Bayesian Contact Tracing for Communicable Respiratory Diseases

by

Ayman M. Shalaby

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Purpose: The purpose of our work is to develop a system for automatic contact tracing with the goal of identifying individuals who are most likely infected, even if we do not have direct diagnostic information on their health status. Control of the spread of respiratory pathogens (e.g. novel influenza viruses) in the population using vaccination is a challenging problem that requires quick identification of the infectious agent followed by large-scale production and administration of a vaccine. This takes a significant amount of time. A complementary approach to control transmission is contact tracing and quarantining, which are currently applied to sexually transmitted diseases (STDs). For STDs, identifying the contacts that might have led to disease transmission is relatively easy; however, for respiratory pathogens, the contacts that can lead to transmission include a huge number of face-to-face daily social interactions that are impossible to trace manually.

Method: We developed a Bayesian network model to process context awareness proximity sensor information together with (possibly incomplete) diagnosis information to track the spread of disease in a population. Our model tracks real-time proximity contacts and can provide public health agencies with the probability of infection for each individual in the model. For testing our algorithm, we used a real-world mobile sensor dataset of 80 individuals, and we simulated an outbreak.

Result: We ran several experiments where different sub-populations were infected and diagnosed. By using the contact information, our model was able to automatically identify individuals in the population who were likely to be infected even though they were not directly diagnosed with an illness.

Conclusion: Automatic contact tracing for respiratory pathogens is a powerful idea, however we have identified several implementation challenges. The first challenge is scala-

bility: we note that a contact tracing system with a hundred thousand individuals requires a Bayesian model with a billion nodes. Bayesian inference on models of this scale is an open problem and an active area of research. The second challenge is privacy protection: although the test data were collected in an academic setting, deploying any system will require appropriate safeguards for user privacy. Nonetheless, our work illustrates the potential for broader use of contact tracing for modelling and controlling disease transmission.

Acknowledgements

First and foremost, I'd like to express my gratitude to my supervisor Professor Daniel Lizotte for providing me with the opportunity to research with him. His invaluable support, guidance and encouragement have provided me with the motivation to constantly excel.

Moreover, I'd like to thank everyone in my academic "family" that I had for the past two years while studying here at University of Waterloo. I would like to send my sincere thanks to my brightest and smartest colleagues and lab-mates John Finnson, Younos Aboulnaga, Yuke Yang, Huangdong Meng and Hadi Hosseini for their lovely support, high intellectual discussions and warm sleepless nights we had together. I'm sure you all will excel in your life and professional careers. I'd like also to thank my mum Nehad, my sister Dina and brother Ahmed for all there endless emotional support and sacrifices they had to help me grow. Last but not least, I'd like to thank all the professors and administrative assistants here at the school of computer science at the University of Waterloo for giving me the opportunity for graduate education at this outstanding institution. My learning experience here was invaluable.

Dedication

To my parents.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Definition: Emerging Novel Influenza Viruses

The evolution of novel influenza viruses in humans that leads to influenza epidemics and pandemics is a biological phenomenon that can not be stopped. The evolved virus strains are usually combined with genes from human, mammals and/or bird flu. All existing data suggest that vaccination is our best line of defence against the morbidity and mortality of emerging outbreaks. As we witnessed during the time-line of the 2009 H1N1 Influenza pandemic, the swine flu emerged in Mexico in March in 2009 and in the first few weeks, the virus spread worldwide in 30 countries (as of May 2011) [41]. Unfortunately, the process of manufacturing the vaccine for a novel virus usually takes about 6 to 9 months, during which the virus would complete its spread.

Recently in the public health and health informatics research communities, there are

two efforts (complementing one another) to battle future influenza pandemics:

- Early detection of the novel influenza viruses by deploying robust, real-time and global syndromic surveillance systems and outbreak detection algorithms.
- Effective response to the outbreak by isolation and quarantining rather than relying on vaccination alone as a prevention method.

1.2 Early Detection of Novel Influenza Outbreaks

The data-driven syndromic surveillance systems involve collecting data from different data sources which are used for other different purposes. These data sources include emergency department (ED) visits, chief complaints, over the counter (OTC) drug sales from pharmacy stores, work or school absenteeism data, electronic health records, laboratory test orders, internet search queries (e.g. Google Flu Trends), etc. The promise of these deployed syndromic surveillance systems is to detect the outbreak as early as possible by detecting an unexpected rise in the signals in the monitored data-stream. The reason behind collecting multiple data sources rather than monitoring a single source such as ED chief complaints is that data sources vary in their quality and timeliness of sources and therefore they vary in their sensitivity, specificity and timeliness for the syndromic surveillance purposes. For example, monitoring ED chief complaint could be very correlated to a possible outbreak but there might be a delay of several days because the patients do not necessarily go and visit a clinic or hospital immediately once they start suffering from certain types of symptoms. On the other hand, internet search queries or work absenteeism data might be a very noisy data source, but they might provide a timely pre-diagnosis public health

indicator. Hence, there is an urgent need for information fusion in order to ensure overall high sensitivity, high specificity and timely detection of any possible outbreaks.

A second stage in the sydromic surveillance systems is called *syndromic classification*. Traditionally, the outbreak detection relies on observations of astute clinicians or laboratory-confirmed diagnoses. Since none of the data streams directly represent specific diagnostic information, the data streams are mapped to common syndromic categories. There are different approaches of mapping the syndromic surveillance data. The mapping can be done via a machine learning classifier [9] or by a rule based classifier [11]. The syndromic classes depend on the input data-stream(s) and the outbreak detection algorithm. Chapman et al. [9] developed a trained naive Bayes classifier that classifies chief triages from the University of Pittisburgh Medical Center into seven categories: respiratory, gastrointerstinal, neurological, rash, constitutional and hermorrhagic.

In a standard syndromic surveillance system, there will be outbreak detection algorithms that monitor these syndromic categories over space and time. We will talk in details about the different approaches for the syndromic surveillance algorithms in chapter 2.

1.3 Effective Response of Novel Influenza Outbreaks and Contact Tracing

Control of the spread of respiratory pathogens (e.g. novel influenza viruses) in the population using vaccination is a challenging problem that requires quick identification of the infectious agent followed by large-scale production and administration of a vaccine. This

takes a significant amount of time. In addition to the different vaccination strategies (e.g. random mass vaccination, age structured vaccination), isolation and quarantining of infected individuals is another effective method used by the public health agencies to control the spreading of infectious diseases [22]. Isolation is effective against any infectious disease; however it can be very hard to detect infectious individuals in the population when:

- Symptoms are ambiguous or easily misdiagnosed (e.g. 2009 H1N1 influenza outbreak shared many symptoms with many other influenza like illnesses)
- When the symptoms emerge after the individual become infectious.

In these cases, contact tracing is critical to identify newly infected cases [13] by tracing all potential past contacts with the detected infected individuals. Contact tracing is being used to control sexually-transmitted infections (STIs) (e.g. HIV). When a person reports symptoms of STI, he/she will be tested. If the test is positive, the person will be isolated, treated and asked for a list of sexual partners in the past 6 to 12 months, because these partners have higher likelihood of being infected compared to the general population. These contacts will be tested and if proven positively infected, they will be isolated, treated and further contact tracing will be done.

Contact tracing was not only used to control the spreading of STIs but also to control respiratory pathogen outbreaks such as SARS and influenza outbreaks [14, 42]. For control of STIs, the contacts that might have led to a possible transmission are easily identified (as long as infected individuals are willing to collaborate), however for respiratory pathogens, the contacts that lead to infections are face to face daily social contacts and interactions, which are impossible to trace.

The efforts in the previous work [42, 14] to use contact tracing to control SARS in Hong Kong and to control influenza were ineffective because of the inaccurate assumption that

the contacts that might have caused the transmission are only with the household of the identified infected individual [7].

Although social networks were used to assist epidemiologists in their response to emerging pandemics and/or epidemics [21, 24], epidemiologists currently do not have access to realtime information about daily social interactions of individuals to assist them in epidemiological models or contact tracing. In this project, we propose a novel contact tracing model in the form of a Bayesian Network (BN) that uses sensor information in cellphones (e.g. WiFi, GPS, Bluetooth) to automatically calculate for each individual in the population his/her probability of having been infected by a respiratory pathogen. Public health agents can rely on these probabilities for their decisions of isolations and/or vaccination. Our Bayesian model will solve the complexity and ineffectiveness of the labour intensive manual contact tracing processes being currently used by the public health agencies and it will have the potential to effectively isolate any future novel influenza viruses at the source rather than battling them around the planet by vaccination.

We have developed a novel probabilistic graphical framework and we designed a Bayesian inference algorithm for the contact tracing problem. In order to test the effectiveness of our Bayesian model we need a rich data-set of locations (via GPS and indoor WiFi) and social interactions (via Blue-tooth).

1.4 Thesis Outline

The following chapters are organized as follows. Chapter 2 gives a literature review of the different approaches for outbreak detection algorithms. Chapter 3 gives a mathematical background on Bayesian Networks and different learning, approximate and exact inference algorithms. Chapter 4 gives a formal description of the Bayesian Contact Tracing model

and its extensions for an individual based outbreak detection algorithm. In chapter 5 we simulated different outbreaks and tested the developed Bayesian network. In chapter 6 we will discuss the challenges of scaling the contact tracing model and the future work of large scale implementation of exact inference algorithms using Hadoop and Giraph.

Chapter 2

Literature Review

In recent years, sydnromic surveillance systems and outbreak detection algorithms have received considerable attention as public health agencies worldwide receive increasing pressure to battle emerging infectious diseases and bio-terrorist attacks. A large body of work exists in the machine learning and data-mining literature on outbreak detections and anomaly detection in healthcare-related data-streams. In this chapter, we will give an overview of the recent work in the literature on the topic of syndromic surveillance system and then we will discuss couple of proposals that discuss using contact tracing to battle infectious diseases.

2.1 Syndromic Surveillance Systems

Outbreak detection algorithms that currently being used in syndromic surveillance systems are typically classified into frequentist or Bayesian inference algorithms and into group-based and individual-based outbreak detection algorithms. The most deployed outbreak

detection algorithm in the existing syndromic surveillance systems is *"scan statistics"* algorithm by Kulldorff and Nagarwalla [27]. Although this work did not take the time dimension into account, later work generalized this method in the *"space-time scan statistic"* which includes the time-dimension in the outbreak algorithm [28, 26]. Extensions to the *scan statistic* and the *space-time scan statistic* algorithm consider using the expectation of the spatial counts of certain syndromes based on the historical data [35, 23].

Another recent proposal for a multivariate Bayesian spatio-temporal scan statistics (MBSS) framework [34] for early outbreak detection by Neill et al. Given a set $\boldsymbol{M}$ of multiple data streams D_m for $m = 1 \cdots M$, a set of spatial locations s_i for $i = 1 \cdots I$ and time sample $t = 0 \cdots T$. Define $c_{i,m}^t$ as the number of counts from the data source D_m, at the time sample t and at spatial location i. Moreover, we define a set of space-time regions $s \in S$ where each space-time region is non-empty set of locations s_i and time steps $W(s)$. We define a set of event types E_k for $k = 1 \cdots K$. Following the multiple hypothesis testing approach in the original Kulldorff's model, Neill [34] applied Bayes theorem to define the posterior probability of the null and alternative hypothesis given the data streams:

$$\Pr(H_1(s, E_k)|D) = \frac{\Pr(D|H(s, E_k)) \Pr(H_1(s, E_k))}{\Pr(D)} \tag{2.1}$$

$$\Pr(H_0|D) = \frac{\Pr(D|H_0) \Pr(H_0)}{\Pr(D)} \tag{2.2}$$

where $\Pr(H_1(s, E_k))$ and $\Pr(D)$ are the prior probabilities of the alternative and null (i.e. no event occurs) hypothesis respectively. $\Pr(D|H(s, E_k))$ and $\Pr(D|H_0)$ are the likelihoods of the data D given the alternative and null hypothesis respectively. The probability

of the data D is equal to $\Pr(D) = \Pr(D|H_0)\Pr(H_0) + \sum_{s,E_k} \Pr(D|H_1(s,E_k))\Pr(H_1(s,E_k))$.

The advantage of this model is its ability not to only to detect outbreaks but also to distinguish between different types of outbreaks.

Another Bayesian approach to the outbreak detection problem is given by Eagle et al. in [20], where it models a broad set of diseases and also models the spatio-temporal effect of the outbreaks using individual level data. The Bayesian network of this model is show in Figure 2.1.

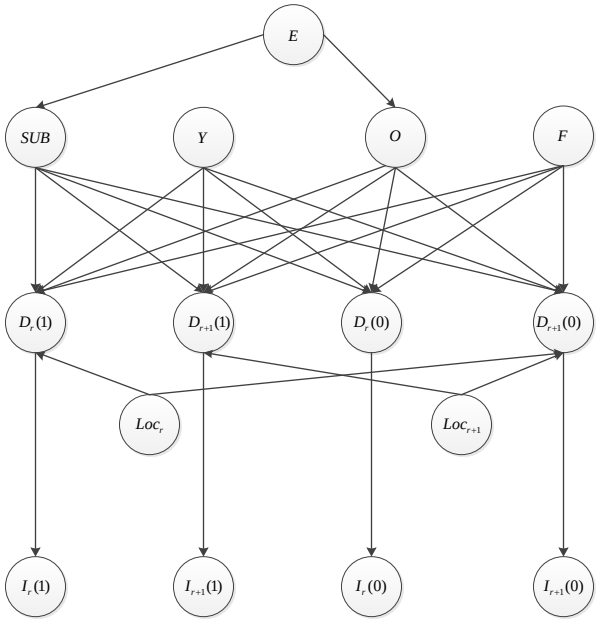


Figure 2.1: An individual based Bayesian outbreak detection model 4.1

The goal of the model is to compute the posterior probability of each outbreak disease type given the individual based emergency department chief complaint data. There is a

set of hidden random variables that describe the behaviour of the outbreak. There random variables are E, SUB, Y, O and F. The random variables E represents whether there is an ongoing outbreak outbreak and it takes two values: yes, no. The random variables SUB represents the subregions where the outbreak took place. In this proposal, the authors assumed a uniform popularity distribution SUB given that there is an ongoing outbreak (i.e. E =yes) and popularity of value *none* if E =no. The random variable O represents the different types of the outbreaks . In Bayesian network above, the random variable O takes 13 values, which are anthrax (two values) , plague (two values), smallpox, tularemia, botulism and hemorraghgic fever (two values), influenza, cryptosporidiosis and hepatitis A and a value of *none*. The conditional probability table (CPT) of random variable O given E was inferred from historical outbreak data. The random variable Y takes a numeric value which represents how many days into the outbreak. The random variable F represents the probability of an individual both having the simulated outbreak and going to the ED.

There are also a set of random variables that represents the state of the individuals such as $D_r(i)$ and Loc_r. $D_r(i)$ represents the hidden diseases state of individual i at time step r. This random variable takes the same values as the parameter O in addition to a value *other*. The random variable Loc_r is an observed random variable which represents the locations at which this individual had been to. Finally, there is the random variable $I_r(i)$ which represents the chief complaints of the individuals when they visited the emergency departments. In this model, the authors defined 55 values that the random variable $I_r(i)$ can take. The CPT of the random variables $I_r(i)$ given $D_r(i)$ can be elicited from a medical expert.

The goal of the outbreak detection problem is to infer:

$$P(E = yes|Data) = \sum_{d \neq none} P(O = d|Data) \tag{2.3}$$

Using Bayes Rule,

$$P(O = d|Data) = \frac{P(Data|O = d)P(O = d)}{\sum_{c} P(Data|O = c)P(O = c)} \tag{2.4}$$

the authors [20] derived mathematical formulas to calculate the prior, likelihood and posterior probabilities of the equation 2.4. The key contribution of the model is using individual based data for the outbreak detection problem, which inspired our work described in later chapters.

The major drawbacks of these studies is that they are not using any personalized user data such as location or history of social interactions in the outbreak detection problem. This leads to very low outbreak detection power for these syndromic surveillance systems[10]. Therefore, these existing syndromic surveillance systems are mainly used for situational awareness purposes more than for outbreaks detection.

2.2 Contact Tracing Studies

Another approach to the outbreak detection problem is to use novel data streams such as the history of social interactions. Given the rise of the Bluetooth and RFID enabled cellphones and devices, it has been come feasible and relatively easy for each individual to log the surrounding devices within meters ranges. A recent study [19], used RFIDs to

collect data about the social interactions of 200 people over the course of 70 days. This data stream could be very promising in the battle against of infectious diseases. In particular, after the rise of Blue-tooth low energy (BLE) devices [36]. However, the study [19] only discussed the data collection of this data. In chapter 4, we will discuss how to use that data-stream of proximity data for early detection and tracking the spread of infectious diseases in the population.

Chapter 3

Bayesian Networks

Suppose that we have a set of random variables that represent a group of correlated phenomena, features or observations such as detected features in a video frame, weather phenomena, etc. Our goal is to compute the distribution of $p(\mathbf{x}|\theta)$, where $\mathbf{x}$ is a set of hidden and correlated random variables and θ is a set of observations. In this chapter, we will discuss the representation and the computations of this *inference* problem.
The joint probability distribution between a set of random variables $\mathbf{M}$ can be represented by the *chain rule* as follows:

$$p(y_{1:\mathbf{M}}) = p(y_1)p(y_2|y_1)p(y_3|y_1,y_2)p(y_4|y_1,y_2,y_3)p(y_M|y_{\mathbf{M}-1}\cdots y_1) \tag{3.1}$$

If we assume that y_m is a a discrete random variable of $\mathbf{N}$ values, y_m can be represented in a table of $\mathrm{O}(\mathbf{N})$.

$p(y_2|y_1)$ is represented in a table of $\mathrm{O}(\mathbf{N}^2)$. $p(y_4|y_1,y_2,y_3)$ is represented in a table of $\mathrm{O}(\mathbf{N}^4)$ and hence $p(y_M|y_{\mathbf{M}-1}\cdots y_1)$ is represented in a table of $\mathrm{O}(\mathbf{N}^{\mathbf{M}})$. These are

called conditional probability tables (CPTs). Hence for large number of correlated random variables and large state space per variable, we can not store the corresponding CPT in memory or disk.

In order to simplify our representation of the joint probability distribution, we rely on a concept of conditional independence (CI), such that for a graph of nodes that represent a set of random variables, the presence of an edge between any two nodes in this graph represents the conditional dependence between these two nodes and the lack of this edge represents the conditional independence of these two nodes given the rest of the nodes in the graph.

There are different types of probabilistic graphical models, but in this thesis we are interested in the directed graphical models known as Bayesian networks.

3.1 Bayesian Networks Terminology

In this section, we will introduce a set of terminologies, related to Bayesian networks [33]:

- **Graph:** A graph $G(M, E)$ is a graph of M nodes and E edges.
- **Root:** is a node with no parent nodes.
- **Leaf:** is a node with no children.
- **Cycle or Loop:** is a series of connected nodes such that we can get back to the starting node by following directed edges.
- **Neighbours:** of a node is the set of nodes that are are all immediately connected to that node. Although this is a terminology that is related to undirected graphs, we will use it later in the section describing the inference algorithms.

- **Clique:** is a group of nodes that are all neighbours to each other. This is also a terminology that is used for undirected graphs, but we will use it in describing the junction tree algorithm in subsequent sections.

3.2 Conditional Probability Distribution

As we discussed in the previous section, representing the CPT is infeasible if we are dealing with nodes that have large number of parents and a big state space. A possible compact representation of the CPT is noisy-or or generalized linear model. Noisy-or representation is mainly used to represent binary nodes, and generalized linear model is mainly used for compact representation of CPTs of multi-valued discrete nodes and continuous nodes. We are interested in noisy-or representation of CPTs, because in modelling the problem of interest we are using binary random variables as will be discussed in Chapter 4.

In order to explain the concept of noisy-or nodes, assume that we have a very simple Bayesian network of three binary nodes A, X and Z. Nodes X and Z are parents for node A as show in Figure 3.1.

CPT representation of the conditional probability $p(A|X,Z)$ is a table of 2^3-1 entries. If we examine the effect of each node X and Z in isolation on A:

$$p(A=1|X=1,Z=0)=0.8$$

$$p(A=1|X=0,Z=1)=0.6$$

The key assumption here that the nodes X and Z are causally independent in terms of

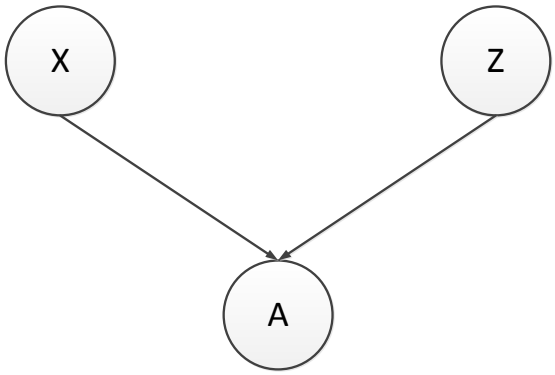


Figure 3.1: Simple Bayesian network of three nodes A, X and Z. Nodes X and Z are parents for node A

effect on node A and hence the combined effect of both nodes is :

$$p(A = 1|X = 1, Z = 1) = 0.48$$

As shown in table 3.1, we can construct the total CPT which has 8 entries by only knowing two values, which are the independent effect of the parent nodes (X , Z) on A. Hence the noisy-or representation allowed us to represent the CPT in a compact form of $O(N)$ instead of $O(2^N)$, where N is the number of parent nodes. Let's assume that we have a node A that has N parents, $C_1 \cdots C_N$, the probability $P(A|par(A))$ given by:

$$\begin{aligned} P(A|par(A)) &= P(A|C_1 \cdots C_N) \\ &= 1 - \prod_i (1 - P(A|C_i)) \\ &= 1 - \prod_i (1 - p_i) \end{aligned} \tag{3.2}$$

where p_i is the independent *influence probability* of each parent.

X	Z	$p(A = 0\|X, Z)$	$p(A = 1\|X, Z)$
0	0	1	0
0	1	1-0.6	0.6
1	0	1-0.8	0.8
1	1	(1-0.6) $\times$ (1-0.8)	0.6 $\times$ 0.8

Table 3.1: Noisy-OR CPD of a node with two parents

3.3 Exact Inference Algorithms

In this section, we will discuss in details two famous exact inference algorithms that are used to perform exact inference on Bayesian networks. These two algorithms are the variable elimination algorithm and the junction tree algorithm.

3.3.1 Variable Elimination Algorithm

The variable elimination algorithm is based an idea called **bucket elimination** [43].

Consider the directed graphical model in Figure 3.2 [25]. It's a Bayesian network of 8 nodes. Without a loss of generality, let's assume that all the eight nodes are binary.

The joint probability distribution of this model is given by [33]:

$$\mathrm{P(C, D, I, G, S, L, J, H) = P(C)P(D|C)P(I)P(G|I, D)P(S|I)P(L|G)P(J|L, S)P(H|G, J)} \tag{3.3}$$

The variable elimination algorithm is applicable to the inference problem for both directed and undirected graphs with the same computational complexity [43].

The process of transforming a directed graph to undirected one is called **moralization**. Figure 3.3 shows the undirected version of the Bayesian network in Figure 3.2. Note that

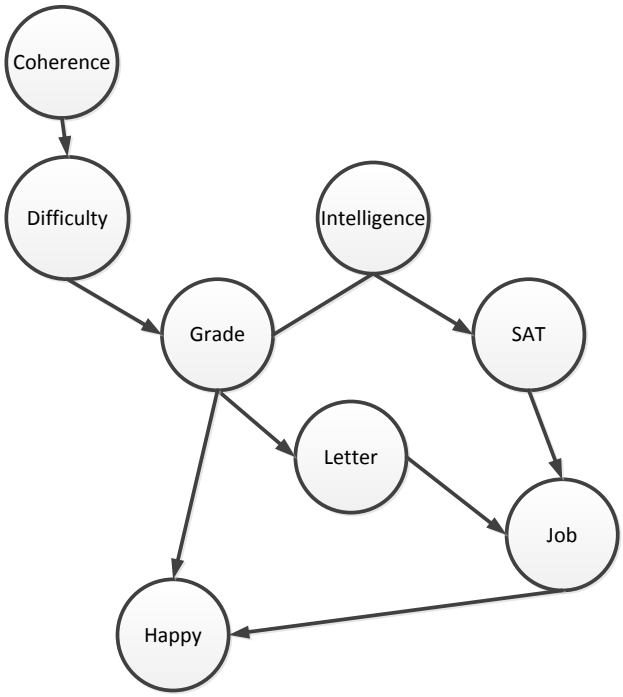


Figure 3.2: Simple Bayesian network of eight nodes that represent a student performance at college [25]

there are more edges in the undirected version than in directed version. The moralization process links any nodes that share the same edges, to ensure that conditional independence properties in the undirected graph matches the ones in the directed graph [33]. To transform a directed graph to undirected one, we define a *factor* ζ for each conditional probability distribution in the directed graph. For example:

$$\mathrm{P(C,D,I,G,S,L,J,H) = P(C)P(D|C)P(I)P(G|I,D)P(S|I)P(L|G)P(J|L,S)P(H|G,J)} \tag{3.4}$$

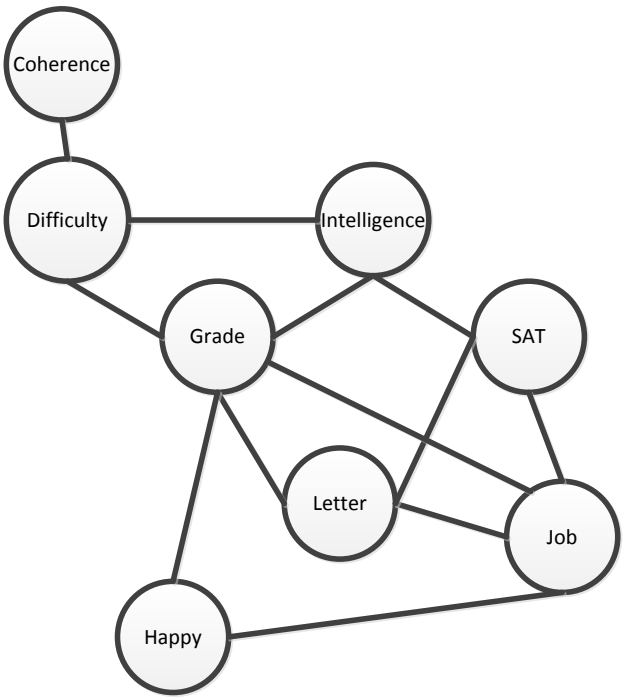


Figure 3.3: A Unidirected graph representation of the student performance graph after the moralization process [25]

$$\mathrm{P(C,D,I,G,S,L,J,H)} = \zeta(\mathrm{C})\zeta_D(\mathrm{D,C})\zeta_I(\mathrm{I})\zeta_G(\mathrm{G,I,D})\zeta_S(\mathrm{S,I})\zeta_L(\mathrm{L,G})\zeta_J(\mathrm{J,L,S})\zeta_H(\mathrm{H,G,J}) \tag{3.5}$$

Let's assume that we want to compute $P(J)$, this can be easily computed by marginalizing over the rest of the random variables in the joint probability distribution.

$$P(J) = \sum_C \sum_D \sum_I \sum_G \sum_S \sum_L \sum_H \mathrm{P(C,D,I,G,S,L,J,H)} \tag{3.6}$$

$$P(J) = \sum_{C} \sum_{D} \sum_{I} \sum_{G} \sum_{S} \sum_{H} \sum_{L} \zeta(\mathrm{C})\zeta_D(\mathrm{D},\mathrm{C})\zeta_I(\mathrm{I})\zeta_G(\mathrm{G},\mathrm{I},\mathrm{D})\zeta_S(\mathrm{S},\mathrm{I}) \times \zeta_L(\mathrm{L},\mathrm{G})\zeta_J(\mathrm{J},\mathrm{L},\mathrm{S})\zeta_H(\mathrm{H},\mathrm{G},\mathrm{J}) \tag{3.7}$$

Obviously the summation in Equation 3.7 has 2^7 terms. We compute the marginalization faster by pushing the summation inside the multiplication, as follows:

$$P(J) = \sum_{C} \sum_{D} \zeta_J(\mathrm{J},\mathrm{L},\mathrm{S}) \sum_{I} \zeta_L(\mathrm{L},\mathrm{G}) \sum_{H} \zeta_H(\mathrm{H},\mathrm{G},\mathrm{J}) \times \sum_{I} \zeta_I(\mathrm{I})\zeta_S(\mathrm{S},\mathrm{I}) \sum_{D} \zeta_G(\mathrm{G},\mathrm{I},\mathrm{D}) \sum_{C} \zeta(\mathrm{C})\zeta_D(\mathrm{D},\mathrm{C}) \tag{3.8}$$

If we start from right to left, this will lead to a temporary potential.

$$\gamma_1(C,D) = \zeta(\mathrm{C})\zeta_D(\mathrm{D},\mathrm{C}) \tag{3.9}$$

If we sum over C this will lead to a second temporary potential as a function of D.

$$\gamma_2(D) = \sum_{C} \gamma_1(C,D) \tag{3.10}$$

After multiplying the temporary potential and marginalizing over D, we get:

$$\gamma_3(G,I,D) = \zeta_G(\mathrm{G},\mathrm{I},\mathrm{D})\gamma_2(D) \tag{3.11}$$

$$\gamma_4(G,I) = \sum_{D} \gamma_3(G,I,D) \tag{3.12}$$

Repeating the last two steps and marginalizing over I:

$$\gamma_5(G,I,S) = \zeta_I(\mathrm{I})\zeta_S(\mathrm{S},\mathrm{I})\gamma_4(G,I) \tag{3.13}$$

$$\gamma_6(G, S) = \sum_I \gamma_5(G, I, S) \tag{3.14}$$

And so on. This process is called variable elimination (VE).

Computational complexity and weakness of VE algorithm

The computation complexity of the VE algorithm is linear in:

- size of the model (number of factors, number of variables).
- size of the largest factor generated.

The size of the largest factor generated is exponential in the size of the largest potential. The largest potential could be found in the original graph or one of the temporary potentials generated during the variable elimination process. The process of generating the temporary potentials is dependent on the elimination order of the random variable. The process of finding the elimination order that will lead to the smallest size of potentials is NP hard [5]. Moreover, the exact-inference problem given the correct elimination order is NP hard itself [25].

Another disadvantage of VE is that if we want to compute the probabilities of different nodes given the same evidence, we have to recompute the VE algorithm from scratch; this will lead to inefficiency in the computations especially for large graphical models.

3.3.2 Junction-Tree Algorithm

The junction tree algorithm overcomes the deficiencies of the VE by sharing many of the computational steps when we are trying to compute the conditional probabilities of different

nodes given the same evidence. In this subsection, we will sketch the basic overview of the junction tree algorithm but for details see the text by Koller and Friedman [25].

The junction tree algorithm starts with the same steps of the variable elimination algorithm such as moralization of the directed graph, adding edges according to a certain elimination order. The elimination order is chosen to transform the undirected graph generated from the VE process to a certain type of graph called ***chordal graph***. A chordal graph is a graph where each cycle of length ≥ 4 has an edge connects all non-adjacent nodes [33]. For example, Figure 3.4 is not a chordal graph because cycle 1,2,3,4,5,6 is chordless, but in Figure 3.5 after connecting nodes (2,5) and (3,5)the graph will be a chordal graph. This process is called *triangulation*.

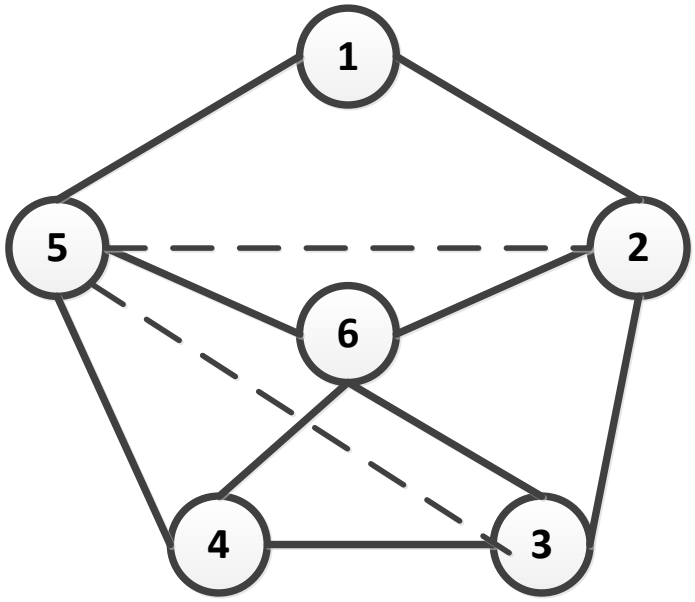


Figure 3.4: Chordal Graph

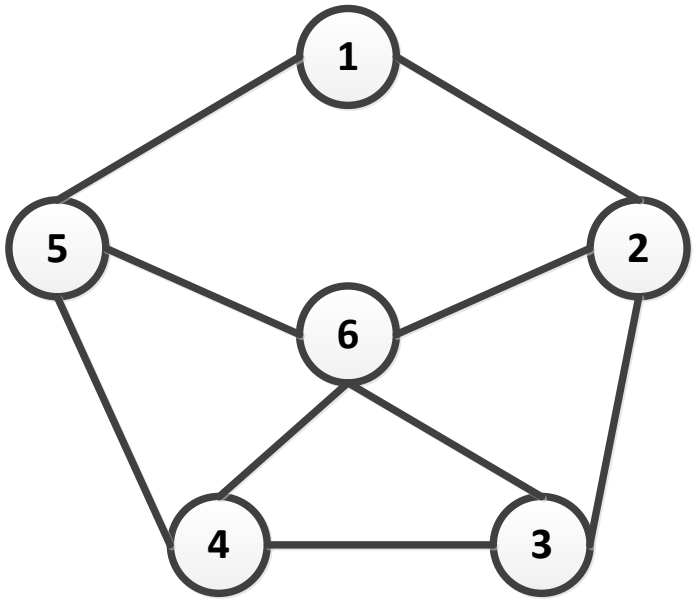


Figure 3.5: Non-Chordal Graph

Chordal graphs have two interesting and useful properties:

- Although finding the maximum number of cliques is computationally hard, in general it is computationally efficient to find the maximum cliques on chordal graphs.
- Chordal graphs enjoy a property called the ***running intersection property***, such that each nodes (i.e. cliques) that contain a give variable are connected components.

The last step in the junction tree algorithm is a process that is very similar to applying the belief propagation algorithm on a tree but it will be applied to a junction tree. The message passing is done where the cliques are the nodes and the separators are the edges.

For illustrative purposes, we built a chordless graph for the student Bayesian network as shown in Figure 3.6 [33].

Figure 3.6: Junction tree of the student graph

The original model has the following form:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C(G)} \psi_c(\mathbf{x}_c) \tag{3.15}$$

The message passing algorithm passes messages from leaves to roots. Initially, the evidence messages at the separators initially is set to 1. The message from node i to node j.

$$m_{i \to j}(S_{ij}) = \sum_{C_i \setminus S_{ij}} \psi_i(C_i) \tag{3.16}$$

Once the clique node j receives the belief message from node i and updates its belief state.

$$\psi_i(C_i) \alpha \psi_i(C_i) \prod_{j \in ch_i} m_{j \to i}(S_{ij}) \tag{3.17}$$

At the root clique node, the potential $\psi_r(C_r)$ represents the posterior probability of the root clique given the evidence from the rest of the graph.

Now the forward message passing is done, the message passing algorithm starts to send messages from the root clique node to the leaf. The messages from sent back from i to j

is divided by the message from j to i.

$$m_{i \to j}(S_{ij}) = \frac{\sum_{C_i \setminus S_{ij}} \psi_i(C_i)}{m_{j \to i}(S_{ij})} \tag{3.18}$$

After sketching out the basic idea of the junction tree algorithm, it is obvious that it takes $O(|c|k^{w+1})$ in time and space, where $|c|$ is the number of cliques and w is the size of the largest clique. The main advantage of the junction tree algorithm over the variable elimination algorithm is that junction tree algorithm always us to calculated the posteriori probabilities and enter evidence without re-running the algorithm from scratch every single time.

3.4 Approximate Inference Algorithms

As we have seen both the variable elimination and the JT algorithms are both exponential in the tree-width. In the worst case scenario, the algorithm can be exponential in the number of nodes. Hence, these exact inference algorithms could be intractable for large networks. In fact, it was shown that the exact inference problem is N-P hard problem[39]. There are many proposals for approximate inference algorithms for the Bayesian networks such as loopy belief propagation, convex belief propagation [18], mean field approximation [38], Gibbs sampling based approximate inference [40], etc. Discussing the different approximate inference algorithms and their usages and trade-offs is beyond the scope of this thesis.

3.4.1 Loopy Belief Propagation

The basic idea of the loopy belief propagation is to apply the belief propagation algorithm on the original graph. Applying belief propagation on graph with loops does not guarantee accurate results or convergence, but it often works well [2].

We define the following notations:

- $\lambda_Y(X)$ is the message from X to child node Y.
- $\pi_X(U)$ is the message to X from parent node U
- $\lambda_X(X)$ is the message to X to itself if the node itself is observed.
- $\lambda^{(t)}(X)$ is the message at iteration t
- α is a normalizing constant

The incoming messages to the node X at t are shown in Figure 3.7 and be represented by the following equations:

$$\lambda^{(t)}(x) = \lambda_X(x) \prod_j \lambda_{Y_j}^{(t)}(x) \tag{3.19}$$

$$\pi^{(t)}(x) = \sum_{uk} P(X = x|U = u) \prod_k \pi_X^{(t)}(u_k) \tag{3.20}$$

The output messages from node X at $t+1$ are shown in Figure 3.8 and be represented by the following equations:

$$\lambda_X^{(t+1)}(u_i) = \alpha \sum_x \lambda^{(t)}(x) \sum_{u_k : k \neq i} P(x|u) \prod_{k \neq i} \pi_X^{(t)}(u_k) \tag{3.21}$$

$$\pi_{Y_j}^{(t+1)}(x) = \alpha \pi^{(t)}(x) \lambda_X(x) \prod_{k \neq i} \lambda_{Y_k}^{(t)}(x) \tag{3.22}$$

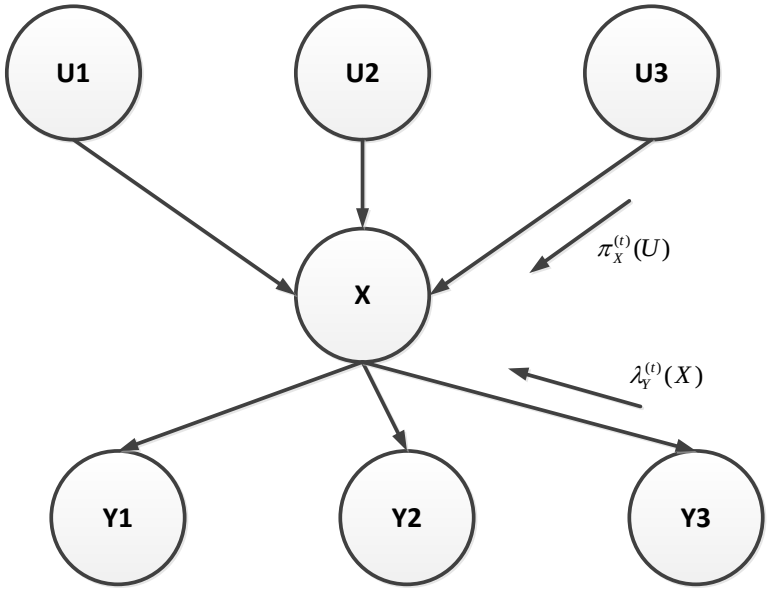


Figure 3.7: Incoming messages to the node X at time t

The beliefs (i.e. π) of all nodes are updated in parallel until convergence. The algorithm is initialized by setting the beliefs of all nodes to 1. At each iteration, all nodes calculate their outgoing messages based on the input messages from their neighbours at the previous time step . The algorithm converges if the beliefs of all node at a time step is very close to the beliefs of the previous time step. If the algorithm doesn't converge, it will oscillate between two values [2].

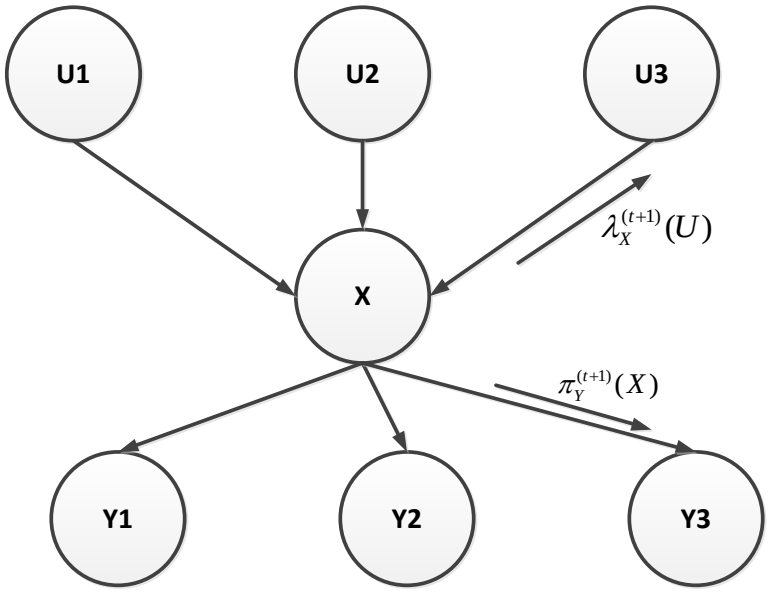


Figure 3.8: Output messages from the node X at time $t+1$

Chapter 4

Contact Tracing Model

In this chapter, we will discuss the process of building a Bayesian network to model the contact tracing problem. Although the contact tracing problem is a time-series problem, which suggests that dynamic Bayesian networks and/or HMM would be an excellent fit for modelling the problem, we used a non-dynamic Bayesian network. The reason behind our choice is that DBNs and HMMs assume that the nodes and edges are identical at each time-step. This would mean that at each time step we would need to represent all possible interactions between any pair of individuals, *whether contact was made or not*. Not only this will lead to the explosion in the number of edges needed per time step, which is of order $T \times \frac{n!}{(n-2)!}$ but also will lead to intractability of the inference algorithms as the size of the maximum cliques will increase as we have seen in Chapter 3. After first describing the model we will discuss the similarities and difference with the DBN model in section 4.3.3.

4.1 Contact Tracing Problem

Contact tracing concept is currently being used by public health agencies to control the spread of sexually transmitted diseases (STDs) [16, 13, 15]. Contact tracing is fundamentally linked to the individual-level spread of infection and in particular, the network of potential transmission. In STDs, the contact tracing is done by questioning the identified positively infected individuals about the history of their social interactions (i.e sexual partners) in the past few months. This infected individual will be treated and/or quarantined. The public health agencies will go and further examine and test the list of susceptible individuals. Whoever is positively infected in this list will be treated, isolated and/or asked for a list of history of social contacts (i.e. sexual partners). Then further contact tracing will be done. Two points to notice here:

1. Contact tracing is a very labour intensive process, making scaling and applying contact tracing to the rest of the population impossible and infeasible.

2. For STDs, the infected individuals can easily remember the history of sexual partners, as along as they are willing to participate.

Our contribution in the thesis is to use the concept of contact tracing to control the spreading of respiratory infectious diseases such as SARS, flu, etc. Doing contact tracing on respiratory pathogens instead of STDs is challenging because the transmission can happen because of any face to face social interaction, and these interactions are impossible to track over an extended period of time. Secondly, the spread of respiratory pathogens in the population is much faster than the spread of STDs. Hence, we are proposing the following:

- Proposing using a novel data source for the contact tracing problem, which is the context awareness data from cell phones.

- We designed a tool based on Bayesian inference that will allow public health agents to effectively do contact tracing

4.2 Context-Awareness Data

Location-aware devices (e.g. cellphones, tablets, etc.) are spread all over the world that allow us to know a precise location of each individual. However, in 2012 there was a spread of using Bluetooth low energy (BLE) enabled devices [36]. The data collected from BLE-enabled devices enables all the devices to be contextually aware by not only knowing the location of the device but also by collecting information about the surrounding devices and regions within a proximity area of 1-10 meters. There are recent studies that discus the idea of using cellphones to do periodic Blue-tooth scan of the surrounding MAC address of the Blue-tooth devices in 1-10 meters range [12, 6, 7, 3].

Our goal in this thesis is to use these proximity data from contextually aware devices to track the social interaction of individuals and to build mathematical models that will assist the public health agencies with their contact tracing task.

4.3 Bayesian Network for Contact Tracing problem

In section, we will discuss the process of building a Bayesian network for the contact tracing problem. Let us assume that we have a time-series data-set that logs the face to face interactions of a group of individuals at each-time step. Our goal is build a probabilistic model that uses this time-series proximity data and health-care status (i.e. infected or not) of a subset of the population at certain time steps and provides us with the likelihood of infection of the rest of population at each time step.

4.3.1 Model Structure

The contact tracing Bayesian network is generated by the following rules:

- The total number of nodes in the Bayesian is $N \times T$, where N is the total number of individuals in the data-set and T is the total number of time-steps.
- Each node is a binary node that represents whether the person is infected at this time step. Each node takes two values: infected or not infected. We write the node for person I at time-step t as I_t.
- There is an edge between I_t and $I_t + 1$
- If there is a face-to-face contact between person I and person J at time t, there are two edges $(I_t, J_t + 1)$ and $(J_t, I_t + 1)$.
- At each time step, there will be edges between the nodes that represents the transitive closure of contacts at this time step. For example at a certain time step t, there are two contacts between individuals (2, 19) and (2, 5), then will be 6 edges $(2_t, 5_{t+1}), (2_t, 19_{t+1}), (5_t, 19_{t+1}), (5_t, 2_{t+1}), (19_t, 2_{t+1})$ and $(19_t, 5_{t+1})$ as shown in Figure 4.1.

4.3.2 Model Parameters

There are two types of nodes for which we must define model parameters.

- **Initial nodes** at the first time-step have no parents; for these nodes we must specify the prior probability of infection $P(I_0 == 1)$, which could be elicited from experts or learned from previous data.

- **Internal Nodes** at subsequent time-steps are defined using a Noisy-OR model. In such a model, the probability $P(I_t|parents)$ is computed using the "influence probability" for each of the parents. In general, the influence probability of I_{t-1} on I_t should be high, since we expect infected individuals to remain infected on the timescales we examine. The influence probability of J_{t-1} on I_t should reflect expert knowledge about the probability of disease transmission based on social contact.

For illustration, assume that we have a data-set of five individuals (A,B,C,D,E) over 4 time-steps. The data-set consists of the history of contacts of these five individuals as shown in the following Table 4.1:

time step	Contact 1	Contact 2
1	A	B
1	B	C
1	C	D
2	A	E
2	C	B
3	A	D
4	A	C
4	C	E

Table 4.1: A sample contact data of five individuals over 4 time-steps

Following the rules that mentioned above, we used the sample data to generate the model in Figure 4.1.

Given the model in Figure 4.1, a public health agent can use it to answer questions such as: If person A is positively infected at the second time-step what is the probability of infection of everyone else of the population at time-step 4? Or, if C and E are positively infected at the third time step, what is the probability of infection of everyone else at the last time step?

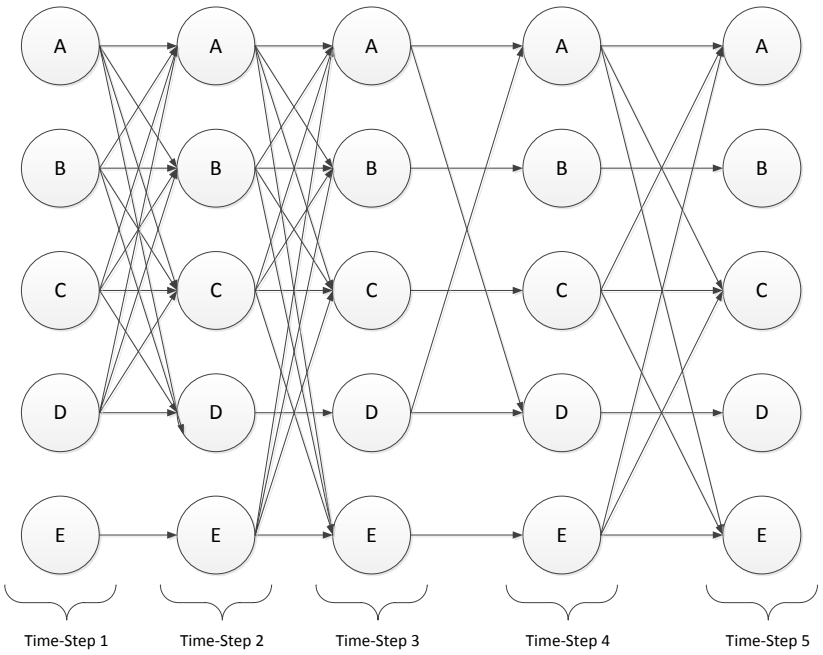


Figure 4.1: A contact Bayesian network generated for the data in table 4.1

Depending on the size and the complexity of the model, we may use an exact inference algorithm or an approximate inference algorithm to answer these questions. Given the answers public health agents can better prioritize the list of individuals that they examine first, and once they find a positively infected person, they set his add as positively infected to the evidence and rerun the inference algorithm on the rest of the individuals to re-compute their probability of infection.

4.3.3 Network Size Versus Full DBN

Note that in Figure 4.1, that the total number of edges is dependent on the data that was used to build the model. In the worst case scenario, where all the nodes interacted with each other at all time steps, the number of edges will be equal to $T \times {}^{N}\mathrm{P}_2 + T \times N$. In case we implemented our model using DBN there will be $2 \times T \times {}^{N}\mathrm{P}_2 + T \times N$ edges in the model that will increase the sizes of the cliques. It is natural to think about this model as a DBN. However, in the DBN formulation, we know from the contact information that many edges could be omitted and we would get exactly the same results with much less computation. An intelligent inference algorithm could do this automatically; however our approach is to "pre-compile" the DBN formulation into a bayes net with many fewer edges so that off-the-shelf methods can be applied. To illustrate the increase of the number of nodes and hence the size of cliques in the DBN representation, consider Figure 4.2. Variables A and B represent infectious status of two individuals. Variable C_{AB} represent whether there is a contact between A and B. Variable T_{AB} and T_{BA} represent whether there is transmission from A to B or from B to A, respectively. Note that in the DBN representation there will be 7 random variables to represent the possible virus transmission between any pair of individuals instead of 4 random variables in the BN version of the model.

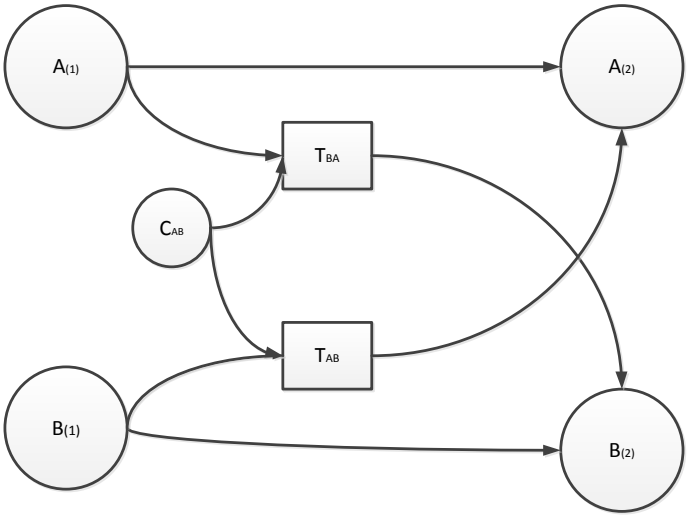


Figure 4.2: DBN representation of the contacts between two individuals

Chapter 5

Outbreak Simulations

In this chapter, we build our contact tracing Bayesian network model based on a real-world sociopatterns data-set [19]. The dataset that we used to build the contact tracing model consists of monitored face-to-face proximity contacts of 80 visitors of a science museum in Europe. The original data are collected by RFID tags placed in the name tags of the attendees that every 20 seconds scan the surrounding RFIDS within 1 to 1.5 meters range. In section 1, we will describe the data-set and how we used it to build the Bayesian network model. In section 2, we will show simulation results for different types of outbreaks. As proof of concept for using the contact tracing data to track the spread of pandemics in the population, for simplicity we assumed without the loss of generality that the outbreak could be an influenza outbreak. However, the model can be easily used to model different infectious diseases by using appropriate probabilities. In the first type of outbreaks we examine, there is a single person infected in the population who walked to the museum. In the second type of outbreaks we examine, there are two individuals who are positively infected. In the third type of outbreaks is where there are ten individuals who are positively

infected. The fourth type of outbreaks is where there is ten individuals. For each type of outbreaks, we will study the effect of these positively infected on the rest of the population in terms of the likelihood of infection. Finally, in section 3, we will discuss the validation of our model over different types of outbreaks.

5.1 Bayesian Contact Tracing Model

In this section, we will describe the process of generating the Bayesian network using the sociopatterns data. The ideal data-set to validate our model would be a dataset that is collected tracking the proximity contacts of a group of individuals over the course of few days because of the incubation period of virus . That would be the contacts within the household, at work, at transportations, etc.

5.1.1 The Sociopatterns Data-Set

The data was collected at the Science Gallery in Dublin, Ireland from April 17 to July 17, 2009. The data collection is done by using active Radio-Frequency Identification Devices (RFID) placed in the name-tags of the visitors of the exhibition. These RFID tags exchange ultra low power messages every 20 seconds with the surrounding RFID devices in 1 to 1.5 meters range. The data generated by these RFID sensors can give us a good approximation of the face-to-face interactions among the visitors of the exhibition. Although the data has been collected over the course of 69 days, we used the data from only one day because the exhibitions expect new visitors every single day and hence the collected proximity data from different days are not correlated. The number of visitors per day to the exhibition vary per day from 80 to 305 visitors.

We built a Bayesian network as described in chapter 4, using the following parameters:

- The transition probability of infection for each node to the same node at the next time step is 0.9615. $P(I_{t+1} = 1|I_t = 1) = 0.9615$
- The prior probabilities of infection for each node at time step 0 is 0.05. $P(I_0 = 1) = 0.05$
- The CPDs of all nodes are noisy-OR nodes with two types of parents. The parents could be the same node at the previous time step of influence probability of 0.9615 or different nodes with influence probability of 0.8. $P(I_{t+1} = 1|J_t = 1) = 0.8$.

These probabilities were chosen after trial and error to be able to be easily visualize the effect of the outbreak on the short-period data-set (i.e. 38 time-steps) we are using. We tested different time step lengths the dataset. A very short window of 20 seconds will generate a large a number of nodes in the model but fewer edges per time step. We re-sampled the data using a window size of 15 minutes instead of 20 seconds. Since the exhibition was open for ten hours, that generated 39 samples. A real contact tracing dataset both the time-scale and the duration of the data-set will be larger.

5.2 Simulated Outbreaks

In this section, we describe three different types of simulated outbreaks:

- Outbreaks where there is only a single person infected in the population.
- Outbreaks where there are two individuals who are positively infected with an infectious disease.

- Outbreaks where there are five individuals who are positively infected with an infectious disease.
- Outbreaks where there are ten individuals who are positively infected with an infectious disease.

Firstly, we ranked the individuals in the data-set according to their *social rank*. We define social rank of an individual as the total number of face to face contacts this person made during the process of the collection of the data set. Depending on the type of outbreaks, we assumed that certain individuals in the population are positively infected with the infectious disease (e.g. novel influenza virus). We ran our Bayesian network model on a Linux machine of 64 GB RAM. We tested different exact inference algorithms such as junction tree algorithm and variable elimination algorithm using Matlab and the Bayes net toolbox [32], but all ran out of memory. Therefore, we used an approximate inference algorithm known as Pearl's inference algorithm or the belief propagation algorithm. Our goal is to infer the the progression of the likelihood of infection of every single individual as the infected individuals start to make more face-to-face contacts with the rest of the population. We visualised the progression of the likelihood of individuals' infection in heatmaps, where the x-axis represents the IDs of the individuals and the y-axis represents the time-steps. Each block in the heatmap represents the probability of infection of the individual with this ID at this time-step.

5.2.1 Single Individuals Outbreaks

In this section we simulated two outbreaks where:

1. Most social individual is positively infected.

Table 5.1: Social Rank of the individuals in the dataset

ID	Social Rank	ID	Social Rank	ID	Social Rank	ID	Social Rank
51	1	43	21	21	41	27	61
64	2	13	22	38	42	75	62
54	3	80	23	55	43	44	63
35	4	5	24	46	44	57	64
29	5	73	25	33	45	40	65
30	6	48	26	72	46	65	66
3	7	34	27	25	47	18	67
26	8	63	28	79	48	56	68
31	9	23	29	7	49	24	69
11	10	2	30	58	50	39	70
6	11	71	31	15	51	8	71
9	12	16	32	22	52	61	72
62	13	74	33	12	53	50	73
36	14	47	34	49	54	76	74
17	15	1	35	53	55	78	75
10	16	68	36	69	56	19	76
37	17	45	37	42	57	41	77
32	18	67	38	60	58	66	78
4	19	14	39	59	59	52	79
20	20	70	40	77	60	28	80

2. Least social individual is positively infected.

The heatmap of the outbreak where the most social individual is positively infected of ID 51 is shown in Figure 5.1. The heatmap of the outbreak where the least social individual is positively infected of ID 52 is shown in Figure 5.3.

5.2.2 Two Individuals Outbreaks

In this section we simulated two outbreaks where:

1. Two most social individuals are positively infected.

2. Two least social individuals are positively infected.

As shown in Figure 5.5, the two most social individuals of IDs 51 and 64 have probability of 1.0 in the heatmap (i.e. set as evidence). In Figure 5.7, the two least social are 28 and 52 are set as positively infected.

5.2.3 Five Individuals Outbreaks

In this section we simulated two outbreaks where:

1. Five most social individuals are positively infected.

2. Five least social individuals are positively infected.

As shown in Figure 5.9, the five least social individuals of IDs 19, 28, 41, 52 and 66 have probability of 1.0 in the heatmap (i.e. set as evidence). In Figure 5.11, the five most social are 29, 35, 51, 54 and 64 are set as positively infected.

5.2.4 Ten Individuals Outbreaks

In this section we simulated two outbreaks where:

1. Ten most social individuals are positively infected.

2. Ten least social individuals are positively infected.

As shown in Figure 5.13, the ten least social individuals of IDs 8, 19, 28, 41, 50, 52, 61, 66, 76 and 78 have probability of 1.0 in the heatmap (i.e. set as evidence). In Figure 5.15, the ten most social are 3, 11, 26, 29, 30, 31, 35, 37, 51, 54 and 64 are set as positively infected.

Figure 5.1: Heatmap of the likelihood of infection where the most social person is positively infected: (a) individuals 1-40

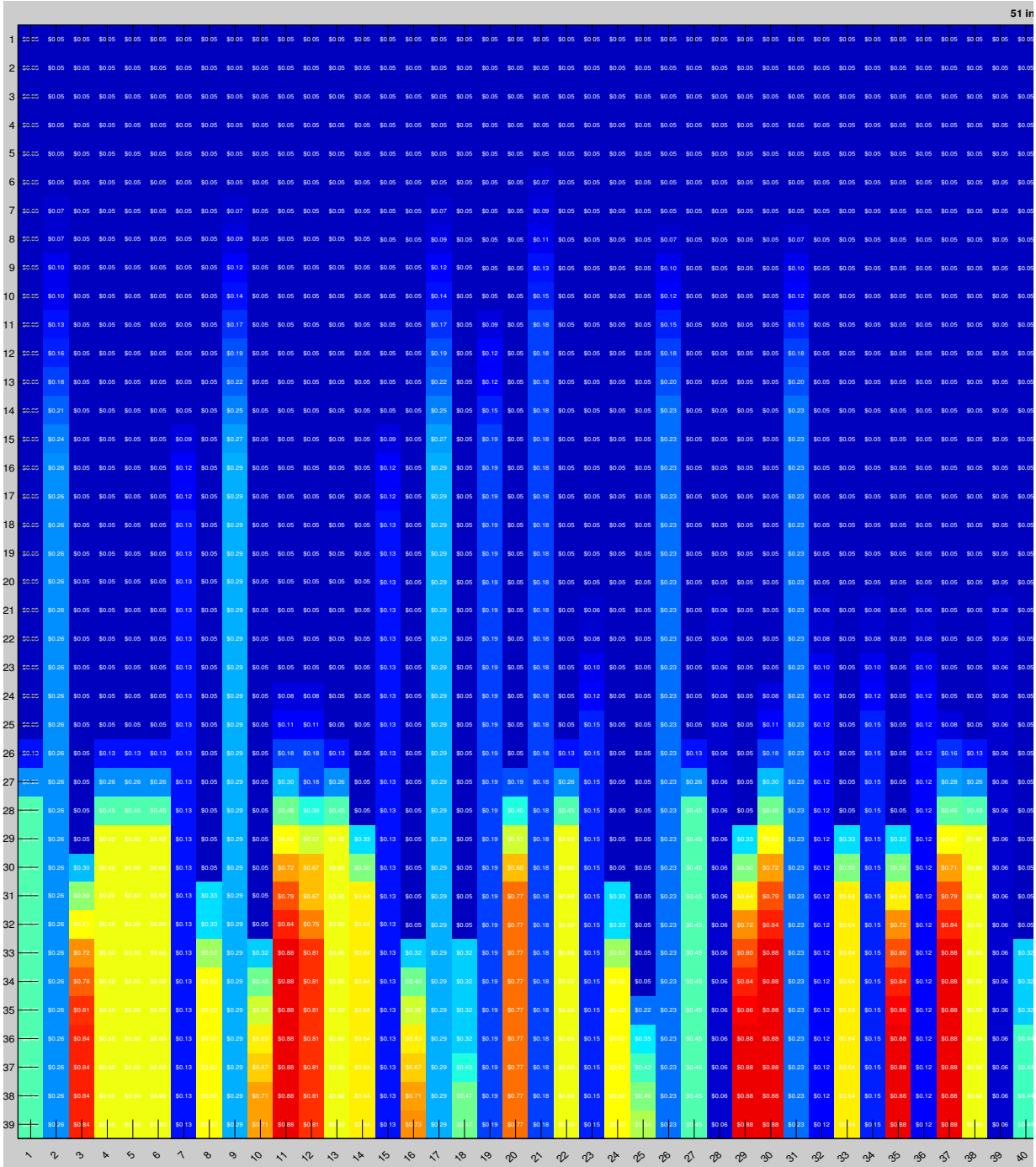


Figure 5.2: Heatmap of the likelihood of infection where the most social person is positively infected: (b) individuals 41-80

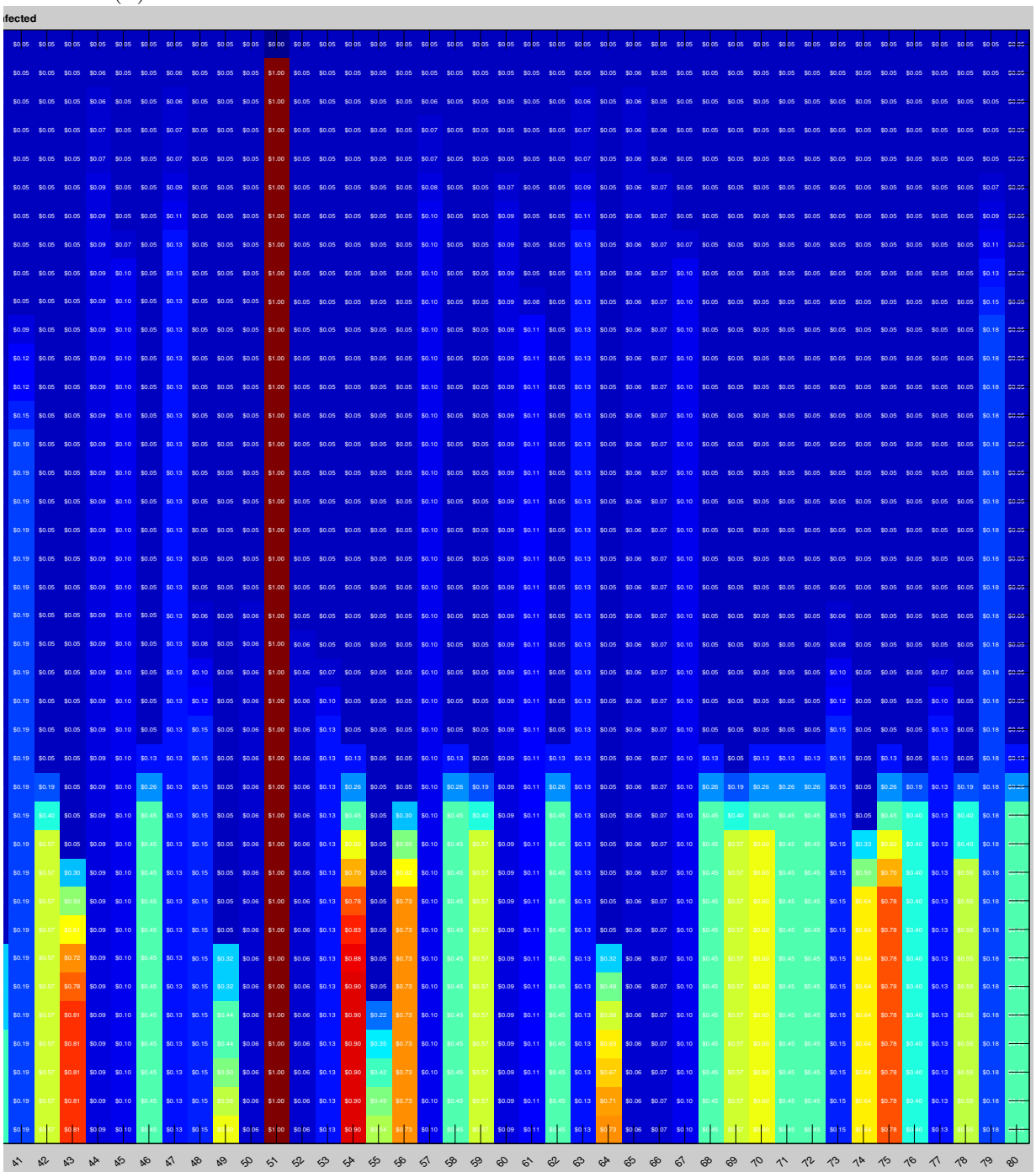


Figure 5.3: Heatmap of the likelihood of infection where the least social person is positively infected: (a) individuals 1-40

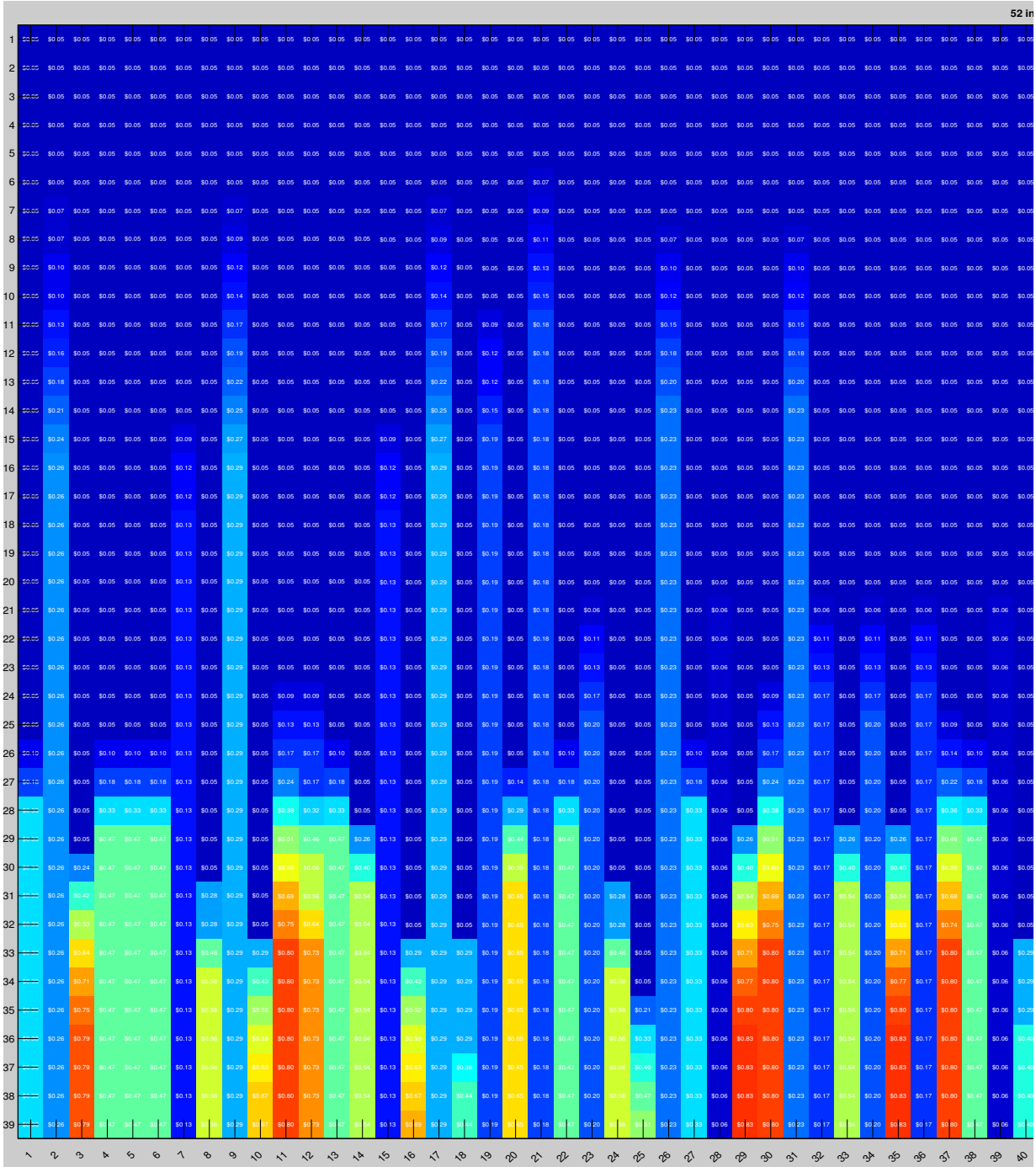


Figure 5.4: Heatmap of the likelihood of infection where the least social person is positively infected: (b) individuals 41-80

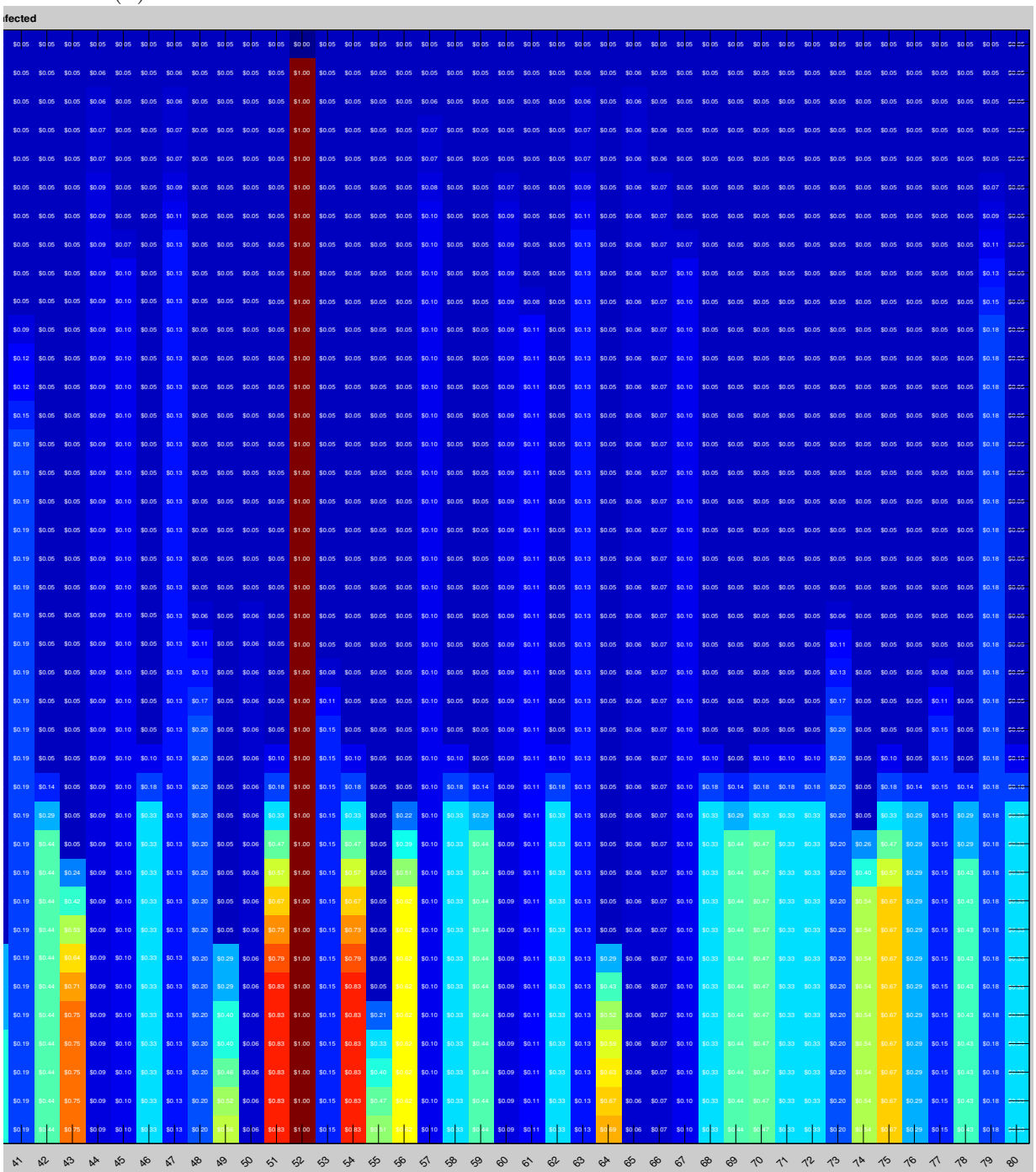


Figure 5.5: Heatmap of the likelihood of infection where the most social two individuals are positively infected: (a) individuals 1-40

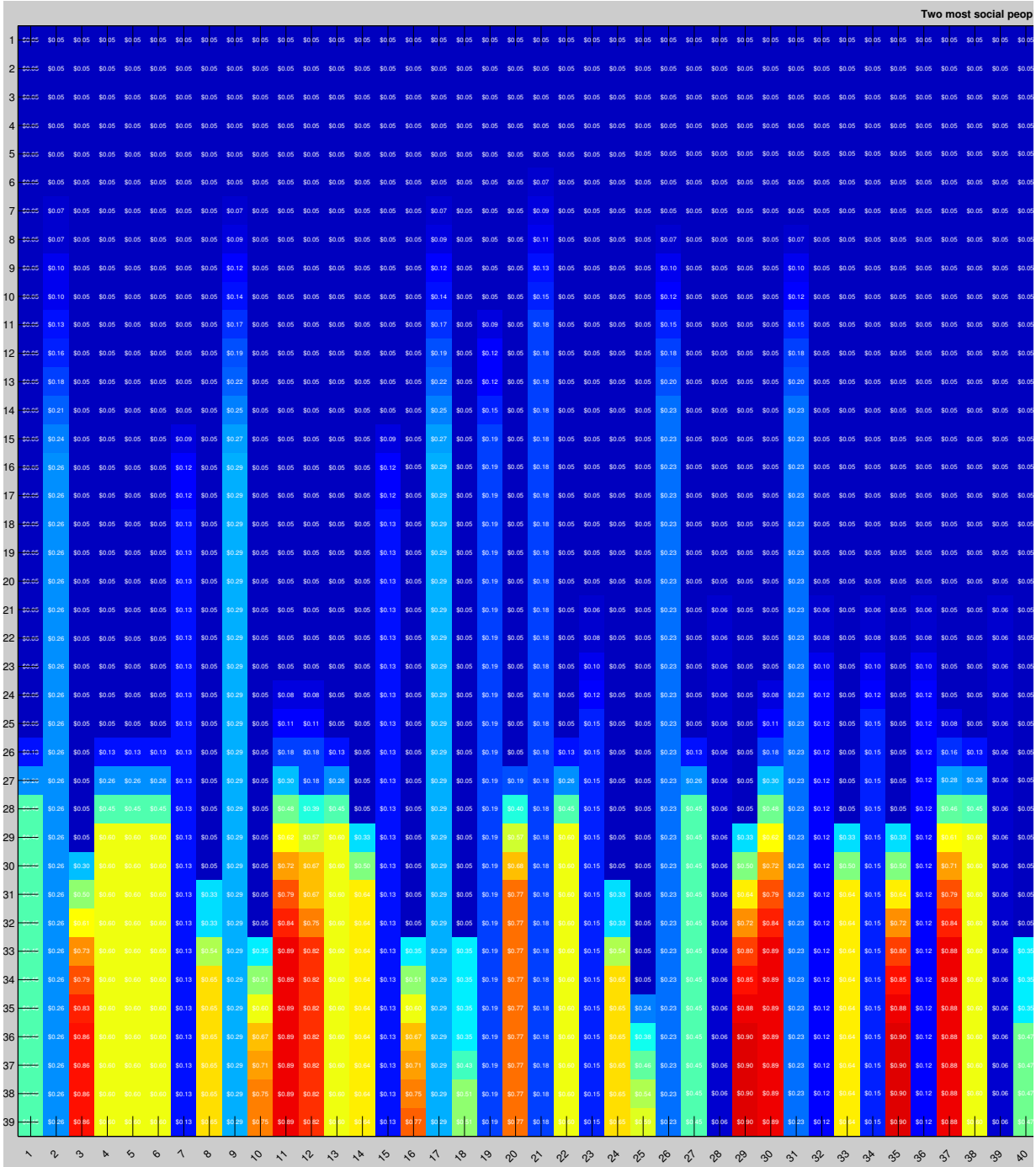


Figure 5.6: Heatmap of the likelihood of infection where the most social two individuals are positively infected: (b) individuals 41-80

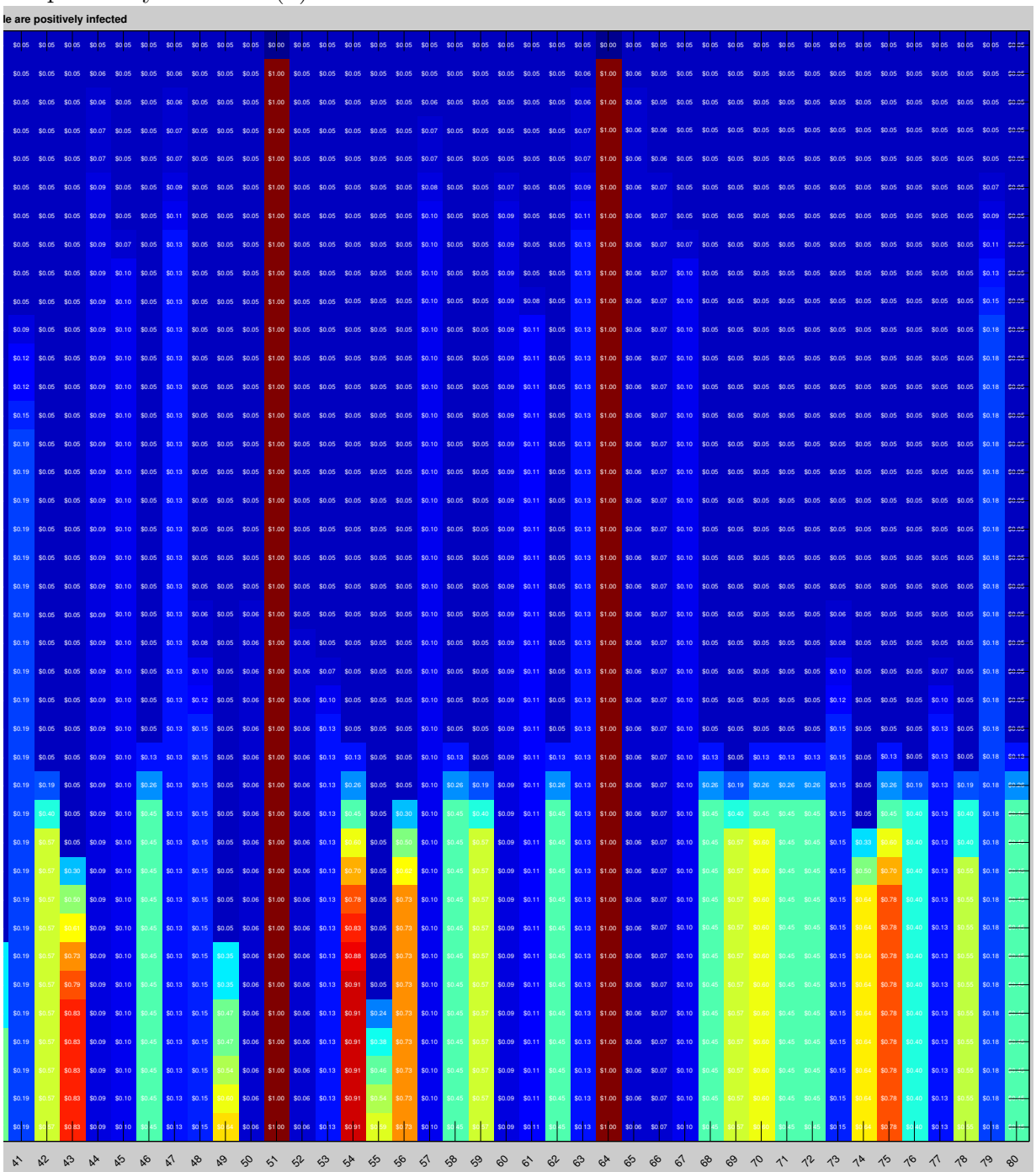


Figure 5.7: Heatmap of the likelihood of infection where the least social two individuals are positively infected: (a) individuals 1-40

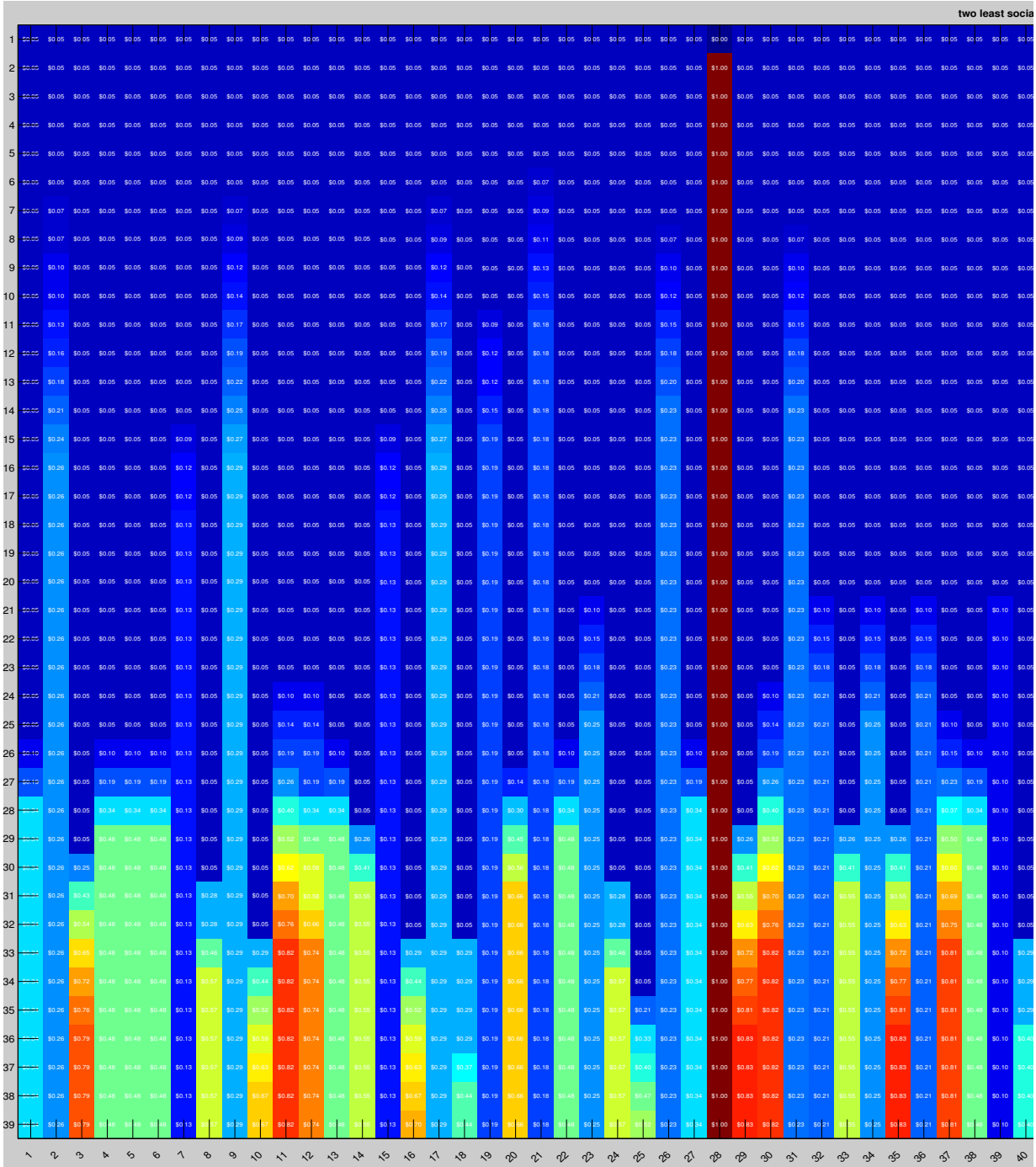


Figure 5.8: Heatmap of the likelihood of infection where the least social two individuals are positively infected: (b) individuals 41-80

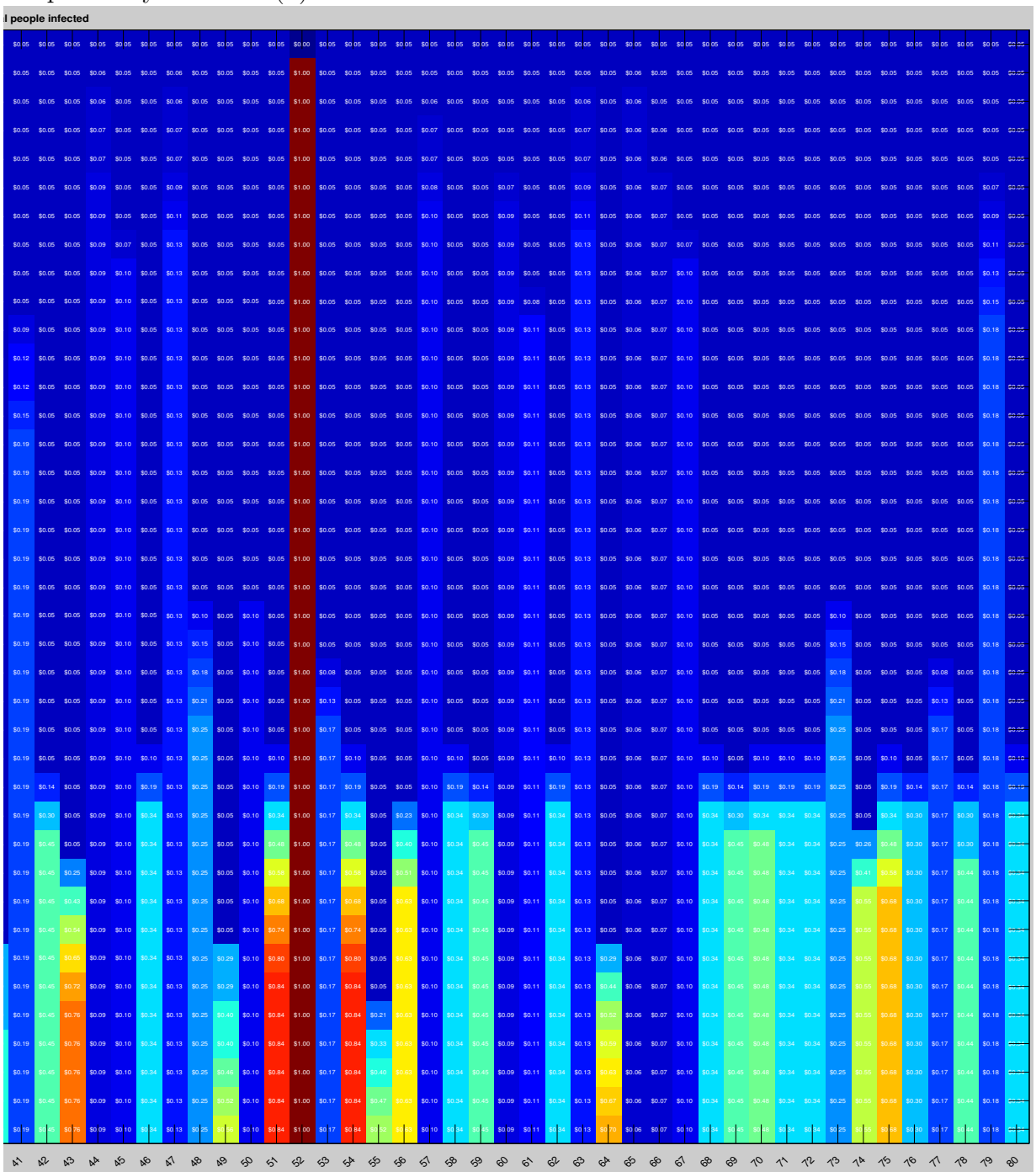


Figure 5.9: Heatmap of the likelihood of infection where the least social five individuals are positively infected: (a) individuals 1-40

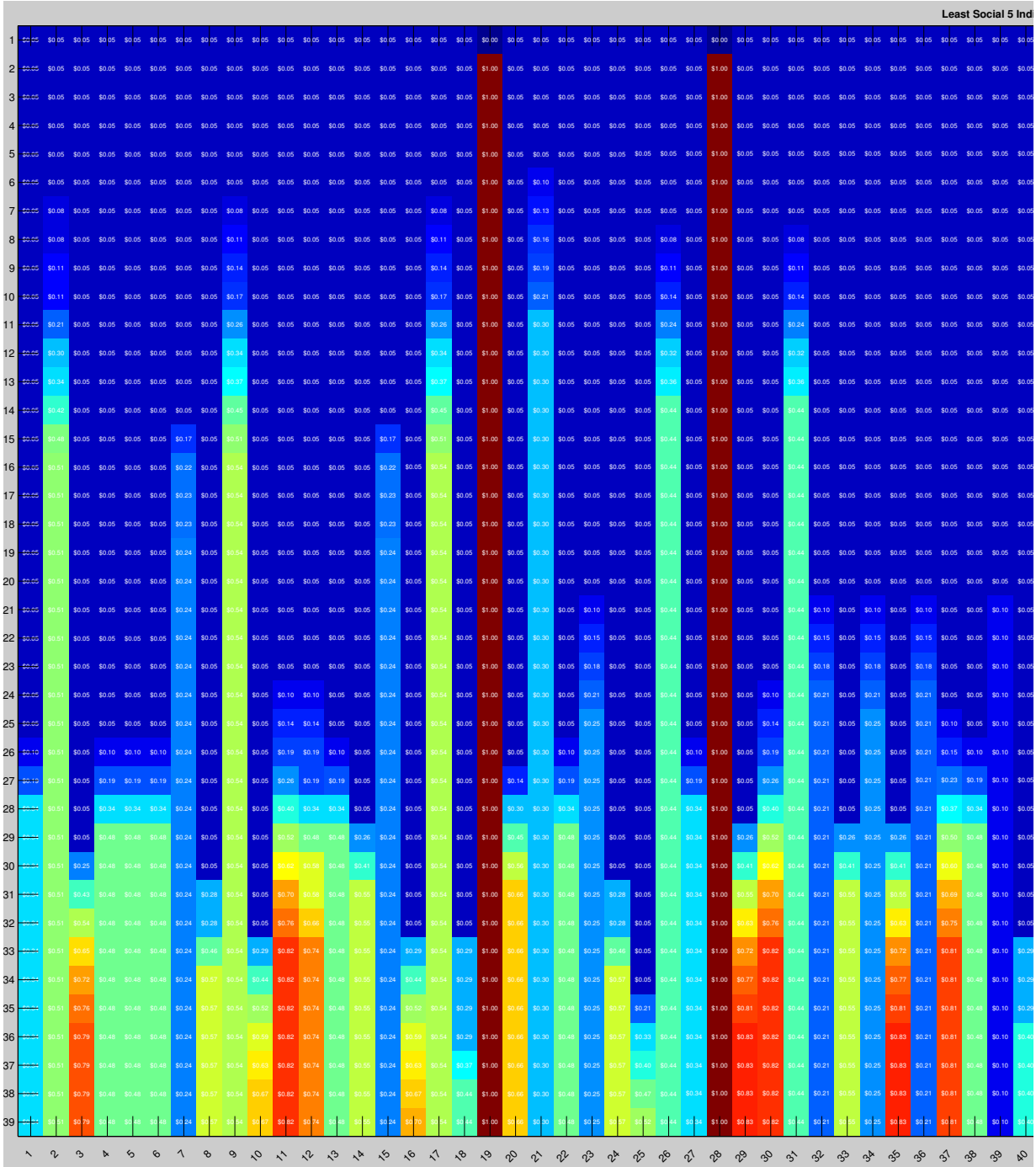


Figure 5.10: Heatmap of the likelihood of infection where the least social five individuals are positively infected: (b) individuals 41-80

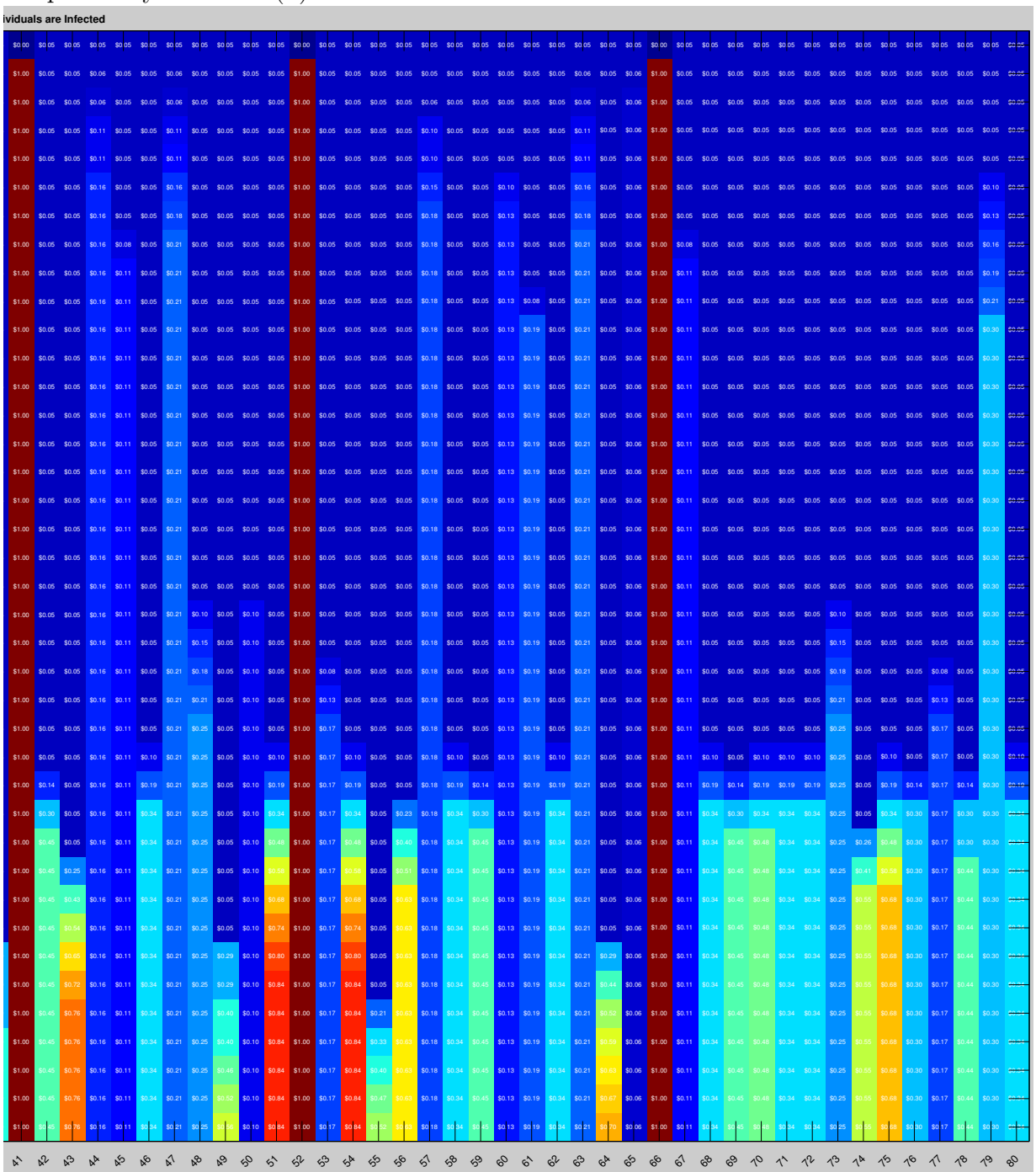


Figure 5.11: Heatmap of the likelihood of infection where the most social five individuals are positively infected: (a) individuals 1-40

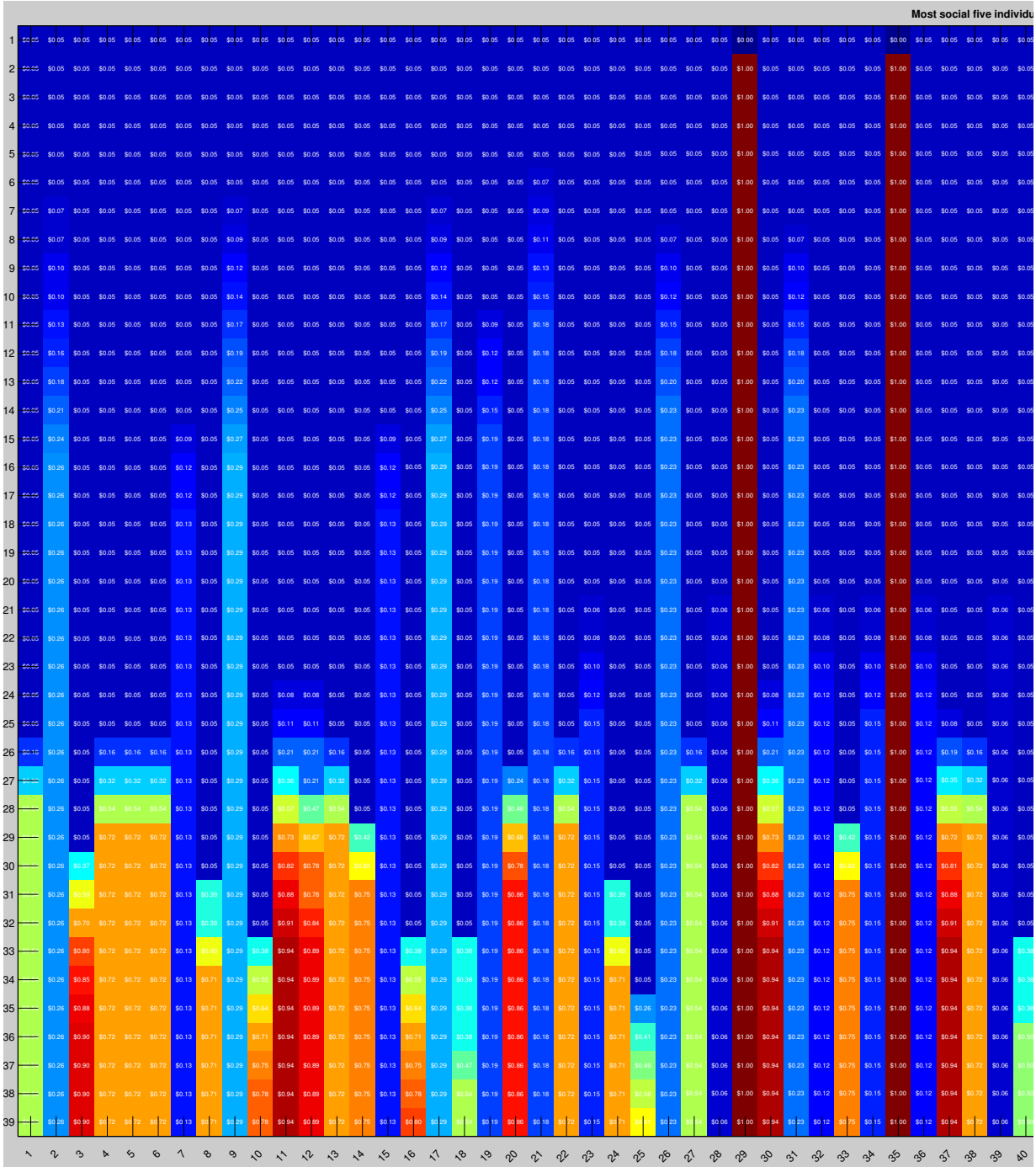


Figure 5.12: Heatmap of the likelihood of infection where the most social five individuals are positively infected: (b) individuals 41-80

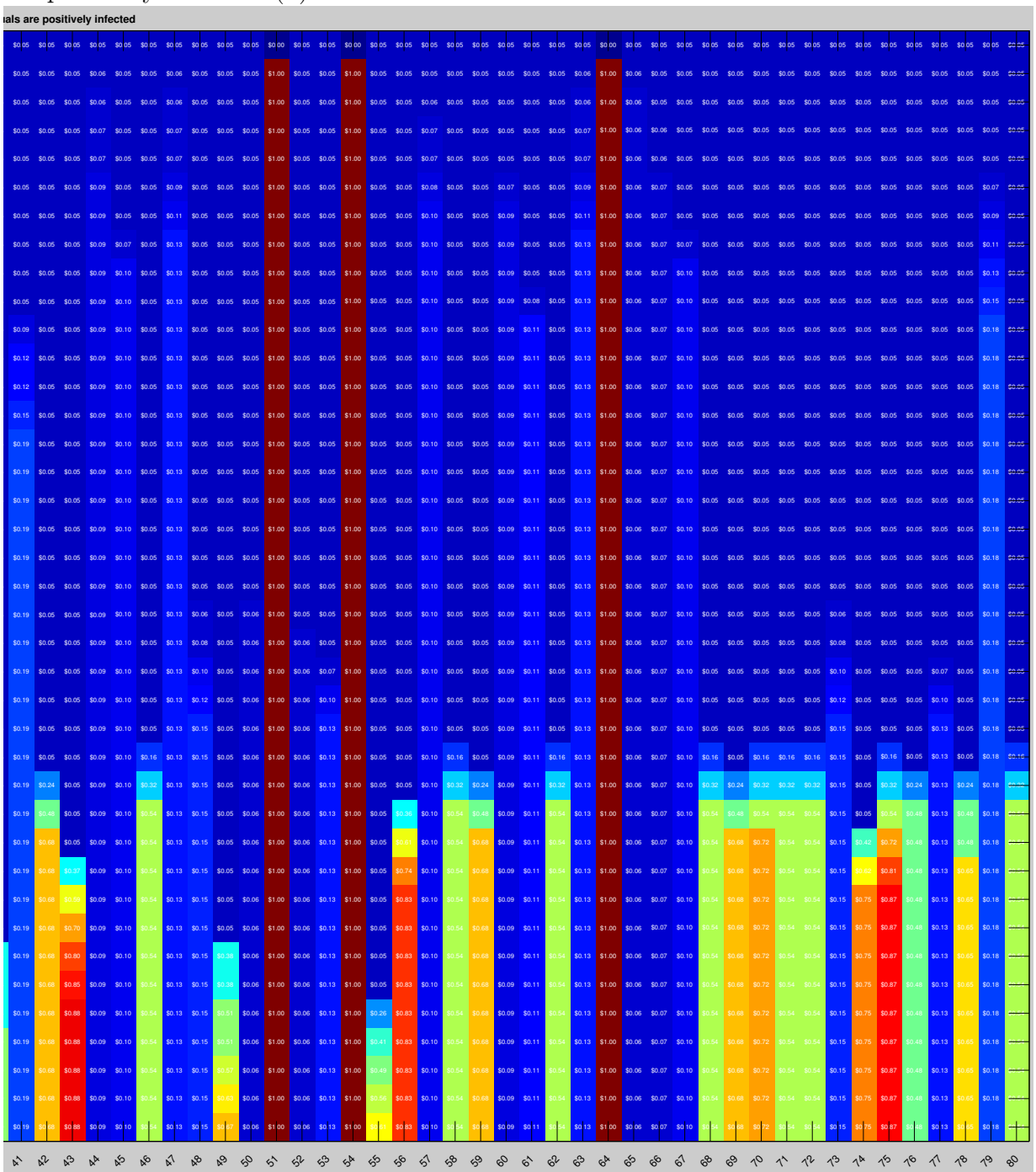


Figure 5.13: Heatmap of the likelihood of infection where the least social ten individuals are positively infected: (a) individuals 1-40

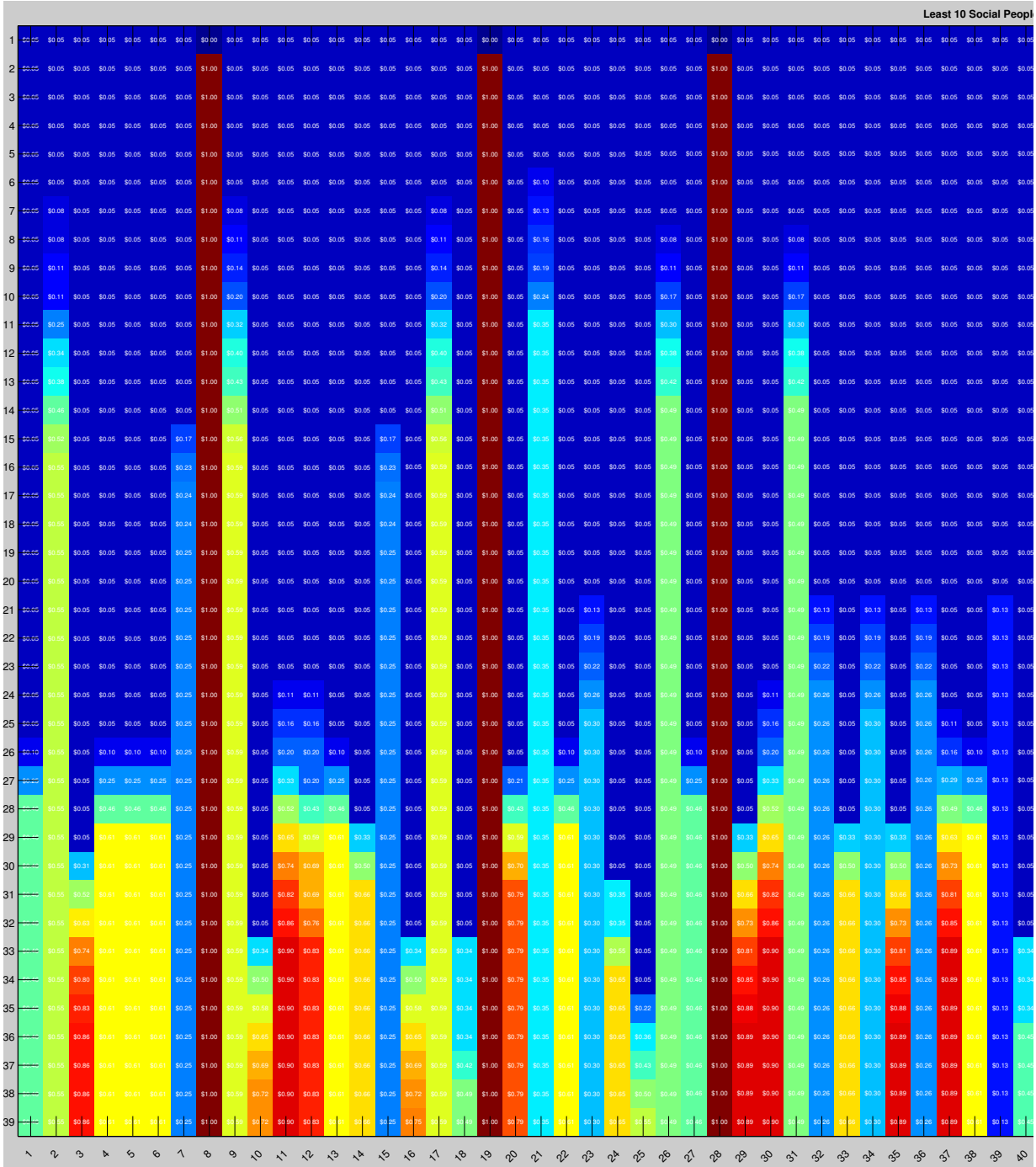


Figure 5.14: Heatmap of the likelihood of infection where the least social ten individuals are positively infected: (b) individuals 41-80

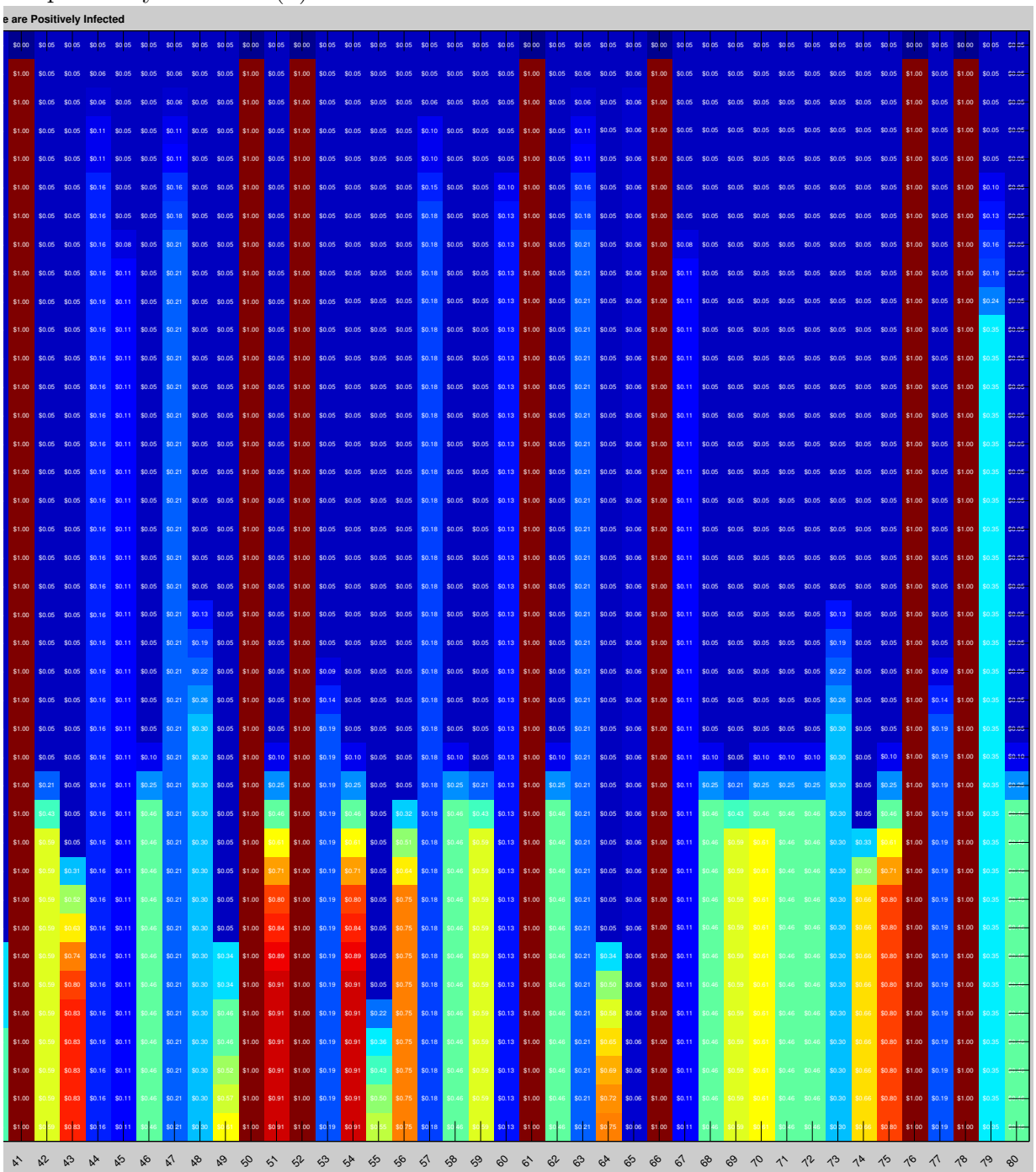


Figure 5.15: Heatmap of the likelihood of infection where the most social ten individuals are positively infected: (a) individuals 1-40

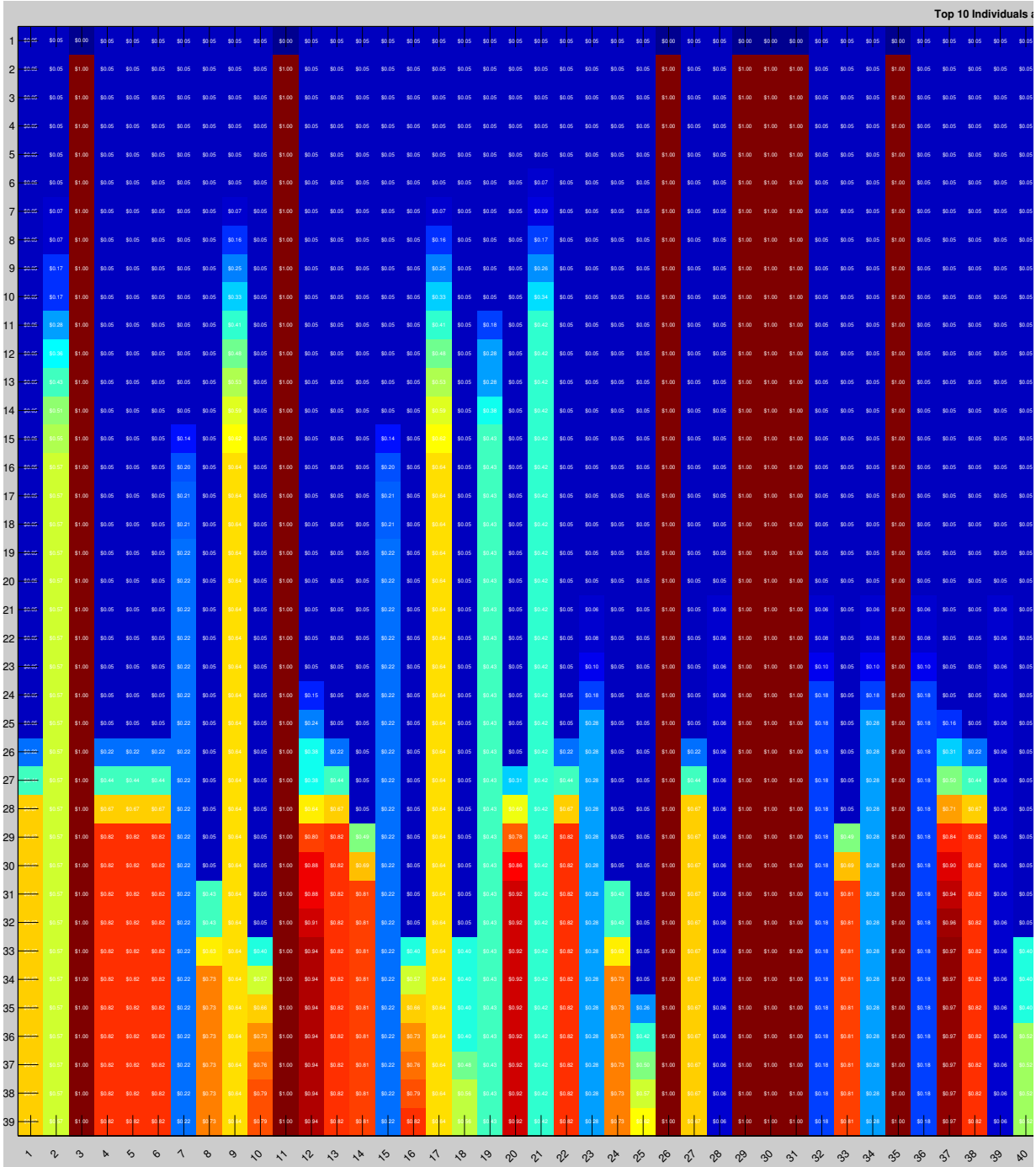
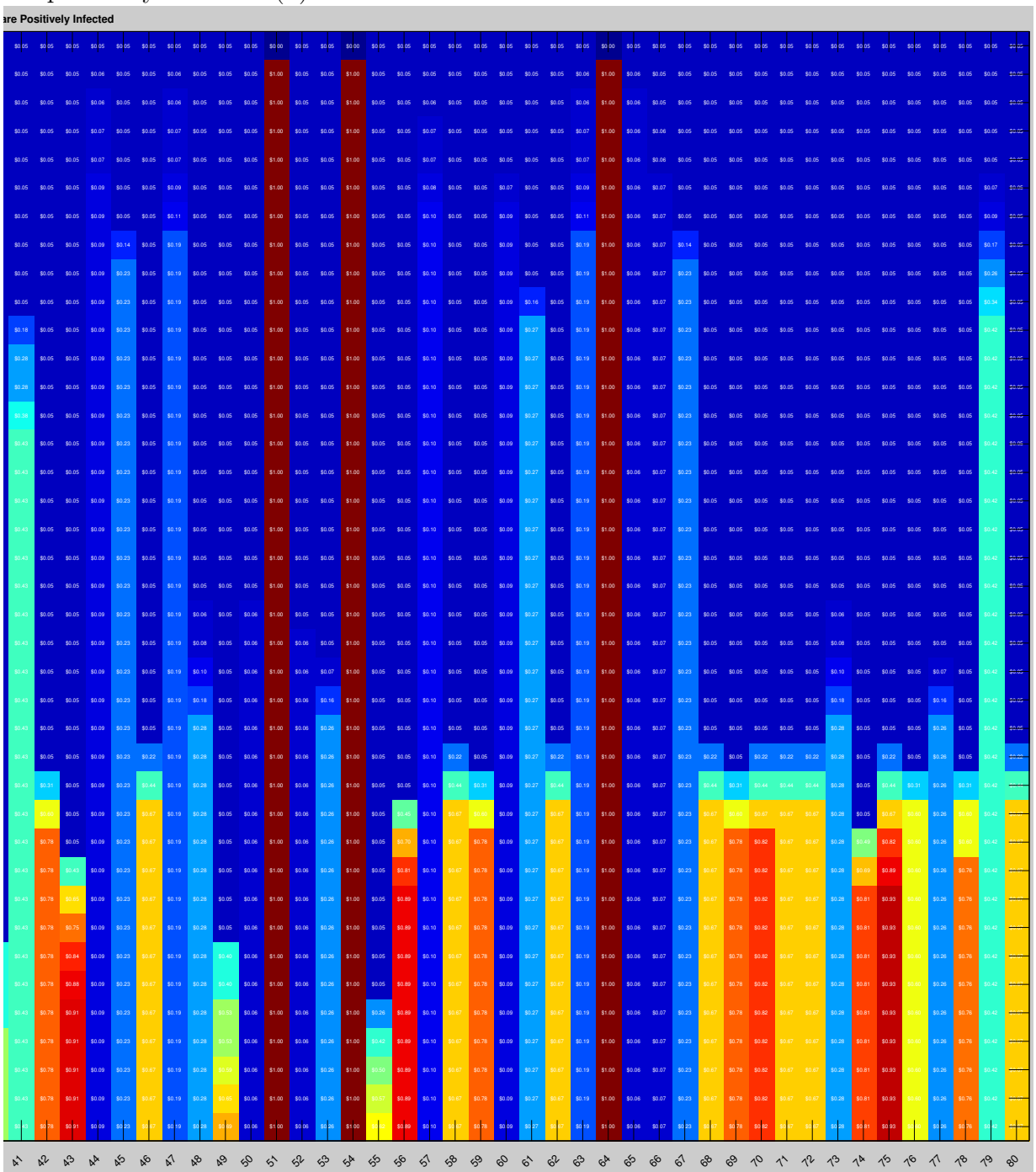


Figure 5.16: Heatmap of the likelihood of infection where the most social ten individuals are positively infected: (b) individuals 41-80



5.3 Model Validation

In the previous section, we presented the heatmaps of different types of outbreaks, where the infected individuals have different social ranks. In this section, in order to validate our model, we implemented a heuristic to measure the severity of the outbreak due to the infectious disease. In particular, we are concerned about:

- How fast the infectious disease will be spread in the population?
- What is the percentage of the population that will be likely to be infected at the last time step where the data was collected?

To investigate this, we will assume that if the likelihood of infection of a certain individual at a certain point of time is 0.6 or higher, this person is highly susceptible to infection and should be identified.

For the outbreaks we simulated at the previous section, we produced two parameters as shown in Table 5.3:

- The time step at which there will be an individual will have a probability of infection greater than or equal 0.6.
- The total number of individuals who will have probability of infection greater than or equal 0.6 at the last timestep.

As shown in Table 5.3, when we compare the effect of the outbreak caused by the most social person to the outbreak caused by the least 5 social individuals, we see that the *single* most social individual can possibly cause 7 more infections in the population and the first infection happened two time steps earlier than the outbreak caused by the least 5 social individuals.

Outbreak type	Time steps to the first infection	Total number of infections at the last time step
10 most social	15	39
10 least social	29	28
5 most social	29	25
5 least social	31	11
2 most social	29	19
2 least social	32	8
1 most social	29	18
1 least social	32	7

Table 5.2: Comparison of the effect of the different type of outbreaks

5.4 Backward Contact Tracing

In the previous section, we simulated few outbreaks where we set the health status of the infected individuals at the first time step and then we examined the virus propagation in the population. In this section we will set the health status of the infected individuals at the last time step and our goal is to identify the source of the infection. As shown in Figure 5.4, we simulated one outbreak where set the least ten social individuals to be positively infected at the last time step. The simulation results show that individuals 7 and/or 15 are most likely to be the seed of the virus.

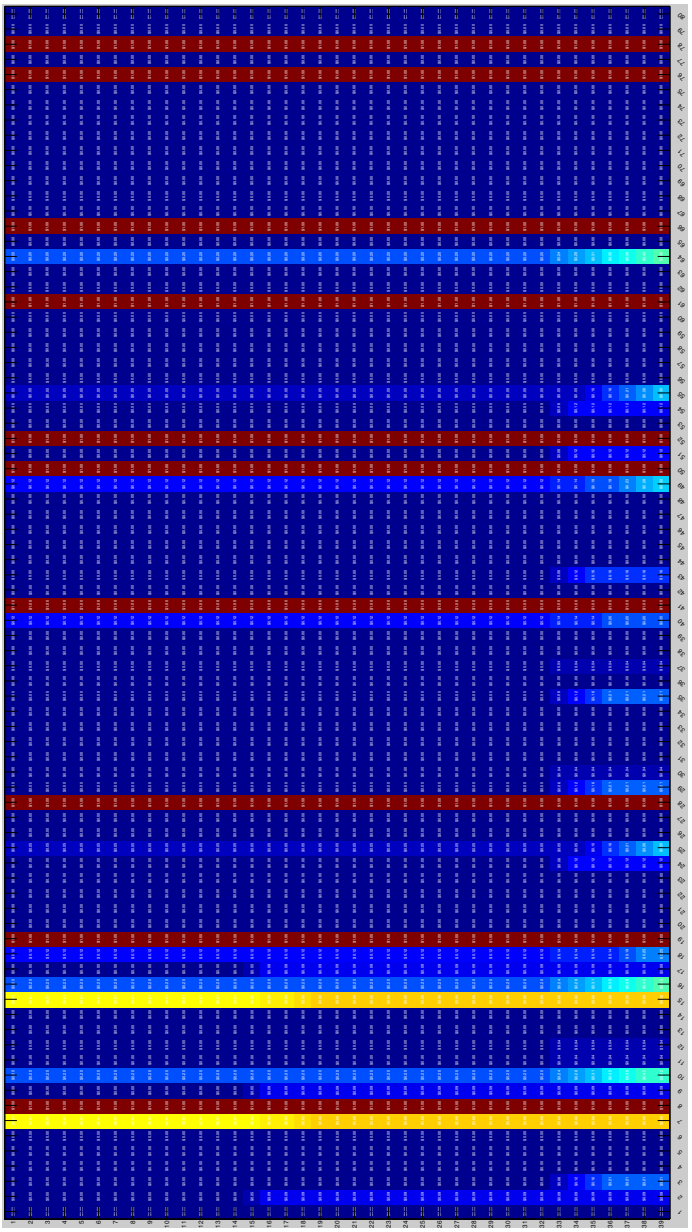


Figure 5.17: Backward Contact Tracing

Chapter 6

Conclusion and Future Work

In order to advocate the usage of context awareness data in public health applications, in this thesis we:

- Presented an approach that uses context awareness data-stream (e.g. proximity social contacts data) to track the spread of pandemics in the population.
- Proposed a Bayesian network model that allows us to perform contact-tracing effectively.

There are few possible extensions to our model.

Firstly, As we discussed in chapter 2, syndromic surveillance systems can be used as an alarm to the public health agencies to detect any possible anomalies in the monitored data-streams. There are many proposals for syndromic surveillance systems, that uses aggregated and un-aggregated data such as emergency department chief complaint, over the counter sales mediations, zip codes of patients, face-book posts, twitter posts, etc. To

the best of our knowledge, there are no proposal of a syndromic surveillance system and/or outbreak detection algorithm that uses context awareness data.

Although the model we built in Chapter 3 is for the contact tracing problem, we can append to it more data-streams such as ED chief triage for each individual, OTC medication sales, location of each individuals, etc. and use the appropriate CPDs for the model to answers questions such as: given all these data-streams in addition to the context-awareness data, what is the likelihood that there is an outbreak in this region at that time-step?

Secondly, In Chapters 3 and 4, we assumed the probabilities of CPDs and CPTs of the infectious disease. In order to use the proposed Bayesian models in real-world we should incorporate realistic probabilities of the CPTs and CPDs to be correlated to the mathematical models of the infectious disease(s) under investigation (e.g. basic reproduction number) [4]. For example, the transmission probability of SARS is different than the transmission probabilities of seasonal influenza.

Thirdly, we can incorporate the length of time two individuals interacted. That could be done be using probability of transmission that is dependent of the length of the interaction and that will definitely lead to more random variables in the model.

Finally, using the exact and approximate inference algorithms on large models can be challenging. As we have seen in Chapter 3, the complexity of exact inference in the models is exponential in the size of the model. There is a proposal that parallelize the junction tree algorithm and the clique implementation using Hadoop and map-reduce [1]. In our experiments we used off the shelf Matlab packages; I believe in order to scale the implementation of the inference algorithm, we should further investigate and implement the clique processing of the algorithms using Big Data packages such as Hadoop [29], Hive [8], Pig [37], Hbase [17], Apache Girpah [31], etc.

References

[1] Exact inference in bayesian networks using mapreduce. http://www.slideshare.net/ydn/5-exact-interfaceshadoopsummit2010. Hadoop Summit, 2010-09-30.

[2] Amen Ajroud, Mohamed Nazih Omri, Habib Youssef, and Salem Benferhat. Loopy belief propagation in bayesian networks: origin and possibilistic perspectives. *arXiv preprint arXiv:1206.0976*, 2012.

[3] Yaniv Altshuler, Nadav Aharony, Micky Fire, Yuval Elovici, and Alex Sandy Pentland. Incremental learning with accuracy prediction of social and individual properties from mobile-phone data. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 969–974. IEEE, 2012.

[4] Roy M Anderson, Robert M May, and B Anderson. *Infectious diseases of humans: dynamics and control*, volume 28. Wiley Online Library, 1992.

[5] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in ak-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[6] Alain Barrat, C Cattuto, V Colizza, F Gesualdo, L Isella, E Pandolfi, J-F Pinton, L Ravà, C Rizzo, M Romano, et al. Empirical temporal networks of face-to-face human interactions. *The European Physical Journal Special Topics*, 222(6):1295–1309, 2013.

[7] Alain Barrat and Ciro Cattuto. Temporal networks of face-to-face human interactions. In *Temporal Networks*, pages 191–216. Springer, 2013.

[8] Edward Capriolo, Dean Wampler, and Jason Rutherglen. *Programming Hive*. O'Reilly, 2012.

[9] Wendy W Chapman, John N Dowling, Michael M Wagner, et al. Classification of emergency department chief complaints into 7 syndromes: a retrospective analysis of 527,228 patients. *Annals of emergency medicine*, 46(5):445–455, 2005.

[10] WW Chapman, M Conway, JN Dowling, FC Tsui, Q Li, LM Christensen, H Harkema, T Sriburadej, and JU Espino. Challenges in adapting an natural language processing system for real-time surveillance. *Automatically tracking diabetes using information in physicians notes*, page 7, 2011.

[11] Jagan Dara, John N Dowling, Debbie Travers, Gregory F Cooper, and Wendy W Chapman. Evaluation of preprocessing techniques for chief complaint classification. *Journal of biomedical informatics*, 41(4):613–623, 2008.

[12] Nathan Eagle, Alex Sandy Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009.

[13] Ken TD Eames and Matt J Keeling. Contact tracing and disease control. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1533):2565–2571, 2003.

[14] Ken TD Eames, Cerian Webb, Kathrin Thomas, Josie Smith, Roland Salmon, and J Mark F Temple. Assessing the role of contact tracing in a suspected h7n2 influenza a outbreak in humans in wales. *BMC infectious diseases*, 10(1):141, 2010.

[15] MR FitzGerald, D Thirlby, and CA Bedford. The outcome of contact tracing for gonorrhoea in the united kingdom. *International journal of STD & AIDS*, 9(11):657–660, 1998.

[16] Geoffrey P Garnett and Roy M Anderson. Contact tracing and the estimation of sexual mixing patterns: the epidemiology of gonococcal infections. *Sexually transmitted diseases*, 20(4):181–191, 1993.

[17] Lars George. *HBase: the definitive guide*. O'Reilly Media, Inc., 2011.

[18] Tamir Hazan and Amnon Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. *arXiv preprint arXiv:1206.3262*, 2012.

[19] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011.

[20] Xia Jiang and Gregory F Cooper. A bayesian spatio-temporal method for disease outbreak detection. *Journal of the American Medical Informatics Association*, 17(4):462–471, 2010.

[21] Matt J Keeling and Ken TD Eames. Networks and epidemic models. *Journal of the Royal Society Interface*, 2(4):295–307, 2005.

[22] Matt J Keeling and Pejman Rohani. *Modeling infectious diseases in humans and animals*. Princeton University Press, 2011.

[23] KP Kleinman, AM Abrams, M Kulldorff, R Platt, et al. A model-adjusted space-time scan statistic with an application to syndromic surveillance. *Epidemiology and Infection*, 133(3):409–420, 2005.

[24] Alden S Klovdahl. Social networks and the spread of infectious diseases: the aids example. *Social science & medicine*, 21(11):1203–1216, 1985.

[25] Daphne Kollar and Nir Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[26] M Kulldorff, WF Athas, EJ Feurer, BA Miller, and CR Key. Evaluating cluster alarms: a space-time scan statistic and brain cancer in los alamos, new mexico. *American Journal of Public Health*, 88(9):1377–1380, 1998.

[27] Martin Kulldorff. A spatial scan statistic. *Communications in Statistics-Theory and methods*, 26(6):1481–1496, 1997.

[28] Martin Kulldorff. Prospective time periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 164(1):61–72, 2001.

[29] Chuck Lam. *Hadoop in action*. Manning Publications Co., 2010.

[30] Steffen L Lauritzen and David J Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 157–224, 1988.

[31] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010.

[32] Kevin Murphy et al. The bayes net toolbox for matlab. *Computing science and statistics*, 33(2):1024–1034, 2001.

[33] Kevin P Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[34] Daniel B Neill and Gregory F Cooper. A multivariate bayesian scan statistic for early event detection and characterization. *Machine learning*, 79(3):261–282, 2010.

[35] Daniel B Neill, Andrew W Moore, Maheshkumar Sabhnani, and Kenny Daniel. Detection of emerging space-time clusters. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 218–227. ACM, 2005.

[36] Juha-Matti Nikki et al. Bluetooth low energy. 2012.

[37] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1099–1110. ACM, 2008.

[38] Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. the MIT press, 2001.

[39] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1):273–302, 1996.

[40] Sujit K Sahu and Gareth O Roberts. On convergence of the em algorithmand the gibbs sampler. *Statistics and Computing*, 9(1):55–64, 1999.

[41] Gavin JD Smith, Dhanasekaran Vijaykrishna, Justin Bahl, Samantha J Lycett, Michael Worobey, Oliver G Pybus, Siu Kit Ma, Chung Lam Cheung, Jayna Raghwani, Samir Bhatt, et al. Origins and evolutionary genomics of the 2009 swine-origin h1n1 influenza a epidemic. *Nature*, 459(7250):1122–1125, 2009.

[42] Corien M Swaan, Rolf Appels, Mirjam EE Kretzschmar, and Jim E van Steenbergen. Timeliness of contact tracing among flight passengers for influenza a/h1n1 2009. *BMC Infectious Diseases*, 11(1):355, 2011.

[43] Nevin Lianwen Zhang and David Poole. Exploiting causal independence in bayesian network inference. *arXiv preprint cs/9612101*, 1996.