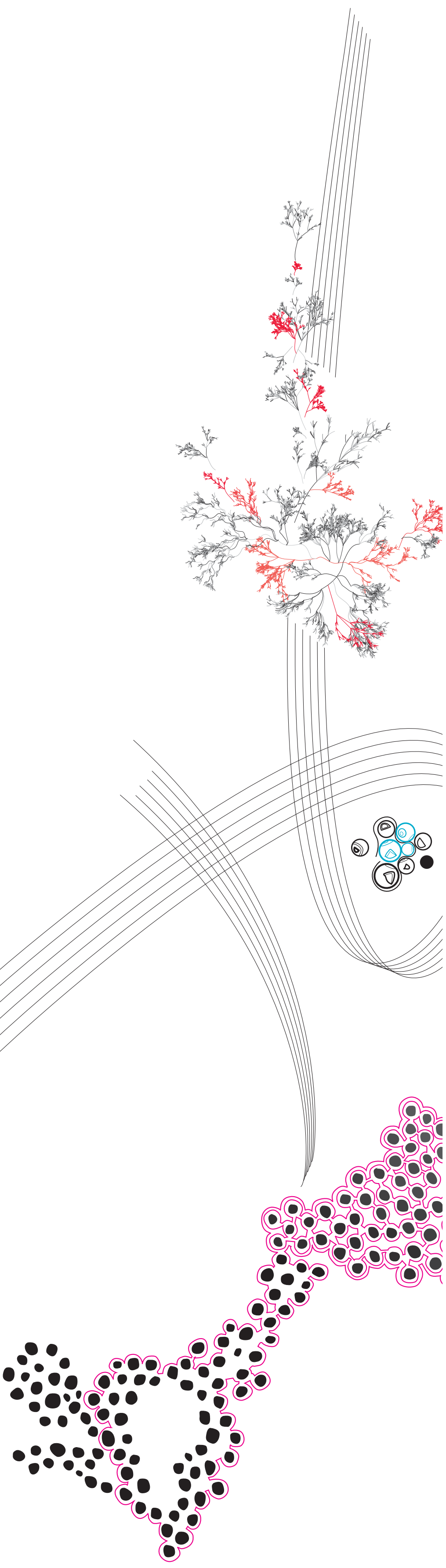

MSc Thesis Applied Mathematics

# Analysis of the Weighted Clique Heuristic for Community Detection using Cascade Data

Andreas Georgiou

Supervisor: Nelly Litvak

May , 2022

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

## Preface

This project was written as the final assignment of my masters degree in Applied mathematics in the university of Twente. Due to the ongoing (at the time) pandemic, most of the time researching and writing it was spend in Greece. I want to thank my supervisor Nelly Litvak for her patience and understandig as well as for agreeing to supervise me online despite the inherit difficulties. Finally I want to thank my friends and family for their support during this period.

Andreas Georgiou

# Analysis of the Weighted Clique Heuristic for Community Detection using Cascade Data

A. Georgiou[*]

May , 2022

[*]Email: a.georgiou@student.utwente.nl

## Abstract

In this study we investigate why weight assignment algorithms that utilise cascade data are suitable for community detection. In combination with the above, the most probable way of cascade propagation in a network with community structure is also examined. The SI epidemic model is used for the cascades. Intuitively enough, we find out that given sufficient rates, the SI epidemic is more likely to infect every individual in the community it originated before proceeding to the other community. Moreover, under this type of epidemic the weights assigned to the edges of the network describe meaningful communities and as a result can be used by a community detection algorithm.

*Keywords*: Community detection, Cascade data, SI model.

## Contents

## 1 Introduction

Being applicable in a vast amount of fields, ranging from marketing all the way to epidemiology, it is not surprising that community detection has received a lot of attention in the field on network science in the last 20 years.

Networks consist of vertices and edges. In some networks the way the vertices are connected through the edges hints at an interesting underlying structure, which we call a community. A community consists of a group of vertices that are densely connected internally. The most common example of network is a social network, where the vertices represent people and an edge between two vertices represent a relation, the most frequent being friendship. If we observe the social network of a individual, it is highly likely that his friends are divided into different, possibly non-overlapping, groups (school, work, university, etc). These groups are what we call the communities of this network and the goal of community detection is to be able to find groups of this kind in any given network.

Community detection itself is an ill-posed problem. We do not know how many, if any, communities are on a network. Moreover communities can manifest in many different ways, overlapping, disjoint, etc. For this reason many heuristics for community detection have been developed in the last 20 years, most of them summarized in [9], [10]. However due to the nature of the problem, the ground truth communities are required to verify the effectiveness of a method. For that reason, benchmark networks have been established to test new and existing methods alike [17].

The methods mentioned in the previous paragraph have the disadvantage of requiring complete knowledge of the network. However in many cases this knowledge is not available. It is common practice to model and study infectious diseases on networks [16], [15]. Moreover, it has been studied how the community structure of a network affects the spreading of a disease [13]. However when modeling human behaviour and relations on networks it is highly unlikely that the connections (edges) between the vertices (individuals) are known. Should the communities of such a graph need to be found, a alternative approach is needed.

In [2], cascade data are used in order to find the communities of the network. Cascade data contain information propagating through the network. For instance in this project the cascades we use are infectious diseases. If we cannot observe the network, but rather only a cascade that propagates through it, can we infer its communities?

Many papers about community detection through cascades have been published in the last decade. Many of which [6], [11], [12], [20] present new methods and algorithms. Our project draws its main motivation from [20]. In this paper a new heuristic for community detection through cascade data is proposed: the CLIQUE algorithm. Instead of inferring the communities of the original network, CLIQUE first creates a surrogate graph. This is done by observing a set of cascades propagating though the network and then weighting any edge that was found.

To the best of our knowledge these heuristics have not been mathematically analysed in the past. The contribution of this work is twofold. First we seek insight and mathematical justification onto why CLIQUE and perhaps other weight assignment algorithms for community detection might work. Second, we investigate which is the most probable way that a cascade can propagate into a network with community structure, and how this way of cascade propagation affects community detection algorithms.

The report is structured as follows. In Chapter 2 we present the necessary background material. In Chapter 3 we discuss about the setup of the project and we introduce the problem. Afterwards, in Chapter 4 numerical results are presented and discussed. The two following chapters contain the major bulk of theoretical results. In Chapter 5 we study the analytical properties of the weights that were introduced in Chapter 3, and in Chapter 6 we investigate which is the most probable order that an SI epidemic can spread in a network with community structure. In Chapter 7 we use the results from Chapter 6 to expand the work done in Chapter 5. Chapter 8 discusses small perturbations to the order established in Chapter 6. The report is concluded with a discussion and a reflection on the results.

# 2 Background and Preliminaries

This chapter serves to introduce background material that will be used throughout the report. First we introduce the SI epidemic model on a network with community structure. As we will see in Chapter 3, our networks will always have two communities and the cascades we will be using will be SI epidemics.

Later we will have to work with increments of infection times. These infection times will always be hypoexponentially distributed. As of this we present an overview of this distribution. Moreover in Chapter 5 we will need to calculate the Laplace transform of these increments. For this reason we will give the definition of the Laplace transform of a general continuous random variable, as well as the Laplace transform of a hypoexponential random variable.

## 2.1 Community based SI process

SI process is one of the many compartmental models in epidemiology. These models assign labels to the individuals. In SI processes the label of an individual can either be susceptible (S) or Infected (I). Susceptible individuals can become infected, but infected individuals stay infected forever [16].

In this project we assume that the population is divided into two non overlapping communities $C_q,\ q = 1, 2$. We denote as $S_q, I_q, N_q$ the susceptible, infected and total population of each community. Furthermore we have two infection rates. Individuals of the same community infect each other with rate $\lambda_{in}$. While individuals that belong to different communities infect each other with rate $\lambda_{out}$. It is a natural assumption that $\lambda_{in} > \lambda_{out}$.

Finally the way in which the process evolves in discrete time is explained by the following equations:

$$\begin{aligned} S_1^{\{t+1\}} &= -\lambda_{in} I_1^{\{t\}} \frac{S_1^{\{t\}}}{N_1} - \lambda_{out} I_2^{\{t\}} \frac{S_1^{\{t\}}}{N_1}, \\ S_2^{\{t+1\}} &= -\lambda_{in} I_2^{\{t\}} \frac{S_2^{\{t\}}}{N_2} - \lambda_{out} I_1^{\{t\}} \frac{S_2^{\{t\}}}{N_2}. \end{aligned} \tag{1}$$

## 2.2 Hypoexponential Distribution

**Definition 2.1** (Hypoexponential distribution)**.** Let $X_k,\ k = 1, 2, .., \nu,\ \nu \in \mathbb{N}$ be independent exponentially distributed random variables each with rate $\lambda_k$. Then $X = \sum_{k=1}^{\nu} X_k$ is hypoexponentially distributed with rate $(\lambda_1, \ldots, \lambda_\nu)$. We denoted it as $X \sim Hypo(\lambda_1, \ldots, \lambda_\nu)$.

**Definition 2.2** (Laplace transform)**.** Let $T$ be a continuous random variable with density $f_T(y)$. Then the Laplace transform of $T$ is defined as:

$$\mathcal{L}\{T\}(y) = \int_0^\infty f_T(t) e^{-yt} dt,\ t \in \mathbb{R}^+, y \in \mathbb{C}.$$

Let $T$ be an exponentially distributed random variable with rate $\lambda$. Then the Laplace transform of $T$ is given by:

$$\mathcal{L}\{T\}(y) = \frac{\lambda}{y+\lambda}.$$

Let $T_k,\ k = 1, 2, .., \nu,\ \nu \in \mathbb{N}$ be independent exponentially distributed random variables each with rate $\lambda_k$. Then the Laplace transform of $T = \sum_{k=1}^{\nu} T_k$ is given by:

$$\mathcal{L}\{T\}(y) = \prod_{k=1}^{\nu} \mathcal{L}\{T_k\}(y) = \prod_{k=1}^{\nu} \frac{\lambda_k}{y+\lambda_k}.$$

## 3 Problem Description

In this section we present the general setup of our research. We begin by giving some already known definitions as a reminder. Then, we proceed by defining some new quantities that we will be using throughout the project. Finally we introduce our main goal.

### 3.1 Setup

**Definition 3.1** (Simple Graph)**.** Let $G(V, E)$ be a graph, $V$ a set of vertices and $E$ a set of edges. $G$ is simple when there is no more than one edge between any two vertices and no edge has both its endpoints on the same vertex.

**Definition 3.2** (Complete Graph)**.** A simple graph $G(V, E)$ is complete when all possible pairs of vertices are connected with an edge.

In this project we will focus on complete graphs. We do that due to the many benefits they offer. We are dealing with infections and on a complete graph each vertex can infect any other vertex. Moreover, since the problem we are working on is very hard, it is only natural to study it on complete graphs as a first step.

Furthermore, we assume that the vertices of $G$ are separated into two non-overlapping communities, $C_1$ and $C_2$. That is, we assume that there exists a partition of the vertex set $V$, $\mathbb{A} = \{C_1, C_2\}$ , $|C_1| = N_1$, $|C_2| = N_2$, such that $C_1 \cup C_2 = V$, $C_1 \cap C_2 = \emptyset$ . Of course there can be a number of ways to partition the set of vertices, but here we consider the simplest one. We deliberately choose to simplify our setup as much as possible so that we can focus on the problem without having to worry about technical issues that might arise from a complicated setup.

### 3.2 Cascades and Infection Order

**Definition 3.3** (Cascade)**.** A cascade $c$ that propagates on a graph $G(V, E)$ with $|V| = N \in \mathbb{N}^*$ nodes is a record of vertices and activation times. I.e. $c = (i, t_i)_{i=1}^{N}$, $i \in V$ and $t_i$ being the infection time of vertex $i$ in the cascade.

We observe a set $\mathcal{C}$ of SI cascades propagating through $G$. Each cascade $c \in \mathcal{C}$ starts from a uniformly picked vertex and lasts until all vertices of $G$ have become infected. In our case, only information of the community of the initial vertex is relevant. The particular vertex that starts the process does not need to be specified as all vertices of a community are interchangeable, since we are working on a complete graph. Each infected vertex transmits the disease to its neighbours with rate $\lambda_{in}$ if they belong in the same community and with rate $\lambda_{out}$ if they do not. Recall from Chapter 2 that $\lambda_{in} > \lambda_{out}$.

Since vertices of the same community are interchangeable in the symmetric complete graph, the order in which the cascade $c$ evolved in $G$ can be described completely through a $N \times 1$ vector $O$ which we define as:

**Definition 3.4** (Infection Order)**.** For a cascade $c$ on a two communities graph $G(V, E)$ , with $|V| = N \in \mathbb{N}^*$, we define the infection order $O \in \{1, 2\}^N$ on the vertices of $G$ as $O = \big(Q(1),\ Q(2),\ \dots,\ Q(N)\big)$. Where

$$Q(i) = \begin{cases} 1 & i \in C_1, \\ 2 & i \in C_2. \end{cases}$$

Where the vertices are numbered as in $c$.

**Definition 3.5** (Smooth Order)**.** An infection order $O$ is called *smooth* and denoted by $O^*$, when it has $N_1$ ones followed by $N_2$ two's or vice versa. $O^*$ corresponds to the cascade where one community is completely infected before any transmission to the other community occurs.

Note that many different cascades result in the same $O$. Moreover we denote by $t_i$ the time vertex $i$ became infected in $c$. Finally for any $i, j \in V$, we use the relation $i \prec j$ to indicate that $i$ was infected before $j$. Subsequently, $i \prec j \iff t_i < t_j$.

## 3.3 Weights

After observing each $c$, we assign a weight to each edge $\{i, j\}$ , $i, j \in V,\ i \prec j$ that could have transmitted the disease. We denote this weight as $w(i, j)$ and define it as in [20] as:

$$w(i, j) = \frac{e^{-\alpha\Delta_{i,j}}}{\sum_{l:l\prec j} e^{-\alpha\Delta_{l,j}}},$$

where $\Delta_{i,j} = |t_i - t_j|$ and $\alpha \in \mathbb{R}_+^*$is a parameter that regulates how much the order in which the vertices where infected affects the value of $w(i, j)$.

Observe that the weight we just defined is random and also that the randomness of $w(i, j)$ is twofold. One source of randomness is that the times between infections, $\Delta$, are random, and the other source is the random order $O$ of the infection. Therefore not only $t_i$ are hypoexponentially distributed random variables, but the rate of their distribution is random as well. To be able to analyze $w(i, j)$ , we must find ways to remove this randomness.

We start with removing the second source of randomness. We do this simply by conditioning on $O$. Doing so immediately makes the distribution of all $\Delta_{i,j}$ known. Moving to the other one, the first natural approach is to examine the expectation of the weights. However taking the expectation leads us to the following expression:

$$\mathbb{E}\left(w(i, j \mid O)\right) = \mathbb{E}\left(\frac{e^{-\alpha\Delta_{i,j}}}{\sum_{l:l\prec j} e^{-\alpha\Delta_{l,j}}}\ |O\right).$$

Here on the right hand size we have expectation of a fraction. Deriving an analytical expression, calculating and working with this quantity is cumbersome. Instead we can work with a first order Taylor approximation. Through this we approximate the expectation of the fraction with the fraction of the expectations. This results in the approximate expected weights conditioned on $O$.

**Definition 3.6** (Approximate expected weights)**.** On a complete graph $G(V, E)$ we define the approximate expected weights between vertices $i,\ j \in V,\ i \prec j$ given the infection order $O$ if as

$$\mu\left(i, j \mid O\right) = \frac{\mathbb{E}(e^{-\alpha\Delta_{i,j}} \mid O)}{\sum_{l:l\prec j}\mathbb{E}(e^{-\alpha\Delta_{l,j}} \mid O)}.$$

**Remark 1.** $\mu\left(i, j \mid O\right) \neq \mu\left(j, i \mid O\right)$

**Remark 2.** If $j \prec i$ in $O$ then $\mu\left(i, j \mid O\right) = 0$

Through the approximate expected weights we can define the Total Approximate Expected (TAE) weight of each community. These weights will be the main subject of study of this project.

**Definition 3.7** (TAE weights)**.** On a complete graph $G(V, E)$ with two communities we define the total approximate weights for $C_1,\ C_2$ as well as the weight between them given the infection order $O$ as:

$$W\left(C_1, C_1 \mid O\right) = \frac{2\sum_{i,j\in C_1, i\prec j}\mu\left(i, j \mid O\right)}{N_1(N_1 - 1)}, \tag{2}$$

$$W\left(C_2, C_2 \mid O\right) = \frac{2\sum_{i,j\in C_2, i\prec j}\mu\left(i, j \mid O\right)}{N_2(N_2 - 1)}, \tag{3}$$

$$W\left(C_1, C_2 \mid O\right) = \frac{\sum_{i\in C_1, j\in C_2, i\prec j}\mu\left(i, j \mid O\right) + \sum_{i\in C_2, j\in C_1, i\prec j}\mu\left(i, j \mid O\right)}{N_1N_2}. \tag{4}$$

Even though we have two communities we will refer to all three weights as community weights. We normalize the TAE weights by dividing them with the total number of edges that take part in their calculation, as communities might not be of the same size. For $W(C_1, C_1|O)$ and $W(C_2, C_2|O)$ that would be the total edges that belong to $C_1$ and $C_2$, which are $N_1(N_1-1)/2$ and $N_2(N_2-1)/2$ respectively. As for $W(C_1, C_2|O)$ it is $N_1N_2$ edges. Since each total weight is divided by the number of edges, what we actually compute is an approximate mean weight per edge. Furthermore, the conditioning on $O$ will be omitted whenever there is no ambiguity.

### 3.4 Problem Statement

The primary goal of this project is to investigate if and under which infection orders, circumstances and conditions, for the observed set of cascades $\mathcal{C}$ it holds that:

$$\sum_{c\in\mathcal{C}} W(C_1, C_1|O) > \sum_{c\in\mathcal{C}} W(C_1, C_2|O), \tag{5}$$

$$\sum_{c\in\mathcal{C}} W(C_2, C_2|O) > \sum_{c\in\mathcal{C}} W(C_1, C_2|O). \tag{6}$$

This is equivalent to stating that the total edge weights inside the communities are greater than the total edge weights between the communities. Taking into consideration that many cascade based community detection algorithms like the ones used in [20] use weight assignment, proving this is a important step towards understanding how they work.

## 4 Numerical Examples

In this section we will numerically compute the TAE weights (2)-(4). After the computation we will compare their values with each other in an attempt to verify whether inequalities (5), (6) hold. We do this on two separate graphs, $K_6$ and $K_{50}$ for various infection orders and visualise the results.

### 4.1 Infection times

The TAE weights are composed by the approximate expected weights. Recall that the approximate expected weights are defined as:

$$\mu(i,j \mid O) = \frac{\mathbb{E}(e^{-\alpha\Delta_{i,j}} \mid O)}{\sum_{l:l\prec j} \mathbb{E}(e^{-\alpha\Delta_{l,j}} \mid O)}.$$

In order to numerically compute the TAE weights we must compute the expectation of the $e^{-\alpha\Delta_{i,j}}$ terms. To do this we need to know the distribution of the infection times $t_i$.

To that end we first present a well known result from probability theory.

**Lemma 1.** Let $X_k,\ k = 1,2,..,\nu$ be independent exponentially distributed random variables each with rate $\lambda_k$. Then if $X = min\{X_1, X_2, \ldots, X_\nu\}$, it is exponentially distributed with rate $\sum_{k=1}^{\nu} \lambda_k$.

Recall that $S_q,\ I_q,\ q = 1,2$ denote the number of susceptible and infected vertices in community $C_q$. We will use this Lemma to prove the following theorem.

**Theorem 3.** Consider a cascade on a complete graph with two communities $C_1$ and $C_2$. Then conditioned on the infection order $O$, the infection time of a vertex $i,\ t_i$, is distributed as:

$$t_i \sim Exp\big(\lambda_{in}(S_1I_1 + S_2I_2) + \lambda_{out}(S_1I_2 + S_2I_1)\big) + t_{i-1}.$$

*Proof.* Every infected vertex is trying to infect a susceptible vertex. However a susceptible vertex will be infected by only one of the already infected vertices. The independent identically distributed random variables $T^j_{in}$ who express the time until it gets infected by vertex $j$ of the same community are exponentially distributed with rate $\lambda_{in}$. The independent identically distributed random variables $T^j_{out}$ who express the time until it gets infected by vertex $j$ of the other community are exponentially distributed with rate $\lambda_{out}$. Moreover at any given time the $I_1 + I_2$ infected vertices are infecting each of the $S_1 + S_2$ susceptible vertices. W.l.o.g we can assume $i \in C_1$. If we want vertex $i$ to be the next infected we require:

$$t_i = min\{T^1_{in}, T^2_{in}, \ldots, T^{I_1}_{in}, , T^1_{out}, .., T^{I_2}_{out}\} + t_{i-1},$$

where $t_{i-1}$ is the infection time of vertex $i-1$. In the expression above there are $S_1I_1 + S_2I_2$ $T_{in}$ terms and $S_1I_2 + S_2I_1$ $T_{out}$ terms. Using this together with Lemma 1 completes the proof. $\square$

**Remark 4.** Unless we condition on a order of infections $O$ the rate at which $t_i$ is distributed is random as $S_1,\ S_2,\ I_1,\ I_2$ are not deterministic, since $t_i \sim Exp(f(S_1, S_2, I_1, I_2)) + t_{i-1}$.

## 4.2 Setup for the numerical calculations

To be able to perform the numerical calculations of the TAE weights, we need to fix most of the quantities. First of all we set $N_1 = N_2 = N/2$. We chose to have equally sized communities so that the results are not biased due to the size. Moreover in the following chapters we will explicitly require the two communities to have the same size. Thus, requiring it here as well makes the report more consistent. By fixing the community sizes we know exactly how many weights each community has. We also fix the infection rates as $\lambda_{in} = 0.66$ and $\lambda_{out} = 0.33$. These numbers are picked for convenience. Any pair of $\lambda_{in}, \lambda_{out}$ would suffice as long as $\lambda_{in}/\lambda_{out} > 1$. We will elaborate on this in Chapter 6.

As a result of fixing the infection rates, the TAE weights become functions of only $\alpha$. As of this we will calculate the TAE weights for $\alpha$ varying from 0 to 100 with a unit step at a time. Unit step gives sufficient information on how the TAE weights depend on $\alpha$. However, had we had a more efficient code, we would be able to study even smaller increments of $\alpha$.

Before we proceed with the presentation of the results we want to point out that we do not expect inequalities (5) and (6) to hold for $\alpha = 0$, regardless of the infection order. That is because when $\alpha = 0$ each weight $\mu\big(i, j \mid O\big)$ is of the form $1/\sum_{i \prec j} 1$. Thus for $\alpha = 0$ the weight of an edge $\{i, j\}$ is just the probability that $j$ was infected by one of the already infected vertices. That makes every already infected vertex equally likely to infect $j$ regardless of the community they are in, and that contradicts what we are trying to show.

## 4.3 Results for single Orders

We will start by computing TAE weights given stand alone orders $O$. First we start with $K_6$, as it is the smallest complete graph where a community structure is meaningful. Moreover the results we obtain for $K_6$ can be "easily" verified, in contrast with the results from $K_{50}$.

The first order we examine is (1,1,1,2,2,2), shown in Figure 1. Recall that we call this a smooth infection order. Afterwards we will start perturbing this smooth order until no two vertices of the same community were infected consecutively. The results are presented in Figure 4.

Figures 1-4 depict the values of the TAE weights plotted against the values of $\alpha$ on $K_6$ for four different orders.

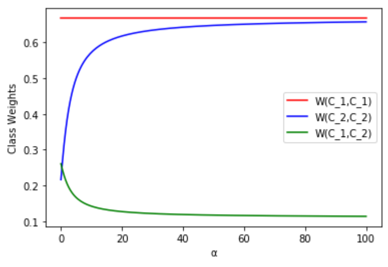

FIGURE 1: Smooth order on $K_6$.

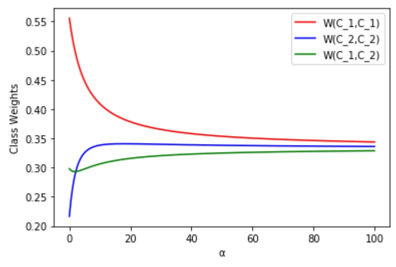

FIGURE 2: $K_6$ with order 112122.

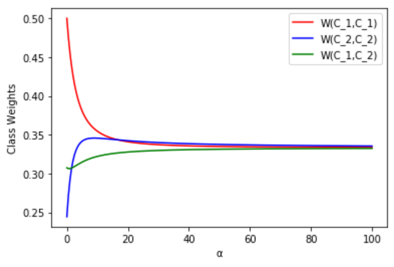

FIGURE 3: $K_6$ with order 112212.

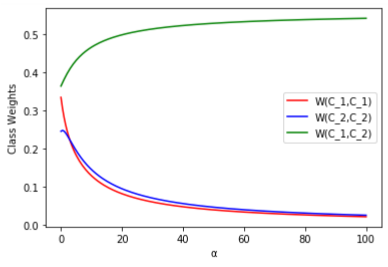

FIGURE 4: $K_6$ with order 121212.

Figures 1-4 show that inequalities (5) and (6) do not always hold. For example, they hold in Figure 3 up until $\alpha$ exceeds some value, while on 4 they do not hold at all. Specifically in Figure 4 we expected weight (4) to be the largest, due to the structure of the order $(1, 2, 1, 2, 1, 2)$. This order is always alternating between communities and that points away from a community structure. Moreover, even for the orders for which our inequalities hold, they only hold for $\alpha$ that exceed some specific value, $\alpha^*$.

However $K_6$ is too small and thus some of the results we got from it are misleading. Figures 1-4 suggest that the TAE weights are monotonic in $\alpha$. We will see bellow that this is not always true. For a more accurate description on how the total expected approximate weights behave we consider a bigger network, $K_{50}$.

For $K_{50}$ we again compute the TAE weights (2) - (4) for four different orders. The first one is a smooth order, Figure 5. The second is identical to the first with the only difference being that the last vertex of $C_1$ is moved to the end of the order, Figure 6. The third order has the vertices of $C_1$ split into two groups around the nodes of $C_2$, Figure 7. Finally the last order alternates between the communities, Figure 8.

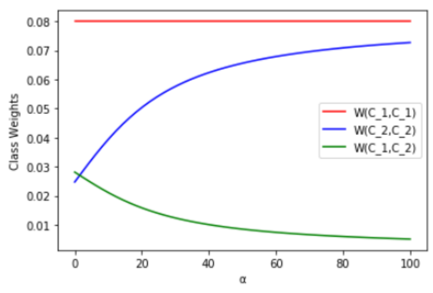

FIGURE 5: Smooth order on $K_{50}$.

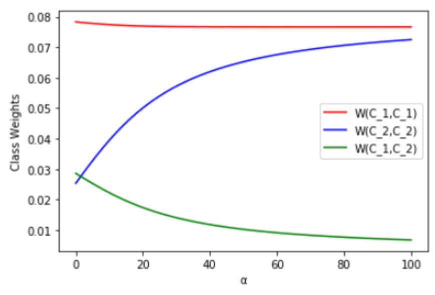

FIGURE 6: Small perturbation on the smooth order on $K_{50}$.

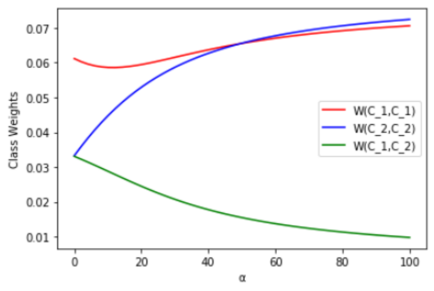

FIGURE 7: A bigger perturbation on the smooth order on $K_{50}$.

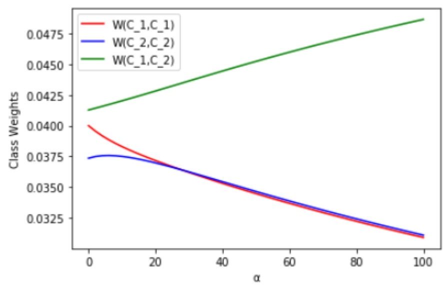

FIGURE 8: Alternating order on $K_{50}$.

The TAE weight of $C_1$, $W(C_1, C_1)$ is not monotone in Figure 7. We could not observe this in the smaller graph. Moreover, inequalities (5), (6) still hold for the smooth order.

## 4.4 Results on averaging on all possible Orders

Now let us assume that $\mathcal{C}$ contains all the $N!/N_1!N_2!$ possible orders $O$ of an SI cascade on $K_N$. We will calculate (2)-(4) for every $c \in \mathcal{C}$ and sum the corresponding weights together. Unfortunately due to how rapidly the number of permutations increase with $N$, we are only able to work on small graphs here. The two graphs we consider are $K_6$ and $K_{10}$. $\sum_{c\in\mathcal{C}} W(C_1, C_1), \sum_{c\in\mathcal{C}} W(C_2, C_2), \sum_{c\in\mathcal{C}} W(C_1, C_2)$ are shown in the Figures bellow.

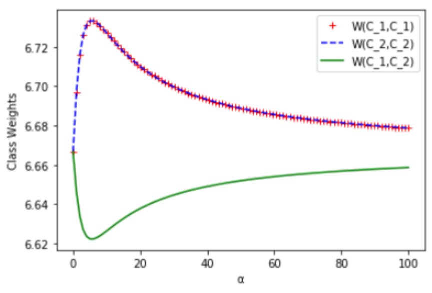

FIGURE 9: $K_6$.

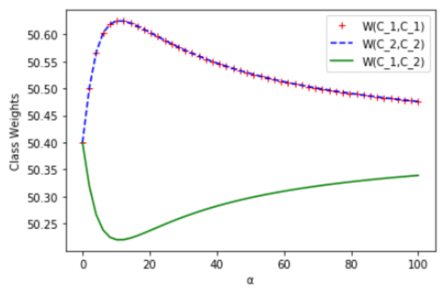

FIGURE 10: $K_{10}$.

As shown in Figures 9, 10 above, our inequalities hold when we consider every possible cascade. Although this result appears to be promising, it is only a heuristic justification for community detection form cascade data. This is due to cascades not being equally likely to appear.

In this chapter we numerically computed the TAE weights in an attempt to verify whether inequalities (5), (6) hold. The most interesting results we obtained are the following. First, the weights can be monotonic in $\alpha$ given an order $O$. However this is not true for every $O$. This is an interesting analytical property. The second result is that our inequalities always hold if the infection order is smooth. We will further examine these results in the following chapter. Finally we saw that the weights describe meaningful communities when the vertices of the same community are infected one after another.

# 5 Analytical properties of the Weights

In this chapter we focus on proving theoretical results about the TAE weights. We investigate the limiting behaviour when $\alpha \to \infty$ and the monotonicity. To do this we will first show that the expected weights, $\mu(i,j)$, behave differently when $j = i+1$ and when $j \neq i+1$. Afterwards we introduce a new quantity $\sigma$. With the help of $\sigma$ we present a way to calculate the limits of the TAE weight given any infection order. Afterwards we condition on a smooth infection order and study the monotonicity of the TAE weights. Finally we prove the two inequalities (5) and (6), conditioned on a smooth infection order.

We start by studying the limiting behaviour of the TAE weights when $\alpha \to \infty$. We expect these limits to exist and be real numbers.

## 5.1 Limiting behavior when $\alpha \to \infty$

Recall the definition of the approximate expected weight between vertices $i,\ j$:

$$\mu(i,j) = \frac{\mathbb{E}(e^{-\alpha\Delta_{i,j}})}{\sum_{l:l\prec j}\mathbb{E}(e^{-\alpha\Delta_{l,j}})}.$$

In the previous section we gave a formula for the distribution of the infection time of vertex $i, t_i$ in Theorem 3 and briefly touched upon the distribution of $\Delta_{i,j} = |t_i - t_j|$. When vertex $j$ was infected right after vertex $i$, then $\Delta_{i,j} \sim Exp(\lambda_j)$. Here $\lambda_j$ is an abbreviation for the actual rate. The use of an abbreviation makes our formulas more compact and clear. The actual rate is a function of the number of infected and susceptible vertices in both communities and the corresponding infection rates. When vertices $i,\ j$ are not infected consecutively then $\Delta_{i,j} \sim Hypo(\lambda_{i+1}, \dots,\ \lambda_j)$. For the following we will use the term $\lambda_h,\ h \in \mathbb{N}$ as a abbreviation for the rate of the distribution of $\Delta_{i,h}$. Now we can distinguish two cases for $\mu(i,j)$:

1. $j = i+1$,
2. $j \neq i+1$.

Case 1:

$$\begin{aligned}\mu(i,j) &= \frac{\mathbb{E}(e^{-\alpha\Delta_{i,j}})}{\sum_{l:l\prec j}\mathbb{E}(e^{-\alpha\Delta_{l,j}})}\\ &= \frac{\frac{\lambda_j}{\lambda_j+\alpha}}{\sum_{h=2}^{j}\prod_h^j \frac{\lambda_h}{\lambda_h+\alpha}}\\ &= \frac{1}{1+\sum_{h=2}^{j-1}\prod_h^{j-1}\frac{\lambda_h}{\lambda_h+\alpha}}.\end{aligned}$$

Taking the limit will make the term in the denominator vanish:

$$\lim_{\alpha\to\infty} \frac{1}{1+\sum_{h=2}^{j-1}\prod_{h}^{j-1}\frac{\lambda_h}{\lambda_h+\alpha}} =$$
$$\frac{1}{1+\lim_{\alpha\to\infty}\sum_{h=2}^{j-1}\prod_{h}^{j-1}\frac{\lambda_h}{\lambda_h+\alpha}} =$$
$$\frac{1}{1+0} = 1.$$

In this case $\lim_{\alpha\to\infty}\mu(i,j) = 1$.

Case 2:

$$\begin{aligned}\mu(i,j) &= \frac{\mathbb{E}(e^{-\alpha\Delta_{i,j}})}{\sum_{l:l\prec j}\mathbb{E}(e^{-\alpha\Delta_{l,j}})}\\ &= \frac{\prod_{h=i+1}^{j}\frac{\lambda_h}{\lambda_h+\alpha}}{\sum_{h=2}^{j}\prod_{h}^{j}\frac{\lambda_h}{\lambda_h+\alpha}}\\ &= \begin{cases}\frac{1}{1+\sum_{h=2}^{j-1}\prod_{2}^{h}\frac{\lambda_h+\alpha}{\lambda_h}} & i=1,\\ \frac{1}{1+\sum_{h=2}^{i}\prod_{h}^{i}\frac{\lambda_h}{\lambda_h+\alpha}+\sum_{h=i+1}^{j-1}\prod_{i+1}^{h}\frac{\lambda_h+\alpha}{\lambda_h}} & \text{else.}\end{cases}\end{aligned}$$

We take the limit once more:

$$\begin{aligned}&\lim_{\alpha\to\infty}\begin{cases}\frac{1}{1+\sum_{h=2}^{j-1}\prod_{2}^{h}\frac{\lambda_h+\alpha}{\lambda_h}} & i=1,\\ \frac{1}{1+\sum_{h=2}^{i}\prod_{h}^{i}\frac{\lambda_h}{\lambda_h+\alpha}+\sum_{h=i+1}^{j-1}\prod_{i+1}^{h}\frac{\lambda_h+\alpha}{\lambda_h}} & \text{else.}\end{cases}\\ &=\begin{cases}\frac{1}{1+\lim_{\alpha\to\infty}\sum_{h=2}^{j-1}\prod_{2}^{h}\frac{\lambda_h+\alpha}{\lambda_h}} & i=1,\\ \frac{1}{1+\lim_{\alpha\to\infty}\sum_{h=2}^{i}\prod_{h}^{i}\frac{\lambda_h}{\lambda_h+\alpha}+\lim_{\alpha\to\infty}\sum_{h=i+1}^{j-1}\prod_{i+1}^{h}\frac{\lambda_h+\alpha}{\lambda_h}} & \text{else.}\end{cases}\\ &=0.\end{aligned}$$

Since:

$$\lim_{\alpha\to\infty}\sum_{h=2}^{j-1}\prod_{2}^{h}\frac{\lambda_h+\alpha}{\lambda_h}\to\infty,$$

and

$$\lim_{\alpha\to\infty}\sum_{h=2}^{i}\prod_{h}^{i}\frac{\lambda_h}{\lambda_h+\alpha}=0.$$

Thus in this case $\lim_{\alpha\to\infty}\mu(i,j)=0$.

So we have found out how the individual approximate expected weights behave when $\alpha$ tends to infinity. That is, we found out that only the $\mu(i,j)$ defined on consecutive vertices do not vanish and they tend to 1. We will now define $\sigma$.

**Definition 5.1** (Infection counter)**.** Given an infection order $O$ and communities $C_{q_1},\ C_{q_2},\ q_1, q_2\ = 1, 2$ we define $\sigma(C_{q_1}, C_{q_2}|O)$ as a count of how many times the infection moved from $C_{q_1}$ to $C_{q_2}$.

Note that for a community, the number of $\mu(i,j)$ defined on consecutive vertices is precisely $\sigma$. Thus we can conclude the following:

$$\begin{aligned}
\lim_{\alpha\to\infty} W(C_1,C_1) &= \frac{\sigma(C_1,C_1)}{N_1(N_1-1)/2}, \\
\lim_{\alpha\to\infty} W(C_2,C_2) &= \frac{\sigma(C_2,C_2)}{N_2(N_2-1)/2}, \\
\lim_{\alpha\to\infty} W(C_1,C_2) &= \frac{\sigma(C_1,C_2)}{N_1N_2}.
\end{aligned} \tag{7}$$

And in case of a smooth infection order:

$$\begin{aligned}
\lim_{\alpha\to\infty} W(C_1,C_1|O^*) &= \frac{N_1-1}{N_1(N_1-1)/2} = \frac{2}{N_1}, \\
\lim_{\alpha\to\infty} W(C_2,C_2|O^*) &= \frac{N_2-1}{N_2(N_2-1)/2} = \frac{2}{N_2}, \\
\lim_{\alpha\to\infty} W(C_1,C_2|O^*) &= \frac{1}{N_1N_2}.
\end{aligned}$$

It is clear that in the smooth infection order the limits of the intra-community weights are much larger than the the limit of the inter-community weight. We know that the TAE weights are continuous functions of $\alpha$. Moreover we just found a way to calculate their limiting behaviour for any infection order. A natural way to tie the continuity of the TAE weights with their limiting behaviour is monotonicity.

## 5.2 Monotonicity of the weights

As we observed from the numerical results, the TAE weights are not always monotonic (Figure 7). For that reason we will examine the monotonicity of the TAE weights conditioned on a smooth infection order. We choose the smooth infection order for two reasons. First, as we saw in Figure 1 and Figure 5 the TAE weights should be monotone in the smooth infection order. Second, it reflects best a strong community structure. Since the TAE weights are defined as a summation of approximate expected weights, $\mu(i,j)$, it is only natural to study the monotonicity of these $\mu(i,j)$ first.

We will repeat what we did for the limiting behaviour. We distinguish between two categories of $\mu(i,j)$. Similarly as before the first category is when $j = i + 1$. In that case:

$$\mu(i,j) = \frac{1}{1 + \sum_{h=2}^{j-1} \prod_{h}^{j-1} \frac{\lambda_h}{\lambda_h + \alpha}}.$$

The $\frac{\lambda_h}{\lambda_h+\alpha}$ terms are decreasing. Therefore we can conclude that $\mu(i,j)$ is an increasing function of $\alpha$ when defined on consecutive vertices.

Now in case when $j \neq i+1$:

$$\mu(i,j) = \begin{cases} \frac{1}{1+\sum_{h=2}^{j-1} \prod_{2}^{h} \frac{\lambda_h+\alpha}{\lambda_h}} & i = 1, \\ \frac{1}{1+\sum_{h=2}^{i} \prod_{h}^{i} \frac{\lambda_h}{\lambda_h+\alpha} + \sum_{h=i+1}^{j-1} \prod_{i+1}^{h} \frac{\lambda_h+\alpha}{\lambda_h}} & \text{else.} \end{cases}$$

When $j \neq i+1$ and $i = 1$, $\mu(i,j)$ is a decreasing function of $\alpha$, as the $\frac{\lambda_h+\alpha}{\lambda_h}$ terms are increasing in $\alpha$. However when $j \neq i+1$ and $i \neq 1$ the monotonicity is not so straightforward to infer as the denominator has both increasing and decreasing terms. So we need to analytically study these $\mu(i,j)$. Talking the derivative of such an expression is not helpful at all. Even if we differentiate we have no means of establishing the sign of the derivative. To avoid this we can simply set $\lambda_h = 1$ w.l.o.g. Thus the quantity we need to study is of the form:

$$f(\alpha) = \frac{1}{1 + \sum_{h=1}^{m} (1+\alpha)^{-h} + \sum_{h=1}^{n} (1+\alpha)^{h}}.$$

To get a feeling for the monotonicity we first consider the following heuristic. We can further simplify $f(\alpha)$ by considering only the terms that have the biggest impact. Through this second simplification the original expression reduces to:

$$g(\alpha) = \frac{1}{1 + (1+\alpha)^{-1} + (1+\alpha)^{n}},$$

and

$$g'(\alpha) = \frac{(1+\alpha)^{-2} - n(1+\alpha)^{n-1}}{(1 + (1+\alpha)^{-1} + (1+\alpha)^{n})^2}.$$

Now we have to examine the sign of the derivative.

$$\begin{aligned} &g'(\alpha) < 0 \\ &(1+\alpha)^{-2} < n(1+\alpha)^{n-1} \\ &\frac{1}{n} < \frac{(1+\alpha)^{n-1}}{(1+\alpha)^{-2}} \\ &\frac{1}{n} < (1+\alpha)^{1+n} \\ &\alpha > \sqrt[1+n]{\frac{1}{n}} - 1. \end{aligned}$$

Now let

$$h(n) = \sqrt[1+n]{\frac{1}{n}} - 1,$$

then:

$$h'(n) = \frac{\frac{\ln(x)}{(x+1)^2} - \frac{1}{x\cdot(x+1)}}{x^{\frac{1}{x+1}}}.$$

Ideally we would like to find the maximum value that $h(n)$ can attain, $h_{max}(n)$. Then we could infer that for any $\alpha > h_{max}(n)$, $\mu(i,j)$ is decreasing in this case as well. Unfortunately we cannot analytically solve the equation $h'(n) = 0$. However it sufficient to observe that for any $n \in \mathbb{N}$ the following is true:

$$1 \geq \sqrt[1+n]{\frac{1}{n}}.$$

That means:

$$h(n) \leq 0, \ \forall n \in N.$$

Although it was not a solid proof, this heuristic indicates that if $j \neq i+1$, then $\mu(i,j)$ is a decreasing function of $\alpha$, for any $\alpha > 0$. We will regard this result as true and we will use it whenever needed for the rest of the project. As a consequence, $\mu(i,j)$ is an increasing function of $\alpha$ only when it is defined on consecutive vertices, and at any other case it is decreasing.

## 5.3 Monotonicity of the weights in $O^*$

First we give this basic property of the approximate expected weights in the following lemma:

**Lemma 2.** For the approximate expected weight defined as:
$\mu\big(i,j\big) = \frac{\mathbb{E}\left(e^{-\alpha\Delta_{i,j}}\right)}{\mathbb{E}\left(\sum_{l:l\prec j} e^{-\alpha\Delta_{l,j}}\right)}$, in every cascade $c$ the following holds:

$$\sum_{i\prec j} \mu(i,j) = 1, \forall\ j.$$

This is easy to verify though a simple summation.

*Proof.*

$$\begin{aligned}\sum_{i\prec j} \mu(i,j) &= \sum_{i\prec j} \frac{\mathbb{E}\big(e^{-\alpha\Delta_{i,j}}\big)}{\mathbb{E}\big(\sum_{l:l\prec j} e^{-\alpha\Delta_{l,j}}\big)} \\ &= \frac{\mathbb{E}\big(\sum_{i\prec j} e^{-\alpha\Delta_{i,j}}\big)}{\mathbb{E}\big(\sum_{l:l\prec j} e^{-\alpha\Delta_{l,j}}\big)} \\ &= 1.\end{aligned}$$

□

**Remark 5.** We can use Lemma 2 to rewrite sums that are not equal to 1.

We can write:

$$\sum_{i \prec j} \mu(i,j) = \sum_{i=1}^{j-1} \mu(i,j) = 1.$$

Then for $\beta \in \mathbb{N}^*$:

$$\sum_{i=\beta}^{j-1} \mu(i,j) = 1 - \sum_{i=1}^{\beta} \mu(i,j).$$

In this section we will investigate the monotonicity of the TAE weights conditioned of a smooth infection order. We do this with the help of Lemma 2 and the results of the previous section. We assume the smooth order started from community 1. We start with weight (2). Then we proceed with weight (4) and finally we examine weight (3).

**Theorem 6.** We observe a cascade on a complete graph $K_N$ whose vertices are divided into two non-overlapping communities $C_1,\ C_2,\ |C_1| = N_1$ and $|C_2| = N_2$. The cascade follows a smooth infection order, and we assume that it started in $C_1$. Then:

1. $W(C_1, C_1)$ is a constant.
2. $W(C_2, C_2)$ is increasing in $\alpha$.
3. $W(C_1, C_2)$ is decreasing in $\alpha$.

*Proof.* Recall the definition of TAE weight (2):

$$W(C_1, C_1) = \frac{2 \sum_{i,j \in C_1, i \prec j} \mu(i,j)}{N_1(N_1 - 1)}.$$

Since we want to make use of Lemma 2 we rewrite the numerator of $W(C_1, C_1)$ by breaking apart the big sum into smaller sums. One for each $j \in C_1$.

$$\sum_{i,j \in C_1, i \prec j} \mu(i,j) = \mu(1,2) + \sum_{i \in C_1, i \prec 3} \mu(i,3) + \cdots + \sum_{i \in C_1, i \prec N_1} \mu(i, N_1).$$

According to Lemma 2 each term on the right hand side is equal to 1. Therefore:

$$\sum_{i,j \in C_1, i \prec j} \mu(i,j) = N_1 - 1.$$

Note that it is not equal to $N_1$ since there is no term for vertex 1.

Thus the weight of community 1 can be written as:

$$\begin{aligned} W(C_1, C_1) &= \frac{2\sum_{i,j\in C_1, i\prec j} \mu(i,j)}{N_1(N_1-1)} \\ &= \frac{2(N_1-1)}{N_1(N_1-1)}. \end{aligned}$$

And thus:

$$W(C_1, C_1) = \frac{2}{N_1}, \tag{8}$$

which is a constant.

Now we continue with TAE weight (4). We follow the same approach. We rewrite the numerator by breaking apart the big sum into smaller sums. To avoid confusion we have to clarify that when we work with a smooth infection order that started in $C_1$, the vertices of $C_1$ are $(1, 2, 3, \dots, N_1)$, while the vertices of $C_2$ are $(N_1+1, N_1+2, \dots, N_2)$.

$$\sum_{i\in C_1, j\in C_2, i\prec j} \mu(i,j) = \sum_{i\in C_1} \mu(i, N_1+1) + \sum_{i\in C_1} \mu(i, N_1+2) + \cdots + \sum_{i\in C_1} \mu(i, N_2).$$

The first term of the right hand side satisfies Lemma 2,

$$\sum_{i\in C_1, j\in C_2, i\prec j} \mu(i,j) = 1 + \sum_{i\in C_1} \mu(i, N_1+2) + \cdots + \sum_{i\in C_1} \mu(i, N_2).$$

Hence we can write weight (4) as:

$$W(C_1, C_2) = \frac{1 + \sum_{i\in C_1} \mu(i, N_1+2) + \cdots + \sum_{i\in C_1} \mu(i, N_2)}{N_1 N_2}. \tag{9}$$

Observe that every $\mu(i,j)$ above is defined on non-consecutive vertices and thus they are decreasing functions of $\alpha$. As a result $W(C_1, C_2)$ is a decreasing function of $\alpha$ as well.

Finally we will examine $W(C_2, C_2)$. Instead of working with the weight itself we can rewrite every term on the numerator in relation (9) by using the remark of Lemma 2. This results in the following pattern:

$$\begin{aligned} &\sum_{i\in C_1} \mu(i, N_1+2) = 1 - \mu(N_1+1, N_1+2), \\ &\sum_{i\in C_1} \mu(i, N_1+3) = 1 - \mu(N_1+1, N_1+3) - \mu(N_1+2, N_1+3), \\ &\qquad\qquad\qquad \dots, \\ &\qquad \sum_{i\in C_1} \mu(i, N_2) = 1 - \sum_{i=N_1+1}^{N_2-1} \mu(i, N_2). \end{aligned}$$

Observe that the summation of the sums on the right hand sides above are actually the numerator of the TAE weight (3). Then once again we will rewrite $W(C_1, C_2)$.

$$W(C_1, C_2) = \frac{N_2 - \sum_{i,j \in C_2, i \prec j} \mu(i,j)}{N_1 N_2}. \tag{10}$$

By multiplying equation (10) by $\frac{2}{N_2 - 1}$ we can make $W(C_2, C_2)$ appear and through some basic operations, we obtain the following expression for TAE weight (3):

$$W(C_2, C_2) = \frac{2 - 2N_1 W(C_1, C_2)}{N_2 - 1}. \tag{11}$$

Since we showed before that $W(C_1, C_2)$ is decreasing in $\alpha$ it follows from relation (11) that $W(C_2, C_2)$ is increasing in $\alpha$. $\square$

Moreover we can use equation (11) to find an expression that links all 3 TAE weights.

First we multiply equation (11) with $N_2 - 1$.

$$(N_2 - 1)W(C_2, C_2) = 2 - 2N_1 W(C_1, C_2).$$

Then we divide by $N_1$.

$$\frac{N_2 - 1}{N_1} W(C_2, C_2) = \frac{2}{N_1} - 2W(C_1, C_2).$$

Finally we use equation (8):

$$\frac{N_2 - 1}{N_1} W(C_2, C_2) = W(C_1, C_1) - 2W(C_1, C_2).$$

Or:

$$W(C_1, C_1) = \frac{N_2 - 1}{N_1} W(C_2, C_2) + 2W(C_1, C_2).$$

### 5.4 Proof of inequalities (5), (6) conditioned on $O^*$

In this section will directly prove inequalities (5), (6) conditioned on a smooth infection order. The proof will follow naturally from equations (8),(10).

**Theorem 7.** When conditioned on a smooth infection order $O^*$ that started from $C_1$, inequality (5) always holds. Assume there exist some $\alpha^*$ for which $W(C_2, C_2) = \frac{2}{2N_1 + N_2 - 1}$. Then for any $\alpha > \alpha^*$ inequality (6) is true.

*Proof.* For inequality (5) we will substitute $W(C_1, C_1), W(C_1, C_2)$ by relations (8) and (10). This results in the following:

$$\begin{aligned} W(C_1, C_1) > W(C_1, C_2) &\iff \\ \frac{2}{N_1} &> \frac{N_2 - \sum_{i,j \in C_2, i \prec j} \mu(i,j)}{N_1 N_2} \iff \\ 2 &> \frac{N_2 - \sum_{i,j \in C_2, i \prec j} \mu(i,j)}{N_2}. \end{aligned}$$

The quantity on the right hand side of the inequality is at most 1 so inequality (5) holds trivially when we have a smooth infection order.

For inequality (6) we will substitute $W(C_1, C_2)$ with relation (10) and work from there.

$$
\begin{aligned}
&W(C_2, C_2) > W(C_1, C_2) \iff \\
&W(C_2, C_2) > \frac{N_2 - \sum_{i,j \in C_2, i \prec j} \mu(i,j)}{N_1 N_2} \iff \\
&W(C_2, C_2) > \frac{2}{2N_1 + N_2 - 1}.
\end{aligned}
$$

Assume that $W(C_2, C_2)$ attains the value $2/(2N_1 + N_2 - 1)$ for some $\alpha^*$. We also showed that $W(C_2, C_2)$ is increasing in $\alpha$. Thus inequality (6) holds $\forall\ \alpha >\ \alpha^*$. $\square$

In this chapter we first studied the limiting behaviour of the $\mu(i, j)$ and the TAE weights. With the use of $\sigma$ we found a formula to calculate the asymptotic behaviour of the TAE weights for any given infection order $O$. Afterwards we studied the monotonicity of the $\mu(i, j)$. With the use of a heuristic we found out that unless $j$ is infected exactly after $i$ $\mu(i, j)$ are decreasing functions of $\alpha$. Using this result, we conditioned on a smooth infection order and studied the monotonicity of the TAE weights. Finally we used their monotonicity to prove inequalities (5), (6). Of course analogous results are derived if we assume that the smooth order originated in $C_2$.

# 6 Is the smooth infection order the most probable order?

Almost the entirety of the previous chapter was dedicated to the behaviour of the TAE weights in a smooth infection order. In this section we will try to justify the exclusive use of the smooth infection order by showing that whenever $\lambda_{in}$ is larger than $\lambda_{out}$ then the smooth infection order becomes the most probable order.

We will do this by using the fact that a SI process is a Markov chain. This enables us to use a dynamic programming approach. We begin by defining some elements of the markov chain. Afterwards we present a brief explanation on how the dynamic programming approach works. Then we proceed by giving a concrete example on $K_6$. Finally we will attempt to generalise for any complete graph $K_N$. For this chapter we explicitly require that $N_1 = N_2$.

## 6.1 SI process

### 6.1.1 State and stage

**Definition 6.1** (Markov chain)**.** A stochastic process is called markov when it has the memoryless property.

**Remark 8.** The memoryless property refers to future states of the system only depending on the current state and not the whole history.

**Theorem 9.** Let $K_N$ be a complete graph with $N$ vertices. We assume the vertices are divided into two communities $C_1,\ C_2$, each having $N_1,\ N_2$ vertices respectively. The SI process on $K_N$ with state space $\mathbb{S} = \{(I_1, I_2)\},\ I_1 = \{0, 1, 2, \dots, N_1\}, I_2 = \{0, 1, 2, \dots, N_2\}$ is a Markov chain. $I_1,\ I_2$ are the number of infected vertices in each community.

*Proof.* Since the rate of a new infection only depends on the current number of infected and susceptible vertices, (1), the SI process satisfies the Markov property. $\square$

**Definition 6.2** (Stage of the process)**.** We define the stage of the SI process as $\mathcal{S} = I_1 + I_2$. A stage contains all the states $(I_1, I_2)$ for which $I_1 + I_2 = \mathcal{S}$ holds.

### 6.1.2 Transition probabilities

The process starts on either state $(1, 0)$ or $(0, 1)$ with equal probability. Afterwards it advances to a new state with some probability until it reaches state $(N_1, N_2)$. The transition probability from state $s$, to state $s^{'}$ is denoted as $\mathbf{P}(s^{'}|s)$ and is equal to:

$$\mathbf{P}(s^{'}|s) = \begin{cases} \frac{S_1 I_1 \lambda_{in} + S_1 I_2 \lambda_{out}}{[(S_1 I_1 + S_2 I_2)\lambda_{in} + (S_1 I_2 + S_2 I_1)\lambda_{out}]} & \text{if } s = (I_1, I_2), s^{'} = (I_1 + 1, I_2), \\ \frac{S_2 I_2 \lambda_{in} + S_2 I_1 \lambda_{out}}{[(S_1 I_1 + S_2 I_2)\lambda_{in} + (S_1 I_2 + S_2 I_1)\lambda_{out}]} & \text{if } s = (I_1, I_2), s^{'} = (I_1, I_2 + 1), \\ 1 & \text{if } s = (N_1, I_2), s^{'} = (N_1, I_2 + 1), \\ 1 & \text{if } s = (I_1, N_2), s^{'} = (I_1 + 1, N_2), \\ 0 & \text{else.} \end{cases} \quad (12)$$

## 6.2 Dynamic programming for establishing the most probable order

First we define the cost of a state $s$, $Cost(s)$. and the cost of an infection order $O$.

**Definition 6.3** (Cost of a state)**.** The cost of a state $s$ is defined recursively as the maximum of the costs of the states that can be reached from $s$,$(s_1, s_2)$, multiplied by the respective transition probabilities.

$$Cost(s) = max\{Cost(s_1)\mathbf{P}(s_1|s), Cost(s_2)\mathbf{P}(s_2|s)\}.$$

In the special case where $s = (N_1, I_2)$ or $s = (I_1, N_2)$, $Cost(s) = 1$.

**Definition 6.4** (Cost of an infection order)**.** The cost of an infection order $O$ is the product of the costs of the respective states $s_1, s_2, \ldots, s_N$ that are part of the order.

**Remark 10.** Every infection order $O$ contains **exactly** $N$ states.

Since SI process is a Markov chain we can use dynamic programming to establish the most probable order. Dynamic programming is an algorithmic optimisation technique that breaks a maximisation (minimisation) problem into smaller sub-problems in a recursive manner [18]. Our problem is a maximisation problem. We seek the infection order ( sequence of states) that has the largest $Cost$, under the condition $\lambda_{in} = \kappa\lambda_{out},\ \kappa > 1$.

The algorithm starts at the final stage $\mathcal{S} = N$ at state $s = (N_1, N_2)$. We set $Cost(N_1, N_2) = 1$. At each step we move backwards, to the previous stage $\mathcal{S} = N - 1$ and for each state $s \in \mathcal{S}$ we compute its cost. The algorithm stops at stage $\mathcal{S} = 1$ after computing $Cost(1, 0)$ and $Cost(0, 1)$. After we have computed the cost of each state we can calculate the cost of each infection order.

## 6.3 Most probable order on $K_6$ using dynamic programming

In this section we will follow the dynamic programming approach on $K_6$. We assume $\lambda_{in} = \kappa\lambda_{out},\ \kappa > 1$. First we give the transition probability matrix.

*Transition probability matrix for $K_6$*

$$\begin{array}{ccccccccccccccc|c}
(0,1) & (1,0) & (0,2) & (1,1) & (2,0) & (0,3) & (1,2) & (2,1) & (3,0) & (1,3) & (2,2) & (3,1) & (2,3) & (3,2) & (3,3) & \\
0 & 0 & \frac{2\kappa}{2\kappa+3} & \frac{3}{2\kappa+3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (0,1) \\
0 & 0 & 0 & \frac{3}{2\kappa+3} & \frac{2\kappa}{2\kappa+3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (1,0) \\
0 & 0 & 0 & 0 & 0 & \frac{\kappa}{\kappa+3} & \frac{3}{\kappa+3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (0,2) \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & (1,1) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3}{\kappa+3} & \frac{\kappa}{\kappa+3} & 0 & 0 & 0 & 0 & 0 & 0 & (2,0) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & (0,3) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\kappa+1}{4\kappa+5} & \frac{2\kappa+4}{4\kappa+5} & 0 & 0 & 0 & 0 & (1,2) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2\kappa+4}{4\kappa+5} & \frac{2\kappa+1}{4\kappa+5} & 0 & 0 & 0 & (2,1) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & (3,0) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & (1,3) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & (2,2) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & (3,1) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & (2,3) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & (3,2) \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & (3,3)
\end{array}$$

It is worth noting the existence of a symmetry with respect to the communities. E.g the probability $P((0,3)|(0,2)) = P((3,0)|(2,0))$. We could only study half of the process by exploiting this symmetry but since this is a numerical example we will study it completely.

We will now perform a backwards pass in a dynamic programming manner.

STAGE 6:
We start at stage $\mathcal{S} = 6$. The only state is $s = (3,3)$ and as we mentioned earlier $Cost(3,3) = 1$.

STAGE 5:
There are two states in stage $\mathcal{S} = 5$. These are $(3,2)$ and $(2,3)$. From both we transition to state $(3,3)$ with probability 1 and $Cost(3,3) = 1$. Thus $Cost(3,2) = Cost(2,3) = 1$.

STAGE 4:
There are three states in stage $\mathcal{S} = 4$. For the first two,$(3,1),(1,3)$ there is only one transition for each, which is to a state of unit cost, hence $Cost(3,1) = Cost(1,3) = 1$. For the third one $(2,2)$, there are two equally probable transitions towards unit cost states. Thus $Cost(2,2) = \frac{1}{2}$.

STAGE 3:
There are four states in stage $\mathcal{S} = 3$. For the first two,$(3,0),(0,3)$ there is only one transition for each, which is to a state of unit cost, hence $Cost(3,0) = Cost(0,3) = 1$. The costs of the remaining two states, $(1,2)$ and $(2,1)$, will be equal. For reference we will calculate $Cost(1,2)$.

$$Cost(1,2) = max\left\{\frac{2\kappa+1}{4\kappa+5}Cost(1,3), \frac{2\kappa+4}{4\kappa+5}Cost(2,2)\right\}$$

Substitute $Cost(1,3) = 1,\ Cost(2,2) = \frac{1}{2}$.

$$Cost(1,2) = max\left\{\frac{2\kappa+1}{4\kappa+5}, \frac{1}{2}\frac{2\kappa+4}{4\kappa+5}\right\}$$

Recall we have assumed that $\kappa > 1$. Then:

$$\begin{aligned}
&\frac{2\kappa+1}{4\kappa+5} > \frac{1}{2}\frac{2\kappa+4}{4\kappa+5} \iff \\
&2\kappa+1 > \kappa+2 \iff \\
&\kappa > 1.
\end{aligned}$$

Thus $Cost(1,2) = Cost(2,1) = \frac{2\kappa+1}{4\kappa+5}$.

Stage 2:
There are three states in stage $\mathcal{S} = 2$. We start with state $(1,1)$. From $(1,1)$ there are two equally probable transitions to $(2,1)$ or $(1,2)$. Recall that at the previous stage we showed that $Cost(2,1) = Cost(1,2)$. As a result $Cost(1,1) = \frac{1}{2}Cost(1,2)$. We know that $Cost(0,2) = Cost(2,0)$. We calculate the former for reference.

$$Cost(0,2) = max\left\{\frac{\kappa}{\kappa+3}Cost(0,3), \frac{3}{\kappa+3}Cost(1,2)\right\}$$

Substitute $Cost(0,3) = 1$ and $Cost(1,2) = \frac{2\kappa+1}{4\kappa+5}$. Then:

$$Cost(0,2) = max\left\{\frac{\kappa}{\kappa+3}, \frac{3}{\kappa+3}\frac{2\kappa+1}{4\kappa+5}\right\}$$

Recall we have assumed that $\kappa > 1$. Then:

$$\begin{aligned}
&\frac{\kappa}{\kappa+3} > \frac{2\kappa+1}{4\kappa+5}\frac{3}{\kappa+3} \iff \\
&\kappa > 1 \textbf{ or } \kappa < -\frac{6}{8}.
\end{aligned}$$

So $Cost(0,2) = Cost(2,0) = \frac{\kappa}{\kappa+3}$.

Stage 1: In the last stage there are only two stages: $(1,0)$ and $(0,1)$. We compute $Cost(0,1)$ for reference.

$$\begin{aligned}
Cost(0,1) &= \max\left\{\frac{2\kappa}{2\kappa+3}Cost(0,2), \frac{3}{2\kappa+3}Cost(1,1)\right\} \\
&= max\left\{\frac{2\kappa^2}{(2\kappa+3)(\kappa+3)}, \frac{6\kappa+3}{2(2\kappa+3)(4\kappa+5)}\right\}
\end{aligned}$$

From here it sufficient to observe:

$$\begin{aligned}&\kappa > 1 \iff \\ &\frac{2\kappa^2}{(2\kappa+3)(\kappa+3)} > \frac{6\kappa+3}{2(2\kappa+3)(4\kappa+5)}.\end{aligned}$$

Thus $Cost(0,1) = \frac{2\kappa}{2\kappa+3}Cost(0,2)$.

Bellow we present a table showing the cost of each state when $\kappa > 1$. It is clear that the most probable paths are the ones that follow the smooth infection orders.

| Stage | State | Cost |
|---|---|---|
| 6 | (3,3) | 1 |
| 5 | (2,3) | 1 |
| | (3,2) | 1 |
| 4 | (1,3) | 1 Cost(2,3) |
| | (3,1) | 1 Cost(3,2) |
| | (2,2) | 1/2 |
| 3 | (0,3) | 1 Cost(1,3) |
| | (1,2) | $\frac{2\kappa+1}{4\kappa+5}$Cost(1,3) |
| | (2,1) | $\frac{2\kappa+1}{4\kappa+5}$Cost(3,1) |
| | (3,0) | 1 Cost(3,1) |
| 2 | (0,2) | $\frac{\kappa}{\kappa+3}$Cost(0,3) |
| | (1,1) | $\frac{1}{2}$Cost(1,2) |
| | (2,0) | $\frac{\kappa}{\kappa+3}$Cost(3,0) |
| 1 | (1,0) | $\frac{2\kappa}{2\kappa+3}Cost(0,2)$ |
| | (0,1) | $\frac{2\kappa}{2\kappa+3}Cost(2,0)$ |

We have shown that in $K_6$ whenever $\lambda_{in} > \lambda_{out}$ then the smooth infection order is the most probable. However $K_6$ is only but an instance. Our aim is to generalise for any complete graph $K_N$ with two equally sized communities.

## 6.4 Most probable order on $K_N$

In this section we will attempt to generalise the results we obtained for $K_6$. That is, show that if $\lambda_{in} = \kappa\lambda_{out}$ and $\kappa > 1$ then the smooth order is the most probable order.

However, having a general graph $K_N$ makes the use of dynamic programming for every state impossible. The reason for this is that the state space can be arbitrarily large. However we can walk around this roadblock by exploiting the structure of the process. Firstly, the process is symmetric with respect to the communities. That is, the probabilities $P\big((J_1+1, J_2)|(J_1, J_2)\big)$ and $P\big((J_2, J_1+1)|(J_2, J_1)\big)$ are both equal to:

$$\frac{J_1(N/2-J_1)\lambda_{in} + J_2(N/2-J_1)\lambda_{out}}{[[(J_1(N/2-J_1)+J_2(N/2-J_2)]\lambda_{in} + (J_1(N/2-J_2)+J_2(N/2-J_1)\lambda_{out}]}.$$

That means we can ignore half the states in each stage. Moreover we can distinguish two types of stages. Those that $\mathcal{S} < N/2$ and those that $\mathcal{S} > N/2$. We will use the fact that for the stages that $\mathcal{S} > N/2$ the cost of the states that are part of the smooth order is equal to 1. I.e states of the form $(N_1, \beta)$, or $(\beta, N_2)$ $\beta \in \mathbb{N}$.

We start by proving the following theorem.

**Theorem 11.** Assume we have an SI process on a complete graph $K_N$ with two equally sized communities $C_1, C_2$ with $|C_1| = N_1, |C_2| = N_2$. For the infection rates $\lambda_{in}$ and $\lambda_{out}$ holds that $\lambda_{in} = \kappa\lambda_{out},\ \kappa > 1$. Then for any state of the SI process of the form $(N_1 - 1, N_2 - \beta)$ the optimal cost is obtained by transitioning to a state that is part of the smooth infection order, i.e to state $(N_1, N_2 - \beta)$.

*Proof.* We will show this by using induction. First we work with state $(N_1 - 1, N_2 - 1)$. The possible transitions from $(N_1 - 1, N_2 - 1)$ are either to state $(N_1, N_2 - 1)$ or to state $(N_1 - 1, N_2)$. It easy to verify that both transitions have probability 1/2. Moreover $Cost(N_1, N_2 - 1) = Cost(N_1 - 1, N_2) = 1$. Thus $Cost(N_1 - 1, N_2 - 1) = 1/2$. Now for the induction base we use state $(N_1 - 1, N_2 - 2)$. We set,

$$P_1 = \mathbf{P}(N_1, N_2 - 2 | N_1 - 1, N_2 - 2).$$

Then,

$$\begin{aligned} Cost(N_1 - 1, N_2 - 2) &= max\left\{P_1 Cost(N_1, N_2 - 2), (1 - P_1)Cost(N_1 - 1, N_2 - 1)\right\} \\ &= max\left\{P_1, (1 - P_1)/2\right\}. \end{aligned}$$

Since we want to make the transition to the smooth path most probable we require:

$$\begin{aligned} &P_1 > \frac{1 - P_1}{2} \iff \\ &P_1 > 1/3. \end{aligned}$$

By substituting $P_1$, (12), it is easy to verify that the inequality holds iff $\kappa > 1$.

$$\begin{aligned} &\frac{\kappa(N_1 - 1) + N_2 - 2}{\kappa(N_1 - 1 + 2N_2 - 4) + (2N_1 - 2 + N_2 - 2)} > \frac{1}{3} \iff \\ &\kappa > 1. \end{aligned}$$

Now for the induction step we assume that our statement holds for state $(N_1, N_2 - \beta)$. That means that

$$\begin{aligned} Cost(N_1 - 1, N_2 - \beta) &= max\left\{P_2 Cost(N_1, N_2 - \beta), (1 - P_2)Cost(N_1 - 1, N_2 - \beta + 1)\right\} \\ &= P_2 Cost(N_1, N_2 - \beta) = P_2, \end{aligned}$$

where,

$$P_2 = \mathbf{P}(N_1, N_2 - \beta | N_1 - 1, N_2 - \beta) = \frac{\kappa(N_1 - 1) + N_2 - \beta}{\kappa[N_1 - 1 + (N_2 - \beta)\beta] + (N_1 - 1)\beta + (N_2 - \beta)}.$$

Now we have to show that it also holds for state $(N_1 - 1, N_2 - \beta - 1)$.
Set $\mathbf{P}(N_1, N_2 - \beta - 1 | N_1 - 1, N_2 - \beta - 1) = P_3$. We have

$$\begin{aligned} Cost(N_1 - 1, N_2 - \beta - 1) &= max\left\{P_3 Cost(N_1, N_2 - \beta - 1), (1 - P_3)Cost(N_1 - 1, N_2 - \beta)\right\} \\ &= max\left\{P_3, P_2(1 - P_3)\right\}. \end{aligned}$$

Where,

$$P_3 = \frac{\kappa(N_1-1)+N_2-\beta-1}{\kappa[N_1-1+(N_2-\beta-1)(\beta+1)]+[(N_1-1)(\beta+1)+N_2-k-1]}.$$

Now all that is left to do is show $\kappa > 1 \iff P_3 > P_2(1-P_3)$.

$$P_3 > P_2(1-P_3) \iff$$

$$\kappa^2(N_1-1)[N_1-N_2+2\beta]+\kappa[(N_1-N_2)(1-2\beta)-\beta(\beta-1)-(N_1-N_2)^2]+(N_2-\beta)^2+\beta-N_1N_2 > 0.$$

Instead of explicitly finding the roots, we will set $\kappa = 1$ and check whether it is indeed a root. Doing so indeed causes everything to cancel out. Therefore $\kappa = 1$ is a root of the polynomial in the left side of the inequality. Thus when $\kappa > 1$ implies $P_3 > P_2(1-P_3)$. And $Cost(N_1-1, N_2-\beta-1) = P_3$ and the optimal transition is towards the smooth order. Thus the proof is completed. □

We continue with proving that the smooth is the most probable order when $\kappa > 1$. Using Theorem (11) by setting $\beta = N_2$ we conclude that from state $(N_1-1, 0)$ the optimal transition that maximises the cost is the one towards the smooth order. That is towards state $(N_1, 0)$. From here we could do another induction, this time on $(N_1-\beta, 0)$. Unfortunately we cannot do that as we have no means of knowing the cost of states $(N_1-\beta, 1)$.

Instead we will use another approach. Recall that in our process, once a community is infected completely,then all subsequent transitions happen with probability 1. Also every infection order has exactly $N_1 + N_2$ transitions. Then it follows logically that the sooner one community is completely infected, the bigger the $Cost(O)$ , since it will have more terms equal to 1. Lastly, for some orders, we can calculate their total cost, as it is a product of the probabilities of the transitions.

**Theorem 12.** Assume we have an SI process on a complete graph $K_N$ with two equally sized communities $C_1, C_2$ with $|C_1| = N_1, |C_2| = N_2$. For the infection rates $\lambda_{in}$ and $\lambda_{out}$ holds that $\lambda_{in} = \kappa\lambda_{out},\ \kappa > 1$. Then a smooth infection order is asymptotically the most probable. That is, for any non-smooth infection order $O'$:

$$\lim_{\kappa\to\infty} \frac{Cost(O')}{Cost(O_1^*)} = 0.$$

*Proof.* We will make use of the dynamic programming approach once more.

The cost of a state is the total probability of reaching $(N_1, N_2)$ from it. Each transition probability is a function of $\kappa$. So there is a one to one correspondence between infecting a community as soon as possible and maximizing cost, since $Cost(State) \leq 1$. The cost of the smooth order $O_1^*$ is given by:

$$Cost(O_1^*) = \prod_{n=1}^{N_1-1} \frac{n\kappa}{n\kappa + N_2}$$

In a smooth infection order a whole community is infected by stage $N/2 = N_1$. Assume that exist an infection order $O^{'}$ for which $Cost(O^{'}) > Cost(O_1^*)$. We will consider two cases.

First case: $O^{'}$ infects a whole community by stage $N_1$ as well. Then $O^{'}$ is a smooth order as well.

Second case: $O^{'}$ infects a whole community by stage $N_1 + \beta,\ \beta \in \mathbb{N}$. That means that in that order at least one inter community transition occurred. The first of which contributed to the total cost of $O^{'}$ a term of the form $\frac{\rho_2}{\rho_1\kappa+\rho_2}$, $\rho_1, \rho_2 \in \mathbb{N}$. In the simplest case where only such transition happened and it happened in stage 2 the cost of $O^{'}$ is given by:

$$Cost(O^{'}) = \frac{N_2}{N_2 + (N_1-1)\kappa} \prod_{n=1}^{N_1-1} \frac{(N_2-n)n\kappa + n}{[(N_2-n)n + N_1 - 1]\kappa + n + (N_1-1)(N_2-n)}.$$

Here we have assumed that $O^{'}$ started at a vertex of community 1 and then infected the entire community 2. It is clear that $Cost(O^{'})$ is of order $\mathcal{O}(\kappa^{N_1-2})$, where $Cost(O_1^*)$ is of order $\mathcal{O}(\kappa^{N_1-1})$.

Thus:

$$\lim_{\kappa\to\infty} \frac{Cost(O^{'})}{Cost(O_1^*)} = 0.$$

Thus either there exist no order $O^{'}$ such that $Cost(O^{'}) \geq Cost(O^*)$ asymptotically, or if it exists, it is another smooth order, in this case $Cost(O^{'}) = Cost(O^*)$. $\square$

In this chapter we tried to justify the exclusive use of the smooth infection order. Intuitively the smooth order should be the most probable given sufficient infection rates and indeed, our findings support that intuition. More specifically the last theorem of the chapter states precisely that.

# 7 Averaging over many cascades

All the work we did in the previous chapters had a major simplification. The set of cascades $\mathcal{C}$ contained only a single cascade $c$ in it. Here we drop this simplification and we examine the two inequalities (5),(6) when we sum the TAE weights over many cascades. However, in light of the work done in the previous chapter it makes sense to focus on $\mathcal{C}$ containing cascades that follow a smooth infection order. We again require $N_1 = N_2$.

## 7.1 Repeating the same smooth cascade

Assume that every cascade $c \in \mathcal{C}$ follows a smooth infection order that started at $C_1$. In this case everything we proved in Chapter 5 automatically applies here as well and we have nothing new to prove.

## 7.2 Different smooth cascades

Here we will examine the case where $\mathcal{C}$ contains cascades that follow a smooth infection order but this time the cascades can originate from any of the two communities. For clarity we denote the smooth infection order that started at $C_1$ by $O_1^*$ and the one that started at $C_2$ by $O_2^*$. First consider a simple case: $\mathcal{C}$ contains two cascades, one that follows $O_1^*$ and one that follows $O_2^*$. Then inequalities (5),(6) hold.

$$\sum_{c\in\mathcal{C}} W(C_1, C_1) > \sum_{c\in\mathcal{C}} W(C_1, C_2)$$
$$W((C_1, C_1)|O_1^*) + W((C_1, C_1)|O_2^*) > W((C_1, C_2)|O_1^*) + W((C_1, C_2)|O_2^*)$$

To show that this inequality holds we will use two simple facts: The values of the TAE weights $W(C_1, C_1)$, $W(C_2, C_2)$ swap values in the two smooth cascades but $W(C_1, C_2)$ remains the same.

To elaborate, in $O_1^*$ vertices $\left\{1, 2, \ldots, \frac{N}{2}\right\} \in C_1$ but in $O_2^*$ these vertices $\left\{1, 2, \ldots, \frac{N}{2}\right\} \in C_2$. So the weights themselves remain numerically the same but they belong to different communities. Therefore, in $O_1^*$ the numerator of (2) is equal to:

$$\sum_{i,j\in C_1, i\prec j} \mu(i, j|O_1^*) = \mu(1, 2) + \sum_{i=1}^{i=2} \mu(i, 3) + \cdots + \sum_{i=1}^{\frac{N}{2}-1} (i, \frac{N}{2}).$$

But in $O_2^*$ the numerator of (3) is equal to:

$$\sum_{i,j\in C_2, i\prec j} \mu(i, j|O_2^*) = \mu(1, 2) + \sum_{i=1}^{i=2} \mu(i, 3) + \cdots + \sum_{i=1}^{\frac{N}{2}-1} (i, \frac{N}{2}).$$

This symmetry is caused by the equally sized communities.

Thus we can re write the inequality as:

$$W(C_1, C_1|O_1^*) + W(C_2, C_2|O_1^*) > 2W(C_1, C_2|O_1^*).$$

This can be broken down into two inequalities:

$$W(C_1, C_1|O_1^*) > W(C_1, C_2|O_1^*),$$

which we showed that in always holds in Chapter 5, and

$$W(C_2, C_2|O_1^*) > W(C_1, C_2|O_1^*),$$

where in Chapter 5 we showed that it holds for any $\alpha > \alpha^*$.

A generalisation follows naturally.

**Theorem 13.** Assume $\mathcal{C}$ has $\xi_1$ cascades that follow $O_1^*$ and $\xi_2$ cascades that follow $O_2^*$. Then inequalities (5),(6) hold.

*Proof.* We have the following inequality:

$$\begin{aligned}
&\sum_{c\in\mathcal{C}} W(C_1, C_1) > \sum_{c\in\mathcal{C}} W(C_1, C_2) \Rightarrow \\
&\xi_1 W((C_1, C_1)|O_1^*) + \xi_2 W((C_1, C_1)|O_2^*) > (\xi_1 + \xi_2) W((C_1, C_2)|O_1^*) \Rightarrow \\
&\xi_1 W((C_1, C_1)|O_1^*) + \xi_2 W((C_2, C_2)|O_1^*) > (\xi_1 + \xi_2) W((C_1, C_2)|O_1^*).
\end{aligned}$$

This, again, holds term-wise for any any $\alpha > \alpha^*$. □

In this chapter we showed that inequalities (5),(6) hold when we sum over cascades that follow a smooth infection order. And this is independent of the community where the cascade originates from.

# 8 Small Perturbations on the Smooth Order

Throughout the project we have thoroughly studied the behaviour of the TAE weigths under a smooth infection order. In this final chapter we will investigate how a small perturbation of the smooth infection order affects the behaviour of the TAE weights. The perturbation we choose is a very simple one. Assuming we have a smooth order $O_1^*$ we move the last vertex of community 1 to the end of the order. Formally, we move the $N_1$ vertex to the $N$ position of our order. We will be referencing this new order as $O_{1,p}^*$. Thus $Q_{1,p}^*(N_1) = 2$ and $Q_{1,p}^*(N) = 1$ . Recall that we had a numerical example of this order on $K_{50}$ back in Chapter 4, in Figure 6. So we have a strong indication of what we can expect, namely $W(C_1, C_1)$ to become a decreasing function of $\alpha$ with this perturbation. Recall that $W(C_1, C_1|O_1^*)$ was a constant equal to $\frac{2}{N_1}$. But before we examine the monotonicity we will first have a look at the limiting behaviour of the TAE weights conditioned on $O_{1,p}^*$.

## 8.1 Limiting Behavior in $O_{1,p}^*$

Recall from Chapter 5 the formula we found for calculating the limiting behaviour of the TAE weights conditioned on any infection order by using equations (7). For $O_{1,p}^*$ specifically we have that $\sigma(C_2, C_2|O_{1,p}^*) = N_2 - 1$ since the order in which the vertices of $C_2$ were infected remains as it was in $O_1^*$. Moreover, $\sigma(C_1, C_1|O_{1,p}^*) = N_1 - 2$, $\sigma(C_1, C_2|O_{1,p}^*) = 2$. Thus the limits of the TAE weights when $\alpha \to \infty$ conditioned on $O_{1,p}^*$ are:

$$\begin{aligned}
\lim_{\alpha\to\infty} W(C_1, C_1|O_{1,p}^*) &= \frac{N_1 - 2}{N_1(N_1 - 1)/2}, \\
\lim_{\alpha\to\infty} W(C_2, C_2|O_{1,p}^*) &= \frac{2}{N_2}, \\
\lim_{\alpha\to\infty} W(C_1, C_2|O_{1,p}^*) &= \frac{2}{N_1 N_2}.
\end{aligned}$$

## 8.2 Monotonicity in $O_{1,p}^*$

**Theorem 14.** Conditioned on the infection order $O_{1,p}^*$, $W(C_1, C_1)$ is a decreasing function of $\alpha$ and $W(C_2, C_2)$ is an increasing function of $\alpha$.

*Proof.* We begin by investigating the monotinicity of $W(C_1, C_1|O_{1,p}^*)$. We follow the same approach as we did in Chapter 5, which is breaking up the big sum in the numerator into smaller sums and use Lemma 2.

$$\begin{aligned}\sum_{i,j\in C_1, i\prec j}\mu(i,j) &= \mu(1,2)+\sum_{i=1}^{2}\mu(i,3)+...+\sum_{i=1}^{N_1-2}\mu(i,N_1-1)+\sum_{i=1}^{N_1-1}\mu(i,N)\\ &= N_1-2+\sum_{i=1}^{N_1-1}\mu(i,N).\end{aligned}$$

Observe that $\sum_{i=1}^{N_1-1}\mu(i,N)$ is a sum of $\mu(i,j)$ that are defined only on non consecutive vertices. In Chapter 5 we used a heuristic to show that $\mu(i,j)$ that are defined on non consecutive vertices are decreasing functions of $\alpha$. Therefore $W(C_1,C_1|O^*_{1,p})$ is a decreasing function of $\alpha$. This is exactly what we observed in Figure 6.

Since we did not move any vertex of $C_2$, the monotonicity of $W(C_2,C_2|O^*_{1,p})$ is the same as in $W(C_2,C_2|O^*_1)$, increasing in $\alpha$. □

Finally, we examine $W(C_1,C_2|O^*_{1,p})$ by using the same method as above. We do this separately from the other two weights as we will not give a formal proof but rather a heuristic.

$$\begin{aligned}\sum_{i\in C_1, j\in C_2, i\prec j}\mu(i,j)+\sum_{i\in C_2, j\in C_1, i\prec j}\mu(i,j) &= \sum_{i\in C_1}\mu(i,N_1+1)+\sum_{i\in C_1}\mu(i,N_1+2)\\ &\quad+\cdots+\sum_{i\in C_1}\mu(i,N-1)+\sum_{i\in C_2}\mu(i,N)\\ &= N_2-\sum_{i,j\in C_2, i\prec j}\mu(i,j)+\sum_{i\in C_2}\mu(i,N),\end{aligned}$$

where:

$$\sum_{i\in C_2}\mu(i,N)=\sum_{i\in C_2/\{N-1\}}\mu(i,N)+\mu(N-1,N).$$

So $\sum_{i\in C_2}\mu(i,N)$ is composed of $N_2-1$ decreasing functions and a single increasing function $\mu(N-1,N)$. In particular $\mu(N-1,N)$ is the smallest valued weight in $O^*_{1,p}$, as it is the weight of the last infection. Then we can conclude that $\sum_{i\in C_2}\mu(i,N)$ is a decreasing function of $\alpha$ as well. Thus $W(C_1,C_2|O^*_{1,p})$ is a decreasing function of $\alpha$.

In this section we studied if and how a small perturbation in a smooth infection order affects the behaviour of the TAE weight (2)-(4). What we found is a swift in the monotonicity of $W(C_1,C_1)$ which was constant in $O^*_1$ but became decreasing in $O^*_{1,p}$. As for the other two TAE weights, their monotonicity remains intact.

# 9 Conclusion, further research and limitations

The main goal of this project was to provide a mathematical explanation onto why algorithms for community detection though cascade data that utilise weight assignment work. We found out that a major catalyst is the way that the cascade propagates through the network. We showed that for sufficiently bigger intra-community rate the smooth infection order is asymptotically the most probable. On top of that we established that in the smooth order the intra-community weights are much bigger than the inter-community weight. That is, we assume that given a smooth infection order, an algorithm like CLIQUE [20] is guaranteed to work. Proving this could be a natural continuation of this project. Apart from this, we feel that there are many more aspects that need to be considered and examined.

First of all, do our findings hold when the communities are not of equal size? Recall that for some results in was necessary to impose this restriction on the size of the communities. More specifically on Chapter 6 it we had to impose this condition. In that instance we did it in order to preserve the symmetry that the process had. This symmetry allowed the process to evolve in a identical way regardless of the community it originated in. This reduced the study of the whole process (N states), to the study of half the process ($N/2$ states). Furthermore we used the equal sized communities for the argument that smooth order infects a community at stage $N/2$. If we drop the assumption of equal communities this symmetry disappears. Will the smooth infection order still be the most probable? Another case where we required the two communities to be of equal size is Chapter 7. This made $W(C_1, C_2|O_1^*) = W(C_1, C_2|O_2^*)$ and enabled for a swift proof. Should we drop the assumption $N_1 = N_2$ the weights will become unequal as well. Do our findings still hold?

In this project we studied two infection orders: $O^*$ and $O_{1,p}^*$. Can we define a "distance" between these infection orders? We also know that the two inequalities (5), (6) hold for both of these orders. That hints that these infection orders are not that "far" apart. How big can this "distance" between any order $O$ and the smooth order become, before inequalities (5), (6) do not hold for $O$?

It is our duty to pinpoint limitations that our research has. First, most benchmark networks for community detection have many communities. However the networks that we worked with in this project have only two. As we mentioned this was done in an effort to simplify the problem and in turn make the study easier. Nevertheless this is a big drawback. We have no means of verifying whether our findings extend, or even hold, when the network have many communities. How our research extends if we allow the communities to overlap? Moreover these benchmark networks usually have much bigger communities. We were only able to perform numerical calculations up until 50 nodes in every community. Just by making the code used for the numerical calculations a bit more efficient, we would be able to simulate realistic sized networks. Furthermore all the networks we used were assumed to be complete. This was done in order to make the existence of each edge deterministic. A natural extension is to move to random graphs, however in such a case we have to make sure that the probability for intra-community edges is sufficiently high (or even 1).

Despite of the drawbacks and the limitations we believe this work was a solid first step into understanding how and why algorithms such as CLIQUE would work. Relaxing some of the assumptions we made in this project would be a natural way into generalising the results.

## References

[1] Linda J.S. Allen. Some discrete-time si, sir, and sis epidemic models. *Mathematical Biosciences*, 124(1):83–105, 1994.

[2] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Cascade-based community detection. *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*, 2013.

[3] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Influence-based network-oblivious community detection. pages 955–960. IEEE, 12 2013.

[4] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Efficient methods for influence-based network-oblivious community detection. *ACM Transactions on Intelligent Systems and Technology*, 8:1–31, 1 2017.

[5] Tanmoy Chakraborty, Ayushi Dalmia, Animesh Mukherjee, and Niloy Ganguly. Metrics for community analysis. *ACM Computing Surveys*, 50:1–37, 7 2018.

[6] Yen-Liang Chen, Ching-Hao Chuang, and Yu-Ting Chiu. Community detection based on social interactions in a social network. *Journal of the Association for Information Science and Technology*, 65:539–550, 3 2014.

[7] Michele Coscia, Fosca Giannotti, and Dino Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4:512–546, 6 2012.

[8] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84:066106, 12 2011.

[9] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 6 2009.

[10] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, 7 2016.

[11] Caitlin Gray, Lewis Mitchell, and Matthew Roughan. Bayesian inference of network structure from information cascades. *IEEE Transactions on Signal and Information Processing over Networks*, 6:371–381, 2020.

[12] Ling He, Wenzhong Guo, Yuzhong Chen, Kun Guo, and Qifeng Zhuang. Discovering overlapping communities in dynamic networks based on cascade information diffusion. *IEEE Transactions on Computational Social Systems*, pages 1–13, 2021.

[13] Wei Huang and Chunguang Li. Epidemic spreading in scale-free networks with community structure. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(01):P01014–P01014, jan 2007.

[14] Brian Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83:016107, 1 2011.

[15] Matt J Keeling and Ken T.D Eames. Networks and epidemic models. *Journal of The Royal Society Interface*, 2:295–307, 9 2005.

[16] István Z. Kiss, Joel C. Miller, and Péter L. Simon. *Mathematics of Epidemics on Networks*, volume 46. Springer International Publishing, 2017.

[17] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78:046110, 10 2008.

[18] Art Lew and Holger Mauch. *Dynamic Programming : A Computational Tool*. Springer International Publishing, 2007.

[19] Liudmila Prokhorenkova, Alexey Tikhonov, and Nelly Litvak. Learning clusters through information diffusion. pages 3151–3157. ACM, 5 2019.

[20] Liudmila Prokhorenkova, Alexey Tikhonov, and Nelly Litvak. When less is more: Systematic analysis of cascade-based community detection. *ACM Transactions on Knowledge Discovery from Data*, 16:1–22, 1 2020.

[21] Liqing Qiu, Zhongqi Yang, Shiwei Zhu, Xiangbo Tian, and Shuqi Liu. Comim: A community-based algorithm for influence maximization under the weighted cascade model on social networks. *Intelligent Data Analysis*, 26:205–220, 1 2022.

[22] Maryam Ramezani, Ali Khodadadi, and Hamid R. Rabiee. Community detection using diffusion information. *ACM Transactions on Knowledge Discovery from Data*, 12:1–22, 4 2018.

[23] MANUEL GOMEZ RODRIGUEZ, JURE LESKOVEC, DAVID BALDUZZI, and BERNHARD SCHÖLKOPF. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2:26–65, 4 2014.

[24] Jaron Sanders, Alexandre Proutière, and Se-Young Yun. Clustering in block markov chains. *The Annals of Statistics*, 48, 12 2020.

[25] Mohammad Sattari and Kamran Zamanifar. A cascade information diffusion based label propagation algorithm for community detection in dynamic social networks. *Journal of Computational Science*, 25:122–133, 3 2018.

[26] Daiki Suzuki and Sho Tsugawa. Effects of hidden users on cascade-based community detection, 2021.