AUCKLAND UNIVERSITY OF TECHNOLOGY

MASTER THESIS (MCIS)

INTELLIGENT INFLUENCE MAXIMISATION IN ONLINE SOCIAL NETWORKS

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER AND INFORMATION SCIENCES

June 2016

By

Chang Jiang

School of Engineering, Computer and Mathematical Sciences

Declaration of Authorship

I, Chang Jiang, declare that this thesis titled, 'Intelligent Influence Maximisation in Online Social Networks' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has not previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Emily. J

Date: 16/Jun/2016

Abstract

An online social network can be defined as a set of socially relevant individuals with some patterns of interactions or contacts among them, which are connected by one or more online relations [1][2]. Online social networks provide platforms for users to share information or statuses and to communicate with their families and friends online. These complex, emergent, dynamic and heterogeneous networks have been developed on an unprecedented scale. With the prosperous development of online social networks, many marketers have exploited the opportunities and attempt to select influential users within online social media to influence other users through online 'word-of-mouth' effect or viral marketing approaches, which are now replacing traditional marketing strategies [3]. Such 'word-of-mouth' effect and viral marketing approaches can enhance brand awareness and achieve the marketing objectives of companies with limited resources. In this situation, the propagation of influence to online users with limited resources over the largest possible range, known as influence maximisation, is an important problem. The solution of influence maximisation is known to be NP-hard. Hence, approximation approaches are better replacements with guarantee [4][5][6].

This thesis explores appropriate approaches for solving the influence maximisation problem effectively and efficiently. Based on the existing problems of influence maximisation in online social networks, influence maximisation is developed through two different approaches; centralised and decentralised. In centralised approaches, all tasks are completed by a single central component. By contrast, decentralised approaches share the workload by distributing the computational tasks to individuals.

Classic influence diffusion models with static and predefined probabilities are too ideal, as they consider only the physical link connections [3][5], whereas online social networks contain additional subjective factors, such as, user preference. User preference plays an important role in influence maximisation, but is not considered in most of the existing influence maximisation models. To alleviate these problems, we proposed a Preference-based Trust Independent Cascade Model, which is founded on a classic centralised approach. This develops influence maximisation in terms of both user preference and trust connection (physical link connection). Based on these two factors, the Preference-based Trust Independent Cascade Model computes the influence propagation probabilities. In this way, hub users in an online social network, who are interested in the promoted items, can be selected as influential users. In experimental results, the

Preference-based Trust Independent Cascade Model demonstrated better performance than other existing approaches.

Furthermore, by reviewing the previous researches and implementing experiments, we discover that centralised approaches are generally inefficient because they limit the stability and scalability of large-scale, dynamic online social networks. To overcome this problem, we propose a novel decentralised approach called Stigmergy-based Influence Maximisation Model, which simulates the influence propagation process by ants crawling across the network topology. The model mimics the key behaviours of ants, i.e., path selection and pheromone allocation. The former identifies the next node to reach when an ant faces multiple options; the latter deposits pheromone on the specific nodes based on the heuristics when an ant explores a possible influence-diffusion path. The superior performance and operating time of the Stigmergy-based Influence Maximisation Model was confirmed in comparison experiments against existing approaches.

Acknowledgements

Master's study is really a tough challenge that I could not have completed without sincere help from many people.

First of all, I would like to express the sincerest appreciation to my supervisor, Dr. Quan Bai. Over the past year, Dr. Quan Bai provided me with really helpful guidance and support for my Master's study and research with great patience, enthusiasm and motivation. Through the process of my research, besides his guidance and support, Dr. Quan Bai gave me a free hand to develop my research and opportunities to express my own ideas and opinions as well. Furthermore, during the toughest period of my thesis, he provided me with much encouragement to cheer me up.

I would also like to give my great appreciation to Weihua Li (a Ph.D Candidate of Auckland University of Technology) for his continuous encouragement and support for my thesis and research over the past year, especially in the programming part. Weihua Li was very patient and passionate throughout our discussions. I also give my thanks to my parents for their unconditional encouragement and support.

I would also like to give my gratitude to all of the members of Department of Computer and Mathematical Sciences for their kind support and assistance. I am very grateful to my proof reader, Leonie Pipe, for giving me useful recommendations and comments. My thanks also go to Auckland University of Technology for all of the support and facilities provided for me, which have made this thesis feasible.

Finally, I give my appreciation to all of the people who lent me their helping hands during my research process.

Publications

Throughout the year of my Master's study, the following papers were published.

1. Chang Jiang, Weihua Li, Quan Bai and Minjie Zhang. Preference Aware Influence Maximisation. In Proceedings of *International Joint Agents Workshop and Symposium (IJAWS), 2015*. Springer. (Accepted)

2. Weihua Li, Chang Jiang, Quan Bai and Minjie Zhang. Stigmergy-based Influence Maximisation in Social Networks. In *Proceedings of Pacific Rim International Conferences on Artificial Intelligence (PRICAI), 2016*. (Accepted)

Contents

List of Figures

Chapter 1

Introduction

1.1 Influence Maximisation (IM) in Online Social Networks

1.1.1 Online Social Networks

With the development of Web 2.0 having been greatly promoted, online social media, such as, Facebook, LinkedIn, Instagram, Wechat, Tercent QQ and Sina Weibo, have gained popularity and have developed on an unprecedented scale recently. According to Statista[1], Facebook and Twitter had more than 1.59 billion and 305 million monthly active users, respectively, in the fourth quarter of 2015.

An online social network can be defined as a set of socially relevant individuals with some patterns of interactions or contacts among them, which are connected by one or more online relations [1][2]. Online social networks provide platforms for users to communicate and share information with their relatives and friends online effectively, connecting users across the world. The various online social networks are fundamentally classified by their node degree distributions [7]. The most frequently mentioned categories of online social networks are randomly distributed, scale-free, large-scale, and complex networks.

[1]http://www.statista.com

1.1.2 Characteristics of Online Social Networks

Online social networks are self-organizing, emergent, dynamic, and complex. They are most commonly characterised as scale-free and small-world effect [8][1]. The earliest example of a scale-free network was probably proposed by Price et al. [9]. They defined a scale-free network as a social network whose node degree distribution adheres to or is at least infinitely close to a power law form, indicating that the network is independent of the social network scale [1]. Price et al. derived the power-law distribution of their network along with a parameter α valued between 2.5 and 3. The main feature of a scale-free social network is inhomogeneity [10][8]. Most of the nodes in a social network have few physical link connections, while the connections of a few nodes far exceed the average number of physical link connections within the social network.

The second characteristic of online social networks is the small-world effect [8][1]. The small-world effect was experimentally confirmed by Stanley [11], who demonstrated that letters passed through a series of users in an online social network reach their pre-defined target user only within a few steps. Existing researches suggest that the number of required steps is around six. Through this experiment, Stanley discovered that almost any two nodes in the online social network even non-neighbouring nodes can be connected by a very short path. In addition, small-world online social networks are governed by two important parameters, i.e., the clustering coefficient and the diameter.

1.1.3 Emergent Complexity of Online Social Networks

Online social networks are complex, meaning that they are composed of interconnected components, consequently an online social network is inherently related to its interconnected components. In general, emergent complexity can be described as the behaviours of each component in an online social network interacting in the way that the behaviour formed of the whole via coordinating with each other is complex [12]. In other words, when describing the behaviour of the whole, the behaviours of each component in the online social network may be regarded as simple. This is because the behaviours of many of these components in an online social network are emergent, so that the behaviour of the whole can not be deduced directly from the behaviours of each component. The term ‘interconnected’ is the key to understand the complexity of online social networks. Hence, in order to know an online social network in a further step, we must understand not only the behaviours of each component but also the coordination between the components, which ultimately accomplish the behaviour of the whole. The coordination, however, is difficult to understand.

1.1.4 Influence Maximisation (IM)

Exploiting the development of online social networks, many marketers attempt to influence online social media users through the 'word-of-mouth' effect or viral marketing approaches, rather than pursue traditional marketing strategies [3]. The remarkable merit of 'word-of-mouth' effect and viral marketing approaches is that they increase the market share for companies with limited resources.

In this situation, the propagation of influence to users with limited resources over the largest possible range is an important issue, and is called influence maximisation (IM). In general, influence refers to the ability to sway or change the thoughts, beliefs or actions of an individual or a community [13]. Based on the aforementioned, IM can be formally defined as follows: Given a set of nodes S and the initial active set S_0, define the influence spread of set S as $\sigma(S)$, which is the prospective number of active nodes activated by set S. Based on the limited resources related to parameter k, find a k-top node set S that propagates influence over the largest possible range [3]. In addition, to clarify IM, consider the following motivating scenario. Suppose an organisation develops a new mobile phone and plans its online promotion. Because of limited resources, the organisation needs to select a limited number of influential users to experience this product and promote it to other users connected to the influential users through the online social network for the organisation. Eventually, the product will reach and be accepted by a large number of users. Thus, provided that the influential users are properly and completely selected, the influence spread will be effective. However, IM problems are known to be NP-hard, and some approximation approaches are considered as better replacements with guarantee. The two most widely used influence diffusion models in IM development are the Independent Cascade (IC) and Linear Threshold (LT) models [5][3].

1.2 Existing Problems in Influence Maximisation (IM)

Describing sophisticated online social media, classic IM models are insufficient for the following reasons.

First, most of the existing approaches consider only the propagation links/channels in IM (called trust connection (TC) in this thesis). Assume that two users following each other on Twitter. In this situation, both users are connected by a TC. If the TC only is considered, the IM will be treated as a simplified probabilistic problem. Furthermore, influence probabilities are predefined and static, whereas many real-world applications are dynamic. Meanwhile, in a social network, the IM is affected by user preferences for particular items [6]. Unfortunately, most of the existing IM approaches ignore this factor. In addition, the existing models have low effectiveness and efficiency, and most of them are time-consuming.

Second, most of the existing IM models are based on centralised approaches, which are usually inefficient in large-scale, dynamic networks. In particular, they limit the stability and scalability of the social networks, because all tasks in a centralised approach are performed by a central component. Furthermore, the seed selection algorithms that run the classic influence diffusion models are time-consuming. Finally, centralised approaches require complicated computation.

1.3 Research Motivations and Objectives

In this thesis, we explore approaches for achieving an effective and efficient IM. Since the IM problem is NP-hard, provided that the influential users can be selected properly and completely, the influence spread can be effectively achieved. To resolve the existing problems of IM field, two IM approaches have been proposed; centralised and decentralised. As a centralised approach, we propose the Preference-based Trust Independent Cascade (PTIC) Model, which accounts for both user preference (UP) and trust connection(TC). As a decentralised approach, we propose the Stigmergy-based Influence Maximisation (SIM) Model, which is based on ant algorithms and stigmergy.

1.3.1 The Preference-based Trust Independent Cascade (PTIC) Model

To resolve the existing problems of influence diffusion models and motivated by the prosperous development of viral marketing as mentioned above, we attempt to achieve research motivations as follows:

- To improve the effectiveness and efficiency of seed selection for IM, take both UP and TC into consideration;
- For better compatibility with dynamic environments, compute the influence propagation probability based on UP and TC.

Based on the motivations above, the Preference-based Trust Independent Cascade (PTIC) Model is proposed in this thesis. The PTIC model is based on a classic influence diffusion model, namely, the Independent Cascade (IC) Model. In the PTIC model, it takes into account not only UP but also TC. Furthermore, the PTIC model computes the influence propagation probability based on UP and TC.

1.3.2 The Stigmergy-based Influence Maximisation (SIM) Model

To alleviate the existing problems of centralised approaches, we develops IM using a novel decentralised approach. Different from centralised approaches, decentralised approaches share the workload by distributing the computational tasks to individual in the network. The research motivations of using the decentralised approach are as below:

- To improve the effectiveness and efficiency of IM seed selection in future steps, simulate influence propagation as an crawling across the network topology.
- To reduce the computational complexity of seed selection, distribute the task of seed selection among multiple ants.

To achieve the motivations mentioned above, we exploits a novel decentralised approach, i.e., stigmergy-based algorithm, to tackle the IM problem. In the meanwhile, the Stigmergy-based Influence Maximisation (SIM) Model is proposed. We simulate the influence propagation process as ants' crawling across the network topology. Furthermore, the SIM model mimics the key behaviours of ants; namely, path selection and pheromone allocation. The former aims to identify the next node to reach when an ant faces multiple options. Based on the heuristics, the latter deposits pheromone on specific nodes when an ant explores a possible influence-diffusion path.

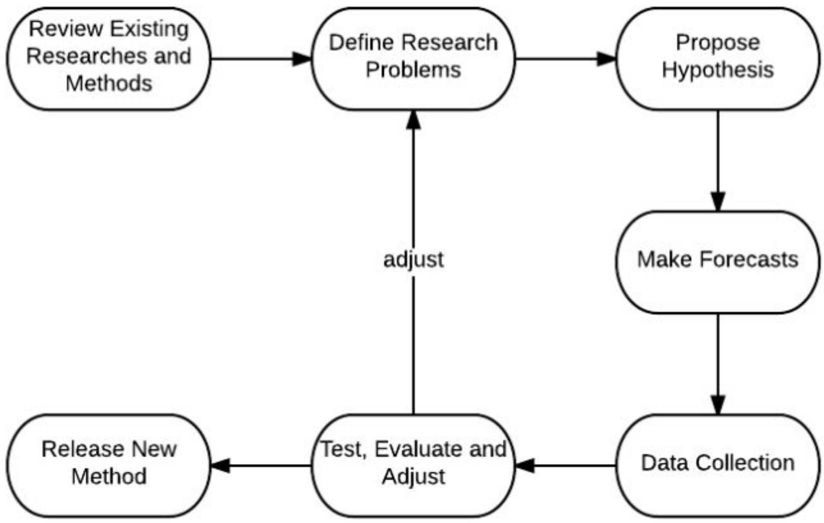

FIGURE 1.1: Research Methodology

1.4 Research Methodology

This section presents the research methodology. In general, research can be defined as innovative activities that advance existing knowledge by systematic and theoretical methods [14]. A research methodology provides the scientific criteria for executing of the research in a given field of study. Figure 1.1 is a flowchart of the methodology proposed for the current research. The first of the seven procedures reviews the existing researches and methods related to online social networks and IM. Guided by the literature review, the second procedure defines the research problems. The third procedure proposes hypotheses based on the defined research problems. At this stage, the results of the research are uncertain. The fourth procedure forecasts the expected output of the experiments. The fifth procedure acquires the preliminary data in preparatory experiments. The research problems, hypotheses, forecasts, and data are tested, evaluated, and adjusted in the sixth procedure. Procedures 2-6 are iterated until no further adjustment is required. Finally, the new method is released.

1.5 Major Contributions of Thesis

In the contemporary IM research field, most researchers investigate IM problems by using centralised approaches, where the network topology is available. Whereas, in some situations, the seed set is supposed to be selected without a clear network topological structure. Motivated by this background, this thesis presents two different approaches, one centralised, the other decentralised, that alleviate the disadvantages of existing approaches to IM problems. First, it proposes the PTIC model as an optimised centralised approach. Unlike many existing models, the PTIC model considers both UP and TC. In this way, the PTIC model not only selects hub users who are genuinely interested in the promoted items, but also connects them to users with similar preferences, thereby propagates the influence of the item. This approach significantly improves the effectiveness and efficiency of IM. Furthermore, rather than predefining the influence propagation probability, the PTIC model bases the probability on the UP and TC, which automatically adapts the PTIC model to dynamic environments. Experiments also confirm that by considering both UP and TC, the PTIC model outperforms trust-only and random methods. Second, we develops a novel decentralised approach based on stigmergy (the SIM model), which utilizes the ants' crawling across the network topology to simulate the influence propagation process. The whole process of the ants' crawling is modelled on the key behaviours of ants (path selection, pheromone allocation, and seed selection). The performance of the SIM model is experimentally compared against those of traditional seed selection algorithms, namely, greedy selection, degree-based selection and random selection. In these experiments, the SIM model proves its superior effectiveness and efficiency.

1.6 Thesis Organisation

The remainder of this thesis is organised as follows.

Chapter 2 reviews the previous researches related to IM in online social networks. The review covers existing IM methods, classic influence diffusion models with examples, popular IM algorithms (greedy and community partition algorithms), methods used for discovering user preferences, the existing researches related to stigmergy-based approaches and ant algorithms.

Chapter 3 presents the PTIC model which accounts for both UP and TC. The problem of applying the PTIC model to IM is described and the PTIC model is formally defined. The framework of the PTIC model is interpreted in detail. In the framework, there are totally four modules. Each of the four modules is accompanied by a corresponding algorithm presented in pseudo code to clarify its operation. The PTIC model is evaluated in two experiments, one employing the TC parameter only while the other employing both UP and TC parameters. In both experiments, the PTIC model is competed against trust-only and random methods.

Chapter 4 presents the proposed SIM model. First, the problem is defined and formal definitions are proposed accordingly. The SIM model simulates the influence propagation process as ants' crawling through the network topology. Hence, this chapter models the whole process of the ants' crawling within the network and the whole process includes ant behaviours as follows: start a tour, select paths, complete a tour, allocate pheromone, and select seeds. Especially, for each key behaviour of ants, i.e., path selection, pheromone allocation, and seed selection, this chapter presents a corresponding algorithm through using pseudo code describing the operation of the behaviour. In addition, The SIM model is evaluated in two experiments on three different sizes (500, 750, and 1000) of the same social network. The first experiment aims to evaluate the effectiveness of the algorithms, i.e., the total number of users activated by the seed set, whereas the second algorithm compares their efficiencies, i.e., the runtime of seed selection. The stigmergy-based algorithm is competed against greedy selection, degree-based selection and random selection algorithms.

The concluding chapter, Chapter 5, highlights the contribution of this thesis and presents suggestions for future work related to this research.

Chapter 2

Literature Review

2.1 Existing Influence Maximisation (IM) Methods

IM is first proposed by Domingos and Richardson as a probabilistic problem [15][16], and is first researched as a discrete optimisation problem by Kempe et al. [3]. The latter authors demonstrate the NP-hardness of solution to the IM problem. NP-hard problems are notoriously difficult to solve. Thus, instead of finding the solutions for NP-hard problems, approximation approaches are better replacements that offer guaranteed results [4][5][6].

Seed selection in IM is classically performed by degree ranking, greedy algorithm, and heuristic methods [5][6]. Degree ranking method selects seeds based on their degree relationship ranking. In general, all nodes in the network are ranked in descending order from the highest-degree node to the lowest-degree node, and seeds are selected from the highest-degree node. The greedy algorithm selects seeds by making optimal local choices at each step. The greedy algorithm is derived from the principles of problem solving heuristics. Heuristic techniques are any methods used to learn, discover or solve problems by taking advantage of an appropriate method to achieve the immediate objectives, which accelerates the progress towards an optimal solution. However, the quality of the solutions is not guaranteed.

2.2 Classic Influence Diffusion Models

Influence is propagated on the foundation of influence diffusion models through the network. This section presents and interprets several fundamental and improved diffusion models; the Linear Threshold (LT) model, the Independent Cascade (IC) model, the General Threshold (GT) model and the Heat Diffusion (HD) model. Rogers [17] defines diffusion as a procedure triggered by an innovation, such as a new product or a new technique, which is communicated among the individuals in a social network via channels. These fundamental models, especially the IC model, have been extensively studied. For example, Chen et al. propose the first Scalable Heuristic Algorithm based on the LT model and a new Degree Discount Heuristics based on the IC Model [5][18].

In general, a social network can be modelled as a directed or an undirected graph $G = (V, E)$, where the set of vertices V represents the individuals in the network, and the set of edges E represents the relationships among the individuals in the network [19][5]. In a graphical context, the IM problem can be defined as follows: Given a social network graph G, a selected influence diffusion model M, and a predefined number k, the purpose of IM is to select k influential vertices (seeds) in the known network by implementing the influence diffusion model M, and computes the prospective number of vertices influenced by these k influential vertices as the influence spread.

2.2.1 Independent Cascade (IC) Model

The Independent Cascade (IC) model is a fundamental diffusion model. In the IC model, V_t is the set of vertices activated at round t, where vertex $v \in V_t$ and $\vec{vu} \in E$. Assume that vertex u is not activated but will be activated by vertex v at round $t+1$ with probability p. If vertex v has only one chance to activate its neighbour u, vertex u is either activated at round $t+1$ or otherwise, is never activated by vertex v. In addition, if l number of neighbours attempt to activate vertex u at round t, the activation proceeds in a stochastic sequence, and the probability of $u \in V_{t+1}$ is $1-(1-p)^l$. This process which basically defines the principles of the IC model continues until there is no vertex can be activated. [5]

The IC model has been improved in subsequent studies. Chen et al. [19][5] propose a new efficient model named 'Degree Discount Heuristics'. In each round, whenever a neighbour of an inactive node becomes active, the degree of the inactive node decreases by one. After that, find out the node with the highest node degree in the network and select it into seed set. Degree discount heuristics is an iterative procedure that markedly reduces the operating time relative to other existing models. In fact, the performance of degree discount heuristics model nearly matches that of the greedy algorithm based on the IC Model. However, this model only can be achieved by small probabilities since this model is not sensitive when the probabilities are large. This model is thus inapplicable to real-world situations, in which the probabilities of influencing individuals should be large.

Except degree discount heuristics method, Wang et al. [20] propose an alternative heuristic algorithm based on the IC model, which easily handles influence propagation within large-scale networks. Wang et al. effectively reduce the operating time by limiting the computation related to local influence regions of nodes. Furthermore, they define a tunable parameter that controls the balance between performance and operating time. Liu et al. [21] propose a time-constrained influence maximisation model, called the Latency Aware Independent Cascade Model, which is developed on the foundation of the IC model and greedy algorithm, and guarantees strong performance. However, the greedy algorithm simulation is computationally expensive. In order to ensure the scalability of their approach, they propose influence-spreading path-based methods to estimate the influence spread of a given seed set. Saito et al. propose an approach that predicts influence diffusion probabilities by using the expectation-maximisation algorithm, which is also founded on the IC model [22]. Gomez-Rodriguez et al. propose a scalable algorithm, called NETINF, which tracks diffusion and influence paths through networks by exploring the sub-modularity of the objective function based on the IC modelling [23]. Wang et al. [24] propose a novel algorithm, i.e., Community-based Greedy Algorithm, for exploring top-k influential nodes. This algorithm is again built on the IC model, and detects communities as well as selecting influential nodes.

2.2.2 Linear Threshold (LT) Model

Another fundamental model is the Linear Threshold (LT) model. Given vertex v is activated, the neighbourhood of vertex v is defined as $N(u) = v|(v,u) \in E$, and R_{vu} denotes the propagation influence between the active vertex v and an inactive neighbour u. In addition, let $V(u)$ denote the set of active vertices in $N(u)$, where $V(u)$ is a subset of $N(u)$. Most important is the critical threshold $theta$, which determines whether u is activated. Specifically, if $\Sigma_{v\in V(u)} R_{vu} \geq \theta$, then vertex u becomes active. In other words, vertex u will become active when the value of its active neighbours' entire influence on vertex u exceeds the defined threshold $theta$. Similarly to vertex v, vertex u will then propagate influence to other inactive vertices, and some of which will activate. This process iterates until no vertex can be activated. [19]

Goyal et al. [25] propose an efficient and effective algorithm based on the LT model called 'SIMPATH', which optimises the computation of influence spread by searching the simple paths in the neighbourhood. According to specific situations, this algorithm divides the given graph into proper induced sub-graphs, and calculates the influence spread of each vertex in each sub-graph. Finally, it computes the sum of influence spread of each sub-graph. They define a unique parameter η, which is used for balancing the quality of the influential vertices against the runtime. There are three main novel methods in this algorithm for improving the quality of seed selection and optimizing the computation. In addition, Goyal et al. propose Vertex Cover Optimisation and Look Ahead Optimisation, which deduct estimation calls in the first iteration and improve the efficiency in the remaining iterations, respectively. However, at present, this model can only be operated under the LT model. Furthermore, Chen et al. propose the first Scalable Heuristic Algorithm based on the LT model. Nevertheless, network analysis approaches, such as, community partition, are not involved in the Scalable Heuristic Algorithm. Community partition could improve the efficiency and effectiveness of IM in further study.

2.2.3 General Threshold (GT) Model

The General Threshold (GT) model is an extension of the IC and LT models. Assume that there are an inactive user v_i and its active neighbour set N. In order to measure whether user v_i will become active, joint influence probability of set N should be calculated. Joint influence probability of set N is expressed as $p_{v_i}(N)$. If $p_{v_i}(N) \geq \theta_{v_i}$, where θ_{v_i} is the activation threshold of user v_i, then user v_i becomes activated. The joint influence probability $p_{v_i}(N)$ is computed by Equation 2.1. [4]

$$p_{v_i}(N) = 1 - \Pi_{v_m \in S}(1 - p_{v_m,v_i}) \tag{2.1}$$

2.2.4 Heat Diffusion (HD) Model

Heat diffusion is a common and well known physical phenomenon. Heat transfers from an initial point with high temperature to a point with low temperature. This is similar with the process of influence propagation. Influence propagates from influential users (analogous to initial hotspots in heat diffusion) to "cooler" users. In a heat diffusion context, the procedures of influence propagation can be interpreted as follows: At the initial time t_0, initialize all of the vertices in the network have no heat. One vertex v_j is then selected as an initial point and be given some heat. At some time t_1, heat from v_j simultaneously reaches all of its neighbours. At some later time t_2, these heated neighbour vertices simultaneously transfer heat to their neighbours. Repeating this process gives the number of vertices influenced by v_j in the network during a specified time interval t. The amount of influence spread of each vertex in the network can be obtained by repeating this process over all of the vertices. [26]

Chang [19] propose a novel community and degree heuristics based on the HD model. This method is developed with CDH-Kcut and CDH-Shrink algorithms, which are improved on the fundamental of original Shrink and Kcut algorithms (These two algorithms are interpreted in detail in Subsections 2.2.2 and 2.2.3, respectively). Chang's model proceeds through two primary phases; partition and selection phases. The partition phase is used to detect communities and the selection phase is conducted to select the influential vertices for the seed set. Both phases implement the CDH-Kcut and CDH-Shrink algorithms. The CDH-Kcut algorithm is utilized to partition the network into communities and select the influential vertices; the CDH-Shrink algorithm is used to identify the structure of each community by detecting the location of community hub in the partition phase, and it is able to adjust the fundamental nodes in the selection phase. Even though this method is implemented based on the HD model, Chang's model does not take weighted graphs into consideration for developing this method, rendering the method too idealistic for real world issues. In addition, this method can only detect static communities, whereas dynamic community detection is required for current development of social network.

2.3 Algorithms Applied for Influence Maximisation (IM)

Many algorithms have been developed by various researchers in IM field. This section introduces some of the more frequently used algorithms related to IM problems.

2.3.1 Greedy Algorithm

Kempe et al. [3] demonstrate that IM is NP-hard. As an approximation, they propose a simple greedy algorithm within the factor $(1 - 1/e)$. Instead of running a specific algorithm, Kempe et al. conduct Monte-Carlo simulations of diffusion models to get an accurate estimation for the prospective number of influence spread. In addition, a large number of previous research papers employ Monte-Carlo simulations for selecting influential vertices [27][5][28]. However, this simple greedy algorithm with Monte-Carlo simulations is time-consuming, as it has to compute the influence spread of each vertex in each round before selecting the influential vertices for the seed set.

Leskovec et al. [29] propose an optimisation for finding the influential vertices based on the simple greedy algorithm, which is called the Cost Effective Lazy Forward (CELF) scheme. This optimisation takes use of the sub-modularity property of spread functions. The fundamental concept is that the marginal gain of a vertex in the current round is always less than that in previous rounds. Consequently, the number of influence propagation estimation calls gradually reduces as the iterations proceed. In addition, the sub-modularity property can ensure the amount of influence spread increasing when adding a new vertex into the seed set. Furthermore, Goyal et al. [30] propose an optimised CELF algorithm named the CELF++ algorithm, which has been reported by Goyal et al. that the CELF++ algorithm operates 35 to 55 percent faster than the original CELF algorithm.

Chen et al. [5] also improve the original greedy algorithm, and hence propose the NewGreedy and MixGreedy algorithms. The NewGreedy algorithm comprises NewGreedyIC and NewGreedyWC. Chen et al. remove all edges in graph G that are not involved in influence propagation, and thus a new graph G' is generated. In this way, all of the vertices in graph G' are reachable by the seed set S. Generating graph G' is meritorious not only because all of its vertices are reachable by the seed set S, but also the size of the vertices influenced by the vertices activated by the seed set S are also determinable, which simplifies the computation. Chen et al.'s MixGreedy algorithm combines the time advantage of the NewGreedy algorithm with the CELF optimisation. Specifically, it conducts the NewGreedy algorithm in the first round and the CELF optimisation in the remaining rounds, which makes the time efficiency improved in subsequent steps.

2.3.2 Community Detection Algorithms

Discovering the structure of communities in an online social network is essential for understanding the future behaviour of the network. With the prosperous development of online social networks, detecting communities has become a basic process in network science [31]. Community detection has been applied in many existing researches, and numerous methods for detecting communities in online social networks have been proposed.

Hierarchical clustering algorithms have greatly contributed to community detection knowledge. Hierarchical clustering, also called hierarchical clustering analysis, identifies the hierarchical communities in a network. A representative hierarchical clustering algorithm is the Shrink algorithm [32], which is known as an unsupervised clustering algorithm for hierarchical network with no parameter identification. The shrink algorithm takes advantage of the combination of the modularity optimisation approaches and the density-based clustering, and operates under the following basic principles: First, it exploits density-based clustering approach to detect clusters based on the density of vertices. Second, it verifies the quality of the clustering results through modularity optimisation approaches. The remarkable merit of Shrink Algorithm is its ability to detect not only clusters, but also the structure of the network, i.e., the hubs and outliers.

Huang et al. [32] propose two critical parameters for the Shrink algorithm; the structural similarity and dense pair. The structural similarity σ can be calculated by exploiting the Equation 2.2 below. The more mutual neighbours vertices v and u have, the higher structural similarity these two vertices will obtain. If $\sigma(u,v)$ is the largest structural similarity among vertices u and v, and their adjacent neighbours, then (u,v) can be defined as a dense pair. The structural similarity and dense pair parameters facilitate the Shrink algorithm by increasing the ease and accuracy of clustering detection.

$$\sigma(u,v) = \frac{\Sigma_{x\in\Gamma(u)\cap\Gamma(v)} w(u,x) \times w(v,x)}{\sqrt{\Sigma_{x\in\Gamma(u)} w(u,x)^2 \times \sqrt{\Sigma_{x\in\Gamma(v)} w(v,x)^2}}} \tag{2.2}$$

In Equation 2.2, $w(u,x)$ is the number of mutual friends between users u and x.

Another hierarchical clustering algorithm is the community partition method based on node similarity proposed by Ying el al. [33]. In this algorithm, each node is initialized as a community in the network at the beginning. The communities are then merged iteratively based on the neighbourhood similarity. The approach proposed by Ying et al. has lower computational complexity and has been applied in many types of networks, indicating its effectiveness and efficiency in community detection.

Besides hierarchical clustering methods, another classification of community partition methods is named spectral graph partitioning methods, such as, Kcut algorithm. Kcut relies on the eigenvectors of the Laplacian matrix of a graph. Depending on their graph partitioning techniques, these methods are divided into two categories. In the first category, the graph can become bipartite by exploiting the leading eigenvector of the graph. The representative algorithm is the SM algorithm, which computes the second smallest generalized eigenvector and performs a linear search for graph partition. In the second category, a k-way partitioning of the graph is computed through multi-eigenvectors. The representative algorithm is the NJW algorithm, which computes a predefined k number of smallest generalized eigenvectors and builds a matrix $Y = \{\mu_1, \mu_2, \ldots, \mu_n\}$. The graph is partitioned by a standard k-means algorithm based on the normalized unit length of each row of Y, regarding each row as a point. [19] However, as this algorithm is not able to detect the location of vertices, the only available filter condition for selecting seed candidates is the degree of vertices.

Some other methods can be utilized to achieve community partition. For instance, Girvan and Newman [34][35][36] propose the hierarchical divisive algorithm for community detection. In this algorithm, the edges between nodes are removed based on their betweenness, which denotes the shortest paths between pairs of nodes. This is an iterative process and it will remove edges continuously until the modularity of the community detection reaches the maximum value. Furthermore, Blondel et al. [37] propose a fast modularity optimisation method on the foundation of the modularity proposed by Girvan and Newman. This method partitions nodes into communities by merging them into supernodes. The formation of supernodes is interactively until the value of modularity no longer increases. In addition, Donetti and Munoz [38] propose an alternative spectral algorithm, which supposes that in a proper community detection, the values of the nodes in a network are similar to their corresponding eigenvector components in the same community.

2.4 User Preference (UP)

UP also plays an important role in IM, although the existing preference analysis approaches are more often exploited in recommender systems than in IM problems [39][40]. By considering UP, users in a social network can be clustered into different community partitions based on their common preferences. However, many existing researches rarely take UP into account in IM problems. There are two popular methods can be used for discovering user preferences and they are content-based filtering and collaborative-based filtering.

2.4.1 Content-based Filtering Method

Content-based filtering method refers to obtain UPs by comparing the attribute profiles of items with the user profiles. Each user in the network has a profile of recording preferences, and each item has a profile of its corresponding attributes. For example, mobile phone profiles may include brands, screen sizes, systems, and camera pixels etc. The user profile consists of UPs for different mobile phones. In content-based filtering, the compared descriptions of item attributes and UPs are used as a tool of ranking UPs for items. In general, Vector Space Model (VSM) is used for presenting the features of users and items. [39]

2.4.2 Collaborative-based Filtering Method

Collaborative-based filtering method can be defined as a process of retrieving the data or rating patterns using collaborative techniques among multi-agents and data sources. This method obtains a UP by finding the past behaviours of the user, such as ratings and comment history. By utilizing the rating history of the user, the method finds other users with similar behaviours (rating patterns) to predict the preference of the user. Collaborative-based filtering can be divided as memory-based and model-based methods. This method is advantageous as it requires no information related to users and items. [41][42]

2.4.2.1 Memory-based Collaborative Filtering Method

As for memory-based collaborative filtering, it utilizes the similarity between users or items. The computation of preference similarity (ps) can be conducted by the Pearson correlation coefficient (Equation 2.3) and the cosine similarity (Equation 2.4), which are most frequently exploited for similarity measurement in collaborative filtering. [41][39] In addition, memory-based collaborative filtering methods can be further divided into item-based and user-based collaborative filtering.

$$sim(u,v) = \frac{\Sigma_i\{r_{u,i} - \bar{r_u}\} \times \{r_{v,i} - \bar{r_v}\}}{\sqrt{\Sigma_i\{r_{u,i} - \bar{r_u}\}^2} \times \sqrt{\Sigma_i\{r_{v,i} - \bar{r_v}\}^2}} \tag{2.3}$$

In Equation 2.3, $r_{u,i}$ denotes the rating of user u for item i, and $\bar{r_u}$ denotes the average rating of user u.

$$cos(u,v) = \frac{\Sigma_i\{r_{u,i}\} \times \{r_{v,i}\}}{\sqrt{\Sigma_i\{r_{u,i}\}^2} \times \sqrt{\Sigma_i\{r_{v,i}\}^2}} \tag{2.4}$$

Item-based Collaborative Filtering Method

Item-based collaborative filtering is operated on the foundation of an item-centric manner. The relationships between any two items in the network are represented in an item-item matrix. The preferences of a particular user are predicted by comparing the user's existing preferences with the item-item matrix.

User-based Collaborative Filtering Method

As for user-based collaborative filtering, this method can find users who are similar to the target user by computing the ps between the target and candidate users. The drawback of this method is that the relationships between users are unstable. A small change related to user information probably causes the replacement of the whole community of similar users.

2.4.2.2 Model-based Collaborative Filtering Method

The premise of model-based collaborative filtering assumes that there is an underlying model used for managing the way how users rate items. The users' rating patterns are discovered by machine learning algorithms and data mining. Typical models in model-based collaborative filtering include factorization and latent semantic models, and Markov decision process-based models. [41]

2.5 Ant Algorithms and Stigmergy

Ant colonies have fascinated the researchers of computer science field in recent years. The reasons why are that ant colonies can be regarded as decentralised systems and the societies they live is high-structured. In some situations, ant colonies are able to complete complicated tasks that exceed the individual abilities of an ant, which is able to provide methods for solving complicated optimisation and decentralised control problems in an adaptive, flexible and interoperable way. There are two major characteristics of ant behaviours, one is indirect communication, the communication among ants is indirect, they communicate through leaving a kind of chemical substance on the trails; the other one is self-organizing, they can accomplish a task without any control even the task is complicated.

As for ant colonies, stigmergy consists in the main body of ant colony knowledge, as it is a particular mechanism exploited for indirect communication among ants to control and coordinate their tasks. Ant and stigmergy-based algorithms do not much rely on the network topology, and the computation is decentralised. In natural environments, stigmergy-based systems have been demonstrated that they can be utilized for generating complicated and robust behaviours in the systems even if each ant has limited or even no intelligence. Nest building is the representative example of stigmergy.

Some researchers have applied stigmergy for applications in computer science field. For example, Dorigo et al. introduce how to solve the Travel Salesman problem (TSP) [43] by leveraging ant and stigmergy-based algorithms, where the pheromone allocation is concerning the distances among the cities [44]. To be more specific, TSP is in order to solve the problem about the shortest distance for a tour by giving m cities in graph G, and each city should be visited once and only once. Each city is treated as a start point for each ant and m number of ants start their travels at the same time by visiting the m cities sequentially. If there are more than one edge an ant facing, it can be decided by a probability function related to the distance and the amount of pheromone. On each edge, an artificial pheromone trail is updated continuously, which is used to assist ants to make decisions and build tours. When an ant completes a tour, the amount of pheromone on the edges it crawled through will be updated. In addition, pheromone is evaporated over time t. By leaving different amount of pheromone on the corresponding edges, it indicates the quality of the tour. The solution of TSP is presented as an Hamiltonian circuit.

Ahmed et al. propose a stigmergy-based approach for modelling dynamic interactions among web service agents in decentralised environments [45][46]. As for the many existing composition approaches, one of the main challenges is lacking of supports for distributed working environments. In this approach, digital pheromone and pheromone store are used for facilitating control and coordinate objects within web service composition. This approach is operating in a fully decentralised working environment. Each agent requests for an abstract workflow consisting of several sub-tasks. It is capable of searching for suitable resources and services for constructing its workflow by traversing directly connected service agents in the network automatically. In addition, it deposits/withdraws digital pheromone from pheromone stores of other agents automatically as well, which is depending on the quality of the provided services. The experimental results also demonstrate that stigmergy-based approach is able to adapt the web service composition efficiently and effectively by the self-organisation and indirect communication behaviours among service agents stimulated by artificial pheromone. However, this approach has not been applied for real world applications and it does not take trust into consideration.

Takahashi et al. propose anticipatory stigmergy model with allocation strategy for sharing near future traffic information related to traffic congestion management in a decentralised environment [47]. Some of the existing cases utilize past traffic information to estimate the traffic congestion, which deducts the accuracy of the estimation. Nevertheless, Takahashi et al. use near future intention submitted by users as traffic data resources for estimation, which can improve the accuracy effectively. In this research, Takahashi et al. compare the anticipatory stigmergy approach with other five approaches. Through implementation, the experimental results also present that the anticipatory stigmergy with allocation strategy has a better performance than other methods. However, this approach does not take dynamic environments, such as, accidents, into consideration. In addition, the scale of the maps used for experiments are small.

Besides developing applications with stigmergy, some distributed optimisation approaches are also developed using stigmergy, e.g., the ant colony optimisation (ACO) metaheuristic. It is inspired by ant foraging. This method is used to find reasonable minimum cost paths over a graph G with a set of defined constrains. This method also can be used to solve TSP.

2.6 Summary of Literature Review

The above reviews highlight the limitations of existing centralised approaches. First of all, we note that classic diffusion models, such as, the IC and LT models, consider only TC, which results in that IM is treated as a simplified probabilistic problem. In addition, the influence probabilities are predefined and static. Second, UP plays a critical part in IM since users with common preferences will more easily influence each other than users with dissimilar preferences. However, this factor is not taken into consideration in most of the existing approaches. In general, centralised approaches have low effectiveness and efficiency.

Moreover, to improve the efficiency of seed selection algorithms in the IM problem, researchers have developed a variety of methods. Chen et al. study the efficient influence maximisation by improving the original greedy selection algorithm and propose a novel seed selection approach, namely, degree discount heuristics for the uniform IC model, which assigns the same probability to all edges. Their model efficiently improves the seed selection of IM. [5]. Leskovec et al. propose a novel algorithm that finds the influential nodes (seeds) based on a simple greedy algorithm. Their so-called Cost Effective Lazy Forward (CELF) scheme [29], which deducts the running time of seed selection. Furthermore, Goyal et al. extend the original CELF algorithm to an optimised algorithm, called CELF++ algorithm, which reduces the running time of subsequent steps and it has been demonstrated that the CELF++ algorithm is 35 to 55 percent faster than the original CELF algorithm [30]. Zhang et al. research the least Cost Influence Problem (CIP) in multiplex networks, and alleviate its problems by mapping a set of networks into a single network via lossless and lossy coupling schemes [48]. However, all of these approaches are applicable only to static networks, and require discovery of the network topology. Specifically, they cannot handle the dynamics of social networks. Meanwhile, traditional approaches are not applicable when the global perspective is unavailable.

Chapter 3

The Preference-based Trust Independent Cascade (PTIC) Model

3.1 Introduction

In this chapter, we tend to explore an appropriate centralised approach for the IM problem. Through the review of the previous literature related to classic influence diffusion models, the existing problems are identified as follows:

- Most of the existing approaches, such as the IC and LT models, only take TC into account, which reduces IM to a simplified probabilistic problem;
- The influence propagation probabilities are predefined and assumed as static;
- User preferences to particular items are not considered;
- The existing models are ineffective and inefficient.

Motivated by the prosperous development of viral marketing and in order to overcome the limitations of existing approaches, a Preference-based Trust Independent Cascade (PTIC) Model is proposed by considering both UP and TC. Furthermore, the influence probabilities in the PTIC model are computed based on the UP and TC rather than predefine them. In this approach, hub users in a social network, who are interested in the promoted items, can be selected as influential users. Thereby, the effectiveness and efficiency of IM can be significantly improved. In addition, as verified in experiments, the PTIC model outperforms the trust-only and random methods.

As for the developmental process of the PTIC model, there are two critical factors, i.e., the UP and TC, involved in. UP, which specifies the preference degree of a particular user for an item. It is a critical subjective factor of influence propagation. It is mainly involved in community partitioning by the hierarchical clustering algorithms, and in seed selection by degree-based ranking. Depending on the data of the proposed system exploited, the methods used to retrieve user preferences can be classified into two categories: one is content-based filtering method and the other is collaborative-based filtering method. Content-based filtering method is based on the comparison between user profiles and item attributes, whereas collaborative-based filtering method is based on identifying the users with similar rating patterns and rating histories to predict the rating of a particular user for an item. The proposed PTIC model is developed based on a collaborative-based filtering method. However, users with common preferences are partitioned into the same community regardless of whether or not they know each other, which may lower the quality of the seed set selection. Hence, we must check that two users with common preferences share a corresponding trust connection in the network, such a connection will guarantee an influence propagation channel between users to propagate influence.

The remainder of this chapter is organized as follows. In Section 2, it describes the problem and formally defines the PTIC model. In Section 3, the framework of the PTIC model is described in detail. Experiments are presented in Section 4. Finally, this chapter is concluded in Section 5.

3.2 Problem Description and Formal Definitions

3.2.1 Problem Description

Suppose that an organisation plans to promote a particular product (i_x) in an online social network. Due to limited resources, the organisation needs to select a limited amount of influential users to experience the product and promote it to their connected users. Ideally, the selected users will maximize the influence in the network. The purpose of IM is to select k influential vertices, also called seeds, from the social network.

The prospective number of vertices influenced by the selected seeds is regarded as the achieved influence.

3.2.2 Formal Definitions

In this chapter, we assume that there are m users, n edges, and x items in a social network. Here, a social network is modelled as a graph $G = (V, E)$, where V denotes the set of users, and E denotes the set of edges among the users. There are two types of edges in G, i.e., preference edges and trust edges, representing the UPs and TCs among the users, respectively (see Definition 5 and 6).

Definition 1: A **user** is defined as a vertex v_j in the network. Each user has a set of neighbours, i.e., $N_j = \{v_j \in V | v_j, v_i \in E\}$. Each vertex in N_j ($v_i \in N_j$) has a Trust Edge (see Definition 6) to v_j.

Beside users, the network also contains a set I of items. An item $i_x \in I$ is a particular product that has been or will be promoted to the users in the network. User v_j's preference to item i_x is presented as the result of the ratings given by v_j.

Definition 2: Rating r_{jx} is the preference degree of user v_j for item i_x. The rating set $R_j = (r_{j1}, r_{j2}, r_{j3}, ..., r_{jx})$ is the set of all ratings previously given by v_j.

Definition 3: Common Preference for Item (cpi_{ijx}) is defined as the ratings that any two users in the network gave for item they both rated, where cpi_{ijx} denotes that item i_x is the common preference of user v_i and v_j.

$$cpi_{ijx} = 1 - \frac{|r_{ix} - r_{jx}|}{r_x.max - r_x.min} \tag{3.1}$$

In Equation 3.1, r_{ix} is the rating given to item i_x by user v_i, and $|r_{ix} - r_{jx}|$ indicates the rating difference of i_x between users v_i and v_j. The quantity $r_x.max - r_x.min$ is the difference between the maximum and minimum rating value for item i_x.

Definition 4: Common Preference Similarity (cps_{ij}) is defined as the similarity of the ratings given by network users v_i and v_j for all items rated by both two users. Users with common preferences will be computed for CPS and labelled CPS as weight on preference edges.

$$cps_{ij} = \frac{\sum_{x \in I} cpi_{ijx}}{I.count} \tag{3.2}$$

Equation 3.2 shows the CPS calculation between users v_i and v_j. I denotes the item set rated by both v_i and v_j, and I.count indicates the number of items in I.

Definition 5: A **Preference Edge** pe_{ij} denotes the preference relationship between two users, i.e. v_i and v_j. Users v_i and v_j will have a preference edge pe_{ij} when their cpi_{ijx} is calculable. The weight of pe_{ij} can be denoted as $w(pe_{ij}) = cps_{ij}$.

Definition 6: A **Trust Edge** te_{ij} denotes the trust relationship between two users, i.e. v_i and v_j. The weight of te_{ij} can be represented as $w(te_{ij}) = user\ distance$, which is computed from the n-dimensional coordinates of the information (attributes) in the user profiles (n is depending on the number of attributes). The range of user distance is from 0 to 1.

Definition 7: **Influence Probability** p_{ij} is defined as the likelihood that the influence will propagate from user v_i to user v_j. The range of p_{ij} is from 0 to 1. p_{ij} is computed by the product of cps_{ij} and the user distance between users v_i and v_j and it can be expressed as Equation 3.3.

$$p_{ij} = w(pe_{ij}) \times w(te_{ij}) \tag{3.3}$$

Definition 8: A **Community** C_r, refers to a set of users in any scale that have common preference(s). Users in a community have a compact relationship related to common preference among each other, even for those who are not linked directly. A social network G can be partitioned into a number of communities, i.e., $G = C = C_1, C_2, C_3, ..., C_r$. Here we assume that there are no intersections between the communities.

The reason why we conduct community partition is that assuming we plan to propagate influence within a given community C_i, selecting influential users from community C_i to propagate influence is more efficient than selecting influential users from other communities [49]. Each community presents not only the CPS between any two users in the network, but also the relationship related to the common preference among a set of users in a community.

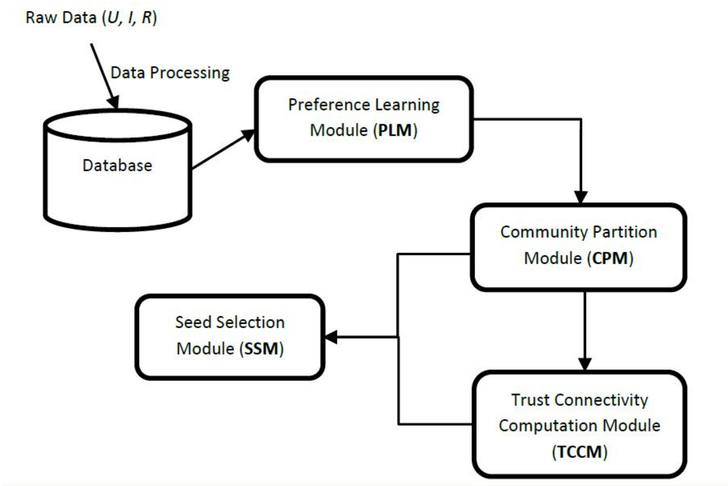

FIGURE 3.1: Framework of the PTIC Model

3.3 The Preference-based Trust Independent Cascade (PTIC) Model

The framework of the PTIC model is shown in Figure 3.1. There are four modules in the PTIC model, i.e., the Preference Learning Module (PLM), the Community Partition Module (CPM), the Trust Connectivity Computation Module (TCCM), and the Seed Selection Module (SSM). At the beginning of this model, the CPSs (see Definition 4) between the users will be computed and evaluated by the PLM. The computation of CPS can be regarded as the preparation of the CPM, as the CPSs are the weights of the preference edges. After computing the CPSs, the CPM will be conducted by partitioning the users into communities based on their CPSs. The TCCM computes the user distances based on the user profiles, thus providing the TC. Finally, the outputs of the CPM and TCCM are input to the SSM which selects the influential nodes.

3.3.1 Preference Learning Module (PLM)

In this module, the CPS is calculated based on the rating differences between two users, v_i and v_j. The smaller the average of their rating differences is, the higher the CPS will be. The cps_{ij} between v_i and v_j can be calculated by using Equations 3.1 and 3.2.

Algorithm 1 Common Preference Similarity (CPS) Computation Algorithm

Input: V, R

Output: CPS

1: Load the Users and Ratings from the process dataset
2: **for** $\forall v_i \in V$ **do**
3: **for** $\forall v_j \in V \wedge i > j$ **do**
4: Find common rating pairs $R', r_i, r_j \in R', R'.I_i = R'.I_j$
5: Calculate CPS between user v_i and user v_j, cps_{ij}
6: $cps_{ij} = cps_{ji}, cps_{ij} \in CPS, and\ cps_{ji} \in CPS$
7: **end for**
8: **end for**

Algorithm 1 shows the process for calculating the CPS. The algorithm accepts the user set V and user-item rating set R as inputs. The network is defined as an undirected network; that is, a user and a counterpart are compared once only.

As for the CPS, it represents the common preference similarity between any two users in the network (see Definition 4). The common preference similarity among users within a community, it can be obtained by partitioning the network into communities. Users with close CPSs will be partitioned into the same community, and each node in a community will be related by a common preference similarity. In other words, the members of a community share a common preference even when those users are not directly linked.

3.3.2 Community Partition Module (CPM)

In CPM, the community partition approach is derived from the community detection algorithm proposed by Ying et al. [33]. All of the users are randomized, and each individual tries to merge with its closest neighbour. This procedure will be conducted iteratively until the similarity among the communities (referring to clusters in the algorithm) reaches a certain threshold σ.

The user preference clustering algorithm is shown in Algorithm 2. This algorithm accepts two input variables, a user set V and a UP matrix P. The output C indicates the tree-like hierarchical UP cluster. The algorithm terminates when the similarity among the communities reaches a certain threshold. In each iteration, the nodes merge with their neighbours with the closest CPSs. All edges of the merged nodes are updated accordingly. In the merging process, as for those common neighbours' edges, the one with higher weight is selected, while the edges of all non-common neighbours' nodes

Algorithm 2 User Preference Clustering Algorithm

Input: V, CPS

Output: C

Load the user dataset,V and user preference matrix,P
Randomize the user set
Initialize cluster, $C.size = V.size$
while average similarity among clusters $> \sigma$ **do**
    for $\forall v_i \in V$ **do**
        for $\forall v_j \in V \wedge i > j$ **do**
            Find v_j neighbour v_j with the maximum cps_{ij} in P
            if $cps_{ij} > threshold\ t_s$ **then**
                merge (v_i, v_j) into a new node v_n
                $V.size - 1$, assign v_n to a new cluster C_n
                for $\forall v_c \in \Gamma(v_i) \cap \Gamma(v_j)$ **do**
                    $cps_{nc} = max(cps_{cj}, cp_{ci})$
                end for
                for $\forall v_c \in \Gamma(v_j) \cup \Gamma(v_j)$ **do**
                    $cps_{nc} = cps_{ci} = cps_{cj}$
                end for
            end if
        end for
    end for
end while

directly point to the merged node. After a number of iterations, a tree-structured cluster is generated.

3.3.3 Trust Connectivity Computation Module (TCCM)

The main purpose for computing TC is to ensure that users with common preferences establish an influence propagation channel. Without the TC, users with common preferences in the same community cannot partake in the influence propagation. The TC is computed based on the information (attributes) in the user profiles. When necessary, this information is quantified and stored in a n-dimensional coordinate (n is depending on the number of attributes). The TC is then quantified by the user distance, calculated by Equation 3.4.

$$w(te_{ij}) = 1 - \sqrt{\sum_{a_m \in A} \left(\frac{v_i.a_m - v_j.a_m}{a_m.max - a_m.min}\right)^2} \tag{3.4}$$

In Equation 3.4, $w(te_{ij})$ indicates the weight on the TC between users v_i and v_j, and $v_i.a_m - v_j.a_m$ denotes the distance difference of the attribute a_m of users v_i and v_j, where a_m is an element of the attribute set A.

Algorithm 3 User Trust Connectivity Computation Algorithm

Input: V, A
Output: T

Load the User dataset V including all the users' attribute set A
for $\forall v_i \in V$ **do**
  for $\forall v_j \in V \wedge i > j$ **do**
    Calculate the weight of TC between user v_i and v_j
    $w(te_{ij}) = w(te_{ji})$
  end for
end for

As mentioned above, the weight on TC is depending on the users' attributes. In Algorithm 3, the input is user set V and user attribute set A, and the output is a user trust matrix T.

3.3.4 Seed Selection Module (SSM)

With the involvement of community partition and TC computation, users have not only common preferences, but also TC in the network. The influence probability p_{ij} is calculated based on the product of the UP and TC between users v_i and v_j. After computing the influence probability p_{ij}, hub users who are interested in the promoted item will be selected as the influential users (seed set). The seeds are selected based on a heuristic method in this chapter. Based on budget, this module selects p number of seeds with high influence spread.

Algorithm 4 aims to calculate the activated users influenced by the seed set in the network by using the IC model. The input is the seed set $\{v_a\}$, and the output is a set of activated users V_a in the entire network. p_{ai} denotes the influence probability between two users v_i and v_a (see Definition 7). Each iteration of Algorithm 4 (Lines 3-14) finds user v_a's neighbour set $V_n, v_i \in V_n$. If v_i is inactive and its Influence Propagation Probability (IPP) is larger than the threshold, then v_i is activated and its neighbours $\Gamma(v_i)$ are influenced by the IPP', where $IPP' = IPP \times p_{ix}$ $(x \in \Gamma(v_i))$. This algorithm is recursive, meaning that it invokes itself inside the procedure (Line 11). For example, if v_a is an element of the seed set, then $v_a's$ $IPP = 1$, the IPP of $v_a's$ neighbour v_i is $1 \times p_{ai}$, and the

Algorithm 4 Influence Propagation Algorithm using IC Model

Input: $\{v_a\}, \{v_a\} \subseteq Seeds\ Set$
Output: V_a

Initialize $IPP = 1$ if not a recursive invoke
$v_a.activeStatus = true$
for $\forall v_i \in \Gamma(v_a)$ **do**
    if $v_a.activeStatus = true$ **then**
        Next
    end if
    if $p_{ai} \times IPP \geq propagation\ threshold$ **then**
        Generate a random decimal $d_r, 0 \leq d \leq 1$
        if $d_r \leq p_{ai} \times IPP$ **then**
            $v_i.activeStatus = true$
            Update IPP, input v_i as variable and invoke self - Recursive
        end if
    end if
end for

IPP of $v_i's$ neighbour v_j is $1 \times p_{ai} \times p_{ij}$. Hence the IPP steadily reduces as the number of hops of influence propagation increases.

3.4 Experiments and Analysis

In this section, two experiments are conducted to compare the performances of the PTIC model with two other approaches, i.e., the random approach and the trust-only approach. In the random approach, the seeds are randomly selected from users. In the trust-only approach, seed selection is based on only the TC weight.

We estimate the total number of activated users influenced by the seed set generated by the trust-only and random approaches. If a user is selected into the seed set, that user will attempt to influence and activate its neighbours in the network. The activation probability of the neighbours is determined by the weight of the influence propagation channel.

In Experiment 1, the weight of the influence propagation channel is determined by both UP and TC; in Experiment 2, the influence propagation depends only on the TC.

3.4.1 Data Selection

Experiments are performed on the Movielens[1] dataset, a stable benchmark dataset released in April of 1998 that contains the ratings of 1682 movies from 943 users. To filter the noise data, users who input less than 50 ratings have been removed from the dataset. Furthermore, users with ambiguous or false attributes are also eliminated from the experiments. After the data preprocessing, 441 users remains in the experimental dataset.

3.4.2 Experimental Results

The evaluation results are plotted in Figures 3.2 and 3.3. In both figures, the x-axis denotes the size of the seed set, i.e., the number of selected influential users, and the y-axis refers to the number of activated users in the entire network.

Figure 3.2 compares the performances of the three approaches in Experiment 1. Among the 441 users, approximately 280 users are capable of activating their neighbours (the remaining users can activate only themselves). In order to ensure the accuracy of an individual's influence spread, we conduct each trial for 100 runs and compute their average. The seed set with multiple elements is treated by the same method. The seed set is increased by retaining the selected users and adding new users.

As shown in Figure 3.2, the PTIC model outperforms the two alternative approaches. In terms of the cost performance, the appropriate size of the seed set is 10, the elbow point in the PTIC model. In addition, the number of activated users in the network reached 272, significantly higher than the other two approaches.

Figure 3.3 presents the performance of the three approaches in Experiment 2. This experiment is conducted in trust-only network. Trust-only network is a homogeneous network, which only takes physical links into consideration. In this situation, the seeds selected by trust-only approach should best suit the network compared with other approaches. Although the trust-only approach yields the best performance, the PTIC model still performs very well, and is far superior to the random approach. When the size of seed set is small (ranging from 4 to 6), the seeds selected via the trust-only approach can activate approximately 300 users, whereas those selected by the PTIC

[1] https://grouplens.org/datasets/movielens

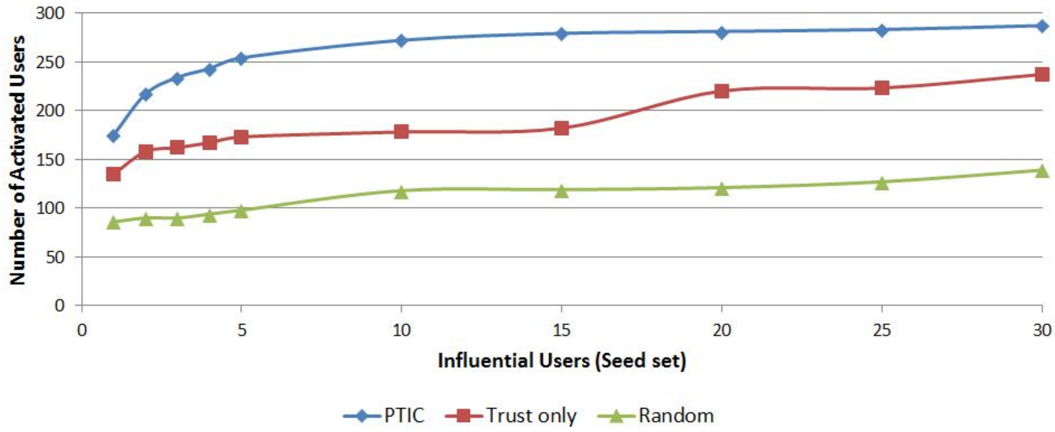

FIGURE 3.2: Preference Evaluation of Three Models Considering User Preference and Trust

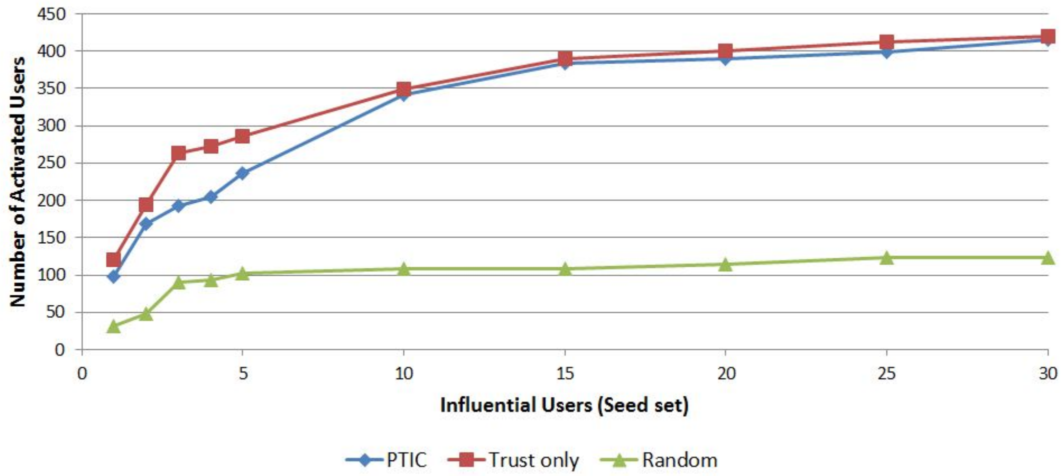

FIGURE 3.3: Preference Evaluation of Three Models Considering Trust Only

model can activate 250 users. Furthermore, when the seed size reaches 10, the number of activated users is very similar in the trust-only approach and the PTIC model.

Based on the above discussion, we could claim that when the resources are limited, the PTIC model yields the best performance among the three tested approaches.

3.5 Summary of The Preference-based Trust Independent Cascade (PTIC) Model

In this chapter, in order to resolve the existing problems of some classic influence diffusion models, we propose the PTIC model and apply it to the IM problem on the foundation of UP and TC. In addition, compared with previous researches, we compute the influence probabilities based on the UP and TC rather than predefine them, which can improve the quality of the seed selection significantly. The experimental results also prove that the PTIC model has a better performance than trust-only and random approaches. Hence, by including both UP and TC, the PTIC model is able to propagate influence in a largest possible range, realising an effective and efficient dissemination of limited resources.

Chapter 4

The Stigmergy-based Influence Maximisation (SIM) Model

4.1 Introduction

In this chapter, a decentralised approach is developed for the IM problem. Different from centralised approaches, decentralised approaches can distribute the workload, further improving the effectiveness and efficiency of seed selection. The existing problems of centralised approaches are presented as follows:

- Centralised approaches are ineffective and inefficient;
- Centralised approaches are complicated to compute.

To solve the existing problems above, the Stigmergy-based Influence Maximisation (SIM) model is proposed by exploiting a novel decentralised approach, i.e., stigmergy-based approach. Stigmergy is an important proportion of knowledge from ant colonies. Ant colonies have recently fascinated researchers in computer science field since ant colonies can be utilized to solve complex distributed control and optimisation problems. The essential characteristics of decentralised approaches, e.g., ant and stigmergy algorithms, are that they are able to facilitate online social networks by improving their adaptability, flexibility and interoperability. Decentralised networks can be regarded as multi-agent systems because decentralised approaches tend to distribute the workload to individuals, and each individual cooperates with other individuals in task implementation. The individuals' actions eventually lead to a global convergence.

There are two kinds of decentralised approaches in terms of communications. One relies on the direct communications among the individuals, such as cellular automata [50], in which each cell in the grid adapts its state by examining its adjacent neighbours based on a set of rules. While, the other focuses on the indirect communications, in which individuals read or analyse the 'stimulations' left by their peers. One of the typical approaches is ant and stigmergy algorithm [44]. The French entomologist, Pierre-Paul Grasse defines stigmergy as "stimulation of workers by the performance they have achieved", which is associated with two major features of ants [51]. First, the communications among ants are indirect. More specifically, stigmergy is a particular indirect communication mechanism by which ants exploited to harmonize their daily tasks with each other. Their indirect communication is conducted through depositing pheromone on the trails, which is a kind of chemical substance that evaporates over time. Second, ants' activities are self-organized. Individual ants can complete a complicated task independently and autonomously. Stigmergy has been developed and applied to diverse problems, such as, communication network routing, exploratory data analysis, and diagram drawing.

There are two obvious advantages of applying the ant and stigmergy algorithm to the IM problem. First, as for seed selection, the ant and stigmergy algorithm can improve the performance and the operating time effectively and efficiently since ants are able to separately accomplish even complicated tasks without any control. Second, since the stigmergy-based approach is an optimisation process, so that it can continuously guide the output of seed selection towards the optimal solution.

In this chapter, we tackle the IM problem by a novel decentralised approach, the Stigmergy-based Influence Maximisation (SIM) model which simulates the influence propagation process as ants' crawling across the network topology. Furthermore, the ant's key behaviours, including path selection and pheromone allocation, have been modelled for selecting suitable nodes to achieve the IM. Path selection aims to identify the next node to approach when an ant faces multiple options. While, the objective of pheromone allocation is to deposit pheromone on specific nodes based on the heuristics when an ant explores a possible influence-diffusion path. Experiments have been conducted to evaluate the performance of the SIM model by comparing against traditional seed selection algorithms; namely, greedy selection, degree-based selection and random selection algorithms. The results demonstrate that the proposed model is more advanced by considering both effectiveness and efficiency. Moreover, the SIM model can dramatically reduce the computational overhead, compared with centralised approaches.

The remainder of this chapter is organized as follows. Section 2 systematically elaborates the SIM modelling approach, including the problem description, formal definitions, path selection and pheromone operations. In Section 3, the performance of the SIM model is experimentally evaluated. Finally, the paper is concluded in Section 4.

4.2 The Stigmergy-based Influence Maximisation (SIM) Model

The SIM model tends to select appropriate influential candidates by considering the influence strengths among users and the assembled influential effect. In this model, numerous ants walk simultaneously and update their shared environment by allocating pheromone. The influence propagation process is simulated by the crawling behaviours of ants. The influential users can be identified when the pheromone distribution in the network starts to converge. The amount of pheromone at each node then determines the seed selection. The SIM model will be elaborated in the following subsections.

4.2.1 Problem Description

Suppose that an organisation plans to promote a particular product (i_x) in a large-scale online social network. Due to limited resources and insufficient time, the organisation needs to select k initial candidates as influential users, who will experience the product as soon as possible and hopefully recommend it to others in their social cycle. Ideally, the k influential users exert the maximum influence in the social network.

4.2.2 Formal Definitions

Definition 1: A **Social network** is defined as a weighted graph $G = (V,E)$ with a clear topological structure, where $V = \{v_1, v_2, ..., v_n\}$ stands for the nodes (users) in the network, and $E = \{e_{ij} | v_i \in V \wedge v_j \in V, v_i \neq v_j\}$ denotes the edges (relationships) among the nodes. A particular edge can be represented as a three-tuple, i.e., $e_{ij} = (v_i, v_j, w_{ij})$, where w_{ij} is the weight of e_{ij} which represents the influence strength. Each node v_i has a set of neighbours $\{v_j | v_j \in \Gamma(v_i), e_{ij} \in E\}$. While, $v_i.p$ indicates the pheromone amount (see Definition 4) accumulated on corresponding node v_i, which can be regarded as an attribute of v_i. Similarly, the weight w_{ij} is denoted by using the notation $e_{ij}.w$ to indicate its association with the edge e_{ij} in this chapter.

Definition 2: An **Ant** a_m is defined as an autonomous agent in the network G, which crawls across the network topology of G. a_m can be represented as a three-tuple, i.e., $a_m = (m, q^n_m, T^n_m)$, indicating that ant a_m carries q^n_m pheromone during tour T^n_m (see Definition 3). There exist a number of ants, $A = \{a_1, a_2, ..., a_n\}$, in the social network, and they continue to crawl through the network. As they travel, the ants discover and evaluate the amount of pheromone on their current and nearly nodes. However, the ants cannot directly communicate with each other.

Definition 3: A **tour** $T^n_m =< v_1, v_2, ..., v_n >$ is defined as the path walked by ant a_m in the $n-th$ round. Specifically, ant a_m randomly selects a starting point, then crawls from one node to an adjacent node until it reaches the end point v_e, where $\Gamma(v_e) \subset T^n_m \cup |\Gamma(v_e)| = 1$.

Definition 4: Pheromone refers to a kind of chemical substance deposited by the ants. In this context, the pheromone passes the information and heuristics from an ant to its peers based on its experience. q^n_m denotes the total amount of artificial pheromone carried by ant a_m in the $n-th$ round. Once a_m has completed its tour, its pheromone is distributed to each node of T^n_m.

4.2.3 Path Selection

In this context, path selection is one of the ant's basic behaviours, that describes how an ant a_m located at node v_i selects the next node to reach among multiple choices $V_c = \{v_j | v_j \in \Gamma(v_i) \wedge e_{ij} \in E\}$.

Basically, the path selection decision depends on two factors; the pheromone amount on v_j, i.e., $v_j.q$, and the weight of the corresponding edge $e_{ij}.w$. The path selection behaviour is modelled as an probabilistic event by using Equation 4.1, where p_{ij} denotes the probability that an ant walks from node v_i to v_j.

$$p_{ij} = \begin{cases} \frac{e_{ij}.w \cdot v_j.q}{\sum_{v_x \in \Gamma(v_i)} e_{ix}.w \cdot v_x.q}, & e_{ij} \in E \\ 0, & e_{ij} \notin E \end{cases} \tag{4.1}$$

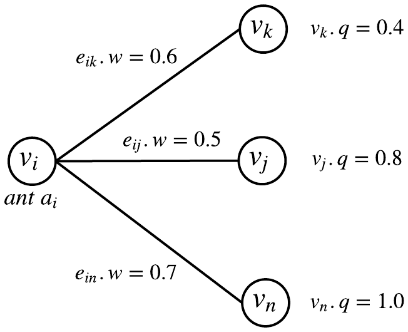

FIGURE 4.1: Path Selection of an Ant

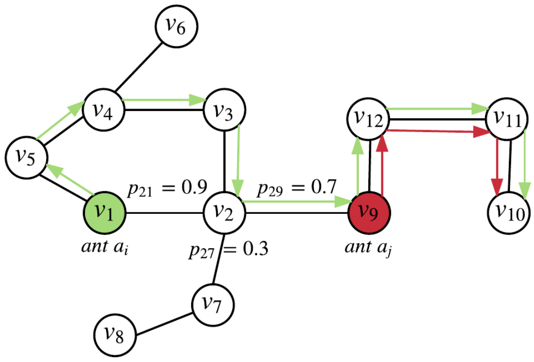

FIGURE 4.2: Path Selection of Multiple Ants

To demonstrate path selection, we provide two concrete examples. In Figure 4.1, ant a_i starts from node v_i and confronts three options, i.e., v_k, v_j and v_n. The decision is made by considering both the targeting nodes' pheromone amount and the influence strength / weight of the corresponding edges, as indicated in Equation 4.1. In this diagram, the probability of choosing node v_j is calculated as: $p_{ij} = e_{ij}.w \cdot v_j.q/(e_{ij}.w \cdot v_j.q + e_{ik}.w \cdot v_k.q + e_{in}.w \cdot v_n.q) = 0.8 \times 0.5/(0.4 \times 0.6 + 0.8 \times 0.5 + 1.0 \times 0.7) = 29.85\%$

In Figure 4.2, another example is demonstrated, where two ants, labelled a_i and a_j, walk through the same network. The path selection principles prevent the ants from choosing previously self-visited nodes within the same tour, but they can intersect nodes previously visited by other ants in either the current or previous iterations.

Each ant keeps performing an iterative process: walking and selecting path. This process stops when the ant reaches the end point. The iterative process triggered by ant m in round n is recorded in a path vector called tour T_m^n. The tour formation is described in Algorithm 5.

Algorithm 5 presents the process of how an ant complete a tour. The input to the touring procedure includes an ant a_m and the index of round n, and the output is a tour T_m^n. Line 3 specifies the criterion for walking to the next node. Lines 5-10 implement the target candidate selections, where the predefined threshold σ filters out the low-probability candidates. Lines 11-17 indicate the path selection process. The iterative walking process terminates when all of the current node v_s's neighbours reside in the tour list T_m^n.

Algorithm 5 Tour Formation Algorithm

Input: a_m, n

Output: $T_m^n, T_m^n \subseteq V$

Initialize a_m and random select a starting point $v_s, v_s \in V$
Initialize a tour list $T_m^n = \emptyset$
while $\exists \Gamma(v_s) \wedge \Gamma(v_s) \not\subset T_m^n$ **do**
  Initialize candidate list $V_c = \emptyset$
  for $\forall v_i \in \Gamma(v_s) \wedge v_i \notin T_m^n$ **do**
    Compute the probability p_{si} using Equation 4.1.
    if $p_{si} > \sigma$ **then**
      $V_c = V_c \cup \{v_i\}$
    end if
  end for
  if $V_c \neq \emptyset$ **then**
    Determine the next node $v_n \in V_c$ using Equation 4.1.
    $T_m^n = T_m^n \cup \{v_n\}$
    $v_s = v_n$
  else
    $v_s = null$
  end if
end while

4.2.4 Pheromone Operations

4.2.4.1 Sub-network Generation

Sub-network generation is the preliminary step of the pheromone operations. After ant a_m completes a tour T_m^n, a corresponding sub-network $G_m^n = (V_m^n, E_m^n)$ will be generated based on the path walked by a_m. V_m^n incorporates all nodes in tour T_m^n and their valid first-layer neighbours $\Gamma(T_m^n)$, thus, $V_m^n = T_m^n \cup \Gamma(T_m^n)$. Meanwhile, the edge set E_m^n include all links among V_m^n.

The total amount of pheromone q_m^n during tour T_m^n depends on the total number of nodes in the sub-network, i.e., $|V_m^n|$. Each node in the sub-network contributes one proportion of pheromone. Figure 4.3 illustrates a typical example of generated sub-network. In this example, an ant sequentially walked from node v_a to node v_e. Based on these nodes, search for their valid first-layer neighbours. In this way, the sub-network is generated.

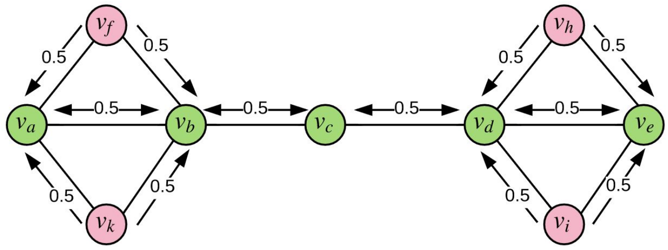

FIGURE 4.3: Pheromone Allocation in A Tour with Five Nodes

4.2.4.2 Pheromone Allocation

In general, pheromone allocation refers to how ants deposit the biological information on the nodes after the ants visit the nodes in the network. The distribution of pheromone plays an important role in stigmergy algorithms, since it updates the context according to the relevant impact factors. In this way, the solution becomes continuously more optimised.

In the current setting, the pheromone distribution is based on the size of the sub-network. The shorter the length path and the larger the sub-network, the more pheromone will be allocated on each node. The number of connected neighbours of node v_i in sub-network G_m^n is computed by Equation 4.2. Equation 4.3 computes the pheromone accumulation of node v_m in tour T_m^n. The accumulated pheromone is the sum of all pheromone contributions given by the direct neighbours $\Gamma(v_m)$.

$$v_i.N = |\{v_i | v_i \in V_m^n \wedge \Gamma(v_i) \in T_i^n\}| \tag{4.2}$$

$$v_m.\Delta q = \begin{cases} \sum_{v_i \in \Gamma(v_m)} \frac{1}{v_i.N}, & v_m \in T_m^n, v_i.N \neq 0 \\ 0, & v_m \in T_m^n, v_i.N = 0 \end{cases} \tag{4.3}$$

Figure 4.3 illustrates an example of a specific sub-network $G_m^n = (V_m^n, E_m^n)$, where the tour travelled by ant a_m is represented as $T_m^n = < v_a, v_b, v_c, v_d, v_e >$, $V_m^n = \{v_a, v_b, v_c, v_d, v_e, v_f, v_k, v_h, v_i\}$ and E_m^n includes all edges among the nodes in V_m^n, $|E_m^n| = 12$ in this example. Because node v_f is the direct neighbour of two nodes in tour T_m^n, both v_a and v_b obtain half of a unit pheromone from v_f. Meanwhile, node v_b contributes 0.5 unit of pheromone to v_a and v_c, but v_f and v_k are beyond the scope of this allocation. Therefore, the pheromone gains at nodes v_a and v_b are calculated as 1.5 and 2.0, respectively.

Algorithm 6 implements the pheromone allocation process initiated by ant a_m in tour T_m^n. The distribution is based on the topology of the explored sub-network G_m^n. The input is a specific tour T_m^n and the output is a pheromone amount update. Specifically, this algorithm alter the context of the network by updating the pheromone amount at each node of the tour path. Lines 1-9 initialize and construct the sub-network G_m^n. Lines 10-12 obtain the denominator of Equation 4.3 at each node which allocates pheromone to the nodes in the tour path. Lines 13-14 compute the variations of the pheromone.

Algorithm 6 Pheromone Allocation Algorithm

Input: T_m^n

Output: *pheromone changes for all the nodes in* T_m^n

1: Initialize sub-network graph $G_m^n = (V_m^n, E_m^n), V_m^n = \emptyset, E_m^n = \emptyset$
2: **for** $\forall v_i \in T_m^n$ **do**
3:     **for** $\forall v_j \in (\Gamma(v_i) \cup v_i)$ **do**
4:         $V_m^n = V_m^n \cup \{v_j\}$
5:         **if** $p_{ij} > 0 \wedge i \neq j$ **then**
6:             $E_m^n = E_m^n \cup \{e_{ij}\}$
7:         **end if**
8:     **end for**
9: **end for**
10: **for** $\forall v_n \in V_m^n$ **do**
11:     Compute $v_n.N$ using Equation 4.2
12: **end for**
13: **for** $\forall v_m \in T_m^n$ **do**
14:     $v_m.q = v_m.q + v_m.\Delta q$, using Equation 4.3
15: **end for**

4.2.4.3 Pheromone Evaporation

Pheromone evaporation is a common biological phenomenon, which reduces the amount of allocated pheromone over time. In ant and stigmergy algorithms, pheromone evaporation discourages the convergence to a locally optimal solution. Pheromone simultaneously evaporates through each node within the scope of the whole network. At a justified time, all of the nodes in the network will have evaporated a predefined unit of pheromone. The pheromone evaporation at each node is quantified by using Equation 4.4, where Δt is the time difference and λ is the evaporation speed of the pheromone.

$$EQ = e^{\frac{\Delta t}{\lambda}}, \lambda \neq 0 \tag{4.4}$$

4.2.5 Seed Selection

Seed selection aims to select the set of influential users that will propagate influence to other members from a specific network. Various classic seed selection approaches are available. Among these degree-based seed selection chooses the nodes with high node degree. Intuitively, users with large friend cycles can influence more users in the social network than users with few friends. However, in general, two connected users with very high degree tend to have many common friends, so that selecting one or both of these high influencers will make little difference to the impact. Another well-known approach is greedy selection, which aims to obtain the maximum influence marginal gain through selecting the seeds. However, this approach is not applicable to large-scale network because of its high computational overhead. Random selection is also applied to some cases, but generally gives poor performance because it is not based on any heuristics.

In stigmergy-based algorithms, seed selection depends on the amount of pheromone allocated on each node. The selection is similar to the degree-based approach, but influential users are determined by ranking the pheromone degree of each node in stigmergy-based algorithms.

Algorithm 7 Seed Selection Algorithm

Input: $n, k, \lambda, \Delta t, G = (V, E)$
Output: V_s

Initialize ant set $A = \{a_1, a_2, ..., a_n\}$ which contains n ants.
Initialize seed set $V_s = \emptyset$
All the n ants start to crawl in network G in the distributed servers.
while !convergence **do**
    Compute EQ using Equation 4.4.
    for $v_i \in V$ **do**
        $v_i.q = v_i.q - EQ$
    end for
    Sleep for Δt
end while
Sort V order by q descend
for $\forall v_i \in V$ **do**
    if $|V_s| < k$ **then**
        $V_s = V_s \cup \{v_i\}$
    end if
end for

In Algorithm 7, the input includes the number of ants n, the size of the seed set k, evaporation speed λ, time difference Δt and the network $G = (V, E)$. Lines 1-2 initialize the ants and seed sets. Lines 3 mimics the ants' autonomous behaviours in the network by calling Algorithms 5 and 6. Lines 4-10 implement the global pheromone evaporation process. Lines 11-16 select the seeds from the updated environment.

4.3 Experiments and Analysis

4.3.1 Experiment Setup

Experiments are conducted on MovieLens[1] dataset. This stable benchmark dataset released in February of 2003, and it contains approximately one million ratings of 3,900 movies given by 6,040 users. To filter the noise data, users who input less than 50 number of ratings are removed from the dataset. There are no explicit links among the users, but the implicit links can be generated according to the item ratings. Moreover, in order to control the computing time, we vary the scale of the network sub-graphs as 500, 750 and 1000 in each experiment.

The node degree distributions of the sub-graphs scaled to sizes of 500, 750 and 1000 are presented in Figures 4.4, 4.5 and 4.6, respectively. All of these sub-graphs exhibit a power-law distribution pattern of the node degrees, as in most real networks [52].

4.3.2 Global Pheromone Distribution

As explained in Section 3.4, all of the artificial ants crawl across the social network and allocate pheromone after completing their tours, meanwhile the allocated pheromone continuously evaporates over time. The total amount of the outstanding pheromone in the social network is regarded as the global pheromone.

The global pheromone distributions of the 500, 750 and 1000-sized sub-graphs are plotted in Figures 4.7, 4.8, and 4.9, respectively. In all cases, the pheromone amount increases steadily towards a certain level, then oscillates around that level. This indicates a near-equilibrium stare between pheromone allocation and evaluation. At this point,

[1] http://grouplens.org/datasets/movielens/

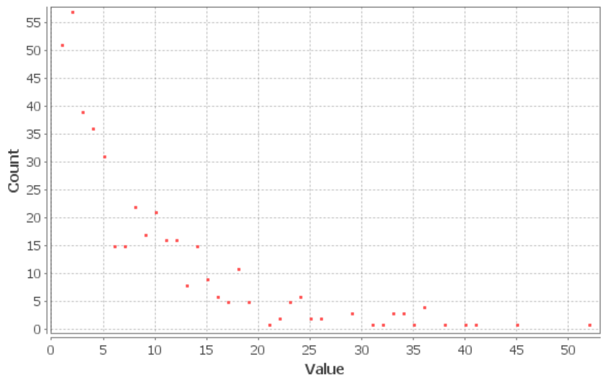

FIGURE 4.4: The Degree Distribution (size = 500)

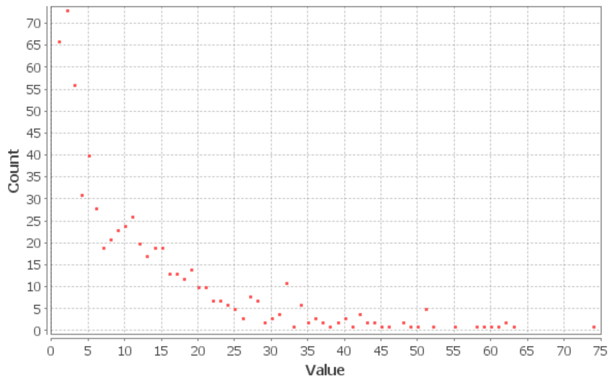

FIGURE 4.5: The Degree Distribution (size = 750)

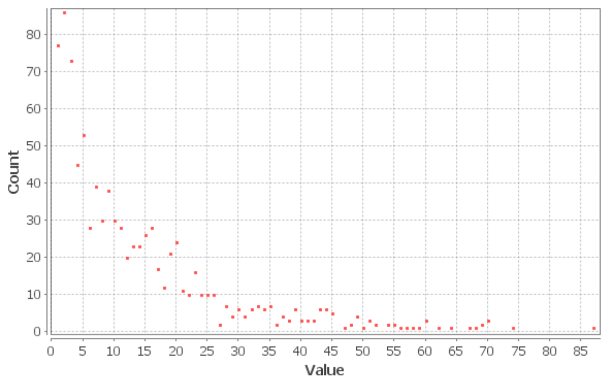

FIGURE 4.6: The Degree Distribution (size = 1000)

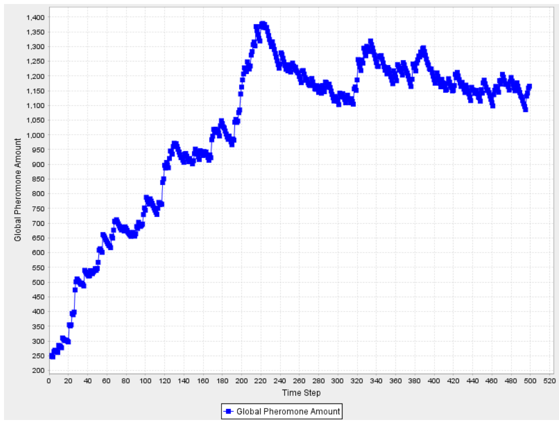

FIGURE 4.7: Global Pheromone Distribution (size=500)

the sequential pheromone ranking list changes only marginally, indicating that convergence is reached.

4.3.3 Experimental Results

We conducted two experiments by using the same social network in three different sizes (500, 750 and 1000). The first experiment aims to evaluate the influence effectiveness of the stigmergy-based algorithm, i.e., the total number of users activated by the seed set. While, the second experiment tends to evaluate the efficiency, i.e., the running time of seed selection. In both experiments, the stigmergy-based algorithm is competed against the greedy selection, degree-based selection and random selection.

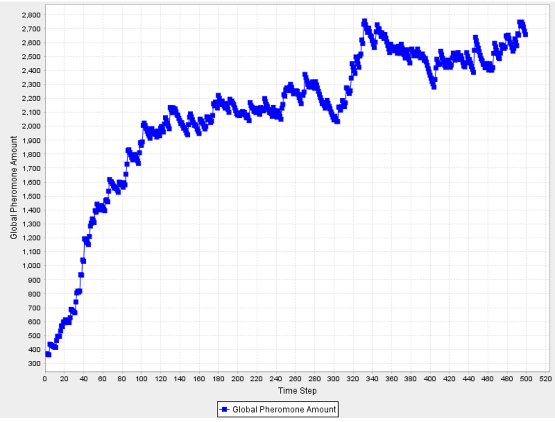

FIGURE 4.8: Global Pheromone Distribution (size=750)

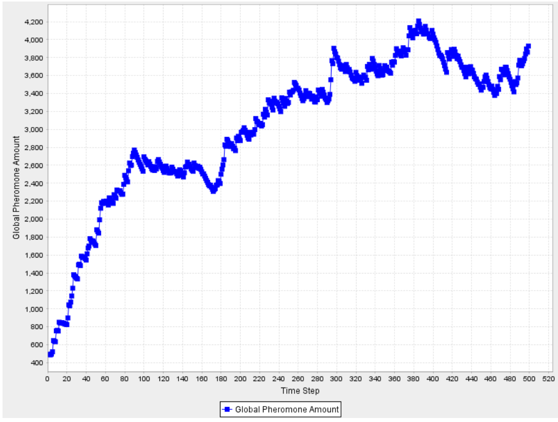

FIGURE 4.9: Global Pheromone Distribution (size=1000)

In the first experiment, seeds selected from the proposed model are input to the IC model. The influence effectiveness of the SIM model is compared with those of classic algorithms, the comparisons among the four algorithms in sub-networks of size 500, 750 and 1000 are presented in Figures 4.10, 4.11 and 4.12, respectively. The stigmergy-based algorithm performs better than both degree-based selection and random selection in all cases, and its performance is similarly to the greedy selection at the network size of 500. On larger sub-graphs, the influence effectiveness of stigmergy-based selection deteriorates slightly, but remains above that of the other algorithms.

The second experiment analyses the efficiency of the four seed selection algorithms by comparing their running time. In the stigmergy-based algorithm, the runtime includes the initiation and pheromone operations of the ants. The efficiencies of the other three algorithms are evaluated in the IC model. Figures 4.13, 4.14 and 4.15 compare the efficiencies of the four algorithms in sub-networks of sizes 500, 750 and 1000, respectively. Greedy selection is by far the most computationally expensive algorithm, and its running time is proportional to the size of the seed set. The efficiencies of random selection and degree-based selection are very similar. The stigmergy-based algorithm is only slightly more efficient than degree-based selection, but is much more efficient than the greedy selection, and the computational cost does not increase a lot with the expansion of network.

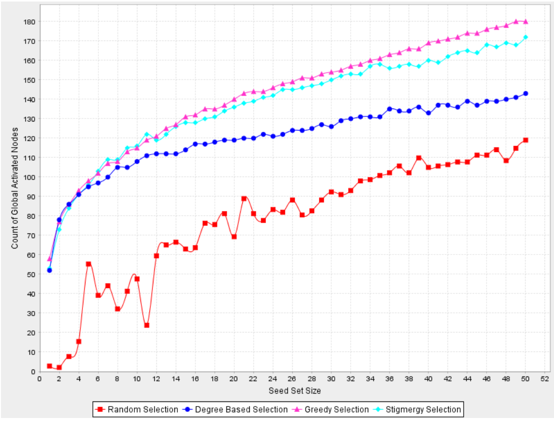

FIGURE 4.10: Influence Effectiveness Comparison (size=500)

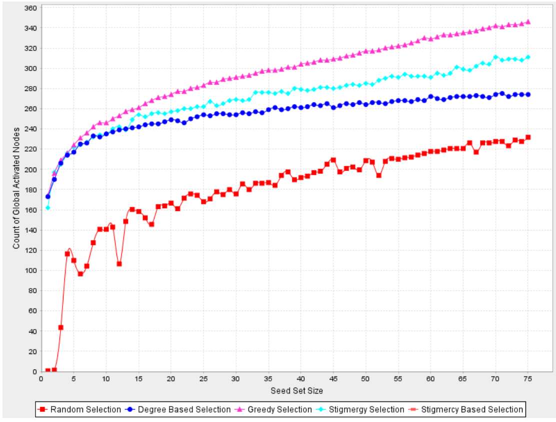

FIGURE 4.11: Influence Effectiveness Comparison (size=750)

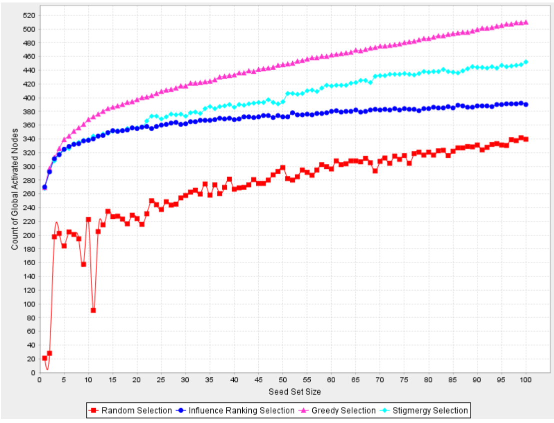

FIGURE 4.12: Influence Effectiveness Comparison (size=1000)

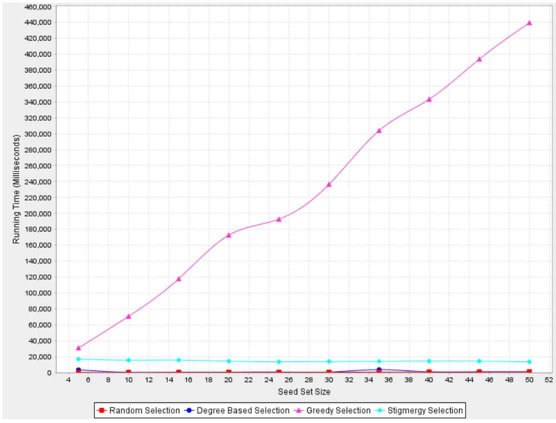

FIGURE 4.13: Efficiency Comparison (size=500)

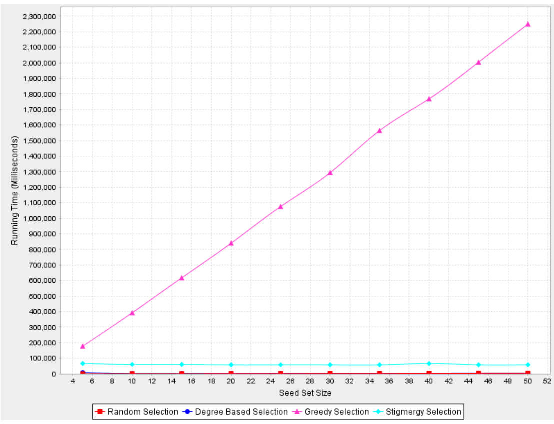

FIGURE 4.14: Efficiency Comparison (size=750)

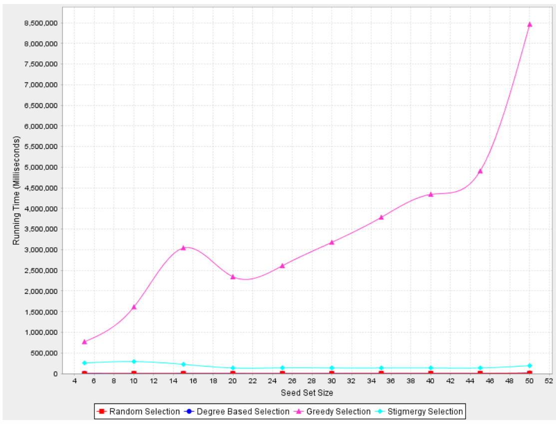

FIGURE 4.15: Efficiency Comparison (size=1000)

In summary, according to the experimental results, we can conclude that the stigmergy-based algorithm performs better than the traditional algorithms by considering both effectiveness and efficiency.

4.4 Summary of The Stigmergy-based Influence Maximisation (SIM) Model

In this chapter, we introduce a novel approach, i.e., stigmergy-based algorithm, that tackles the IM problem in a decentralised environment. In the meanwhile, the SIM model has been proposed and systematically elaborated, and its performance is experimentally evaluated in comparison experiments. Experimental results demonstrate that the SIM model performs better than the traditional seed selection approaches, including greedy selection, degree-based selection and random selection, by considering both effectiveness and efficiency. Moreover, the SIM model is applicable to large-scale networks and functions even without a global perspective.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

With the development of online marketing, understanding how the influence should be propagated over the widest possible range with finite resources is critically important. To this end, we must find practical solutions for the IM problem. However, the solution of IM problem is NP-hard. Approximation approaches that guarantee a solution are desirable alternatives. Provided that the influential users can be properly selected and processed, the influence spread can be regarded as successful. The primary motivation of this thesis is to explore appropriate approximation approaches to the IM problem which would obtain effective and efficient solutions. Based on the existing problems as aforementioned, IM is developed by exploiting two approaches, i.e., centralised and decentralised approaches, in this thesis.

Many existing researches related to IM are proposed based on classic centralised approaches, such as, the IC and LT models. In order to alleviate the inherent problems in classic influence diffusion models, an optimised centralised approach, i.e., the PTIC model, is proposed by considering not only UP but also TC and this model is developed on the foundation of the IC model. The UP is involved in the whole process of the PTIC model. In the PLM, user preferences are used to calculate the CPS (see Subsection 3.3.1). In the CPM, users with similar CPSs are partitioned into communities (as explained in Subsection 3.3.2). Moreover, the PTIC model calculates the influence propagation probabilities based on the UP and TC. The TC is used to ensure the users with similar preferences are interconnected, which facilitates the influence propagation. In this situation, users who have not only high node degree but also preferences for the promoted items will be selected as influential users in the online social network. As for the implementation of the PTIC model, we conduct two experiments, one considering the factor TC only, the other considering both of UP and TC. The PTIC model is competed against two existing approaches, i.e., trust-only and random approaches, within these two experiments. The experimental results confirm that the PTIC model outperforms the trust-only and random approaches.

Through the process of reviewing the existing literature and conducting experiments, we discover that centralised approaches are normally inefficient. Centralised approaches limit the stability and scalability of online social networks, especially when those networks are dynamic and large-scale. This is because all tasks in the network are completed by a central component in centralised approaches. Furthermore, centralised approaches also require complicated computations. In order to resolve the discovered problems of centralised approaches, we proposed a novel decentralised approach, called the SIM model, which adopts a stigmergy-based approach. Compared with centralised approaches, decentralised approaches are more effective and efficient since the tasks are distributed among the individuals. Moreover, task distribution also decreases computational complexity. The main idea of the SIM model is to simulate the influence propagation process as ants crawling across the network topology. The whole process of ants crawling are modelled in the SIM model, especially their key behaviours of ants, i.e., path selection, pheromone allocation, and seed selection. Path selection is used to decide the next node to approach when an ant faces multiple choices. Pheromone allocation aims to deposit a proper amount of pheromone on the nodes based on the heuristics, and is implemented when an ant explores a possible influence-diffusion path. Seed selection is to determine the seed candidates based on the pheromone rankings of nodes in the network. In addition, two experiments are implemented in the same network with three different sizes (500, 750 and 1000), which aims to evaluate the effectiveness and efficiency of the stigmergy-based approach. In these experiments, the SIM model is competed against several traditional seed selection algorithms, namely, the greedy selection, degree-based selection, and random selection algorithms. Based on the experimental results, the SIM model outperforms the other three approaches by taking not only effectiveness but also efficiency into consideration.

5.2 Future Work

Both of the proposed approaches could be developed in future works. As for the centralised approach, it could be implemented in other classic influence diffusion models, such as, LT model [18]. In addition, besides UP and TC, the model could account for positive and negative influence effects. Furthermore, the trust relationship between users could be regarded as directed rather than undirected. For example, if a user u_a trusts user u_b on a directed graph, user u_b will probably distrust user u_a. Regarding the decentralised approach, machine learning algorithms could be employed to improve the performance of the stigmergy-based algorithm in the IM problem. Meanwhile, we could consider a hybrid approach for developing a more practical model, for example, by combining the stigmergy-based approach with reinforcement. In addition, the digital pheromone can be exploited, which is able to be withdrawn/deposited by ants from their pheromone stores based on the importances of nodes in the online social network.

Bibliography

[1] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[2] Alexandra Marin and Barry Wellman. Social network analysis: An introduction. *The SAGE handbook of social network analysis*, pages 11–25, 2011.

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

[4] Sabbir Ahmed and CI Ezeife. Discovering influential nodes from trust network. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 121–128. ACM, 2013.

[5] Wei Chen, Yajun Wang, and Siyu 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.

[6] Jingyu Zhou, Yunlong Zhang, and Jia Cheng. Preference-based mining of top-k influential nodes in social networks. *Future Generation Computer Systems*, 31: 40–47, 2014.

[7] Michael Small, Lvlin Hou, and Linjun Zhang. Random complex networks. *National Science Review*, 1(3):357–367, 2014. doi: 10.1093/nsr/nwu021. URL http://nsr.oxfordjournals.org/content/1/3/357.abstract.

[8] Xiao Fan Wang and Guanrong Chen. Complex networks: small-world, scale-free and beyond. *Circuits and Systems Magazine, IEEE*, 3(1):6–20, 2003.

[9] D Price. Statistical studies of networks of scientific papers. In *Statistical Association Methods for Mechanized Documentation: Symposium Proceedings*, volume 269, page 187. US Government Printing Office, 1965.

[10] Xiao Fan Wang and Guanrong Chen. Pinning control of scale-free dynamical networks. *Physica A: Statistical Mechanics and its Applications*, 310(3):521–531, 2002.

[11] Fredrik Liljeros, Christofer R Edling, Luis A Nunes Amaral, H Eugene Stanley, and Yvonne rAberg. The web of human sexual contacts. *Nature*, 411(6840): 907–908, 2001.

[12] Yaneer Bar-Yam. *Dynamics of complex systems*, volume 213. Addison-Wesley Reading, MA, 1997.

[13] Behnam Hajian and Tony White. On the interaction of influence and trust in social networks. In *Proceedings of the 1st Workshop on Incentives and Trust in E-Commerce*, pages 63–75, 2012.

[14] Chakravanti Rajagopalachari Kothari. *Research methodology: Methods and techniques*. New Age International, 2004.

[15] Pedro Domingos and Matt 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.

[16] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.

[17] Everett M Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

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

[19] Su-Hua Chang. Efficient influence maximization in social network via community characteristics. 2011.

[20] Chi Wang, Wei Chen, and Yajun Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3):545–576, 2012.

[21] Bo Liu, Gao Cong, Dong Xu, and Yifeng Zeng. Time constrained influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 439–448. IEEE, 2012.

[22] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-based intelligent information and engineering systems*, pages 67–75. Springer, 2008.

[23] Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4):21, 2012.

[24] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1039–1048, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1. doi: 10.1145/1835804.1835935. URL http://doi.acm.org/10.1145/1835804.1835935.

[25] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM)*, pages 211–220. IEEE, 2011.

[26] Myungcheol Doo and Ling Liu. Extracting top-k most influential nodes by activity analysis. 2012.

[27] Wei Chen, Chi Wang, and Yajun 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.

[28] Masahiro Kimura and Kazumi Saito. *Tractable models for information diffusion in social networks*, pages 259–271. Springer, 2006.

[29] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie 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.

[30] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.

[31] Zhidan Feng, Xiaowei Xu, Nurcan Yuruk, and Thomas AJ Schweiger. *A novel similarity-based modularity function for graph partitioning*, pages 385–396. Springer, 2007.

[32] Jianbin Huang, Heli Sun, Jiawei Han, Hongbo Deng, Yizhou Sun, and Yaguang Liu. Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 219–228. ACM, 2010.

[33] Ying Pan, De-Hua Li, Jian-Guo Liu, and Jing-Zhang Liang. Detecting community structure in complex networks via node similarity. *Physica A: Statistical Mechanics and its Applications*, 389(14):2849–2857, 2010.

[34] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[35] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[36] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[37] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[38] Luca Donetti et al. Detecting network communities: a new systematic and efficient algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(10): P10012, 2004.

[39] Dinh Q Phung, Svetha Venkatesh, et al. Preference networks: Probabilistic models for recommendation systems. In *Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70*, pages 195–202. Australian Computer Society, Inc., 2007.

[40] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl. Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 127–134. ACM, 2002.

[41] Mohammad Ali Abbasi, Jiliang Tang, and Huan Liu. Trust-aware recommender systems. *Machine Learning book on computational trust, Chapman & Hall/CRC Press*, 2014.

[42] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.

[43] KP Wang, L Huang, CG Zhou, W Pang, et al. Traveling salesman problem. In *Conference on Machine Learning and Cybernejcs*, volume 3, pages 1583–1585, 2003.

[44] Marco Dorigo, Eric Bonabeau, and Guy Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16(8):851–871, 2000.

[45] Ahmed Mostafa, Minjie Zhang, and Quan Bai. Stigmergic modeling for web service composition and adaptation. In *PRICAI 2012: Trends in Artificial Intelligence*, pages 324–334. Springer, 2012.

[46] Atahar Mostafa, Minjie Zhang, and Quan Bai. Trustworthy stigmergic service composition and adaptation in decentralized environments. *Service Computing, IEEE*, 2014. doi: 10.1109/TSC.2014.2298873.

[47] Junji Takahashi, Ryo Kanamori, and Takao Ito. A preliminary study on anticipatory stigmergy for traffic management. In *Proceedings of the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pages 399–405. IEEE, 2012.

[48] Huiyuan Zhang, Dung T Nguyen, Huiling Zhang, and My T Thai. Least cost influence maximization across multiple social networks. 2015.

[49] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[50] Daniel Shiffman, Shannon Fry, and Zannah Marsh. *The nature of code*, chapter 7 Cellular Automata, pages 323–330. D. Shiffman, 2012.

[51] Eric Bonabeau. Editor's introduction: stigmergy. *Artificial Life*, 5(2):95–96, 1999.

[52] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world.* Cambridge University Press, 2010.