ABSTRACT

MALLICK, SHAILAJA. Spread of Opinion in a Social Network!. (Under the direction of Do Young Eun and Christopher Healey.)

When a new product enters a market already dominated by an existing product, will it survive and coexist with this dominant product? Most existing research has represented the coexistence of two competing products spreading or being adopted as overlaid graphs of social networks with a common set of individuals. However, for the survival of a weaker product on the same graph, it has been established that the stronger product dominates and eliminates the weaker competition. In this thesis, we take a step towards narrowing this gap so that even a new or weaker product may survive with a positive market share. Specifically, we attempt to identify a locally optimal set of users over time to form an influencer community that can be targeted by a company launching a new product under a given budget constraint. We model the system as a competing Susceptible-Infected-Susceptible (SIS) epidemic and employ perturbation techniques that identify budget parameters to produce a positive market share in a cost-efficient manner. Our simulations with real-world graph datasets and performance comparisons against standard centrality measures demonstrate that with our choice of target nodes, a new product can establish itself, rather than being dominated and eventually pushed out of the market under the same budget constraint.

Although SIS-based models determine users' opinions towards a product using exogenous model parameters, online forums or reviews are the most common real-world medium for users to express their opinions. This in turn helps other potential purchasers before making a buying decision. Important features of a product evaluated in reviews make it easier for a user to build an informed decision. We take advantage of online reviews posted in e-commerce websites to analyze their sentiment and visualize the results as a node–link graph highlighting how reviews help users form opinions. We construct a sentiment-weighted property list assigning 1, 0, or -1 to positive, neutral, or negative senti-

ment for each important product property discussed in the reviews. This labeled dataset acts as input to machine learning algorithms that can predict a user's preferences based only on property sentiment extracted from their reviews. These preferences are integrated into our SIS model to optimize which users to target and which product features to emphasize. To better represent our findings, we employ visualization to provide information about the structure of the users and their relationships. Our sentiment visualization uses node–link graphs to present users, their relationships, and the spread of influence through review sentiment about product features. This illustrates how potential purchasers form opinions on the basis of a product's features combined with influence from their neighbors.

Our novel contributions include: (1) perturbation to modify bi-SIS epidemic model parameters in a competitive product domain; (2) optimal budget allocation for advocate community creation; (3) sentiment analysis of review datasets to choose and weight important product aspects and properties for bi-SIS elements; and (4) a text-to-visual representation of the integrated bi-SIS/sentiment model showing how influence is spread between potential product adopters.

Spread of Opinion in a Social Network!

by
Shailaja Mallick

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2023

APPROVED BY:

Rada Chirkova	Khaled Harfoush
Do Young Eun Co-chair of Advisory Committee	Christopher Healey Co-chair of Advisory Committee

DEDICATION

To my husband, VT, for his endless support and belief in me.

BIOGRAPHY

Shailaja Mallick is from Odisha, India, and graduated with a Bachelor of Technology in Computer Science and Engineering from Veer Surendra Sai University of Technology (formerly University College of Engineering), Burla in 2011. She then joined Tata Consultancy Services, Bangalore as an Assistant Systems Engineer and then left the job as Systems Engineer to pursue a Masters degree in 2013. Following her Masters from Chalmers University of Technology, Sweden, she joined the Computer Science department at NC State University as a Ph.D. student in 2017. Her research interests include applied machine learning, Natural Language Processing, epidemiological models and network science, and upon graduation, she intends to pursue a career in industry.

ACKNOWLEDGEMENTS

I would like to thank Dr. Eun for all his support and guidance over the past few years. Thank you for introducing me to the diverse field of epidemiology within the context of Computer Science; I learnt a lot and grew as a better researcher. I would also like to thank Dr. Healey for believing in me, and for his constant unconditional support and encouragement throughout this journey. His guidance and help has been invaluable towards the completion of this work. I am very fortunate to have had the opportunity to work with him. In addition, I would also like to thank Dr. Chirkova and Dr. Harfoush for agreeing to be on my committee, taking out time to listen to my research progress, and offering helpful feedback.

I have also been fortunate to be able to work under Dr. Battestilli, both as a TA and Mentored Teaching Fellow for two semesters. I gained valuable experience of better understanding the teaching methodologies and designing course structure. I extend special thanks to Dr. Rouskas for being considerate and supporting me throughout my PhD journey.

This journey would be incomplete without all the unconditional love and support from my Raleigh friends. Thank you Vijay bhai, Vineeta didi, Avni, Seema, Kamal bhai, Arnika, Manas bhai, Sony, and Soumya bhai for all the fun times we have spent together. I could not have asked for better friends than you all. I would like to thank Akanxa didi and Sandeep bhai for all the movie night dinners we have had. The time spent with you all over the weekends was a stress buster. Thank you Xi and Jason, I am going to miss our weekly Sunday badminton sessions and the poker nights.

Finally, I will always be thankful to my family. My parents and my sister for their endless love, support, and in general being there for me in this process. Thank you Aja, Ai, Ma, and Nida for the unconditional love, belief, wishes, and presence in this journey.

This PhD journey would not have been possible without the encouragement, love, belief, support (both during the highs and lows) of my loving husband, Manish. His joyful nature and perennial smile have made my bad days good and kept me going through this process.

Words cannot express how grateful and thankful I am. Last but not the least, the pandemic would not have been easier without the cuddles, smiles, licks, play-times with my loving and dearest MOMO. You have been a constant source of happiness for the last three years and will be for the years to come.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

In recent years the emergence and wide-spread use of online social network platforms such as Facebook and Twitter have become the paradigm for spreading rumors or launching campaigns to gain attention. Product adoption, marketing campaigns, and the spread of rumor/misinformation or even malware has led to an intense competition between consumption and propagation. These social networks play a fundamental role as a medium for the spread of such events, ideas, or influence among its users. In such social networks, based on friend of friend relationships information such as rumors, ideas, or products evolve and spread very quickly. For example, the spread of fake news over social networks like Facebook, Twitter and WhatsApp have been documented heavily with respect to recent elections. The way an individual forms an opinion is an interplay between the user's behavior/opinion towards a product and the influence of their friends in the network. In order to understand how "word-of-mouth" plays an important role in the extent to which ideas/information are adopted, it is vital to understand how adoption dynamics propagate within an underlying social network.

In this thesis, we investigate the spread of *properties* within a network of *entities* over time. This general description represents many real-world domains. My focus in this thesis is the spread of *influence and opinion* between *individuals* considering product purchases. We model spread based on research studying the diffusion of a disease in epidemiology. A collection of related models have been proposed based on susceptibility, infection, and recovery. We use these models as a foundation for investigating how individuals choose products based on an evaluation of the importance and capabilities of a product's properties. Companies can use this knowledge to enter new markets by identifying and growing a small group of *influencers* who spread their knowledge about a product and its properties to their neighbors. This can allow companies to successfully compete against existing, dominant products in a cost effective manner.

In Chapter 2 we focus on modeling product competition, especially when a company is launching a new product alongside an already existing dominant product in the market. To define the spread of information within communities or a social network, we turn to the use of graphs representing epidemic disease outbreak and its spread. Some of the standard epidemic models for single-virus spread in a network are Susceptible-Infected (SI), Susceptible-Infected-Susceptible (SIS), and Susceptible-Infected-Recovered (SIR). These models have also been extended to analyze multiple viruses spread in a network. The model that aptly captures the dynamics of competitive viruses is a variant of SIS is *bi-SIS* [Ker27; Pra12b] where an individual oscillates between infected and recovered. *bi-SIS* is the underlying model of our work. In this part of the thesis, we propose techniques to quantify and improve the market share of the new/weaker product when it competes with an existing dominant product in the market, where the underlying network follows the bi-SIS spreading mechanism. Under limited budget constraints, a new company launching its product would aim for survival, and the best strategy in this situation is to target certain users via advertisement/promotional offers to form a small support users group (community) who, with their positive reviews, can spread motivating recommendations about the new

product. The goal behind promoting such communities is to increase the rate of influence so as to maximize the chances of the product's survival. Specifically, a new company would initially be resource-constrained, and wants to identify an optimal set of individuals to form a support user group to maximize its (positive) market share as much as possible.

Various mathematical models, such as variants of the SIS epidemic model [Omi09; Liu19], opinion dynamics-based models such as De-Groot, Hegselmann-Krause, Friedkin-Johnsen, and others [Pro17; Pro18], or the Linear Threshold (LT)/Information Cascade (IC) model [Kem03; Che10] have been proposed to capture the dynamics of competing products. Most of the work around these models focuses on exogenous model parameters that determine how a user forms an opinion (in terms of probability) and is influenced by its neighbors towards buying any of the competing products. However, this is not exclusively the case. Users' purchasing decisions are influenced not only by their neighbors, but also by comparing product features on e-commerce websites.

In Chapter 3, we consider a real-world dataset of smartphone reviews to understand and analyze what factors in addition to word-of-mouth influence a user to form an opinion towards product adoption. We utilize machine learning algorithms, and natural language processing individuals' to understand people's buying behavior from product reviews posted on e-commerce websites. With the available reviews dataset, the aim is to classify the reviews based on each of the properties discussed in a review into positive, negative, or neutral sentiments. This classification of reviews based on products' features will help potential future buyers make an informed decision by evaluating such positive and negative feedback constructively. In this part of the thesis, we study the extraction of products' features, especially from reviews of two competing smartphone brands, and perform sentiment analysis to generate a three-valued property list ($-1, 0, 1$ for negative, neutral, or positive sentiment) of each of the features for each review. We aim to determine how well different supervised machine learning models predict the review rating given by a user about a product based on the user's property sentiment vector. This property list uses the

star rating of a review as a proxy for the overall sentiment about the product. As users have either a positive, negative or neutral sentiment towards a product's feature, we also use visualization of users as a node-link graph to highlight which features have higher weights for given users, along with their opinion towards each of the products. Through this visualization dashboard, we hope to provide an intuitive medium to potential customers so that they can navigate through the reviews and form an opinion about a product.

Finally, in Chapter 4, we focus on the integration of predicted star-ratings into the original bi-SIS model. The goal is to better understand how users form opinions towards a product that is then propagated among their neighbors. As the current bi-SIS model assumes constant edge weight, by incorporating the star-rating into the underlying model, we can better reflect how individuals actually form an opinion and how this opinion spreads as influence among neighbors.

CHAPTER 2

Targeted Advertisement to Communities in Social Networks

2.1 Introduction

Motivation: In this competitive world of rapid technological advancements, one of the key strategic decisions that a company launching a new product has to make is how to position itself [Coo87; MW94]. It can carve out a niche by differentiating itself from its competitor, either by its product design or brand image [Fuc10; Jal14]. When a new product enters the market, more often than not, there is another existing dominant product. For instance, Nintendo dominated the video game console market in the US for almost a decade until the introduction of the PlayStation (PS) [Gal02]. Over the next few years, PS was able to establish and eventually dominate the video game console market, forcing Nintendo to focus on hand-held consoles. Subsequently, similar to the PS-Nintendo dynamics, Xbox initially entered a PS-dominated market, and currently, both coexist. Figure 2.1 shows that before the introduction of Xbox 360, PS 2 already dominated the US market and since the emergence of Xbox, it took some time to mark its presence in the market against its

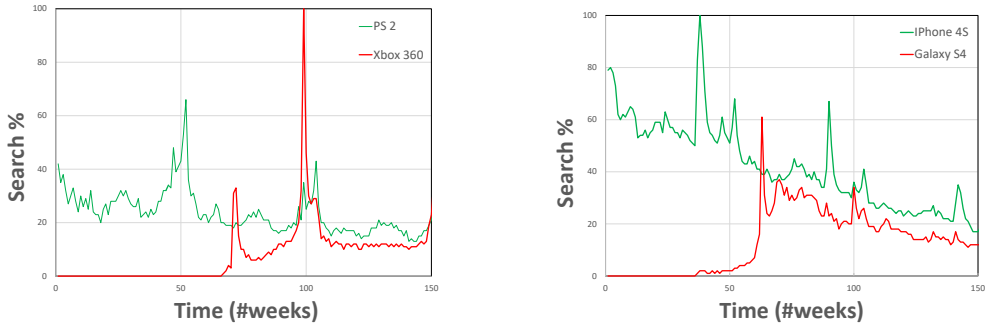


Figure 2.1 Real web-search percentage over a time period (in weeks) showing the introduction of a new product (Galaxy S4 and Xbox 360) alongside an existing product (iPhone 4S and PlayStation 2).

competitor (similar for iPhone 4S and Galaxy S4)[1]. In such situations, marketing techniques such as advertisements serve as a medium of communication and play a key role for the firm to achieve this differentiation. There has been evidence that products with no mass advertisements have gained popularity over time as the companies have focused on targeting advertisements to niche communities [Dal94]. The effectiveness of targeting a small portion of users (communities) who can further spread and influence their neighborhood network via word-of-mouth, rather than focusing on the entire market, has proven beneficial and is also recognized by businesses in the market [Kot90; Yan06].

When it comes to modeling product competition where individuals are the adopters, the interactions between them can be represented as edges of a graph, which are linked via socially meaningful relationships like friendship or information exchange. Such social networks have become the paradigm for studying the spread of competing products and their adoption among individuals [Kal85]. Although here we stress "product" adoption and its influence among neighbors, we can draw a similar analogy within different contexts, for example, having an opinion (or preference) towards a political party/idea and how targeted advertisement influences this opinion.

Our Contributions: In this paper, we take a step forward and propose techniques to

[1]Real-data collected from Google Trends for the United States region which provides "insights into search pattern"(`https://trends.google.com/trends/?geo=US`).

quantify and improve the market share of the new/weaker product when it competes with an existing dominant product. Under limited budget constraints, this new company launching its product would aim for survival, and the best strategy in this situation is to target certain users via advertisement/promotional offers to form a small support users' group (community) who, with their positive reviews, can spread motivating information about this new product. The idea behind promoting such communities (additional edges among the users of the graph) is to increase the rate of influence so as to maximize the chance of its survival. Specifically, the new company that is initially resource-constrained wants to identify an optimal set of nodes to form such communities to maximize its (positive) market share.

When it comes to quantifying the resulting market share on a general graph, there is no known closed-form solution for the positive fixed point (market share) even for a single-SIS model. The situation is further complicated when competition comes into play (bi-SIS). To tackle this challenge, we employ a perturbation approach to optimally assign resources under a specific budget structure based on model parameters and approximate the resulting market share of the original losing/weaker product. Our proposed budget structure also accounts for a minimum budget below which the company, despite spending positive amounts on advertising and expecting a non-zero return on investment, still obtains zero market share and thus cannot survive. By leveraging our theoretical findings, we develop an algorithm–a modified version of "Iterated Local Search" [Lou03]–that returns a locally optimal choice of nodes at each iteration, under a total budget constraint with heterogeneous costs for different users. We also demonstrate the efficacy of our algorithm by comparing it against different benchmarks such as the choice of top-k nodes via centrality measures over real network topology with various cost constraints for the users in the community.

2.2 Background & Related Work

In this section, we provide a brief background and discuss the existing literature on modeling and analysis of SIS epidemic models (both single and multi-virus). We describe recent results with respect to population ecology, epidemiology, and epidemic thresholds. Next, we discuss our modeling choice of the bi-SIS epidemic model in the context of competing products. We present a comparison of the bi-SIS epidemic model with other models such as opinion dynamics based models or LT/IC models based on how each approach captures products' competition in a network.

The classic epidemic model draws its inspiration from an important principle in theoretical biology known as *competitive exclusion*. The principle states that no two species can forever occupy the same ecological niche. Mathematical modeling of the evolution of the total number of healthy and infected individuals, referred to as *population dynamics*, was first undertaken by Lotka [Lot56] and Volterra [Vol28], independent of each other. Their work was seminal and is referred to as the Lotka-Volterra model. It suggests that indefinite coexistence of two or more species (such as a predator and prey) limited by the same resource is impossible. The Lotka-Volterra model has been used not only in biological studies but also in other fields such as physics and computer science. It has been extended to analyze the competition between species when there are more than one resource, and to understand the science behind extinction and persistence of a disease.

Apart from ecology, the Lotka-Voltera model has also been used to analyze how diseases, such as flu or sexually transmitted diseases, spread and reach an endemic/pandemic state [CC99; CC96]. Standard epidemic models for single-virus spread in a network include SIS (flu-like, with no immunity), SIR (with complete immunity for a lifetime) and Susceptible-Infected-Recover-Susceptible (SIRS, with temporary immunity). Many of the well-studied epidemiological models assume that the network is fully connected and every person has the same probability of contact with one another. This class of epidemiological

model is called a *homogeneous model* [And92; Bai75]. Kephart and White [Kep92] adopted a modified homogeneous model, which represents the communication pattern among individuals on directed graphs. Wang *et al.* [Wan03] and Ganesh *et al.* [Gan05] generalised Kephart and White's model without assuming homogeneous connectivity or any particular graph topology. They found that for single-virus spread under the SIS epidemic model, the epidemic threshold is determined by the leading eigenvalue of the adjacency matrix of an arbitrary graph. However, their results do not hold in general as was shown using approximation. Van Meigham *et al.* [VM09] generalised Wang and Ganesh's results using mean field theory and showed that for any arbitrary graph, the epidemic threshold is equal to the largest eigenvalue of the adjacency matrix. Prakash *et al.* [Pra12a] further showed that for all propagation models, the only parameters that determine the epidemic threshold are the leading eigenvalue and a model-dependent constant. All of these models are based on Markov models that are simplified using mean-field approximation techniques to handle large network sizes. To eliminate the limitation of using Markov models on large networks, Wang *et al.* [Wan16] proposed a discrete-time absorbing Markov process to characterize virus propagation in any topology. In their work, they have derived bounds on the virus extinction rates by using a transition matrix of the Markov process, which is validated by their simulation results.

The vast majority of papers are based on a single-virus, despite the fact that many real systems exhibit multiple viruses propagating in a network. Prakash *et al.* [Pra12b] in their work have shown what happens when two viruses propagate in a network and how the dynamics of the SIS epidemic model change. In a two virus competition scenario, a node can be Susceptible or Infected with virus A or virus B. Prakash focused on competition between two types of viruses spreading on the same graph, with the assumption that an individual cannot be infected with two viruses at the same time. The theoretical and numerical results indicate that at a steady state (or at the end of a sufficiently large time horizon), the stronger virus wins ("winner takes all") and completely eliminates the weaker virus.

Beutel *et al.* [Beu12] extended the work of Prakash to consider the case where two viruses in a fully connected network provide partial immunity against each other. Specifically, they added a state where a node can be simultaneously infected with virus A and virus B. They showed that there is a phase transition, which implies that below a critical value the stronger virus dominates and wins over the weaker virus. Above this critical value, the weaker virus survives and there is coexistence. The authors characterise the set of equilibria but do not provide any conditions for convergence to these equilibria. In [Xu11], Xu *et al.* also studied a similar model on a non-trivial graph structure. Their model is probabilistic and not based on differential equations (ODE), distinguishing it from the papers mentioned above. Paré *et al.* [Par18] proposed a similar model but on a general graph structure and provided analysis of different equilibrium conditions, specifically when both the viruses die out to produce a healthy equilibrium condition.

2.3 Choice of bi-SIS

When two products compete with each other on a network, the eventual outcomes/conditions can be one of the following: absolute dominance, where one of the products dominates by wiping out its competitor and is adopted widely; die-out, where none of the products are adopted; or coexistence, where both of the products are adopted within the network [Pra12b; Beu12; Sah14; Wei16]. Various mathematical models, such opinion dynamics-based models (i.e., analysis of how opinions form/spread in a population) proposed by De-Groot, Hegselmann-Krause, and Friedkin-Johnsen, variants of epidemic models [Omi09; Liu19], and others [Pro17; Pro18] have been proposed to capture the dynamics of competing products. Most of the work around classic opinion dynamics-based models focus on showing eventual consensus (everyone agrees to one of the opinions/rumors eventually), which has also been extended to capture the scenario of coexistence of two opinions (or clusters of opposing opinions). These works have shown coexistence by either relying on

external factors in the form of stubborn agents (initially or currently forming bias towards an opinion), zealots (who never alter their initial stance) or via the inclusion of confidence bounds beyond which individuals do not interact [Gar08; Yan12; Dan13; Mob15; Ame17; Muk20; Ana20; Ngu20]. The addition of stubborn agents already guarantees that at least one of the opinions (the one followed by the stubborn agents) is always adopted and hence results in clusters of opinions. With the inclusion of these additional factors in the model setups, eventual consensus or complete wipe-out is infeasible.

The propagation of information, influence maximization, and idea or opinion/rumor diffusion in a network has also been studied extensively via Linear Threshold (LT)/ Information Cascade (IC) models [Kem03; Che10]. In these models, individuals can either be in an active (i.e., influenced) or an inactive (i.e., free of influence) state. Here, the overarching assumption is that individuals either never change states or can only switch between active or inactive states randomly [Gra78; Pat10], with all nodes (users) becoming activated eventually. This is not always true in the context of products/opinions, as people tend to change their preferences/opinions/views from time to time.

The Susceptible-Infected-Susceptible (SIS) epidemic model has been used to study the spread of viruses in a network of populations [Laj76a; VM09]. However, the *bi-SIS* model, a variant of the SIS model (single virus), has been used to characterize the spreading dynamics of two competing products/opinions in a network [Pra12b; Sah14]. In such a model, via word-of-mouth or by being influenced by their neighbors, users adopt one of the two competing products (get infected) and gradually recover to become susceptible to infection again. Unlike the opinion dynamics-based model or LT/IC model, the bi-SIS model does not necessitate that the individuals have an opinion, positive or negative, ex-ante; in contrast, it allows individuals to switch states from being susceptible (inactive) to infected (active) and vice-versa. It has proved to be effective in capturing eventual consensus or coexistence in any general arbitrary graph, including special graphs such as complete or random [Wan03; Li12; Pra12b; San15; Yan17]. It can be seen that the bi-SIS model aptly

captures competitiveness without the need for any prior assumptions or dependence on a specific network structure. Thus, the bi-SIS model allows for all three possible state spaces while providing conditions for each of the outcomes i.e., die-out, absolute dominance, and coexistence.

Although the bi-SIS model is effective in capturing all possible outcomes, most of the existing work focuses only on stability analysis of equilibrium points or identifying which of the three possible outcomes a system converges to [VM09; Li12; Pra12b; Sah14; Dos21]. Furthermore, none of these works explicitly quantify the magnitude of existence; for example, the market share of either one or both of the competing products, especially in a general graph setting. Note that with respect to opinion dynamics, for example, the simple linear French-De-Groot model with stubborn agents [FJ56; Pro17] has shown quantifying results on a general graph. However, as mentioned earlier, opinion dynamics-based models do not capture all three possible states and thus pose a limitation when it comes to the analysis of the market share of competing products.

2.4 Network-Based bi-SIS – A Primer

Basic Notations: Let $\mathscr{G}(\mathscr{N},\mathscr{E})$ denote a general, undirected, connected graph with a set of nodes $\mathscr{N}=\{n_1,n_2,\ldots,n_N\}$ and set of edges $\mathscr{E}$, represented by $n_i,n_j\in\mathscr{N}$, and $e_{i,j}\in\mathscr{E}$, respectively. $\mathbf{A}\triangleq[a_{ij}]$ is the symmetric adjacency matrix of the graph $\mathscr{G}$, with $a_{ij}=1$ if $e_{i,j}\in\mathscr{E}$ and zero otherwise. We denote vectors $\mathbf{v}\in\mathbb{R}^N$ and matrices $\mathbf{M}\in\mathbb{R}^{N\times N}$ as bold lowercase and uppercase letters, respectively. For a non-negative matrix $\mathbf{M}$, we write $\lambda=\lambda(\mathbf{M})>0$ for its largest eigenvalue (spectral radius) [Mey00]. We use $\mathbf{S}_{\mathbf{v}}\triangleq\text{diag}(\mathbf{1}-\mathbf{v})$ to represent the diagonal matrix with the i^{th} diagonal entry being $(1-v_i)$. Let $\mathbf{1}$ ($\mathbf{0}$) be an N-dimensional vector of all ones (zeros). We write $\mathbf{x}\gg\mathbf{y}$ to indicate $x_i>y_i\ \forall\ i$.

Consider the system dynamics under a single-SIS setting where a susceptible node becomes infected with rate $\beta>0$ from *each* of its infected neighbors. Let $p_i(t)$ represent

the probability that node $n_i \in \mathcal{N}$ is infected at time $t \geq 0$. Once infected, the node recovers at a rate $\delta > 0$, and again becomes susceptible to infection. The system dynamics are then characterized by the system of ODEs i.e., for each $n_i \in \mathcal{N}$,

$$\frac{d\,p_i(t)}{d\,t} = \beta(1-p_i(t))\sum_{j\in\mathcal{N}} a_{ij}\,p_j(t) - \delta\,p_i(t) \tag{2.1}$$

Earlier works [Laj76b; VM09] in single-SIS have proved that when the effective infection rate $\tau \triangleq \beta/\delta$ is larger than the critical threshold $\tau^* = 1/\lambda(A)$, the virus survives ($\mathbf{x}^* \gg \mathbf{0}$) and infects a non-zero portion of the population; otherwise it dies-out ($\mathbf{x}^* = \mathbf{0}$) where $\mathbf{x}^*$ corresponds to the globally asymptotically stable fixed point of (2.1).

The system dynamics change accordingly when two viruses or products compete on *overlaid* graphs. Under such a bi-SIS setting, at any given time $t \geq 0$, a node infected with product $\mathbf{P}_1$ or product $\mathbf{P}_2$ infects its susceptible neighbors at a rate of β_1 or β_2 and they recover with a rate of δ_1 or δ_2, respectively. The effective infection rate then becomes $\tau_1 = \beta_1/\delta_1$ or $\tau_2 = \beta_2/\delta_2$ for the two products. Let $x_i(t)$ and $y_i(t)$ represent the probabilities that node $n_i \in \mathcal{N}$ adopts $\mathbf{P}_1$ or $\mathbf{P}_2$ at time $t \geq 0$. Note that a node cannot adopt both products at the same time. The system dynamics can then be written as the following *bi-SIS* model

$$\begin{aligned}
\frac{d\,x_i(t)}{d\,t} &= \beta_1\big(1-x_i(t)-y_i(t)\big)\sum_{j\in\mathcal{N}} a_{ij}\,x_j(t) - \delta_1 x_i(t) \\
\frac{d\,y_i(t)}{d\,t} &= \beta_2\big(1-x_i(t)-y_i(t)\big)\sum_{j\in\mathcal{N}} b_{ij}\,y_j(t) - \delta_2 y_i(t),
\end{aligned} \tag{2.2}$$

where the adjacency matrices $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$ corresponds to $\mathcal{G}_1(\mathcal{N}, \mathcal{E}_1)$ and $\mathcal{G}_2(\mathcal{N}, \mathcal{E}_2)$, respectively. The state transition diagram as seen from a node in the network is shown in Figure 2.2.

Let $\mathbf{x}^*, \mathbf{y}^* \gg \mathbf{0}$ be all the positive fixed points corresponding to single-SIS, that is, without any competition. The bi-SIS system (2.2) converges to the following types of equilibrium points $\forall\, n_i \in \mathcal{N}$ [Sah14; Dos21]:

(i) None of the products are adopted (virus-free equilibrium) when $\tau_1\lambda(A) \leq 1$ and

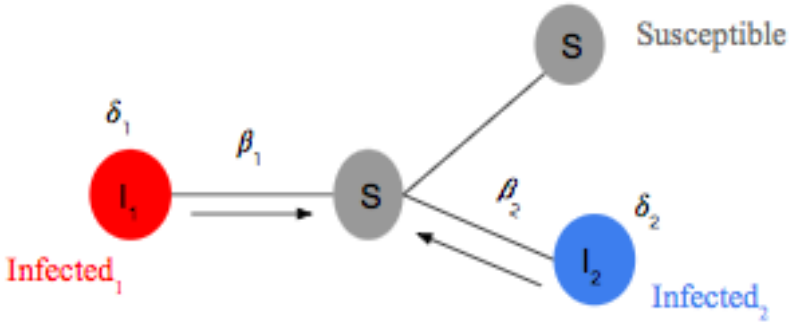


Figure 2.2 State Diagram for a node in the graph under SI_1I_2S model. The node is in S state if it is not affected by the competition (does not own a product). It is in state I_1 if it owns $\mathbf{P}_1$ and in state I_2 if it owns $\mathbf{P}_2$. The transition from S to I_1 or I_2 happens depending on the neighbors of each node.

$\tau_2\lambda(B)\leq 1$, and the system converges to fixed point $(\mathbf{0},\mathbf{0})$[2].

(ii) Only one of the products is adopted (single-virus equilibria) when $\tau_1\lambda(\mathbf{S}_{\mathbf{y}^*}A)>1$ and $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}B)\leq 1$, and the system globally converges to fixed point $(\mathbf{x}^*,\mathbf{0})$. Similarly, when $\tau_1\lambda(\mathbf{S}_{\mathbf{y}^*}A)\leq 1$ and $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}B)>1$, the system globally converges to $(\mathbf{0},\mathbf{y}^*)$.

(iii) Both products survive (coexistence equilibria) when $\tau_1\lambda(\mathbf{S}_{\mathbf{y}^*}A)>1$ and $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}B)>1$, and the system globally converges to $(\mathbf{x},\mathbf{y})\gg(\mathbf{0},\mathbf{0})$.[3]

2.5 System Model & Problem Setup

We begin with an examination of the problem of how to target individual users and recruit them to participate in communities or forums, with the goal of maximizing the long term market share of a new product. In our context, a community or online discussion forum provides a platform where a group of interacting users actively exchange information about common interests. A pertinent example can be a subreddit—a community in Red-

[2]When $\tau_1\lambda(A)>1$ and $\tau_2\lambda(B)\leq 1$ (or vice versa), virus 2 (or virus 1) dies out independent of the competition and the system reduces to the single-virus case. The conditions (ii) and (iii) are the only non-trivial cases that are satisfied when $\tau_1\lambda(A)>1$ and $\tau_2\lambda(B)>1$ (both viruses compete for survival)

[3]Here, $\mathbf{x},\mathbf{y}$ correspond to coexistence fixed points.

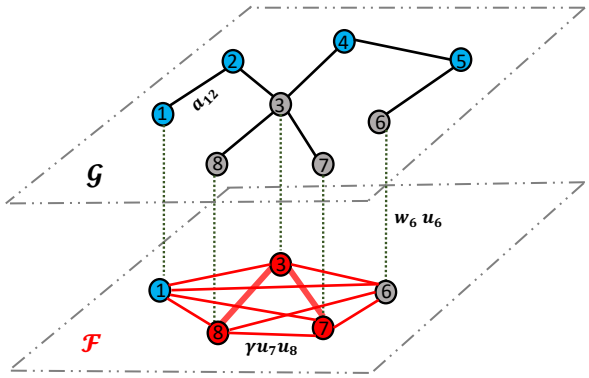


Figure 2.3 Illustration of users forming *community* (Nodes 1,3,6–8 in layer $\mathscr{F}$). The blue (red) nodes denote users adopting Product 1 (Product 2) at any given instance, and gray nodes are the users who does not own any of the products.

dit [Red09a]—which is dedicated to discussions around specific topics such as smartphones (`r/Smartphones`) [Red09b] or games (`r/gaming`) [Red07]. We will focus on establishing one such community C_{new}, and assume that any interactions or discussions between users in C_{new} are solely about the company's products or offerings, and not about its competitor's products.

Suppose that a user the company targets chooses to engage in C_{new} discussions with a certain probability. We can safely assume that the more aggressive the company is with its advertising efforts targeted towards the user, the higher is the probability of the user participating in C_{new}. In real life, these targeted advertisements take various forms such as product placement, early-bird discounts, or promotional offers, and cost a certain amount. This will help us in determining the modelling aspects of the cost structure for our problem. Before proceeding with the details, we discuss the dynamic system of ordinary differential equations (ODEs) for both single and bi-SIS.

Let $p(u_i)$ denote the probability with which any user $u_i \in \mathscr{N}$ actively engages in such a forum. We will assume that all users participating in the community are able to communicate with one another, and $\mathbf{P}_2$'s influence will spread across this community with rate $\gamma\beta_2$ where γ is a factor capturing the rate of influence relative to β_2. To clearly visualize this creation of a community, we draw it as a separate layer in Figure 2.3. The top layer $\mathscr{G}$

denotes the original graph where the mutual follower relationships are defined according to adjacency matrix $\mathbf{A}$ and layer $\mathscr{F}$ denotes the newly created community/forum in which a user $u_i \in \mathscr{N}$ participates with probability $p(u_i)$. The spread of influence within this community is akin to adding an edge of weight γ for any two users u_i and u_j *known* to be actively participating in the community. If the two users were already connected prior to being a part of the community, that is, $a_{ij} = 1$ (a_{12} in Figure 2.3), then γ is simply an additional reinforcement of their connection as a consequence of interacting through multiple media. The thick red lines in Figure 2.3 highlight this reinforcement whereas the black lines indicate the original interaction links. The probability with which both u_i and u_j engage in the forum thus becomes $p(u_i)p(u_j)$ ($p(u_7)p(u_8)$ in Figure 2.3), and the total weight of edges connecting them is, on average, given by $a_{ij} + \gamma p(u_i)p(u_j)$.

From the mean field interpretation of the bi-SIS model (2.2), $\beta_2(a_{ij} + \gamma p(u_i)p(u_j))$ is the overall expected rate at which the infected (already owns $\mathbf{P}_2$) node n_j causes the susceptible (owns neither product) node n_i to adopt $\mathbf{P}_2$. By substituting $b_{ij} = a_{ij} + \gamma p(u_i)p(u_j)$ in (2.2), and interpreting $x_i(t)$ and $y_i(t)$ as the probabilities with which node n_i owns $\mathbf{P}_1$ and $\mathbf{P}_2$ at time $t \geq 0$, we can then write our system dynamics as

$$\begin{aligned} \dot{x}_i(t) &= \beta_1(1 - x_i(t) - y_i(t)) \sum_{j \in \mathscr{N}} a_{ij} x_j(t) - \delta_1 x_i(t) \\ \dot{y}_i(t) &= \beta_2(1 - x_i(t) - y_i(t)) \sum_{j \in \mathscr{N}} [a_{ij} + \gamma p(u_i)p(u_j)] y_j(t) - \delta_2 y_i(t). \end{aligned} \tag{2.3}$$

Using $\mathbf{u} = [p(u_i)]_{u_i \in \mathscr{N}}$ to denote the vector of probabilities with which users participate in C_{new}, we can write the above system in a matrix-vector form as follows:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \beta_1 \text{diag}(\mathbf{1} - \mathbf{x}(t) - \mathbf{y}(t)) \mathbf{A}\mathbf{x}(t) - \delta_1 \mathbf{x}(t) \\ \dot{\mathbf{y}}(t) &= \beta_2 \text{diag}(\mathbf{1} - \mathbf{x}(t) - \mathbf{y}(t)) \left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right] \mathbf{y}(t) - \delta_2 \mathbf{y}(t). \end{aligned} \tag{2.4}$$

As can be seen in (2.4), the extra edge weights added to the original adjacency matrix $\mathbf{A}$ as a result of creating communities in the manner described earlier are in the form of a

rank one update given by $\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T$. We will use this special form of the update to aid our analysis. Our propagation model (2.4) yields the equilibrium equations

$$\mathbf{x} = \tau_1 \text{diag}(\mathbf{1}-\mathbf{x}-\mathbf{y})\mathbf{A}\mathbf{x} \tag{2.5}$$

$$\mathbf{y} = \tau_2 \text{diag}(\mathbf{1}-\mathbf{x}-\mathbf{y})\left[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T\right]\mathbf{y}, \tag{2.6}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the fixed points of (2.4) for $\mathbf{P}_1$ and $\mathbf{P}_2$, respectively, and $\tau_i = \beta_i/\delta_i$ for $i = 1,2,\ldots,N$ as defined in Section 2.4.

We next present the problem of maximizing the market share $\bar{y} = \frac{1}{N}\sum_{i\in\mathcal{N}} y_i$ (where $\mathbf{y} = [y_i]$ is the fixed point of (2.4) for $\mathbf{P}_2$) under a limited budget constraint, say C by an optimization problem.

$$(\mathcal{O}) \qquad \begin{aligned} \underset{\gamma,\mathbf{u}}{\text{maximize}} \quad & \bar{y} = \frac{1}{N}\sum_{i\in\mathcal{N}} y_i \\ \text{subject to} \quad & \sqrt{\gamma}\sum_{i\in\mathcal{N}} w_i p(u_i) \le C \\ & p(u_i)\in[0,1] \quad \forall\, u_i \in \mathcal{N}. \end{aligned}$$

We structure the cost constraint in this optimization problem in a heterogeneous manner. Under this setting, w_i is the cost of recruiting user u_i in a newly created forum. u_i will be included in the forum with probability $p(u_i)$ such that the average cost of having u_i in the forum would be $w_i p(u_i)$ ($w_6 p(u_6)$ in Figure 2.3). This setting is very general and can capture various scenarios such as w_i being larger (more expensive to recruit) for more popular users/influencers as would be the case in reality. The company also allocates a portion of its budget towards γ, which captures the overall communication efficiency across the channels established for user interaction, as part of creating the discussion forum. We reflect this in the budget constraint in our optimization problem $\mathcal{O}$, with the presence of

the $\sqrt{\gamma}$ term. This ensures that the scaling between γ and $\mathbf{u}$ is the same as $\gamma\mathbf{u}\mathbf{u}^T$ as in (2.6). To be more specific, for any choice of $\mathbf{u}$ up to a multiplicative constant, we can allocate C between γ and $\mathbf{u}$ without affecting the fixed point $\mathbf{y}$. Our scaling ensures that $\hat{\mathbf{u}} = c\mathbf{u}$ and $\hat{\gamma} = \gamma/c^2$ for any constant c renders the terms $\gamma\mathbf{u}\mathbf{u}^T$ unchanged. Any other scaling would force us to consider every possible value of $\mathbf{u}$ and γ *separately* and thus obfuscate the meaning of the budget constraint for our optimization problem and subsequently our analysis on the optimal choice of $(\gamma, \mathbf{u})$ on the maximal market share $\bar{y}$. This completes our problem set up, and we are now ready to address its analysis in the next section.

2.6 Main Results

In this section, we begin by first analyzing the budget constraint in $\mathcal{O}$ and the minimum amount of budget needed to ensure positive market share of $\mathbf{P}_2$, $\bar{y} > 0$. We also establish a key result on the behavior of $\bar{y}$ as the entries $p(u_i)$ of the vector of probabilities $\mathbf{u}$ (probability with which any user $u_i \in \mathcal{N}$ actively engages in the forum) are perturbed. We then develop our main algorithm, which is an iterative procedure of updating $\mathbf{u}$, consequently helping towards obtaining an improved market share of $\mathbf{P}_2$ under the same budget constraint. We defer to the Appendix all the technical proofs of the results presented in this section.

2.6.1 Minimum Budget for $\bar{y} > 0$

To begin, we provide the following proposition regarding the budget constraint in $\mathcal{O}$.

Proposition 1. *Any solution* $(\gamma^*, \mathbf{u}^*)$ *to the optimization problem* $\mathcal{O}$ *always satisfies the budget constraint with equality, that is,*

$$\sqrt{\gamma^*} \sum_{i \in \mathcal{N}} w_i p(u_i)^* = C.$$

Proposition 1 tells us that the optimal market share is always obtained by spending

all the available budget. This follows from the monotonicity property[4] of the ODE system (2.4) in that $\bar{y}$ turns out to be an increasing function of γ and $\mathbf{u}$, albeit being nonlinear. See Appendix A.2 for details.

For general bi-SIS models, $\mathbf{P}_2$ needs to satisfy a threshold type survival condition $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T]) > 1$ in order to eventually secure a positive market share. Note that this survival condition is basically the type of equilibrium point (iii) with $\mathbf{B} = \mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T$, as outlined in Section 2.4. This condition captures the relationship between the initial market share of $\mathbf{P}_1$, given by $\bar{x}^* = 1/N\sum_{i\in\mathcal{N}} x_i^*$, and the budget C, which is a function of the pair $(\gamma, \mathbf{u})$. Specifically, for a given τ_2 and any $\mathbf{u}$ fixed up to a multiplicative constant, the survival condition indicates that when $\bar{x}^*$ is large, the term $\gamma\mathbf{u}\mathbf{u}^T$ also needs to be large (to ensure a sufficiently large eigenvalue[5]) to guarantee long-term survival for $\mathbf{P}_2$, indicating a large budget is necessary. This implies that as the market share of $\mathbf{P}_1$ increases, the minimum budget required for $\mathbf{P}_2$ to survive also increases, which can also be seen from the heatmap[6] in Figure 2.4. For an arbitrarily chosen $\mathbf{u}$, fixed up to a multiplicative constant, Figure 2.4 illustrates how the steady-state market share of $\mathbf{P}_2$ varies with C and $\mathbf{x}^*$. The red curve in Figure 2.4 indicates the minimum budget required to attain $\bar{y} > 0$, and any amount spent below the minimum budget leads to a zero market share. In other words, a positive budget spent by the company towards its product placement does not necessarily guarantee even a small non-zero return. The company launching a new product must account for the existing competitor's market share to decide the minimum budget to ensure its non-zero long-term market share.

[4]Arising from the fact that the bi-SIS system (2.2) is a *monotone dynamical system* [Dos21].

[5]This follows from the fact that the spectral radius of a non-negative matrix is a non-decreasing function of its elements [Mey00] and the terms in $\mathbf{S}_{\mathbf{x}^*} = \text{diag}(\mathbf{1}-\mathbf{x}^*)$ is decreasing in $\mathbf{x}^*$.

[6]We numerically simulate the ODE system (2.4) for a small graph of dolphins' population (62 nodes, 159 edges) [Lus03].

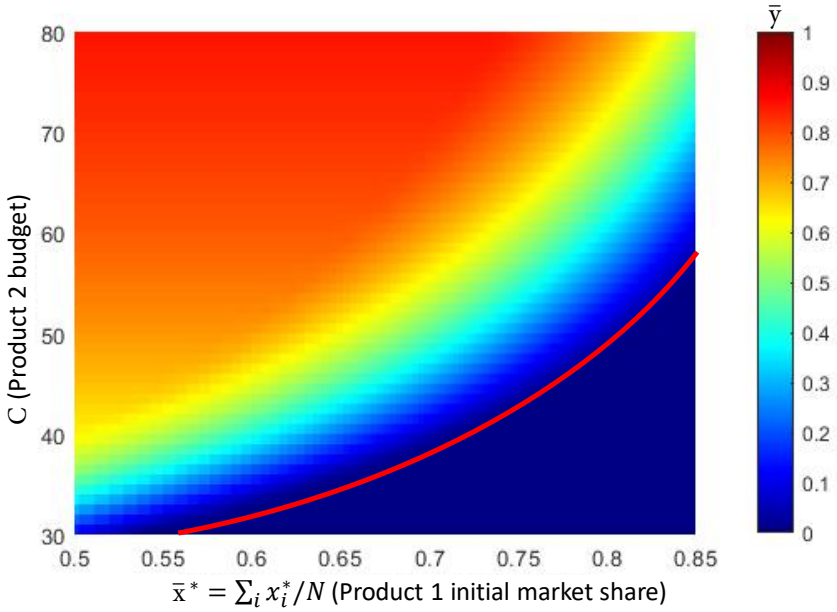


Figure 2.4 Heat map illustration of the market share as a function of budget C and $\bar{x}^*$.

2.6.2 Analysis of P_2's Market Share

We now turn our attention towards establishing a key result on the sensitivity of $\bar{y}$, with respect to entries $p(u_i)$ of $\mathbf{u}$. This result will form a basis towards developing the algorithm which we present later in this subsection.

For a general graph with adjacency matrix $\mathbf{A}$, it is impossible to obtain the optimal solution to $\mathscr{O}$ in a closed form, due to the absence of a closed-form expression of $\bar{y}$. Even for the single-SIS model in a general graph setting, there is no known closed-form solution for the positive fixed point. The situation is further complicated when competition comes into play in the bi-SIS model. Therefore, to gain better insight into $\bar{y}$ and how it changes with different model parameters, we begin by applying a small perturbation to $p(u_i)$ for each $u_i \in \mathscr{N}$. As previously mentioned, change in γ does not affect the fixed point in (2.5), (2.6) when the budget is held constant, since the term $\gamma\mathbf{u}\mathbf{u}^T$ remains the same under the aforementioned scaling for γ and $\mathbf{u}$ with the same budget C. The only way to obtain an increase in $\bar{y}$ is by *heterogeneously* changing the entries $p(u_i)$, and thereby obtaining an *updated* version which is no longer a multiplicative constant of the initial $\mathbf{u}$.

To proceed, we first define the terms *critical curve* and *critical parameters*, which we use throughout the rest of this section. The critical curve is the set of parameters at which the aforementioned survival condition is satisfied with equality, and we call any such set of parameters *critical* if they are on the critical curve. Let $(\gamma^c, \mathbf{u}^c)$ denote a pair of critical parameters for which $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma^c\mathbf{u}^c\mathbf{u}^{c\,T}]) = 1$. For any given pair, let $\boldsymbol{\nu}^c = [\nu_i^c]$ be the PF eigenvector[7] of $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma^c\mathbf{u}^c\mathbf{u}^{c\,T}]$. From the threshold condition we know that $\bar{y} = 0$ when $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma^c\mathbf{u}^c\mathbf{u}^{c\,T}]) = 1$. As a first step towards obtaining a positive $\bar{y}$, we apply perturbation $\alpha_i > 0$ ($\boldsymbol{\alpha} = [\alpha_i]$) to any i^{th} entry $p(u_i)^c$ of $\mathbf{u}^c$. The updated entry is then given by $p(u_i)^c + \alpha_i$, and we have the following result.

Theorem 1. *The market share of $\boldsymbol{P}_2$ as a function of the perturbation $\alpha_i > 0$ to $p(u_i)^c$ can be written as*

$$\bar{y}(\alpha_i) = \zeta\, \nu_i^c\, \alpha_i + O(\alpha_i^2), \tag{2.7}$$

where $\zeta > 0$ is a constant independent of i.

Theorem 1 tells us that the magnitude of increase in $\bar{y}$ due to an increase in $p(u_i)^c$ (that is, α_i) is roughly proportional to ν_i^c with an error term[8]. This implies that there is a greater reward in terms of market share to invest in nodes with larger ν_i^c. Although the result from Theorem 1 helps in obtaining a direction leading to an increase in $\bar{y}$, the updated $\mathbf{u}^c$ as a result of perturbation no longer lies on the critical curve. To obtain the next direction of increment in $\bar{y}$, we re-scale the updated $\mathbf{u}^c$ such that the re-scaled system operates on the critical curve again, to which we can apply Theorem 1 with the newly re-scaled parameters to find the next local move (perturbation). In what follows, we utilize this observation as a basis towards designing our iterated search algorithm, which helps in obtaining a "better" $\mathbf{u}$ for a larger market share under the same budget constraint.

[7]The Perron-Frobenius (PF) eigenvector is the eigenvector corresponding to the largest eigenvalue of a non-negative, irreducible matrix [Mey00].

[8]Note that Theorem 1 shows the sensitivity of $\bar{y}$ with respect to 'one' particular $\alpha_i > 0$ for which we expect to see the best possible increase in the market share. Any $\alpha_i < 0$ results in a zero market share $\bar{y} = 0$ since $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma^c\mathbf{u}^c\mathbf{u}^{c\,T}]) \leq 1$.

Algorithm 1: Iterated Local Search for bi-SIS

Input: $\mathbf{A}, \tau_1, \tau_2, \mathbf{w}, \epsilon, \theta$
Output: $\mathbf{u}$
Initialize:
  $\mathbf{v}(0) = \boldsymbol{\nu}^c$ /* PF eigenvector of $\mathbf{S}_{\mathbf{x}^*}\mathbf{A}$ */
  $\mathbf{u}(0) = \mathbf{u}_0$ /* as in Lemma 1 */
  $\gamma(0) = \gamma_0$ /* as in Lemma 1 */
  $\epsilon(0) = \epsilon$
for *k = 1 to # iterations* **do**
  $\epsilon(k) \leftarrow \min\{\epsilon, min_{i \in S(\theta)} p(u_i)(k-1)\}$
  $\boldsymbol{\alpha}(k) \leftarrow \mathscr{P}(\mathbf{v}(k-1), \epsilon(k), \mathbf{u}(k-1))$ /* Solution to the optimization problem */
  $\mathbf{u}(k) \leftarrow \mathbf{u}(k-1) + \boldsymbol{\alpha}(k)$
  $\tilde{\gamma}(k) \leftarrow \gamma(0) - (\hat{\lambda} - 1/\tau_2) \frac{\mathbf{v}(k-1)^T \mathbf{S}_{\mathbf{x}^*}^{-1} \mathbf{v}(k-1)}{[\mathbf{v}(k-1)^T \mathbf{u}(k)]^2}$
  $\mathbf{v}(k) \leftarrow$ PF eigenvector of $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \tilde{\gamma}(k)\mathbf{u}(k)\mathbf{u}(k)^T]$
end

2.6.3 Iterated Local Search for bi-SIS

Suppose we have an initial pair of model parameters which satisfy the threshold condition, and thus form a critical pair $(\gamma^c, \mathbf{u}^c)$. These chosen parameters would then satisfy $C = \sqrt{\gamma^c} \sum_{i \in \mathscr{N}} w_i p(u_i)^c$. By keeping the budget constant, we intend to move in a direction that obtains the best possible increase in $\bar{y}$ from its initial value of $\bar{y} = 0$ at the critical point. We do so by applying a perturbation to $p(u_i)^c$, which needs to have both positive and negative components to ensure that the budget constraint remains binding. Also, Theorem 1 indicates that the increase in $\bar{y}$ is roughly proportional to v_i^c. We leverage this result and the perturbation approach to formulate another optimization problem $\mathscr{P}(\boldsymbol{\nu}, \epsilon, \mathbf{u})$. Let $S(\theta) = \{u_i \in \mathscr{N} | p(u_i) > \theta\}$ be the set that determines which entries of $\mathbf{u}$ to update, for a given threshold $\theta > 0$. Since Theorem 1 holds in a small region around the critical curve, we limit the maximum size of the perturbation to $\epsilon > 0$. Specifically, for a given initial $\mathbf{u}^c$, we define $\tilde{p}(u_i) \triangleq p(u_i)^c + \alpha_i \; \forall \, i \in \mathscr{N}$ where $\alpha_i \in [-\epsilon, \epsilon]$ and consider the following:

Algorithm 2: Solution to $\mathscr{P}(\boldsymbol{\nu}, \epsilon, \mathbf{u})$

Input: $\boldsymbol{\nu}, \mathbf{w}, \epsilon$

Output: $\boldsymbol{\alpha}$

1 $W = \sum_{i \in \mathscr{N}} w_i/2$

2 $\sigma = \text{sort}(\nu_i/w_i, \text{descend})$ `/* ∀i ∈ 𝒩 Obtain permutation (ordering) of sorted array */`

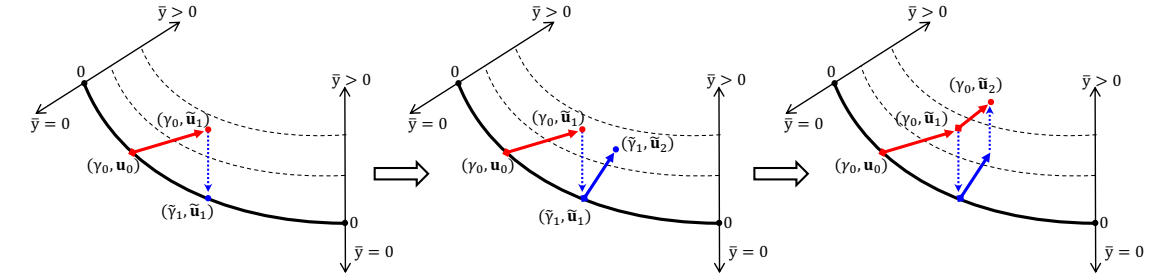
3

$$\alpha_{\sigma(i)} = \begin{cases} \epsilon & \text{for } \sum_{i=1}^{k-1} w_i \leq W \\ \epsilon\left[\left(2(W - \sum_{j=1}^{k-1} w_j)/w_k\right) - 1\right] & \text{for } i = k \text{ s.t. } w_k > W \\ -\epsilon & \text{for } i = k+1, \cdots, N \end{cases}$$

$$\begin{aligned} \mathscr{P}(\boldsymbol{\nu}, \epsilon, \mathbf{u}) \qquad & \underset{\alpha_1, \cdots, \alpha_N}{\text{maximize}} \sum_{i \in \mathscr{N}} \alpha_i \nu_i \\ & \text{subject to} \sum_{i \in \mathscr{N}} w_i \alpha_i = 0, \\ & \qquad S(\theta) = \{u_i \in \mathscr{N} | p(u_i) > \theta\} \qquad \text{where } \theta > 0, \\ & \qquad \alpha_i \in [-\epsilon, \epsilon] \qquad \forall i \in S(\theta), \\ & \qquad \alpha_i = 0 \qquad \forall i \in \mathscr{N} \setminus S(\theta). \end{aligned}$$

In the above optimization problem, the step size ϵ is a small constant that controls the maximal size of the perturbation in $\mathbf{u}$. To avoid the possibility of having negative entries in the perturbed $\mathbf{u}$, we choose the step size at the k^{th} iteration such that $\epsilon(k) = \min\{\epsilon, \min_{i \in S(\theta)} p(u_i)(k-1)\}$, as in line 7 of Algorithm 1.

The solution $\boldsymbol{\alpha}$ to $\mathscr{P}(\boldsymbol{\nu}, \epsilon, \mathbf{u})$ is based on the decreasing order of (ν_i/w_i) such that it satisfies the given constraints. This solution is essentially a relaxation of the well-known *0-1 Knap-sack* problem [Gar79], as shown in Algorithm 2, which for a given initial critical parameter pair $(\gamma^c, \mathbf{u}^c)$ leads to a direction resulting in the best possible local increase in $\bar{y}$. We illustrate this direction in Figure 2.5(a), where the red solid arrow shows the direction of increase in $\bar{y}$ for a given starting pair of critical parameters, say $(\gamma_0, \mathbf{u}_0)$ (representing



(a) First step in the direction of increment in $\bar{y}$ by perturbing $\mathbf{u}_0$ to $\tilde{\mathbf{u}}_1$.

(b) Adjusting γ_0 to $\tilde{\gamma}_1$ to obtain the next direction of increment in $\bar{y}$.

(c) $(\gamma_0, \tilde{\mathbf{u}}_2)$ pair gives the potential increment in $\bar{y}$

Figure 2.5 Schematic of Algorithm 1.

$(\gamma^c, \mathbf{u}^c)$). As illustrated, we have $\bar{y} > 0$ for the updated parameters $(\gamma_0, \tilde{\mathbf{u}}_1)$, which are no longer a critical pair. In order to obtain the next direction, our approach is to adjust $(\gamma_0, \tilde{\mathbf{u}}_1)$ to $(\tilde{\gamma}_1, \tilde{\mathbf{u}}_1)$ such that it lies on the critical curve (the thick black curve in Figure 2.5(a) on which $\bar{y} = 0$) again. The blue dotted arrow in Figure 2.5(a) shows this change in $(\gamma_0, \tilde{\mathbf{u}}_1)$. Note that the new pair of critical parameters, essentially obtained by reducing γ_0 to $\tilde{\gamma_1}$, will then be associated with a lower budget. We incorporate this auxiliary step of reduction in γ_0 only for the purpose of finding the next direction of increment in $\bar{y}$. The resulting $\tilde{\gamma_1}$ from the original γ_0 (or γ^c) in this transformation can be written as (see Appendix A.4 for the detailed derivation)

$$\tilde{\gamma}_1 = \gamma_0 - [\hat{\lambda} - 1/\tau_2] \frac{\boldsymbol{\nu}^{c\,T} \mathbf{S}_{\mathbf{x}^*}^{-1} \boldsymbol{\nu}^c}{(\boldsymbol{\nu}^{c\,T} \tilde{\mathbf{u}}_1)^2}, \tag{2.8}$$

where $\hat{\lambda} = \lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \gamma_0 \tilde{\mathbf{u}}_1 \tilde{\mathbf{u}}_1^T])$. We then solve the optimization problem $\mathscr{P}(\boldsymbol{\nu}, \epsilon, \mathbf{u})$ now with a newly updated PF eigenvector $\tilde{\boldsymbol{\nu}}$ of $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \tilde{\gamma_1} \tilde{\mathbf{u}}_1 \tilde{\mathbf{u}}_1^T]$, whose solution can then be used to update $\tilde{\mathbf{u}}_1$ to $\tilde{\mathbf{u}}_2$, as shown by the blue solid arrow in Figure 2.5(b). Since the auxiliary step of reducing γ_0 to $\tilde{\gamma}_1$ is only for the purpose of finding the next direction of increment in $\bar{y}$ from the critical curve, the iteration at this point would correspond to $(\gamma_0, \tilde{\mathbf{u}}_2)$, as shown in Figure 2.5(c). While we can perform this step repeatedly to find the feasible directions for increment for $\bar{y}$, our simulation results show that we can obtain a large increase in $\bar{y}$ with

just a few iterations.

Now, we shift our attention to the initialization of our iterative algorithm (Algorithm 1). Recall that we defined the critical curve as the set of critical parameters satisfying the threshold survival condition with equality. Lemma 1 below provides a feasible initial pair of critical parameters.

Lemma 1. *Let* $\mathbf{v} \in \mathbb{R}^N$ *be the PF eigenvector of* $\mathbf{S}_{\mathbf{x}^*}\mathbf{A}$ *associated with* $\lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{A})$. *Then, for the choice of* $\mathbf{u}_0$ *and* γ_0 *such that*

$$\mathbf{u}_0 \propto \mathbf{S}_{\mathbf{x}^*}^{-1}\mathbf{v}, \quad \textit{and} \quad \gamma_0 = \frac{1/\tau_2 - \lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{A})}{\mathbf{u}_0{}^T \mathbf{S}_{\mathbf{x}^*}\mathbf{u}_0},$$

we have $\tau_2 \lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \gamma_0 \mathbf{u}_0 \mathbf{u}_0^T]) = 1$.

2.7 Experimental Setup & Results

In this section, we present simulation results on real-world graphs under the cost constraint for a given initial $(\gamma, \mathbf{u})$ pair according to Lemma 1. We first briefly describe the simulation setup for our experiments, which includes the fixed budget structure and then describe the baseline methods that we use to compare its performance against our approach.

2.7.1 Simulation Setup

We use the following two real-world social network based graph datasets for the simulations, which are undirected, connected graphs from the SNAP repository [Les14].

1. Social Circles–Facebook: This dataset consists of a friends list of users using the Facebook app, which contains 4,039 nodes and 88,234 edges [McA12].

2. Deezer Europe Social Network: A user–user network of European members of the music streaming service Deezer. Nodes are Deezer users and edges are mutual follower relationships between users. It contains 28,281 nodes and 92,752 edges [Roz20].

As $\mathbf{P}_1$ already exists in the network and $\mathbf{P}_1$ stronger than $\mathbf{P}_2$, the effective infection/adoption rates are initially $\tau_1\lambda(\mathbf{S}_{\mathbf{y}^*}\mathbf{A})>1$ and $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{A})\leq 1$. In the simulation results, we are interested in the immediate increment of $\lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{B})$ (where $\mathbf{B}=\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T$) from its initial value $\tau_2\lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{B})=1$ for a given initial pair $(\gamma^c,\mathbf{u}^c)$. For more iterations, the eigenvalue will no longer lie in a small region around $1/\tau_2$.

We set the budget as per Proposition 1 and run our simulation for two scenarios: (i) when the cost of resource allocation is homogeneous across all nodes, i.e., $w_i=1$ for all $i\in\mathcal{N}$ and, (ii) when the cost of resource allocation is heterogeneous, that is, one instance of $\mathbf{w}$ sampled uniformly at random.

2.7.2 Baseline Methodology Comparison

To evaluate the performance of our Algorithm 1, we compare it against standard centrality measures such as degree-based centrality and eigenvector-based centrality, as baseline algorithms. Degree-based centrality is a measure of node connectivity based on the number of links (degree) each node has with others. This is useful in identifying popular nodes in a network who can further connect to a wider network [New06; New18]. Similarly, eigenvector-based centrality (EVC) measures a node's influence on other nodes in the network. This is done by assigning an importance score based on the fact that a node is important when it is linked to another important node. EVC identifies not only the influence of a node on a direct connection, but also the impact on the entire network [Was94]. We chose to compare against these measures to gain a holistic analysis against such well-known centrality methods which are generally used to understand the relationship structure in a social network [Fre78; Fri91].

In each of these baseline methodologies, while the total budget remains the same, that is, $C=\sum_i\sqrt{\gamma}w_i p(u_i)$, we assign the cost (either homogeneous or heterogeneous) only to the top-k nodes of the community based on the ratio of its degree and importance score for degree and eigenvector centrality measures, respectively, as outlined in Algorithm 3. For our simulation results, in each of the centrality measures, k in top-k varies from 10,

Algorithm 3: Baseline Methodology

Input: $\mathbf{A}, \tau_1, \tau_2, \gamma_0, k, C, \mathbf{w}, \mu$

`/* μ = chosen centrality measure */`

Output: **u**

1 T = {top-k nodes according to μ}

2 $p(u_i) = \frac{\mu_i w_i}{\sum_{j \in T} \mu_j w_j}$ for all $i \in T$

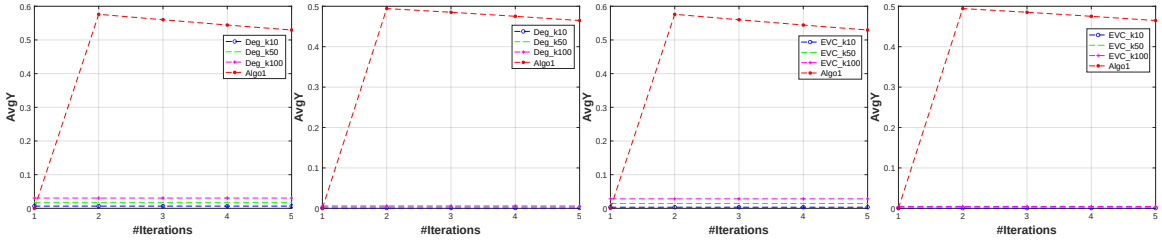
3 $p(u_i) = 0$ for all $i \in N \setminus T$

50, and 100, beyond which the number of degrees per node decreases to a smaller value of around 10-15 and an importance score per node goes close to zero and hence, need not be taken into consideration.

2.7.3 Simulation Results

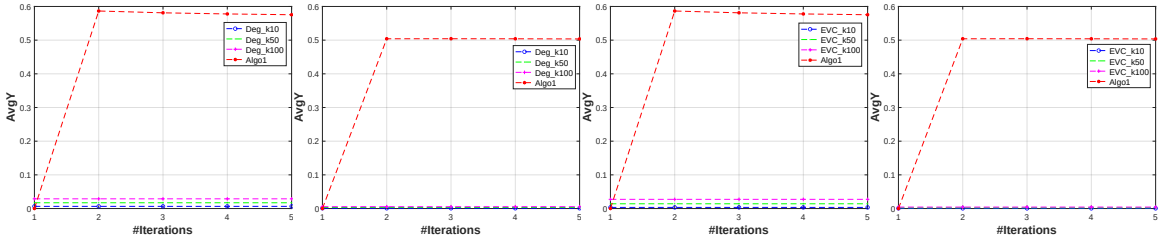
Here, we present the comparison results for both the homogeneous and heterogeneous cost distribution for the above mentioned real-world graph datasets. To visualize our system's two dimensions, we represent the market share on the y-axis as $AvgY = {}^{1}/_{\mathrm{N}} \sum_{i \in \mathcal{N}} y_i$ and the number of iterations on the x-axis. We run the algorithm only for a very few iterations as mentioned in Section 2.6.3. If the algorithm is run for a larger number of iterations, $\bar{y}$ decreases and we aim for the best possible increase in $\bar{y}$.

The results show that for the locally optimal choice of nodes returned by our algorithm, under both heterogeneous and homogeneous cost distribution, the initial weaker/new product has a higher market share. This is better than that achieved using the baseline methods, thus validating our theoretical results, as shown in Figure 2.6 and 2.7. It also indicates that, irrespective of the size of the graph, our algorithm leads to a better market share for an initially weaker product. Note that similar numerical results can indeed be obtained for even larger graphs than shown here, for any choice of budget, C, as long as the cost assigned to each u_i is not too large. The details are omitted here to avoid repetition. With respect to the worst-case time complexity of Algorithm 1, as the *for* loop (line 6)



(a) Facebook: Algo1 vs Degree-based **(b)** Deezer: Algo1 vs Degree-based **(c)** Facebook: Algo1 vs EVC-based **(d)** Deezer: Algo1 EVC-based

Figure 2.6 Change in $\bar{\mathbf{y}}$ (AvgY) for homogeneously distributed cost against Degree-based & Eigenvector-based centrality measure.



(a) Facebook: Algo1 vs Degree-based **(b)** Deezer: Algo1 vs Degree-based **(c)** Facebook: Algo1 vs EVC-based **(d)** Deezer: Algo1 vs EVC-based

Figure 2.7 Change in $\bar{\mathbf{y}}$ (AvgY) for heterogeneously distributed cost against Degree-based & Eigenvector-based centrality measure.

executes for a fixed number of iterations (constant upper bound), the run-time does not depend on N and hence, the time complexity is $\mathcal{O}(1)$. However, the *for* loop derives the solution to the optimization problem $\mathscr{P}(\boldsymbol{\nu}, \epsilon, \mathbf{u})$ (based on fractional knapsack [Ish77]) whose time complexity is $\mathcal{O}(N\,log\,N)$, where N is the number of nodes. Hence, nesting the $\mathcal{O}(N\,log\,N)$ inside the outer $\mathcal{O}(1)$ loop, the overall run time results in $\mathcal{O}(N\,log\,N)$.

2.8 Conclusion

We have developed a heuristic algorithm that returns an optimal set of nodes that form a community, that is, the best possible pair of $(\gamma^c, \mathbf{u}^c)$. This set of nodes when targeted with advertisements under a given budget constraint results in a positive market share of the initially weaker product $\mathbf{P}_2$. The algorithm uses an iterated local search technique to

identify the set of nodes that will be influenced through advertisement so that $\mathbf{P}_2$ can enter the network with a positive market share. We also compare our proposed algorithm against baseline algorithms using two different datasets. The results indicate that Algorithm 1 outperforms both when the budget is assigned homogeneously or heterogeneously to nodes that are part of the community. We believe that our work is the first step towards finding an optimal size of community for a new firm (launching a product) to compete against a product dominant in a market. By basing our theoretical analysis using the bi-SIS epidemic model and extensive simulation on different real-world graphs we demonstrate the efficacy of our algorithm, including its improved performance over the standard centrality measures.

CHAPTER 3

Visualizing Product Features to Determine Users' Purchasing Decisions

3.1 Introduction

With the exponential growth of e-commerce websites, users now have access to a large number of options to buy at the click of a mouse. Due to such ease of accessibility, in order to improve the shopping experience, it has become a common practice by online merchants to provide the ability for users to express their opinions or views regarding the products they buy in the form of text such as blogs, forum discussions, or product reviews. The range of issues that users share in their reviews vary from a particular product, movie, hotel, restaurant, or travel destination. These online reviews can be in the form of a text, or a rating (for example, a star-rating on the range of 1 to5), or both. The reviews act as a medium for potential customers to evaluate a decision before making any purchase. From a product manufacturer's point of view, it is helpful to gather useful insights regarding

different aspects of a product. As the number of product reviews being posted online is increasing rapidly, it is becoming especially difficult for potential customers to make an informed decision whether to buy the product.

Understanding how users form an opinion towards a product or what factors they take into account before making any purchasing decision is a non-trivial task. With the increase of user-generated content and the amount of public data currently available in the form of users' purchase histories, information from surveys, and online reviews posted in e-commerce websites, analyzing buying behavior is now possible. For example, as mentioned earlier, e-commerce websites provide extensive information about products that people browse before a purchase. These buying decisions are based on aspects such as quality or price of the product mentioned in reviews, ratings, and purchase history. From a customer's point of view, it is advantageous to compare product properties prior to forming a buying decision. From a company's perspective, reviews provide a medium to gather useful insights regarding which properties of a product motivate a potential customer to make a buying decision and how much weight such properties carry. With the help of these insights, we hope to understand the market structure more accurately and propose which product properties, especially for products with a lower market share, to enhance and improve to aid in an increase in the spread of the product's adoption.

Online reviews reflect a user's perception or sentiment towards the purchased product. Such user-generated data through reviews acts as a source for opinion mining or sentiment analysis, which includes techniques to detect and extract opinions or sentiment from text based on context. Although there are different areas of opinion mining, we focus on feature-based opinion mining where particular features of products are analyzed to determine their sentiment as positive, negative or neutral in the review. Features of a product are attributes, properties or other aspects of the product. For example, in the case of a smartphone, battery, camera quality, price, and reliability correspond to some of its features. In this work, we extract a product's important properties from reviews given on e-commerce websites, which

is further used to analyze sentiment by generating a property vector with three possible values 1, 0, or 1 for negative, neutral, or positive sentiment, respectively, per property in each review. These property sentiment vectors can then be used to predict the star rating given by a user. The predicted ratings act as an estimate of a user's opinion about a product. We provide the details of the proposed methodology and background of sentiment analysis and machine learning algorithms in Section 3.2.

Our overall goal here is to accurately construct property sentiment vectors based on a user's review text. It is *not* to construct a model to convert property sentiment vectors into estimated star ratings, since these are available directly from the review itself. Instead, our machine learning model construction and evaluation is designed to validate that we are constructing property sentiment vectors that accurately capture a user's overall sentiment about a product. The property sentiment vectors can then act as a proxy for the overall review sentiment and can be used as a more detailed representation of a user's views about a product for follow-on analysis and visualization.

With the large amount of data available, it is difficult to gain insights and extract an overall idea from the raw data alone, due to the sheer volume of data and the time needed to analyze it. An interactive visual representation can not only enhance users' insights into the data but also help in understanding its structure [Šil10]. Using product reviews as an example, the combination of a user's opinion, important product properties, and the information about users' social relationships enhances knowledge about the domain's structure while providing meaningful insights. Although there are various forms of visualization best suited to specific domains, data sources, and data formats, we focus on sentiment visualization within a social network. Sentiment visualization is a form of information visualization that deals with the presentation and visual analysis of "thoughts or feelings expressed by an individual" [Alm19]. Social network visualization falls within the area of graph visualization, typically represented as a node–link structure or an adjacency matrix. We use sentiment visualization to capture and present insights extracted from the product sentiment analysis

of online reviews of two competing products. The foundation for our visualization system integrates results obtained from sentiment analysis into the bi-SIS epidemic model (where users form opinions based on neighbors' influence).

To be specific, we study the extraction of a product's important properties based on product reviews to perform sentiment analysis to generate a negative–neutral–positive score for a set of pre-chosen product properties. We use these property sentiment scores to build a model to predict star ratings for review text. The predicted rating and its corresponding product sentiment vector act as a proxy for the probability of the reviewer purchasing the product or being influenced by their neighbors in a social network. As users have a positive, negative, or neutral sentiment towards a product's important properties, we use visualization to present node–link graphs that highlight properties with a higher importance for potential purchasers, along with their opinions towards the product. In this research, we study the problem of feature-based opinion mining from online product reviews in three steps:

1. Analyzing review datasets to extract the importance of a product's properties, rather than estimating these with exogenous model parameters as seen in bi-SIS or opinion dynamics based models.

2. Applying sentiment analysis and supervised machine learning on text discussing a product's properties to predict a review rating that acts as an estimate of a user's opinion about a product prior to a buying decision.

3. Developing a text-to-visual representation in the form of a node–link graph to highlight a product property's importance and to indicate how each user's social links influence forming an opinion regarding purchasing decisions.

3.2 Background & Related Work

Sentiment Analysis: In this section we provide a brief background and an overview of prior research related to text analysis techniques such as sentiment analysis or opinion mining, and the methodology that we use to predict a consumer's rating of a product. We focus on supervised machine learning models that use reviews with associated star ratings to build our prediction engine.

In recent years, the growing amount of text data, especially in product reviews, has driven the desire to analyze and extract contextual information from such data [Liu12b]. Sentiment analysis is a text analysis technique that automates the process of extracting a writer's feelings and emotions from text. Given text data, one common approach is to categorize it using *sentiment polarity:* positive or negative. This can be performed at different scales including document level, sentence level, or entity level [Liu12a]. The document level determines whether a document as a whole expresses positive or negative sentiment. The sentence level assigns sentiment polarity to each sentence in a document. The entity level targets identify exactly what people like or dislike from their opinions [Fan15].

Extensive research has been performed to study how words and phrases can be classified into positive or negative sentiment, a priori [Hat97]. Although such pre-classification is often helpful in many cases, the implication of positive or negative polarity can be entirely different based on context. For example, the polarity of smartphone product reviews has been estimated based on the positive and negative orientation of the review [Kum17]. However, identifying polarity at the level of product characteristics further enhances the sentiment analysis process. Such aspect-based sentiment analysis identifies aspects in a given review about a product and determines which class of sentiment the aspect belongs to, providing more granular sentiment polarity that can be aggregated to augment or replace overall review-based sentiment orientation.

3.2.1 Aspect-Based Sentiment Analysis

To estimate a user's preference score, we need a method to obtain a sentiment score from their review text. Because each of the reviews are based on a product's properties, we employ aspect-based sentiment analysis, where aspects in a given review are classified by sentiment polarity.

For a lexicon-based approach, the sentiment orientation and subjectivity of each term in a sentence usually defines how the sentiment polarity of the term is calculated. The polarity of an aspect determines if it implies a positive or negative sentiment on the range $[-1, 1]$. Subjectivity quantifies the amount of personal opinion and factual information contained in the text and lies on the range $[0, 1]$. A higher subjectivity means the text or data contains more personal opinion than factual information [Naz20]. For example, consider the sentence "I do not like this food item, it is very bland". The sentence has a negative sentiment polarity of -1 for the aspect *food* and a subjectivity of 1, implying that it is a personal opinion rather than factual information.

3.2.2 Machine Learning

In its simplest form, machine learning is designed to identify patterns that characterize individual samples and how they relate to one another. Practical use cases for machine learning include email spam filtering, fraud detection on social networks, online stock trading, face and shape detection, medical diagnosis, traffic prediction, character recognition, and product recommendations. Real-world examples include the self-driving Google cars, Netflix showcasing movies a person might like, online recommendation engines like friend suggestions on Facebook, "more items to consider" or "products you might like" on Amazon, and credit card fraud detection [Alz18]. A machine learning problem typically has three aspects: (i) the task that needs to be learned, (ii) a performance evaluation of a learned model, and (iii) model improvement based on its performance [Car83; Sam88].

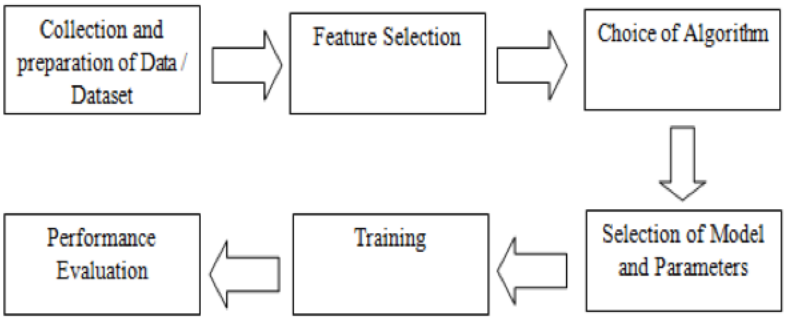


Figure 3.1 Components of Machine Learning Model

Independent of the algorithm adopted for achieving a task, the generic model of machine learning consists of the six components as shown in Figure 3.1.

Depending on the algorithm that is employed for training the machine learning model and the type of data set that is available, machine learning is broadly classified into supervised learning, unsupervised learning, and semi-supervised learning [Lan96]. Although we provide a brief background on each of these models, we focus on the supervised learning mechanism for this research work.

In supervised learning, a pre-labeled data set is used to train a model to predict an output *target* value based on properties of its corresponding data sample. This implies that a *training data set* exists where each entry is labeled with a correct output value. Normally, supervised learning is used to build prediction engines. First, the labeled data set is split into a *training set* and a *test set*. The training set is used by a machine learning algorithm to construct a model to predict the labeled value. Samples in the test set are then provided to the final model, and the label the model estimates is compared to the known, correct label to measure the model's accuracy.

Unlike supervised learning, unsupervised learning uses unlabeled data, that is, data with no output values or labels that an algorithm can use to model relationships between the sample and its label. Given this, an algorithm must use patterns and structures from the input data to characterize relationships within the dataset. The most well-known un-

supervised algorithms fall into the area of clustering. Here, a *similarity metric* is defined to calculate the similarity between any pair of samples. Once built, the similarity metric is used to build a similarity matrix of all pairwise similarities. The similarity matrix forms the basis for most clustering algorithms, since it allows the algorithms to group subsets of samples that are similar to one another.

With respect to semi-supervised learning, only part of the given input data is labeled and the rest is unlabeled. Here, a combination (hybrid technique) of supervised and unsupervised learning algorithms are used to predict labels. The other machine learning technique (in addition to supervised and unsupervised earning) that is also commonly used in an interactive environment to learn from actions and feedback is the reinforcement learning (RL). Unlike supervised learning where a set of correct actions are provided to perform a task, in RL rewards and punishments are used to reinforce positive and negative behavior. Also, unlike unsupervised learning where the aim is to find similarities or differences between data points, the goal of RL is to find an optimal model that maximises the total reward [Sut18].

3.2.2.1 Supervised Learning

Supervised learning uses an algorithm to learn a mapping function between input variables $\{x\}$ and an output value or label y, $y = f(x)$. Given a pre-labeled data set, the learning process *learns* the mapping using the training set. How this is done depends on the particular algorithm used. Once the learning process is complete, the final model is applied to the test set, comparing model predictions to known, correct labels to measure the accuracy of the model. Given appropriate accuracy and a representative training set, the model can then be used to predict labels for new, unlabeled data [Rus02]. Supervised learning can be further divided into two types of problems.

1. **Classification.** A classification problem uses a label from a discrete set of possible categories such as yes–no, true–false, or positive–neutral–negative. Well-known al-

gorithms to build predictive models for this type of problem include support vector machines (SVM), decision trees, and k-nearest neighbors (KNN).

2. **Regression.** A regression problem predicts a continuous variable value such as temperature forecasting or stock market movement. Algorithms used to solve regression problems include linear regression, logistic regression, polynomial regression, and Bayesian linear regression.

Given the main focus of our work is based on predicting the star rating (*e.g.,* one to five stars) of a product based on the product's properties, we employ classification-based supervised learning algorithms to solve our problem. We provide background on each of the supervised algorithms that we compared when we discuss how we built a model to predict star ratings from user reviews.

3.2.3 Sentiment Visualization

In this section, we provide a brief background and an overview of prior research related to the area of sentiment visualization. Exploiting an individual's sentiment towards an item based on its properties and their relevance to the individual can be performed through the visualization of Natural Language Processing-based (NLP-based) sentiment analysis results. Sentiment visualization has gained attention among researchers over the past years. We provide background details on techniques which are closely related to our work.

Gregory et. al. developed IN-SPIRE, a visual analytic tool for document analysis that, among other capabilities, helps users explore documents and classify them based on their content. IN-SPIRE supports sentiment visualization where the sentiment of lexicons are categorized into four *affects*: positive/negative, virtue/vice, cooperative/conflict, and pleasure/pain [Gre06]. As shown in Figure 3.2(a), each pair of affects has a common hue, where the more positive a pair the lighter its saturation.

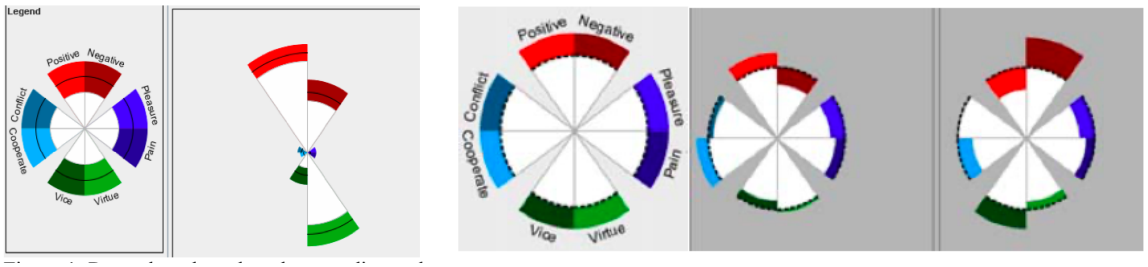
In order to determine how affect varies across a set of documents, IN-SPIRE uses a

Nightingale chart or rose chart [Nig59]. Nightingale charts were originally used by Florence Nightingale during the Crimean war to demonstrate to the British generals how improving sanitary conditions in field hospitals could prevent the majority of deaths during the war. A Nightingale chart splits a circle into segments representing different categories (e.g., months in Nightingale's original charts). Each slice or "petal" can be further subdivided to represent subcategories within each category (preventable deaths, deaths by wounds, and deaths by other causes in Nightingale's original charts). IN-SPIRE uses this approach to subdivide the circle into eight petals representing the eight different affect endpoints. Colour is used to identify each affect: red for positive/negative, green for virtue/vice, and two blues for cooperative/conflict and pleasure/pain. Lighter saturations identify positive affects and darker saturations identify negative affects. A petal uses a dark line to identify the median, with the colour regions below and above the line representing the second and third quartiles, respectively (Figure 3.2(a,b), left sub-image).

IN-SPIRE also visualizes whether a document's text (e.g., a product discussed in a customer review) is relatively more positive or negative across aspect endpoints using the concept of deviation from an expected value. The unit circle is represented as a dotted circle with a unit radius (Figure 3.2(b), right two sub-images). Colour segments that extend beyond the unit circle have a higher than expected value. Colour segments within the unit circle have a lower than expected value.

IN-SPIRE's sentiment charts identify whether a product is discussed in a document (e.g., a customer review) in a relatively more positive or negative way. A circle (as shown in Figure 3.2(b)) portrays the expected affect value. The positioning of the coloured regions in each petal visualizes larger than expected values (outside the unit circle) or smaller than expected values (inside the circle).

The authors demonstrate IN-SPIRE in a case study of reviews for five different products. Figure 3.3 shows the affect scores of two products being discussed in their respective reviews: an Apex DVD player and a Nokia cell phone, where Figure 3.3(a) shows that although both

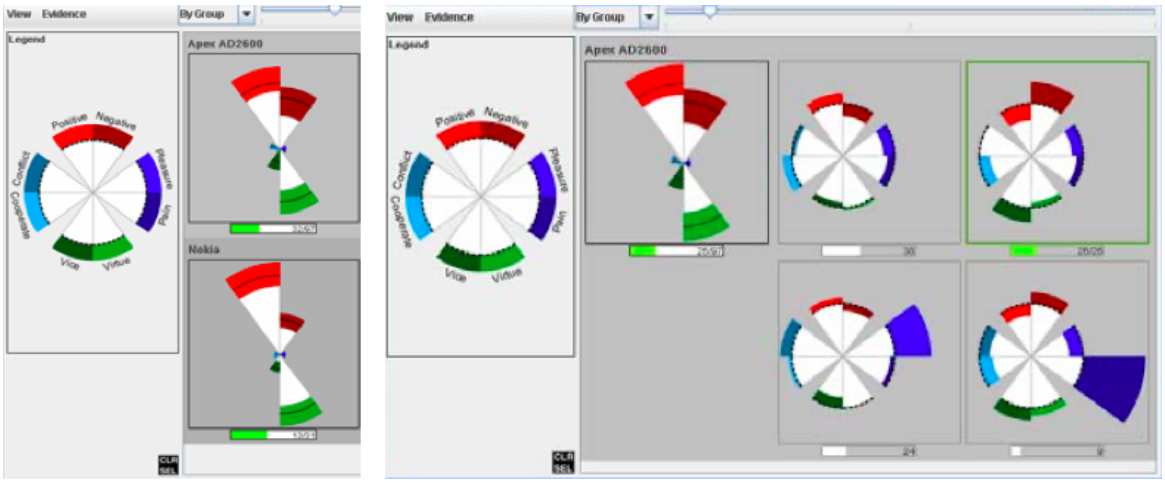


(a) **(b)**

Figure 3.2 IN-SPIRE visualization tool: (a) rose plot showing median and quartile variation, (b) rose plot showing deviation from expected value.

the products have positive reviews, Apex has stronger negative ratings than Nokia. Figure 3.3(b) illustrates how each of the properties of the Apex DVD player mentioned in the reviews deviate from their expected values. We see that for each of the words associated with the Apex DVD player in all the reviews, there are four sub-clusters with different affect variations. For example, one of the sub-clusters has a large Pain petal (Figure 3.3(b), lower right corner) indicating a higher deviation than expected. Similarly, the lower left rose plot in the figure has an affect category with a large Pleasure petal, again a higher deviation than the expected value. One can observe from these plots that two of these sub-clusters are more positive than expected and the other two are more negative than expected. This implies that there are sub-clusters within all reviews about the Apex DVD player which have both larger positive and negative affect categories than the expected value. It is interesting to see how different types of visual exploration techniques help users to gather insights from multiple documents in a single visualization.

OpinionBlock is a related visualization tool developed by Oelke et. al. that creates a feature-based visual summary to support interactive analysis of reviews and ratings [Oel09]. OpinionBlock provides a summary representation of high-dimensional feature vectors to present an overview of relevant information. The feature vectors summarize the expressed opinion for each attribute, where the corresponding value of a vector indicates if the attribute was discussed in a positive, negative, or neutral manner. The framework displays different shades of polarity: blue for positive, red for negative, and higher saturation for larger values. The intervals for the four shades of blue–red colours are determined by the



(a) Comparison of Affect scores

(b) Summary of Apex products

Figure 3.3 Sentiment visualization of two products in a rose plot with different colors

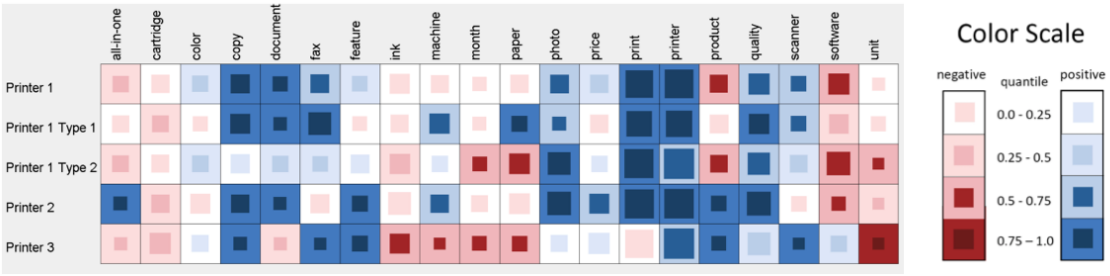


Figure 3.4 OpinionBlock visualization tool showing summary report of attributes of three printers mentioned in reviews.

quantiles of the positive or negative attributes [Oel09]. Figure 3.4 provides the summary report of three printers where each row represents a product and each column represents an attribute. The cell's colour identifies an attribute's polarity (blue or red) and the quartile where its opinion fell. The size of the inner square shows the number of customers that commented on the given attribute (i.e., the importance of the attribute based on its frequency of occurrence in customer reviews).

OpinionBlock also provides an interactive circular correlation map that allows analysts to detect relationships between attributes and their user-provided scores. The map represents a feature vector with a line from the review ID on the right of the review's score (on a five-point scale) in the center, and then from the review score to each attribute mentioned in the review on the left [Oel09]. Review score–attribute lines are coloured based on

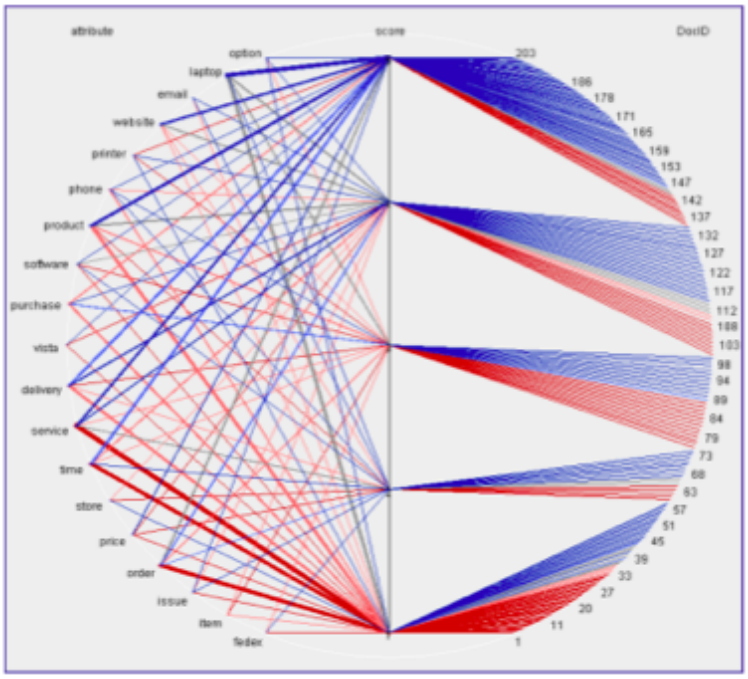


Figure 3.5 OpinionBlock circular correlation map showing all customers comments.

the opinions expressed. A line's thickness represents how often a particular attribute was present in the set of reviews. Figure 3.5 shows an example of a correlation map for reviews of Microsoft's Notebook.

Similar to OpinionBlock, OpinionSeer is another visual analytic tool developed by Wu et. al. that employs a radial structure and uses feature-based opinion mining techniques to model the uncertainty in reviews, specifically in hotel reviews [Wu10]. Here, uncertainty refers to the uncertainty of a customer's opinion when they comment both positively and negatively about a hotel. OpinionSeer presents two views, an opinion wheel and tag clouds, to analyze customer feedback. The opinion wheel is a combination of an opinion triangle based on a scatterplot and an opinion ring that forms a radial visualization (Figure 3.6). The figure shows how customers' opinions in hotel reviews are correlated with trip type, customer group, and age range.

The three vertices of the opinion triangle represent positive, negative, and uncertain opinions, where the position of each customer inside the triangle is located according to

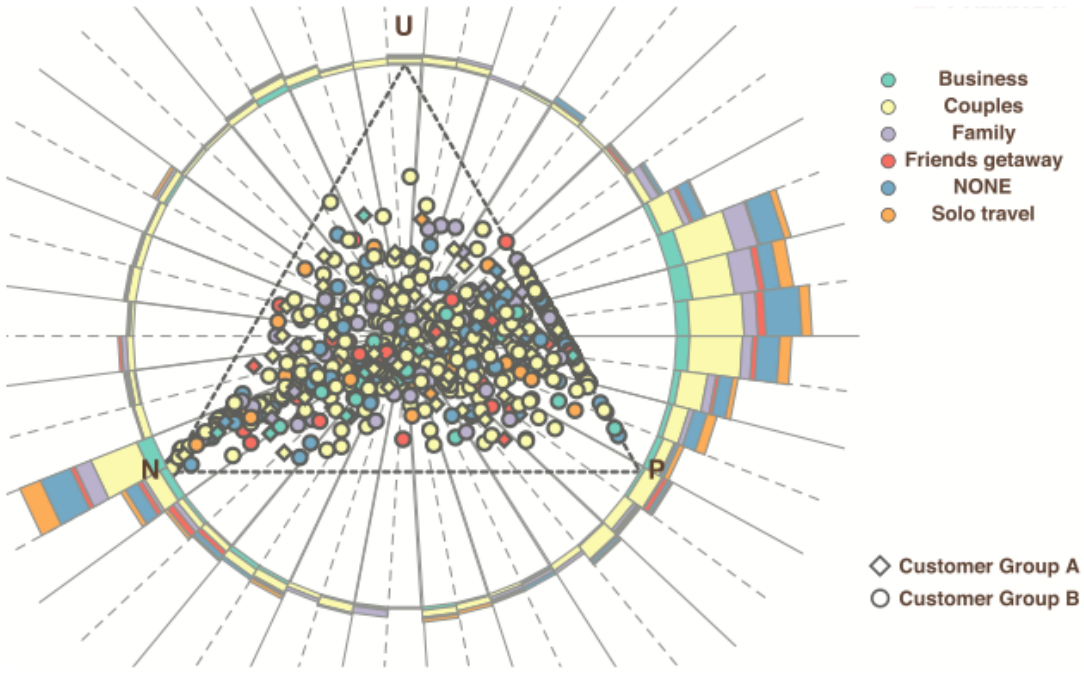


Figure 3.6 OpinionSeer – showing how trip types are related to customer groups where distribution of trip types of the two customer groups are shown by dashed and straight lines.

the distance from each of the vertices (Figure 3.7(a), left). The tag cloud provides a visual summary of customers' reviews for a certain hotel group. For example, the center of Figure 3.7(a) shows two tag clouds based on opinions associated with Regions A and B. Customers from these regions are highlighted in the opinion triangle. In order to display category information, for example age range, radial layouts of colour and size are projected on the circumference of the ring. The color represents different age groups and the size represents the number of customers within that age group (Figure 3.7(b)).

A final visualization tool is OpinionFlow, which is based on information diffusion models modelled with Markov processes (Wu et. al. [Wu14]). This means that the opinion of users at time t depend only on their opinions at time $t-1$. OpinionFlow focuses on visualizing the diffusion of opinions, especially via tweets. The visual analysis system consists of three major components: data pre-processing, diffusion modeling, and interactive visualization. The visual representation includes a user flow layout based on a density map that shows

(a) (b)

Figure 3.7 (a) OpinionSeer – Opinion triangle where two out of 3 selected region's tag clouds of opinion words are shown on the right, (b) Illustration of category information circumscribed to categorical ring.

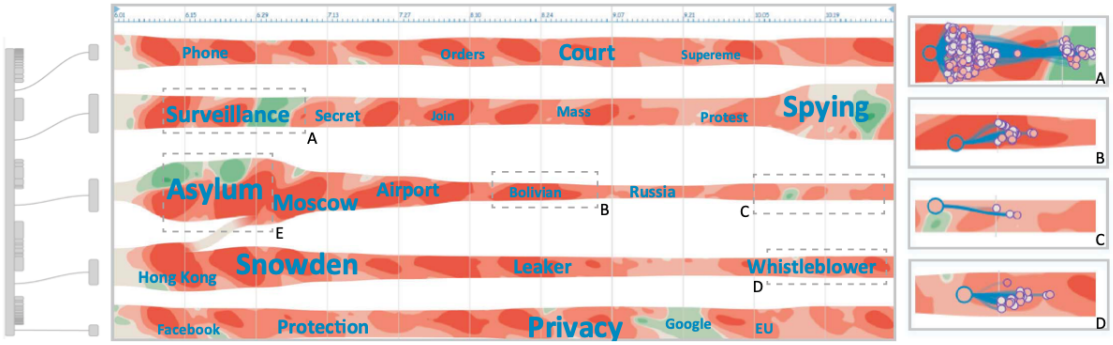


Figure 3.8 OpinionFlow – showing opinion diffusion paths of PRISM data

the diffusion of positive (green) and negative (red) opinions among users, together with user shifts in attention between topics. Each horizontal stream in the layout conveys the temporal variation of the number of users who post tweets on a topic, where the keywords placed on the strip identify a particular topic and their size represents keyword frequency within the topic. Figure 3.8 shows an example of the opinion diffusion of five topics taken from Twitter on PRISM, an electronic surveillance and data mining program from NASA. In the figure, regions A and D show the opinion diffusion paths for Fox News, with strong and weak opinions, respectively. Regions B and C show the opinion diffusion paths for a common user with strong and weak opinions, respectively [Wu14].

Although these analytic systems provide an interactive method to view and summarize users' opinions based on different dimensions, none of them show the exact opinion value

or social relationships between users. Due to the absence of an opinion value, these approaches cannot show how users form opinions based on product properties mentioned in each review, instead aggregating the total number of positive or negative words mentioned in all reviews of a product. Also, the systems do not take into account how users are influenced by their neighbors in a social network like Facebook or Instagram. We plan to address these limitations by visualizing a combination of social relationships and users' opinions in a product feature domain. Although our specific interest is in tracking the diffusion of influence based on feature sentiment over time, this same approach could be applied to other domains with similar needs and data properties.

3.3 Methodology

This section discusses how we plan to approach the problem of predicting product preference ratings based on properties of importance extracted from online reviews. We apply text analytics to determine the sentiment polarity assigned to each property in each review. The end goal is the use of a supervised learning technique to construct a predictive model of preference ratings based on user reviews. These ratings act as a proxy to each user's probability of adopting a product. Finally, we visualize users' opinions towards a product and relationships with their neighbors.

Before proceeding with natural language processing (NLP) and text analytics, we collect publicly available review datasets, in particular of smartphones. We extract the top ten most frequent terms (words) in the reviews that correspond to product properties using named entity recognition (NER). Examples for a smartphone review include display, size, and charger to weight of the smartphone. Once we have the important property list, the dataset is pre-processed in a traditional manner. We then apply NLP techniques to analyze, understand, and extract meaningful insights from the textual data. The specific NLP methods that we include are part-of-speech tagging, NER, and sentiment analysis. This is done

using open-source NLP Python libraries such as spaCy, NLTK, Gensim, and TextBlob.

3.3.1 Natural Language Processing

NLP is an area of artificial intelligence focused on ways to computationally interpret human languages in the form of text. The foundations of NLP lie in a number of disciplines such as linguistic analysis, computer science, mathematics, and artificial intelligence [Man99]. Prior to applying NLP, text is pre-processed to prepare it for analysis. Common pre-processing steps include the following [Nad11].

1. **Tokenization.** Identify terms (words or phrases) within the text corpus. Tokenization is used to decompose a text document or document collection into its constituent entities.

2. **Stop Word Removal.** Common terms often contain no semantically meaningful information, so they are removed prior to NLP. Although stop word lists vary across different software libraries and are often tailored to a domain of interest, typical examples include terms like *the*, *a*, *an*, *always*, *sometimes*, and so on.

3. **Stemming.** Stemming attempts to convert different variations of a term into its common root form. For example, *run*, *runs*, *running*, and *ran* would all be stemmed to the base term *run*. This allows for accurate frequency counting of terms, since different variations are bundled into a single representation. Stemming was initially proposed by Lovins [Lov68]. Currently, one of the most commonly used stemming algorithms is Porter stemming [Por80].

4. **Lemmatization.** Similar to stemming, lemmatization attempts to collapse different forms of the same root term. Unlike stemming, however, which strips the ending of a term, lemmatization considers the morphological context of a term to identify its lemma. For example, the Porter stem of *ponies* is *poni*, but the lemma is *pony*.

Although viewed as more informative than stemming since it considers surrounding term context, lemmatization is also more difficult to implement correctly.

5. **Part of Speech Tagging.** Each term can be associated with a part-of-speech using sentence analysis with algorithms like Hidden Markov Models. Part-of-speech tagging can disambiguate synonyms (e.g., *We broke our fast* versus *I like fast cars*, where *fast* is a noun and an adjective, respectively). It can also improve other techniques like named entity recognition (NER).

After pre-processing, for each property or aspect of a product we apply a rule-based matching technique to identify adjectives, adverbs, and verbs associated with the aspect. Next, we perform sentiment analysis to estimate a score based on a sentiment lexicon dictionary that defines terms as positive or negative. Properties not mentioned in the text are assigned a neutral sentiment.

3.3.2 Aspect-based Sentiment Analysis

To estimate user's preference scores, we need a method to obtain a sentiment score from their review text. Because each of the reviews is based on a product's properties, we employ aspect-based sentiment analysis, where aspects in a given review are classified by sentiment polarity.

For a lexicon-based approach, the sentiment orientation and subjectivity of each term in a sentence usually defines how the sentiment polarity of the term is calculated. The polarity of an aspect determines if it implies a positive or negative sentiment on the range $[-1, 1]$. Subjectivity quantifies the amount of personal opinion and factual information contained in the text and lies on the range $[0, 1]$. A higher subjectivity means the text or data contains more personal opinion than factual information [Naz20]. For example, consider the sentence "I do not like this food item, it is very bland". The sentence has a negative sentiment polarity of -1 for the aspect *food* and a subjectivity of 1, implying that it is a

personal opinion rather than factual information. We incorporate only the polarity score to determine the sentiment of an aspect in a review. This produces a three-valued property list (negative or -1, neutral or 0, and positive or $+1$) for each aspect of importance used in each review.

3.3.3 Predicting Star-rating

Overall user preferences are estimated using the individual aspect sentiments discussed in the previous section. We combine the aspect polarities into a vector that acts as input to a supervised machine learning technique. Supervised machine learning (ML) takes a dataset of data samples that have been labeled with a known, correct category. In our context, a data sample is an aspect vector constructed from review text together with the review's five-star rating acting as its label. This dataset is subdivided into a *training set* and a *test set*. The training set is used as input to a supervised ML algorithm. The algorithm is designed to associate properties in the training set to predict the appropriate target classification. As a practical example, in our investigation we take a dataset of product reviews and construct aspect vectors for each. Assume a review has a five-star rating, producing five possible classifications (labels) for each review. Different supervised ML algorithms use different approaches, but in all cases the result is a model that takes a review's aspect vector as input and returns a predicted category (or star rating, in our application) as a result. Once the model is built, the review text in the test set is used to generate aspect vectors and corresponding estimated preferences (star ratings). Since the test set has also been correctly labeled, we can compare the estimate and the known, correct label to calculate an accuracy for the model. This allows us to apply different supervised ML algorithms to different review datasets to see how their accuracies compare. To this end, we investigate the following supervised ML algorithms: decision trees, random forest, support vector machines (SVMs),

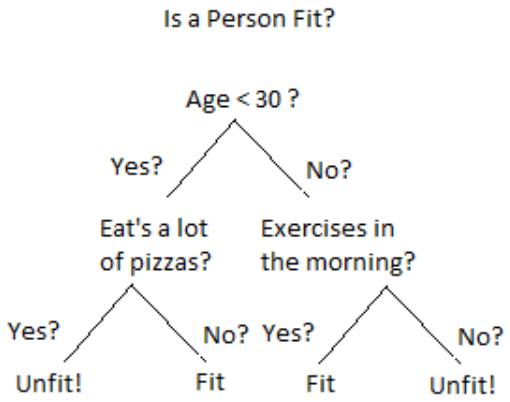


Figure 3.9 A decision tree that classifies a person's level of fitness based on their age, whether they exercise, and whether they eat a lot of pizzas

and k-nearest neighbors (KNN).

Decision Tree. A decision tree classifier is based on a tree structure where internal nodes denote splitting conditions based on specific *decision* properties, edges represent splitting values, and leaf nodes represent subsets of the dataset assigned a specific label. A decision tree is a hierarchical representation where decision nodes are used to make choice decisions and leaf nodes are the output of these decisions [Saf91]. Figure 3.9 represents the concept of a decision tree[1] that shows whether a person is fit or not based on age, physical activity, and eating habits.

The decision tree classification starts at the root node (the entire dataset) and recursively progresses until it reaches leaf nodes with class labels for predicting the output of the samples contained at the leaf [Tan20]. Internally, in order to decide whether to move to the left or right sub-tree, a split condition is applied. The goal is to select a split condition that divides the dataset *homogeneously*. The notion of *impurity* was introduced to mathematically measure a split condition. The condition with the lowest impurity value is chosen [Bre17]. There are different criteria proposed for the split condition such as the GINI index or information gain.

[1]Image Source: Decision Tree Classification

- **GINI index.** The GINI index is employed to determine the quality of a split. A lower GINI index value indicates a higher purity and thus results in a better split of each attribute. If D is a dataset with l different class labels, GINI is calculated as $\text{GINI}(D) = 1 - \sum_{i=1}^{l} p_i^2$, where p_i is the probability that class i is in D [Rai04].

- **Information Gain.** Information gain is the difference between the entropy of a class and the conditional entropy of the class and a selected property. If the information gain has a higher value, it implies the property is more useful for classification [Azh13]. Mathematically, information gain IG is

$$\text{IG}(D, f) = \text{entropy}(D') - \text{entropy}(D', f)$$

  where D' is the dataset after split, f is the property under consideration, and entropy is the sum of the probability of each label times the log probability of that same label.

Random Forest. As the name suggests, a random forest is an ensemble of individual decision trees whose predictions are combined to determine a final outcome. It is based on the technique of random property selection and bagging where multiple trees are generated with replacement [Bre01; Ho95; Ona16]. The term bagging comes from **b**ootstrap **agg**regat**ing** where bootstrapping is a method of randomly creating repeated samples of data (multiple trees) with replacement and then aggregating the results of each tree to predict the final outcome [Bre96].

In random forests, when a training set is created, about one-third of the sample, known as *out-of-bag data*, is withheld. This is used to obtain an estimate of classification error as trees are added to the forest. Unlike a decision tree where the most important property based on the entire training set is identified to split a node, in a random forest the best property is located among the random subset of properties available. Figure 3.10 shows an example of a random forest model predicting a class label based on majority voting.

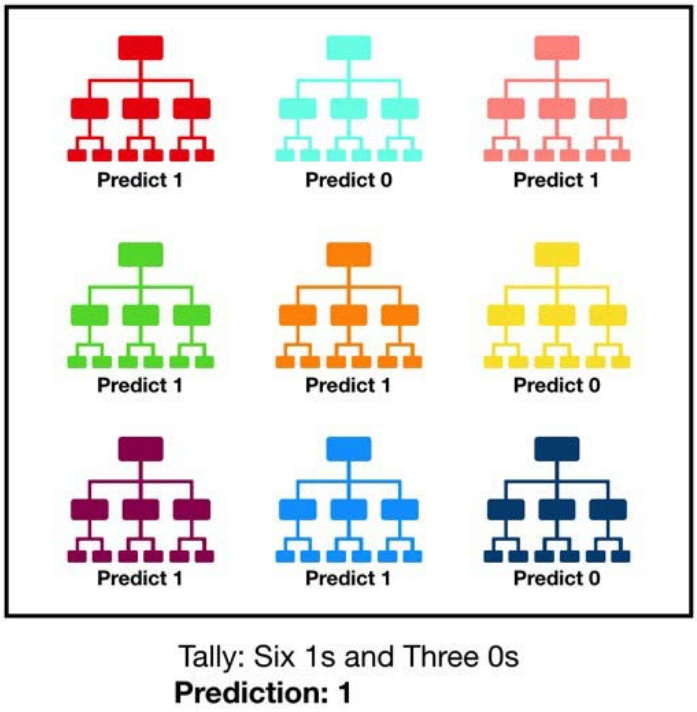


Figure 3.10 Random forest model

The random forest algorithm runs as follows.

1. Select k random samples from a dataset containing N records, $k < N$.
2. Construct individual decision trees for each of the k samples.
3. Generate output from each decision tree.
4. Use majority voting to determine the final outcome (or class label).

SVM. Support vector machines construct a hyperplane in an n-dimensional space, where n is the number of properties that distinctly classify a dataset's data points. There is a possibility of having multiple hyperplanes between any two data points, however, the objective is to maximize the distance between the data points of different classes. Here, hyperplanes act as a decision boundary that classifies the data points [Ton01; Nob06]. If the number of input properties is $n = 2$, then the hyperplane is a line, if it is $n = 3$, then it is

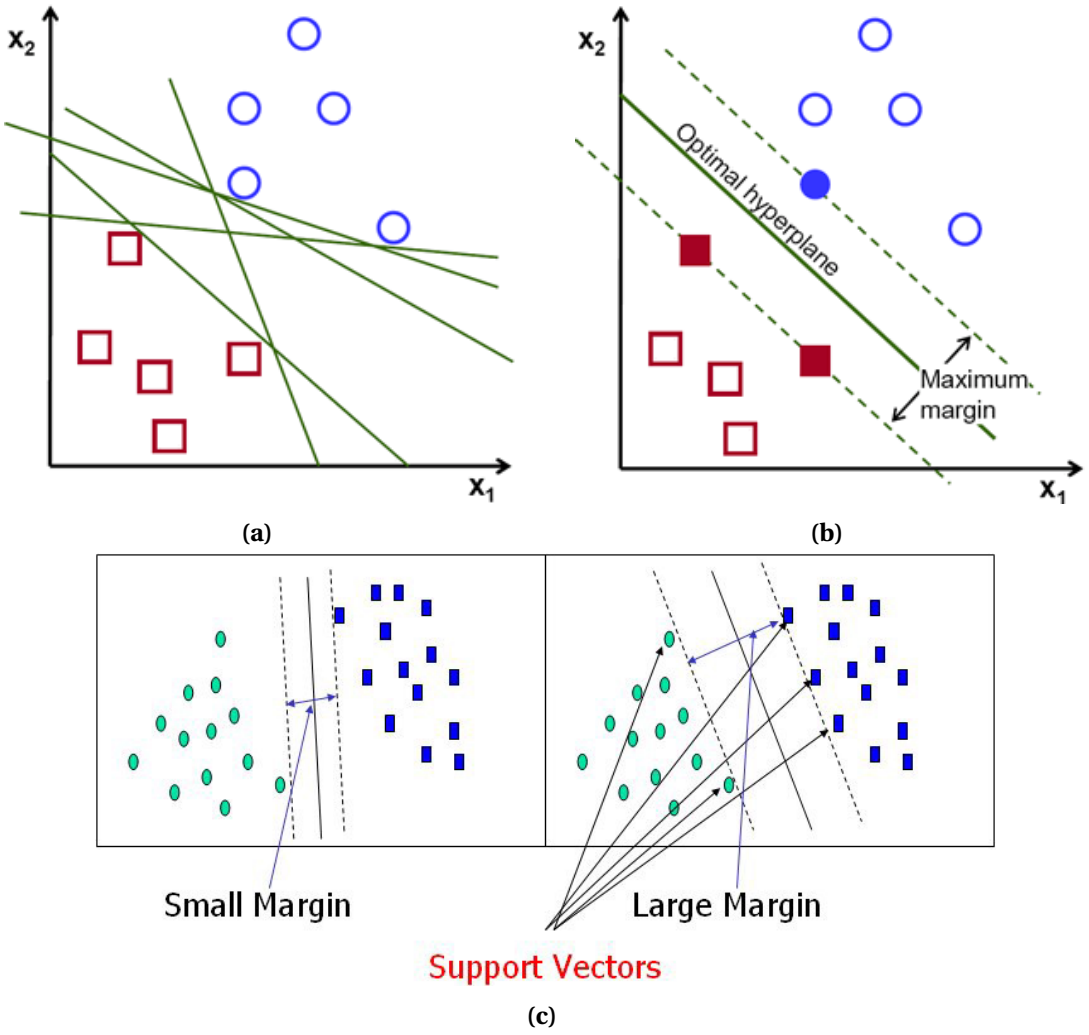


Figure 3.11 Examples of hyperplanes dividing two classes of points in 2D: (a) candidate hyperplanes; (b) maximum margin hyperplane that maximizes distance from data points in each class; (c) the effect of a support vector on a hyperplane relative to its distance from the hyperplane

a two-dimensional plane. For $n > 3$ hyperplanes exist in $(n-1)$-dimensional hyperspace. The goal is to find the hyperplane with a *maximum margin*, or maximum distance from data points of each class (Figure 3.11a–b).

Support vectors are points (n-dimensional vectors) in each class. The closer a point is to a hyperplane, the stronger its influence on that hyperplane. Removing the support vector (point) will influence the hyperplane's position and orientation to maintain a maximum margin (Figure 3.11c[2] [Wan05]). The combination of hyperplanes and support vectors

[2]Image Source: Support Vector Machine—Introduction to Machine Learning Algorithms

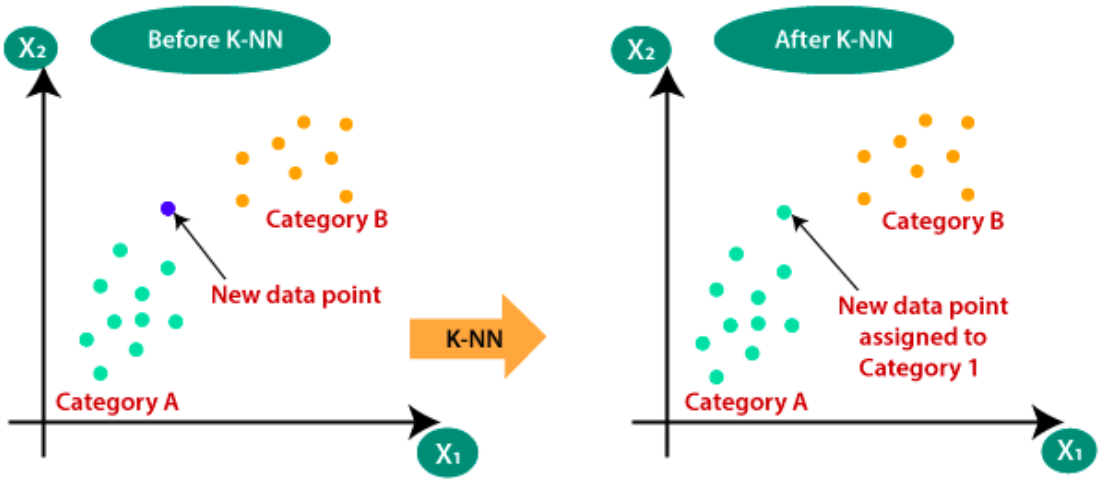


Figure 3.12 Illustration of how KNN algorithm classifies a data point based on its distance to another data point.

produce an SVM model.

KNN. k-nearest neighbors determines the classification of a given data sample based on the categories of the k nearest data points. KNN captures the idea of similarity by calculating the distance between two data points. However, KNN is heavily dependent on k as the choice of k determines the success of classification of the output labels [Guo03]. The output label is predicted based on the most frequent class of the k nearest neighbors. Although there are various ways of calculating pairwise distance, the preferable method is Euclidean distance. One of the important features of this algorithm is that it does not make any assumptions about the underlying data. Figure 3.12 shows how a new data point is classified based on the distance to its nearest neighbors[3].

The KNN algorithm runs as follows [Zha17].

1. Select k as the number of neighbors to a *new* data point.

2. Identify the data point's k nearest neighbors using Euclidean distance.

3. Assign a category to the new data point based on the most frequent category from

[3]Image Source: KNN Algorithm for Machine Learning

the k neighbors.

3.3.4 Evaluation

As we are applying multiple supervised learning methodologies, the next step is to evaluate these models and try to improve classification accuracy. For any classification problem, there are four possible outcomes.

1. **True positive (TP).** The model predicts a point belongs to class A, and the point belongs to class A.

2. **True negative (TN).** The model predicts the point does not belong to class A, and the point does not belong to class A.

3. **False positives (FP).** The model predicts a point belongs to class A, but the point does not belong to class A.

4. **False negatives (FN).** The model predicts a point does not belong to class A, but the point does belong to class A.

These outcomes can be used to evaluate a model using *precision*, *recall*, *accuracy*, and F_1 *score*.

Precision represents the number of points correctly classified as A relative to all points classified as A, that is, precision $= {}^{\mathrm{TP}}/_{\mathrm{TP}+\mathrm{FP}}$. Recall represents the number of points correctly classified as A relative to the total number of points that belong to class A, that is, recall $= {}^{\mathrm{TP}}/_{\mathrm{TP}+\mathrm{FN}}$.

F_1 score is a representation of a classifier's accuracy based on the harmonic mean of precision and recall. $F_1 = 2\left(\mathrm{precision} \cdot \mathrm{recall}/_{\mathrm{precision}+\mathrm{recall}} \right) = {}^{\mathrm{TP}}/_{\mathrm{TP}+{}^{1}/_{2}(\mathrm{FP}+\mathrm{FN})}$. Finally, accuracy is the percentage of points correctly classified, $\sum_{\forall \mathrm{class}_i} {}^{\mathrm{TP}_i}/_{||\mathrm{class}_i||}$.

Based on these, the classification metric that we use is accuracy. As a next step, we incorporate the predicted preference values into the existing bi-SIS epidemic model where

each node represents the probability of a user forming an opinion towards a product.

3.3.5 Visualization

As mentioned earlier, we use sentiment visualization to capture and present insights extracted from the product sentiment analysis of online reviews of two competing products. The foundation for our visualization system is integration of results obtained from sentiment analysis into the bi-SIS epidemic model. Given a bi-SIS model, we produce normalized user preferences from sentiment analysis and present them in the visualization tool as proxies for the probability of a user being infected (i.e., adopting either of the products). Specifically, users' opinion scores are a combination of their estimated preferences combined with their neighbors' influence. In addition to the infection probability, we include estimated preferences for the high importance product properties that we have determined are most likely to influence a user's purchasing decision.

Our visualization design focuses on two areas: graph visualization and multidimensional visualization. Our graph visualization displays the user–neighbor relationships as a node–link graph where nodes represent users and links represent relationships between users. Multidimensional visualization uses different visual features (e.g., colour, size, texture, and so on) to encode information in "feature layers" attached to the nodes and edges. Past work in our research laboratory has identified the strengths and limitations of common features, used both in isolation and in combination with one another [Hea12]. We believe it will be possible to present four dependent variables against an independent variable, which in our case allows us to visualize different results from analyzing each user's reviews relative to the top ten product properties.

Graph Visualization: The first step towards developing the visualization tool requires construction of a node–link graph visualization. Nodes represent users. The node's degree or number of adjacent edges (links) represent the neighbors associated with the user. Links

in the graph encode relationships between different entities such as email sent and received, friendship status, and so on. From a company's perspective, the graph structure identifies influential users (nodes with higher degree) who can more quickly spread information. The goal of the graph visualization is to provide an overview of how users form opinions based on the sentiment extracted from their reviews as well as neighbour influence identified using social networking algorithms. An example of the proposed node–link visualization is shown in Figure 3.13 [Gho04].

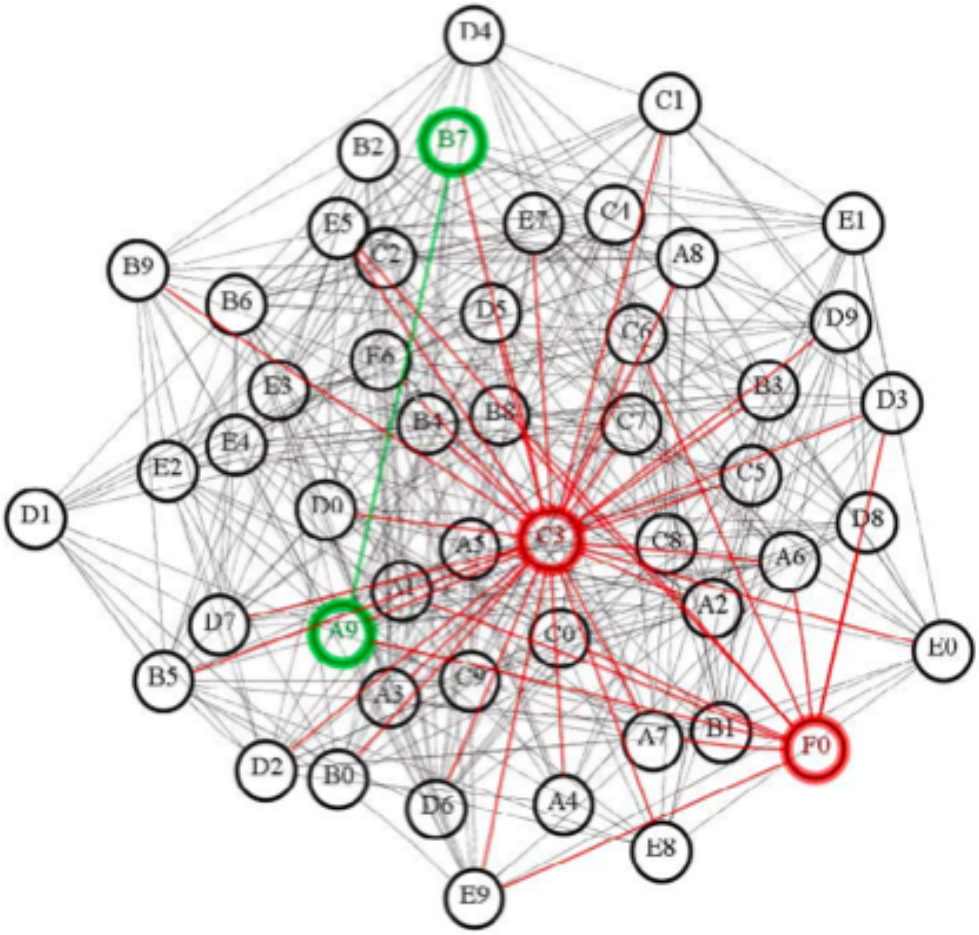


Figure 3.13 Visualization of node-link undirected graph

Graph Visualization Challenges. One of the main problems associated with graph visual-

ization is scalability. As the number of nodes and edges in a graph increases, it becomes difficult to visually explore the graph. To address visual clutter in large graphs, we use the three-step process of visual data exploration known as the Information Seeking Mantra [Shn03]: *overview first, zoom and filter, then details-on-demand*. This highlights nodes (or users) that a company would be most interested in identifying. While providing users the opportunity to visually explore an overview of the data, it should also provide insight to support interacting with the data. Common graph-based interactivity includes mouse over for more detailed information via tooltips, or single and double click to select and manipulate nodes and edges [Kha11]. To address information scale, different strategies have been proposed. Three of the most common techniques are zoom + pan, overview + detail, and focus + context. Zoom + pan allows a user to shift their viewpoint throughout a dataset, zooming in for more detail as needed. Overview + detail initially provides an overview of the dataset. The user can select subsets for more detailed analysis. The subset "zooms in" to replace the overview. A combination of "zoom in" and "zoom out" is used to hierarchically examine the dataset at different levels of detail. Finally, overview + context presents an overview and a mechanism to reveal additional detail at specific areas of user interest within the overview. A common example of this approach is a lens-based system where a magnifying lens can be positioned to increase or decrease the magnification factor to examine data beneath the lens at different levels of detail.

For our purpose, we focus on an overview + detail approach, which provides a combined overview and detail view. The *overview* feature in our scenario provides a complete, compressed form of the user graph along with varying node size to capture the notion of larger nodes for higher probabilities of product adoption. The *detail* aspect zooms in to provide additional detail using the Magic Lens approach [Kei02]. The basic idea is to provide a different sub-view overlaid on the overview using the concept of a magnifying glass. The detail region shows additional information while the remainder of the visualization remains unaffected. This provides context about how the overview and detail regions relate to one

another. The zooming technique not only displays data objects (nodes in our case), but also an automatically varying data representation to present more details at higher zoom levels [Kei02]. For example, we present the nodes as icons at an intermediate zoom level and as labeled objects with information such as opinion scores and important property polarity at a high resolution. We also bundle nodes into clusters with aggregated property values using a metric of user similarity based on a product's properties at the overview level, then allow the lens to decompose clusters into their constituent components.

Multivariate Visualization: Individual nodes and edges in the node–link diagram can display multiple attribute values by treating them as glyphs or icons. Glyphs are graphical entities that use visual features such as size, shape, and color to represent one or more attributes. Figure 3.14 shows a variety of glyphs where graphical properties are controlled by the underlying multivariate data point that the glyph represents. Examples of glyphs and the visual features they can manipulate include the following [War02].

- Profiles: Glyph height and color.
- Stars: Axes radiating from a central origin.
- Anderson/metroglyphs: Rays of varying length radiating from a central origin.
- Stick figures: Limbs and limb joints that vary length, angle, and color.
- Trees: Branching structure with length, thickness, and angles of branches.
- Autoglyph: Colored boxes.
- Boxes: Boxes with varying height, width, and depth.
- Chernoff Faces: Faces that vary the size and position of the eyes, nose, mouth, curvature of the mouth, and angle of the eyebrows.
- Arrows: Arrows that vary their length, width, taper, and color of base and head.

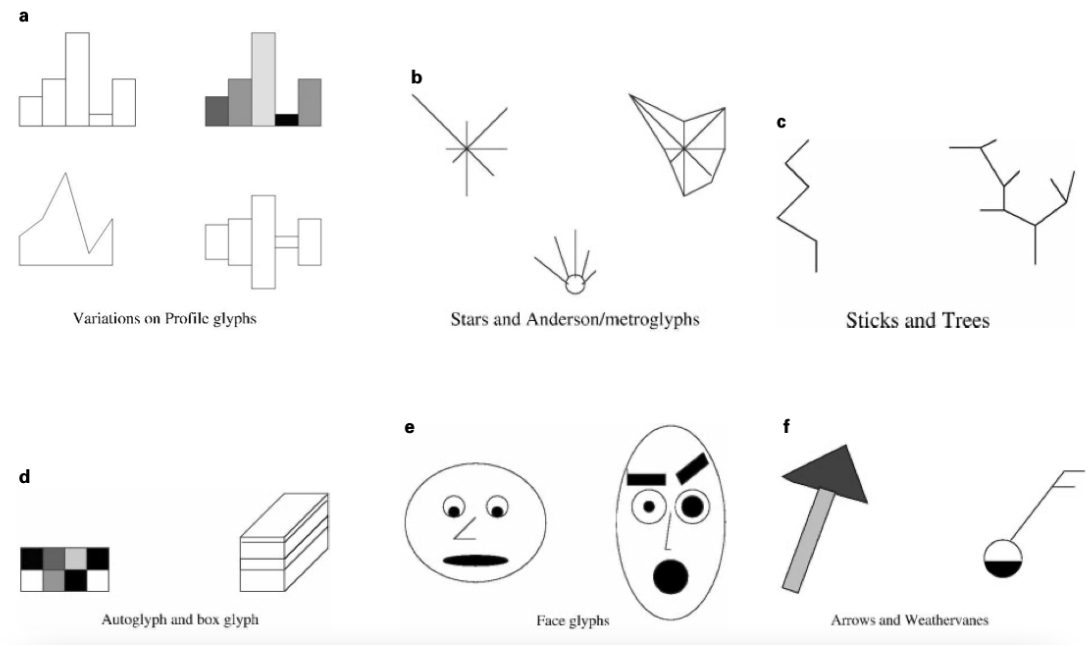


Figure 3.14 Examples of Glyphs

- Weathervanes: Bulbs with varying length and flag representation, a standardized meteorology glyph.

In our domain, glyphs in the form of a scatterplot encodes a user's preference and weight for important product properties. Significant work in visualization has been performed to investigate the capabilities, strengths, and limitations of fundamental visual features when applied to glyphs, both in isolation and in combination [Hea12]. Although glyphs can effectively represent multiple attribute values, there is still a fairly small limit to the number of values that can be visualized simultaneously in an efficient manner. We use different visual variables such as hue, saturation, brightness, size, shape, and so on to indicate how each of the property attributes vary between reviews, as shown in Figure 3.15. These visual channels are ordered based on how accurately they can be perceived: 1) position along a common scale; 2) position along non-aligned scale; 3) length, angle or slope; 4) area; 5) volume or curvature; 6) shading or colour saturation [Cle84].

For the design of our glyphs, we use eight criteria proposed by Chung et. al. [Chu15]:

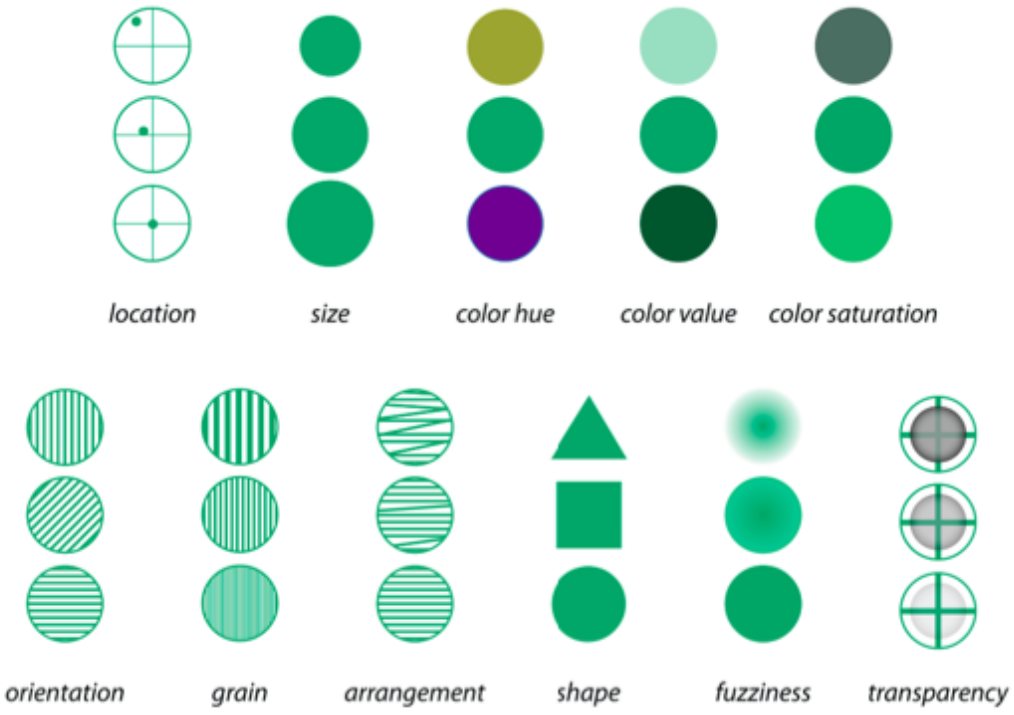


Figure 3.15 Visual Variables

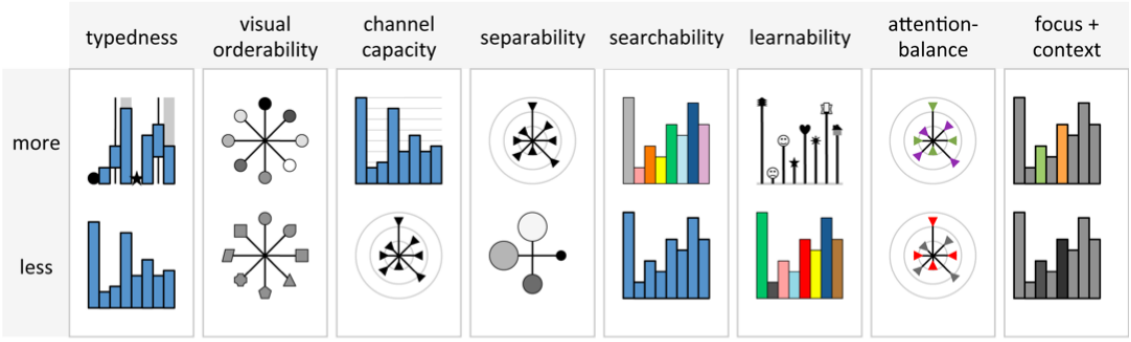


Figure 3.16 Glyph Design Criteria

typedness, visual orderability, channel capacity, separability, searchability, learnability, attention balance, and focus and context, as shown in Figure 3.16. Based on this understanding of product properties and their preference weights through an interactive visualization tool, analysts can explore the attributes of each product and the relationships between users to better understand how influence is spread and what the buying probabilities of different users might be.

By taking all these details into account, and with the use of interactive filtering, interactive zooming, and multivariate glyphs, analysts can capture the notion of social network properties like cliques (clusters), high degree nodes, or influential nodes. The identification of cliques can also be related to the creation of a community, as seen in our bi-SIS epidemic modeling framework. Through these exploration techniques, we aim to provide the opportunity for users to obtain a detailed overview of their data by improving insight quality and speed of acquisition through a visual analytics process.

3.4 Visualization Dashboard

As a next step towards having a visual medium to better analyze each user's opinion value for the product, we develop and design a visualization dashboard. This dashboard uses the node–link diagram as our visualization technique to represent the user–user relationship along with a set of additional functions to aid the visualization process. We discuss the details of the aspects/layout used in the design process of the dashboard in the following sections.

3.4.1 Dashboard Layout

We divide the dashboard layout into 3 rows: (i) the first row shows the user–user relationship as a node–link diagram, (ii) the second row provides a slider for the viewer to select reviews for a specific time range, and (iii) we provide drop-down menus to select any particular user's review along with the ability to view clusters of users based on a product's properties included in the users' reviews.

We have designed the above-mentioned layout to provide a holistic view that includes an overview of all users' behavior, together with mechanisms to analyze individual users or clusters of users. We aim to make the visualization process intuitive so that viewers can navigate the dashboard easily while identifying all the information relevant to their

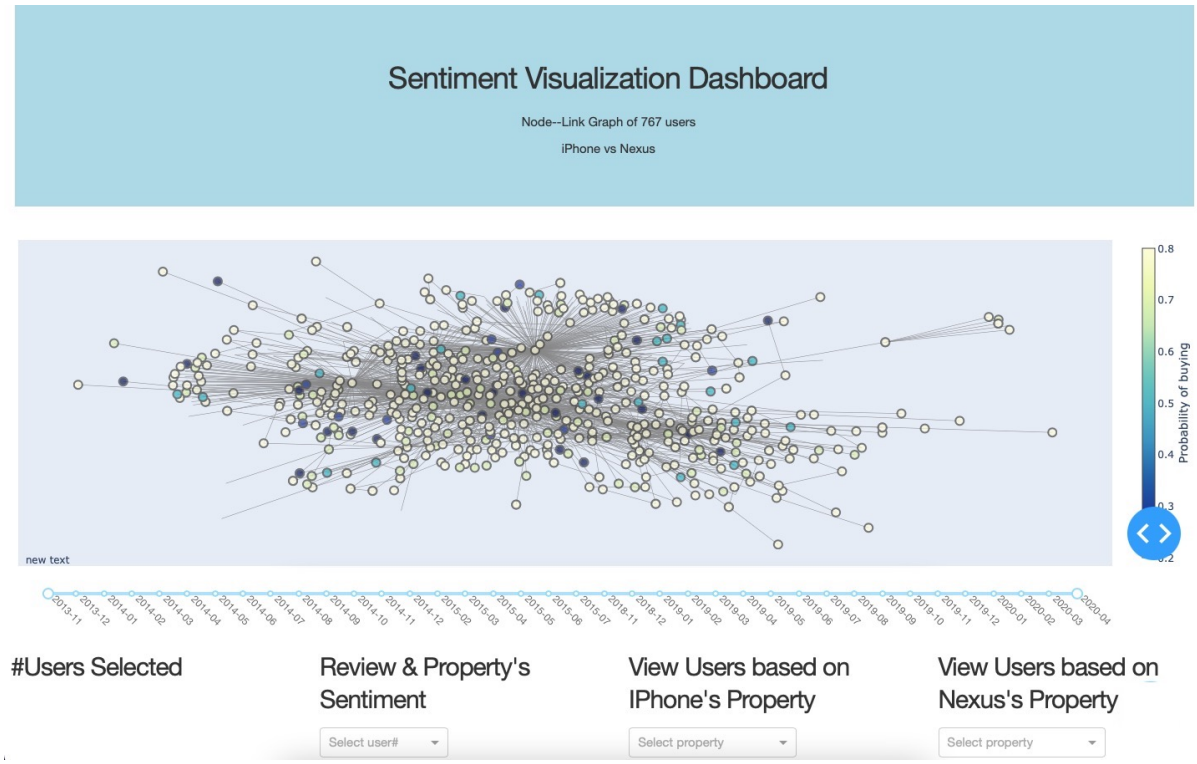
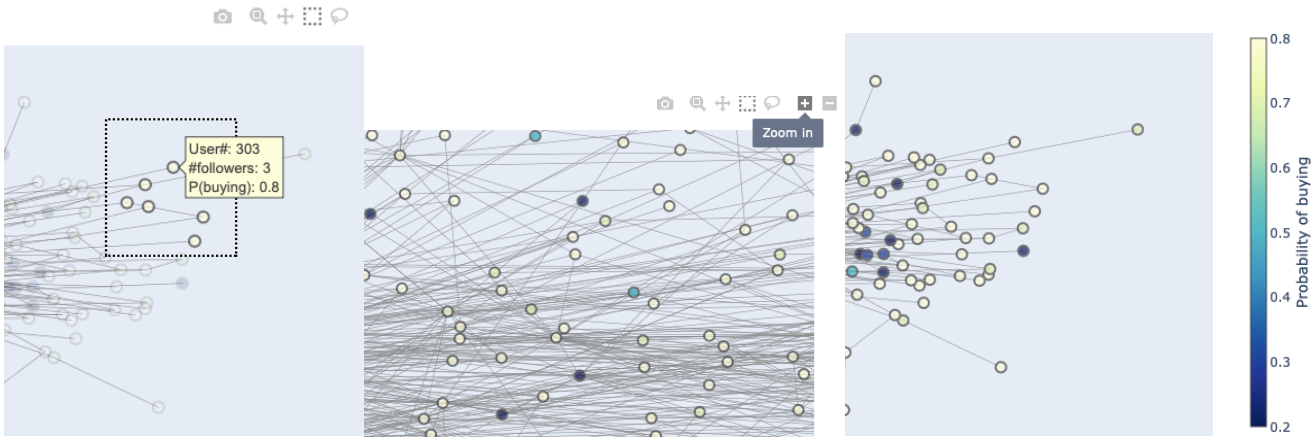


Figure 3.17 Visualization Dashboard



(a) Box-select feature **(b)** Zoom feature **(c)** Coloring Scheme

Figure 3.18 Summarized view of each of the basic functionalities incorporated in the dashboard.

investigations.

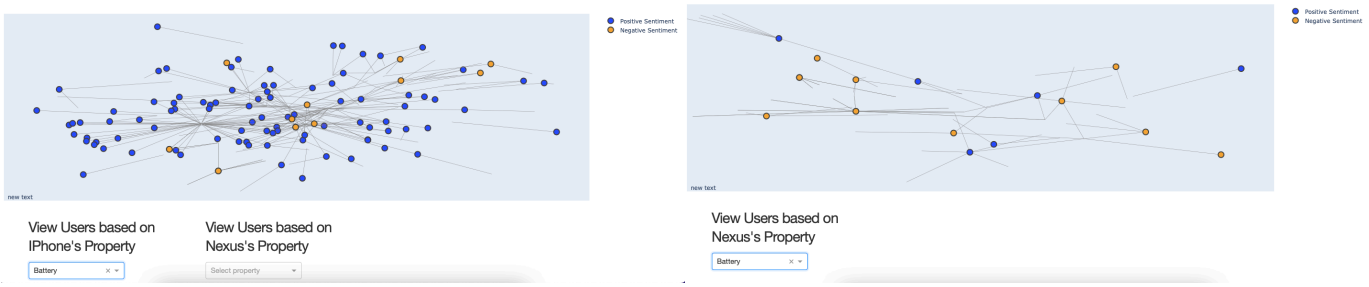
Node–link Diagrams: Although there are different visualization techniques to represent graphs/social networks, node–link diagrams depict the relationship between users more

naturally and intuitively. In such diagrams, nodes in the form of circles represent users and links/edges in the form of line segments represent relationships between the users. Figure 3.17 shows an example of a node–link diagram of 767 users depicting their social relationship of follower–followee.

While node–link diagrams are easy to analyze and understand, they become cluttered as the number of nodes and edges increases. To address this issue, we have included tool-tips and inspection capabilities in the dashboard to give more detailed information about each of the nodes' behaviors. For example, the zoom-in and box-select feature enable viewers to magnify and focus on specific areas based on the viewer's current needs and interests. The box-select feature displays detailed information about a user such as their name, their review text, and whether the properties mentioned in the review were described in a positive, negative, or neutral way (Figure 3.18(a)). Along with this functionality, hovering over a node in the graph displays the user's name, number of followers, and opinion (as a probability value) of the user towards the product. Figure 3.18(b) shows the zoom feature along with the hovering functionality.

Another important aspect of the graph visualization is the coloring scheme, which allows observation of user's opinion values while guiding the user's attention to specific regions. While displaying the entire graph becomes less effective for large networks, we use a color grading scheme to denote the variation of opinion among users. The closer the opinion score is to 1, the darker the color, and the closer it is to 0, the lighter the color, as shown in Figure 3.18(c).

Clustering: Although our dashboard provides detailed information related to a user through the box-select feature, it still lacks the capability to view users based on product properties. We capture this by providing the viewers a drop-down menu to select any property of interest. Upon selecting a property, the dashboard displays a cluster of users (in the form of a node–link diagram) that have reviewed the property either positively or negatively. For example, Figure 3.19 shows a comparison between the users who have



(a) iPhone **(b)** Nexus

Figure 3.19 Graph showing cluster of users who have reviewed positively and negatively about iPhone's and Nexus's 'Battery'.

reviewed iPhone's and Nexus's battery property. Here, blue represents positive sentiment and orange represents negative sentiment towards the display. A viewer can hover over users to examine their name, number of followers, and opinion value towards the selected property. Through this color-coded sentiment differentiation, it is easier to identify and navigate even when the number of users increases. The cluster of nodes based on the product's property makes it possible to obtain a snapshot of how many users have posted positively or negatively about a product property in a single glance. It aids the visualization process by making it more visually appealing and clutter-free.

Slider: A timeline slider allows a viewer to navigate through different periods of time to select reviews and other related information based on time frame. For example, the slider can help to view reviews posted over a specific year or range of years. This also helps reduce visual clutter as only certain users' information will be shown in the graph over any period of time. A viewer can still select using the box-select feature to expose clusters of users based on product properties. Figure 3.20 shows an example of reviews posted during a subset of months in a year together with the information on the selected number of users. Note that for our scenario we have varied year numbers in the slider, but this can be modified to represent any data attribute per the viewer's requirement.

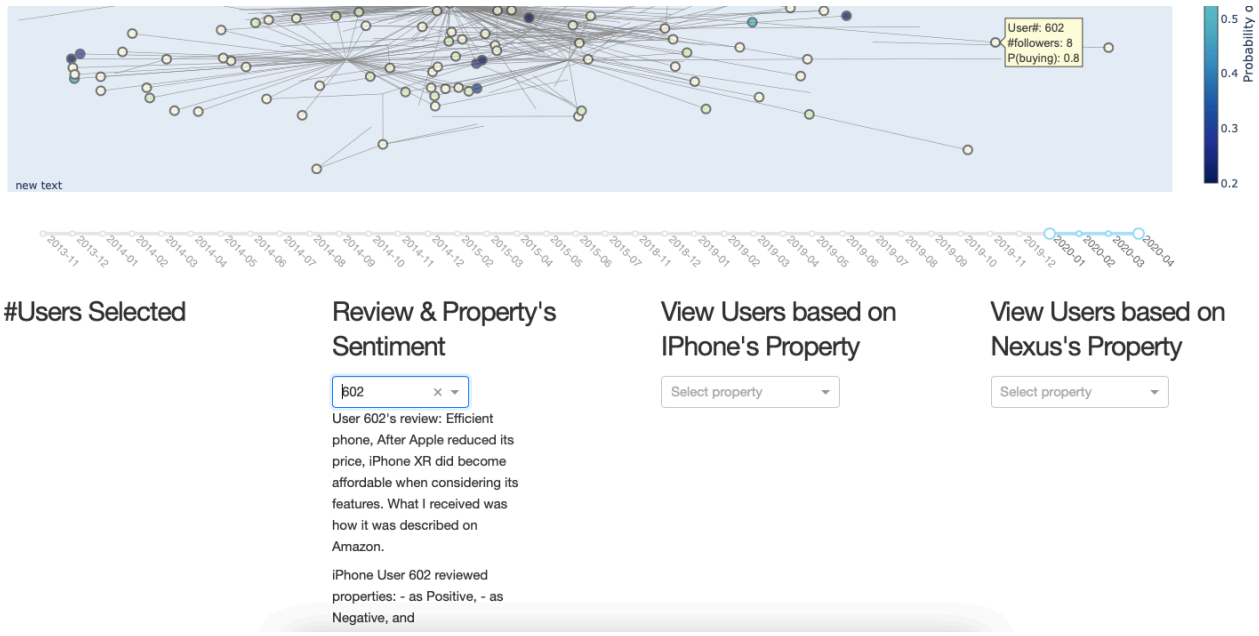


Figure 3.20 Timeline slider showing the number of users who posted reviews for the period of Jan 2020 – April 2020.

3.5 Conclusion

In order to more accurately represent individuals' product adoption behavior in the bi-SIS social network graph, we applied NLP and supervised machine learning to online reviews about smartphones. NLP identified important smartphone properties and for a given user's review assigned a sentiment polarity to each property based on how it was described in the review text. The polarities form an aspect polarity vector. The set of vectors for all reviews are used as input to a supervised ML algorithm, producing a model that estimates the five-star review rating based on a user's stated sentiments about a product in their review text. The rating and aspect sentiments augment a user's social network graph node with estimated product property preferences, a more accurate weight than the typically assumed constant weights used in traditional bi-SIS. Results showed that decision trees, random forests, and SVMs produced models with comparable performance.

As a next step, the predicted star-rating is normalized to denote the probability with

which a user is willing to adopt a product. We use these values to develop a visualization dashboard that provides viewers the ability to examine user reviews. The dashboard highlights users who have reviewed product properties positively or negatively through variation of hue and brightness. Additional functionalities provided in the dashboard include selecting reviews based on a specific time period, selecting users of interest, zooming in and out of the node–link graph, and hovering over a user's node to query more detailed information about the user. Based on these dashboard features/functionalities, we aim to provide a visual interface to the users and their reviews for analysis and comparison of a product's properties prior to making a buying decision.

CHAPTER 4

Integration to bi-SIS Model

4.1 Introduction

A consumer's product adoption behavior is a combination of multiple factors such as interaction with their friends/network neighbors, market trends, online reviews, and the consumer's opinion about the product, to name a few. Although for a potential buyer these factors (and many more) contribute to making an informed buying decision, we focus on two important aspects: influence from neighbors and star-rating as seen in online reviews. To better describe the process of the spread of product adoption behavior, we provide an integrated model which includes bi-SIS epidemic spreading parameters and star-rating as an indicator of the probability of buying a product based on properties described in the reviews. This approach provides an improved description of product adoption by addressing the issue of how and by what factors neighbors influence a potential buyer of a product. By focusing on how a certain product's properties described in reviews determine the opinion a user has about a product and how it helps in the spread of the product in the network, we

aim to bridge the gap between the endogenous influence parameter used in a SIS epidemic model and how users actually form opinions from reviews.

We use a modified bi-SIS model that incorporates the star-rating provided by a potential buyer's neighbor into the bi-SIS model. Specifically, we focus on how incorporating the star-rating from reviews into the model better captures neighbors' influences on potential buyers versus the original exogenous influence parameter from the bi-SIS model. Even though our proposed model reflects how user opinions from reviews influence product adoption behavior in a network, the unavailability of a public dataset that has both information regarding social networks and reviews makes it difficult to validate our proposed model and other theoretical results. Hence, we resort to numerical simulation where the probability values or star-rating for each review from the supervised machine learning model is provided as an exogenous constant parameter that does not change over time. We simulate using real-world graphs for two scenarios where, in one case, all the users provide reviews about $\mathbf{P}_1$ and $\mathbf{P}_2$, and in the other, only a few users provide reviews. The simulation results show that two products co-exist in both scenarios. Although the model has the limitation of not being completely accurate (as it assumes reviews do not change over time) in terms of how users form opinions and what factors contribute to making a purchasing decision, we hope the integration of a star-rating is one step closer to better understanding how product properties impact the opinion formation process and thereby, the spread of product adoption in a network of individuals.

4.2 Model

Based on the bi-SIS model, we incorporate the additional star-rating factor as provided by a user in a review, which is equivalent to the normalized value of the probability with which each user i owns one of the competing products. The following ODEs denote the system dynamics when star-rating is integrated into the bi-SIS model:

$$\dot{x}_i = \beta_1(1-x_i-y_i)\sum_{j\in\mathcal{N}} a_{ij}x_j + \beta_1(1-x_i-y_i)\sum_{j\in\mathcal{N}} a_{ij}\mathbb{1}_{j\in\mathcal{R}_1} r_{xj}x_j - \delta_1 x_i \tag{4.1}$$

$$\dot{y}_i = \beta_2(1-x_i-y_i)\sum_{j\in\mathcal{N}} b_{ij}y_j + \beta_2(1-x_i-y_i)\sum_{j\in\mathcal{N}} b_{ij}\mathbb{1}_{j\in\mathcal{R}_2} r_{yj}y_j - \delta_2 y_i \tag{4.2}$$

Under this setting, at any time $t \geq 0$ a user adopting $\mathbf{P}_1$ ($\mathbf{P}_2$) influences their susceptible neighbors (potential buyers) at rate β_1 (β_2) and they recover (i.e., disown the product) with rate δ_1 (δ_2). The effective adoption rate then becomes $\tau_1 = \beta_1/\delta_1$ ($\tau_2 = \beta_2/\delta_2$) for $\mathbf{P}_1$ ($\mathbf{P}_2$). Let $x_i(y_i)$ represent the probability that user $i \in \mathcal{N}$ adopts $\mathbf{P}_1$ ($\mathbf{P}_2$) at time $t \geq 0$. Note that a user cannot adopt both products at the same time. The term $r_{xj} > 0$ ($r_{yj} > 0$) denotes the star-rating provided by the user j regarding $\mathbf{P}_1$ ($\mathbf{P}_2$) which is equivalent to the probability that user j owns $\mathbf{P}_1$ ($\mathbf{P}_2$) and has the corresponding opinion regarding it. The indicator function $\mathbb{1}_{j\in\mathcal{R}_1}$ is 1, where $\mathcal{R}_1 \subset \mathcal{N}$, if user j has posted a review about $\mathbf{P}_1$ otherwise 0 (Similarly for $\mathbf{P}_2$ where $\mathcal{R}_2 \subset \mathcal{N}$). When a user does not provide any review, the system of equations is equivalent to the original differential equation of bi-SIS.

By rearranging the terms in 4.1 and 4.2, it yields

$$\dot{x}_i = \beta_1(1-x_i-y_i)\sum_{j\in\mathbb{N}} a_{ij}\left[1+\mathbb{1}_{j\in\mathcal{R}_1} r_{xj}\right]x_j - \delta_1 x_i \tag{4.3}$$

$$\dot{y}_i = \beta_2(1-x_i-y_i)\sum_{j\in\mathbb{N}} b_{ij}\left[1+\mathbb{1}_{j\in\mathcal{R}_2} r_{yj}\right]y_j - \delta_2 y_i. \tag{4.4}$$

For the theoretical analysis, we focus on the scenario when $R_1 = R_2 = N$. The integrated bi-SIS model (Eqns. 4.3 and Eqns. 4.4) yields the following equilibrium equations when $\dot{x}_i = 0$ ($\dot{y}_i = 0$):

$$\frac{x_i^*}{1-x_i^*-y_i^*} = \tau_1 \sum_j a_{ij}\big[1+r_{xj}\big]x_j^* \tag{4.5}$$

$$\frac{y_i^*}{1-x_i^*-y_i^*} = \tau_2 \sum_j b_{ij}\big[1+r_{yj}\big]y_j^*, \tag{4.6}$$

for $i \in 1 \ldots N$ where x_i^* and y_i^* are, respectively, $\mathbf{P}_1$ and $\mathbf{P}_2$ equilibrium adoption probabilities of user i. When $\tau_1 > 1/\lambda(\mathbf{A})$ and $\tau_2 > 1/\lambda(\mathbf{B})$, the above equilibrium equations suggests that the competing spreading model has at least one of the following three equilibrium points:

1. None of the products are adopted, i.e., all the users own neither of the two products ($x_i^* = y_i^* = 0 \; \forall i \in N$).

2. $\mathbf{P}_2$'s absolute dominance, i.e., all the users own only $\mathbf{P}_2$ ($x_i^* = 0$, $y_i^* > 0 \; \forall i \in N$).

3. $\mathbf{P}_1$'s absolute dominance, i.e., all the users own only $\mathbf{P}_1$ ($x_i^* > 0$, $y_i^* = 0 \; \forall i \in N$).

The co-existence equilibrium is given when the equilibrium points (2) and (3) are unstable and then it gives rise to a fourth equilibrium point where $x_i^* > 0, y_i^* > 0 \; \forall i \in N$. Given τ_2, the $\mathbf{P}_1$ survival threshold is the critical value where such a coexistence equilibrium emerges. This corresponds to the threshold value τ_{1c}, $x_i^*|_{\tau_1=\tau_{1c}} = 0$ and $\frac{dx_i^*}{d\tau_1}|_{\tau_1=\tau_{1c}} > 0$ for all $i \in N$ [Sah14]. Taking the derivative of equilibrium equation 4.5 with respect to τ_1 gives,

$$\frac{\frac{dx_i^*}{d\tau_1}(1-y_i^*) + x_i^*\frac{dy_i^*}{d\tau_1}}{(1-x_i^*-y_i^*)^2} = \tau_1 \sum a_{ij}[1+r_{xj}]\frac{dx_j^*}{d\tau_1} + \sum a_{ij}[1+r_{xj}]x_j^* + \sum a_{ij}\frac{dr_{xj}}{d\tau_1}x_j^*$$

At the survival threshold, $\tau_1 = \tau_{1c}$, $x_i^* = 0$. Substituting these values in the above equation yields,

$$\frac{1}{1-y_i^*}\frac{dx_i^*}{d\tau_1}\bigg|_{\tau_1=\tau_{1c}} = \tau_{1c} \sum a_{ij}[1+r_{xj}]\frac{dx_j^*}{d\tau_1}\bigg|_{\tau_1=\tau_{1c}}$$

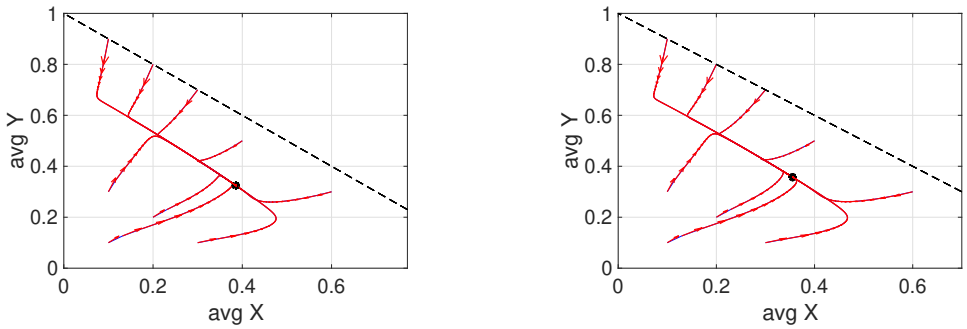


Figure 4.1 Coexistence of two products based on whether (a) all or (b) some percentage of users post reviews when $\tau_1\lambda(\mathbf{S_{y^*}AD_{rx}})>1$, $\tau_2\lambda(\mathbf{S_{x^*}BD_{ry}})>1$.

Rewriting the above equation, we get

$$\left.\frac{dx_i^*}{d\tau_1}\right|_{\tau_1=\tau_{1c}} = \tau_{1c}(1-y_i^*)\sum a_{ij}[1+r_{xj}]\left.\frac{dx_j^*}{d\tau_1}\right|_{\tau_1=\tau_{1c}},$$

where we find that the survival threshold τ_{1c} is the value for which a nontrivial solution exists for $\frac{dx_i^*}{d\tau_1}|_{\tau_1=\tau_{1c}}>0$ and the survival threshold is given by

$$\tau_{1c} = \frac{1}{\lambda(\mathbf{S_{y^*}AD_{rx}})},$$

where $\mathbf{S_{y^*}} = \text{diag}(1-\mathbf{y}^*)$ and $\mathbf{D_{rx}} = \text{diag}(1+r_x)$. Similarly for τ_{2c}.

4.3 Coexistence Equilibrium Numerical Results

We consider an undirected, connected graph (103 nodes, 239 edges) called Autonomous System (AS-733) from the SNAP repository [Les14]. We generated two additional graphs overlaid on the same set of nodes by modifying the original graph (AS-733-A with $\lambda(\mathbf{A})=$ 12.16), removing and adding edges while ensuring connectivity between the nodes. The graph AS-733-B has 741 edges with $\lambda(\mathbf{B})=15.5$. Table 4.1 summarizes the parameter values that satisfy coexistence of two products with a fixed point.

To visualize our system in two dimensions, we use $\bar{X} \triangleq (1/N)\sum_{i\in\mathcal{N}} x_i$ on the x-axis (denoted as avg X in Figure 4.1), and $\bar{Y} \triangleq (1/N)\sum_{i\in\mathcal{N}} y_i$ on the y-axis (denoted as avg

Table 4.1 Simulation Parameters

τ_1	τ_2	$\lambda(\mathbf{S_{y^*}AD_{rx}})$	$\lambda(\mathbf{S_{x^*}BD_{ry}})$
0.5	0.21	2.78	5.77

Y in Figure 4.1). We plot trajectories of the bi-SIS system starting from different initial points in the state space D to observe their convergence, with red arrows representing the trajectories' direction of movement at various time intervals. Here, the state space D is the region that lies below the dotted-line (in Figure 4.1), ensuring $x_i + y_i \leq 1$ for all $i \in \mathcal{N}$, for every initial point. Figure 4.1(a) shows the convergence to coexistence equilibrium when each of the products are competing over graphs AS-733-A and AS-733-B, respectively, where the eigenvalues $\lambda(\mathbf{A})$ and $\lambda(\mathbf{B})$ are significantly different and all the users have provided reviews. We also demonstrate the case where some percentage (20%) of $\mathbf{P}_1$ users do not provide any reviews, as seen in reality. Even in this scenario, the simulation results show convergence to coexistence equilibrium as shown in Figure 4.1(b), under the same value of τ_1 and τ_2 as before.

CHAPTER 5

Conclusion & Future Work

In this dissertation, we focus on the impact of adaptive behavior of competing products within a social network. Although our main area of interest has been understanding product adoption behavior, the same dynamics can also be applied to the analysis of the spread of misinformation or opinion formation regarding an idea/event/entity. With the growing demands for marketing strategies to attract customers and the use of influencers on social networks for product promotion, understanding how a new product with a fixed budget can be launched and adopted more rapidly or what properties of a product help in its adoption/improvement is crucial. Specifically, we look at two aspects of product adoption behavior. First, we propose techniques to quantify and improve the market share of a new product when it competes with another dominant product in the market. We do so by formulating a budget structure by which advertisements or promotional offers can be targeted to a small user group (community) who can then spread positive reviews within the network. Second, we analyze, implement, and design a visualization dashboard based on a product's properties being discussed in reviews posted on e-commerce websites. We aim to

make viewer's product buying decisions easier by using the dashboard rather than manually reading text in the form of reviews. Through a visual interface, it is more efficient and more effective to navigate, compare and contrast a product's features, star-rating (indicative of user's product adoption behavior), and how users perceive each of the features of the product (positive, negative or neutral).

In Chapter 2, we characterize how and under what budget structure an optimal set of users can form a community who can further propagate product adoption via word-of-mouth within a social network. Our heuristic algorithm is based on an iterated local search that identifies the set of users who can be influenced through advertisements. In this way a new or weaker product can better position itself in the market with a positive market share. Finally, by building our theoretical analysis on the bi-SIS epidemic model and conducting extensive simulations on different real-world graphs, we demonstrate the efficacy of our algorithm, including its improved performance over the standard centrality measures of a network. In Chapter 3, we focus on extracting properties/features of a product from online reviews. We apply sentiment analysis through text analysis and natural language processing techniques to assign a $1, -1$ or 0 sentiment score to the top eight properties of a product. These properties are identified based on how the product is discussed in each review. To predict a star-rating from this three-valued sentiment polarity vector we apply supervised machine learning algorithms. We then use the star-rating prediction model to design a visualization dashboard based on a node-link graph and additional associated functionalities to help viewers make an informed buying decision. The types of functionalities include an overview + detail view, a node-link graph to examine clusters of nodes based on specific properties, and a timeline slider to view user reviews over a specific time period. Finally, in Chapter 4, we propose an integrated bi-SIS model which characterizes how both neighbors' influence and star-rating provided by a user promotes product adoption behavior in a social network.

Currently, the integrated model assumes fixed exogenous infection rates and star-ratings

for analyzing system dynamics. It does not take into account how a user's opinion (influencing other users or posting online reviews) changes dynamically over time as we apply the parameters obtained over a specific snapshot of time. Also, due to the unavailability of a public dataset that contains both the user's social connection information and text reviews, it is difficult to analyze and apply integrated parameters to our model. As future work, it would be interesting to endogenize the model parameters and then analyze how the system dynamics change in a competitive product adoption behavior setting. From the visualization interface perspective, it would also be interesting to extend and implement dynamic updates of how users' opinions regarding a product's properties change over time for multiple products and in different domains/categories.

REFERENCES

[Alm19] Almjawel, A. et al. “Sentiment analysis and visualization of amazon books’ reviews”. *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. 2019, pp. 1–6.

[Alz18] Alzubi, J. et al. “Machine learning from theory to algorithms: an overview”. *Journal of Physics: Conference series*. Vol. 1142. 1. 2018, p. 012012.

[Ame17] Amelkin, V. et al. “Polar Opinion Dynamics in Social Networks”. Vol. 62. 11. 2017, pp. 5650–5665.

[Ana20] Anagnostopoulos, A. et al. “Biased opinion dynamics: When the devil is in the details”. 2020.

[And92] Anderson, R. M. et al. *Infectious diseases of humans: dynamics and control*. 1992.

[Azh13] Azhagusundari, B, Thanamani, A. S., et al. “Feature selection based on information gain”. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)* **2**.2 (2013), pp. 18–21.

[Bai75] Bailey, N. T. et al. *The mathematical theory of infectious diseases and its applications*. 1975.

[Beu12] Beutel, A. et al. “Interacting viruses in networks: can both survive?” *ACM SIGKDD*. 2012.

[Bre01] Breiman, L. “Random Forests”. *Machine Learning* **45**.1 (2001), pp. 5–32.

[Bre96] Breiman, L. “Bagging predictors”. *Machine learning* **24**.2 (1996), pp. 123–140.

[Bre17] Breiman, L. et al. *Classification and regression trees*. 2017.

[Car83] Carbonell, J. G. et al. “An overview of machine learning”. *Machine learning* (1983), pp. 3–23.

[CC96] Castillo-Chavez, C. et al. “Competitive exclusion in gonorrhea models and other sexually transmitted diseases”. Vol. 56. 2. 1996, pp. 494–508.

[CC99] Castillo-Chavez, C. et al. “Competitive exclusion and coexistence of multiple strains in an SIS STD model”. Vol. 59. 5. 1999, pp. 1790–1811.

[Che10] Chen, W. et al. “Scalable influence maximization in social networks under the linear threshold model”. *IEEE international conference on data mining*. 2010, pp. 88–97.

[Chu15] Chung, D. H. et al. “Glyph sorting: Interactive visualization for multi-dimensional data”. *Information Visualization* **14**.1 (2015), pp. 76–90.

[Cle84] Cleveland, W. S. & McGill, R. “Graphical perception: Theory, experimentation, and application to the development of graphical methods”. *Journal of the American statistical association* **79**.387 (1984), pp. 531–554.

[Coo87] Cooper, R. G. & Kleinschmidt, E. J. “New products: what separates winners from losers?” Vol. 4. 3. 1987, pp. 169–184.

[Dal94] Dalgic, T. & Leeuw, M. “Niche marketing revisited: concept, applications and some European cases”. 1994.

[Dan13] Dandekar, P. et al. “Biased assimilation, homophily, and the dynamics of polarization”. Vol. 110. 15. 2013, pp. 5791–5796.

[Dos21] Doshi, V. et al. “Competing Epidemics on Graphs–Global Convergence and Coexistence”. *IEEE INFOCOM*. 2021.

[Fan15] Fang, X. & Zhan, J. “Sentiment analysis using product review data”. *Journal of Big Data* **2**.1 (2015), pp. 1–14.

[Fre78] Freeman, L. C. “Centrality in social networks conceptual clarification”. *Social networks* **1**.3 (1978), pp. 215–239.

[FJ56] French Jr, J. R. “A formal theory of social power”. *Psychological review* **63**.3 (1956), p. 181.

[Fri91] Friedkin, N. E. “Theoretical foundations for centrality measures”. *American journal of Sociology* **96**.6 (1991), pp. 1478–1504.

[Fuc10] Fuchs, C. & Diamantopoulos, A. “Evaluating the effectiveness of brand-positioning strategies from a consumer perspective”. 2010.

[Gal02] Gallagher, S. & Park, S. H. “Innovation and competition in standard-based industries: A historical analysis of the US home video game market”. *IEEE transactions on engineering management* **49**.1 (2002), pp. 67–82.

[Gan05] Ganesh, A. et al. “The effect of network topology on the spread of epidemics”. *IEEE INFOCOM*. 2005.

[Gar79] Garey, M. R. & Johnson, D. S. *Computers and Intractability*. Vol. 174. 1979.

[Gar08] Gargiulo, F. et al. “The saturation threshold of public opinion: are aggressive media campaigns always effective?” 2008.

[Gho04] Ghoniem, M. et al. “A comparison of the readability of graphs using node-link and matrix-based representations”. *IEEE symposium on information visualization*. 2004, pp. 17–24.

[Gra78] Granovetter, M. “Threshold models of collective behavior”. Vol. 83. 6. 1978, pp. 1420–1443.

[Gre20] Greenbaum, A. et al. “First-order perturbation theory for eigenvalues and eigenvectors”. Vol. 62. 2. 2020, pp. 463–482.

[Gre06] Gregory, M. et al. “User-directed sentiment analysis: Visualizing the affective content of documents”. *Proceedings of the Workshop on Sentiment and Subjectivity in Text*. 2006, pp. 23–30.

[Guo03] Guo, G. et al. “KNN model-based approach in classification”. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. 2003, pp. 986–996.

[Hat97] Hatzivassiloglou, V. & McKeown, K. “Predicting the semantic orientation of adjectives”. *35th annual meeting of the association for computational linguistics and 8th conference of the european chapter of the association for computational linguistics*. 1997, pp. 174–181.

[Hea12] Healey, C. G. & Enns, J. T. “Attention and Visual Memory in Visualization and Computer Graphics”. *IEEE Transactions on Visualization & Computer Graphics* **18**.7 (2012), pp. 1170–1188.

[Ho95] Ho, T. K. “Random Decision Forests”. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*. Montéal, Canada, 1995, pp. 278–282.

[Ish77] Ishii, H. et al. “Fractional knapsack problems”. *Mathematical Programming* **13**.1 (1977), pp. 255–271.

[Jal14] Jalkala, A. M. & Keränen, J. “Brand positioning strategies for industrial firms providing customer solutions”. 2014.

[Kal85] Kalish, S. “A new product adoption model with price, advertising, and uncertainty”. Vol. 31. 12. 1985, pp. 1569–1585.

[Kei02] Keim, D. A. “Information visualization and visual data mining”. *IEEE transactions on Visualization and Computer Graphics* **8**.1 (2002), pp. 1–8.

[Kem03] Kempe, D. et al. “Maximizing the spread of influence through a social network”. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003, pp. 137–146.

[Kep92] Kephart, J. O. & White, S. R. “Directed-graph epidemiological models of computer viruses”. *Computation: the micro and the macro view*. World Scientific, 1992, pp. 71–102.

[Ker27] Kermack, W. O. & McKendrick, A. G. “A Contribution to the Mathematical THeory of Epidemics”. *Proceedings of the Royal Society of London A* **115**.772 (1927), pp. 700–721.

[Kha11] Khan, M. & Khan, S. S. “Data and information visualization methods, and interactive mechanisms: A survey”. *International Journal of Computer Applications* **34**.1 (2011), pp. 1–14.

[Kot90] Kotler, P. & Franke, G. *Marketing: an introduction*. 1990.

[Kum17] Kumari, U. et al. “Sentiment analysis of smart phone product review using SVM classification technique”. *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. 2017, pp. 1469–1474.

[Laj76a] Lajmanovich, A. & Yorke, J. A. “A deterministic model for gonorrhea in a nonhomogeneous population”. *Mathematical Biosciences* **28**.3-4 (1976), pp. 221–236.

[Laj76b] Lajmanovich, A. & Yorke, J. A. “A deterministic model for gonorrhea in a nonhomogeneous population”. Vol. 28. 3. 1976, pp. 221 –236.

[Lan96] Langley, P. *Elements of machine learning*. 1996.

[Les14] Leskovec, J. & Krevl, A. *SNAP Datasets: Stanford Large Network Dataset Collection*. `http://snap.stanford.edu/data`. 2014.

[Li12] Li, C. et al. “Susceptible-infected-susceptible model: A comparison of N-intertwined and heterogeneous mean-field approximations”. Vol. 86. 2. 2012, p. 026116.

[Liu12a] Liu, B. “Sentiment analysis and opinion mining”. *Synthesis lectures on human language technologies* **5**.1 (2012), pp. 1–167.

[Liu12b] Liu, B. & Zhang, L. “A survey of opinion mining and sentiment analysis”. *Mining text data*. 2012, pp. 415–463.

[Liu19] Liu, J. et al. “Analysis and control of a continuous-time bi-virus model”. 2019.

[Lot56] Lotka, A. J. “Elements of mathematical biology”. 1956.

[Lou03] Lourenço, H. R. et al. “Iterated local search”. *Handbook of Metaheuristics*. 2003, pp. 320–353.

[Lov68] Lovins, J. B. “Development of a Stemming Algorithm”. *Mechanical Translation and Computer Linguistics* **11**.1–2 (1968), pp. 22–31.

[Lus03] Lusseau, D. et al. “The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations”. *Behavioral Ecology and Sociobiology* **54**.4 (2003), pp. 396–405.

[Man99] Manning, C. & Schutze, H. *Foundations of statistical natural language processing*. 1999.

[McA12] McAuley, J. J. & Leskovec, J. “Learning to discover social circles in ego networks.” *NIPS*. Vol. 2012. 2012, pp. 548–56.

[Mey00] Meyer, C. D. *Matrix analysis and applied linear algebra*. Vol. 71. 2000.

[Mob15] Mobilia, M. “Nonlinear q-voter model with inflexible zealots”. Vol. 92. 1. 2015, p. 012803.

[MW94] Montoya-Weiss, M. M. & Calantone, R. “Determinants of new product performance: a review and meta-analysis”. Vol. 11. 5. 1994, pp. 397–417.

[Muk20] Mukhopadhyay, A. et al. “Voter and majority dynamics with biased and stubborn agents”. Vol. 181. 4. 2020, pp. 1239–1265.

[Nad11] Nadkarni, P. M. et al. “Natural language processing: an introduction”. *Journal of the American Medical Informatics Association* **18**.5 (2011), pp. 544–551.

[Naz20] Nazir, A. et al. “Issues and challenges of aspect-based sentiment analysis: a comprehensive survey”. *IEEE Transactions on Affective Computing* (2020).

[New18] Newman, M. *Networks*. 2018.

[New06] Newman, M. E. et al. *The structure and dynamics of networks.* 2006.

[Ngu20] Nguyen, V. X. et al. “Dynamics of opinion formation under majority rules on complex social networks”. Vol. 10. 1. 2020, pp. 1–9.

[Nig59] Nightingale, F. *A Contribution to the Sanitary History of the British Army During the Late War with Russia*. London, UK: John W. Parker, 1859.

[Nob06] Noble, W. S. “What is a support vector machine?” *Nature biotechnology* **24**.12 (2006), pp. 1565–1567.

[Oel09] Oelke, D. et al. “Visual opinion analysis of customer feedback data”. *2009 IEEE symposium on visual analytics science and technology*. 2009, pp. 187–194.

[Omi09] Omic, J & Van Mieghem, P. “Epidemic spreading in networks—variance of the number of infected nodes”. 2009.

[Ona16] Onan, A. et al. “Ensemble of keyword extraction methods and classifiers in text classification”. *Expert Systems with Applications* **57** (2016), pp. 232–247.

[Par18] Paré, P. E. et al. “A Coupled Bi-Virus Spread Model in Networked Systems”. *IEEE American Control Conference*. 2018.

[Pat10] Pathak, N. et al. “A generalized linear threshold model for multiple cascades”. *IEEE International Conference on Data Mining*. 2010, pp. 965–970.

[Por80] Porter, M. F. “An Algorithm for Suffix Stripping”. *Program: Electronic Library and Information Systems* **14**.3 (1980), pp. 130–137.

[Pra12a] Prakash, B. A. et al. “Threshold conditions for arbitrary cascade models on arbitrary networks”. Vol. 33. 3. 2012, pp. 549–575.

[Pra12b] Prakash, B. A. et al. “Winner takes all: competing viruses or ideas on fair-play networks”. *ACM World Wide Web*. 2012.

[Pro17] Proskurnikov, A. V. & Tempo, R. “A tutorial on modeling and analysis of dynamic social networks. Part I”. Vol. 43. 2017, pp. 65–79.

[Pro18] Proskurnikov, A. V. & Tempo, R. “A tutorial on modeling and analysis of dynamic social networks. Part II”. Vol. 45. 2018, pp. 166–190.

[RB12] "Rafael Bru Rafael Cantó, R. S. & Urbano", A. “"A Brauer’s theorem and related results"”. Vol. "10". "1". "2012", "312 –321".

[Rai04] Raileanu, L. E. & Stoffel, K. “Theoretical comparison between the gini index and information gain criteria”. *Annals of Mathematics and Artificial Intelligence* **41**.1 (2004), pp. 77–93.

[Red07] Reddit. *r/gaming*. 2007.

[Red09a] Reddit. *Reddit*. 2009.

[Red09b] Reddit. *r/Smartphones*. 2009.

[Roz20] Rozemberczki, B. & Sarkar, R. “Characteristic Functions on Graphs: Birds of a Feather, from Statistical Descriptors to Parametric Models”. *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM ’20)*. 2020, 1325–1334.

[Rus02] Russell, S. & Norvig, P. “Artificial intelligence: a modern approach” (2002).

[Saf91] Safavian, S. R. & Landgrebe, D. “A survey of decision tree classifier methodology”. *IEEE transactions on systems, man, and cybernetics* **21**.3 (1991), pp. 660–674.

[Sah14] Sahneh, F. D. & Scoglio, C. “Competitive epidemic spreading over arbitrary multilayer networks”. Vol. 89. 6. 2014, p. 062817.

[Sam88] Samuel, A. L. “Some studies in machine learning using the game of checkers. II—recent progress”. *Computer Games I* (1988), pp. 366–400.

[San15] Santos, A. et al. “Bi-virus SIS epidemics over networks: Qualitative analysis”. *IEEE Transactions on Network Science and Engineering* **2**.1 (2015), pp. 17–29.

[Shn03] Shneiderman, B. “The eyes have it: A task by data type taxonomy for information visualizations”. *The craft of information visualization*. 2003, pp. 364–371.

[Šil10] Šilić, A. & Bašić, B. D. “Visualization of text streams: A survey”. *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. 2010, pp. 31–43.

[Smi14] Smith, H. L. *Monotone dynamical systems: an introduction to the theory of competitive and cooperative systems*. 2014.

[Sut18] Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction*. 2018.

[Tan20] Tangirala, S. “Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm”. *International Journal of Advanced Computer Science and Applications* **11**.2 (2020), pp. 612–619.

[Ton01] Tong, S. & Koller, D. “Support vector machine active learning with applications to text classification”. *Journal of machine learning research* **2** (2001), pp. 45–66.

[VM09] Van Mieghem, P. et al. “Virus spread in networks”. Vol. 17. 1. 2009, pp. 1–14.

[Vol28] Volterra, V. “Variations and fluctuations of the number of individuals in animal species living together”. *ICES Journal of Marine Science* **3**.1 (1928), pp. 3–51.

[Wan05] Wang, L. *Support vector machines: theory and applications*. Vol. 177. 2005.

[Wan16] Wang, X. et al. “Virus propagation modeling and convergence analysis in large-scale networks”. Vol. 11. 10. 2016, pp. 2241–2254.

[Wan03] Wang, Y. et al. “Epidemic spreading in real networks: An eigenvalue viewpoint”. *IEEE Reliable Distributed Systems*. 2003.

[War02] Ward, M. O. “A taxonomy of glyph placement strategies for multidimensional data visualization”. *Information Visualization* **1**.3-4 (2002), pp. 194–210.

[Was94] Wasserman, S., Faust, K., et al. *Social network analysis: Methods and applications*. Vol. 8. 1994.

[Wei16] Wei, X. et al. “Cooperative spreading processes in multiplex networks”. Vol. 26. 6. 2016, p. 065311.

[Wu10] Wu, Y. et al. “OpinionSeer: interactive visualization of hotel customer feedback”. *IEEE transactions on visualization and computer graphics* **16**.6 (2010), pp. 1109–1118.

[Wu14] Wu, Y. et al. “Opinionflow: Visual analysis of opinion diffusion on social media”. *IEEE transactions on visualization and computer graphics* **20**.12 (2014), pp. 1763–1772.

[Xu11] Xu, S. et al. “A stochastic model of multivirus dynamics”. Vol. 9. 1. 2011, pp. 30–45.

[Yan12] Yang, H.-X. et al. “Convergence to global consensus in opinion dynamics under a nonlinear voter model”. Vol. 376. 4. 2012, pp. 282–285.

[Yan17] Yang, L.-X. et al. “A bi-virus competing spreading model with generic infection rates”. Vol. 5. 1. 2017, pp. 2–13.

[Yan06] Yang, W.-S. et al. “Mining social networks for targeted advertising”. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS)*. Vol. 6. 2006, 137a–137a.

[Zha17] Zhang, S. et al. “Learning k for knn classification”. *ACM Transactions on Intelligent Systems and Technology (TIST)* **8**.3 (2017), pp. 1–19.

APPENDICES

A.1 Existing Results from Literature

We use the following well-known result for non-negative, irreducible matrices heavily throughout the paper.

Theorem 2. *(Perron-Frobenius)[Mey00] Let* $\mathbf{A}$ *be a non-negative, irreducible matrix. Then,* $\lambda(\mathbf{A})$ *is a strictly positive real number, and the corresponding eigenvector* $\mathbf{v}$ *where* $\mathbf{Av} = \lambda(\mathbf{A})\mathbf{v}$ *is also strictly positive. We call* $\lambda(\mathbf{A}) > 0$ *and* $\mathbf{v} \gg \mathbf{0}$ *the* ***PF eigenvalue*** *and* ***PF eigenvector*** *of the matrix, respectively.*

Proposition 2. *[RB12] Let* $\mathbf{M}$ *be an arbitrary* $N \times N$ *matrix with eigenvalues* $\lambda_1, \lambda_2, \cdots, \lambda_N$*, and let* $\mathbf{v}_k$ *be an eigenvector associated with eigenvalue* λ_k *for* $k \in \{1, \cdots, N\}$*. Then for any arbitrary vector* $\mathbf{q} \in \mathbb{R}^N$*,* $\mathbf{v}_k$ *is also an eigenvector of the matrix* $\mathbf{M}' = \mathbf{M} + \mathbf{v}_k\mathbf{q}^T$ *with the corresponding eigenvalue being* $\mu_k = \lambda_k + \mathbf{q}^T\mathbf{v}_k$*.*

A.2 Proof of Proposition 1

Proof. Consider two pairs of parameters $(\gamma, \mathbf{u})$ and $(\hat{\gamma}, \hat{\mathbf{u}})$, such that for all $i, j \in \mathcal{N}$, we have $\gamma p(u_i)p(u_j) \leq \hat{\gamma}\hat{p}(u_i)\hat{p}(u_j)$, with strict inequality for at least one pair of i, j. This also implies that $\sqrt{\gamma}\sum_i w_i p(u_i) < \sqrt{\hat{\gamma}}\sum_i w_i \hat{p}(u_i)$, meaning that the company needs to spend more on $(\hat{\gamma}, \hat{p}(u_i))$ than on $(\gamma, p(u_i))$. For both set of parameters, we assume that the respective ODE systems satisfy the coexistence condition (iii) from Section 2.4. Let $\mathbf{y}$ denote the resulting fixed point for $\mathbf{P}_2$ under parameters $(\gamma, \mathbf{u})$, and let $\mathbf{z}$ denote the fixed point under parameters $(\hat{\gamma}, \hat{\mathbf{u}})$. We will show that $\bar{y} < \bar{z}$, implying that it is always better to spend more if the budget allows for it.

Let $(\mathbf{x}, \mathbf{y})$ denote the fixed points of the ODE system (2.4), and $(\hat{\mathbf{x}}(t), \hat{\mathbf{y}}(t))$ denote the solution of the ODE system (2.4) with parameters $(\hat{\gamma}, \hat{\mathbf{u}})$, with the initial point $(\hat{\mathbf{x}}(0), \hat{\mathbf{y}}(0)) = (\mathbf{x}, \mathbf{y})$ (that is, starting from the fixed points of the system with original parameters $(\gamma, \mathbf{u})$).

Then, we have[1]

$$\begin{aligned}
\left.\frac{d\hat{\mathbf{x}}(t)}{dt}\right|_{t=0} &= \beta_1 \text{diag}(\mathbf{1}-\mathbf{x}-\mathbf{y}(t))\mathbf{A}\mathbf{x} - \delta_1\mathbf{x} = 0 \\
\left.\frac{d\hat{\mathbf{y}}(t)}{dt}\right|_{t=0} &= \beta_2 \text{diag}(\mathbf{1}-\mathbf{x}-\mathbf{y})\left[\mathbf{A}+\hat{\gamma}\hat{\mathbf{u}}\hat{\mathbf{u}}^T\right]\mathbf{y} - \delta_2\mathbf{y} \\
&> \beta_2 \text{diag}(\mathbf{1}-\mathbf{x}-\mathbf{y})\left[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T\right]\mathbf{y} - \delta_2\mathbf{y} = 0,
\end{aligned}$$

where the inequality is due to the assumption that $\gamma p(u_i)p(u_j) \leq \hat{\gamma}\hat{p}(u_i)\hat{p}(u_j)$, with strict inequality for at least one pair of $i, j \in \mathcal{N}$. Therefore, there exists an $\epsilon > 0$ for which, we have $\mathbf{x} \geq \hat{\mathbf{x}}(\epsilon)$ and $\mathbf{y} \leq \hat{\mathbf{y}}(\epsilon)$. From [Dos21], we know that the bi-SIS system is strongly monotone [Smi14] for $\mathbf{x}, \mathbf{y} \in (0,1)^N$. This basically means that $\mathbf{y} < \hat{\mathbf{y}}(\epsilon) \ll \mathbf{y}(t+\epsilon)$ for all $t > 0$, and since $\mathbf{z} = \lim_{t\to\infty} \mathbf{y}(t+\epsilon)$, we have $\mathbf{y} \ll \mathbf{z}$. This implies that $\bar{y} < \bar{z}$, completing the proof. □

A.3 Proof of Theorem 1

We begin with analyzing the impact of change in γ on $\mathbf{P}_2$ fixed point $\mathbf{y}$ (2.6) by taking the derivative of (2.6) with respect to γ evaluated at $(\gamma^c, \mathbf{u}^c)$ which yields,[2]

$$\left.\frac{\partial y_i}{\partial \gamma}\right|_{c.p.} = \tau_2(1-x_i^*)\sum_{j\in\mathcal{N}}(a_{ij} + \gamma^c p(u_i)^c p(u_j)^c)\left.\frac{\partial y_i}{\partial \gamma}\right|_{c.p.}. \tag{1}$$

Having $\frac{\partial \mathbf{y}}{\partial \gamma} > 0$ implies that it has a positive slope and $\mathbf{y}$ is an increasing function of γ. Similarly, the derivative of (2.6) with respect to $p(u_r)$ results in,

$$\left.\frac{\partial y_i}{\partial p(u_r)}\right|_{c.p.} = \tau_2(1-x_i^*)\sum_{j\in\mathcal{N}}(a_{ij} + \gamma p(u_i)^c p(u_j)^c)\left.\frac{\partial y_i}{\partial p(u_r)}\right|_{c.p.} \tag{2}$$

The expression in (1) and (2) demonstrates that $\partial \mathbf{y}/\partial\gamma = [\partial y/\partial\gamma]_i$ and $\partial \mathbf{y}/\partial p(u_r) =$

[1] for any two vectors $\mathbf{p} = [p_i]$ and $\mathbf{q} = [q_i]$, the inequality $\mathbf{p} > \mathbf{q}$ implies that $p_i \geq q_i$ with strict inequality for at least one entry. If for every entry, we have $p_i > q_i$, then we use $\mathbf{p} \gg \mathbf{q}$ to denote the ordering relationship.

[2] We define *c.p.* to refer to the point at which we take derivative of (2.6), that is, $\left.\frac{\partial y_i}{\partial \gamma}\right|_{c.p.} \triangleq \left.\frac{\partial y_i}{\partial \gamma}\right|_{(\gamma,\mathbf{u})=(\gamma^c,\mathbf{u}^c)}$.

$[\partial y/\partial p(u_r)]_i$ are eigenvectors of the matrix $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T]$ corresponding to eigenvalue τ_2. In fact, by the PF Theorem 2, they are precisely the first eigenvector of the $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T]$ (which is a non-negative, irreducible matrix) up to some multiplicative constants. For $\partial\mathbf{y}/\partial p(u_r)$, this multiplicative constant can be different for all $r\in\mathcal{N}$, and is directly related to the magnitude of increase in $\mathbf{y}$ caused by each $p(u_r)$. As a first step towards determining these constants for each $p(u_r)$, we give the following Lemma with respect to the change in $\bar{\mathbf{y}}$ for any increase in $\lambda(\gamma,\mathbf{u})$.

Lemma 2. *Let* $\lambda(\gamma,\mathbf{u})\triangleq\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T])$*, and* $(\gamma^c,\mathbf{u}^c)$ *be any pair of critical parameters. Let* $\bar{\mathbf{y}}(\gamma,\mathbf{u})\triangleq\mathbf{1}^T\mathbf{y}(\gamma\mathbf{u})/N$ *denote the market share as a function of any feasible* $(\gamma,\mathbf{u})$*. Then, the market share can be written as,*

$$\bar{\mathbf{y}}(\gamma,\mathbf{u})=\frac{\tau_2\mathbf{1}^T\mathbf{v}(\gamma^c,\mathbf{u}^c)}{N\sum_{i=1}^{N}\frac{v_i(\gamma^c,p(u_i)^c)^3}{(1-x_i^*)^2}}\left(\lambda(\gamma,\mathbf{u})-1/\tau_2\right)+O\left(\left(\lambda(\gamma,\mathbf{u})-1/\tau_2\right)^2\right) \tag{3}$$

Proof. Before proceeding with the proof of Lemma 2, we first provide another lemma which we use further.

Lemma 3. *Let* $\epsilon_1,\epsilon_2>0$ *be any small constants such that* $\gamma(\epsilon_1)$ *and* $\mathbf{u}(\epsilon_2)$ *are perturbations around* $\gamma,\mathbf{u}$*. Then, for all* $\epsilon_3>0$ *such that* $\lambda(\gamma,\mathbf{u})=\lambda(\gamma^c,\mathbf{u}^c)+\epsilon_3$*, there exists* $\epsilon_4>0$ *such that* $\mathbf{y}=\epsilon_4\mathbf{v}(\gamma,\mathbf{u})$.

Proof. From (1) one can observe that it is an eigenvalue problem of the form $\mathbf{A}\nu=\lambda\nu$, that is,

$$\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T])\Big|_{\gamma=\gamma^c}=1/\tau_2. \tag{4}$$

Among all possible solutions for eigenvalue, (4) is acceptable; according to the PF theorem, only the dominant eigenvector of the matrix, $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T]$ (being non-negative, irreducible), has all positive entries, which allows $\frac{dy_i}{d\gamma}>0$. Similarly, it can be shown that

for the change of $\mathbf{y}$ with respect to $\mathbf{u}$, only the dominant eigenvector of the matrix has all positive entries, which allows $\left.\frac{dy_i}{dp(u_i)}\right|_{p(u_i)=p(u_i)^c} > 0$.

From (1) we have that $1/\tau_2 = \lambda(\gamma^c, \mathbf{u}^c) > 0$ and $\frac{dy_i}{d\gamma} \triangleq [v]_i$ (or any scaled eigenvector), that is, $\mathbf{y} = \epsilon \mathbf{v}(\gamma, \mathbf{u})$ is the eigenvector of matrix $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T]$.[3] For $\lambda(\gamma^c, \mathbf{u}^c) > \lambda(\gamma, \mathbf{u})$, $1/\tau_2$ cannot be the eigenvalue and only possible solution is $[v]_i = 0$. Hence, for $\lambda(\gamma, \mathbf{u}) > \lambda(\gamma^c, \mathbf{u}^c)$ (with any change in γ or $\mathbf{u}$) i.e., for any small arbitrary constant ϵ, $\lambda(\gamma, \mathbf{u}) = \lambda(\gamma^c, \mathbf{u}^c) + \epsilon$, $1/\tau_2$ is the eigenvalue with $\mathbf{y} = \epsilon \mathbf{v}(\gamma, \mathbf{u})$, completing the proof. □

Now, consider the fixed point equations for the ODE system (2.4), which are given by

$$\mathbf{x} = \tau_1 \text{diag}(\mathbf{1} - \mathbf{x} - \mathbf{y})\mathbf{A}\mathbf{x} \tag{5}$$

$$\mathbf{y} = \tau_2 \text{diag}(\mathbf{1} - \mathbf{x} - \mathbf{y})\left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y}. \tag{6}$$

Let $\mathbf{S}_\mathbf{x} \triangleq \text{diag}(\mathbf{1} - \mathbf{x})$, where $\mathbf{x}$ is the fixed point from (5). From (6), we have the following expressions:

$$\frac{\mathbf{1}^T \mathbf{y}}{\tau_2} = (\mathbf{1} - \mathbf{x})^T \left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} - \mathbf{y}^T \left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} \tag{7}$$

$$\frac{\mathbf{y}^T \mathbf{S}_\mathbf{x}^{-1} \mathbf{y}}{\tau_2} = \mathbf{y}^T \left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} - \mathbf{y}^T \mathbf{S}_\mathbf{x}^{-1} \text{diag}(\mathbf{y}) \left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y}, \tag{8}$$

where the $\mathbf{y}$ terms from $\text{diag}(\mathbf{1} - \mathbf{x} - \mathbf{y})$ are separated out for aid in further analysis.

We aim to rewrite the above equations in terms of eigenvalues and eigenvectors. Let $\mathbf{M} \triangleq \mathbf{S}_\mathbf{x}\left[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T\right]$, and let $(\lambda_k, \mathbf{v}_k)$ for $k \in \{1, \cdots, N\}$ be the (right) eigenvalue-eigenvector pairs of $\mathbf{M}$ such that $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_N$. We have the following proposition regarding orthogonality of the right eigenvectors.

Proposition 3. *For all $k \neq j$, the right eigenvectors $\mathbf{v}_k$ of $\mathbf{M} = \mathbf{S}_{\mathbf{x}^*}[A + \gamma \mathbf{u}\mathbf{u}^T]$ satisfy $\mathbf{v}_k^T \mathbf{S}_\mathbf{x}^{-1} \mathbf{v}_j = 0$.*

Proof. Since $\mathbf{x}^*$ has strictly positive entries, the diagonal matrix $\mathbf{S}_{\mathbf{x}^*}$ is invertible, and has a *square root* given by $\mathbf{S}_{\mathbf{x}^*}^{1/2}$ which is also invertible. Observe that $\mathbf{S}_\mathbf{x}^{-1/2} \mathbf{M} \mathbf{S}_{\mathbf{x}^*}^{1/2} = \mathbf{S}_{\mathbf{x}^*}^{1/2}[\mathbf{A} +$

[3] $\mathbf{v}(\gamma, \mathbf{u}) = \mathbf{v}_1(\gamma, \mathbf{u})$ corresponds to the first eigenvector of matrix $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T]$.

$\gamma \mathbf{u}\mathbf{u}^T]\mathbf{S}_{\mathbf{x}^*}^{1/2} \triangleq \hat{\mathbf{M}}$, and hence $\mathbf{M}$ and $\hat{\mathbf{M}}$ are similar matrices, sharing the same set of eigenvalues [Mey00], with the right eigenvector of $\hat{\mathbf{M}}$ corresponding to eigenvalue λ_k given by $\mathbf{S}_{\mathbf{x}^*}^{-1/2}\mathbf{v}_k$. Moreover, $\hat{\mathbf{M}}$ is a symmetric matrix due to $\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T$ being symmetric, and its eigenvectors are orthogonal to each other. This implies that for all $k \neq j$, we have $\mathbf{v}_k^T \mathbf{S}_{\mathbf{x}^*}^{-1/2} \mathbf{S}_{\mathbf{x}^*}^{1/2} \mathbf{v}_j = \mathbf{v}_k^T \mathbf{S}_{\mathbf{x}^*}^{-1} \mathbf{v}_j = 0$, completing the proof. □

Assume that the eigenvectors are normalized such that $\mathbf{v}_k \mathbf{S}_{\mathbf{x}}^{-1} \mathbf{v}_k = \mathbf{1}$ for all $k \in \{1, \cdots, N\}$. The fixed point $\mathbf{y}$ can then be written as a linear combination of these eigenvectors as

$$\mathbf{y} = \sum_{k=1}^{N} c_k \mathbf{v}_k. \tag{9}$$

From the above equation, we have

$$\frac{\mathbf{1}^T \mathbf{y}}{\tau_2} = \sum_{k=1}^{N} \tau_2^{-1} c_k \mathbf{1} \mathbf{v}_k \tag{10}$$

$$\frac{\mathbf{y}^T \mathbf{S}_{\mathbf{x}}^{-1} \mathbf{y}}{\tau_2} = \sum_{k=1}^{N} \tau_2^{-1} c_k^2. \tag{11}$$

Substituting (9) in (7) gives us

$$\begin{aligned}
\frac{\mathbf{1}^T \mathbf{y}}{\tau_2} &= (\mathbf{1}-\mathbf{x})^T \left[\mathbf{A}+\gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} - \mathbf{y}^T \left[\mathbf{A}+\gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} \\
&= \mathbf{1}^T \mathbf{S}_{\mathbf{x}} \left[\mathbf{A}+\gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} - \mathbf{y}^T \mathbf{S}_{\mathbf{x}}^{-1} \mathbf{S}_{\mathbf{x}} \left[\mathbf{A}+\gamma \mathbf{u}\mathbf{u}^T\right]\mathbf{y} \\
&= \sum_{k=1}^{N} c_k \lambda_k \mathbf{1}^T \mathbf{v}_k - \sum_{k=1}^{N} \sum_{j=1}^{N} c_k c_j \lambda_k \mathbf{v}_j^T \mathbf{S}_{\mathbf{x}}^{-1} \mathbf{v}_k \\
&= \sum_{k=1}^{N} c_k \lambda_k \mathbf{1}^T \mathbf{v}_k - c_k^2 \lambda_k.
\end{aligned} \tag{12}$$

where the fourth equality comes by using the orthonormality of the eigenvectors, as in

Proposition 3. Similarly, by substituting (9) in (8), we obtain

$$\begin{aligned}
\frac{\mathbf{y}^T\mathbf{S}_\mathbf{x}^{-1}\mathbf{y}}{\tau_2} &= \mathbf{y}^T\big[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T\big]\mathbf{y}-\mathbf{y}^T\mathbf{S}_\mathbf{x}^{-1}\mathrm{diag}(\mathbf{y})\big[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T\big]\mathbf{y}\\
&= \mathbf{y}^T\mathbf{S}_\mathbf{x}^{-1}\mathbf{S}_\mathbf{x}\big[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T\big]\mathbf{y}-\mathbf{y}^T\mathbf{S}_\mathbf{x}^{-1}\mathrm{diag}(\mathbf{y})\mathbf{S}_\mathbf{x}^{-1}\mathbf{S}_\mathbf{x}\big[\mathbf{A}+\gamma\mathbf{u}\mathbf{u}^T\big]\mathbf{y}\\
&= \sum_{j=1}^{N}\sum_{k=1}^{N} c_j c_k\left(\lambda_k\mathbf{v}_j^T\mathbf{S}_\mathbf{x}^{-1}\mathbf{v}_k-\sum_{l=1}^{N} c_l\lambda_l\sum_{i=1}^{N}\frac{[\mathbf{v}_j]_i[\mathbf{v}_k]_i[\mathbf{v}_k]_i}{(1-x_i)^2}\right)\\
&= \sum_{j=1}^{N} c_j^2\lambda_j-\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} c_j c_k c_l\lambda_l\sum_{i=1}^{N}\frac{[\mathbf{v}_j]_i[\mathbf{v}_k]_i[\mathbf{v}_k]_i}{(1-x_i)^2} \qquad (13)
\end{aligned}$$

where the fourth equality is again due to the orthonormality of the eigenvectors. By equating (10) with (12), (11) with (13), and rearranging the terms, we get the following two equations.

$$\sum_{k=1}^{N}(\lambda_k-\tau_2^{-1})c_k\mathbf{1}^T\mathbf{v}_k=\sum_{k=1}^{N} c_k^2\lambda_k \qquad (14)$$

$$\sum_{j=1}^{N}(\lambda_j-\tau_2^{-1})c_j^2=\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} c_j c_k c_l\lambda_l\sum_{i=1}^{N}\frac{[\mathbf{v}_j]_i[\mathbf{v}_k]_i[\mathbf{v}_k]_i}{(1-x_i)^2}. \qquad (15)$$

Now, we observe that $\mathbf{x}$ and $\mathbf{y}$ are actually functions of γ and $\mathbf{u}$. For $(\gamma,\mathbf{u})$, that is, they go to some *critical* combination of the parameters which satisfy the coexistence conditions for $\mathbf{P}_2$ with equality, we have $\mathbf{x}(\gamma^c,\mathbf{u}^c)=\mathbf{x}^*$ and $\mathbf{y}(\gamma^c,\mathbf{u}^c)=\mathbf{0}$ from Lemma 3. λ_k, $\mathbf{v}_k$ and c_k are also a functions of $(\gamma,\mathbf{u})$, with $\lambda_1(\gamma^c,\mathbf{u}^c)=\lambda_1^c=\tau_2^{-1}$.

From Lemma 3, we know that for any $(\gamma,\mathbf{u})$ such that $\lambda(\gamma,\mathbf{u})=\lambda(\gamma^c,\mathbf{u}^c)+\epsilon_1$ for some $\epsilon_1>0$, there exists an $\epsilon_2>0$ such that $\mathbf{y}=\epsilon_2\mathbf{v}(\gamma^c,\mathbf{u}^c)$. In fact $\mathbf{y}=\alpha\mathbf{v}(\gamma^c,\mathbf{u}^c)+\beta\mathbf{w}$, where $\mathbf{w}$ is a vector orthogonal to $\mathbf{v}(\gamma,\mathbf{u})$, and β goes down to zero faster than α as $(\gamma,\mathbf{u})\to(\gamma^c,\mathbf{u}^c)$, or alternatively as $\lambda(\gamma,\mathbf{u})\to\lambda(\gamma^c,\mathbf{u}^c)\tau_2^{-1}$. In view of this, suppose that α and β can be expressed as $\alpha=\alpha_0(\lambda_1-\lambda_1^c)^p+o((\lambda_1-\lambda_1^c)^p)$ and $\beta=\beta_0(\lambda_1-\lambda_1^c)^q+o((\lambda_1-\lambda_1^c)^q)$, where $q>p>0$. Then, since c_1 is the component of $\mathbf{y}$ from (9) associated with $\mathbf{v}$, and c_k for all $k>1$ are associated with vectors orthogonal to $\mathbf{v}$, we have

$$c_1=r_1(\lambda_1-\lambda_1^c)^p+o((\lambda_1-\lambda_1^c)^p)$$

and

$$c_k = r_k(\lambda_1 - \lambda_1^c)^q + o((\lambda_1 - \lambda_1^c)^q).$$

Substituting these in the left hand side of (14) for when $\lambda_1 \to \lambda_1^c$ shows that it is of the order

$$r_1 \mathbf{1}^T \mathbf{v}_1 (\lambda_1 - \lambda_1^c)^{p+1} + \sum_{k=2}^{N} r_k \mathbf{1}^T \mathbf{v}_k (\lambda_k - \tau_2^{-1})(\lambda_1 - \lambda_1^c)^q + o\left((\lambda_1 - \lambda_1^c)^{\min(q,p+1)}\right),$$

and doing the same for the right hand side gives us

$$r_1^2 \lambda_1 (\lambda_1 - \lambda_1^c)^{2p} + \sum_{k=2}^{N} r_k^2 \lambda_k (\lambda_1 - \lambda_1^c)^{2q} + o\left((\lambda_1 - \lambda_1^c)^{2p}\right).$$

Equating the two orders gives us two cases. *Case A:* $q \geq p+1$, which means that $p+q = 2p$ implying $p = 1$; *Case B:* $q < p+1$, implying $q = 2p$. To find the correct values of p and q, we conduct the same analysis on (15).

We obtain that the left hand side of (15) is of the order

$$r_1^2 (\lambda_1 - \lambda_1^c)^{2p+1} + \sum_{k=2}^{N} r_k^2 (\lambda_k - \tau_2^{-1})(\lambda_1 - \lambda_1^c)^{2q} + o\left((\lambda_1 - \lambda_1^c)^{\min(2q,2p+1)}\right).$$

For the right hand side term, we mention both the first and second order terms:

$$r_1^3 \lambda_1 (\lambda_1 - \lambda_1^c)^{3p} \sum_{i=1}^{N} \frac{v_{1i}}{(1-x_i)^2} + o((\lambda_1 - \lambda_1^c)^{3p}) + O((\lambda_1 - \lambda_1^c)^{2p+q}).$$

where $2p+q > 3p$ implying that $O((\lambda_1 - \lambda_1^c)^{2p+q})$ is of higher order than $o((\lambda_1 - \lambda_1^c)^{3p})$ and $\mathbf{v}_1 = [v_{1i}] = [v_i]$ corresponds to the first eigenvector of $\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T]$. Equating the orders of the left and right hand sides, we get the following two cases: *Case C:* $2q \geq 2p+1$, in which case $2p+1 = 3p$ and $2q = 2p+q$, implying $p = 1$ and $q = 2$; *Case D:* $2q < 2p+1$,

which gives us $2q = 3p$ and $2p+1 = 2p+q$. Observe that Cases A and C are the only ones complementing each other, while the others pairs are contradictory. Hence, it must be true that $p = 1$ and $q = 2$. By equating the corresponding powers of the left hand side and right hand side, we get

$$r_1^2(\lambda_1 - \lambda_1^c)^3 = r_1^3\lambda_1(\lambda_1 - \lambda_1^c)^3 \sum_{i=1}^{N} \frac{v_i}{(1-x_i)^2}$$
$$\implies r_1 = \left[\lambda_1 \sum_{i=1}^{N} \frac{v_i}{(1-x_i)^2}\right]^{-1}$$

This allows us to rewrite (9) as

$$\mathbf{y} = \frac{\mathbf{v}(\gamma, \mathbf{u})}{\lambda_1(\gamma, \mathbf{u}) \sum_{i=1}^{N} \frac{v_i(\gamma, \mathbf{u})}{(1-x_i(\gamma, \mathbf{u}))^2}} \left(\lambda_1(\gamma, \mathbf{u}) - \lambda_1(\gamma^c, \mathbf{u}^c)\right) + O\left((\lambda_1(\gamma, \mathbf{u}) - \lambda_1(\gamma^c, \mathbf{u}^c))^2\right),$$

where we have made the dependence on $(\gamma, \mathbf{u})$ more obvious. A Taylor series expansion with respect to $\lambda_1(\gamma, \mathbf{u})$ and centred around $\lambda_1(\gamma^c, \mathbf{u}^c) = \tau_2^{-1}$ then gives us[4]

$$\mathbf{y} = \frac{\tau_2 \mathbf{v}(\gamma^c, \mathbf{u}^c)}{\sum_{i=1}^{N} \frac{v_i(\gamma^c, \mathbf{u}^c)}{(1-x_i^*)^2}} \left(\lambda_1(\gamma, \mathbf{u}) - \tau_2^{-1}\right) + O\left((\lambda_1(\gamma, \mathbf{u}) - \tau_2^{-1})^2\right),$$

Summing over the entries of $\mathbf{y}$ and then dividing by N gives us the final result, thus completing the proof. □

Now, we use the results from Lemma 2 towards the proof of Theorem 1. Note that the term $\lambda(\gamma, \mathbf{u}) - 1/\tau_2$ on the RHS in Lemma 2 is actually based on given $(\gamma^c, \mathbf{u}^c)$ at the critical point, that is, $\mathbf{x} = \mathbf{x}^*, \mathbf{y} = \mathbf{0}$, which we use in the following proof.

Proof of Theorem 1. From first-order approximation, we know that $f(x) \approx f(a) + f'(a)(x -$

[4] $\lambda_1(\gamma, \mathbf{u})$ corresponds to the first eigenvalue from PF Theorem and is equivalent to $\lambda(\gamma, \mathbf{u})$

a). Utilizing this, for the eigenvalue $\lambda(\gamma^c, \mathbf{u})$ with respect to change in u_i based on given $\lambda(\gamma^c, \mathbf{u}^c)$ at critical point we have,

$$\lambda(\gamma, \mathbf{u}) \approx \lambda(\gamma^c, \mathbf{u}^c) + \lambda'(\gamma^c, \mathbf{u}^c)\Delta\mathbf{u} \tag{16}$$

Utilizing Theorem 1 [Gre20], we know that at $u_i = u_i^c$, $\lambda'(\gamma^c, \mathbf{u}^c) = \mathbf{q}^*\mathbf{M}'\mathbf{p}$ where $\lambda'(\gamma^c, \mathbf{u}^c)$ and $\mathbf{M}'$ are, respectively, the derivatives of $\lambda(\gamma, \mathbf{u})$ and $\mathbf{M}$ at $p(u_i) = p(u_i)^c$. And $\mathbf{M} = \mathbf{S}_{\mathbf{x}^*}(\mathbf{A} + \gamma\mathbf{u}\mathbf{u}^T)$, with largest eigenvalue $\lambda(\gamma^c, \mathbf{u}^c)$ corresponding to right eigenvector $\mathbf{p} = \mathbf{v}$ and left eigenvector $\mathbf{q} = \mathbf{v}^T\mathbf{S}_{\mathbf{x}^*}^{-1}$. Substituting this in (16) and from Lemma 1, we have

$$\lambda(\gamma^c, \mathbf{u}) - \frac{1}{\tau_2} = \mathbf{q}^*\mathbf{M}'\mathbf{p}\Delta\mathbf{u}, \tag{17}$$

where $\Delta\mathbf{u} = p(u_i) - p(u_i)^c$. The derivative of $\mathbf{M}$ with respect to $p(u_i)$ gives,

$$\begin{aligned}\frac{\partial\mathbf{M}}{\partial p(u_i)} &= \gamma^c\mathbf{S}_{\mathbf{x}^*}\Big[\frac{\partial\mathbf{u}}{\partial p(u_i)}\mathbf{u}^T + \mathbf{u}\frac{\partial\mathbf{u}^T}{\partial p(u_i)}\Big] \\ &= \gamma^c\mathbf{S}_{\mathbf{x}^*}\Big[e_i\mathbf{u}^T + \mathbf{u}e_i^T\Big]\end{aligned}$$

This gives us the derivative of $\mathbf{M}$ at $p(u_i)$ is $\mathbf{M}' = \gamma\mathbf{S}_{\mathbf{x}^*}\Big[e_i\mathbf{u}^T + \mathbf{u}e_i^T\Big]$. Substituting this on the RHS of (17) gives

$$\begin{aligned}\mathbf{q}^*\mathbf{M}'\mathbf{p}\Delta\mathbf{u} &= \gamma^c\mathbf{v}^T\mathbf{S}_{x^*}^{-1}\mathbf{S}_{\mathbf{x}^*}[e_i\mathbf{u}^T + \mathbf{u}e_i^T]\mathbf{v}(p(u_i) - p(u_i)^c) \\ &= 2\gamma^c\mathbf{v}^T e_i\mathbf{u}^T\mathbf{v}(p(u_i) - p(u_i)^c) \\ &= 2\gamma^c\mathbf{u}^T\mathbf{v}(p(u_i) - p(u_i)^c)v_i\end{aligned}$$

where the term $2\gamma^c\mathbf{u}^T$ always remains constant. Substituting this in (17) gives,

$$\lambda(\gamma^c, \mathbf{u}) - \frac{1}{\tau_2} = 2\gamma^c\mathbf{u}^T\mathbf{v}(p(u_i) - p(u_i)^c)v_i \tag{18}$$

which implies that the change is in the direction of the eigenvector $\mathbf{v} = [v_i]$ for all $i \in \mathcal{N}$. □

A.4 Derivation of (2.8)

From first-order approximation, we know that $f(x) \approx f(a) + f'(a)(x-a)$. Utilizing this, the first-order approximation of $\lambda(\tilde{\gamma}, \tilde{\mathbf{u}})$ at $\tilde{\gamma} = \gamma^c$, i.e., $\Delta\gamma = \gamma^c - \tilde{\gamma}$ for a given $\lambda(\gamma^c, \mathbf{u}^c)$, we have

$$\lambda(\tilde{\gamma}, \tilde{\mathbf{u}}) \approx \lambda(\gamma^c, \mathbf{u}^c) + \lambda'(\gamma^c, \mathbf{u}^c)\Delta\gamma \tag{19}$$

Utilizing Theorem 1 [Gre20] we know that,

$$\lambda'(\gamma^c, \mathbf{u}^c) = \mathbf{q}^* \mathbf{M}' \mathbf{p},$$

where $\lambda'(\gamma^c, \mathbf{u}^c)$ and $\mathbf{M}'$ are the derivatives of $\lambda(\gamma^c, \mathbf{u}^c)$ and $\mathbf{M} = \mathbf{S}_{\mathbf{x}^*}(\mathbf{A} + \tilde{\gamma}\tilde{\mathbf{u}}\tilde{\mathbf{u}}^T)$ at $\tilde{\gamma} = \gamma^c$, respectively. Let $\mathbf{p} = \mathbf{v}^c$ and $\mathbf{q} = \mathbf{v}^{c\,T}\mathbf{S}_{\mathbf{x}^*}^{-1}$ be the right and left eigenvector of $\mathbf{M}$. Substituting this in (19) and from Lemma 1, that is, $\lambda(\mathbf{S}_{\mathbf{x}^*}[\mathbf{A} + \gamma^c \mathbf{u}^c \mathbf{u}^{c\,T}]) = 1/\tau_2$, we have

$$\lambda(\tilde{\gamma}, \tilde{\mathbf{u}}) - \frac{1}{\tau_2} = \mathbf{q}^* \mathbf{M}' \mathbf{p} \Delta\gamma. \tag{20}$$

The derivative of $\mathbf{M}$ with respect to $\tilde{\gamma}$ gives,

$$\frac{\partial \mathbf{M}}{\partial \tilde{\gamma}}\Big|_{\tilde{\gamma}=\gamma^c} = \mathbf{S}_{\mathbf{x}^*}\tilde{\mathbf{u}}\tilde{\mathbf{u}}^T$$

Substituting this on the RHS of (20) and normalizing $\mathbf{p}, \mathbf{q}$ gives

$$\begin{aligned}\mathbf{q}^* \mathbf{M}' \mathbf{p} \Delta\gamma &= \frac{\mathbf{v}^{c\,T}\mathbf{S}_{x^*}^{-1}\mathbf{S}_{\mathbf{x}^*}\tilde{\mathbf{u}}\tilde{\mathbf{u}}^T\mathbf{v}^c}{\mathbf{v}^{c\,T}\mathbf{S}_{x^*}^{-1}\mathbf{v}^c}\Delta\gamma \\ &= \frac{(\mathbf{v}^{c\,T}\tilde{\mathbf{u}})^2}{\mathbf{v}^{c\,T}\mathbf{S}_{x^*}^{-1}\mathbf{v}^c}\Delta\gamma\end{aligned}$$

Rearranging the terms and substituting this in (20) gives,

$$\tilde{\gamma} = \gamma^c - [\lambda(\tilde{\gamma}, \tilde{\mathbf{u}}) - \frac{1}{\tau_2}] \frac{\mathbf{v}^{c\,T} \mathbf{S}_{x^*}^{-1} \mathbf{v}^c}{(\mathbf{v}^{c\,T} \tilde{\mathbf{u}})^2}, \tag{21}$$

where $\lambda(\tilde{\gamma}, \tilde{\mathbf{u}})$ is the largest eigenvalue of $\mathbf{M}$ at $\tilde{\gamma} = \gamma_c$ where γ^c corresponds to γ_0 in (2.8).

A.5 Proof of Lemma 1

Proof. Consider the matrix $\mathbf{M} = \mathbf{S}_{\mathbf{x}^*}(\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T)$. By substituting $\mathbf{u} = \mathbf{S}_{\mathbf{x}^*}^{-1}\mathbf{v}$,[5] it can be written as

$$\mathbf{M} = \mathbf{S}_{\mathbf{x}^*}(\mathbf{A} + \gamma \mathbf{S}_{\mathbf{x}^*}^{-1} \mathbf{v}\mathbf{u}^T) = \mathbf{S}_{\mathbf{x}^*}\mathbf{A} + \gamma \mathbf{v}\mathbf{u}^T,$$

which is now in the form of a rank-one perturbation to the matrix $\hat{\mathbf{M}} = \mathbf{S}_{\mathbf{x}^*}\mathbf{A}$, and is in the same form as in Proposition 2 with $\mathbf{v}$ as the first eigenvector of $\hat{\mathbf{M}}$, and $\gamma \mathbf{u}$ being the additional $N-$dimensional vector. Then, we have from Proposition 2 that $\mathbf{v}$ is also the first eigenvector of $\mathbf{M} = \mathbf{S}_{\mathbf{x}^*}(\mathbf{A} + \gamma \mathbf{u}\mathbf{u}^T)$ associated with eigenvalue $\lambda(\mathbf{M}) = \lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{A}) + \gamma \mathbf{u}^T\mathbf{v} = \lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{A}) + \gamma \mathbf{u}^T \mathbf{S}_{\mathbf{x}^*}\mathbf{u}$. Finally, by rearranging terms in the above expression, we can check that the value of γ for which $\lambda(\mathbf{M}) = 1/\tau_2$ is given by $\gamma = \frac{1/\tau_2 - \lambda(\mathbf{S}_{\mathbf{x}^*}\mathbf{A})}{\mathbf{u}^T \mathbf{S}_{\mathbf{x}^*}\mathbf{u}}$, thus completing the proof.

□

[5] The vector $\mathbf{u}$ can actually be taken as any scalar multiple of $\mathbf{S}_{\mathbf{x}^*}^{-1}\mathbf{v}$, that is $\mathbf{u} = c\mathbf{S}_{\mathbf{x}^*}^{-1}\mathbf{v}$ for any $c \in \mathbb{R}$. We drop the 'c' notation for clarity.