N is the total number of agents $(R + P)$ and each block has the size of $P \times P$. Matrices A and B are the non-zero part of this matrix which represent the interactions among *Regular* agents and interactions between *Regular* agents and *Products*, respectively.

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

$$\mathbf{A}\,\overrightarrow{\mu}_{\mathbf{R}} + \mathbf{B}\,\overrightarrow{\mu}_{\mathbf{P}} = 0 \Rightarrow \overrightarrow{\mu}_{\mathbf{R}} = \mathbf{A}^{-1}(-\mathbf{B}\,\overrightarrow{\mu}_{\mathbf{P}}) \tag{4.19}$$

Now based on this analytical view of the system, we define an optimization method in following section to maximize the product sales through intelligent selection of the *Product* agent linkages.

4.4.2 Node Selection Method

Using the analysis from the previous section, 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. Here, 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:

$$max_u \sum_{1\leq k\leq P} \sum_{i\in A_R} (\rho_i . \overrightarrow{\mu}_{\mathbf{R},i}) \tag{4.20}$$

$\overrightarrow{\mu}_{\mathbf{R},i}$ 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.

The goal of our proposed method is to assign a fixed number of *Product* agents with limited number of connections to a network of *Regular* agents in a way to optimize the objective function presented above. In Equation 4.19, matrix $\mathbf{A}$ and vector $\overrightarrow{\mu}_{\mathbf{P}}$ are known since the static network among the *Regular* agents and the fixed desire vector of the products are both known. We define the matrix $\mathbf{B}$ based on parameters of u_{ij}s. We substitute the probability of interaction, p_{ij}, occurring between agents i and j in matrix $\mathbf{Q}$, by Equation 4.2 of the model.

The partitioning of matrix $\mathbf{Q}$ in Equation 4.18 and the size of matrices $\mathbf{A}$ and $\mathbf{B}$ (Figure 4.3), indicates that the elements of matrix $\mathbf{B}$ are all off the diagonal. Therefore substituting the values of p_{ij} and p_{ji} of Equation 4.2 into Equation 4.14, $\mathbf{B}_{ij} = \frac{1}{N}\, u_{ji}\mathbf{W}(i,j) = \widehat{u} \otimes \mathbf{M}$. Here, $\widehat{u}$ contains all the variables and influence parameters and $\otimes$ indicates the Kronecker product [70].

Therefore, by rewriting Equation 4.19 as:

$$\overrightarrow{\mu}_{\mathbf{R}} = \mathbf{A}^{-1}[\widehat{u} \otimes \mathbf{M}]Vec(\widehat{\mu}_{\mathbf{P}}) \tag{4.21}$$

and using the following identity

$$[\widehat{u} \otimes \mathbf{M}]\; Vec(\widehat{\mu}_{\mathbf{P}}) = Vec(\mathbf{M}\; \widehat{\mu}_{\mathbf{P}}\; \widehat{u}),$$

Equation 4.19 becomes $\overrightarrow{\mu}_{\mathbf{R}} = \mathbf{A}^{-1}Vec(\mathbf{M}\; \widehat{\mu}_{\mathbf{P}}\; \widehat{u})$, which is solved using convex optimization methods. Therefore the optimal assignment of *Product* agents to *Regular* agents is obtained through the following optimization problem:

$$\begin{aligned} \underset{\widehat{u}}{\text{maximize}} \quad & \|\mathbf{A}^{-1}Vec(\mathbf{M}\; \widehat{\mu}_{\mathbf{P}}\; \widehat{u})\|_1 \\ \text{subject to} \quad & x_{ip} \in [-1\ 1],\ \forall i \in A_R, \\ & \sum_{j \in A_R} u_{ij} = cte. \end{aligned} \tag{4.22}$$

To solve this optimization problem we used the CVX toolbox of Matlab which is useful for convex programming and minimized the dual of our objective function.

4.4.3 Experimental Setup

We conducted a set of simulation experiments to evaluate the effectiveness of our proposed node selection method on marketing the items in a simulated social system with a static network.

The parameters of the model for all the runs are summarized in Table 4.4(a). All the results are computed over an average of 30 runs with 100 *Regular* agents and 10 *Product* agents.

In this work, we model four long-lasting groups, ($G_1, \ldots, G_4$), with different feature vector distributions in our social simulation. Moreover, a group value judgment, (S_i), assigned to each group, is drawn from Gaussian distribution. We assumed that the group model has been learned by agents based on their previous experiences, each agent has its own fixed value judgment toward each group of agents and that value has been selected based on the assigned Gaussian distribution of the model. Consequently, this group value judgment affects the connection of agents during the network generation phase, as we described before. Table 5.1(b) shows the mean and standard deviations of the Gaussian distributions assigned to each group. Note that the membership in each group is permanent for all agents and cannot be changed during the course of one simulation.

In the *Regular* and *Product* agent interaction, parameters α and ε are fixed for any interaction and are presented in Table 4.4(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 4.2 and are parametric.

Table 4.2: Parameter settings

(a) Experimental parameters

Parameter	Value	Descriptions
R	100	Number of *Regular* agents
P	10	Number of *Product* agents
$Threshold$	2	Number of links between P and R agents
ε	0.4	Influence factor between P and R agents
α	0.6	Probability of influence between P and R agents
$N_{Iterations}$	10000	Number of iterations
N_{Run}	30	Number of runs

(b) Group model

Group	Mean Value	StDev
G1	0.9	0.05
G2	0.6	0.15
G3	0.4	0.15
G4	0.3	0.1

Finally, the remaining part of the social system setup is 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 one item, is equal to one.

4.4.4 Results

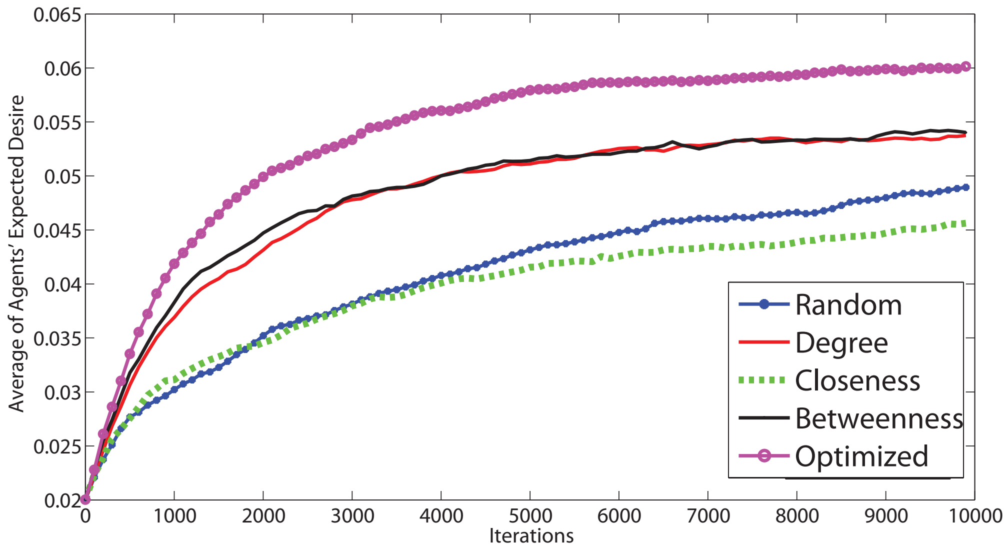

Figure 4.4: The average of agents' expected desire vs. the iterations. The average is across all the products and over 30 different runs. Our proposed method has the highest average in comparison to other methods which shows its capability as a method for targeted advertisement in a social system.

We compare our optimization-based algorithm with a set of centrality-based measures commonly used in social network analysis for identifying influential nodes based on network structure [53]. The comparison methods are:

Degree Assuming that high-degree nodes are influential nodes in the network is a standard approach for social network analysis. Here, we calculated the probability of joining a *Regular* agent based on the out-degree of the agents and attached the *Product* agents according to preferential attachment. Therefore, nodes with higher degree had an increased chance of being selected as an advertising target.

Closeness This is another commonly used influence measure in sociology, based on the assumption that a node with short paths to other nodes has a higher chance to influence them. Here, we averaged the shortest paths of a node to all the other nodes in the network and sorted the nodes according to this measure. Nodes with shorter average path had a higher chance of being selected as a 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 probability of being selected.

Random Finally, we consider selecting the nodes uniformly at random as a baseline.

To evaluate these methods, we started the simulation with an initial desire vector set to 0.02 for all agents, and simulated 10000 iterations of agent interactions. The entire process of interaction and influence is governed based on the previous formulas given in Section 4.3.2 and extracted parameters from the network. At each iteration, we calculated the average of the expected desire value of agents toward all products. Figure 4.4 shows this result for 100 agents and 10 advertisements. As explained before, the desire vector of *Product* agents are fixed for all products; in our simulation is was set to 1 for the product itself and -0.05 for all other products (e.g., $\mu_2 = [-0.05\,1\,-0.05\ldots\,-0.05]$). The results for this condition show that the proposed method creates a higher total product desire in the social system and is more successful than other methods at selecting influential nodes.

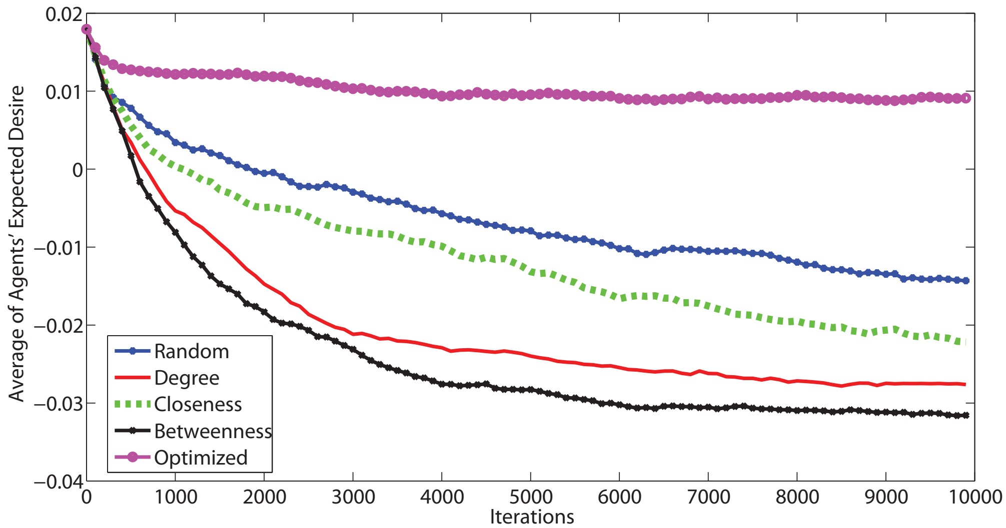

Figure 4.5: The average of agents' expected desire vs. iterations. In this simulation, the negative effect of advertising products against other products has been increased. This result demonstrates that our proposed method is more robust to the commonly occurring condition where increasing the desire toward one item has a higher negative effect on the desire of agent toward other products.

To test the robustness of our algorithm we modified the desire vector of *Product* agents and increased the negative effect of advertisements over other products by factor of three (e.g., $\mu_2 = [-0.15\ 1\ \ -0.15\ldots\ \ -0.15]$). The result of this simulation is shown in Figure 4.5. We can see that in this case the average desire of agents has dropped dramatically for all methods except the proposed algorithm. Even in the cases of having high negative effect toward other products, this algorithm can adapt the node selection in a way to keep the desire of agents high and sell more products.

To estimate the performance of algorithms in selling the products to *Regular* agents, we assumed that agents with expected desire higher than a threshold will purchase the product. Figure 4.6 shows the average of total purchased items by agents with the purchasing threshold as 0.01. Again, we see that our proposed algorithm is the most successful method in advertising and selling products.

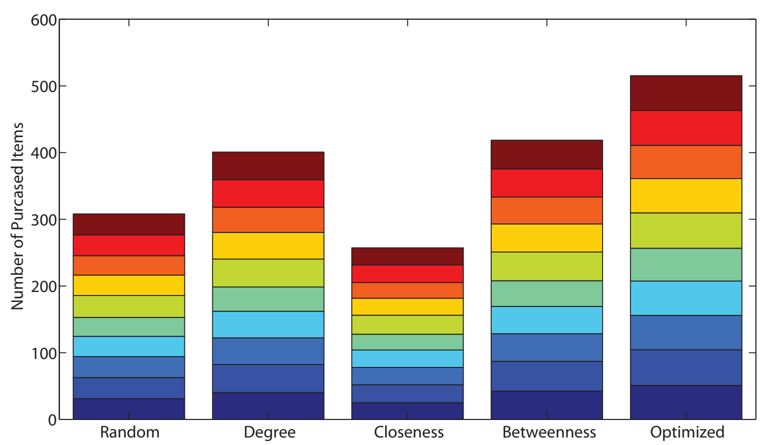

Figure 4.6: The number of sold items vs. different advertising methods. The assumption is that an agent with expected desire greater than 0.01 will purchase the product. Different colors in each bar indicates the number of sold items of each advertised products. As there exist ten different products, the bar is divided into ten parts.

4.5 Hierarchical Influence Maximization

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 section, we present a hierarchical influence maximization approach for product marketing that constructs an abstraction hierarchy for scaling the optimization technique presenting in Section 4.4 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.

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 4.7 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.

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 4.3. 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.

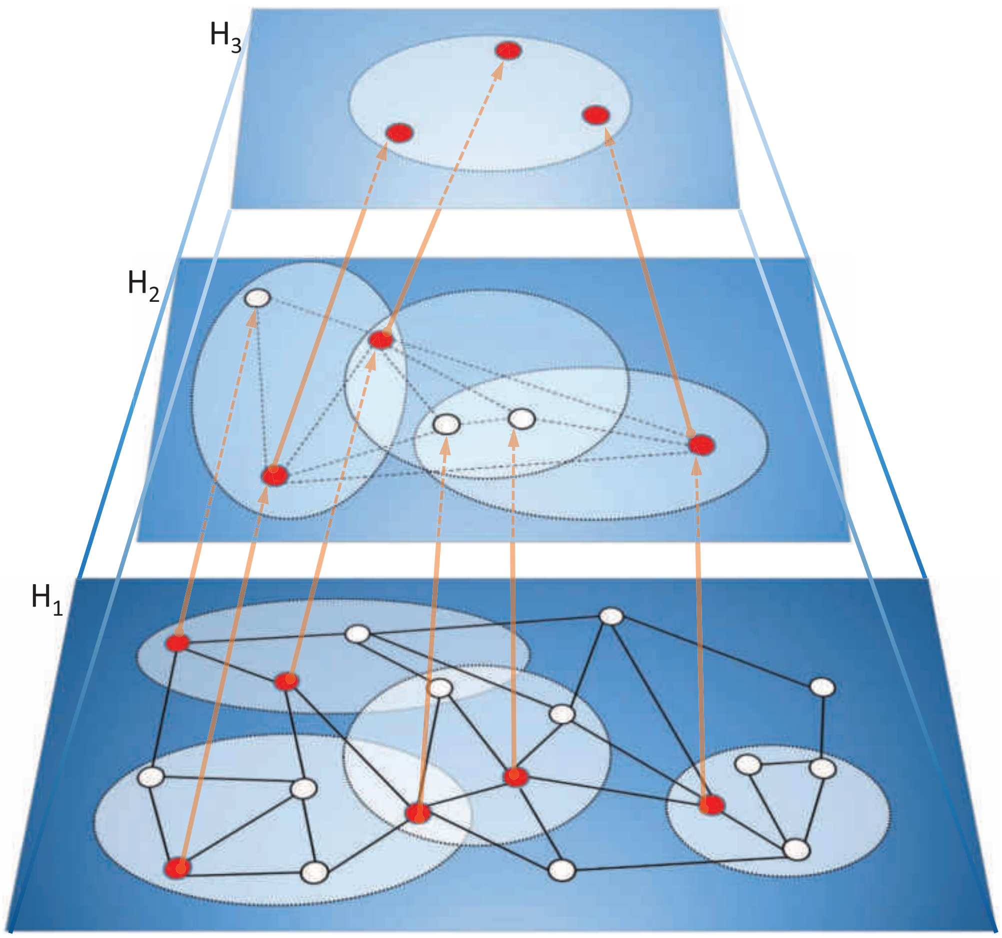

Figure 4.7: At each hierarchical level (H_i) local neighborhoods are created and influential nodes (red) are selected using an optimization technique. Nodes that have been selected at least once as an influential node are transferred to the next level of the hierarchy. At the higher levels, the connection 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.

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 4.5.2. Then we add these influential nodes to the set of influential nodes that have been identified in other neighborhoods in the same hierarchy.

Table 4.3: HIM Algorithm

```
HIM (Agent, E, P, A, A_R, H_max, r)
H = 0
E^H = E
N^H = |A_R|
While stopCriteria do
   H = H + 1
   infList = NULL
   for i = 1 to N^H do
      neighborList = FindNeighborList (i, r, E^H)
      E_i^H = Subgraph (neighborList, E^H)
      E_i^H = AddOutsideWorld (E^H, E_i^H)
      (P_i, A_i) = UpdateMat (E^H, P, A, neighborList )
      L = Optimize (Agent, E_i^H, P_i, A_i)
      infList = infList ⋃ L
      Agent = UpdateAgent (infList)
   end for
   N^H = |infList|
   U = MakeU (Agent)
   stopCriteria = UpdateCriteria (infList, H)
   E^H = UpdateHierarchy (infList)
end while
return U
```

4.5.1 Outside World Effect

When a local neighborhood is detached from the complete network, there exist some boundary nodes which 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 work, 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 4.8 illustrates the abstraction of outside world effect and shows how the model's parameters are calculated between each boundary and virtual node.

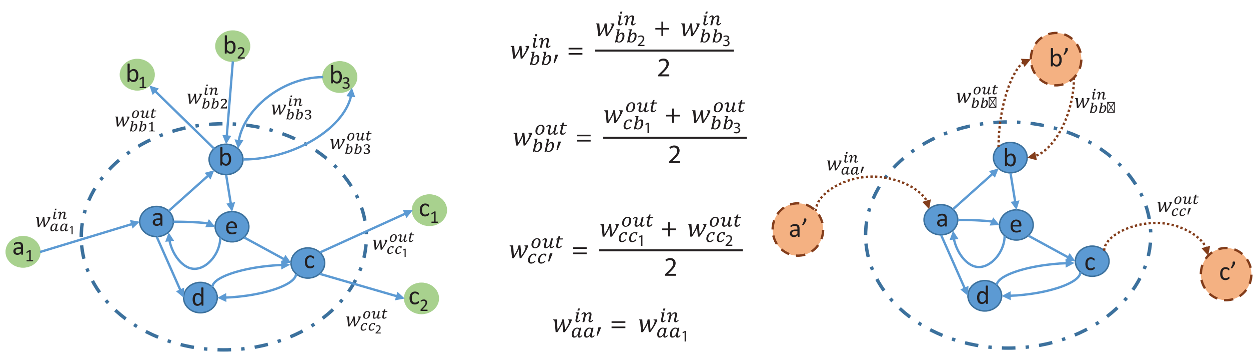

Figure 4.8: The network on the left is an example of a neighborhood around node e; the network on the right is the equivalent network with virtual nodes representing the outside world effect. Here w can be any interaction parameter such as link's weight, α, or ϵ. The direction of the interaction with 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.

4.5.2 *Node Selection*

The process of selecting influential nodes is repeated at each hierarchy and at each local neighborhood surrounding node i. Following previous works [47, 48, 69], 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 pair-wise 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:

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

Using the method described in Section 4.4 for calculating the expectation of all agents' desire vector according to the possibility of an interaction, 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:

$$E[\mathbf{X_i}(k+1)] = E[\mathbf{X_i}(k)] + \mathbf{Q_i}\ E[\mathbf{X_i}(k)] \tag{4.23}$$

where $\mathbf{Q_i}$ is a block matrix representing the interactions among *Regular* agents in the neighborhood and interactions between the *Regular* agents and all the *Products*.

4.5.3 Convergence

In the previous section, we showed how Equation 4.23 can be solved at the steady state and in a global fashion, without giving any guarantee that the state of the system actually reaches the steady state. Here, by using Brouwer fixed-point theorem [59], we prove that each local neighborhood has a fixed-point and solving Equation 4.23 at steady state is a valid choice.

The Brouwer fixed-point 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 4.23, $E[\mathbf{X_i}(k+1)]$ is a continuous function as it is the sum of two continuous ones. Also since $\overrightarrow{X_i}(k+1)$ in Equation 4.6 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. Consequently, we can follow all the calculations of [69] and solve our problem with the proposed optimization algorithm to find the assignment of u_jis in a way to maximize the long-term expected desire vector of agents toward all the products in the market.

4.5.4 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.

4.5.5 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.

4.5.6 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 the runs are summarized in Table 4.4(a). All the 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 4.4(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 4.2 and are parametric. Table 4.4(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.

4.5.7 Results

We compare our hierarchical algorithm with the original optimization method (named OIM) described in [69] and a set of centrality-based measures commonly used in social network analysis for identifying influential nodes based on network structure [53]. The comparison methods are:

Table 4.4: 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

- **OIM:** The Optimized Influence Maximization method, described in Section 4.4, finds the influential nodes globally by using a convex optimization method over the entire 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 4.6 and 4.7 (Section 4.3.1). 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). 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., $\mu_1 = [1\ -0.1\ -0.1 \ldots\ -0.1]$). We used the same network generation technique described earlier for generating customer networks.

4.5.7.1 Performance

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 4.9. Here we selected 150 agents as an optimal number of agents to compare all the algorithms together. With a lower number of agents the assignment of 10 products can not illustrate the potential differences among the methods while with a higher number of agents OIM suffers from scalability issues and the convex optimization method was not feasible due to near singular interaction matrix. In Figure 4.9, 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.

Figure 4.10 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 the ability to scale up linearly to higher number of nodes provides a sub-optimal and yet practical solution in selecting the influential nodes in large networks.

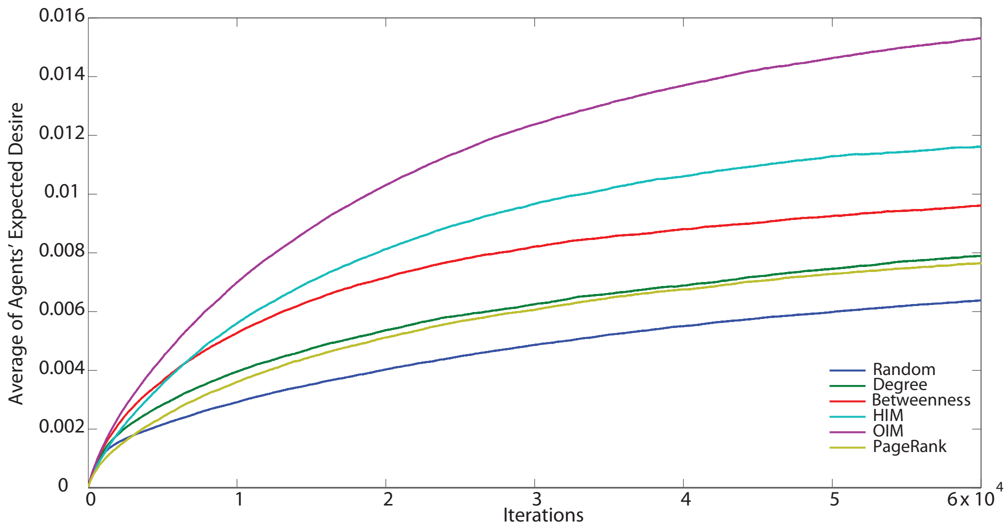

Figure 4.9: The average of agents' expected desire vs. number of iterations, calculated across all products and over 100 runs (10 different runs on 10 different networks). The optimization methods have the highest average in comparison to the centrality measurement heuristics. As the HIM algorithm is a sub-optimal method, its performance is less than the global optimization method.

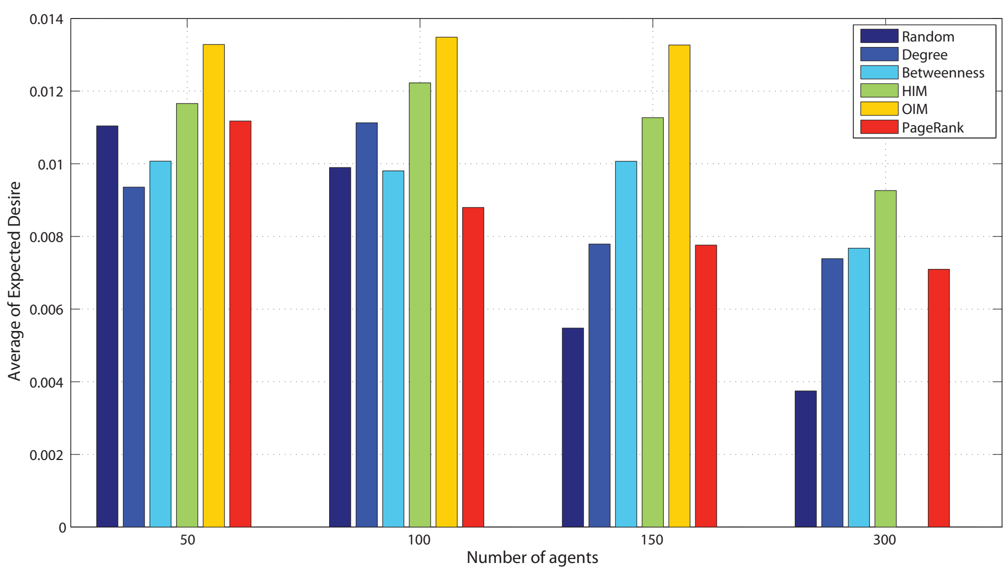

Figure 4.10: The average of the final expected desire vectors for different numbers of $Regular$ agents and 10 $Product$ agents. The optimization based methods (OIM and HIM) outperforms the other methods in selecting the seed nodes. While OIM is more successful than HIM in selecting the influential nodes, it is unable to scale-up to networks with 300 agents and higher.

Table 4.5: 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

4.5.7.2 Runtime

Table 4.5 shows a runtime comparison between the two optimization methods, HIM (hierarchical) 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.

4.5.7.3 Jaccard Similarity

To analyze the differences between the algorithms' selection of influential nodes, we use the Jaccard similarity measurement. This measurement is calculated by dividing the intersection of two selected sets by the union of these sets. Figure 4.11 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.

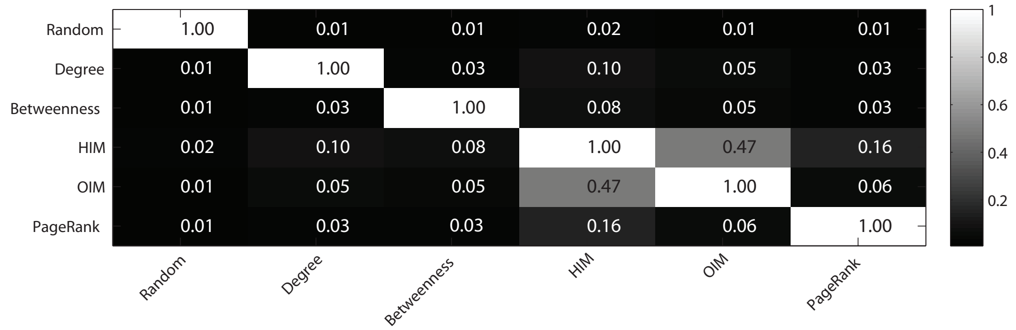

Figure 4.11: The average Jaccard similarity measurements between different methods, calculated over 100 runs (10 runs on 10 different networks). Lighter squares denote greater similarity between a pair of algorithms. Note that HIM's selection of nodes is fairly close to OIM's optimal selection.

4.5.8 Summary

In this section, we present a general hierarchical approach for applying optimization techniques to influence maximization and demonstrate its use for product marketing. 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. One possible extension of this work is to generalize the market simulation to explicitly model the adversarial effects between competing advertisers as a Stackelberg competition. Also in this work 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. [62].

CHAPTER 5: EVALUATION OF HIM ON SOCIAL MEDIA DATASETS

5.1 Increasing the Number of Benchmarks

In the previous chapter we only evaluated our algorithm against centrality measurement methods such as betweenness and degree. Although our proposed algorithms were successful against these centrality measurements, we need to compare it with other influence maximization approaches that have been successful with the LTM and ICM propagation models. For our evaluation, we selected two state of the art influence maximization methods, Prefix excluding Maximum Influence Arborescence (PMIA) and DegreeDiscount, which we describe in the next two sections.

5.1.1 PMIA Algorithm

This scalable heuristic algorithm has been presented by Wang et.al [100] and with its submodular approach, it looks at the network locally with considering the local neighborhood around each node based on the influence radius parameter. The influence radius parameter is an adjustable parameter to control the balance between the running time and the influence spread of the algorithm. PMIA algorithm finds the influence pattern in a local arborescence and then ultimately, estimates the influence propagation in the network. To our knowledge, this algorithm is the best scalable solution to the influence maximization problem in ICM.

5.1.2 DegreeDiscount Algorithm

Degree is frequently used for selecting seeds in influence maximization. Experimental results have shown that selecting vertices with maximum degrees as seeds results in larger influence spread than other heuristics, but is still not as large as the influence spread produced by the greedy algorithms.

The DegreeDiscountIC heuristic algorithm, presented by Chen et al. [20], matches the performance of the greedy algorithms for the IC model, while also improving upon the pure degree heuristic in other cascade models. It basically refines the degree method by discounting the degree of the nodes whenever their neighbor has already been selected as an influential node.

5.2 Using Real-world Datasets

One of the goals of this work was to run the proposed algorithms networks extracted from social media datasets. Therefore, in addition to the synthetic dataset, we also examined the performance and scalability of the HIM algorithm on real-world networks from the Stanford Network Analysis Project (SNAP) library. The advantage of having real-world datasets is the huge size of their networks in addition to the realistic structure of the network which has emerged from user interactions. Based on our model, among all datasets available on SNAP website, the ones with directed links are the best for evaluating our method. We evaluated our method on the following datasets:

- **WikiVote** is a network that contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent Wikipedia users and a directed edge from node i to node j represents that user i voted on user j.

- **Epinions** is a who-trust-whom online social network from a general consumer review site Epinions.com. In this network nodes are members of the site and a directed edge from i to j means j trusts i (and thus i has influence to j).

- **SlashDots** is a technology-related news website known for its specific user community. The website features user-submitted and editor-evaluated technology oriented news. In 2002 Slashdot introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes. The network cotains friend/foe links between the users of Slashdot. The network was obtained in February 2009.

Table 5.1: Statistics of the Real-world Networks

(a) Before Pre-processing

Dataset	WikiVote	SlashDot	Epinion
$\#Nodes$	7K	82K	76K
$\#Edges$	100K	950K	509K
$Average\quad Degree$	14.6	13.4	6.7
$Maximal\quad Degree$	1167	3079	3079
$Diameter$	7	11	14

(b) After Pre-processing

WikiVote	SlashDot	Epinion
2k	72K	20K
38K	840K	3700
31.1	10.5	28.9
714	5059	256
7	13	2

In all the experiments, we applied a pre-processing procedure to the networks to extract a connected network. As a result, all the isolated nodes and all boundary nodes (nodes with the degree of one) have been removed from the network. Tables 5.1(a) and 5.1(b) summarize the statistics of these real world networks before and after the pre-processing stages, respectively.

5.3 Solving the Optimization Problem

In solving our optimization problem presented in equation 4.22, we experimented with different toolboxes and approaches. All the experiments so far, presented in the previous sections and on the synthetic dataset, have used the CVX toolbox for solving the optimization problem in the OIM algorithm. CVX is a Matlab-based modeling system for convex optimization freely available for download (http://cvxr.com/cvx/).

To deal with large datasets, we adopted a new software package GLPK, to solve our optimization problem. The GLPK (GNU Linear Programming Kit) package is intended for solving large-scale linear programming (LP), mixed integer programming (MIP), which is exactly what is required for this problem. GLPK is a set of routines written in ANSI C and organized in the form of a callable library which is also free to download on web (http://www.gnu.org/software/glpk/) .

The main advantages of using GLPK can be summarized as:

- It runs faster and can handle large matrices allowing us to increase the size of local neighborhood and consider larger thresholds for the degree of nodes.

- Instead of solving the problem as convex optimization and converting the continuous output produced by the slow CVX toolbox to binary, the problem is solved as integer linear programming with simplex method. This eliminates the post-processing requirement.

5.4 Experiments

This section presents results from running our algorithms plus the benchmarks mentioned in Section 5.1 on the real-world datasets described in Section 5.2. It was only possible to run the OIM algorithm on the smaller WikiVote dataset with 2K nodes due to the large run time requirements on the other datasets. Also recall that in previous sections we were not able to run OIM on the synthetic networks with more than 200 nodes but here, due to our usage of the GLPK package for optimization, it was possible to run OIM on a 2K node network.

The parameters used in this section, especially the HIM parameters, are the same as the parameters presented in section 4.5.6. The only difference is the number of products and the advertising budget which are equal to 10 and 50, respectively. Also, running the algorithms on 10 different synthetic networks generated with the same parameters was superfluous as we worked with a deterministic real-world data.

Although using a hierarchical approach in this work reduces the problem of dealing with huge interaction matrices, as we cut the network locally and our calculation is performed on a small section of the network, but still in some cases with high degree nodes, HIM is unable to process the inverse matrix in the optimization module. Especially, in real world datasets this issue can be problematic since real social networks often possess a couple of high degree hub nodes and even a local cut of these nodes and its neighbors is almost equivalent to the whole network. In addition to creating huge interaction matrices, these nodes will create star-shape subgraphs which results in an infeasible answer for the optimization part.

There are a couple of solutions for dealing with these very high degree nodes: 1) ignore high degree nodes when we scan through the network and make the assumption that the high connectivity of this node guarantees the future processing of this node while we are looking at the neighbors of other nodes; or 2) ignore some neighbors of this node and reduce the number of nodes in the local network to a reasonable number. This selection of neighbors can be based on different strategies. Here, we chose the first approach in dealing with these nodes. Therefore, in all networks we ignored the nodes with degrees higher than 100. Examining the average degree of all datasets presented in Table 5.1(b) shows that this choice prevents huge matrices and star-shaped subgraphs while yielding a high percentage of nodes to process. By using this heuristic, the following results have been generated for WikiVote and Epinion datasets.

Figure 5.1 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 trend of the HIM algorithm is that it approaches to the global optimization method. The DegreeDiscount heuristic, PMIA, and PageRank algorithms are very close to each other with no significant difference.

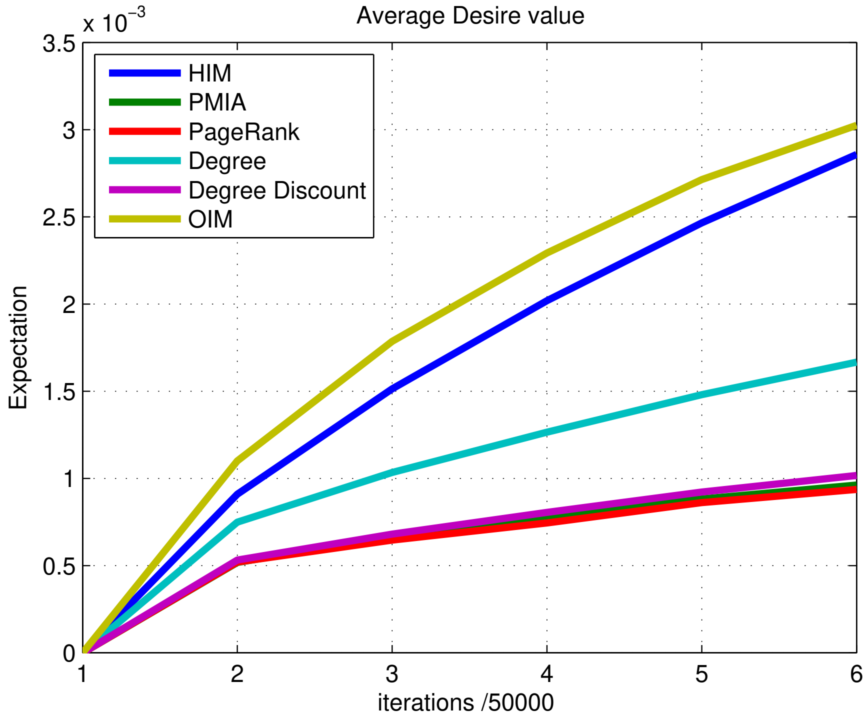

Figure 5.1: The average of agents' expected desire vs. number of iterations in the WikiVote dataset, calculated across all products and over 10 different runs, over 300K iterations. The pre-processed dataset consists of 2K nodes, and the simulation was run over 300K iterations. The optimization methods 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. During the pre-processing step the isolated and boundary nodes have been removed.

While our algorithms outperform the other benchmarks on the WikiVote dataset, on the Epinion dataset the Degree based algorithms perform better. Figure 5.2 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.

Also Figure 5.3 summarizes the final expected desire value of agents for different algorithms and for different datasets. It should be noted that the low value of desire vector is a consequence of having huge networks in which the decision of agents is multiplied by ϵ and α, the parameters that are extracted from the network and are related to the degree of nodes.

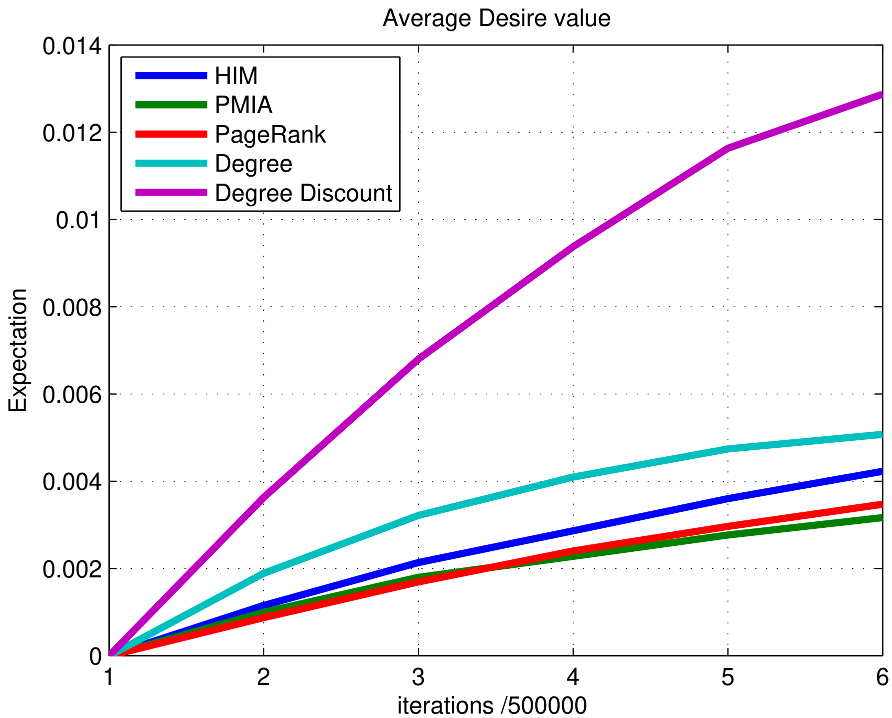

Figure 5.2: The average desire value of the agents in the Epinion dataset over 300K iterations. The pre-processed dataset consists of 20K nodes. During pre-processing the isolated and boundary nodes have been removed.

Based on our results on the Epininon dataset (and after observing the same trend for the SlashDot network) we performed further analysis to identify the characteristics of Epinion dataset that make its results different from the WikiVote and synthetic datasets in order to explain the high performance of the degree based algorithms. Table 5.2 shows the quantile analysis of the pre-processed datasets reporting the maximum degree in the 25% $(50\%, ...)$ lowest degree nodes of the network. Based on this analysis we will see that while the WikiVote network is a very small network compared to other two datasets, the max degree of its bins are higher than the others. Also the maximum degree of the whole network, compared to the number of nodes is much higher than the Epinion and SlashDot networks. Hence we conclude that this network is a more connected network with a more uniform degree distribution.

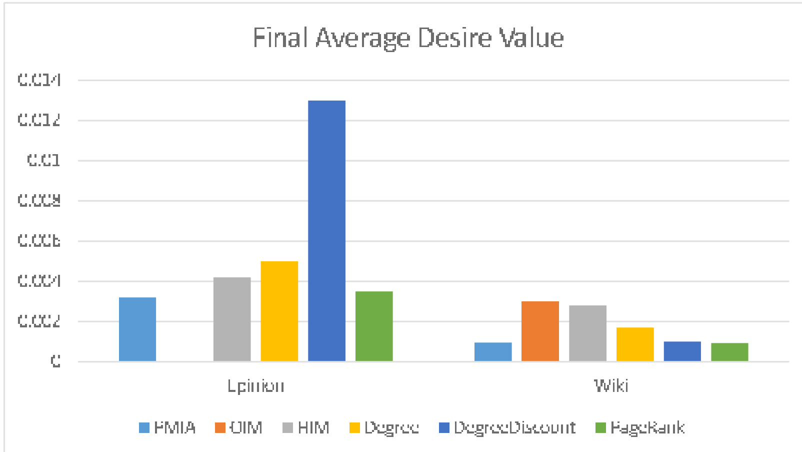

Figure 5.3: The final expected desire value of the agents at the end of the simulation for the different methods and datasets. The OIM algorithm could not be run on the Epinion dataset as a results of its huge network.

Table 5.2: Quantile Analysis on Pre-processed 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

Figures 5.4, 5.5, and 5.6 show the degree histogram of our datasets. In the Epinion and SlashDot datasets we have a small number of nodes with very high degrees while most of the nodes have a degree below 10 in the network. Therefore in these cases we have a sparse network in which few nodes serve as hubs and the rest of the nodes have few connections that aren't necessarily even connected to the high degree nodes. By applying the heuristics of ignoring high degree nodes, we not only missed counting these important nodes in the network but also have no other way to consider them and the ultimately what is selected in the HIM algorithm is the list of unimportant connections with low degrees and no potential to propagate the influence in the network. On the

other hand the degree-based algorithms target these high degree nodes and the algorithms work the best as there are no other important nodes in the network that have the potential of distributing the advertisements. In contrast, in the networks such as WikiVote or the synthetic networks where the degree of nodes is more uniform HIM works well as the nodes in the middle bins are more numerous and better connected to the entire network. Also this increases the chance of not having star shaped subgraphs which jeopardize the optimization process.

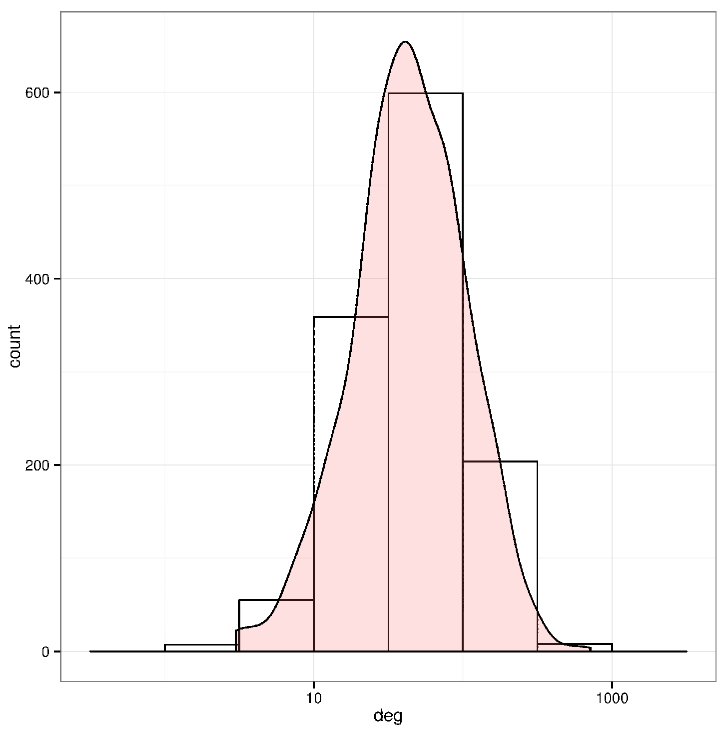

Figure 5.4: The degree histogram of the WikiVote dataset. The x-axis shows the logarithmic scale of degree and the curve shows the kernel density estimation. In this dataset the majority of nodes lie in the middle range and have a degree between 50 to 100.

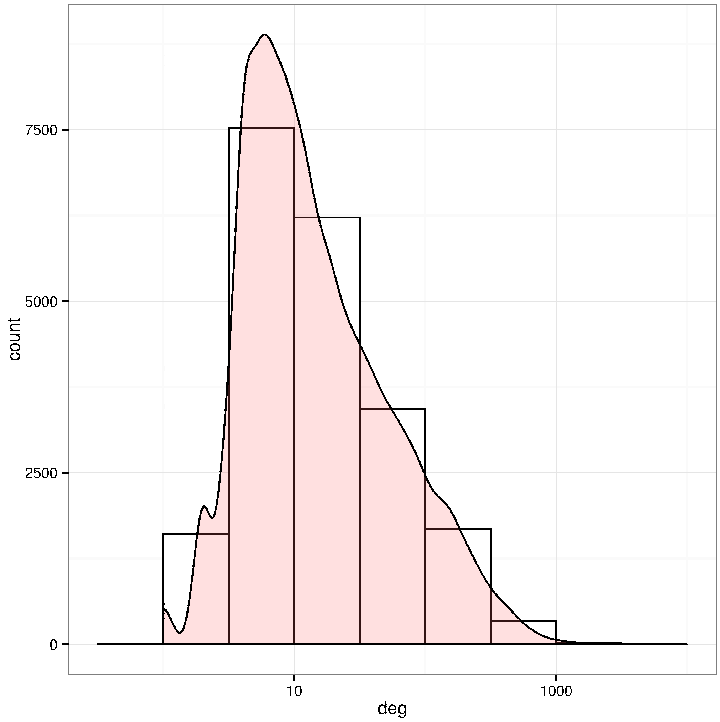

Figure 5.5: The degree histogram of the Epinion dataset. The x-axis shows the logarithmic scale of degree and the curve shows the kernel density estimation. In this dataset the network is so sparse with the majority of nodes possessing a degree less than 10.

Based on the results we have found, we used a degree-based heuristic to select the nodes considered by our optimization approach. Here, we selected the top 1% of high degree nodes in the Epinion dataset and created a subgraph based on the shortest path among these nodes, the same as the procedure we perform in the upper hierarchies in HIM and then we ran the OIM algorithm over the whole processed network. Figure 5.7 shows the result of OIM and other benchmarks on this preprocessed network. The result shows that in this case the OIM outperforms the rest of the benchmarks as it has the best selection among those filtered nodes.

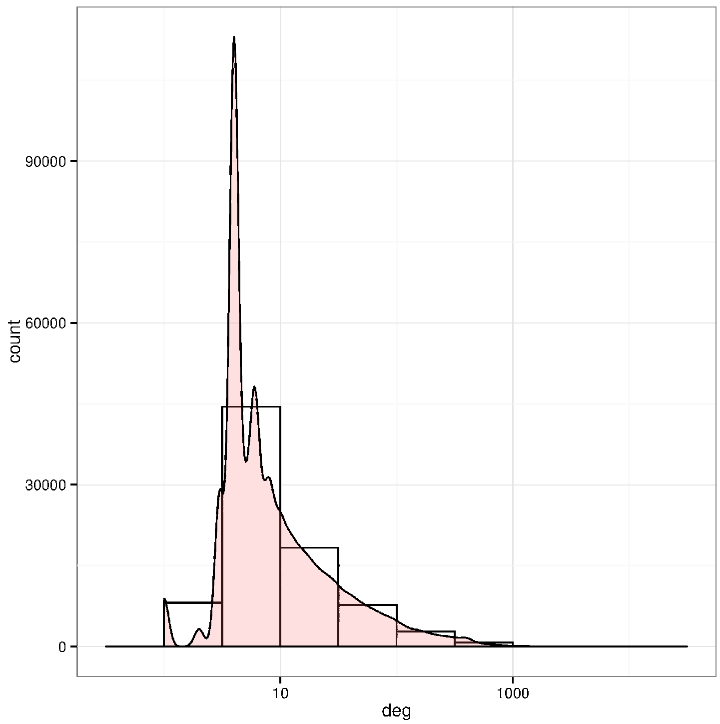

Figure 5.6: The degree histogram of the SlashDot dataset. The x-axis shows the logarithmic scale of degree and the curve shows the kernel density estimation. In this dataset, the same as Epinion dataset, the network is so sparse with the majority of nodes possessing a degree less than 10.

The conclusion is that HIM algorithm can be used to improve scalability factor on the networks with semi-uniform degree distribution. In cases with sparse networks our suggestion is to filter the nodes first and then based on the size of the processed network, apply OIM or HIM to select the influential nodes based on the advertising budget.

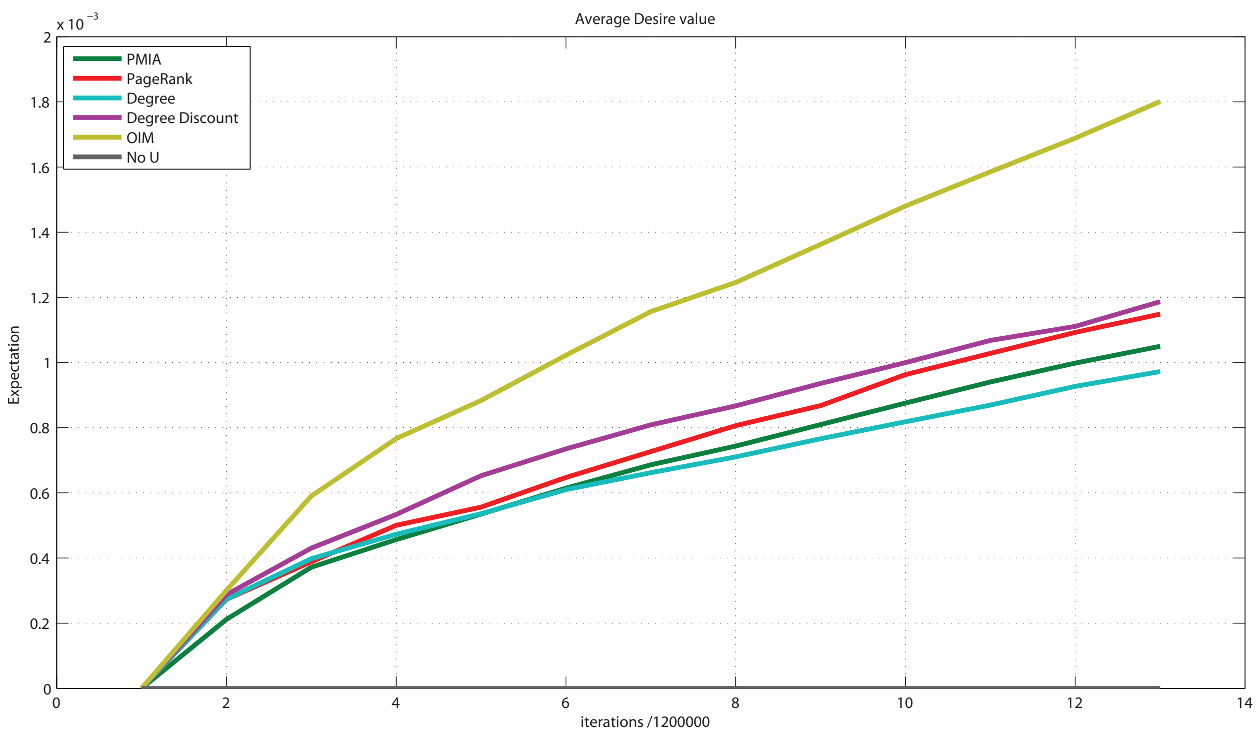

Figure 5.7: The average of agents' expected desire vs. number of iterations in the Epinion dataset, calculated across all products and over 10 different runs, over 300K iterations. The pre-processing consists of selecting the 1% top degree nodes and forming a subgraph based on the shortest path between these nodes. The optimization methods have the best performance in comparison to the other benchmarks.

CHAPTER 6: CONCLUSION

In this dissertation, we address the problem of influence maximization in social networks for the purpose of advertising. In an advertising domain, our goal is to find the influential nodes in a social network as targets of advertisement 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 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. We summarize the major contributions in the following section.

6.1 Summary of Contributions

- Generalized Interaction Model:
  - We presented an interaction model which is the generalized version of the Independent Cascade Model (ICM). This generalized version gives more flexibility in incorporating more complex interaction scenarios. The advantages of our generalized ICM can be listed as:
    1. Once the agent gets activated, it is capable of activating or influencing all other neighbors at any time afterwards. This is not the case in ICM where agents can influence their neighbors only one time step after their own activation.
    2. Influencing the neighbors is not a binary situation as in ICM in which the neighbors completely agree or completely disagree with the influencing agent. In this model agents can have a partial influence on their friends' opinion.
    3. The influence propagation is not assumed to be a progressive activation. Agents

can change their mind at any time based on their interaction with different neighbors and hence with different opinions.

- Simulated Market Model:
  - Here we proposed a dynamic market model where agents could interact with each other and affect the decision of their network neighbors. Buyers and the available products in the market are represented as agents with an assigned desire vector. The elements of the desire vector are random variables showing the desire of the agents toward purchasing each available product and can be changed whenever agents interact with each other. Our market model has the following advantages:
    1. Provides the capability of having multiple products in the market.
    2. Represents budget limitations for advertising available products in the market.
    3. Includes the purchasing history and the correlation prior product purchases into the advertising decision. Our model also considers the effect of social factors, such as group membership, on the buyer's purchase decision.
- Optimized Selection of Influential Nodes:
  - In this thesis we have presented an optimization technique to select the influential nodes in a social network based on the stricture of the network, the dynamic of the interactions, and the restriction of advertising budget. We solve the problem at steady-state assuming that the assignment of advertising would be optimal if all the interactions and decision makings converge.
- Hierarchical Selection of Influential Nodes:
  - We presented a hierarchical approach for solving the influence maximization problem and finding the influential nodes in a social network. This approach examines the net-

work locally and finds the optimized selection of nodes in each neighborhood; in some types of networks it outperforms other benchmarks. The advantages of this approach can be listed as follows:

1. The hierarchical approach gives the flexibility to use any optimization method in finding the influential node and any selection strategy in moving the influential nodes from one hierarchy to another.
2. Since this algorithm looks at the network locally, it gives us the scalability to deal with huge networks.
3. It can easily be configured for different advertisement budgets by adjusting the number of selected nodes propagated between hierarchies.

6.2 Future Work

The approaches proposed in this work have certain limitations and can be improved in many ways. We describe some attempts in the following subsections.

6.2.1 Limitation: Dynamic Networks

In this thesis all the processing and experiments were on the static networks where we had all the nodes and connections fixed. Since our optimization technique is based on the steady-state of the network, using the static network is fair. But one possible solution is to solve the optimization problem in real-time when nodes can enter and leave the network. It would be interesting to find a way to solve the problem of finding influential nodes in complex systems in real time.

6.2.2 Improvement: Adding Learning Model

Having a learning model which is able to learn the features of influential nodes would be another interesting topic which could add value to this work. In this work we don't use learning techniques to generalize the common features of influential nodes in the network. Having learning ability can potentially boost the performance and reduce the run time of the node selection process. Possible challenges of learning methods include sampling the training set and performing feature extraction based on the local network neighborhood.

6.2.3 Improvement: Adversarial Market Model

In the simulated market presented in this work, we did not account for the adversarial marketing situation. Although adopting one product can decrease the interest of the user toward all other available products, there is no accommodation for scenarios where the products are competing with each other or scenarios in which the sequence of advertisement is also important. One possible extension of this work is to design those markets like a Stackelberg competition and add proper constraints into the optimization problem as well.

6.2.4 Improvement: Add Memory for the Agents

In this work, 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 and decision making. Applying a parameter to the model which forces the agents to retain their decisions for a minimum period of simulation time, regardless of external interactions, would ameliorate this issue and make the simulation more realistic. Adding that parameter will change the interaction model and all optimization calculations but would add more value to the current simulation.

LIST OF REFERENCES

[1] A. Agogino and K. Tumer. Team formation in partially observable multi-agent systems. *NASA Ames Research Center, NTIS*, 2004.

[2] R. Albert and A. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, 2002.

[3] A. Anagnostopoulos, R. Kumar, and M. Mahdian. Influence and correlation in social networks. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 7–15. ACM, 2008.

[4] A. Apolloni, K. Channakeshava, L. Durbeck, M. Khan, C. Kuhlman, B. Lewis, and S. Swarup. A study of information diffusion over a realistic social network model. In *2009 International Conference on Computational Science and Engineering*, pages 675–682. IEEE, 2009.

[5] S. Aral and D. Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[6] L. Ardissono, L. Console, and I. Torre. On the application of personalization techniques to news servers on the www. *AI* IA 99: Advances in Artificial Intelligence*, pages 261–272, 2000.

[7] R. Axtell and J. Epstein. Growing artificial societies: Social science from the bottom up. *Washington, DC: Brookings Institution*, 1996.

[8] A. Bagherjeiran and R. Parekh. Combining behavioral and social network data for online advertising. In *IEEE International Conference on Data Mining Workshops (ICDMW'08)*, pages 837–846. IEEE, 2008.

[9] A. Barabasi and E. Bonabeau. Scale-free networks. *Scientific American*, pages 60–69, May 2003.

[10] M. Berryman and S. Angus. Tutorials on agent-based modelling with NetLogo and network analysis with Pajek. *Complex Physical, Biophysical and Econophysical Systems: Proceedings of the 22nd Canberra International Physics Summer School*, pages 351–377, 2009.

[11] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. *Internet and Network Economics*, pages 306–311, 2007.

[12] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. *Internet and Network Economics*, pages 539–550, 2010.

[13] C. Burnett, T. Norman, and K. Sycara. Bootstrapping trust evaluations through stereotypes. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pages 241–248, 2010.

[14] S. Casare and J. Sichman. Towards a functional ontology of reputation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 505–511. ACM, 2005.

[15] S. Casare and J. Sichman. Using a functional ontology of reputation to interoperate different agent reputation models. *Journal of the Brazilian Computer Society*, 11:79–94, 2005.

[16] M. Cha, A. Mislove, and K. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th International Conference on World Wide Web*, pages 721–730. ACM, 2009.

[17] W. Chen, A. Collins, R. Cummings, T. Ke, and et. al. Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 11th SIAM International Conference on Data Mining (SDM'2011)*, 2011.

[18] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1029–1038. ACM, 2010.

[19] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1029–1038. ACM, 2010.

[20] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 199–208. ACM, 2009.

[21] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the IEEE 10th International Conference on Data Mining (ICDM'10)*, pages 88–97. IEEE, 2010.

[22] A. Cuddy, S. Fiske, V. Kwan, P. Glick, S. Demoulin, J. Leyens, M. Bond, J. Croizet, N. Ellemers, E. Sleebos, et al. Stereotype content model across cultures: Towards universal similarities and some differences. *British Journal of Social Psychology*, 48(1):1–33, 2009.

[23] J. Dall and M. Christensen. Random geometric graphs. *Physical Review E*, 66(1):16121, 2002.

[24] P. Dayama, A. Karnik, and Y. Narahari. Optimal incentive timing strategies for product marketing on social networks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[25] P. Dayama, A. Karnik, and Y. Narahari. Optimal incentive timing strategies for product marketing on social networks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS2012)-Volume 2*, pages 703–710. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[26] J. Delgado. Emergence of social conventions in complex networks. *Artificial intelligence*, 141(1-2):171–185, 2002.

[27] J. Denzinger and J. Hamdan. Improving modeling of other agents using tentative stereotypes and compactification of observations. In *Proceedings of Intelligent Agent Technology*, pages 106–112, 2005.

[28] J. Denzinger and J. Hamdan. Improving observation-based modeling of other agents using tentative stereotyping and compactification through kd-tree structuring. *Web Intelligence and Agent Systems*, 4(3):255–270, 2006.

[29] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[30] J. Epstein. Agent-based computational models and generative social science. *Complexity*, 4(5):41–60, 1999.

[31] A. Flache and M. Macy. Social life from the bottom up: Agent modeling and the new sociology. In *Social Dynamics: Interaction, Reflexivity and Emergence. Proceedings of the Agent 2004 Conference*, 2004.

[32] R. Fryer and M. Jackson. A categorical model of cognition and biased decision-making. *The BE Journal of Theoretical Economics*, 8(1):6, 2008.

[33] M. Gaston and M. desJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 230–237, 2005.

[34] M. Gaston and M. desJardins. The effect of network structure on dynamic team formation in multi-agent systems. *Computational Intelligence*, 24(2):122–157, 2008.

[35] M. Gaston, J. Simmons, and M. desJardins. Adapting network structure for efficient team formation. In *Proceedings of the AAAI 2004 Fall Symposium on Artificial Multi-agent Learning*, 2004.

[36] A. Ghanem, S. Vedanarayanan, and A. Minai. Agents of influence in social networks. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[37] R. Glinton, P. Scerri, and K. Sycara. Exploiting scale invariant dynamics for efficient information propagation in large teams. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS): Volume 1*, pages 21–30. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[38] R. Glinton, P. Scerri, and K. Sycara. An investigation of the vulnerabilities of scale invariant dynamics in large teams. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2011.

[39] R. Glinton, K. Sycara, and P. Scerri. Agent-organized networks redux. In *Proceedings of AAAI*, pages 83–88, 2008.

[40] V. Grimm, E. Revilla, U. Berger, F. Jeltsch, W. Mooij, S. Railsback, H. Thulke, J. Weiner, T. Wiegand, and D. DeAngelis. Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science*, 310(5750):987, 2005.

[41] D. Hamilton, S. Sherman, and C. Ruvolo. Stereotype-based expectancies: Effects on information pprocessing and social behavior. *Journal of Social Issues*, 46(2):35–60, 1990.

[42] J. Hartline, V. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. In *Proceeding of the 17th International Conference on World Wide Web*, pages 189–198. ACM, 2008.

[43] R. Hastie and G. Stasser. Computer simulation methods for social psychology. *Handbook of research methods in social and personality psychology*, pages 85–114, 2000.

[44] B. Hendrichson. Latent semantic analysis and Fiedler embeddings. In *Proceedings of the 4th Workshop on Text Mining of the 6th SIAM International Conference on Data Mining*, 2006.

[45] J. Hilton and W. Von Hippel. Stereotypes. *Annual Review of Psychology*, 47, 1996.

[46] B. Horling and V. Lesser. A survey of multi-agent organizational paradigms. *The Knowledge Engineering Review*, 19(04):281–316, 2004.

[47] B. Hung. Optimization-based selection of influential agents in a rural afghan social network. *Master's Thesis, Massachusetts Institute of Technology*, 2010.

[48] B. Hung, S. Kolitz, and A. Ozdaglar. Optimization-based influencing of village social networks in a counterinsurgency. In *Proceedings of the 4th International Conference on Social Computing, Behavioral-cultural Modeling and Prediction*, pages 10–17. Springer-Verlag, 2011.

[49] L. Jussim, C. Fleming, L. Coleman, and C. Kohberger. The nature of stereotypes ii: A multiple-process model of evaluations1. *Journal of Applied Social Psychology*, 26(4):283–312, 1996.

[50] S. Kalick and T. Hamilton III. The matching hypothesis reexamined. *Journal of Personality and Social Psychology*, 51(4):673–682, 1986.

[51] J. Kay. Stereotypes, student models and scrutability. In *Intelligent Tutoring Systems*, pages 19–30. Springer, 2000.

[52] P. Kazienko and M. Adamski. Adrosa–adaptive personalization of web advertising. *Information Sciences*, 177(11):2269–2295, 2007.

[53] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146. ACM, 2003.

[54] D. Kempe, J. Kleinberg, and É. Tardos. Influential nodes in a diffusion model for social networks. *Automata, Languages and Programming*, pages 1127–1138, 2005.

[55] M. Kimura and K. S. nd R. Nakano. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the National Conference on Artificial Intelligence*, 2007.

[56] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. *Knowledge Discovery in Databases: PKDD 2006*, pages 259–271, 2006.

[57] M. Kimura, K. Saito, R. Nakano, and H. Motoda. Finding influential nodes in a social network from information diffusion data. *Social Computing and Behavioral Modeling*, pages 1–8, 2009.

[58] M. Kimura, K. Saito, R. Nakano, and H. Motoda. Extracting influential nodes on a social network for information diffusion. *Data Mining and Knowledge Discovery*, 20(1):70–97, 2010.

[59] D. Leborgne. *Calcul différentiel et géometrie*. Presses universitaires de France, 1982.

[60] J. Leskovec. Amazon product co-purchasing network. In *http://snap.stanford.edu/data/amazon0302.html.*, 2003.

[61] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[62] L. Liow, S. Cheng, and H. Lau. Niche-seeking in influence maximization with adversary. In *Proceedings of the 14th Annual International Conference on Electronic Commerce*, pages 107–112. ACM, 2012.

[63] X. Liu, A. Datta, K. Rzadca, and E. Lim. Stereotrust: A group based personalized trust model. In *Proceeding of the 18th ACM conference on Information and Knowledge Management*, pages 7–16. ACM, 2009.

[64] X. Liu, T. Kaszuba, R. Nielek, A. Datta, and A. Wierzbicki. Using stereotypes to identify risky transactions in internet auctions. In *Proceedings of the IEEE International Conference on Social Computing/ IEEE International Conference on Privacy, Security, Risk, and Trust*, pages 531–520, 2010.

[65] S. Luke, C. Cioffi-Revilla, L. Panait, and K. Sullivan. Mason: A new multi-agent simulation toolkit. In *Proceedings of the 2004 SwarmFest Workshop*, volume 8. Citeseer, 2004.

[66] A. Lyons and Y. Kashima. How are stereotypes maintained through communication?: The influence of stereotype sharedness. *Journal of Personality and Social Psychology*, 85(6):989–1005, 2003.

[67] C. Macrae and G. Bodenhausen. Social cognition: Thinking categorically about others. *Annual Review of Psychology*, 51(1):93–120, 2000.

[68] M. Macy and R. Willer. From factors to actors: Computational sociology and agent-based modeling. *Annual Review of Sociology*, pages 143–167, 2002.

[69] M. Maghami and G. Sukthankar. Identifying influential agents for advertising in multi-agent markets. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 687–694. International Foundation for Autonomous Agents and Multiagent Systems, 2012.

[70] J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics, 2nd Ed.* Wiley, 1999.

[71] D. M. Marx. On the role of group membership in stereotype-based performance effects. *Journal of Social and Personality Psychology Compass*, 3(1):77–93, 2009.

[72] J. McClelland and D. Rumelhart. A distributed model of human learning and memory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2, pages 170–215. MIT Press, 1986.

[73] J. McClelland, D. Rumelhart, et al. *Parallel distributed processing*. Citeseer, 1986.

[74] J. Miller and S. Page. *Complex adaptive systems: An introduction to computational models of social life*. Princeton Univ Pr, 2007.

[75] N. Minar and N. Santa Fe Institute (Santa Fe. The swarm simulation system: A toolkit for building multi-agent simulations. In *Working Paper 96-06-042, Santa Fe Institute, Santa Fe.* Citeseer, 1996.

[76] S. Moss, B. Edmonds, et al. Sociology and simulation: Statistical and qualitative cross-validation. *American Journal of Sociology*, 110(4):1095–1131, 2005.

[77] M. North and C. Macal. Escaping the accidents of history: An overview of artificial life modeling with repast. *Artificial Life Models in Software*, pages 115–141, 2005.

[78] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14):3200–3203, 2001.

[79] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 965–970. IEEE, 2010.

[80] S. Queller and E. Smith. Subtyping versus bookkeeping in stereotype learning and change: Connectionist simulations and empirical findings. *Journal of Personality and Social Psychology*, 82(3):300–313, 2002.

[81] M. Resnick. *Turtles, termites, and traffic jams: Explorations in massively parallel microworlds*. The MIT Press, 1997.

[82] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Behavioral analyses of information diffusion models by observed data of social network. *Advances in Social Computing*, pages 149–158, 2010.

[83] R. Sawyer. Artificial Societies. *Sociological Methods & Research*, 31(3):325, 2003.

[84] M. Schaller, L. Conway, and T. Tanchuk. Selective pressures on the once and future contents of ethnic stereotypes: Effects of the communicability of traits. *Journal of Personality and Social Psychology*, 82(6):861–877, 2002.

[85] T. Schelling. Dynamic models of segregation. *The Journal of Mathematical Sociology*, 1(2):143–186, 1971.

[86] T. Schelling. *Micromotives and macrobehavior*. New York. WW Norton, 1978.

[87] P. Sen and L. Getoor. *Link-based classification*. Technical Report, CS-TR-4858, University of Maryland, Reading, Massachusetts, 2007.

[88] P. Shakarian and D. Paulo. Large social networks can be targeted for viral marketing with small seed sets. In *Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1–8. IEEE, 2012.

[89] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.

[90] P. Singla and M. Richardson. Yes, there is a correlation:-from social networks to personal behavior on the web. In *Proceeding of the 17th International Conference on World Wide Web*, pages 655–664. ACM, 2008.

[91] E. Smith and F. Conrey. Agent-based modeling: A new approach for theory building in social psychology. *Personality and Social Psychology Review*, 11(1):87, 2007.

[92] E. Smith and J. DeCoster. Knowledge acquisition, accessibility, and use in person perception and stereotyping: Simulation with a recurrent connectionist network. *Journal of Personality and Social Psychology*, 74:21–35, 1998.

[93] E. Smith and J. DeCoster. Person perception and stereotyping: Simulation using distributed representations in a recurrent connectionist network. *Connectionist Models of Social Reasoning and Social Behavior*, pages 111–140, 1998.

[94] S. Spencer, R. Gerety, K. Pienta, and S. Forrest. Modeling somatic evolution in tumorigenesis. *PLoS Comput Biol*, 2(8):e108, 2006.

[95] T. Stewart, K. Doan, B. Gingrich, and E. Smith. The actor as context for social judgments: Effects of prior impressions and stereotypes. *Journal of Personality and Social Psychology*, 75(5):1132–1154, 1998.

[96] L. Tesfatsion. Agent-based computational economics: Growing economies from the bottom up. *Artificial Life*, 8(1):55–82, 2002.

[97] L. Tesfatsion. Agent-based computational economics: Modeling economies as complex adaptive systems. *Information Sciences*, 149(4):263–269, 2003.

[98] L. Tesfatsion. Agent-based computational economics: A constructive approach to economic theory. *Handbook of Computational Economics*, 2:831–880, 2006.

[99] D. Van Rooy. Modeling multidirectional, dynamic social influences in social networks. In *Proceedings of the 18th World IMACS/MODSIM Congress*, pages 2962–2968, 2009.

[100] C. Wang, W. Chen, and Y. Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, pages 1–32, 2012.

[101] X. Wang, M. Maghami, and G. Sukthankar. Leveraging network properties for trust evaluation in multi-agent systems. In *Proceeding of the 2011 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Lyon, France*, 2011.

[102] U. Wilensky. Netlogo. *Connected Learning and Computer-Based Modeling, Northwestern University : http://ccl.northwestern.edu/netlogo/*, 1999.

[103] U. Wilensky and M. Resnick. Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology*, 8(1):3–19, 1999.

[104] C. Wilson, B. Boe, A. Sala, K. Puttaswamy, and B. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European conference on Computer systems*, pages 205–218. Acm, 2009.

[105] F. Wu, B. Huberman, L. Adamic, and J. Tyler. Information flow in social groups. *Physica A: Statistical and Theoretical Physics*, 337(1-2):327–335, 2004.

[106] W. Yang, J. Dia, H. Cheng, and H. Lin. Mining social networks for targeted advertising. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences-Volume 06*, pages 137–1. IEEE Computer Society, 2006.