

Aino Tarvainen

INDEPENDENT CASCADE PARAMETER ESTIMATION WITH MARKOV CHAINS

Master of Science Thesis
Faculty of Information Technology and Communication Sciences
Examiners: Henri Hansen
Juho Kanninen
December 2024

ABSTRACT

Aino Tarvainen: Independent Cascade Parameter Estimation with Markov Chains
Master of Science Thesis
Tampere University
Master's Programme in Statistical Data Analytics
December 2024

The study of network diffusion processes has traditionally focused on the field of epidemiology but since the beginning of the Internet age, the application fields have also greatly expanded to information diffusion and individuals' social media activity patterns. To understand these dynamics is a relevant field of study, because it allows us to control them, for instance when we consider the countermeasures against epidemic spread, the utilization of word-of-mouth information propagation in the context of marketing or moderating the sharing of harmful online content.

One of the main parameters in commonly used diffusion process models is the transmission probability, which presents the rate at which something is transferred from one individual to other. In many cases when these models are applied in practice, researchers use some a priori knowledge when assigning their values, but there exists methods by which the transmission probabilities may be reconstructed from observed diffusion data.

However, many of these estimation methods tend to overestimate when the networks contain locally high density neighbourhoods. Therefore, this thesis's objective is to describe and validate a new alternative method using Markov chain modelling, which would result in unbiased estimates in this kind of situation.

We base our estimation method in the Independent Cascade model and our approach in this thesis is to treat Independent Cascade data as a Markov process. The main contributions of the thesis are thus to firstly describe the translation process of Independent Cascade data to Markov chains and secondly, to provide the process for estimating transmission probabilities from this Markov chain translated data.

We tested the proposed method on small synthetic networks using varying sample sizes. Our chosen network topologies include the star-ring and block graphs, which are characterized by their notably highly connected nodes. Our results indicate that the method works as intended, i.e. it is capable of generating high quality estimates that do not contain the kind of bias observed in other methods.

The method described will therefore prove useful when we wish to analyse data from networks of well connected, clique-like neighbourhoods. In addition, since presenting diffusion processes as a Markov process is a simple and intuitive conversion, the method has plenty space for further development and modification.

Keywords: Independent Cascade, Markov Chain, diffusion, transmission probability, parameter estimation, networks

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Aino Tarvainen: Independent Cascade -mallin parametrien estimointi Markovin ketjuilla
Pro gradu -tutkielma
Tampereen yliopisto
Statistical Data Analytics -maisteriohjelma
Joulukuu 2024

Diffuusioprosessien analysointi verkostoissa on perinteisesti keskittynyt epidemiologian alan tutkimuskohteisiin, mutta Internet-aikakauden alkamisesta lähtien sovellusalat ovat laajentuneet enemmän myös informaatiodiffuusioon ja käyttäytymismalleihin sosiaalisessa mediassa. Näiden dynamiikkojen ymmärtäminen on relevantti tutkimusala, koska sen avulla niitä voidaan kontrolloida, kun esimerkiksi harkitaan epidemioiden leviämisen vastakeinoja, vertaisviestinnän hyödyntämistä markkinoinnissa tai vaikkapa haitallisen verkkosisällön jakamisen hillitsemistä.

Yksi yleisesti käytettyjen diffuusiomallien pääparametreista on transmissiotodennäköisyys, joka edustaa todennäköisyyttä, jolla jokin ominaisuus siirtyy yksilöltä toiselle. Monesti kun näitä malleja sovelletaan käytännössä, tutkijat käyttävät hyväksi jotain a priori -tietoa transmissiotodennäköisyyksille, mutta on olemassa menetelmiä, joiden avulla arvot voidaan muodostaa havainnoidusta diffuusiodatasta.

Toisaalta, monilla näistä estimaatiomenetelmistä on taipumus yliarvioida estimaattiarvoja verkostoille, jotka sisältävät naapurustoja, joilla on korkea kaaritiheys. Tämän pro gradu -tutkielman tavoite näin ollen on kuvailla ja todistaa kelpolliseksi uusi, vaihtoehtoinen menetelmä hyödyntäen Markovin ketju -mallinnusta, joka tuottaisi harhattomia estimaatteja tämänkaltaisissa tapauksissa.

Kyseinen estimointimenetelmä perustuu Independent Cascade -malliin, ja sen lähestymistapa estimointitehtävään on käsitellä Independent Cascade -dataa Markov-prosessina. Tutkielman pääkontribuutiot ovat ensiksi kuvata translaatioprosessi, jolla Independent Cascade -data muunnetaan Markovin ketjuiksi, sekä toiseksi selostaa itse estimointimenetelmä, jolla transmissiotodennäköisyydet estimoidaan Markovin ketjuksi muunnetusta datasta.

Esitettyä menetelmää testattiin pienillä synteettisillä verkostoilla käyttäen erikokoisia otantakojoja. Tutkielmaan valitut verkostotopologiat sisältävät nk. star-ring ja blokkigraafeja, joiden ominaisuuksina ovat niiden huomattavan tiheästi toisiinsa kytkeytyneet solmut. Testien tulokset viittaavat siihen, että menetelmä toimii kuten oli oletettu, eli se kykenee tuottamaan korkealaatuisia estimaatteja, joissa ei ole havaittavissa samanlaista vääristymää kuin muissa menetelmissä.

Tämän vuoksi tutkielmassa kuvailtu menetelmä on hyödyksi, kun tavoitteena on analysoida dataa verkostoista, jotka sisältävät hyvin kytkeytyneitä klikkimäisiä naapurustoja. Koska diffuusioprosessien esittäminen Markov-prosesseina on lisäksi yksinkertainen ja intuitiivinen muunnos, tämä menetelmä jättää tilaa myös myöhemmälle kehitykselle ja muunnelmille.

Avainsanat: Independent cascade, Markovin ketju, diffuusio, transmissiotodennäköisyys, parametristimointi, verkostot

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck -ohjelmalla.

PREFACE

After 9 months of work it is finally done, and not at all too soon. I would like to first of all thank my supervisors Henri Hansen and Juho Kanninen for giving me this opportunity and all the support along the way. This project allowed me to apply my skills in creative and motivating ways, and I believe the end product matches its initial intentions.

I want to thank my family for encouraging my ambitions and always being proud of me, and all my friends and peers who have listened patiently to my worries, even if they could not fully understand them. Much thanks also to my workplace and host family abroad, who have supported me and have provided an environment where I could finalise my thesis to its full potential.

Fukuoka, 9th December 2024

Aino Tarvainen

CONTENTS

1.	Introduction	1
2.	Related work	3
3.	Information Diffusion Models	5
3.1	Graphs	5
3.2	The SIR model.	6
3.3	The Independent Cascade model	7
3.4	Markov chains	8
4.	Formulation.	10
4.1	Translation of Independent Cascade episodes to Markov chain.	10
4.2	Transition probability estimation given an independent cascade model.	11
4.3	Transmission probability estimation with Markov chain translation.	14
4.4	Python implementation	15
4.4.1	Libraries and the solver function	15
4.4.2	Improving code performance	15
5.	Data generation	17
5.1	Network models	17
5.1.1	Star-ring network model	17
5.1.2	Block graph	18
5.2	Assigning transmission probabilities to graphs	19
5.3	Generating Independent Cascade data	20
6.	Numerical results	22
6.1	Constant transmission probabilities.	22
6.2	Noisy transmission probabilities	25
6.3	The effect of graph type, order and sample size.	27
7.	Discussion	29
	References.	31

1. INTRODUCTION

The study of network diffusion processes has much interested statisticians in the past few decades as the collection and computing of network data has advanced. By understanding the patterns of epidemics, word-of-mouth information propagation and even the sharing of social media content, we can take action and disrupt these processes to our advantage. In a post-COVID, social media driven world this could mean for example controlling the spread of viral diseases in the physical world and misinformation online.

Diffusion models created by previous researchers most often require transmission probabilities whose values are based on a priori knowledge. It is then natural that many researchers have aspired to develop methods to reconstruct these probabilities from observed propagation data. However, these methods tend to underperform when the network contains high edge density neighbourhoods. This thesis's objective is therefore to describe and validate a diffusion parameter estimation method using a new approach, which will result in unbiased estimates in this setting.

The base framework in this study for stochastic spreading processes is the Independent Cascade (IC) model. Our study approach is to treat the IC data as a Markov process and fit the model using Markov state transition probability estimation. The conversion is very natural since each step in the IC process can be presented as a corresponding transition between states in a Markov model, and as the Markov state transition probabilities will be dependent on the transmission probability values in the IC model, they can be used for estimating the IC parameters. The main contributions of this thesis is then to describe the translation process from IC episodes to Markov chains and the subsequent transmission probability estimation from this Markov state data.

We introduce two Markov estimation types: one where we assume homogeneous transmission probabilities, and one where we assume heterogeneous transmission probabilities. These methods will be demonstrated on synthetic data on two types of simple undirected network structures that contain high local edge density. The data contains IC episodes from both homogenous and heterogeneous transmission probability models.

This study is constructed as follows: In Chapter 2 we review the major transmission probability estimation methods developed so far. Chapter 3 introduces the relevant propagation and stochastic process models used in this study and Chapter 4 details the proposed

method's formulation and implementation. Chapter 5 contains the description of the data used in the experiments and how it was generated. The estimation results using this data are then presented in Chapter 6. The thesis concludes in Chapter 7, where we summarize the test results and discuss the future of the proposed method.

2. RELATED WORK

There are in general two approaches to study diffusion data, which are the Independent Cascade (IC) [1] [2] model and the Linear Threshold (LT) model [3]. As the method proposed in this paper concerns the IC model, we discuss here those studies which are also based on it.

One of the first approaches to estimate Independent Cascade model parameters was coined by Saito et al. [4]. In their approach the task is formulated as a maximum likelihood problem and solved with the Expectation-Maximization (EM) algorithm. This method has then garnered much attention and the likelihood approach has been the basis for later studies such as [5], [6] and [7].

The next approach to estimating network diffusion parameters was to apply Bayesian inference tools introduced by Dutta et al. in their paper from 2018 [8]. The method works by approximating the likelihood function with so-called ABC algorithms [9], thus giving a likelihood-free alternative to the estimation.

Another recent development in improving the IC parameter reconstruction is the inclusion of dynamic message-passing (DMP) equations by Lokhov et al. in [10]. These equations allow us to derive the marginal probabilities of the IC variables in a more efficient and accurate manner compared to the maximum likelihood approach. Lokhov et al. have continued the study of transmission probability estimation under the DMP framework in [11] and [12], in which the authors address the problem of incomplete data.

The DMP approach works well on sparse networks of real-life data, but the existence of local short loops generates bias in the results. The method is therefore limited to tree-like graphs and difficult to apply to intra-community analysis. The bias comes from the DMP method's tendency to overestimate the marginal probabilities in high-order neighbourhoods.

A more thorough overview of the limitations of the DMP estimation is given by Gao et al. in [13] where they introduce a new method based on DMP. In this paper the researchers apply Graph Neural Networks alongside DMP to derive the marginal probabilities of graphs with loops. However, due to computational complexity, the current version of this method struggles with scalability in regards to larger complex networks.

On other hand, several authors have introduced the idea of including additional data to improve the transmission probability reconstruction for social media space dynamics [14] [15] [16] [17]. These papers aim to model information diffusion in online networks regards to the type of content shared and the types of users involved in the sharing. These studies propose that individual users' inherent traits and activity history, and the topic of the shared content can influence the process of how the information spreads in the community. While very fruitful to the study of viral marketing and rumour spreading, these approaches have a very limited field of application.

3. INFORMATION DIFFUSION MODELS

This chapter offers a brief overview on the relevant models used in this thesis' estimation method and a selection of notions important to the method formulation. We begin in section 3.1 with the introduction of graphs, which are our chosen structure for modeling communities. Sections 3.2- 3.3 define the model for representing the information diffusion process. Lastly in section 3.4 we describe the Markov chain model, which is our study's primary probabilistic model.

3.1 Graphs

The basic data structure used in this study is graphs, also referred to as networks. They are a useful tool to study phenomena where the connections between objects hold much importance – for example when studying information diffusion in social networks. The study of graphs is called graph theory and this section contains a summary of its basic notions. The following definitions are adopted from Reinhard Diestel's third edition of *Graph Theory* [18].

A graph G contains a set of nodes (also referred to as vertices) V and edges E , which are 2-element subsets of V . A node corresponds to a single object and an edge presents a connection between nodes: for example, a node could represent a person and an edge then would represent a friendship between two people. Nodes are hereafter referred by their corresponding index number and edges between nodes u and v are noted as a pair (u, v) . When a node u is noted to belong to a graph's node set V or an edge to an edge set E , we simplify the notation to $e \in G$ and $(u, v) \in G$.

The total number of nodes in a graph G is referred to as its order. In formulation a graph order is denoted by $|G|$. The number of edges is subsequently denoted by $||G||$.

Two nodes u and v are considered to be neighbours if there exists an edge (u, v) joining them in the graph. The complete set of neighbours for u is denoted by $N(u)$ and the degree of the node is the number of its neighbours, $|N(u)|$.

Graphs can be considered either directed or undirected. Directed graphs differ from undirected graphs by defining an initial node and a terminal node for each edge. This means that in directed graphs one may be able to travel from a node u to v but not vice versa.

When the initial and terminal nodes are the same node, the edge is called a loop. Directed graphs also allow multiple edges from the same initial and terminal nodes.

In other words, undirected graphs are defined by the lack of initial/terminal node distinction and the exclusion multiple edges and loops. In this study we consider only undirected graphs for the sake of simplicity, but in theory the method could be applied to directed graphs as well after small modifications.

Edges can be assigned to carry additional data. These values are called weights and they can represent the strength of a connection, distance between objects, the type of connection et cetera. In this study the weights are used to express the transmission probabilities between nodes. In the case of undirected graphs this means that we assume the transmission probability between two nodes to be the same regardless of which node the transmission began from.

This study utilizes two types of graph topologies: star-ring and block graphs. These models are described in detail in Chapter 5 where we depict the data generation process for this paper.

3.2 The SIR model

For this study, we will model information diffusion based on the susceptible-infected-recovered (SIR) model first proposed in 1927 by W.O. Kermack and A.G. McKendrick [19]. Although originally specified to model the spread of contagious disease, with COVID-19 being a recent popular topic, the SIR model can be and has been applied to various other fields of study. These examples include modeling user adoption and abandonment phases of online social networks [20], spread of extremist ideology on social media [21] and emotional contagion in the case of emergency evacuations [22].

In the classic SIR model a member of the population is in one of three possible states at any times; At susceptible state the individual is healthy but vulnerable to infection. At infected state the individual is a carrier of the disease and has the capacity to infect others. Lastly, the individual reaches the recovered state when the disease has exited the body. In this last state individuals are no longer able to infect others and they have also gained an immunity against new infections.

The original objective of the SIR model was to estimate the number of individuals at each stage at each time step t , denoted as S_t , I_t and R_t respectively. In Kermack and McKendrick's work [19] the authors formulated the estimation of these values at each time

step with functions

$$\begin{aligned} S_{t+1} &= -\beta S_t I_t \\ I_{t+1} &= \beta S_t I_t - \gamma I_t \\ R_{t+1} &= \gamma I_t, \end{aligned} \tag{3.1}$$

where β is the infection transmission probability coefficient and γ the rate of recovery coefficient of infected individuals.

In this model the infected individuals are able to infect any susceptible node in the population. When we apply the SIR model to networks, the role of topology is emphasized. In this case each node is in one of the model's three stages and each infected node is able to infect susceptible nodes only in its first order neighbourhood. Additionally, infected nodes have the chance to infect others for only one time step with transmission probability $\beta \in [0, 1]$ and move to recovered state immediately after. This is comparable to special case of 3.1 where $\gamma = 1$.

The SIR model has received many adjustments, advancements and extensions since the 1920s and it has become a staple in epidemiological research. Some modifications to the SIR model include the SIS model, in which individuals do not gain immunity after infection and instead revert to the susceptible state, and the SEIR and SEIS models, which add an exposed state (E) where the individual has been infected but is not yet infectious.

These models may describe the stages of an epidemic more accurately, but for our setting of information diffusion, the simple monotonic SIR model will be enough: we will assume that once a person has learned a piece of information, they are unable to unlearn it.

3.3 The Independent Cascade model

The first formulation of cascade models is attributed to Goldenberg et al. [1] [2]. There have since risen a few variations of the cascade models, and the classic model introduced in these papers became to be referred as the independent cascade model (IC). The model was originally intended for studying viral marketing and the word-of-mouth phenomenon.

In a certain sense the IC model is a generalization of the network SIR model. The difference is that instead of a constant transmission probability β , each edge $e = (u, v) \in E$ is assigned an individual transmission probability $p_{u,v} \in [0, 1]$, which is comparable to an edge weight. This introduces a new dynamic to the diffusion model where some channels are more prone to spread the infection than others.

One of the applications of the independent cascade model is the influence maximization problem. Here the objective is to find nodes in the network which, when assigned as the source of the infection, will result in the maximal expected amount of infected nodes

in the diffusion cascade. An overview of the current methods and state of the influence maximization problem can be found in the article [23] by Ye et al. published in 2022.

Usually when applying the IC model, the transmission probabilities $p_{u,v}$ are assigned according to some a priori knowledge. These presumptions are based on the nature of connection between individuals that is assumed to affect the transmission rate. Some examples of these given in Ref. [24] include frequency of interaction, geographical proximity and historical evidence.

3.4 Markov chains

In addition to the Independent Cascade, this thesis will model information diffusion with Markov chains. Markov models are stochastic models for processes that follow the Markov property discovered by Russian mathematician Andrey Markov in early 1900s [25]. The basic assumption of this property is that the future state of a process depends only on the properties of the current state and not on any previous or future state. Notation for Markov chains in this section follows the definitions by MacDonald and Zucchini in [26].

There exists multiple models that apply the Markov property and they are collectively named Markov models. The simplest of the Markov models is the discrete-time Markov chain (DTMC) which observes the sequence of observations at discrete time steps. This is in contrast to the continuous-time Markov chain (CTMC), which is otherwise identical to DTMC, but in which the process moves from state to state over continuous time lapses. The DTMC model is often referred to by authors simply as the Markov chain and CTMC as the Markov process.

The Markov chain assumes that the states of the process are completely observable. When the sequence of states is unobservable but we assume that the observations at each step are dependent on a latent (hidden) process which we can observe, the model is called a Hidden Markov model (HMM) [27]. In other words, we can infer the unobservable Markov state according some dependent variables. For example, we can identify a disease through symptoms such as fever and cough without microscopically observing the actual pathogen in the body. However, for this study we restrict our scope to the simple Markov chain since the states are observable to us. In future research the method introduced in this thesis may be extended to HMMs.

Next we will review the basic notations of Markov chains. Let the discrete-time Markov chain be defined by the state space $S = \{1, 2, \dots, m\}$ and the transition probability matrix $\Gamma = (\gamma_{ij})$. The transition probabilities are thus defined as

$$\gamma_{ij} = P(S_{t+1} = j | S_t = i) \quad (3.2)$$

for all states $i, j \in S$ and time steps $t \in \mathbb{N}$. This is the probability of transitioning to state j in the next time step given the current state is i . Therefore it must follow that $\gamma_{ij} \in [0, 1]$ and $\sum_j^m \gamma_{ij} = 1$.

Transition probabilities can be estimated from data using their maximum likelihood estimator (MLE). Let $n_{i,j}$ be the number of transitions from state i to state j in an observed Markov chain. The MLE for transition probabilities is thus given by

$$\hat{\gamma}_{ij} = \frac{n_{ij}}{\sum_{k=1}^m n_{ik}} \quad (3.3)$$

In other words, we calculate the proportion of transitions beginning in state i and ending in j from all observed transitions from i .

When the model is saturated i.e. we consider all state transition pairs, the number of transitions and subsequently the dimensions of the Γ matrix will be $m \times m$ in total.

4. FORMULATION

In this chapter we define the steps of this thesis's proposed transmission probability method. First, in sections 4.1 and 4.2, we present the formulation of translating IC episodes into Markov chain framework and how we can analytically estimate the Markov transition possibilities given an IC model network. Then, in section 4.3 we present how the method uses the Markov chain translation to estimate transmission probabilities. Lastly, in section 4.4 we specify how the method was applied in Python to get our results showcased in Chapter 6.

4.1 Translation of Independent Cascade episodes to Markov chain

Translating the independent cascade model to Markov chain and vice versa is a very natural process. Both of these function in the discrete time framework and moreover, independent cascades inherently follow the Markov property: the state of the network depends only on the nodes' SIR status at the previous time step.

In the independent cascade to Markov chain translation, the state space of a graph G with order $|G| = n$ will be hereafter defined as $S = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m\}$, where \mathbf{X}_i is a vector of infection statuses of each node. The state vector is therefore $\mathbf{X}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ where $x_{i,u}$ is the SIR state of node u in state \mathbf{X}_i . We represent the status of the node with index values $x_{i,u} \in \{0, 1, 2\}$, where 0 corresponds to the susceptible state, 1 the infected state and 2 the recovered state. As there are three possible status options for one node, the maximum number of unique states in the Markov chain is $m_{max} = 3^n$. Subsequently, the saturated transition probability matrix dimensions come to be $3^n \times 3^n$.

After converting the SIR model network into a Markov state space, IC episodes can be translated to Markov chains. This is done by taking a cascade episode and creating a sequence of Markov states that correspond to the SIR state of the network at each time step of the cascade. For example, a three node network with a cascade where first node 1 is infected, then it infects node 2, which does not infect node 3, would be presented as a Markov chain as $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$, where $\mathbf{X}_1 = (1, 0, 0)$, $\mathbf{X}_2 = (2, 1, 0)$ and $\mathbf{X}_3 = (2, 2, 0)$. This example is also illustrated in Figure 4.1.

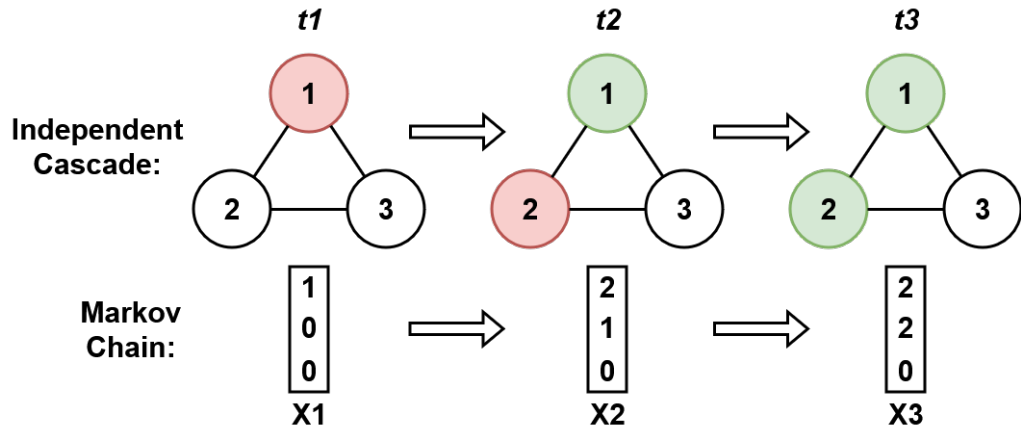


Figure 4.1. A propagation process event illustrated as an IC episode and as a Markov chain translation.

Multiple episodes can be appended into one complete, longer Markov chain. This chain then can be used to estimate the transmission probability. It should be noted that in this case each IC episode must be separated by a "neutral" state, otherwise there would be transitions that go against the SIR principles. In this paper the neutral state is defined as a null vector $X_0 = (0, 0, \dots, 0)$, i.e. there is no infection in the network, and it is used to mark the beginning and end of individual IC episodes.

4.2 Transition probability estimation given an independent cascade model

There are two cases where we can use the IC model to estimate Markov transition probabilities: If we have a sample of IC episodes, we can use maximum likelihood estimation. If we instead have knowledge of the IC model network parameters, the exact transition probabilities can be derived analytically.

In the first case, we translate the sample of observed IC episodes and translate them to a Markov chain. The translated data can then be used to estimate the transition probability matrix Γ using the MLE equation in 3.3.

On the other hand, if the transmission probabilities $p_{u,v}$ of an IC network are known, the transition probabilities can be calculated analytically. This is useful, since the alternate would be to estimate them using Monte Carlo sampling, which is time consuming and includes a degree of variability in results.

Before addressing the process of calculating individual transition probabilities, we must acknowledge that not every state transition is possible in the IC framework. By defining and identifying these cases, we can assign a zero value to these transitions in the

probability matrix Γ without the need of further calculations thus optimizing the computing process.

There are two cases between states for which we know a transition is not feasible according to the SIR mechanics. Both of these cases the transition probability is automatically zero.

The first case is linked to the linearity of the process: a susceptible node may not become recovered before becoming infected, an infected node must become recovered immediately in the next step and a recovered node may not revert to a previous stage in the SIR model. In other words, the set of recovered nodes must be equal to the combined set of recovered and infected nodes in the previous state.

The other infeasible case is when a susceptible node becomes infected even though none of its neighbours were infectious in the previous state. An exception for this is the beginning state of a cascade episode, where the infection seed nodes are set to infected status. This exception is addressed at the end of this subsection.

To summarize these two cases, for a transition (i, j) to be feasible in graph G it must satisfy the following:

$$\begin{aligned} \{u \in G : x_{i,u} \neq 0\} &= \{v \in G : x_{j,v} = 2\} \\ \wedge \\ \neg \exists u \in G : x_{j,u} = 1 \wedge (\forall v \in N(u) \mid x_{i,v} \neq 1) \end{aligned} \tag{4.1}$$

To calculate the transition probability between two Markov states i and j in feasible cases, we essentially derive the probability of nodes being infected according to state i given the previous state is j . In order to do this, we first need a method to calculate the probability of an individual node u being infected given the previous state.

Let S_t be a state in Markov chain at time t , and let $x_u^{(t+1)}$ be the SIR status of node u at time $t + 1$. Let $P(x_u^{(t+1)} = x \mid S_t = \mathbf{X}_i)$ be the probability that node u is in SIR stage x given that the previous state is \mathbf{X}_i .

Now, the probability of a node not becoming infected at time $t + 1$ is a joint probability of all its neighbours failing to infect. In other words,

$$P(x_u^{(t+1)} = 0 \mid S_t = \mathbf{X}_i) = \prod_{v \in N(u)} (1 - p_{u,v} \mathbb{1}_{x_{i,v}=1}), \tag{4.2}$$

where $\mathbb{1}_{x_{i,v}=1}$ is the indicator function that node v was in the infected state in the previous state i .

Successful and unsuccessful infection are mutually exclusive and exhaustive events, thus we can apply the complement rule on 4.2 to derive the probability of a successful infection:

$$P(x_u^{(t+1)} = 1 \mid S_t = \mathbf{X}_i) = 1 - P(x_u^{(t+1)} = 0 \mid S_t = \mathbf{X}_i) \quad (4.3)$$

Let V_i be the set of vulnerable nodes in state i , i.e. it is the set of susceptible nodes which have at least one infected node in their neighbourhood:

$$V_i = \{u \in G : x_{i,u} = 0 \wedge (\exists v \in N(u) \mid x_{i,v} = 1)\} \quad (4.4)$$

Due to the independence of infection events between each vulnerable node, the complete state transition probability simplifies to a joint probability product of individual infection probabilities. Adding the previously defined feasibility requirements, we get the complete formulation for calculating the transition probabilities analytically:

$$\gamma_{ij} = \begin{cases} 0 & \text{if does not satisfy feasibility (4.1)} \\ \prod_{u \in V_i} P(x_u^{(t+1)} = x_{j,u} \mid S_t = \mathbf{X}_i) & \text{otherwise} \end{cases} \quad (4.5)$$

The exception to 4.5 is transitions to and from the neutral state X_0 which is used for separating independent cascades in the Markov chain. The probability distribution for transitions beginning from the state (γ_{0j}) is equivalent to the distribution of seed infection states for a sample of IC episodes.

On other hand, an episode ends when a state has no more nodes in the infected state, i.e. further infections are not possible. In this case the transition probability is 1 for the neutral state and 0 otherwise.

These special cases regarding the neutral state can be expressed as

$$\begin{aligned} \gamma_{0j} &= P(C_1 = \mathbf{X}_j) \\ \gamma_{i0} &= \begin{cases} 0 & \text{if } \exists u \in G : x_{i,u} = 1 \\ 1 & \text{otherwise} \end{cases} \end{aligned} \quad (4.6)$$

where $P(C_1 = \mathbf{X}_j)$ is the probability of state j being the first state of a Markov chain translated IC episode.

4.3 Transmission probability estimation with Markov chain translation

Now that we have formulated the translation of IC episodes into Markov chains and the analytical calculation of transition probabilities on known transmission probabilities, we can present the actual transmission probability estimation process of this study's proposed method.

The method begins with a network G with unknown transmission probabilities p for each of its edges, and a set of IC episodes observed from it at discrete time steps. Note, that this method requires that the graph topology is known a priori and the data is complete for every time step.

Firstly, we generate a Markov state space S for the network and translate the cascades into a Markov chain (see section 4.1). Let $\hat{\gamma}_{ij}$ be the maximum likelihood estimate of transition probability for transition (i, j) given the Markov chain translated data (see section 3.4).

Secondly, we create an IC model G^* based on the original network G and assign some transmission probabilities p^* to it. These are our estimation seed values and they can be chosen randomly or according to researchers' initial presumptions on the actual model parameters. After this, we apply the method defined in 4.2 to G^* and solve model-based transition probabilities $\tilde{\gamma}_{ij}(p^*)$ analytically.

Now that we have calculated the data- and model-based transition probabilities, the estimation problem becomes a non-linear least squares problem:

$$\hat{p} = \arg \min_{p^*} \sum_{i,j} \|\hat{\gamma}_{ij} - \tilde{\gamma}_{ij}(p^*)\|^2 \quad (4.7)$$

In other words, we must update p^* according to some solver algorithm until the difference between Markov chains according to the data and the estimates in p^* is minimized. The simple least squares approach was chosen due to its simplicity and the assumption that the variables are homoscedastic.

There are two versions how this estimation method is applied in this thesis. The first is to estimate an individual transmission probability for each edge in the network, which will be called the heterogeneous method. This is the version described above. The second method is called then the homogeneous method, and in it we estimate one common transmission probability for all edges. This is done by replacing vector p^* in the minimization algorithm with a single parameter β and assigning this value to all edges in G^* .

The simpler homogeneous method is computationally much more efficient and it proves beneficial when we can assume that the transmission probability variability is low and when we are interested in the global transmission probability estimate of the graph. In other cases the heterogeneous method in theory should result in more accurate results.

4.4 Python implementation

Finally in this section we provide the programming details related to the implementation of the proposed method. First we list the relevant libraries used and the specifics of the chosen solver function. Then we introduce some tactics we used to improve the code efficiency.

4.4.1 Libraries and the solver function

The chosen programming language for this study is Python and the experiments were run on version 3.9.12 of the language. For creating and manipulating network data we used the NetworkX library (version 3.1). For solving the non-linear least squares model fit, a function called *optimize.least_squares* from library SciPy (1.13.0) was used.

The solver function *optimize.least_squares* has two required arguments: a function to be minimized and parameter seed value vector. For other arguments, the bounds are set to 0 and 1, otherwise the parameters are left as default. For more details on the function's configuration, see the library documentation: [28].

The minimizing function we give the solver function in this research application returns the residual vector between the data-based transition probabilities $\hat{\gamma}_{ij}$ and analytical probabilities $\tilde{\gamma}_{ij}(p^*)$, which are calculated using the values for p^* from the algorithm's latest iteration.

The default algorithm chosen to minimize the sum of squared residuals in *least_squares* is the Trust Region Reflective (TRF) algorithm, which is found to be robust in bounded non-linear minimization problems. This algorithm solves linear and non-linear least squares by creating sub-problems through approximating the function to be minimized with a simpler function. The simpler function is then minimized over a so-called trust region. A detailed description of the TRF algorithm formulation can be found in [29].

4.4.2 Improving code performance

In section 3.4 it was stated that for a network with n amount of nodes, the maximum number of Markov chain states is 3^n and the number of unique transitions is $3^n \times 3^n$. As one can imagine, the size of the transition matrix becomes unreasonably large quickly for any decent sized network. This becomes a challenge for parameter estimation, since

the model-based transition matrix must be calculated at each iteration of the minimization function. It must also be noted that the increase of edges in the network increases the number of parameters to be estimated, subsequently increasing the complexity of the parameter estimation problem.

However, as it was then established in section 4.2, not every transition is possible in the independent cascade framework due to infeasibility defined in 4.1. For example, for all feasible transitions (i, j) , the transition (j, i) is not feasible due to the fact that a process can not revert to a previous state in the monotonous SIR model.

In a similar manner, not all states are usually possible in an IC model. As the structure of a given network restricts the possibilities of how the infection spreads, this also limits the collection of unique states one can observe in IC episodes. For example, there shouldn't be infected nodes in one part of the network, if the infection has not yet reached it from another part.

By acknowledging these aspects of the Independent Cascade to Markov chain translation, one can increase the estimation performance. Therefore when implementing the method, we construct the Markov state space by including only states present in the data. In addition, we consider only those unique transitions that are found during the Markov chain translation step of the estimation method.

Thanks to these procedures, the number of values to be recalculated at each iteration of the least squares minimization stage decreases to a fraction of the complete transition matrix. At the same time, this eliminates the need to test for feasibility as the transitions are confirmed already by the data to be valid.

As an extra procedure to enhance the code performance in this particular experiment, the generated cascade samples are restricted to begin from the same seed node each episode. This restricts the Markov state space of the data, but it is possible only because we use synthetic data and therefore cannot be applied to empirical cases in all cases. Nonetheless, the other optimizing methods introduced are still applicable to empirical data.

5. DATA GENERATION

All the data used to showcase the proposed method was generated synthetically specifically for this thesis. This was done in order to create an experiment setting where comparisons and analysis can be performed without concerns related to empirical data, such as incomplete or noisy data. This approach also allows us to assess the experiment results with confidence since the real transmission probabilities are known.

The chapter begins with the introduction of the graphs that were chosen to test the method. Then we go over the process of assigning transmission probabilities to these graphs. Lastly we detail how these graphs were then used to generate IC episode data.

5.1 Network models

In this study we consider two types of network topologies. The first one is the so-called star-ring model, which was chosen for its simple topology. The second one is the block graph, which approximates networks one might encounter in real life data, especially in social networks. Both of these models are characterized by high-order local neighborhoods and short loops, which was something the DMP estimation method struggles with (see Chapter 2). The graphs are undirected and we do not include self-loops.

Five different sized graphs were made for each of the graph types. Their sizes are 5, 7, 10, 12 and 15 nodes. Additionally, we consider two types of structure for the block graph. Therefore, we have 15 unique graph topologies to analyze. Having variety both in the sizes and structure of the networks allows us to evaluate the scalability and generalization of the estimation method.

5.1.1 Star-ring network model

A star-ring network model is a hybrid topology of the star and the ring graph. A star graph's characteristic is that it contains a single node with degree $n - 1$ and the rest have only one degree, and the order of the graph is n . In other words, the graph has one node that is connected to every node and the other nodes are only connected to this center node. A ring graph on the other hand is a connected graph of 2-degree nodes, thus the edges create a ring through all the nodes. Combining these two topologies creates a

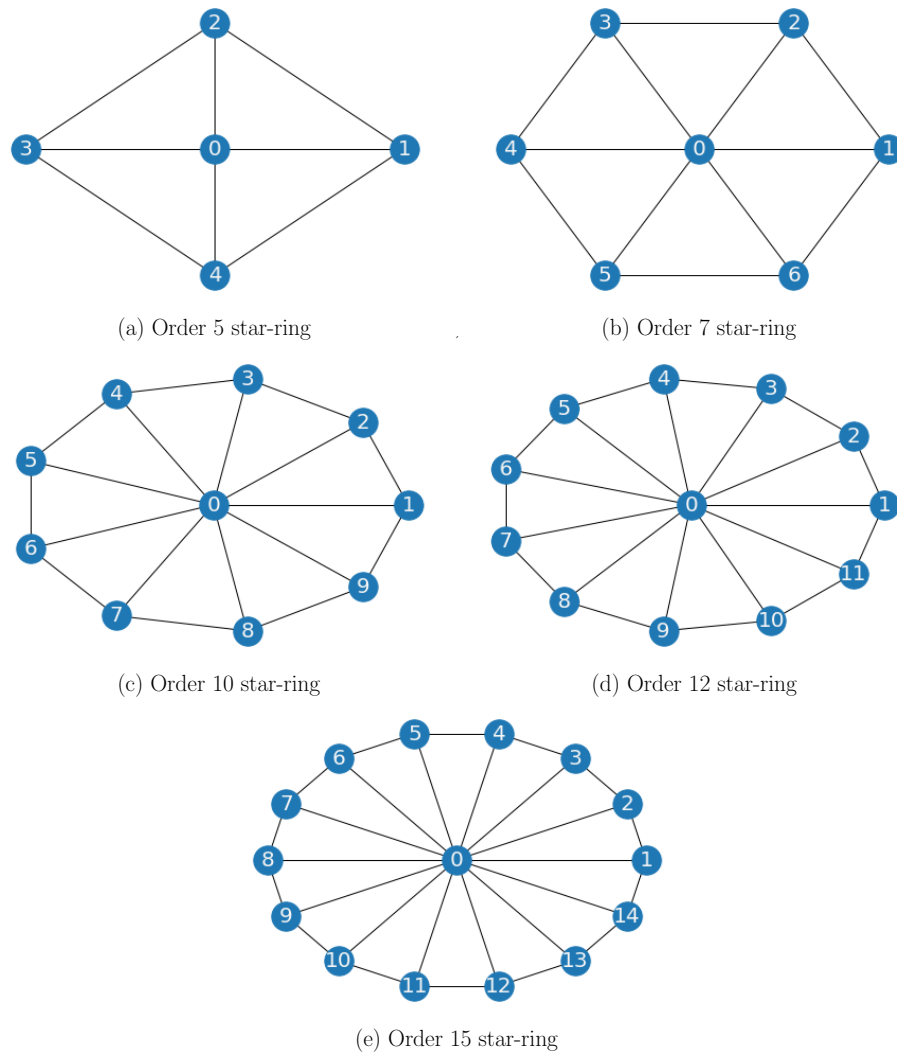


Figure 5.1. Different sized star-ring networks.

star-ring graph, a network where there is one central node that is connected to all other nodes, which create a ring around the central node.

Figure 5.1 contains the diagrams of each star-ring graph used in this study. The graphs' edge counts are given in Table 5.1. The number of edges in a star-ring graph is a linear function $2(n - 1)$, in other words the number of edges increases by two with each additional node in the graph.

5.1.2 Block graph

The second type of network topology we use is a generalization of the block graph, also known as the clique tree, which is an algorithm to create random graphs with highly connected cliques. For creating a block graph, each of the n nodes are first assigned randomly to one of m blocks while ensuring that none of the blocks end up empty. After this

Order	Star-ring	Three-block	Four-block
5	8	5	4
7	12	10	8
10	18	19	16
12	22	28	21
15	28	40	36

Table 5.1. Edge counts for all network types

one of the blocks is chosen as a root block. The remaining blocks are then attached to the graph by randomly choosing one of their member nodes and assigning it to either the root block or block already connected to the root block, thus creating interblock connections.

When all the blocks are attached to the graph, the result is a block graph. Further interblock connections can be made by adjusting the connection coefficient γ , which determines how many nodes are assigned to other blocks after the graph becomes connected. After the blocks are determined, we set their individual edge densities to 1, i.e. every node in a block has an edge with each other.

This model mimics the dynamic of many real life social networks, in which people can be members of multiple communities that are highly connected within but less so with each other: for example a person might belong to a family unit, a workplace and a hobby circle, but all of these communities usually have little overlap in their members. Thanks to this characteristic and the fact that this model also contain many short loops, this graph type was deemed interesting for testing the proposed method.

In this study we consider block graphs with three and four blocks. The connection coefficient is set to zero, meaning that the random assignment of nodes to new blocks is halted immediately when the graph becomes connected. In our graphs this resulted in networks which have at maximum two nodes that belong to multiple blocks.

Figures 5.2 and 5.3 contain all the three-block and four-block graphs generated for this study using the previous description. Separate blocks are marked with different coloured nodes. Grey colour denotes the bottleneck nodes that belong to multiple blocks. The graphs' edge counts are listed in Table 5.1.

5.2 Assigning transmission probabilities to graphs

The graphs will be assigned two different sets of transmission probabilities for their edges. In the first set the graph has the same transmission probability $p_{u,v} = \beta = 0.5$ for all (u, v) . In the second set we add a random noise variable $e \sim N(0, 0.3^2)$ to β , thus

$$p_{u,v} \sim \text{Normal}(0.5, 0.3^2), \quad (5.1)$$

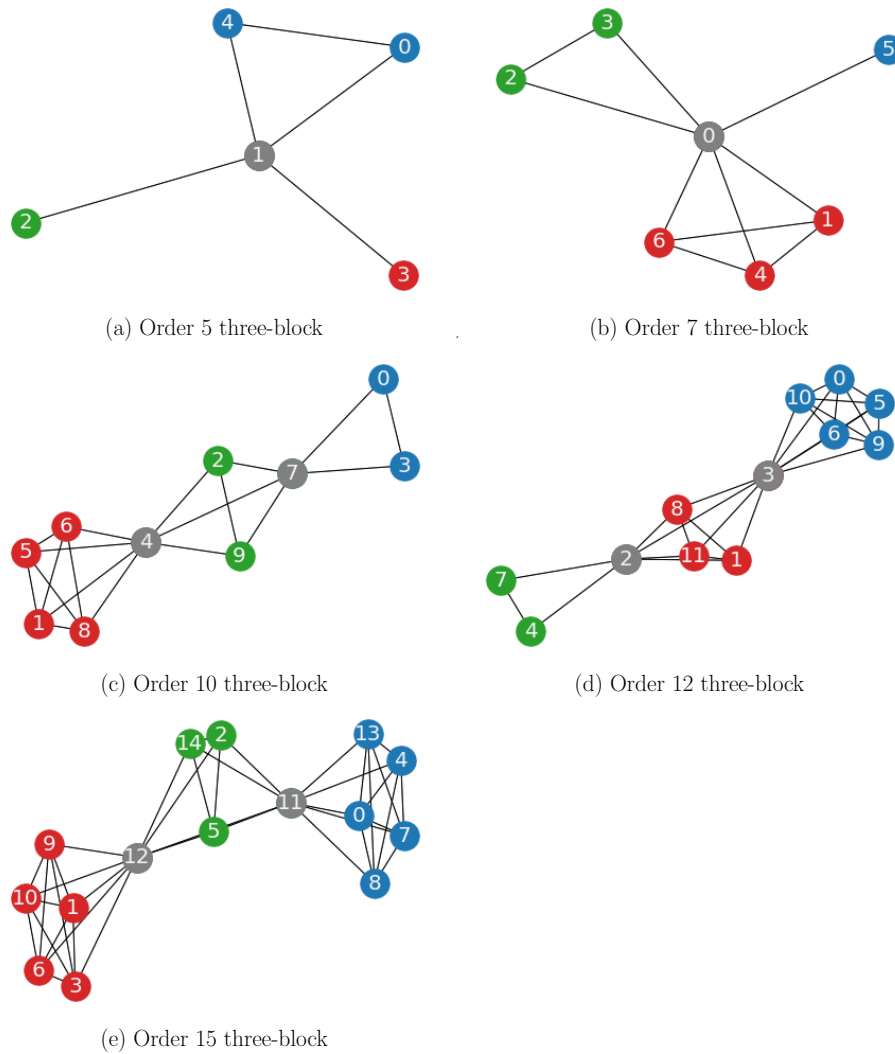


Figure 5.2. Different sized three-block graphs.

with the restriction that probabilities can not be outside values in $[0, 1]$. The noise was randomly generated to data using Python's NumPy library's *random.normal* function [30].

By having both constant and varied transmission probabilities we can study how the estimation is affected by noise. Additionally, we can compare the performance of the homogeneous and heterogeneous methods between these value sets.

5.3 Generating Independent Cascade data

As it was previously established, the data we use for estimating transmission probabilities consists of IC episodes. A single IC episode is a sequence of consecutive time steps in an IC process. Each step is registered as a set of nodes that were infected at that particular step.

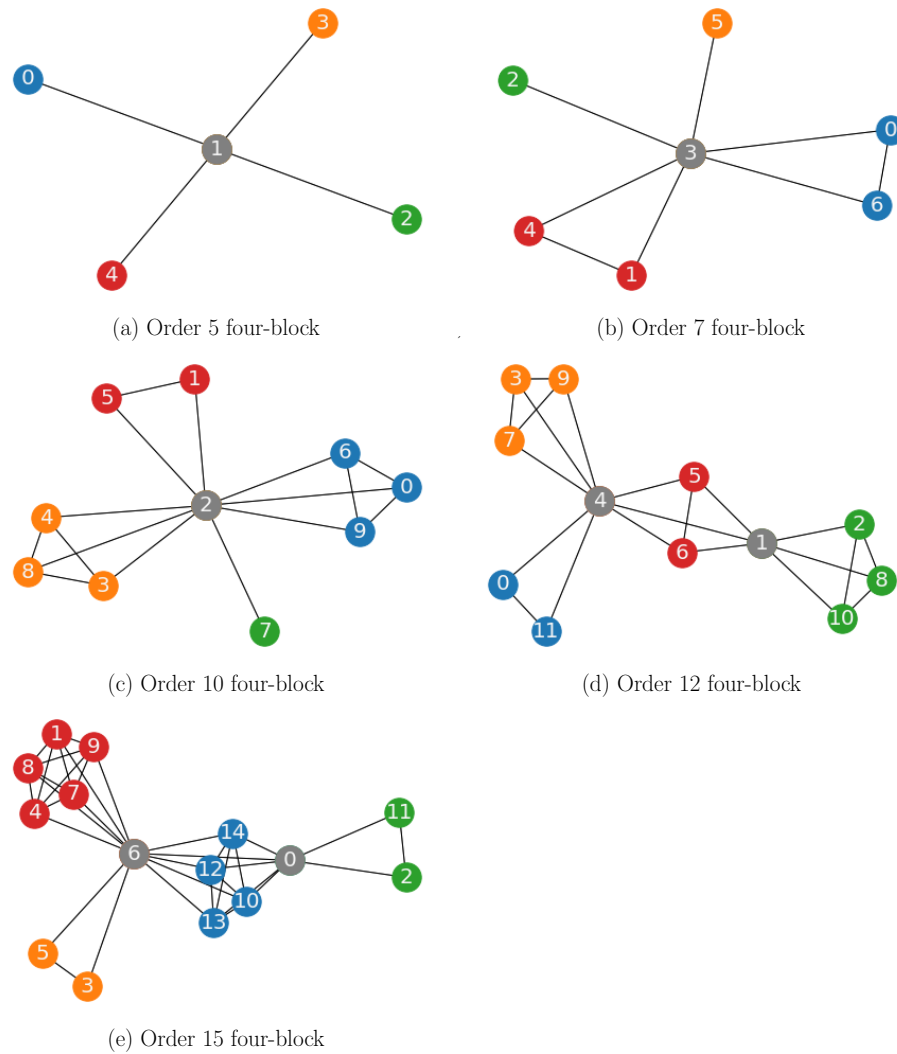


Figure 5.3. Different sized four-block graphs.

In this study we generate IC cascades from a singular seed node which is set to be the graph's 0 index node. IC samples are then generated by at each step randomly selecting nodes to be infected according to their infected neighbours' transmission probabilities.

We test the transmission probability estimation on different sized IC samples. The samples are generated independently from each other and their sizes range from 100 to 1000 with 100 episode intervals. In other words, we have ten independent random samples for each graph and transmission probability set combination.

When applying the principles aimed at optimizing the computation introduced in Chapter 4.4, all of the possible Markov states and transitions may not be observed in the data, especially if the random sample is small.

6. NUMERICAL RESULTS

In this chapter we apply the transmission probability estimation given in chapter 4 to data described in chapter 5. The results are divided into four parts: the first section contains the results from data with constant transmission probabilities and the second section contains the results from data with varying probabilities. Lastly, the third section discusses how the graph type, order and sample size affected the results overall.

The estimation quality is evaluated through calculating the mean squared error (MSE) for each set of estimates in comparison to their actual values. Let $p_{u,v}$ be the real value for transmission probability for edge (u, v) and $\hat{p}_{u,v}$ be its estimate. The MSE of a transmission probability estimation for graph G is therefore

$$\frac{1}{||G||} \sum_{(u,v) \in G} (p_{u,v} - \hat{p}_{u,v})^2 \quad (6.1)$$

The MSE values will be compared between the different graph orders, types and sample sizes.

In addition to calculating MSEs for the estimation results, we also derive a threshold MSE value for each graph type. The threshold value in this study is the Monte Carlo mean estimate for a graph with randomly assigned transmission probabilities. By using a MSE estimate for randomly chosen transmission probabilities, we get a baseline for assessing the gain in accuracy when using the estimation method. For the threshold values were calculated via 10^4 Monte Carlo simulations per graph type.

6.1 Constant transmission probabilities

This subsection contains the estimation results for graphs for which the edge probabilities are assigned one common value $\beta = 0.5$. The resulting MSE values for star-ring graphs are drawn in Figure 6.1, and the results for block graphs with three and four blocks are drawn in Figures 6.2 and 6.3 respectively.

We see that for all graph types, both the homogeneous (green) and heterogeneous (pink) estimation method seems to perform equally to or better than the threshold value (violet). When we compare the two methods with each other, we see that the homogeneous

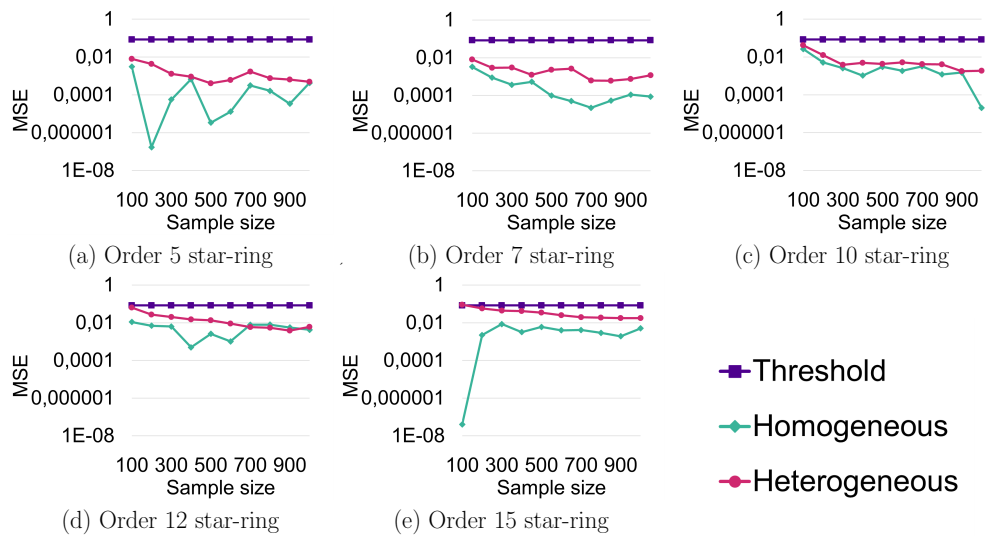


Figure 6.1. MSE values for star-ring graph transmission probability estimation. (Constant transmission probability)

method outperforms the heterogeneous one. In some occasions, the heterogeneous method struggles to get significantly better results compared to the threshold: for example, the MSE values of the 15-order block graph with four blocks in Figure 6.3c does not differ much from the threshold value even when the sample size increases.

On average, the threshold had a MSE value of 0.083, the homogeneous method a value of 0.003 and the heterogeneous method a value of 0.016. The homogeneous method is thus clearly superior to the other methods, and it was able to achieve a MSE value of at least 0.01 for almost all sample sizes in all graph types.

The reason that the homogeneous method outperforms the heterogeneous method here is most likely the fact that the assumption of common constant transmission probability is accurate in this case. Furthermore, the minimization algorithm's search space for the homogeneous method is much smaller than for the heterogeneous method, therefore it is easier for the algorithm to find an ideal local minimum.

If we look at the graphs in Figures 6.1, 6.2 and 6.3, the results of the homogeneous method are often much more volatile compared to the heterogeneous method, which gains accuracy in a steady manner when the sample size increases. Thus, the homogeneous method here seems to be very sensitive to the differences between random samples. For example, the lowest MSE values recorded for the block graphs were achieved on their smallest graphs and largest sample sizes (MSE 2.9E-8 and 2.0E-8), but the star-ring graph had its smallest error in its 15-order graph with only 100 IC episodes (MSE 4.0E-8).

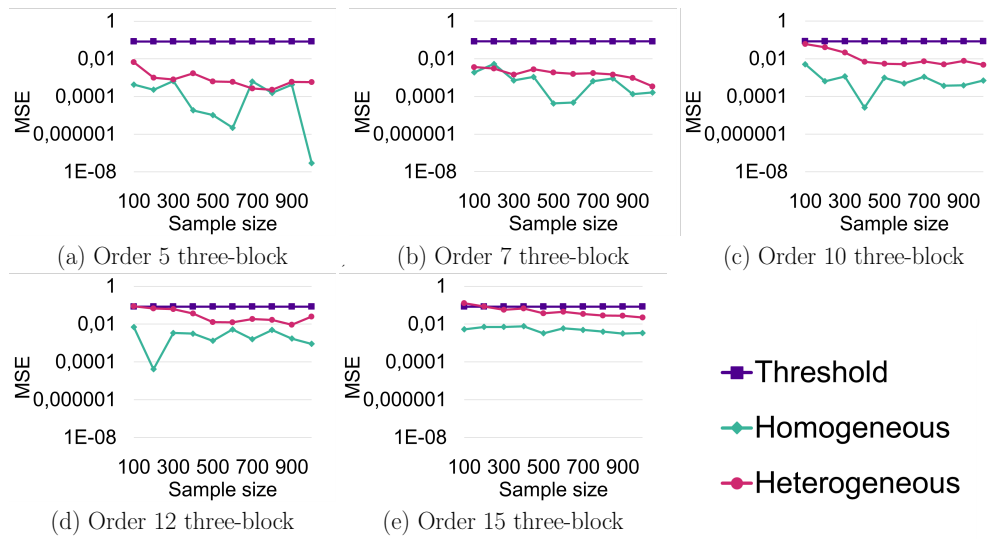


Figure 6.2. MSE values for three-block graph transmission probability estimation. (Constant transmission probability)

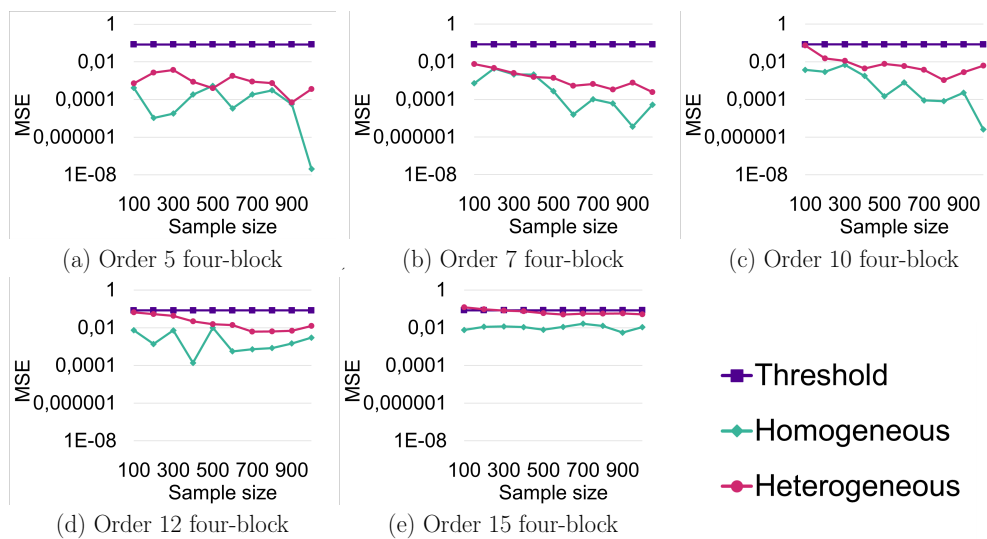


Figure 6.3. MSE values for four-block graph transmission probability estimation. (Constant transmission probability)

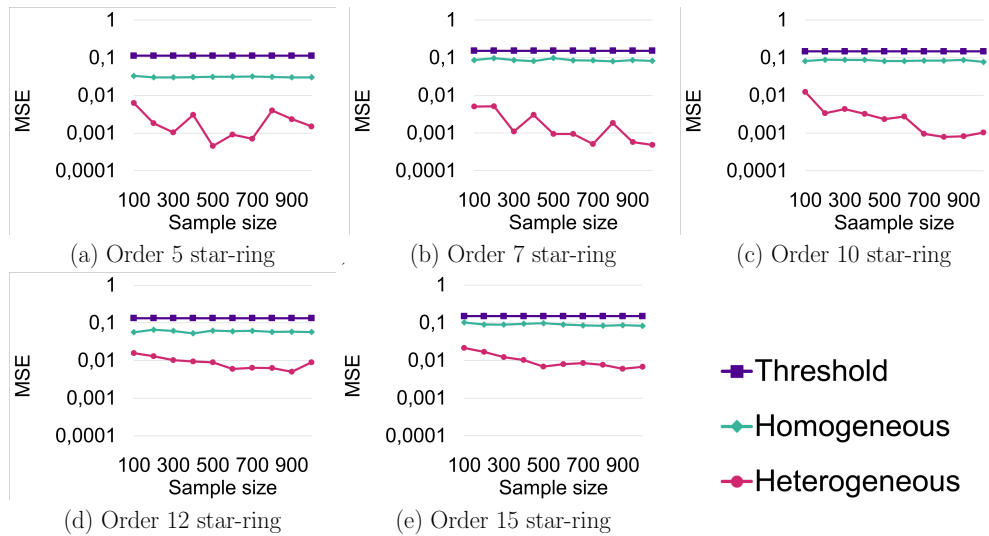


Figure 6.4. MSE values for star-ring graph transmission probability estimation. (Noisy transmission probability)

6.2 Noisy transmission probabilities

This section presents the estimation results for graphs where each of the edges have differing individual transmission probabilities. The MSE values are drawn for each graph type in Figures 6.4, 6.5 and 6.6, similarly to the previous section.

The graphs show that in this case too, the two estimation methods perform at better accuracy than the threshold value. But, contrary to the previous section, now the heterogeneous method clearly has the best performance. When considering all the graph types, orders and sample sizes, the average MSE for the threshold value comes to be 0.144, for the homogeneous method 0.065 and for the heterogeneous method 0.012.

Furthermore, the homogeneous method (blue) now seems to stagnate at a constant error rate, while the MSE values improve for the heterogeneous method (pink) when increasing the sample size. This indicates that there seems to be a singular local minimum for the solver algorithm when we restrict the estimation to a single value, regardless of the sample size.

Though the heterogeneous method clearly produces better results now that the probabilities are not constant, it does not achieve the same level of accuracy as the homogeneous method in section 6.1. Now the lowest MSE recorded are little over 0.001, while the latter method could produce accuracy of well under 0.001. This shows how the added noise in the transmission probabilities increases the complexity of the problem and makes finding ideal local minima more difficult.

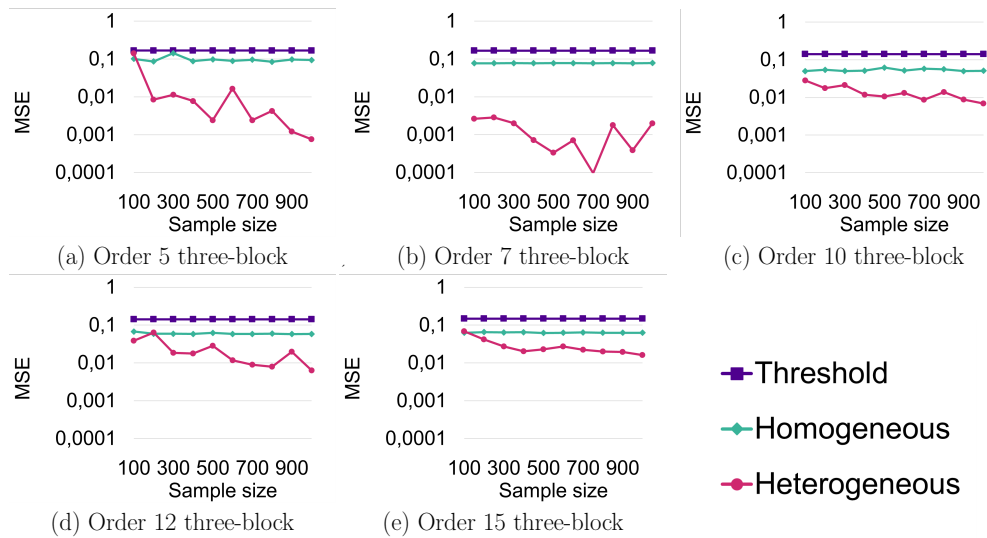


Figure 6.5. MSE values for three-block graph transmission probability estimation. (Noisy transmission probability)

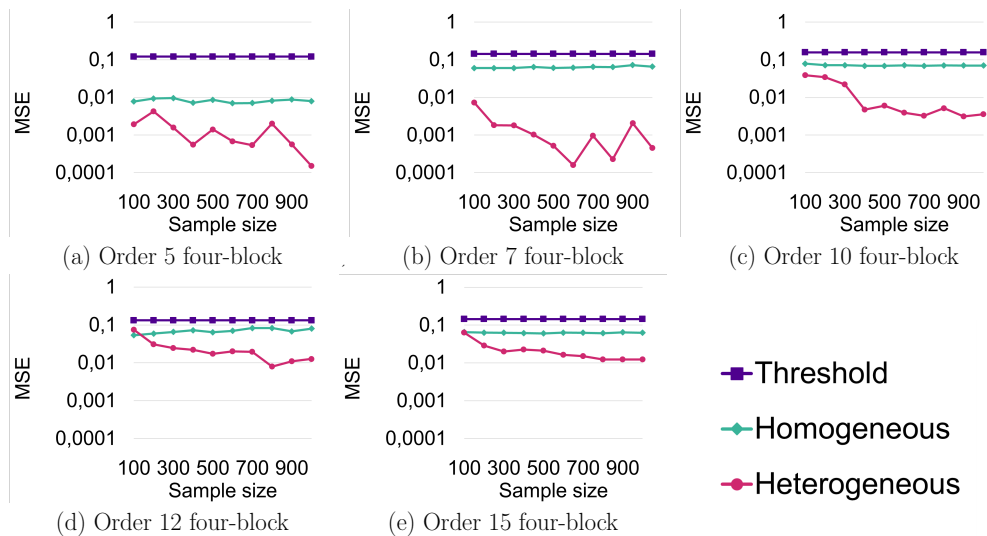


Figure 6.6. MSE values for four-block graph transmission probability estimation. (Noisy transmission probability)

On the other hand, when comparing the performance of the heterogeneous method on constant and varying transmission probabilities, the difference does not seem to be too large: the average MSE for constant data was 0.012 and for varied 0.016. In other words, this suggests that the heterogeneous method, even though more computationally expensive, is able to generate results at the same level of accuracy regardless of the variability of the original parameter values.

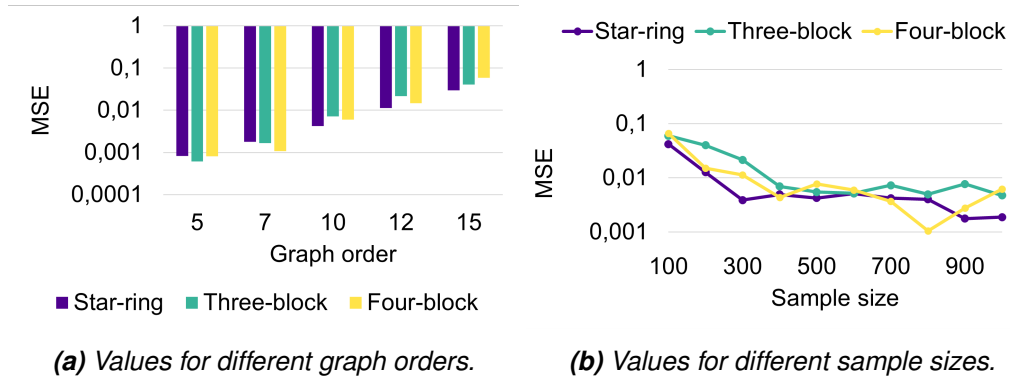


Figure 6.7. Median MSE values for heterogeneous estimation method over every graph type. (Constant transmission probabilities)

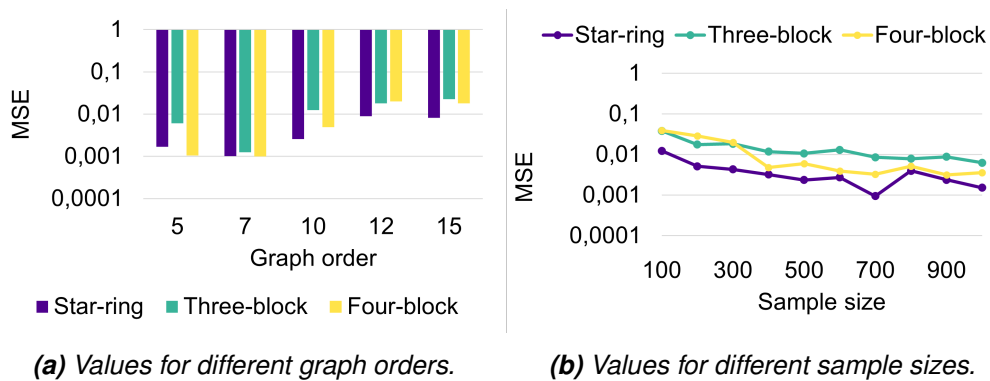


Figure 6.8. Median MSE values for heterogeneous estimation method over every graph type. (Noisy transmission probabilities)

6.3 The effect of graph type, order and sample size

The results in the previous sections already suggested that the order and the number of episodes in the sample have some effect on the method performance. In this section we will look at this effect in more detail.

In Figures 6.7 and 6.8 the median MSE values are drawn for all the graph orders and sample sizes, separated by network type. The first Figure contains the results from constant transmission probabilities and the second the results from noisy transmission probabilities. Both of these results applied to the heterogeneous method.

The results overall suggest that the estimation quality decreases as the graph size increases. This observation is supported by Figures 6.7a and 6.8a, where the median MSE values on average grow when we increase the graph order. It also seems that this trend is quite consistent for all graph types. For example, for the constant transmission probabilities, the median MSE values were closer to 0.001 for 5-order graphs while the values for 15-order graphs were closer to 0.1. In similar fashion, the median values for the noisy transmission probabilities increase from almost 0.001 for the smaller graphs to over 0.01 for the larger graphs.

The reason why the estimation results lose accuracy when increasing the graph order is due to the fact that larger networks usually contain more edges. Therefore, the Markov state space becomes more vast and the algorithm has to fit more parameters, making the discovery of an ideal solution more complicated.

On the other hand, the gain in accuracy over different sample sizes is depicted in Figures 6.7b and 6.8b. We see a decreasing trend in both of the graphs for all network types, therefore increasing the sample size lowers the median MSE. For the constant transmission probabilities, the decrease in MSE from 100 samples to 1000 samples was between 91% to 96% while for the noisy parameter values the decrease was approximately 84%-91%.

There is no particularly strong argument in the data that the method would perform significantly better with one graph type over another, at least when considering only these three graph types. However, there is some evidence in Figure 6.8 that the three-block graph on average results in higher MSE than the other two. Especially when comparing between the different sample sizes, the star-ring graph (purple line) quite consistently has lower median values than the block graphs, while the three-block graph (green line) being slightly higher than the four-block graph (yellow line). This could be explained by the fact that the three-block model usually contains more edges in its graphs compared to the other two (see Table 5.1).

The reason why the method generates somewhat worse estimates on average for the three-block model could be based on the fact that its graphs contained more edges than its four-block and star-ring counterparts (see Table 5.1). This, paired with the fact that also increasing the graph order increases the MSE, would suggest that the number of edges has a much larger effect on the estimation results than the network topology.

7. DISCUSSION

The objective of this thesis was to introduce and validate a new method based on Markov chains to estimate the transmission probabilities of an IC network diffusion model. For this, we tested the presented method on three types of simple network structures with assumptions of both homogeneous and heterogeneous variables. According to these results, the method proves to be effective at estimating the model parameters in adequate accuracy. Therefore, the study can be deemed successful in its objectives.

The test results revealed three aspects which will affect the method's performance. Firstly, the heterogeneous method was observed to perform at equal accuracy for both constant and noisy transmission probabilities, but the homogeneous method performed significantly better when the parameter values were equal. Secondly, increasing the number of episodes was observed to increase the estimation accuracy. And thirdly, a higher graph order and a higher number of edges were observed to lower the accuracy.

The tests were implemented on relatively small, synthetic data sets with high local edge densities. These topologies were proven to be troublesome for the DMP-based estimation method, but this study's method did not show significant bias in its estimates. In other words, this method could prove more useful if one wishes to study the diffusion mechanics within highly connected communities.

Furthermore, the problem of scalability due to the increase of graph edges could be bypassed by utilizing the method's inherent independencies. As the transmission probability between two nodes is only dependent their immediate neighbourhoods, the estimation problem could theoretically be split into subproblems and solved in parallel by identifying suitable subgraphs. To confirm this approach's validity requires more research and testing, but it could be one possible mean to apply the method to large networks without sacrificing accuracy.

There are currently two major limitations with the Markov chain method: First, the method assumes that the network edges are known. If this is not the case, the researcher must assign the connections according to some a priori knowledge. Secondly, the Markov chain estimation also assumes that we are able to observe the IC state of the node at all time points. With incomplete data the researcher must take additional steps to address this.

Therefore, the suggested further development for the method introduced in this paper is to modify it into Hidden Markov model framework. By introducing additional variable information to the model we can make statistical assumptions on the state of a IC network without requiring the complete information on the nodes' status. The tools described in this paper, namely the translation of IC episodes into Markov states and the basic Markov chain based estimation pipeline, should provide a basis for the future possible Hidden Markov model estimation.

Compared to other IC transmission probability estimation methods based on maximum likelihood and dynamic message-passing equations, the Markov Chain method has potential to perform well on networks with short loops and there is not yet any indication it would not work as well on sparser networks. This study only considered relatively small synthetic data for reasons of convenience and accurate method evaluation, but future research using real life data sets should aid with assessing the limits of the method and motivating further improvement.

REFERENCES

- [1] Goldenberg, J., Libai, B. and Muller, E. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review* 9.3 (2001), pp. 1–18.
- [2] Goldenberg, J., Libai, B. and Muller, E. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters* 12 (2001), pp. 211–223.
- [3] Granovetter, M. Threshold models of collective behavior. *American journal of sociology* 83.6 (1978), pp. 1420–1443.
- [4] Saito, K., Nakano, R. and Kimura, M. Prediction of information diffusion probabilities for independent cascade model. *International conference on knowledge-based and intelligent information and engineering systems*. Springer. 2008, pp. 67–75.
- [5] Rodriguez, M. G., Balduzzi, D. and Schölkopf, B. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697* (2011).
- [6] Lamprier, S., Bourigault, S. and Gallinari, P. Influence learning for cascade diffusion models: focus on partial orders of infections. *Social Network Analysis and Mining* 6 (2016), pp. 1–16.
- [7] Mashayekhi, Y., Meybodi, M. R. and Rezvanian, A. Weighted estimation of information diffusion probabilities for independent cascade model. *2018 4th International Conference on Web Research (ICWR)*. IEEE. 2018, pp. 63–69.
- [8] Dutta, R., Mira, A. and Onnela, J.-P. Bayesian inference of spreading processes on networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 474.2215 (2018), p. 20180129.
- [9] Marin, J.-M., Pudlo, P., Robert, C. P. and Ryder, R. J. Approximate Bayesian computational methods. *Statistics and computing* 22.6 (2012), pp. 1167–1180.
- [10] Lokhov, A. Y., Mézard, M. and Zdeborová, L. Dynamic message-passing equations for models with unidirectional dynamics. *Physical Review E* 91.1 (2015), p. 012811.
- [11] Lokhov, A. Reconstructing parameters of spreading models from partial observations. *Advances in Neural Information Processing Systems* 29 (2016).
- [12] Wilinski, M. and Lokhov, A. Prediction-centric learning of independent cascade dynamics from partial observations. *International Conference on Machine Learning*. PMLR. 2021, pp. 11182–11192.

- [13] Gao, F., Zhang, J. and Zhang, Y. Neural enhanced dynamic message passing. *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 10471–10482.
- [14] Barbieri, N., Bonchi, F. and Manco, G. Topic-aware social influence propagation models. *Knowledge and information systems* 37 (2013), pp. 555–584.
- [15] Lagnier, C., Denoyer, L., Gaussier, E. and Gallinari, P. Predicting information diffusion in social networks using content and user's profiles. *Advances in Information Retrieval: 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings 35*. Springer. 2013, pp. 74–85.
- [16] Wang, Z., Zhao, J. and Xu, K. Emotion-based Independent Cascade model for information propagation in online social media. *2016 13th international conference on service systems and service management (ICSSSM)*. IEEE. 2016, pp. 1–6.
- [17] Varshney, D., Kumar, S. and Gupta, V. Predicting information diffusion probabilities in social networks: A Bayesian networks based approach. *Knowledge-Based Systems* 133 (2017), pp. 66–76.
- [18] Diestel, R. Graph theory 3rd ed. *Graduate texts in mathematics* (2005).
- [19] Kermack, W. and McKendrick, A. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A* 115 (1927), pp. 700–721.
- [20] Cannarella, J. and Spechler, J. A. *Epidemiological modeling of online social network dynamics*. 2014. arXiv: 1401.4208 [cs.SI].
- [21] Woo, J., Son, J. and Chen, H. An SIR model for violent topic diffusion in social media. *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*. 2011, pp. 15–19. DOI: 10.1109/ISI.2011.5984043.
- [22] Fu, L., Song, W., Lv, W. and Lo, S. Simulation of emotional contagion using modified SIR model: A cellular automaton approach. *Physica A: Statistical Mechanics and its Applications* 405 (2014), pp. 380–391. DOI: <https://doi.org/10.1016/j.physa.2014.03.043>.
- [23] Ye, Y., Chen, Y. and Han, W. Influence maximization in social networks: Theories, methods and challenges. *Array* 16 (2022), p. 100264. ISSN: 2590-0056. DOI: <https://doi.org/10.1016/j.array.2022.100264>.
- [24] Shakarian, P., Bhatnagar, A., Aleali, A., Shaabani, E. and Guo, R. The Independent Cascade and Linear Threshold Models. *Diffusion in Social Networks*. Cham: Springer International Publishing, 2015, pp. 35–48. DOI: 10.1007/978-3-319-23105-1_4.
- [25] Gagniuc, P. A. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017, pp. 1–8.
- [26] MacDonald, I. L. and Zucchini, W. *Hidden Markov and other models for discrete-valued time series*. Vol. 110. CRC Press, 1997, pp. 6–11.
- [27] MacDonald, I. L. and Zucchini, W. *Hidden Markov and other models for discrete-valued time series*. Vol. 110. CRC Press, 1997, pp. 65–69.

- [28] The SciPy Community. *scipy.optimize.least_squares – SciPy v1.13.0 Manual*. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html#scipy.optimize.least_squares (visited on 04/24/2024).
- [29] Branch, M. A., Coleman, T. F. and Li, Y. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing* 21.1 (1999), pp. 1–23.
- [30] NumPy Developers. *numpy.random.normal – NumPy v1.26 Manual*. URL: <https://numpy.org/doc/stable/reference/random/generated/numpy.random.normal.html> (visited on 04/25/2024).