Mobile Sensing, Simulation and Machine-learning Techniques: Improving Observations in Public Health

A Thesis Submitted to the
College of Graduate and Postdoctoral Studies
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Aydin Teyhouee

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Or

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building
110 Science Place
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Entering an era where mobile phones equipped with numerous sensors have become an integral part of our lives and wearable devices such as activity trackers are very popular, studying and analyzing the data collected by these devices can give insights to the researchers and policy makers about the ongoing illnesses, outbreaks and public health in general. In this regard, new machine learning techniques can be utilized for population screening, informing centers of disease control and prevention of potential threats and outbreaks. Big data streams if not present, will limit investigating the feasibility of such new techniques in this domain. To overcome this shortcoming, simulation models even if grounded by small-size data can represent a simple platform of the more complicated systems and then be utilized as safe and still precise environments for generating synthetic ground truth big data. The objective of this thesis is to use an agent-based model (ABM) which depicts a city consisting of restaurants, consumers, and an inspector, to investigate the practicability of using smartphones data in the machine-learning component of Hidden Markov Model trained by synthetic ground-truth data generated by the ABM model to detect food-borne related outbreaks and inform the inspector about them. To this end, we also compared the results of such arrangement with traditional outbreak detection methods. We examine this method in different formations and scenarios. As another contribution, we analyzed smart phone data collected through a real world experiment where the participants were using an application Ethica Data on their phones named. This application as the first platform turning smartphones into micro research labs allows passive sensor monitoring and sending over context-dependent surveys. The collected data was later analyzed to get insights into the participants' food consumption patterns. Our results indicate that Hidden Markov Models supplied with smart phone data provide accurate systems for foodborne outbreak detection. The results also support the applicability of smart phone data to obtain information about foodborne diseases. The results also suggest that there are some limitations in using Hidden Markov Models to detect the exact source of outbreaks.

Acknowledgements

I would first like to thank my supervisor, Nathaniel D. Osgood, who has introduced me to the world of academics and inspired my future ambitions. The door to Prof. Osgood office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would like to thank my wife, Anahita Safari who has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point and my family, for their spiritual support throughout my life.

Additionally, I thank my lab-mates for their help during this study. Also, I thank all of my other friends in the University of Saskatchewan.

Contents

List of Tables

List of Figures

List of Abbreviations

ABM	Agent Based Model
HMM	Hidden Markov Model
SVM	Support Vector Machine
GIS	Geographic Information System
CDPH	Chicago Department of Public Health

Chapter 1

Introduction

Attempts to understand and describe disease outbreaks and their pattern of spread through computer simulation have become an important and challenging application of computational science in the past decades. Epidemiological models supply tools to analyze the outbreaks as manifestations of a complex system, and to examine various scenarios to prevent and improve response to disease outbreaks. Employing computer software provides the ability to study these "what if" scenarios in a low-risk, low-cost and rapidly executed simulation environment to support rapid learning. In addition, it provides the ability to perform a large number of simulated experiments in a timely fashion. These benefits have led to a high and increasing demand for simulation models. As a result, there are now dozens of software toolkits available for simulation, including, but not limited to, IThink, InsightMaker, Vensim, Repast, Anazlytica, NetLogo, Anylogic, etc. Within this list, Anylogic is notable for supporting mixtures of agent-based, discrete event, and system dynamics simulation methodologies.

Despite advances in food safety controls, foodborne illness (sometimes informally termed "food poisoning") imposes a considerable health burden, causing an estimated one in ten people to fall ill every year, and can be deadly specially in children less than 5 years old [33]. With foodborne outbreaks constituting a significant public health concern, the highest priority for most contemporary foodborne illness surveillance systems is to detect a new source of contaminated food quickly and efficiently. The medical and economic burden of an ongoing outbreak grows rapidly as it progresses, and hence a timely detection of the disease source is of critical importance.

This thesis describes the development and evaluation of an agent-based model simulating the occurrence of foodborne illness in a municipality and its combination with machine learning outbreak detection mechanisms. The model characterizes the relations among people (as food consumers), restaurants (as a major source of foodborne outbreaks) and an outbreak surveillance system within a geographic space using GIS elements. To make the simulation model empirically grounded, ground-truth data collected from clinical documents and publications were used [35, 8, 27]. To improve the accuracy of outbreak detection, the collected synthetic datasets of incident illness cases and vendor contamination records from the model were used to study the efficacy of performing disease outbreak detection using the machine learning approach of Hidden Markov Models (HMMs). Furthermore, we implemented variants of the predictive HMM model achieved from data training and applied it for syndromic surveillance monitoring, rapidly predicting the occurrence of ongoing

outbreaks.

1.1 Motivation

Each year, about 48 million people suffer from foodborne illness in the U.S., of which 128,000 are hospitalized, and 3,000 die from the infection [14]. A similar situation also prevails in Canada. The estimates by the Public Health Agency of Canada (PHAC) suggest that 4 million Canadians annually – or 1 in 8 – become sick from foodborne illnesses. Of these, there are about 11,600 hospitalizations and 238 deaths [45, 4]. An estimated 600 million - almost 1 in 10 people - fall ill after eating contaminated food and 420,000 die each year across the planet [33]. To prevent outbreaks of foodborne disease, local public health administrations routinely investigate restaurants, recording complaints, and responding accordingly [9]. While the public health inspection regime of food vendors successfully prevents many potential illnesses, the dynamic nature of restaurants' kitchens, the human resource constraints on carrying out consecutive inspections, and the time-consuming character of the inspection process allow violations to remain undetected and limit the completeness of foodborne illness prevention. Moreover, most afflicted people never show up at clinics and health care centers, but are greatly curtailed in their activity. Underreporting is one of the main factors that complicates effective surveillance of foodborne illness [29]. Although foodborne illnesses can be severe or even fatal, milder cases are often not detected through routine surveillance. While such cases impose stiff health, quality of life and economic costs, the absence of reliable data regarding this kind of illness in public health incidence records makes it very challenging to rapidly identify a potential outbreak occurrence.

An additional complicating factor reflects the fact that outbreak prediction methods rely heavily on telephone interviews of the clinical registered patients reporting possible foodborne illness, days or weeks after their illness. This makes the situation even worse in two ways. First, the patient will be subject to forgetfulness and spurious mentions regarding food vendors visited during a specified time. These issues makes it hard to effectively prioritize the inspection of possibly contaminated restaurants in an investigation. Second, and in consequence, because of inaccuracies and incompleteness of the data collected and the consequent prolongation of the investigation process, the adverse health and cost impacts of the outbreak will be magnified.

With the advent of mobile technologies, and web-based platforms such as Twitter, Facebook and various mobile-phone applications, a huge amount of data is being harvested from our daily life while those users interact with the surrounding world. Lately, the potential of collecting and translating online reviews and complaints about food vendors into useful information to improve foodborne illness surveillance has caught researchers' attention. In 2015, the Chicago Department of Public Health (CDPH) launched a project designed to improve food safety in Chicago by finding and responding to likely incidents of foodborne illness reported on Twitter [17]. In a similar approach, restaurant reviews from Yelp – a business directory service and crowd-sourced review forum – were inspected by the New York City Department of Health and Mental

Hygiene to find foodborne illness complaints [10].

We further investigated in our model how use of reporting of illnesses via smartphones by a population subgroup equipped with smartphones (referred as sentinels) could improve our inference regarding and response to potential outbreaks. To evaluate this, we investigated the impact of two data collection regimes. In the first – and more traditional – regime, we used clinical data only, reflecting presentation by a small subset of victims of possible foodborne illness to healthcare centers. The second regime supplements such traditional data with reports of illnesses provided by a small sentinel population, constituting just 4% of the total population. It bears noting that the 4% was chosen as the best result of Sara McPhee-Knowles's work [35, 8, 27] where she compares the different scenarios of having 1%, 2% and 4% of the population as sentinels to the baseline and it's impact on different metrics in her model. While this data collection regime could be carried out with a number of technologies – for example, via designated social media channels, call-in lines, and web-based mechanisms – we considered a case in which Ethica Data was utilized, as in the study above; as a result, some scenarios considered a situation in which information regarding participant location was considered as being potentially available for the sentinel group.

1.2 Solution

Having a foodborne illness outbreak detection mechanism for more accurate and timely triggering of outbreak control measures would offer notable public health dividends to foodborne disease outbreak surveillance systems. The classification can be a tool to support switching from a regime of regular restaurant inspections to a regime based more heavily on targeted inspections. Such a classification may further help to find contaminated restaurants more rapidly, so as to minimize the cumulative illness burden and limit economic losses.

The current inspection method most commonly alternates between restaurant checks (in a round-robin fashion) running in a condition absent a suspected outbreak, and an outbreak inspection regime (running in outbreak state declared when reported incident cases of foodborne illness within a period exceed a certain threshold), where recently visited restaurants reported by clinically ill individuals are prioritized for inspection. Our hypothesis was that the burden of foodborne disease could be lowered by supplementing traditional reporting data with information from sentinels who report their symptoms of foodborne illness via their phones, and (further) by designating an outbreak using a time-series based machine learning algorithm – such as in the form of a Hidden Markov Model (HMMs). Within this document, we sought to evaluate this hypothesis with our simulation model, using several different HMMs:

1. A binary HMM for classification as to whether we are facing an outbreak situation: Here, we do not take into account probability of a given restaurant being in contaminated state, and only concentrate on identification of an overall outbreak state. Recognition of such an outbreak is treated as triggering

a comprehensive inspection citywide.

(a) We initially focus on the historical number of traditional reports of presenting individuals to feed our HMM and train it. This trained model later will be used to judge the existence of any contaminated restaurant on the basis of the number of newly reported presenting clinical cases of illness.

(b) In a second step, we use both traditional clinical reports and sentinel self-reported illness counts as our HMM input.

(c) And finally, the HMM outbreak classifier developed in 1b is placed into the simulation model. The model is then used to evaluate the cumulative cases of infection and cumulative contamination period (which are hypothesized to shrink).

2. Within this element of the work, we consider a more articulated HMM for classification as to whether specific restaurants are in contaminated situation. To address that need, we built an HMM with a single global "no-contaminated" state and distinct "contaminated" states for each restaurant. To avoid a combinatorial explosion, we do not have separate states involving more than one restaurant being in contamination, relying on the assumption of a small probability that two restaurants will be simultaneously contaminated, because of the rarity and short duration of contamination. Within the current sub-investigation, to understand the relative importance of different types of information, we will assume that the HMM has no access to highly accurate location information, in the form of either restaurant-specific traffic or an accurate record where a given person has gone – i.e., no harvested location data is available via tools such as Ethica Data to give either such traffic or to support better recollection.

(a) Like 1a, this investigation takes into account only traditional reporting emerging from presentation to a clinical setting, and relies upon each afflicted individual's personal memory of locations in which they might have been poisoned.

(b) In this step, we consider both complaints received through traditional reporting systems from presenting clinical cases, as well as reports of symptoms received directly from the user. As above, only afflicted individuals' personal memory of their visited restaurant locations will be assessed.

(c) Finally, the HMM contamination classifier created in step 2b will be placed into the simulation model as an outbreak triggering mechanism. We then seek to investigate how much use of that classifier changes the cumulative cases of infection, compared with the naive classifier noted in the baseline 1c. We use this HMM in two ways:

i. First, we investigate the impact of using the ABM only to determine if an outbreak occurs across the restaurants as a whole.

ii. Next, we consider an HMM that operates both to determine any occurrence of restaurant contamination and to prioritize visitation of restaurants in such instances, so as to help the surveillance system rapidly identify them based on the probability that this particular restaurant is in the "contaminated" state, as deduced from the HMM.

3. The HMM for this subproject is identical to the one described in 2 above, but in contrast we consider the effects of highly accurate visitation counts for each restaurant .

1.3 Contributions

The main contributions of this thesis to the literature are the following:

1. Evaluation of new technologies in harvesting foodborne illness data and using them as an information stream employable in the inference structure of syndromic surveillance systems. Broad-brush picture of such applications is already given in a work by McPhee-Knowles et al. [28]. Results of this thesis achieved by running a project on food consumption behaviour of 96 university students using a smartphone-based epidemiological data collection system called Ethica Data [1] revealed that use of smartphones that can record locations and offer channels for reporting foodborne illness cases where the individual does not seek medical help could improve our inference about the potential outbreaks.

2. Simulation-based evaluation of several HMMs as outbreak identification systems. We found that such a system offers an excellent potential for detecting foodborne illness outbreak when informed by reporting by even a very small (e.g., 4% of population) sentinel group.

3. Utilization of HMMs in a goal-oriented scheme to identify the source of outbreaks and reduce the duration of an outbreak, resulting in fewer incident cases of illness. While the evaluation of such schemes did not reveal big gains, the creation of such models suggests opportunities for improvements of the models – such as with improved data – that could strengthen results.

1.4 Thesis Outline

In this section, we offer an overview of the balance of the thesis. The current chapter, Chapter 1, described the main motivation for this work, and problems that should be addressed. Furthermore, lines of investigation and main contributions of this thesis were explained. Notes on the remaining chapters are given below:

- Chapter 2 presents background information on topics important for understanding the following chapters of this thesis. This includes description of the simulation mode, an overview of similar work in using data streams via new technologies and descriptions of foundational technologies and formalisms.

- Chapter 3 is designated to cover the problem mentioned in 1.2, which has been published the proceedings of the Conference on Social Computing, Behavioral-Cultural Modeling & Prediction and Behavior Representation in Modeling and Simulation 2017.

- Chapter 4 addresses the problems 2, 3 raised in the Section 1.2.

- Chapter 5 describes the 2016 data collection project run using the smartphone-based Ethica Data system, with a focus on food eating habits (including time and location of eating), and foodborne illness reports collected from a group of students recruited for this purpose. In this chapter, we used some tools and methods to study the harvested data and reveal some interesting results.

- Chapter 6 reports the work described in the paper *Cough Detection Using Hidden Markov Models* presented at the 2019 International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. Although the focus of the paper is not in the domain of foodborne illness, it deals with temporally explicit syndromic surveillance technologies supported by big data and shows how – contrary to the chapter 4 where a bottom-up application of HMMs brings up some limitations due to lack of profile distinction among classes (states) – here we are effectively able to group the identified hidden states of a cough sound into binary groups of cough/non-cough and coughing/non-coughing.

- Chapter 7 provides a summary of the thesis. It notes important limitations of the investigations conducted, and directions for future work that can improve or build on the results of this thesis.

1.5 Publications

- Chapter 3 includes a manuscript entitled "Prospective Detection of Foodborne Illness Outbreaks Using Machine Learning Approaches" by Aydin Teyhouee (AT), Sara McPhee-Knowles (SMK), Cheryl Waldner (CW) and Nathaniel D. Osgood (NDO), published in Proceedings of the 2017 International Conference on Social Computing, Behavioral-Cultural Modeling & Prediction and Behavior Representation in Modeling and Simulation [44]. Authors' contributions are as follows:

  AT drafted the manuscript, contributed in redesigning the simulation model, obtaining synthetic empirical data from the model, implementing the HMM component, and data analysis. NDO supervised the study, provided the skeletal HMM structure, supervised in adapting it to the foodborne illness context and redesigning the simulation model and modified the manuscript. SMK and CW contributed to development of the initial simulation model. CW further advised on some elements of the modeling work and its description.

- Chapter 6 includes a manuscript entitled "Cough Detection Using Hidden Markov Models" by AT and NDO, accepted in Proceedings of the 2019 International Conference on Social Computing, Behavioral-

Cultural Modeling & Prediction and Behavior Representation in Modeling and Simulation [43]. Authors' contributions are as follows:

AT drafted the manuscript, contributed in data preparation, data labeling, implementing the HMM component. NDO supervised the study, provided the skeletal HMM structure, supervised in adapting HMM to cough detection context and modified the manuscript. AT and NDO contributed in obtaining empirical data.

Chapter 2

Background

This chapter primarily focuses on a review of the models which have applied machine learning techniques for detection of disease outbreaks. It also provides a brief introduction to *AnyLogic®*, the tool we used for creating our model and performing all the scenarios, and also agent-based modelling. A brief section is also allocated to Ethica Data [1], a smartphone-based epidemiological data collection system that was utilised in data acquisition for our foodborne study discussed in Chapter 5. The chapter further offers an overview of the machine learning algorithms that we have employed. We additionally provide a background review on Cassandra, a highly scalable distributed database and also Apache Spark, an open source big data processing framework. Both Cassandra and Apache Spark are used for analyzing and investigating foodborne disease related data discussed in detail in Chapter 5.

Section 2.1 covers foundational background on foodborne illness. Sections 2.4 and 2.5 provide an overview of Hidden Markov Modeling and its great performance when dealing with sequential data, and Support Vector Machine classifiers, respectively. These two techniques are used as a means of comparison in outbreak detection, and their performance is discussed in Chapter 3. The Cassandra and Apache spark are covered in Sections 2.6 and 2.7 respectively.

2.1 Literature Review

Within recent years, prospective detection of disease outbreaks in general using machine learning approaches has attracted the attention of researchers [49], [24], [11]. For example, Xia Jiang and Garrick L. Wallstrom in [49] present a Bayesian Network model which not only predicts an outbreak, but can further estimate its size, duration and time of occurrence. The challenge in this new field is to diagnose occurrence of an impending outbreak in a timely enough fashion to aid policy makers and public health agents in taking quick outbreak controlling measurements. However, the applications of such work to foodborne illness outbreaks have thus far been very limited [19].

Machine learning provides a set of tools and methods which can be applied in different problem domains for data analysis. Given that the challenge of detecting foodborne illness outbreaks consists of identifying the evolution of the categorical latent state (outbreak vs. non-outbreak) of a system (municipality) over time in the light of noisy observations (incident cases) strongly influenced by the state, Hidden Markov Models

(HMMs) offer a particularly attractive analysis lens. Here we seek to distinguish between outbreak and non-outbreak states based on our observation of the number of reported illnesses. In this work, we compare the findings of the HMM with the results of a Support Vector Machine (SVM) model, which fails to take into account the temporal context of data. To achieve this end, we will employ synthetic ground truth data from a previously contributed [36] empirically-grounded agent-based model (ABM) of foodborne illness.

2.2 Agent-based Modeling

This section introduces Agent-based modeling as a common tool in modeling complicated systems and how this approach has been utilized to create the foodborne-illness model presented in this thesis.

2.2.1 Anylogic

We used *AnyLogic®* in our project. It is equipped with a variety of tools to build projects using System Dynamics, Agent Based and Discrete Event modeling concepts. It further provides the flexibility to build up a model with combinations of those three mentioned schemas. This is the case in hybrid models where, for example we wish to model the agent behavior continuous in time we use System Dynamics fragments (i.e., systems of equations) inside the agent; however most frequently we use statecharts as a clear and intuitive way of representing human decision logic.

AnyLogic uses a graphical interface for declaratively characterizing many aspects of a model, but is also based on – and compiles models to – *Java*, which makes it possible for a modeller to extend the declarative characterization of simulation models with coding in *Java*. In fact, *Anylogic* extends the **Eclipse IDE Platform**. The *Java* base of AnyLogic provides a high degree of flexibility in extending the simulation models and also in building *Java* applets executable by any standard web browser, subject to security settings.

Models can benefit from maps as layouts. AnyLogic supports both GIS shapefile maps and tile maps from free online providers such as OpenCycleMap, LandMap, and OpenStreetMap. Using the tile maps, geospatial mode-specific routes for agents are derived by AnyLogic from the map data.

2.2.2 Agent-based Modelling Overview

In highly dynamic and complex ecosystems (hospitals, workplaces, cities, etc) disease outbreaks such as foodborne outbreaks depend on a number of individual characteristics of the food vendors and consumers and network of contacts. These constitute external impacts that may be best picked up within the agent based modeling tradition, given the fact that surveillance and outbreak investigation for foodborne disease takes place at an individual level.

To map a system from the realm of real world to the models domain, ABM takes an approach working upwards from the level of individual characteristics. In agent-based modeling methodology, properties and

actions of individual agents of the system are taken into account and the overall behaviour of the model appears as these agents integrate and interact with each other. This is in contrast to the most traditional type of health modeling in System Dynamics and compartmental methodologies, where an aggregate view is taken on the system. For example, Tian and Osgood [47] compare these two approaches in the context of Tuberculosis (TB) transmission, considering smoking as a risk factor. Their results suggest that at the practical application level, greater accuracy and easier extension are what agent based model was offering. They also show a significant difference between agent based modeling and system dynamics when the impact of network structure on TB diffusion was studied, giving more insights into the difference between them in the context of practical decision-making in healthcare [47].

When the system is so complicated that it is almost impossible to understand how it behaves, the ABM approach helps the modeler individualize the key role players in the system (agents) and define their behaviors. Then these building blocks are connected to each other or are put in an environment to communicate and the global behaviour of the modeled system emerges from these interactions. As a result, ABMs can support learning that leads to deeper understanding of the dynamics of complex systems. By examining the emergent patterns arising from representing and running counter-factual "what if" strategies in the model, policy makers can arrive at improved understanding of the tradeoff between policy options.

Each agent has its own variables, parameters and behaviors; an individual for instance who is female (i.e., has a sex parameter), gets sick (as a natural behaviour), and gets older as the model runs (i.e., an age variable). There may be a network of contacts between agents which is used to model the exchange of relevant information. There also can be an environment affecting the agents and being affected by them.

A central part of model development lies in the characterization of the agents playing key roles in the system with a requisite level of details. Technically, an agent has state, parameters and behaviour. By parameters we mean values mostly constant during a simulation run, such as gender or martial status and by variables – changing values, such as current age, or health level. Behaviour not necessarily but often involves decision making logic, which is typically triggered by certain events and conditions. Random factors play an important role in many AB models, reflecting the fact that – among other factors – most decisions are probabilistic. Factors influencing the agent behavior can be external and internal. The external ones typically are same for all agents and are originated by the environment, by service providers. Internal factors are consumer individual preferences and needs, knowledge, history, etc.

Figure 2.1 shows a demo model with three agents: *Agent1*, *Home* and *Factory*. *Agent1* has three parameters which might hold characteristic values about its being, two variables holding varying values while the model is running. Two functions describe the behavior of the agent when triggered by an internal event like *event1* or an external one like a specific global value from the main environment. This agent owns a statechart. Statecharts materialize an agent's interaction logic. The second agent, the *Home* has a very basic structure having a location in terms of two horizontally and vertically pixel-based distances from the origin of the pane sitting in the main class or (if a GIS map is presented) two geographical values as the latitude

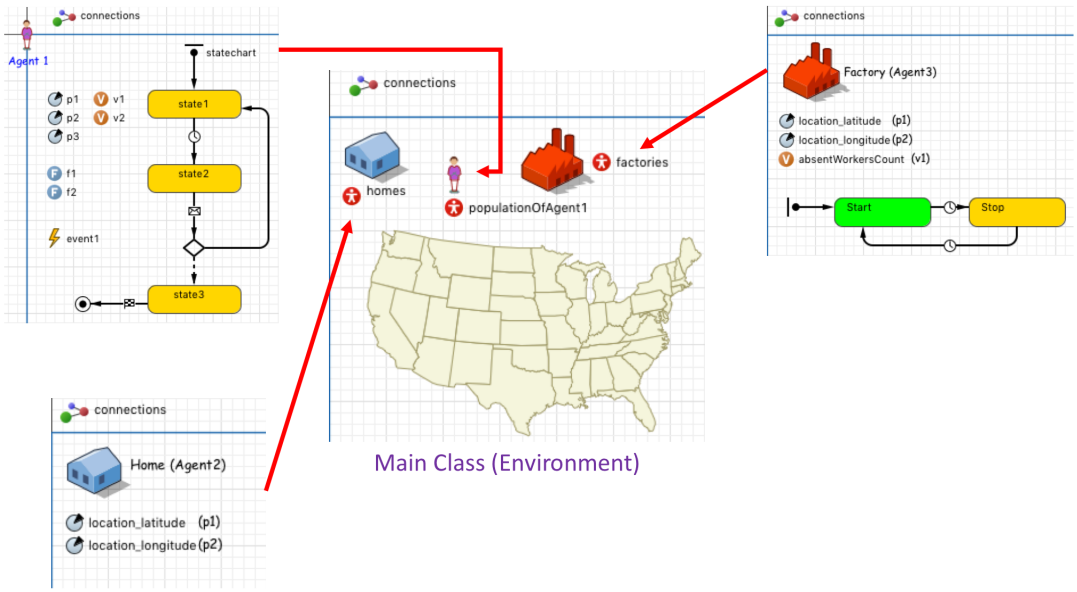


Figure 2.1: An ABM representing three agents. Normally a population of agents live inside the main class which is the environment where all the agents interact with each other.

and longitude of the location. Like the *Home* agent, the *Factory* agent is holding a simple logic in terms of a state chart and two parameters and a variable.

The *main* class, our model's environment is where the population of the agents live; in other words, the Main class holds populations of one or more instances of other classes such as *Home*, and *Factory*. Here, we might assign (randomly or with a logic) a group of 2 or 3 Person agents to a single Home agent. There might be also some underlying networks among these agents living in a home, such as a parent-ship network, children network, husband-wife network or among neighbour homes as neighbourhood network and so on. A class diagram of this model is shown in Figure 2.2, using the Unified Modeling Language (**UML**).

2.3 Ethica

Developed by Ethica Data, Inc., this system originally emerged from the iEpi project at the University of Saskatchewan that in 2009 was used to track the spread of the H1N1 virus in central Canada by means of a mobile sensor system, and subsequently developed to support smartphones. Ethica is the first of its kind in turning smartphones into small-scale research labs. Researchers can use this application to design their studies without needing to have programming skills. Although missing at the time of using this application for our food behavior and foodborne illness research project, eligibility screening, informed consent, and enrolment now can all be performed through the phone and thus requiring physical meetings with participants becomes unnecessary, With Ethica a researchers can evaluate symptoms, and behaviors of the participants through sensors on smartphones/wearables and surveys. They can either get advantage of the web-based access to real-time analytics or have their own analysis over the collected data by accessing database directly or through

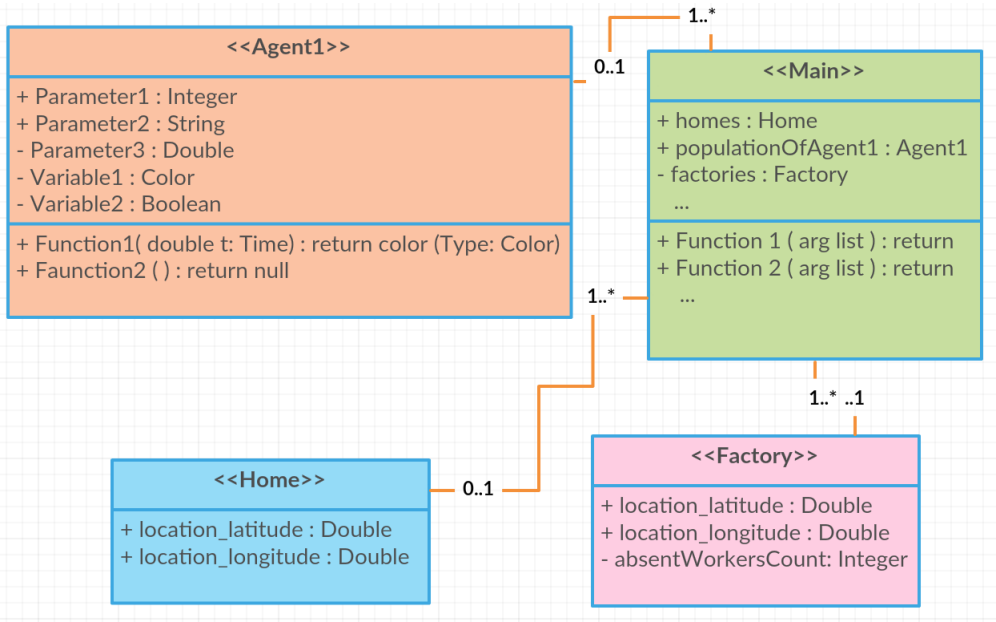


Figure 2.2: A class diagram representing the structure of a model in *Anylogic* which uses *java* programming language behind the scenes.

other frameworks and applications such as Spark, R, or Tableau. Due to end-to-end encryption, from app data capture up until back-end storage, it is well equipped to be utilised in the public health domain where researchers are dealing with sensitive data.

Ethica collects data from most of the data sources without a user's interaction. After participants' enrollment in an Ethica study, given that the study designed by the researchers group requires automated data collection from any of the sensor sources on the phone, and subject to continued approval being granted by the users, their phones' sensors data are captured by Ethica in a fine-grained fashion. After being encrypted, such data is uploaded to a table under the name of that specific sensor located in a database designed for the study and sitting on a secure server. Apart from a wide range of capabilities in collecting sensor data, Ethica also offers great flexibility in designing customized surveys. User-initiated surveys and time-based ones are two of the common survey types, and were used in our study. Similar to the sensor data, responses to the surveys are uploaded into their corresponding table. These responses are saved with a record time specifying the time they are filled out after being popped up on the phone or called up by the participant. This record time for expired and actively canceled surveys is being registered with the values of "-1" and "-2" respectively.

2.4 Hidden Markov Models

Hidden Markov Models (or HMMs) are widely used in speech detection, pattern recognition and classification problems. Given a time horizon, they infer the evolution of the system among a finite set of latent and

non-observable categorical states over that horizon, with each of these states being associated with a specific distribution of observables; it bears emphasis that, as applied here, this procedure is not inferring the structure of the model but instead the value of the parameters governing the evolution of the system [15].

A hidden Markov Model algorithm is trying to model a system producing a sequence of typically noisy external events (observations) generated from its finite and countable internal states, where the internal state changes are hidden to a viewer outside the system, and the current state is always dependent on the immediately previous state only, satisfying the Markov property. This property guarantees that the probability distribution of the immediately subsequent state of the system depends only upon the current state and not past states; in other words, the current state encloses all of the history of the system up to present, enough to probabilistically dictate the future state of the system. The distribution of observables is further treated as depending only on the current state (and are thus treated as independent, conditional on on remaining in that state).

To briefly provide a more structured characterization of HMMs, consider a random variable o. Denote the value of o at time t as o_t that as indicating observations at discrete time chunks. As mentioned before, the observation at time t (that is, o_t) is dependent on an evolving, non-observable ("hidden") and categorical discrete state s_t. Moreover, there is an underlying (and hidden) process governing changes in state: the probability distribution for the current state s_t depends only on the value of the state of the system at $t-1$, s_{t-1}, i.e., $P(s_t|s_{t-1}) = P(s_t|s_{t-1}, ..., s_1)$. Also, given s_t, o_t is independent of any other state and observation, i.e., $P(o_t|s_t) = P(o_t|s_t, o_{t-1}, s_{t-1}, ..., o_1, s_1)$.

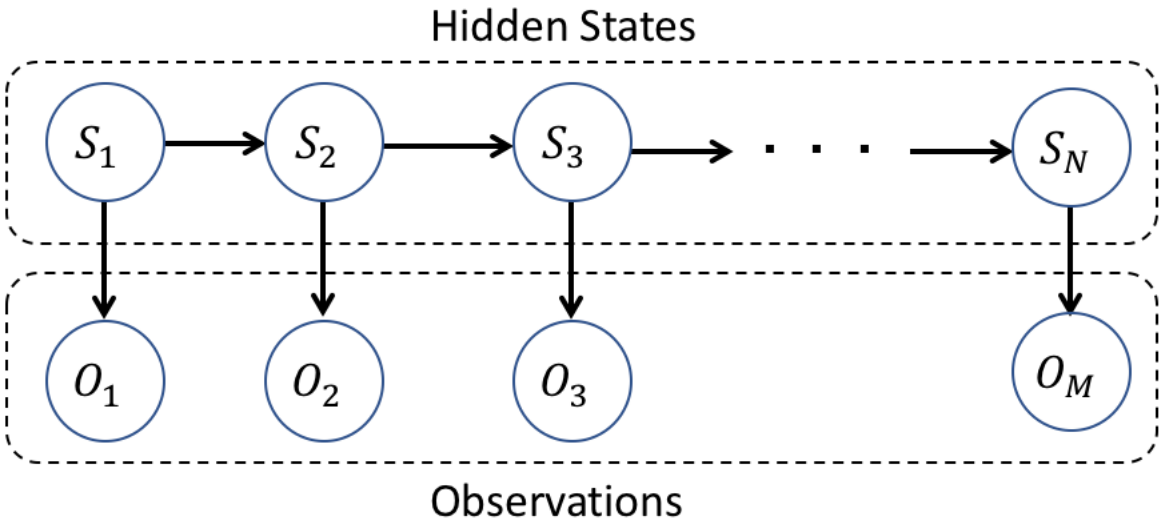


Figure 2.3: A Trellis Diagram representing a Hidden Markov Model.

Figure 2.3 demonstrates the configuration of an HMM in a Trellis Diagram format. In terms of a joint

distribution of a sequence of states and observations, this configuration can be factored as in Equation 2.1.

$$P(s_{1:T}, o_{1:T}) = P(s_1)P(o_1|s_1)\prod_{t=2}^{T} P(s_t|s_{t-1})P(o_t|s_t) \tag{2.1}$$

Within this framework, we can specify a HMM consisting of N states by the following parameters:

- **State Transition Matrix**

  A matrix containing the probabilities of transition from each state $s_i(t)$ to state $s_j(t+1)$, $1 \leq t \leq T$, $1 \leq i, j \leq N$ can be defined as:

  $$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}, \text{ where } a_{ij} = P(s_j(t+1)|s_i(t)) \text{ and } \sum_{j=1}^{T} a_{ij} = 1 \quad \forall 1 \leq i \leq N$$

- **Observation Emission Matrix**

  For those cases where the set of observations consists of only M discrete values, a matrix containing the probabilities of emitting each observation $o_j(t)$ from state $s_i(t)$ is given as follows:

  $$\mathrm{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M1} & b_{M2} & \dots & b_{NM} \end{bmatrix}, \text{ where } \sum_{k=1}^{M} b_{ik} = 1 \quad \forall 1 \leq i \leq N$$

  For a continuous set of observations, the probability density of emitting observation o_t while in state s_i is $P(o_t|s_i)$ where, as required by a probability density, $\int_O P(o|s)do = 1$.

- **State Initialization Probability**

  The probability of the HMM starting in state s_i is given by π_i

Having defined the structural parameters of a HMM, several different kinds of problems can be addressed; two of the most important for our interests are as follows:

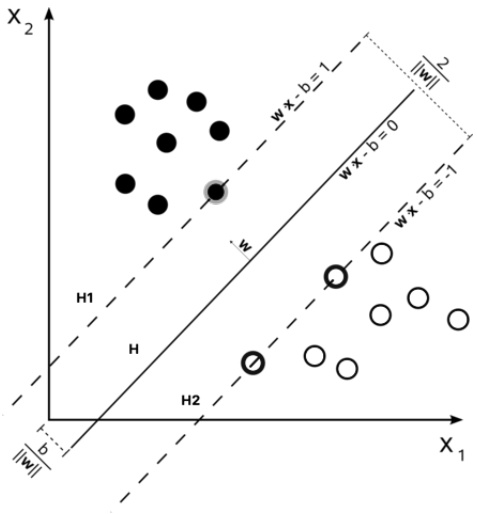
1. Given a set of such models $\Theta_h = (A_h, B_h, \pi_h)$, what which model H^* maximizes that probability, such that $H^* = \text{argmax}_h[P(O|\Theta_h)]$?

2. Given a model $\Theta = (A, B, \pi)$ and a sequence of observations $O = (o_1, o_2, ..., o_T)$, what is the most likely sequence of hidden states S^* emerging from this observation sequence, where $S^* = \text{argmax}_S\, P(O, S|\Theta)$?

2.5 SVM

In contrast to an explicit probabilistic model present in an HMM, a Support Vector Machine (SVM) estimates the decision surfaces directly. In a binary linear SVM problem, for example, the goal is to separate data

points using a decision region. This is achieved with separating hyperplanes in the space of the data points, or in a non-linear transformation of that space.



Source: Wikipedia
Figure 2.4: Maximum-margin hyper-plane and margins for a binary SVM classification.[3]

Figure 2.4 shows a hyperplane in a space of two-components datapoints x defined as a plane H. Such a plane can be presented with a normal vector ω and a scaling parameter b in the form of $\omega^T x - b = 0$. The data points nearest to the hyperplane the points that if removed, would change the position of the H, are called Support vectors. If we define two parallel hyperplanes H1 and H2 so that they pass through the support vectors and separate the two classes of data The two other hyperplanes shown, H1 and H2, are parallel to H and the distance between them is called the margin. The closest positive and negative class datapoints lying on H1 and H2 are defined as support vectors. It is an easy exercise to show that the width of the margin is equal to $\frac{2}{||\omega||^2}$. The goal is to find the optimal hyperplane H which maximizes the margin or minimizes $\frac{||\omega||^2}{2}$ over the weight vector, ω and the scaling parameter b, subject to constraints. Formally, we seek to minimize $L(\omega)$

$$L(\omega) = \frac{1}{2}||\omega||^2$$

subject to:

$$y_i(\omega^T x - b) >= 1; \forall i$$

To address this problem, Lagrangian optimization can be used using the Lagrange multipliers to achieve the optimal hyper-plane parameters. For many problems, the separation can only be achieved using nonlinear surfaces. The key point here is to project the dataset from nonlinear space to a high dimensional eigenspace using kernel functions. A key class of kernel functions used with SVMs map datapoints (e.g., feature vectors for each of a training example and a vector to be classified) into a higher dimension and then calculate a function of the inner product for those mapped vectors in that higher dimensional space. As a result, the

nonlinear problem can become linearly separable, without the need to explicitly represent the separating boundary within the higher-dimensional space.

The approach of increasing dimensionality described above is effective but has a huge computational burden. It can easily proven that the training step in finding a SVM model by solving the optimization problem only needs the training samples to compute pair-wise dot products $\langle \vec{x_i}, \vec{x_j} \rangle$ where $\vec{x_i}, \vec{x_j} \in \mathbb{R}^N$ [18]. It turns out there are functions that given two vectors u and v in $\mathbb{R}^N$, compute the dot product between them in a higher-dimensional $\mathbb{R}^M$ without requiring us to explicitly transform them to $\mathbb{R}^M$. Such functions are called kernel functions, $K(\vec{x_i}, \vec{x_j})$ and can be used to learn nonlinear decision boundaries for SVMs by replacing the pair-wise dot products in a higher-dimensional $\mathbb{R}^M$ space while remaining in $\mathbb{R}^N$.

2.6 Cassandra

Cassandra is a highly scalable distributed database designed to manage large amounts of structured data. It is a masterless cluster, ensuring full service with no single point of failure. It also provides a Cassandra query language shell (cqlsh) command-line interface allowing users to communicate with it. Using this shell, one can execute the Cassandra Query Language (CQL).

2.7 Data Analysis Framework: Apache Spark

Apache Spark is an open source big data processing framework which provides speed and ease of use in extracting complicated analytics from data. Spark makes it quick to write applications in Java, Scala, Python or even R. In addition to the *MapReduce* operation – a programming paradigm which enables scalability over Hadoop clusters [6] – it supports SQL queries, streaming data, machine learning and graph data processing. One can combine all these powerful capabilities to come up with a single pipeline for accessing data and processing it in a fast, distributed and fault-tolerant environment.

Figure 2.5 shows a flow diagram of a Spark setup and how data enters the Spark Streaming library from multiple static or streaming data sources, and Spark Core along with other libraries such as Spark SQL and MLlib used for analytical purposes.

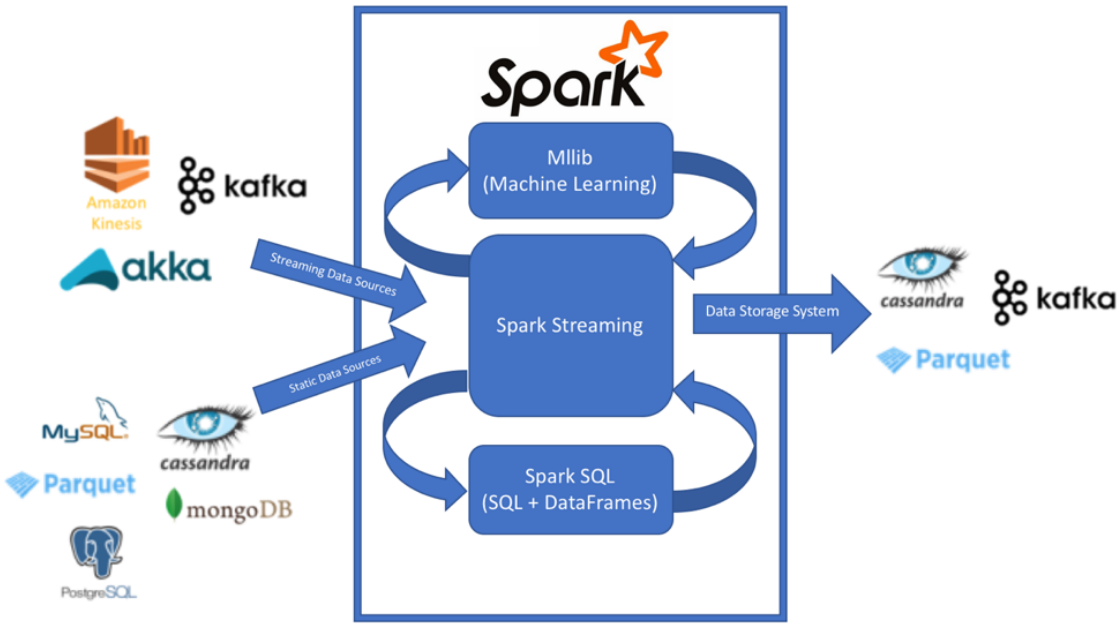


Figure 2.5: Spark Framework Ecosystem - Spark core along with different libraries.

Chapter 3

Simple HMM incorporated ABM

This chapter includes text drawn from a manuscript entitled "Prospective Detection of Foodborne Illness Outbreaks Using Machine Learning Approaches" by Aydin Teyhouee, Sarah McPhee-Knowles, Cheryl Waldner and Nathaniel Osgood, published in Proceedings of the 2017 Social, Cultural, and Behavioral Modeling (SBP-BRiMS) Conference [44]. The author's contributions are described in Chapter 1. The agent-based simulation model in this chapter is an extension of the earlier work by Sarah McPhee-Knowles and the parameter values and assumptions about the interaction among different model agents are derived from her work. All parts of the model associated with the HMM has been added to this previous work [27].

3.1 Introduction

Each year, a large population worldwide suffer from foodborne illness. While the public health inspection regime of food vendors successfully prevents many potential illnesses, the dynamic nature of restaurants' kitchens, the human resource constraints on carrying out consecutive inspections and the time-consuming character of the inspection process allow violations to remain undetected and limit the completeness of food illness prevention. Moreover, numerous food poisoned people who show mild to moderate symptoms of illness never show up at clinics and health care centers, but are greatly curtailed and inconvenienced in their activity. While such cases impose stiff health, quality of life and economic costs, the absence of such data regarding these kind of illness occurrence in public health incidence records makes it challenging to identify a potential outbreak occurrence in a timely fashion.

Once an outbreak is declared, and outbreak investigation is generally launched. Outbreak investigation methods mostly rely on telephone interviews of the clinical registered patients classified as suffering from possible food poisoning, days or weeks after their illness. This phase of work also involves notable challenges. First, a given patient will be subject to forgetfulness about food vendors visited during a specified time period, making it harder to prioritize the most probable contaminated restaurants in an investigation. Second, and as a consequence, because of inaccuracies in the data collected and the prolonged investigation process, the adverse health and cost impacts of the outbreak will be magnified.

The general problem of prospective detection of disease outbreaks using machine learning approaches has attracted the attention of researchers. A central challenge in this new field is to diagnose the occurrence

of an outbreak in a fashion timely enough to help public health authorities in undertaking rapid outbreak control mechanisms. However, the applications of such work to foodborne illness outbreaks has been very limited [19].

Machine learning provides a set of tools which can be applied in different problem domains for data analysis. Given that the challenge of detecting foodborne illness outbreaks consists of identifying the evolution of the categorical latent state (outbreak vs. non-outbreak) of a system (municipality) over time in the light of noisy observations (incident cases) strongly influenced by state, Hidden Markov Models (HMMs) offer a particularly attractive analysis lens. Here we seek to distinguish between these outbreak and non-outbreak states based on our observations of the number of reported illnesses. Although this is not the main goal of our presented work, we compare the findings of the HMM with the results of a Support Vector Machine (SVM) model, which fails to take into account the temporal context of data. In order to assess the accuracy of the machine learning models, we will use synthetic ground truth data from a previously contributed empirically-grounded agent-based model (ABM) of foodborne illness [27] .

As McPhee-Knowles shows in her model [27], and reflecting more recent successes in fieldwork by the authors, we particularly investigate how sentinel-based reporting of illnesses via smartphones where the affected individual does not show up in clinics could improve our inference about the potential outbreaks. To evaluate this, we will simulate two data collection regimes. The first regime complements traditional data with reports of subclinical illnesses provided by a small sentinel population, constituting just 4% of the total population. While this first data collection regime could be carried out with a number of technologies such as designated social media channels, call-in lines, and web-based mechanisms, we note that such a system has been successfully utilized over many months by the authors, employing the Ethica [1] smartphone-based epidemiological data collection system [40, 41]. In the second regime, we will use clinical data only, reflecting presentation by victims of possible foodborne illness to healthcare centers.

3.2 Foodborne Illness Model

Details of the first version of the foodborne illness ABM that serves to generate the synthetic time series is described in a previous contribution [27]. However, because of notable differences between that version and the one contributed in this thesis, we are going to make some general comments about the new version. Both models offer a stylistic depiction of a municipality that includes three main agents (the second model has an extra static Home agent): Consumers, Restaurants and Inspectors as actors. In the scenarios examined here, the municipality included a population of 5000 persons, 100 restaurants and one inspector.

Restaurants as shown in Figure 3.1 can be either in a non-contaminated or contaminated state, with a transition hazard from the former to the latter such that on average, one restaurant per year becomes contaminated.

The inspector can be in one of two modes: Routine inspection and outbreak response. In routine inspection

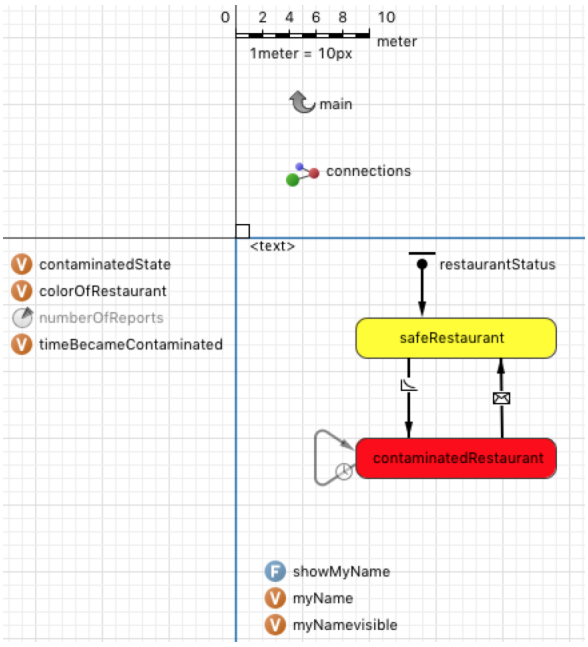


Figure 3.1: Statechart and variables for Restaurant agent

mode, the inspector transitions between restaurants in a round-robin fashion, with contaminated restaurants being subject to a probability of 50% of being diagnosed and rectified (thus transitioning back to a non-contaminated state). An outbreak is assumed to be declared if at least two (as specified in the original McPhee-Knowles model) clinically-presented cases occur. In an outbreak response mode, the inspector makes prioritized visits to restaurants according to the number of times that they have been identified (via faulty individual memory or via the geo-stamped records of sentinels) by those presented at clinics or (for the sentinel scenario) those who have reported their illness via phones. In outbreak response mode, an inspector who visits a contaminated restaurant is assumed to detect that restaurant with complete accuracy, and to eliminate the source of contamination.

Individual persons are associated with certain static and dichotomous degree of care in food handling and storage, and are at any time in one of three health states: Healthy, clinically-presenting ill, and not-clinically-presenting ill. Such a person is treated as requiring to eat one time per day, with each such meal taking place either at home or in a restaurant. 6.7% of this population eat at a restaurant daily, 30.9% three times a week, and 23% eat out once a week; the remaining 39.4% visit a restaurant once every two weeks [5]. Visits to restaurants are remembered by an individual. Absent the app carried by sentinel individuals, recall is imperfect, with a probability-per-week of forgetting a given restaurant of 5% to 20% (Ref [5], appendix A). Both home and restaurant meals are associated with empirically estimated probabilities of triggering foodborne illness, with the probability of a given person becoming ill from a home-cooked meal depending on that person's care in food handling. Within the model, 20% of the population is associated with good food handling skills, while the balance are associated with poor food handling skills [35], [8]. Independent of their source, foodborne illnesses developed in the model are classified clinical and lead to presentation for care with

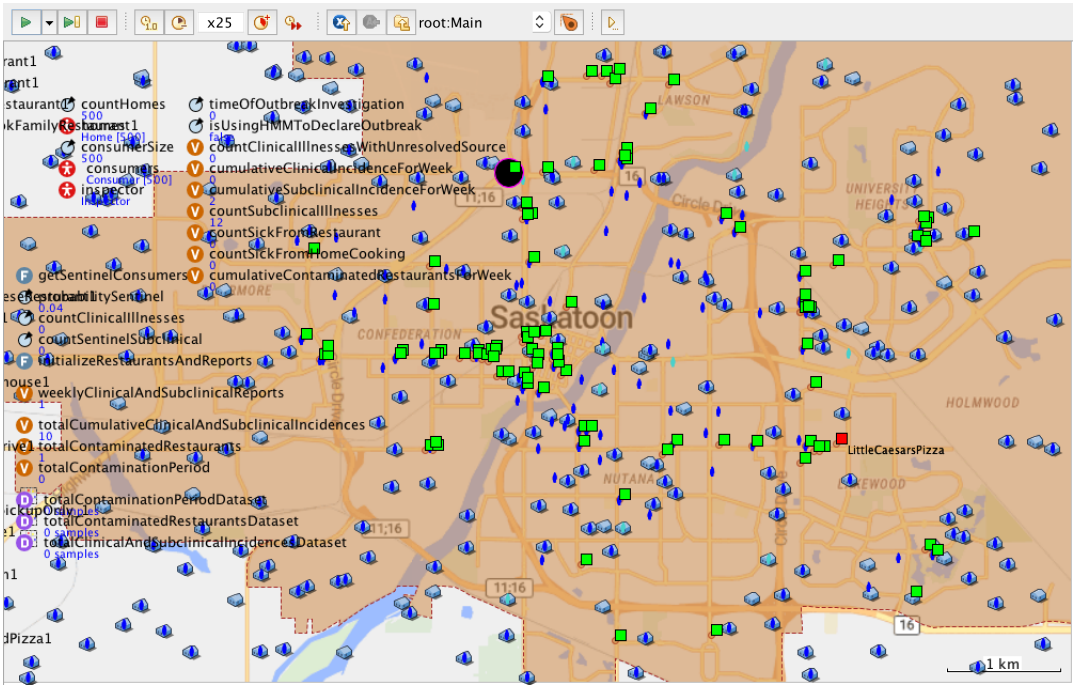


Figure 3.2: A snapshot of Foodborne Illness model

a small probability (0.005), with the remainder remaining not presenting at clinics. Following a fixed period of time (2 days), individuals experiencing foodborne illness symptoms are treated as recovering, and return to a healthy state. For analysis, each week, the model reports the incident case counts of clinically-presenting and not-clinically-presenting illness and the count of contaminated restaurants.

3.3 HMM and SVM Configurations

In this problem, we focus on discrete time characterization, with each time point representing a single week. The system transitions between two states s_t: a state in which the municipality includes a contaminated restaurant (henceforth termed the "outbreak" state) and $s_t = 1$, and one in which no contaminated restaurant is present and $s_t = 0$. Each such state is associated with a distribution for the observables $y_t(t = 1, ..., n)$: clinically-presenting cases and (for the sentinel scenario) not-clinically-presenting cases, where n is the n'th week. That is, for a given state s_t, $y_t|s_t \sim f_k(y_t; \theta_k)$, where $k \in \{0, 1\}$, f_k is a pre-specified density (e.g., univariate or multivariate Gaussian or Poisson) and θ_k are parameters to be estimated. The unobserved state space, $s_t(t = 1, ..., n)$ is modelled by a two-state homogeneous Markov chain of order one with stationary transition probabilities:

$$p_{kl} = P(s_{t+1} = l|s_t = k),$$

where $k, l \in \{0, 1\}$ denote the two states of s_t (0: non-outbreak; 1: outbreak). For example, p_{01} is the probability of switching from the non-outbreak to the outbreak state. Note that in this Markov-dependent mixture model, y_t is conditionally independent of all the remaining variables, given s_t. In Figure 3.3 a binary

HMM classifier is presented.

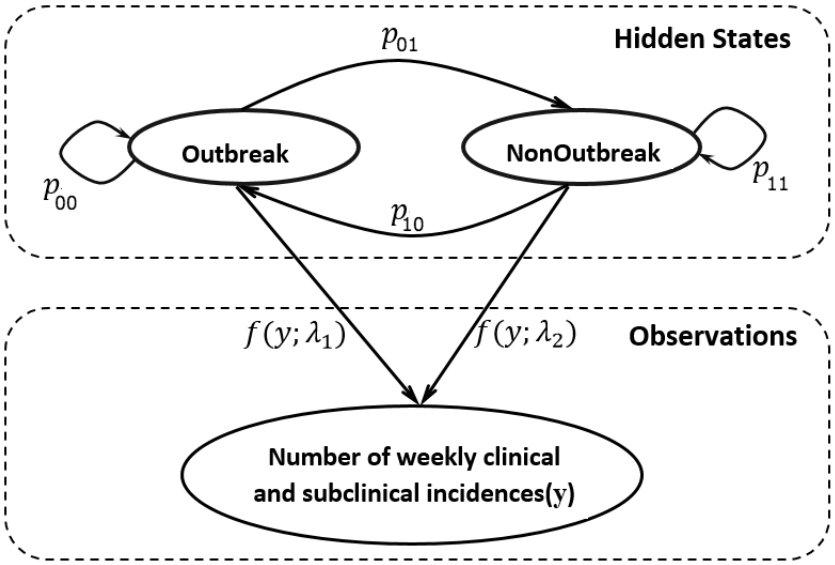


Figure 3.3: A binary HMM

As we are working with counted data in this experiment (number of reported illnesses), the above mentioned f_k is a Poisson density. A Poisson distribution can be used on counts of events where: (1) These events are independent of each other and have no effect on increasing or decreasing the occurrence probability of the other events. (2) The average frequency of the events can be calculated over the analysis time horizon; and finally (3) While asking about the number of the occurred events is meaningful, asking about the number of events that have not happened is senseless. This latter one, highlights the inherent difference between Poisson and Binomial distributions, where in the latter one we know the probability of win (p) as well as the probability of loss (q). So, to make it short, the expected frequency profile for the events of any dataset is definable in the format of a Poisson where all the mentioned conditions are true.

An attraction of HMMs is the fact that it is possible to estimate their parameters using a variety of parameter estimation methods – including the iterative Expectation Maximization (EM) algorithm. A key idea in EM algorithm is to obtain the maximum likelihood estimate of the unknown parameters given the complete set of data (the combination of the expected value of the unknown data given their distribution and the known data) and then iterating the procedure until the estimate converges.

A sample of datapoints (the weekly number of clinically-presenting and not-clinically-presenting illness reports) is shown in Figure 3.5. A preinvestigation over the dataset and by plotting the histograms corresponding to each of the two datapoint clusters, Figure 3.4, one can observe:

1. A low level of illness occurrence, where the weekly incident case count can be modeled as a Poisson distribution with parameter λ_1.

2. And a high level of illness occurrence, where the weekly incident case count can be modeled as a Poisson

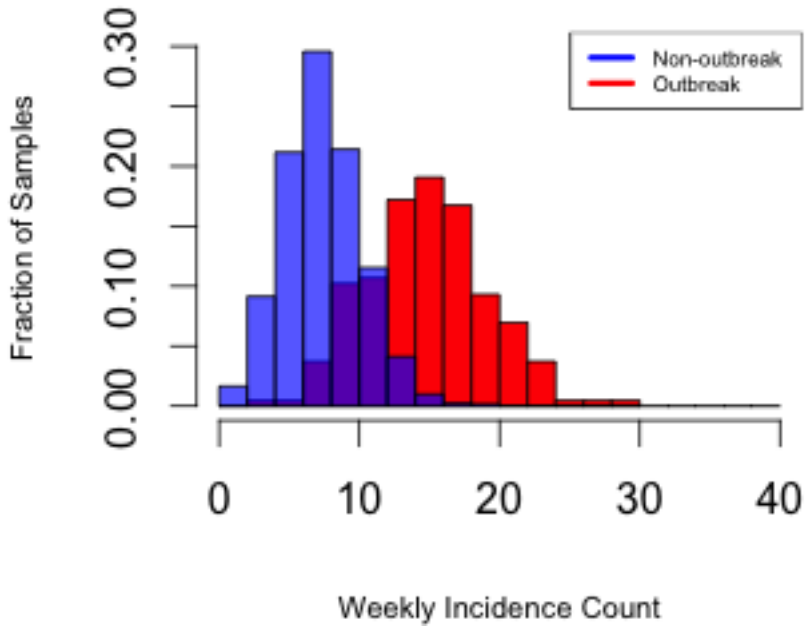


Figure 3.4: Histograms corresponding to each of the two clusters of datapoints

distribution with parameter λ_2.

The iterating process of converging the Poisson distributions' lambda parameter is performed by assigning the two above mentioned observed Poisson distribution parameters to specify a starting model for the EM algorithm: $\Omega_0 = (\pi_0, P_0, b_0)$, where π_0 is the initial matrix, P_0 is the (2×2) transition matrix and b_0 is the (1×2) emission matrix containing the first guessed lambda parameters for each of the Poisson distributions. In this study, a package named mhsmm [32] in the R statistical computing framework(RDevelopment Core Team 2010) was used for parameter estimation. This package performs inference in HMMs as well as hidden semi-Markov models. Input of the HMM consisted of the weekly results from the ABM, including just clinically-presenting illness for the first scenario, and the sum of both the clinically-presenting and not-clinically-presenting instances of illness for the second scenario. For training and cross-validation – a technique used in the training phase to define a data set to test the model in order to limit problems like overfitting, underfitting and get an insight on how the model will generalize to an independent data set – the number of contaminated restaurants in successive weeks was rendered into a dichotomous variable serving as ground truth, assuming that any contaminated restaurant number greater than 1 corresponded to the state of an outbreak (whether declared or not).

To solve the problem with a SVM, we used a package in R named (e1071) [12] for the classification, and – as in the case of the HMM – the results extracted from the ABM were used in two scenarios. In the second scenario, both the clinically-presenting and sentinel not-clinically-presenting reports are considered, while in the first scenario, only the clinically-presenting reports are considered as observations.

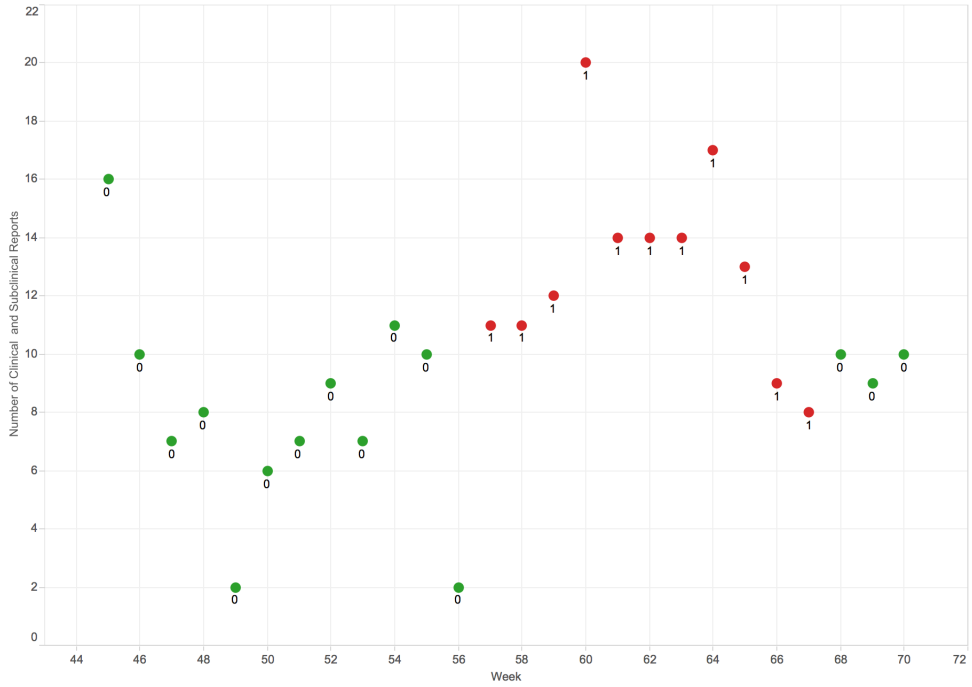


Figure 3.5: A sample of datapoints form week #45 to week #70 - "0" and "1" correspond to No-outbreak and Outbreak states, respectively

3.4 Results

The simulated 10,000-day (almost 27 year) dataset captured from the agent-based model was split up into training dataset (75%) and testing dataset (25%). As is traditional for classification models, the results of classifying the test dataset are characterized in a confusion matrix. This confusion matrix shows how well the model behaved in labeling the existing observations corresponding to each of the classes. The following table shows how the confusion matrix is calculated. Moreover, ***sensitivity*** or ***recall*** and ***specificity*** are two other parameters showing the True Positive Rate (which in our experiment is the percentage of sick people correctly identified as sick) and True Negative Rate (the percentage of healthy people correctly identified as healthy) respectively, and are calculated as follows:

$$Sensitivity = \frac{TruePositive}{TruePositive + FalseNegative}$$

and

$$Specificity = \frac{TrueNegative}{TrueNegative + FalsePositive}$$

It bears emphasis, the positive condition and negative condition encode a state of outbreak as "1" and non-outbreak as "0", respectively.

Table 3.1: Confusion Matrix

Total Population	Predicted Condition Positive	Predicted Condition Negative
Condition Positive	#True Positive	#False Negative
Condition Negative	#False Positive	#True Negative

3.4.1 Results of the Hidden Markov Model (Using both clinically-presenting and not-clinically-presenting case counts)

For the HMM approach, the model was initialized with the following values:

$$\Omega = (\pi_0, P_0, b_0)$$

where the ***initial matrix***, π_0 is:

$$\pi_0 = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}$$

the initial ***transition matrix***, P_0 is:

$$P_0 = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$$

and finally, the initial ***emission matrix***, b_0 is:

$$b_0 = \begin{bmatrix} 1 & 4 \end{bmatrix}$$

These parameters are used by the EM algorithm to produce a maximum likelihood estimate Hidden Markov Model to describe the data.

We evaluated models in the terms of confusion matrix, sensitivity and specificity resulting from a cross-validation procedure over the test data. The model parameters $\Omega = (\pi, P, b)$ and its performance for scenario 2 (the case where our observation includes both clinically-presenting and not-clinically-presenting instances) are described as follows:

$$\pi = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 0.990 & 0.010 \\ 0.055 & 0.945 \end{bmatrix}$$

$$b = \begin{bmatrix} 7.869088 & 15.860456 \end{bmatrix}$$

and the performance is as per Table 3.2:

$$Sensitivity = 0.9318182$$

and

$$Specificity = 0.9840764$$

3.4.2 Results of the Support Vector Machine Model (Using both not-clinically-presenting and clinically-presenting case counts)

The predictive performance of the SVM was measured through a "cross-validation" process over "cost" of constraints violation with 10-fold sampling method and then a model with lowest misclassification error rate with the following parameters was chosen,

1. SVM-Type: C-classification
2. SVM-Kernel: linear
3. cost: 1
4. gamma: 1

The performance of the model over the testing dataset is shown in Table 3.3 and in terms of sensitivity and specificity:

$$Sensitivity = 0.6590909$$

and

$$Specificity = 0.977707$$

3.4.3 Results of the HMM and SVM (Using clinically-presenting case counts)

In this scenario where only the clinically-presenting incidences were considered, both the HMM and SVM approaches failed in labeling the outbreak state. In this case, the number of reported clinically-presenting cases were very rare, and all incidences were labeled as non-outbreak state. Table 3.4, shows the confusion matrix for this scenario.

3.5 HMM-aided Outbreak Triggering System

To investigate whether the HMM could improve syndromic surveillance monitoring and linked disease outbreak detection systems, the ordinary illness triggering method (which is applied once at least two clinically-presenting cases happen) as mentioned in detail in Section 3.2 was replaced with the resulted HMM in Section 3.4.

To carry out this HMM-based outbreak detection mechanism, the ABM uses the HMM parameters calculated in Section 3.4.1 (i.e., the probability distribution for each of the “outbreak” and "non-outbreak" states and the transition probabilities between states) to calculate the updated probability of being in an outbreak state in light of the previous value and the reported not-clinically-presenting and clinically-presenting case counts. If the calculated probability at the beginning of any given week is greater than a threshold (which in this experiment was set to 0.6), a message is sent to the inspector to trigger the transition to the outbreak state investigation. The recorded cumulative count of clinically-presenting and not-clinically-presenting illnesses over 10 years for 12 realizations in two different outbreak declaration regime (HMM outbreak triggering method and the ordinary method) is shown in the Figure 3.6. Results represent a quite significant decrease in the number of illness reports due to the fast detection of contaminated restaurants by applying the HMM outbreak declaration approach. As demonstrated by Figure 3.7, this approach reflects a similar reduction in the time period a given contaminated restaurants remains contaminated before being identified and cleared.

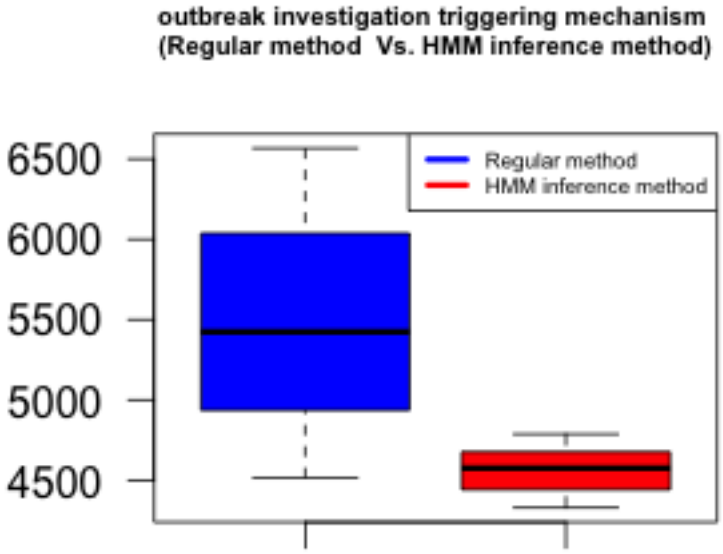


Figure 3.6: Regular and HMM-based outbreak declaration comparison over 12 realizations for each (Number of illness incidences [person/10-years])

3.6 Conclusion

Performing disease outbreak detection based on reported illness cases is an important function for syndromic surveillance systems. We treated the existence of a foodborne illness outbreak as a latent element of state and developed a Hidden Markov model for syndromic surveillance. We evaluated our disease outbreak detection approach using an empirically grounded previously contributed ABM of foodborne illness, comparing the results from HMM to those secured using an SVM approach. Finally, in light of the highly favourable results from the HMM, we further used the foodborne illness ABM to evaluate the public health gains secured through use of a HMM-based outbreak detection trigger, as compared with a traditional one based on case

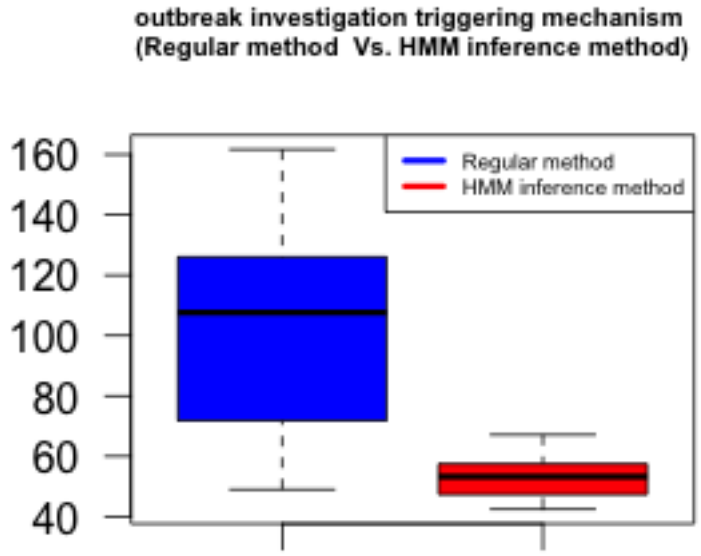


Figure 3.7: Regular and HMM-based outbreak declaration comparison over 12 realizations for each (Contamination period per contaminated restaurants [day/10-years])

counts. Despite the highly noisy data present, and overlapping distributions of incident case counts between the outbreak and non-outbreak states, the results reported in this paper suggest a promising future for the use of hidden state variables to model the changing dynamics of observed surveillance time series, and for HMMs in general in outbreak signal detection. Moreover, the results from the first and second scenarios (considering both clinically-presenting and not-clinically-presenting reports vs. considering only clinically-presenting reports) reveal that use of smartphones that can record locations and offer channels for reporting foodborne illness could improve our inference about the potential outbreaks. Finally, evaluation of HMM-based outbreak triggering mechanisms using ABMs suggest that significant public health gains may be secured when combining new technologies for syndromic surveillance with machine-learning based outbreak signal detection mechanisms. This work suggests promising lines of future work, including in extending our outbreak detection approach with multiple data streams obtained from mobile applications, such as restaurant-specific traffic and illness counts.

Table 3.2: Confusion Matrix for HMM - First Scenario

Total Population	Predicted Condition Positive	Predicted Condition Negative
Condition Positive	41	3
Condition Negative	5	309

Table 3.3: Confusion Matrix for SVM - First Scenario

Total Population	Predicted Condition Positive	Predicted Condition Negative
Condition Positive	29	15
Condition Negative	7	307

Table 3.4: Confusion Matrix for SVM and HMM- Second Scenario

Total Population	Predicted Condition Positive	Predicted Condition Negative
Condition Positive	0	44
Condition Negative	0	314

Chapter 4

Targeted HMM

We begin this chapter with a top down view into our simulation model, with later sections continuing the investigation with regards to the goals defined in Chapter 1.

4.1 Model Architecture

The individual-level character of food consumption and associated variable preferences and risks, the dependence of outbreak response on individual history, and the non-contagious character of many foodborne illnesses suggests that an agent based approach is attractive for this model. People have their own specific preference for eating home-made food or restaurant food. They further possess different levels of skills in food handling, differing attitudes with respect to presenting for care given symptoms, and additional personal characteristics such as frequency and vendor preferences with respect to eating out, motivating an examination at the individual level. While some large scale effects can be observed in aggregate models (such as those defined using System Dynamics tools), most of the interesting phenomena depend on the individual interactions captured most readily in Agent Based and Hybrid Models. As such, all of the modelling here is undertaken within the ABM modelling approach – despite the fact that very visibly higher processing demands of ABM models emerge when the number of agents scales up to the population of a city [27].

4.2 Model Formulation

Guided broadly by the ODD (Overview, Design concepts, and Details) protocol for the specification of agent-based models [16], we offer here the details of our Agent Based Model (ABM).

4.2.1 Overview

Purpose

As mentioned in Chapter 2, the purpose of the ABM presented here is to investigate a series of scenarios examining distinct approaches to triggering foodborne disease outbreaks in a municipal surveillance system, and to use such scenarios to evaluate the outcomes of such scenarios in terms of outbreak duration and cumulative incidence of disease.

State variables

This simulation incorporates four broad types of computational processes: Environment initialization, agent initialization, agent behavior simulation, and agent-agent interaction. Although model logic does not depend on spatial and GIS elements, to improve understanding and visualization, the environment characterizes the mid-western Canadian city of Saskatoon using the GIS component in Anylogic. Because we did not seek a model scope that would make model dynamics dependent on details of agent routes to resources (e.g., for location-based resource selection), to enhance model performance, and to make the routing among agents independent of availability of online resources such as routing databases, straight lines were selected as the routes.

The model contains four different types of entities, each captured in distinct agent populations: Consumers, homes, restaurants, and a (singleton) inspector. The restaurants and homes each exhibit time-invariant latitude and longitude properties (parameters); the values of such parameters are randomly selected during model initialization in such as way as to scatter them over the map of the city; each consumer then resides in exactly one home. Consumers are initialized with parameters that select a random collection of their favorite restaurants (which is of equal size for all agents). Consumers further draw their sequence of possible restaurants to visit with uniform probability; they are also initialized with parameters that select how frequently they eat restaurant food, and whether they posses good food handling habits (a dichotomous attribute) when cooking at home. Specifically, with respect to the frequency of eating at restaurants, consumers are categorized into one of four groups: 6.7% of consumers eat out daily, 30.9% thrice weekly, 23% weekly, and 39.4% of them visit a restaurant once every two weeks [5]. With respect to the riskiness of food handling habits, with 20% probability, consumers are assumed to practice good food handling habits and 80% do not [35, 8].

Consumers are additionally characterized by two state variables: Whether or not they are ill, and – if so – whether they are presenting for care or not. Consumers are further associated with a fixed parameter as to whether or not they represent a sentinel – i.e., whether they are equipped with smartphones enabling them to report their illness. In our model, the value of this parameter was set to 0.04 – indicating that approximately 4% of the total population serves as sentinels [27].

Restaurants have a single, dichotomous, state variable, indicating whether they are in a contaminated or uncontaminated state. To characterize the ongoing risk of restaurant contamination, there is a global hazard rate by which a safe restaurant probabilistically becomes contaminated. A contaminated restaurant is then assumed to return to an uncontaminated state with certainty only if they are investigated by the inspector agent in outbreak investigation mode. Otherwise, the restaurant's contamination in the model may be resolved probabilistically during routine inspection. Following such a routine investigation, the outbreak is assumed to be resolved with a probability of 50% – reflecting the fact that the inspection is not as thorough as it is in an outbreak mode investigation.

The singleton Inspector agent is also associated with a single dichotomous state variable: The inspector is

either in a routine inspection mode visiting restaurants daily in a round robin fashion according to a random ordering set at model initialization, or is – following a triggering indicator – in an outbreak investigation mode, focusing on restaurants most frequently reported as having been visited by ill consumers.

In the model the probability of getting sick from a restaurant is taken to be three times the per day chance of illness from eating a home meal prepared with good safety measures (i.e., approximately $3 \times 1.5E-4$, a value derived from empirical data for formulation of the model extended by this chapter [27]).

Process overview and scheduling

For consumer agent behavior simulation, we have implemented a food consumption scenario in which agents decide every day whether to eat at home or in a restaurant according to their frequency parameter value of eating out (which turns into a dichotomous daily decision of eating at home or a at a restaurant using a uniform distribution). An agent might get ill if they don't adhere to good food handling at home or if they eat at a contaminated restaurant. A statechart representing the consumer's behavior before and after getting sick is shown in Figure 4.1.

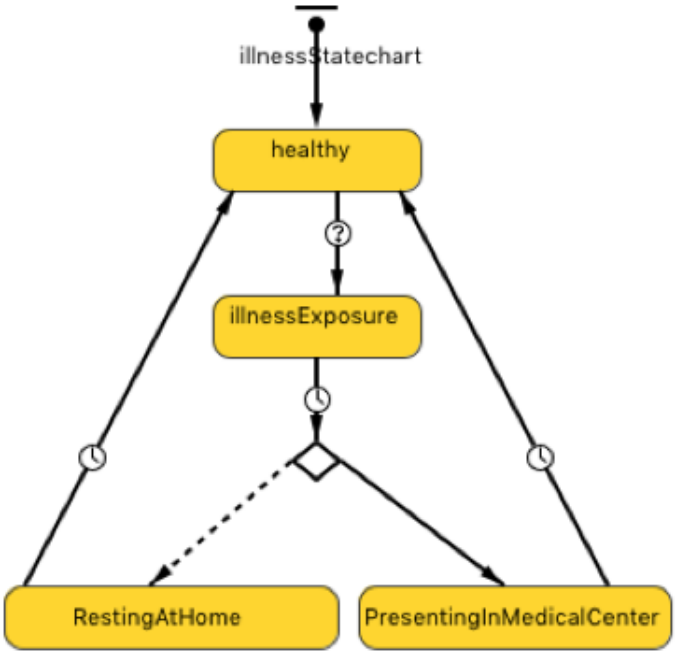


Figure 4.1: Illness Statechart for Consumer Agent

As indicated by this statechart, ill consumers might elect to present at clinics/hospitals, or might decide to remain at home. In the model, 80% of the consumers use poor food handling practices and 20% use safe food handling practices when cooking at home. It is assumed that consumers who use poor food handling practices are twice as likely to contract an illness as those who use safe food handling practices [35, 8]. If they present for medical care or if they are members of the sentinel group, their illness and their visited restaurants during

the last past week are reported and subsequently available to the inspector. When reporting restaurants at the point of care, consumers who are not sentinels are assumed to exhibit a probability of forgetting a visited restaurant a given amount of time ago; that probability rises over the time since visiting that vendor. By contrast, for scenarios positing recording and reporting of data by sentinels, such reporting takes place via smartphone-based location tracking; for simplicity, I follow [27] in assuming that the app-based recording of locations for sentinel perfectly identifies the sets of restaurants that were visited by that sentinel.

In the baseline model, two cases of clinical presentation trigger an outbreak alert, causing the inspector to change mode from routine inspection to an outbreak inspection regime where that inspector makes prioritized visit to the restaurants reported most commonly. If the restaurant visited by the inspector is contaminated, the inspector resolves the contamination and returns to the ordinary inspection regime; otherwise, the inspector begins the targeted visit process anew by removing the name of the visited restaurant from the sorted list and continuing on to visit the next most highly reported restaurant until finding the contaminated restaurant or no reported restaurants remain on that list. In the latter case, the situation is assumed to be interpreted as a circumstance in which the alarm has been based on clinically presenting cases with home food sources. In both of these latter cases, the inspector is assumed to resume routine inspection.

In our previous work [44] discussed in detail at Chapter 3 of this thesis, a Hidden Markov Model (HMM), trained on a collection of synthetic datasets of incident illness cases and vendor contamination records from the empirically grounded simulation model [27] (extended here) was incorporated into an augmented version of that model to replace the triggering mechanism used to provoke outbreak investigations. The baseline experiment of that augmented model from Chapter 3 also supports the baseline experiment here for syndromic surveillance monitoring and disease outbreak detection under two data collection regimes: One involving traditional clinical reporting alone, and the other involving a sentinel population using a smartphone-based app for tracing location of food consumption and illness reporting. Findings of Chapter 3 and [44] suggested that while reliance on clinical presentation data offers poor potential for automatic outbreak detection, it can be highly effective to trigger outbreak response measures based on HMM-based classification even when such HMMs are informed by smartphone-based reporting by even just a very small (4% of population) sentinel group. To avoid repetition, we refer the readers to Chapter 3 and the published paper [44]. These scenarios offer a baseline set of – already competitive – results against which we compare the results of the investigation undertaken here.

4.2.2 Design

When the simulation starts – and after the GIS region is defined on the map – the initial population of restaurants, homes, consumers and an inspector are scattered over the map region. Upon initialization, a global parameter referencing a map from the designated unique name of each restaurants to the corresponding number of reports – initially zero – is further created. The map referenced by this parameter is updated by restaurant food consumers throughout the simulation as they get sick and report their suspected food

sources. All the parameters for the incorporated HMM derived by a model training procedure performed external to the Anylogic environment (as discussed in detail at Appendix B.1) are being deployed into the simulation model environment at the initialization phase.

Once per day for each consumer, an event goes off and lets a consumer decide where to eat. Based on the consumer's restaurant visitation frequency, the consumer might decide to visit a restaurant or eat at home, as realized according to the Movement Statechart in Figure 4.2. According to this statechart, if the decision is to eat out, the consumer leaves the state from being at home and transits to a randomly selected restaurant ("atRestaurant"); upon arrival, that consumer is assumed to immediately order and eat. The contamination state of a restaurant and the parameter indicating whether good food handling practices have been followed are the factors affecting the evolution of the health condition of the consumer for for meals eaten at a vendor or at home, respectively.

If the consumer is sickened by food consumption, that consumer will immediately transition in the illness statechart from the healthy state to one of the two states of resting at home or presenting at a medical center. Regardless of the illness source, and in accordance with literature indicating that only a small fraction of foodborne illness cases are reported, the foodborne illnesses developed in the model are classified as precipitating clinical presentation with a small probability (0.005), with most cases not leading to clinical presentation. Following a period of time fixed at two days regardless as to whether care-seeking was involved, individuals experiencing illness are treated as recovering, and return to a healthy state.

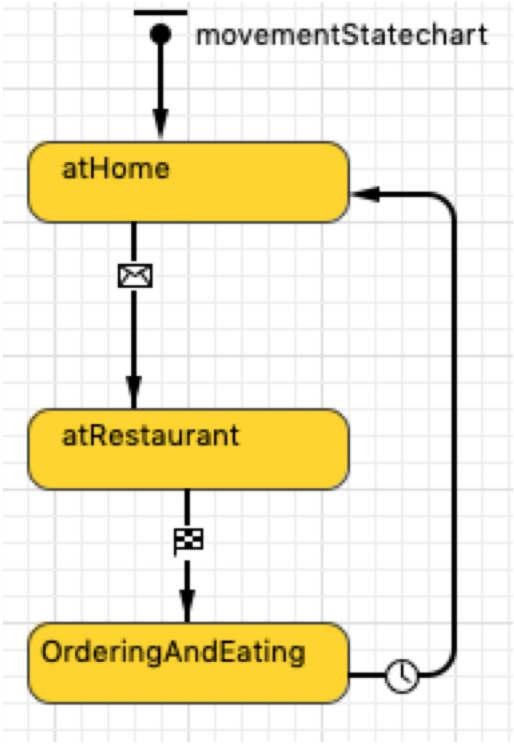


Figure 4.2: Movement Statechart for Consumer Agent

We ran the model for a small population size of 100 people and 5 restaurants over a period of 10 years to evaluate the performance of our new HMM model with a global state of no_outbreak and 5 restaurant-specific states of contamination. At the end of the model run, collected information – including the number of daily reports and the daily visitation count (intended for use in the binomial distribution scenario) for each restaurant and the enumeration of truly contaminated restaurants were exported out of the model through AnyLogic's database functionality.

The exported data was analyzed with the statistical package R, so as to derive the Transition Matrix values and identify the density functions corresponding to any of the states. Later, these matrix tables were imported into the database module inside the simulation model to allow them to be used in setting up the incorporated HMM. To give a better idea to the reader of how these values were obtained, the principle steps of this process are described in detail at Appendix B.1.

4.2.3 Details

An "HMM" event which lives in the Inspector agent class goes off daily, iterates over the restaurants, and calculates the HMM-calculated probability for each restaurant being in one of the underlying states, given the number of daily reports for that restaurant. In accordance with standard theory for HMMs as discussed in detail in Chapter 2 and Section 2.4, the calculated (posterior) probability of being in each hidden state is consists of the product of two values – a likelihood and a prior. The likelihood for a given restaurant-specific state depends heavily on the restaurant-specific observation, and constitutes the likelihood of observing the current observation vector (an observation of the count of illnesses for that restaurant) given that the such a state obtains – i.e., given that this restaurant is in fact contaminated or not. The value of the prior for a given state is derived from the vector of probabilities for the previous time step and the transition matrix; specifically, it represents the probability of being in that state in light of the probability of being in each possible state during the previous time step and the sum of the probability of having transitioned from each such previous state to the current state in the transition from the previous time step to the current time step, where such transition probabilities are specified in the transition matrix. The pseudo-code presented in Appendix B.2 shows how each of these two values are calculated during the model run.

In the following sections, each of the solutions defined in Section 1.2 of Chapter 1 will be discussed, and the performance of the obtained HMM will be evaluated. The results of incorporating that specific HMM into the simulation model are then presented and compared.

4.3 Results: Reports Count Driven HMM

In our first approach only the number of reports by presenting individuals were considered for training the HMM. Since presentation of those health care seeking individuals in clinics and hospitals is a Poisson process where only the average time between events is known, but the exact timing of events is random and also

the events are independent of each other, Poisson distributions were applied to obtain the emission matrix values.

4.3.1 Considering Reports Limited to Clinical Presentation Cases

In this scenario where only the clinical presentation reports were considered. Since only a small fraction of foodborne illnesses result in clinical presentation, within the model time horizon, it was not feasible to collect enough data reporting restaurant-specific counts in the model; while very long runs could be used to collect more such reports, a challenge blocking use of such information for operational decision-making is that the contaminated restaurants would be cleared by the time that analysis is complete. As a result, the HMM components – the transition matrix and the density functions corresponding to low and high number of complaints for each restaurant – were not viable to gather from the collected synthetic ground-truth data. By contrast, in Chapter 3, the clinical presentation data was dense enough for the transition and emission matrices to be inferred because the data was collected for the restaurants as a whole. Having no HMM configured for this – clinical presentation specific – scenario, there was no model to be incorporated into the simulation model and examined.

4.3.2 Considering Reports Not Limited to Clinical Presentation Cases

In this scenario, not only does the HMM consider reports of restaurants visited based on clinically presenting cases, but further considers reports by those reporting illness via their mobile phone application, but not presenting for care. If all the classes other than GNO (Global No-Outbreak) are categorized as a single contaminated state (GNO_NOT), then an ROC curve can be used for evaluation of this binary configuration of the results. Figure 4.3 depicts the ROC curve for this scenario, with the associated AUC (Area Under Curve) quantifying the performance.

ROC curves are insensitive to class balance, so although the occurrence of the restaurants to be contaminated (as our outbreak classes) are rare in comparison to the number of global non-outbreak class, the ROC curve and area under that curve can summarize the performance based on all thresholds, and remain informative.

Having trained the resulting HMM, it was incorporated into the simulation model.

4.3.3 HMM Incorporation into the ABM

HMM Used As Global Outbreak Alerting System (HMM-GOAS) As we mentioned earlier, in the baseline simulation experiment, with the occurrence of two clinically presenting incident cases, the inspector's investigation regime switches from a round-robin mode to outbreak response regime, where the inspector prioritizes visits to restaurants based on the number of times that a restaurant has been reported as being visited by individuals who reported being ill. Here, we seek to replace this triggering mechanism with the

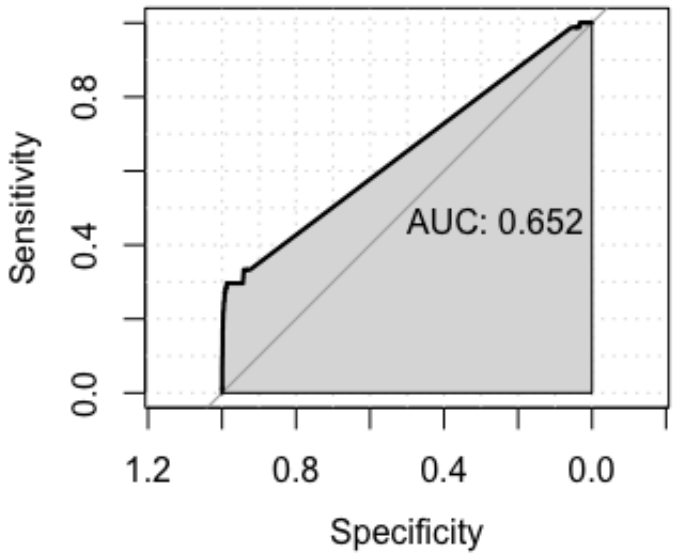


Figure 4.3: HMM Performance considering Clinical and Pseudo-Clinical Cases: ROC Curve and Associated AUC

implemented HMM as defined in the previous section, in which the inspector can switch to an outbreak investigation mode when the HMM detects a case of contamination and sends a global outbreak alert. This application of the HMM will be referred as HMM as Global Outbreak Alerting System (HMM-GOAS) from now on.

HMM Used As Contamination Source Detector (HMM-CSD) Instead of alerting the inspector of a likely outbreak, it would be ideal if the HMM could immediately direct that inspector to the contaminated restaurant, thereby reducing the time until contamination is eliminated. This HMM application will be referred to as HMM as Contamination Source Detector (HMM-CSD) hereafter. This approach would better exploit the potential of the HMM. For this purpose, the HMM-event which goes off daily was modified to iterate over the restaurants; if for any of them the HMM-calculated probability of contamination exceeds a global threshold, the event adds them to a collection of detected restaurants. This collection is provided to the inspector. If it is not empty, the inspector will switch into an outbreak inspection regime, and will investigate – and potentially resolve contamination in – any of those restaurants which have been correctly detected by the HMM as contaminated. Figure 4.4 depicts the change in the inspector's statechart for the realization of this method.

Figure 4.5 shows a box-plot of contamination duration resulting from the traditional outbreak triggering system as compared with the HMM-based systems in form of HMM-GOAS and HMM-CSD, respectively. The average and standard deviation of the presenting/non-presenting illness count temporal density (as characterized by the number of underlying illness per day) is also compared for these the three outbreak

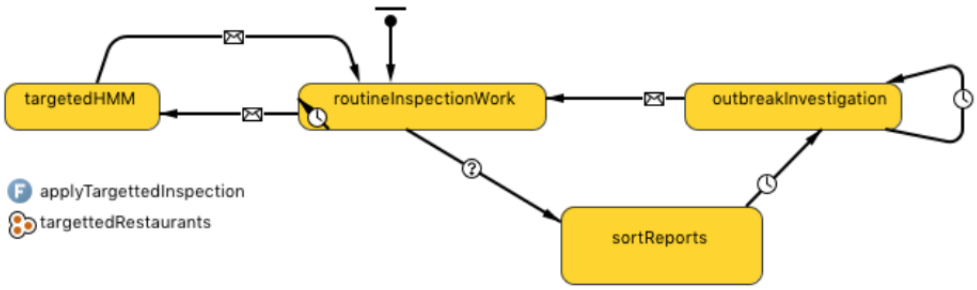


Figure 4.4: Inspector's Statechart - Targeted Inspection

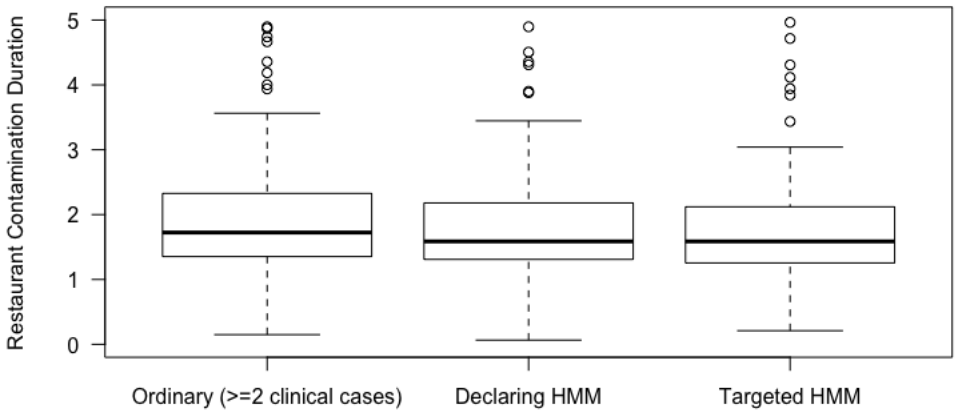


Figure 4.5: Contamination Duration

detection systems, as shown in Table 4.1.

In this scenario, the traditional method performs slightly better than the both of the two HMM-based methods; these two latter methods exhibit very similar performance.

4.4 Results: Reports and Visitations Count Driven HMM

In a second approach (as defined in Problem 3, Chapter 1.2), we investigated the benefit conferred by use of HMMs considering restaurants' daily visitation counts. The motivation lies in the fact that the extra information provided by such counts might aid identification of the risks associated with a given restaurant. The likelihood function is based on assumption that number of complaints from known visitors to each restaurant (as gathered by restaurant visitation count data, such as could be provided by the restaurant's point of sale [e.g., cash register] system) follows a binomial distribution. By running the ABM and harvesting both daily report counts and visitation counts for each restaurant, and knowing from the synthetic ground

truth ABM, for any given day, what restaurant is in a contaminated status, we are able to extract the Transition Matrix values – i.e., the probability of transition from or remaining in the contaminated and non-contaminated states, and also the Emission Matrix values – i.e., the binomial distribution parameters corresponding to each of those two hidden states.

4.4.1 Considering Reports Limited to Clinical Presentation Cases

Similar to what is discussed in section 4.3.1, when limited to considering clinical cases, HMM parameters were not feasible to estimate from the synthetic ground-truth data collected from the ABM due to the scarcity of clinical presentation cases. As a result, no further investigation was undertaken in this scenario.

4.4.2 Considering Reports Not Limited to Clinical Presentation Cases

The evaluation results of employing a Binomial instead of a Poisson distribution is shown in Figure 4.6 in form of a ROC curve and the AUC value.

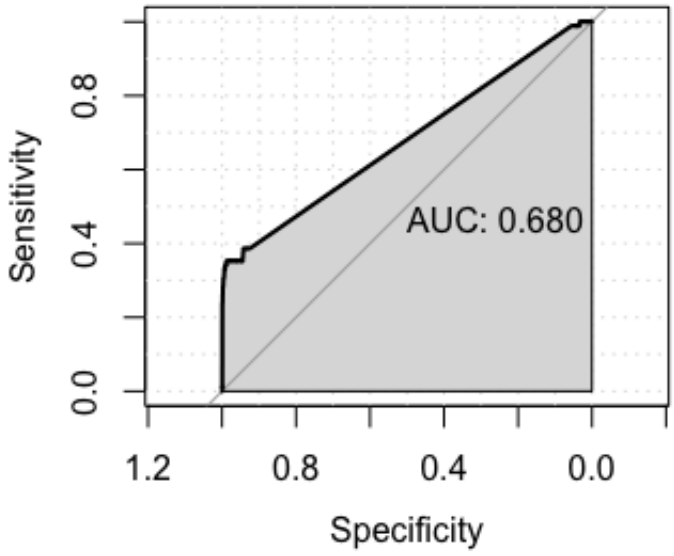


Figure 4.6: ROC Curve and AUC - HMM Performance considering Clinical and Pseudo-Clinical Cases and Visitation Counts

4.4.3 HMM Incorporation into the ABM

In this section, the ordinary outbreak triggering system was replaced with the HMM, so as to use that HMM to trigger the outbreak. Figure 4.7 shows a box-plot of contamination duration for the traditional outbreak triggering system as compared with the HMM-based systems (in form of the HMM-GOAS and HMM-CSD) based on binomial distributions inferred from observations, respectively.

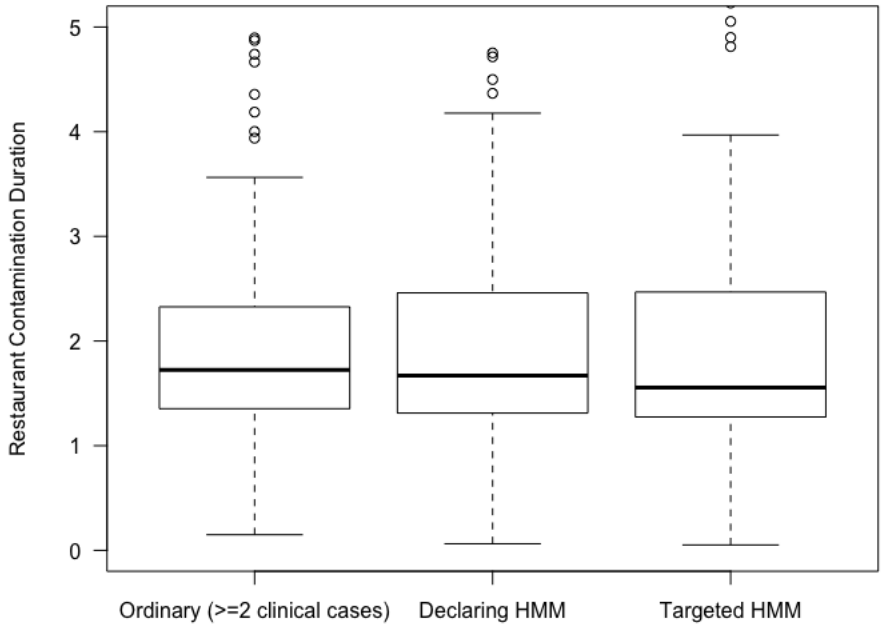


Figure 4.7: Contamination Duration considering Clinical and Pseudo-Clinical Cases and Visitation Counts

For this scenario, the average and standard deviation of the clinically-presenting and no-clinically-presenting cases counts are also compared as depicted in Table 4.2.

4.5 Conclusion

In this chapter, we used synthetic ground-truth data from a foodborne illness simulation model to come up with different configurations of HMMs. Later we incorporated these HMMs into the model to see how they change the effectiveness of outbreak control. Contrary to Chapter 3, where we conducted a top-down assessment depicting the ongoing outbreaks by considering the food-borne illness reports for all the restaurants as a whole, here the investigation focused on a bottom-up, restaurant-specific view. We considered the complaints – whether registered by clinics or sent over the phones – for each restaurant, and sought to extract HMM parameters with the observation values for any single restaurant. Later we used the predicted results in two ways: one considering its used only in alerting to occurrence of a new outbreak (and so making it similar to the HMM configuration on Chapter 3), and the second considering the HMM as a means of identifying the source of outbreaks by predicting the identity of the contaminated restaurant. Both of the approaches showed limitations. Such limitations reflect in part the fact that the statistical signatures associated with contamination for the restaurants were not distinctive enough to enable the HMM to effectively deduce the underlying hidden state based on the observation vector; since the simulation model lacks any distinctive feature for the restaurants (such as proximity to downtown/university, type of restaurants, etc.), the achieved distributions for the restaurants (based on the report counts assigned to them) are very similar.

Table 4.1: Mean and Standard Deviation of presenting/non-presenting illness counts for the Three Methods

	Mean	Std.
No-HMM	0.4965	14.38083
HMM-GOAS	0.5017	14.40924
HMM-CSD	0.5045	14.42962

Table 4.2: Mean and Std. of illness counts with clinical-presentation and no-clinical-presentation for Three Methods - Considering Visitation Counts

	Mean	Std.
No-HMM	0.496	14.380
HMM-GOAS	0.503	14.388
HMM-CSD	0.499	14.305

Chapter 5

Data Analysis with Spark

5.1 Introduction

In January 2016, an innovative study in the University of Saskatchewan under supervision and guidance of Dr. Cheryl Waldner and with support by Dr. Nathaniel Osgood was conducted to evaluate a new technology for gathering data on food consumption and occurrence of gastrointestinal illness, and to use that technology to assess recall and bias associated with recollection of food consumed [39]. This study employed the smartphone app Ethica – details of which were presented in Chapter 2 under Section 2.3).

University students were recruited to install and utilize the app over a multi-month period, and to provide feedback on its usability. The selection of students were chosen as the study population was shaped by the fact that many are learning to cook or eat out frequently, elevating their risk for foodborne illness. Moreover, most have smartphones and frequently use them for data sharing and communication.

For ease of enrollment and management, the initial larger group of participants were divided into two staggered groups, and for each group a separate study was assigned (ethica_study_84 and ethica_study_85). The first group consisted of 40 participants that completed data collection between January 15 and March 27, 2016. The second group consisted of 39 participants that completed the study between January 22 and April 4, 2016. Over a period of 10 weeks, participants of each group were asked to report any gastrointestinal symptoms using the app. During the first 10 days, they were further asked to take photos of their meals and to give a short description of them (either by writing or recording their voice) at their convenience. In a daily and weekly manner, and at specific times of the day, micro surveys containing a few questions about participant's food intake were sent out to the participants through the app. Completion of such surveys had been encouraged at study intake. Reflecting the interest in the accuracy of recollection of food consumption, participants completed an online survey 2 weeks later asking them to recall their food consumption history for days 4 to 10.

The main objective of the project was to measure the extent of participant recall bias and the resulting limitation of current investigation strategies [39]. However, the abundance and value of the collected data over the 10 weeks participation period invite closer and multiperspective investigation. In this chapter, we demonstrated how the core of the data using tools such as the Spark framework can be accessed to derive findings by running analysis procedures over the data. Primary data giving information regarding human

behaviour often provides opportunities to either derive the value of parameters within a simulation model or to estimate – via calibration or machine learning techniques such as Particle Markov Chain Monte Carlo (PMCMC) – values for parameters that allow the model to best match such data. Investigating high volume data (big data) such as the data collected in the project can open new horizons to a researcher and reveal some information regarding the surrounding environments which might be hard to detect directly by visual and intuitional observations. This kind of information can be directly provided to ground the simulation models in a more explicit and accurate way.

The study contained two types of survey-based data collecting regimes: User triggered surveys and time triggered ones:

- User Triggered Surveys
  - Food Consumption
  - Illness Reporting
- Time Triggered Micro-surveys

Figure 5.1 and Figure 5.2 show how users could submit a report of their illness and food consumption behavior, respectively.

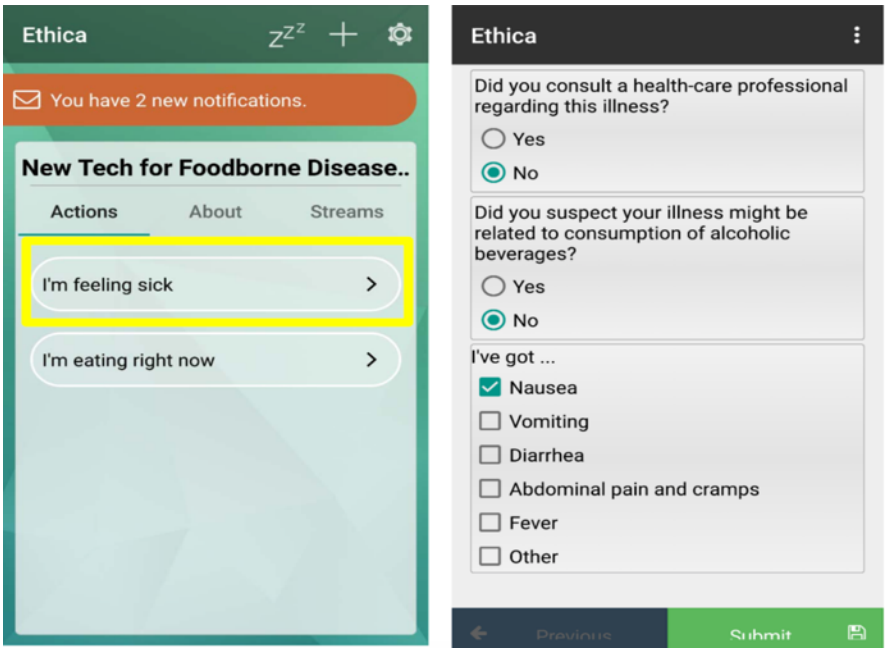


Figure 5.1: Illness Survey used in *ethica_study_84* and *ethica_study_85*.

For the first 10 days of the study, the participants were also asked to answer a set of questions on surveys appearing on their phones around their meal time, a construct termed triggered micro-surveys. These survey

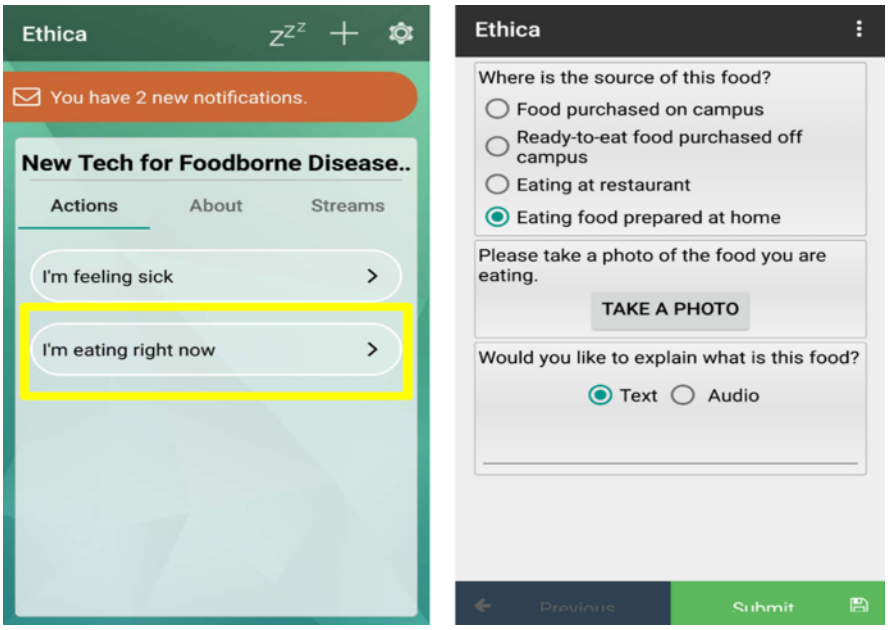


Figure 5.2: Food consumption Survey used in EthicaStudy84 and EthicaStudy85

triggering times were designed so as to correspond to typical breakfast, lunch and dinner times. There were a large number of different micro-surveys included in the study. Figure 5.3 shows a sample micro-survey.

5.2 Data Structure

Since studies in Ethica usually involve collecting one or more automated data sources – such as step counts, screen state, accelerometer and GPS – and given that a large group of participants might be enrolled in the study, researchers will generally be dealing with a large amounts of data, often making it challenging to access and analyze it with traditional instruments. Ethica provides two methods of accessing the raw data: Either by downloading study data for a particular data source as a comma-separated values (CSV) or JavaScript Object Notation (JSON) through a Web—based access, or connecting directly to a study database. More recently, Ethica has also added support for a data access and visualization tool based around Apache Kibana; this Ethica extension is not covered here.

Since our analysis includes a broad set of available data – and potentially cross-linking different types of data – downloading the raw data was not a good fit for our needs. The data cultivated under the study "Exploring New Technologies to Support Investigation of Foodborne Disease" [2] by means of Ethica Data app was saved into a Cassandra Database 2.6. It is saved into two separate databases, each of them covering one of the two cohorts of the study that – as mentioned earlier in this chapter – were enrolled a week apart for ease of managing enrollment process.

Figure 5.4 shows the tables for this study holding the data collected from users' phones.

Following is an example of a single data entry stored in survey_resp table using the Cassandra Query

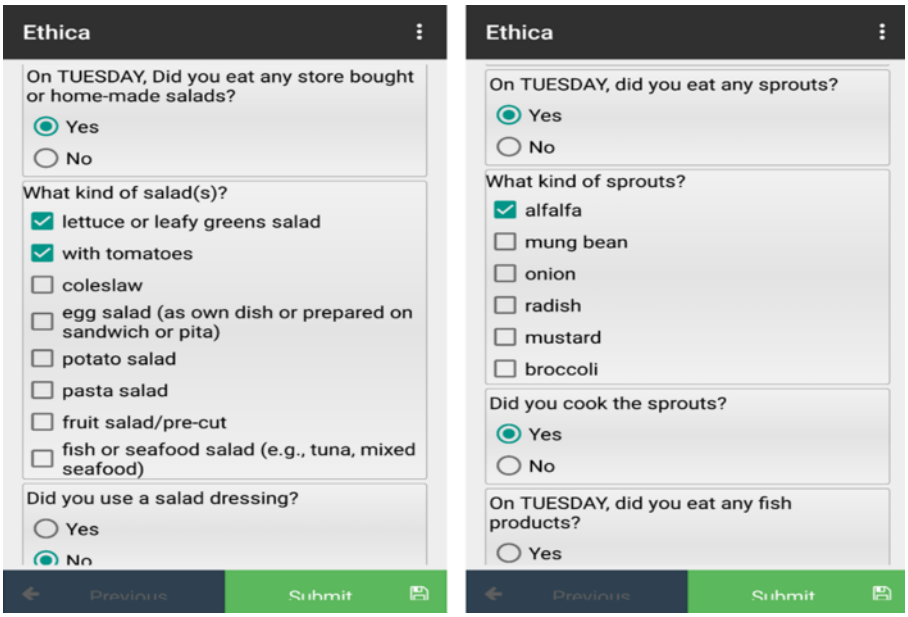


Figure 5.3: A micro-survey used in the study.

survey_resp	wifi	survey_responses_sid	gyroscope
persisted_questions	battery	linear_accel	gps
record_count	survey_responses	accel_variability	
orientation	gravity	battery_events	
accelerometer	survey_resp_sid	persisted_responses	

Figure 5.4: A snapshot of the tables in the *ethica_study_84*, each corresponding to a Sensory dataset.

Language (CQL). Please note that the "resps" column consists of three separate questions stored along with the participant-specific answers in JSON format.

```
user_id | survey_id | subsurvey_id | record_time | rev_no | device_id | \\
duration | resps
---------+-----------+--------------+---------------------------------+--
000 | 111 | 1 | 201*-**-** **:**:**| ****053225741 | 83b7044351f9b19e
| 14 | {1: 'Q/A Set 1', 2: 'Q/A Set 2', 3: 'Q/A Set 3'}
```

The JSON format of these Q/A sets have been extracted and are shown in a readable fashion at Appendix A.1. This example which is a single data entry of the illness reporting survey shows how it is structured within the Cassandra database. The column 'resps' contains information by the participant in response to questions. For the sake of the analysis considered here, notable answers include those reporting illness symptoms, whether the illness might be rooted in alcohol consumption (for excluding it from foodborne illness dataset), and whether the user has consulted with a health-care professional (as a sign of the illness severity).

5.3 Problem Definition

The areas of focus to be pursued in this section are as follows:

1. Food Source Diversity
2. Clinical Presentation Frequency
3. Novelty and breadth in food seeking
4. Capacity to recall
   (a) To avoid eating out
   (b) To avoid going to recent restaurants

In Appendix A.2 the steps of setting a Spark environment and connecting to the data repository via a Cassandra connector package is described. As mentioned above, the survey data were stored in the database in the form of JSON encoding. According to the study design – which involved two distinct waves of enrollment, separated by a week – the data of interest for this analysis were located on two separate databases with the name format of `ethica_study_xx`. For ease of calculation, the tables containing eating and illness report surveys from these two separate databases were aggregated. Please note that – as mentioned in Chapter 5 and Section 5.1 – Ethica's optional expiration time for surveys and capacity for such surveys to be cancelled had resulted in some responses to be filter out before any further inspection. The steps of aggregation and filtration is presented at Appendix A.3 in greater detail.

5.3.1 Food Source Diversity

One of the questions asked of participants through the food reporting survey within Ethica concerned the source of the reported food. The user was allowed to provide one of four options: "Food purchased on campus", "Restaurant Food", "Ready-to-eat food purchased off campus" or "Home food". Such the categories were chosen as to be identical to the categories in the PHAC outbreak investigation surveys. Here, we present our findings on the per-participant portions of such food types reported. Moreover, the daily consumption frequency of each of these food types per user are shown. Hereafter, the following acronyms are used to refer to the food source types:

- "Food purchased on campus" --→ "CampusF"
- "Eating at restaurant" --→ "RestaurantF"
- "Ready-to-eat food purchased off campus" --→ ''R2EF_PurchOffCamp''
- "Eating food prepared at home" --→ "HomeF"

The following table shows a few records of the dataframe achieved from decoding all the required information from the food reporting survey. The step-by-step description of acquiring such a dataframe is presented in Appendix A.4.

```
+-------+---------+----------------------+--------+--------
|user\_id|survey_id|record_time            |foodType        |
+-------+---------+----------------------+--------+--------
|XXX    |***      |2016-01-17 11:53:45.741|HomeF            |
|YYY    |***      |2016-01-19 17:29:22.321|R2EF_PurchOffCamp|
|YYY    |***      |2016-01-19 21:31:17.284|CampusF          |
+-------+---------+----------------------+--------+--------|
```

Figure 5.5 show the fraction of food categories consumed by each participant where each radius points to a user's data. It clearly shows that for most of the participants have had more tendency to consume home cooked food. The snippet of code for extracting the data for plotting these graphs is presented at Appendix A.4.1.

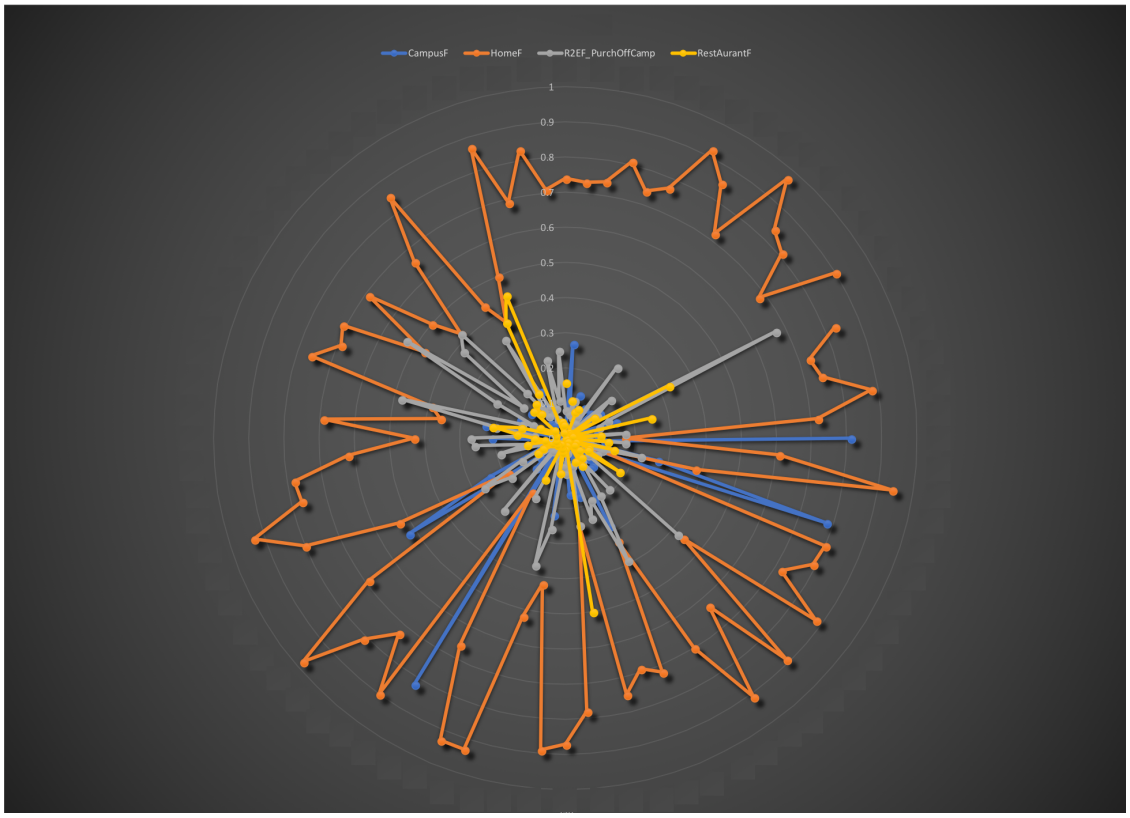


Figure 5.5: Food type consumption fraction per user for total participants

5.3.2 Clinical Presentation Frequency

Given that someone is sick, what is the frequency of clinical presentation? Participants who had reported cases of illness were also asked to specify whether or not they had consulted any health-care professional regarding their illness. A high level review of the collected illness reports by extracting participants' answers to this question revealed that only 3.5% of the reported cases of illness had led to a clinical presentation. Before taking any further steps, we had to make sure that only reports relating to gastrointestinal disease were considered. The analysis therefore excluded any report where the participant had reported "Other" (selected from among a set of foodborne illness symptoms) as the single symptom of illness, or claimed that the illness might be due to alcohol consumption. A high level preview of illness reporting survey data is shown in Figure 5.6.

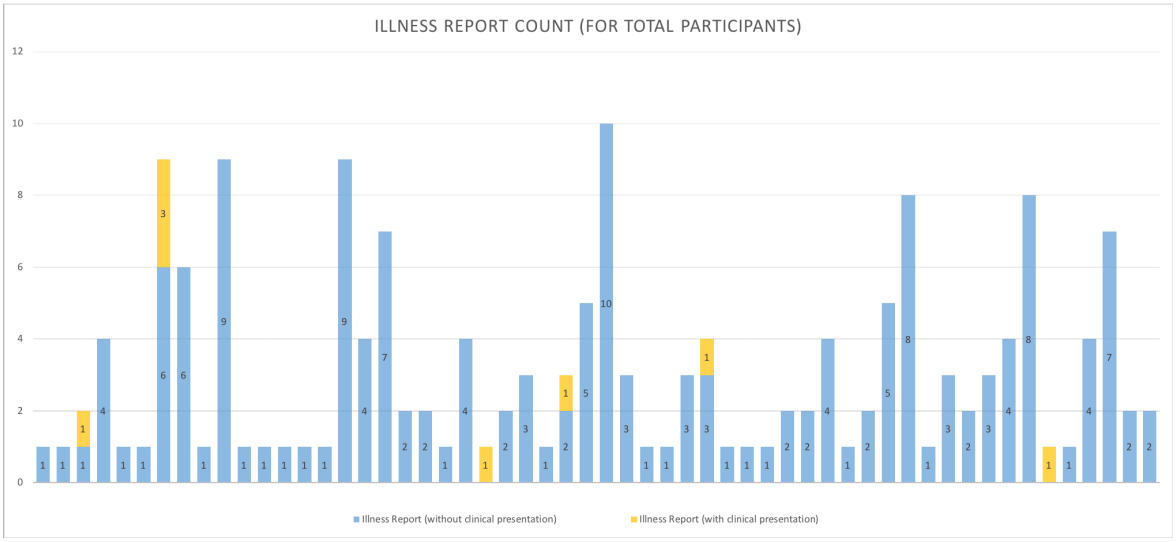


Figure 5.6: Illness Reporting Statistics

This chart shows the information on the reports of clinical presentation of participants against the background of their total illness reports where each of the columns belong to a single participant. As shown a minor fraction of the cases had resulted in clinical presentation.

Down the road, we also sought to find cases of illness reports for each user that have been sent out consecutively in a time window of 24 hours (and in another attempt, 48 hours) and presumably suggesting a common case of illness which had entailed a care-seeking.

The mechanism to find the time lag between each report of illness and the preceding instance of report is given in detail at Appendix A.5.1. It should be noted that all of the records being sent out a few seconds after a previous report (which most likely were related to participants' mistake in data entry) were excluded from the results.

Findings showed that given an illness incidence reported by any of the participants, all other reports in

the next 24 or even 48 hours (hypothetically pointing to a common case of illness) have not resulted in any clinical presentation afterwards.

5.3.3 Novelty in food seeking

To what degree are the participants returning to the same restaurants vs. going to others? Since participants had never been required to report the name of the restaurants at which they had dined, we had to find a way to extract geo-location data out of their food consumption surveys. Unfortunately, in this study, the surveys were not geo-tagged i.e., no GPS location measurement was automatically captured with the survey submission; newer versions of Ethica offer a capacity to tag the surveys with the geo-location information of the phone at the time of survey submission.

Building on the provisional assumption that the food consumption reports regarding restaurant foods had been submitted at the scene, we set up a procedure for cross-linking submitted reports with contemporaneous records from GPS table – which logs the location and time information – based on the extracted record time from the reports.

There were several major issues in applying this method. The first reflected the fact that there might be several cases where, for a report record time, there exist no GPS record; this is mostly because location data collection in Ethica app is occurring with a specific sample rate, which might not lead to a sampling interval overlapping with the survey record time; in other words, there might be cases where for a survey there is no corresponding GPS data at that specific record time. Within this context, it bears noting that record time resolution is in milliseconds.

A second major concern was that there are moments in which no GPS signal is present at the scene. To overcome this problem, instead of looking into one single GPS record time, we could look up all the GPS records within a window corresponding to the food survey report, and then get the average of the latitude and longitude of those selected datapoints. A time window of one minute length was selected for capturing the location information.
By intersecting food consumption reporting and GPS tables, each report became marked with the latitude and longitude of the phone at the time of submission; of them, only those relating to reported consumption of restaurant food were selected.

It is worth bearing in mind that, due to noise, the GPS sensors on cellular phones might record location coordinates a bit off from the exact location of the phone (participant). To address this, a binning process for grouping pairs of locations based on their distance from one another was performed. Square bins and hexagonal bins are the most common for spatial binning and in our case, we used the first method.

Uniqueness fraction is the number of restaurant-sourced eating reports in unique places (unique restaurants) to the total number of eating reports at any restaurant. A scatter plot of such a data is also presented in Figure 5.7.

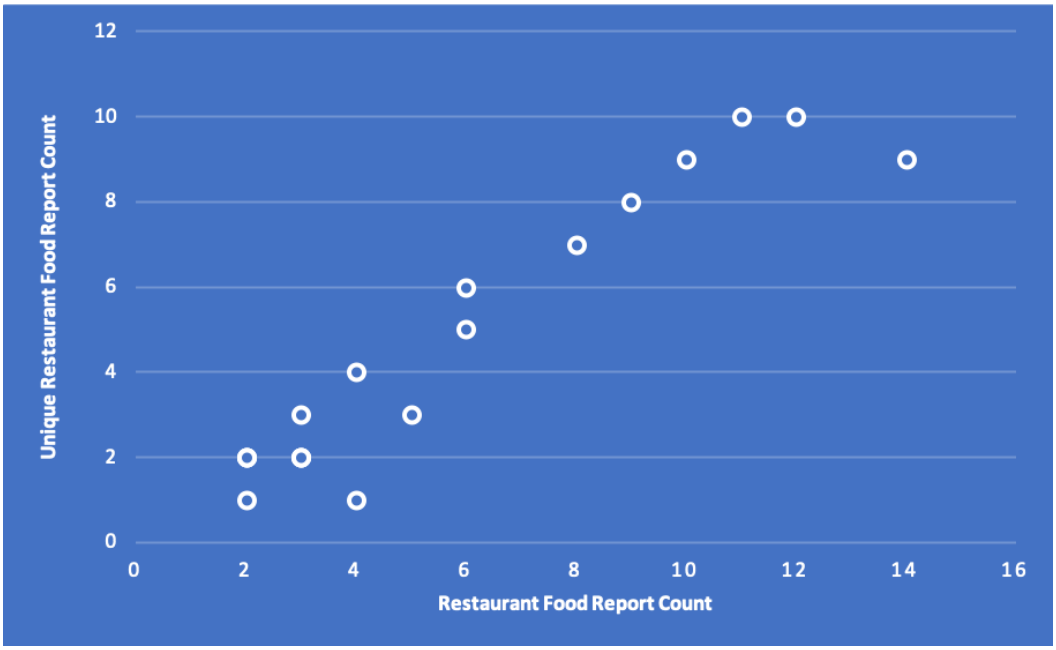


Figure 5.7: Unique restaurant food report counts vs. total restaurant food report counts

5.3.4 Capacity to recall

What are the potential impacts of foodborne illness on eating behavior of affected individuals? The study of change in participants' food consumption behavior when they had a recent foodborne illness experience was the point of focus in this section. I hypothesized that this behaviour change may manifest itself in several ways. Given occurrence of foodborne illness on the part of a participant, a short term impact of might precipitate avoidance in consuming foods prepared at campus vendors or restaurant foods.

A medium- or long- term impact might be observed in the participant's avoidance of eating food in recent restaurants. The following subsections describe my investigation and findings with respect to the validity of these hypotheses.

Post-Illness Avoidance of Eating Out

It is possible that, in many cases, a person who has suffered a foodborne illness will lack a sense as to be where they got sick.

Question: *Is there a statistically significant decrease in the per-day probability of eating out (vs. eating elsewhere) in the day following an illness report?*

To answer this question, we investigated the record times of illness reports and inspected any food consumption behaviour on the part of the participant in a 24-hour span following each such report. Combining records of foodborne illness reports derived in work characterized in Section 5.3.2 and food consumption reports whose derivation was described in Section 5.3.1, the food consumption dataframe was intersected with the illness report dataframe to find instances of food consumption reports that fell into a defined time

span likely to be characterized by illness recovery. This span extended from the record time of the illness report until 24 hours after it. The data preparation and process of intersecting data frames are explained in detail at Appendix A.6.1.

To assess whether occurrence of illness reduced the tendency of eating food prepared by vendors, we sought to find the fraction of total reported food eaten in the period of 24 hours after an illness report that consisted of restaurant food and food purchased on campus. Subsequently we sought to compare these values to the total risky food consumption during the study period for that participant. The pseudo-code of such an operation is explained in detail at Appendix A.6.2

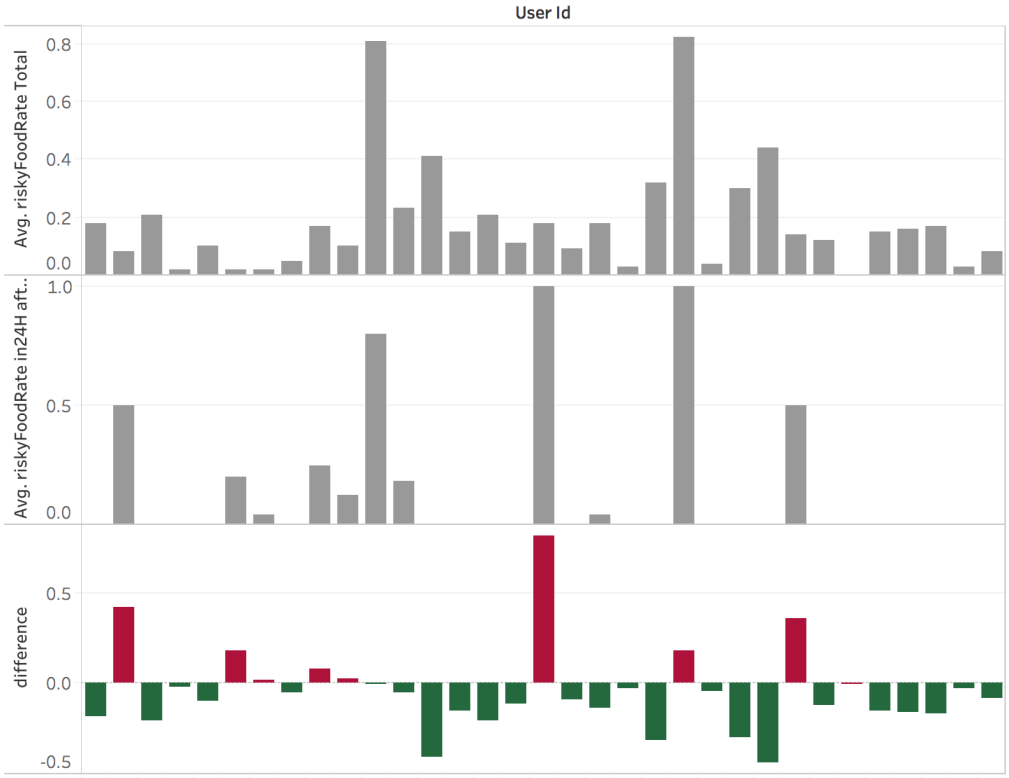


Figure 5.8: Risky Food Consumption: (Top) Average fraction of risky food consumption throughout study. (Middle) Average fraction of risky food consumption within 24 hours after illness report. (Bottom) Difference of the top charts – Bars in green show decrease in risky food consumption. All data is shown on a per-participant basis

Among all those cases who had submitted a minimum of one food consumption survey within a 24-hour time slot, the change in their vendor-prepared food consumption fraction during this period to their total risky food consumption fraction is shown in Figure 5.8. It is worth noting that the decrease in reporting might come not from a deliberate avoidance of vendor-related food consumption, but instead from a tendency to stay home while recovering.

Post-Illness Avoidance of Recent Restaurants

To assess whether the dataset suggests a reduction in behaviour of participants in showing up in recently visited restaurants after an illness report, a length of time before that incidence and after it, and to then compare the representation of food vendors in each of the corresponding windows of time was considered. Since the study had not asked the participants to provide names of food vendors at which they had dined, it was necessary to compare geographic information derived for the food-consumption reports during those time windows. Specifically, eating behaviour for vendor-prepared food across one-week windows before and after illness reports was selected.

Similar to the mechanism described earlier at Section 5.3.3 and also previous subsection, the eating survey dataframe was first joined with the GPS dataframe in a one-minute window fashion to provide geographic context to the vendor-related food eating surveys.

Following the step of lending geographic context to the food consumption reports, the obtained geo-tagged food data was linked with illness report data in order to find locations at which food consumption was reported within the a week before and after any illness incidence reported by each participant. The details of such operations are expressed at Appendix A.6.2.

The results show that there are only 5 participants having reports of restaurant food eating in a time length of one week before and after a case of illness report. Table 5.9 shows the order of eating and illness reports for these 5 participants.

user-id								
1	eating 2016-01-16 8:31 (X_0, Y_0)	illness 2016-01-16 16:42	illness 2016-01-17 13:57	eating 2016-01-17 20:41 (X_0, Y_0)	illness 2016-01-19 9:07			
2	eating 2016-02-09 16:47 (X_00, Y_00)	eating 2016-02-09 16:53 (X_00, Y_00)	eating 2016-02-10 15:09 (X_11, Y_11)	eating 2016-02-10 21:12 (X_22, Y_22)	eating 2016-02-10 21:41 (X_33, Y_33)	eating 2016-02-11 18:34 (X_44, Y_44)	illness 2016-02-11 19:44	eating 2016-02-12 12:34 (X_55, Y_55)
3	eating 2016-02-15 8:06 (X_000, Y_000)	illness 2016-02-18 11:16	eating 2016-02-23 7:13 (X_111, Y_111)					
4	eating 2016-01-26 18:35 (X_0000, Y_0000)	illness 2016-01-28 0:35	eating 2016-01-29 20:55 (X_1111, Y_1111)	eating 2016-01-31 21:25 (X_2222, Y_2222)				
5	eating 2016-02-11 19:16 (X_00000, Y_00000)	illness 2016-02-15 13:01	illness 2016-02-16 16:37	eating 2016-02-17 12:38 (X_11111, Y_11111)	eating 2016-02-21 13:28 (X_22222, Y_22222)			

Figure 5.9: Food reporting locations before and after an illness report for different participants

As shown in this table, participant with user_id=1 is the only one who has dined in the same location before and after her illness report. For all others, the location before and after the illness report are different.

Figure 5.10 shows the distance between each of the assumed restaurant locations before illness report with the ones after this report.

5.3.5 Conclusion

The findings presented confirm that Apache Spark can be an effective manipulation and analysis tool for large amounts of data in a cluster computing platform. In addition, Ethica and similar tools demonstrate a

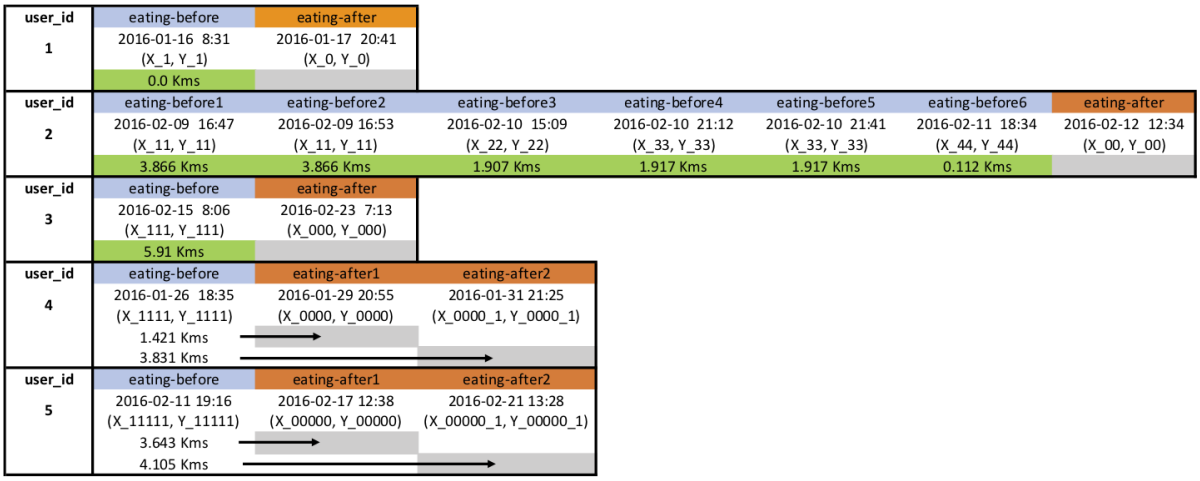


Figure 5.10: Distance between restaurant locations before and after illness report

promise in collecting information, given the great majority of foodborne illness cases that remain clinically unrevealed. This collected information can be used by public health organizations and centers in planning health protection plans in a timely fashion without imposing huge economic burden. The results also suggest that a considerable number of the participants appear to engage in a pattern of novelty seeking in their food seeking behaviour rather than returning to same restaurants. The data also suggest that a majority of affected individuals' consumption of risky foods declined in a 24-hour time span after an illness incident. The fact that the participant population of this project is limited to university students shapes the observed patterns in central ways, including in terms of their limited access to diverse food vendors, avoidance of revisits to recent restaurants after illness was not statistically (in terms of distance) significant. Given the role of places like food courts on the university campus that host multiple restaurants, distance is not likely to represent a robust way of assessing the degree of participants' avoidance from recently visited restaurants.

Chapter 6

Cough Detection

Respiratory infections and chronic respiratory diseases impose a heavy health burden worldwide. Coughing is one of the most common symptoms of many such infections, and can be indicative of flare-ups of chronic respiratory diseases. Whether at a clinical or public health level, the capacity to identify bouts of coughing can aid understanding of population and individual health status. Developing health monitoring models in the context of respiratory diseases and also seasonal diseases with symptoms such as cough has the potential to improve quality of life, help clinicians and public health authorities with their decisions and decrease the cost of health services. In this paper, we investigated the ability to which a simple machine learning approach in the form of Hidden Markov Models (HMMs) could be used to classify different states of coughing using univariate (with a single energy band as the input feature) and multivariate (with a multiple energy band as the input features) binned time series using both of cough data. We further used the model to distinguish cough events from other events and environmental noise. Our Hidden Markov algorithm achieved 92% AUR (Area Under Receiver Operating Characteristic Curve) in classifying coughing events in noisy environments. Moreover, comparison of univariate with multivariate HMMs suggest a high accuracy of multivariate HMMs for cough event classifications.

6.1 Introduction

Symptoms such as cough are important clinical signs. Coughing is the most common symptom in respiratory diseases, and awareness of the occurrence or persistent presence of a cough can provide valuable information to physicians. Detailed awareness of coughing can aid physicians with their treatment on the basis of quantitative assessments such as frequency or intensity as well as qualitative assessments such as dry or wet coughs [42]. Moreover, cough detection analysis has the potential to reduce the cost of health services by – for example – detecting the early signs of diseases and making preemptive diagnosis possible and prescribing basic treatments while they are still effective [22]. However, the benefits of securing reliable, and timely quantification of coughing behavior can also offer benefits beyond the physician's office. Collecting cough data using monitoring devices such as mobile sensors or other devices and analyzing the audio signals of coughs can support remote monitoring of patients with chronic respiratory illnesses or restricted mobility. For such diseases, awareness of flare-ups of coughing can motivate the need to present for care, and can inspire changes

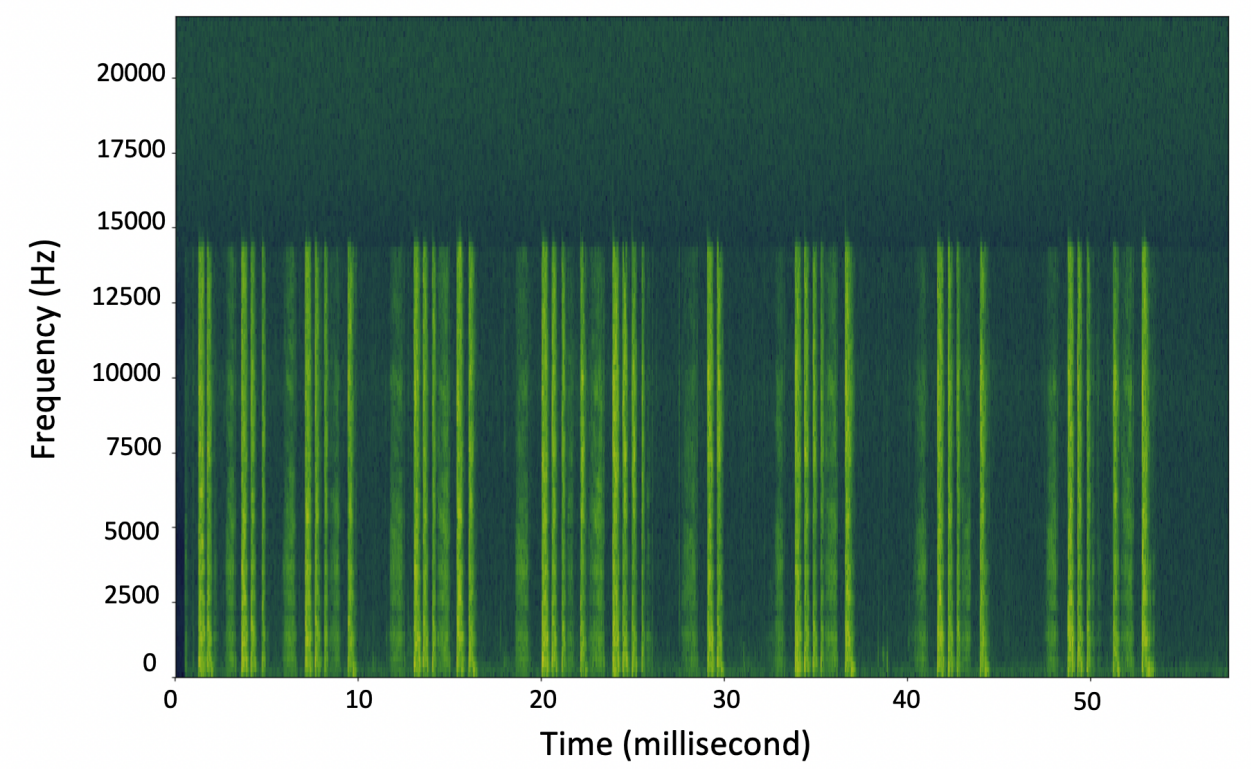


Figure 6.1: A spectrogram of a sample cough

to treatment recommendations. A final and important advantage of cough recognition resides in its potential to provide health authorities with timely surveillance information about emergence of high-burden respiratory conditions, thereby supporting earlier outbreak identification in particular geographic areas, thereby better supporting public health decision making, including the design of public health interventions.

The duration of a cough sound typically varies between 0.2 and 1 second [20], and exhibits a sequence of distinct acoustic patterns. The origin of these patterns is airway narrowing and bifurcation. The airway narrowing is due to a change in the thickness of the airflow walls (inflammation, mucus collection, bronchoconstriction and fibrosis). A typical cough sound usually is composed of three stages: an explosive expiration due to the abrupt opening of glottis, the intermediate stage in which cough sounds are reduced, and the voiced stage due to the closing of the vocal cord. There are a variety of patterns of coughing based on the presence or absence of each of these stages [30].

A visual representation of the spectrum of frequencies of a cough signal as it varies over time is shown in the spectrogram of Figure 6.1, which is depicted as a heat map, with the lowest and highest intensities being represented by dark and light green, respectively.

Several studies have described methods to analyze cough characteristics, considering the subjective interpretation of cough sound recordings and the analysis of spectrograms [21, 7, 13, 31, 46, 48]. There are two main research streams for cough recognition. One stream investigates audio signals frame-by-frame and combines consecutive cough frames as a cough event [26]. The second stream consists of event detection and

cough classification steps. Event detection identifies cough event candidates; each candidate is then classified as a cough or non-cough event [25]. Our work follows the first stream, by seeking to detect cough signals in continuous audio recording using a Hidden Markov Model (HMM).

This paper investigated the performance of an HMM, where each state of that model corresponds to a portion of a typical cough, and where observables represent summaries of information from sound profiles. We further investigated the performance of the model in detecting each state and thus distinguishing a period of time in which a cough was occurring from when it was not. The HMM could further be used to distinguish coughing from non–coughing behaviour when considering a longer period of time, and when the main focus is to identify bouts of cough present in an sound recording events. To achieve this, the acoustic energy was selected as the observable and measurable feature to feed into a univariate HMM. In another attempt, using the frequency or pitch of the sound, the energy spectrum as the observation input was split into a vector of three sub-features as low, mid and high energy bands. Finally, we compared the performance of these two scenarios.

6.2 Materials and Methods

6.2.1 Data Collection and Labeling

The cough data used in this article is collected from recordings of cough sounds from different individuals associated with the Computational Epidemiology and Public Health Informatics Laboratory in the Department of Computer Science at the University of Saskatchewan in relatively noisy environments. A duration of 20 minutes of such cough sounds were manually annotated by the authors.

We divided each audio signal into 25 millisecond time slots (bins) and extracted the following information from each bin: the time corresponding to the mid–point of each bin, the sum of the energy density of frequencies under 2 KHz (low-band energy), the sum of the energy density of frequencies between 2 KHz and 4 KHz (mid-band energy) and – finally – the sum of the energy density of frequencies between 4 KHz and 22 KHz (high-band energy). In light of the limited span of the audio frequency range, no frequencies above 22 KHz were considered. We considered the sum of energy densities as our training features for the Hidden Markov model. In this work, each cough recording was divided into five distinct states/stages, and each 25 millisecond time bin was labeled as to the state with which it was associated. Specifically, we considered three states inside a single cough (states A, B and C), a brief state of silence between each cough inside a bout of coughs (D) and a longer state of silence between bouts of coughs for cough-prone cases (E). Bouts of coughing were considered to trigger additional coughing (thus returning from state D to state A) with higher probability than in a general non-coughing state (state E); alternatively, a bout of coughing could then end, via a transition to state E. Figure 6.2 depicts different coughing states in the time domain. B contrast, a schematic diagram showing posited transitions between different coughing states is demonstrated in Figure 6.3.

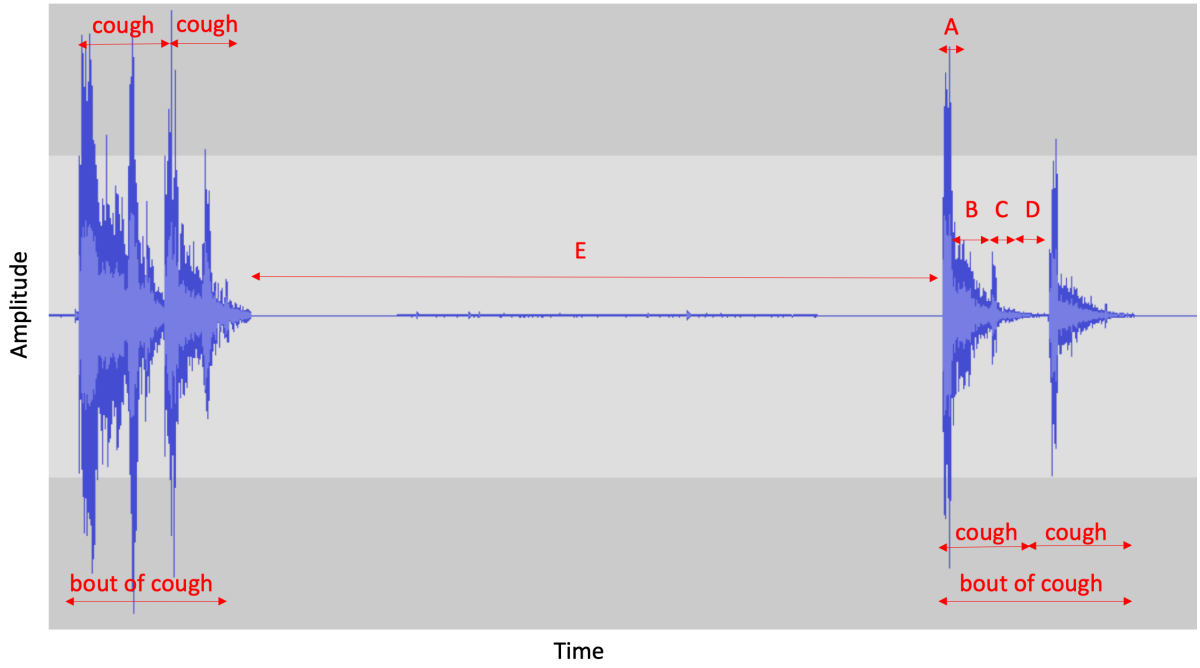


Figure 6.2: Different states of coughing in an acoustic signal of four cough epochs

The length of the cough sounds vary from cough-to-cough, and the distinctions between the successive stages are not always clear – leading to imprecision in human classification of such stages. The beginning of the cough sound was used as the starting point of state A, the start of state B was selected when the sound amplitude was significantly lower than the initial peak and the start of state C was chosen when there was a rise in the sound amplitude after state B.

This work sought to investigate the effectiveness of an HMM in predicting the underlying state of a given time interval of a cough–recording by feeding our model with low, mid and high band energy-density values. Given the characteristics of a single 25 millisecond bin and the energy density values, we investigated the capacity of that model to predict with which state of coughing this bin was associated.

6.2.2 Model Training

The calculated probability for each hidden state is obtained by multiplying two values; one inferred from the observation i.e., the likelihood of observing that hidden state given the current observation vector and the other one derived from the transition matrix – i.e., the probability of being in that specific state according to the probability of having been in different states in the previous time bin. The initial states' values, i.e., the probability of being in any of the hidden states in the initialization step was set assuming that the model starts in state A with the probability of 1.

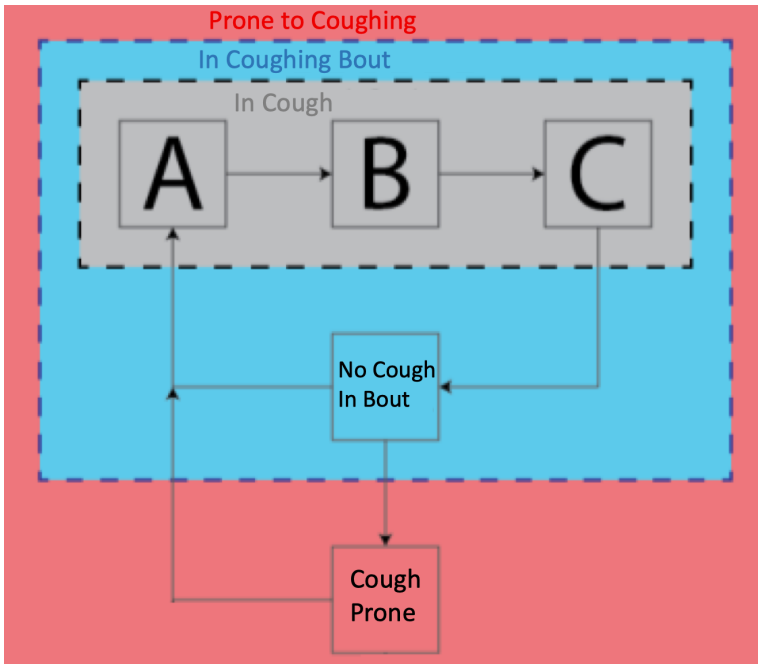


Figure 6.3: Cough transitions captured in the HMM

6.2.3 Model evaluation

We employed a two-fold cross validation approach in a repeated manner for training our model and and used the average AUC – Area Under the Receiver Operating Characteristic [ROC] Curve – of the cross validation steps as the primary evaluation metrics. The confusion matrix, sensitivity, and specificity were considered to further evaluate model performance.

Since the ultimate goal is of this work to classify Cough from Non-Cough (correctly identifying an epoch of cough in a bout of coughs) or Coughing from Non-Coughing (correctly identifying a bout of coughs), we further investigated the capacity to classify audio signals according to two dichotomous categories: Cough vs. Non-Cough, and Coughing vs. Non-Coughing. To accomplish this, we grouped the states in binary format as follows:

- **Cough vs. Non-Cough**: states A, B and C were grouped in a single state of Cough and states D and E as a single state of Non-Cough
- **Coughing vs. Non-Coughing**: States A, B, C and D were grouped as sate of Coughing and E as the state of Non-Coughing.

The details of the preferred classifier will differ depending on our goals. Here, we applied the Youden's index [50] (by applying the "best" argument of the "coords" method from pROC package [37]) to maximize the sum of sensitivity and specificity. The Confusion matrix and the optimal accuracy, sensitivity and specificity are demonstrated in Section 6.3.

Transition and Emission Matrices

Table 6.1 shows a sample of data points extracted from cough signals. The HMM states and transitions captured the posited structure of transition ins between cough stages as shown in Figure 6.3.

Table 6.1: Training data sample

Ground Truth Label	Low–band energy	Mid–band energy	High–band energy
A	31855.85	1155.99	678.39
B	5630.51	895.47	1704.09
B	9672.26	1891.19	1126.83
C	371.24	8.47	2.07
D	189.62	6.65	1.22
E	3.12	0.39	0.06
E	2.16	0.15	0.05
E	1.13	0.10	0.02

At any given time bin, the HMM can be in one of the five (hidden) states of A, B, C, D or E, resulting in the transition matrix shown as Table 6.2. It bears emphasis that there are no transitions between some pairs of states – for example, from A to C, or A to D); the probability of such transitions was treated as zero.

Table 6.2: Transition table for sample data

	A	B	C	D	E
A	$P_{A\|A}$	$P_{A\|B}$	0.0	0.0	0.0
B	0.0	$P_{B\|B}$	$P_{B\|C}$	0.0	0.0
C	0.0	0.0	$P_{C\|C}$	$P_{C\|D}$	0.0
D	$P_{D\|A}$	0.0	0.0	$P_{D\|D}$	$P_{D\|E}$
E	$P_{E\|A}$	0.0	0.0	0.0	$P_{E\|E}$

To calculate the probability P_{xy} of transition from a current state x to any of the probable states y, we first found the probability of leaving a given state to any destination. Based on the HMM assumption of memoryless transition processes, this is given by the reciprocal of the mean residence time (in time bins) within that state. For states exhibiting a single outgoing transition (states A, B, C and E), that probability was employed directly. For state D (which can be followed by either state A and state E), to arrive at the probability of making the transition to each of states A and E, we further multiplied the probability of leaving the state by the empirically observed proportion of transitions from state D to states A and E, respectively.

Since the model in this work makes use of continuous observations, instead of having an emission matrix,

we used density functions extracted from and fitted to empirical observations, where the observations are assumed to be independent from each other, conditional on being in a given state. As a simplifying assumption, the joint likelihood of observing a given vector of low-band, mid-band and high-band energy quantities was approximated as the product of independent likelihood functions (each associated with a univariate probability density function). For a case of univariate HMM where a single observation (i.e., the total energy inside each bin), for any given state, only one empirical density function was defined.

6.3 Results

Two experiments were conducted using the HMM. Experiment A trained and evaluated a univariate HMM considering just a single feature: the total energy in a time-binned audio signal. By contrast, in Experiment B, all the three band of energies were considered as a vector of observations, and a multivariate HMM was trained. Both experiments used the "mhsmm" package in the statistical software R. Both Experiments evaluated the HMMs according to ability to classify, for a given time bin, the particular coughing state as well as dichotomous classification regarding the presence of absence of coughing.

6.3.1 Results of the univariate HMM: Experiment A

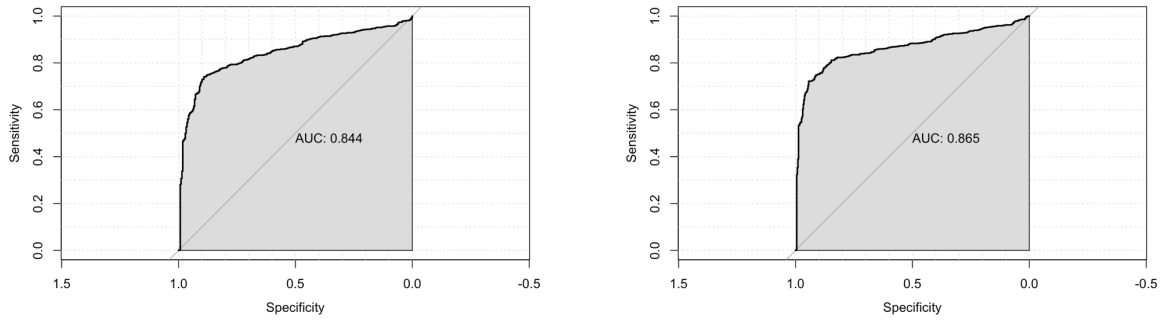
Using the total energy in bins as the single feature, an AUC value of 0.751 and 0.744 was obtained for training and testing sets, respectively. The performance statistics of the model over the testing set – including a confusion matrix, sensitivity, specificity, and accuracy – is shown in Table 6.3. Performance statistics of the testing set for the univariate HMM in cough/nocough and cough-ing/nocoughing classification mode is shown in Table 6.4.

Table 6.3: Performance statistics of the testing set for univariate HMM

		Observed				
		Class: A	Class: B	Class: C	Class: D	Class: E
Predicted	Class: A	31	6	1	1	7
	Class: B	3	45	19	6	25
	Class: C	3	17	29	4	5
	Class: D	3	2	31	21	19
	Class: E	13	9	65	84	714
Sensitivity		0.585	0.570	0.200	0.181	0.927
Specificity		0.986	0.951	0.971	0.947	0.565
Accuracy		0.722				

Table 6.4: Performance statistics of the testing set for the univariate HMM in cough/no_cough and coughing/no_coughing classification mode

	Identifying a cough epoch in bout of coughs		Identifying a bout of coughs	
	Observed		Observed	
	cough	no–cough	coughing	no–coughing
Predicted cough(ing)	247	230	371	214
Predicted no-cough(ing)	30	656	22	556
	Accuracy: 78%		Accuracy: 80%	
	Sensitivity: 89%		Sensitivity: 94%	
	Specificity: 74%		Specificity: 72%	



(a) Cough/Non_cough; AUC 0.844 **(b)** Coughing/Non_coughing; AUC 0.865

Figure 6.4: ROC curve for uni-variate HMM after grouping

To investigate the obtained models performance in classifying Cough from Non-Cough or Coughing from Non-Coughing, the identified states were grouped as per the process discussed in Section 6.2.3 resulting in the following ROC curves shown in Figure 6.4 for Cough/Non-Cough and Coughing/Non-Coughing classifications.

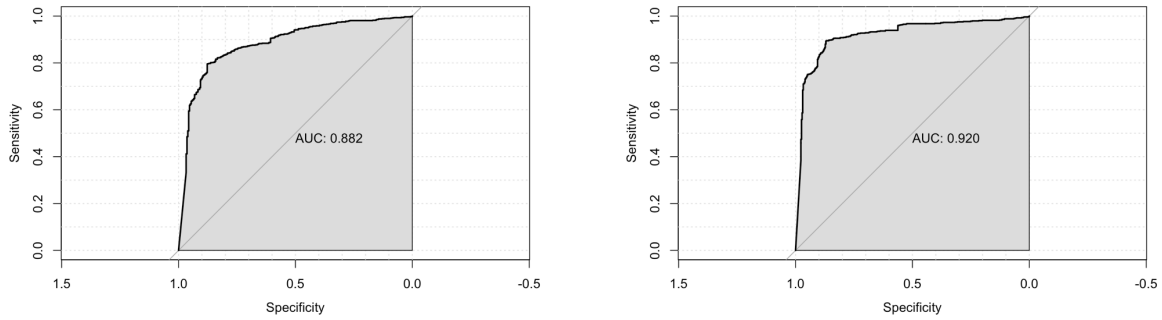
6.3.2 Multivariate HMM Results: Experiment B

The multivariate HMM trained with a vector of three features containing the acoustic energy in low, medium and high bands improved by 6% the performance of the AUC for the testing set, increasing it from 0.744 to 0.789. The AUC for training set was almost the same as for the univariate case, reaching 0.752. The performance statistics of the chosen by Youden's-index-selected multivariate model over the testing set is demonstrated at Table 6.5. Also, the results of the Cough/Non-Cough and Coughing/Non-Coughing classifications as the results of dichotomously grouping the cough states are depicted in Figure 6.5. The AUC for the cases of Cough/Non-Cough and Coughing/Non-Coughing classification were increased by 4.5% and 6.4% when compared to their univariate HMM counterparts.

Using the curves demonstrated in Figure 6.5 and to maximize both the sensitivity and specificity, the best cut-off point was calculated on which the confusion matrix and the optimal accuracy, sensitivity and specificity were obtained, according to Youden's index. Results of using the best threshold in terms of balancing the sensitivity and specificity are shown in Table 6.6.

Table 6.5: Performance statistics of the testing set for multivariate HMM

		Observed				
		Class: A	Class: B	Class: C	Class: D	Class: E
Predicted	Class: A	41	12	0	1	5
	Class: B	4	41	6	0	8
	Class: C	0	21	33	2	10
	Class: D	2	4	43	26	3
	Class: E	6	1	63	87	744
Sensitivity		0.774	0.519	0.223	0.224	0.966
Specificity		0.984	0.984	0.968	0.950	0.600
Accuracy		0.761				



(a) Cough/Non_cough; AUC 0.882 **(b)** Coughing/Non_coughing; AUC 0.920

Figure 6.5: ROC curve for multivariate HMM after grouping

6.4 Conclusion

The HMMs evaluated here demonstrated favorable results, especially when the obtained results were interpreted in terms of the dichotomously problem of distinguishing Coughs from Non_Coughs, or Coughing from Non_Coughing periods. Unsurprisingly, the results presented in this work further suggest that the multivariate HMM demonstrates classification and detection of cough events with higher accuracy than does a univariate HMM. Splitting the energy of cough sounds into three separate bands lead to density functions corresponding to each band which can provide more detailed information to the HMM. The results offer intriguing potential for early-warning outbreak detection in public areas. The prospects for applying such surveillance methods can further be boosted using mobile sensor data – such as from wearable devices and smartphones using platforms such as Ethica, particularly when coupled with transmission modeling and tools such as particle filtering [38, 23, 34]. The approach demonstrated here for cough analysis can provide a foun-

Table 6.6: Performance statistics of the testing set for multi-variate HMM in cough/no_cough and coughing/no_coughing classification mode

	Identifying a cough epoch in bout of coughs		Identifying a bout of coughs	
	Observed		Observed	
	cough	no–cough	coughing	no–coughing
Predicted cough(ing)	243	181	342	82
Predicted no-cough(ing)	34	705	51	668
	Accuracy: 82%		Accuracy: 89%	
	Sensitivity: 88%		Sensitivity: 87%	
	Specificity: 80%		Specificity: 90%	

dation towards support for both clinical research on pulmonary distress at a clinical level and for capturing patient outcomes. The expansion of the HMM model using more detailed training over diverse types of coughs can help physicians with qualitative assessment, such as in distinguishing dry or wet coughs, and inferring diseases associated with such symptoms. Another potential application of this study can be helping to identify the need for symptomatically-triggered treatment of patients suffering from respiratory diseases, particularly in patients that lack ready capacity to communicate their distress, such as in infants and young children, and among adults suffering from dementia or verbal limitations. The technique also offers potential for recognizing animal vocalization and diagnosing animal health status.

While the results presented here demonstrate much promise, the approach applied exhibits significant limitations and room for improvements. The added accuracy associated with multivariate analysis invites investigation not only into alternative bands, but also classification according to a larger number of such bands. The library of cough sounds examined here were greatly limited in their sourcing; results presented here may differ significantly for alternative coughing etiologies, and according to the pulmonary and upper-respiratory character and physical shape of the person coughing, and potentially according to cultural norms involved. Greater variety in sourcing of coughs remains a high priority. Moreover, the classification accuracy exhibited in this study needs to be considered preliminary in light of the limited library of recordings employed here; accuracy of these or other HMM may exhibit markedly different accuracy when considered on other audio recordings containing a variety of background noise or other respiratory-related sounds (e.g., wheezing, clearing of the throat). Finally, it will be important to consider examining other classifiers that provide additional avenues for predictive accuracy, including classifiers that are less theory-based, such as recurrent artificial neural networks or deep learning networks employing recurrent network structures.

Chapter 7

Conclusion & Future Work

This thesis described a novel approach for foodborne illness outbreak detection that combines agent-based simulation modeling with machine learning approaches to investigate whether this combination can better capture outbreak dynamics. Moreover, the thesis demonstrates how variants of machine learning approaches using data analysis can be applied for surveillance monitoring, predicting incident cases of ongoing outbreaks and detecting the source of contamination. Application of Hidden Markov Models (HMMs) were investigated through various scenarios. HMMs also were applied on a cough detection problem where similar to foodborne illness problem, the temporal structure of the input signals were well suited to the speciality of HMMs in reliably capturing those underlying characteristics. This chapter will provide an overview of thesis contributions and highlight potential directions for future work.

7.1 Summary of Findings

7.1.1 Simple HMM incorporated ABM

We developed a Hidden Markov model for syndromic surveillance systems considering the existence of a foodborne illness outbreak as a latent state. Comparing the performance of HMM with an context-insensitive SVM approach, we observed that HMM outperforms SVM in detection of outbreaks. We further evaluated the public health benefits obtained from the HMM with the traditional case-count-based approach. We learned that a simple HMM approach significantly improves detecting outbreak signals. Moreover, the results suggest that using smartphones for recording location and reporting foodborne illness occurrence can help with more reliable and faster detection of occurrence of an outbreak.

7.1.2 Data Analysis with Spark

We investigated foodborne illness data collected through the Ethica application to obtain information required for designing model and targeted interventions. We presented how to exploit the scalable programming language Scala using the Spark framework and Cassandra connector library to analyze big data in a performant fashioni. We further detailed the information obtained from the empirical dataset being analyzed, which captured patterns related to risk and patterns of occurrence of foodborne illness.

7.1.3 Targeted HMM

We explored application of multivariate HMMs in different scenarios on a GIS version of our municipal simulation model and we demonstrated limitations of several alternative HMM architectures.

7.1.4 Cough Detection using HMM

We showed that a multi-class HMM – which faced some limitations when applied for finding sources of contamination in the municipal simulation model – performs effectively for cough identification where the hidden states of the system have a very distinctive distributions corresponding to the observation vector.

7.2 Contributions

The contributions of this thesis can be summarized into three main areas:

1. by serving as one of the first contributions using a mobile platform such as Ethica Data for collecting behavioral data from food consumers and securing insights from analysis of such data.

2. by serving as the first study to systematically investigate the applicability of a machine learning approach for foodborne illness outbreak inference and enhancing the food safety surveillance system.

3. Finally, adapting the methodology successfully applied for foodborne outbreak detection into the respiratory disease domain, and helping with cough detection at an individual level.

The first contribution was relatively novel as it was one of the first Ethica Data applications on a human-subject research to collect data. Analysis of the associated big data was performed with cutting edge technologies to derive metrics for testing our hypotheses.

The second contribution was innovative in terms of evaluating the advantages and disadvantages between employing more frequent data samples (not-clinically-presenting foodborne illness reports) and less frequent (clinically-presenting illness reports) in a binary Hidden Markov Model for outbreak detection. We also investigated the impact of applying an articulated HMM for more accurate detection of contamination sources.

The third contribution was novel work in terms of applying HMMs for cough detection. To this end, cough sounds were studied and analyzed carefully, and common patterns were identified, which were later translated into latent states of a HMM. The obtained HMM demonstrated very favorable results, especially when the classified latent states of cough were interpreted as a dichotomously problem of distinguishing Coughs from Non_Coughs, or Coughing from Non_Coughing periods.

7.3 Future Work

There are multiple possible research directions that could improve the contributions made in this thesis. In this section, we will discuss a few of the limitations of our approach. Further studies in these areas can contribute marked enhancement to the work presented here. The limitations of this thesis can be summarized into three main areas:

1. To improve the accuracy of machine learning models, it will be valuable to expand beyond the sensor and self-reporting data to leverage the vast amount of data from social media platforms such as Twitter, Instagram and Facebook. This can lead to more reliable and higher resolution geographic-specific datasets.

2. To improve the generalizability of this analysis, it would be important to consider demographic groups other than University of Saskatchewan students, so as to avoid limitations associated collecting data on students' food consumption patterns, such as the heavy reliance of students on food court restaurants in close proximity, and limitations in student access to food sources.

3. It will further be important to combine ABM models with statistical methods such as particle filtering or particle Markov Chain Monte Carlo to estimate continuous system states and poorly evidenced parameters, and to predict model trends.

4.

In terms of redesigning the thesis, a key step could be improving the simulation model by characterizing heterogeneity among the restaurants in the model and having such heterogeneity change the visitation pattern of those restaurants. These features could include considering an average meal cost per person, categories of food provided, speed of serving customers, as well as each restaurant's distance to the consumers' residence places (something that the model currently represents, but which does not affect consumer behaviour therein). To further strengthen the model, as mentioned as one of the limitations of the work, redesigning the Ethica-based project and obtaining data from more diverse demographic groups other than University students could be very helpful in more generalizably simulating the food seeking in the ABM.

7.4 Conclusion

Our original hypothesis stated that new technologies in harvesting data in integration with machine learning approaches can improve the detection accuracy of foodborne outbreaks. In order to investigate this hypothesis, we simulated a municipal system consisting of food consumers, food providers (restaurants) and a food inspector, where a group of consumers serve as sentinels in terms of being able to report cases of their illnesses through their mobile phones. We further developed a Hidden Markov model for syndromic surveillance that

sought to classify whether an outbreak was in progress, and incorporated it into the simulation model. The favorable results revealed that significant public health gains may be secured when combining new technologies for syndromic surveillance with machine-learning based outbreak signal detection mechanisms.

We further analyzed the data obtained from a mid-sized real world project which was designed to leverage smartphones to capture food participant consumption behaviour. Results of such analyses were a seal of approval on our previous assertion that technologies such as smartphones can offer new channels for surveillance systems in data collection and better understanding of the ongoing diseases.

Using the same simulation model, we also investigated taking advantage of HMMs in detecting sources of contamination in ongoing foodborne outbreaks. While such an application, as we showed in Chapter 6 for detecting hidden states of a single cough and then inferring the existence of a cough or bout of cough in a sound record was very promising, such configuration for targeting contaminated restaurants showed some limitations. As we discussed earlier in the relevant chapter, these shortcomings may be resolved by introducing some changes in making the restaurants distinctive.

References

[1] Ethicadata. https://www.ethicadata.com/about. Online; Accessed May-2018.

[2] Exploring new technologies to support investigation of foodborne disease. https://nursing.usask.ca/news/2015/20150901newtechnologyfoodbornedisease.php. Online; Accessed Nov-2018.

[3] Support vector machine. https://en.wikipedia.org/wiki/Support_vector_machine#/media/File:Svm_max_sep_hyperplane_with_margin.png. Online; Accessed Apr-2019.

[4] Yearly food-borne illness estimates for canada. https://www.canada.ca/en/public-health/services/food-borne-illness-canada/yearly-food-borne-illness-estimates-canada.html. Online; Accessed June-2014.

[5] Canada's restaurant industry: Putting jobs and economic growth on the menu. http://www.restaurantscanada.org/wp-content/uploads/2016/07/Report_IpsosPublicOpinion_Dec2010.pdf, 2010. Online; Accessed March-2019.

[6] What is mapreduce? https://www.ibm.com/analytics/hadoop/mapreduce, 2019. Online; Accessed March-2019.

[7] Ayman A Abaza, Jeremy B Day, Jeffrey S Reynolds, Ahmed M Mahmoud, W Travis Goldsmith, Walter G McKinney, E Lee Petsonk, and David G Frazer. Classification of voluntary cough sound and airflow patterns for detecting abnormal pulmonary function. *Cough*, 5(1):8, 2009.

[8] Janet B Anderson, Thomas A Shuster, Kelee E Hansen, Alan S Levy, and Anthony Volk. A camera's view of consumer food-handling behaviors. *Journal of the American dietetic association*, 104(2):186–191, 2004.

[9] Monica E Campbell, Charles E Gardner, John J Dwyer, Sandy M Isaacs, Paul D Krueger, and Jane Y Ying. Effectiveness of public health interventions in food safety: a systematic review. *Can J Public Health*, 89(3):197–202, 1998.

[10] MSPH Cassandra Harrison, MS Mohip Jorder, Faina Stavinsky Henri Stern, and MD Sharon Balter. Using online reviews by restaurant patrons to identify unreported cases of foodborne illness—new york city, 2012–2013.

[11] R. Amorós D. Conesa, M.A. Martínez-Beneito and A. López-Quílez. Bayesian hierarchical poisson models with a hidden markov structure for the detection of influenza epidemic outbreaks. *Statistical Methods in Medical Research*, 24(02):206–223, 2015.

[12] K. Hornik A. Weingessel F. Leisch C.C. Chang D. Meyer, E. Dimitriadou and C.C. Lin. e1071. https://cran.r-project.org/web/packages/e1071/e1071.pdf. Misc Functions of the Department of Statistics, TU Wien, 2010.

[13] MJ Doherty, LJ Wang, S Donague, MG Pearson, P Downs, SA Stoneman, and JE Earis. The acoustic properties of capsaicin-induced cough in healthy subjects. *European Respiratory Journal*, 10(1):202–207, 1997.

[14] Centers for Disease Control and Prevention. Cdc 2011 estimates: Findings. cdc estim foodborne illn. u s. january 2014. http://www.cdc.gov/foodborneburden/2011foodborneestimates.html. [Online; Accessed June-2014].

[15] Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence*, 15(01):9–42, 2001.

[16] Volker Grimm, Uta Berger, Donald Deangelis, J Polhill, Jarl Giske, and Steven F. Railsback. The odd protocol: A review and first update. 221:2760–2768, 11 2010.

[17] Hawkins J. Nguyen L. Nsoesie E. Tuli G. Mansour R. Harris, J. and J. Brownstein. Using twitter to identify and respond to food poisoning. *Public Health Management and Practice*, 23(6):577–580, 2017.

[18] Michael I Jordan and Romain Thibaux. The kernel trick. *Lecture Notes*, 2004.

[19] A. Okhmatovskaia K. Morrison, K. Charland and D. Buckeridge. A framework for detecting and classifying outbreaks of gastrointestinal disease. *Online Journal of Public Health Informatics*, 05(01), 2013.

[20] J Korpáš, J Sadloňová, and M Vrabec. Analysis of the cough sound: an overview. *Pulmonary pharmacology*, 9(5-6):261–268, 1996.

[21] J Korpas, M Vrabec, J Sadlonova, D Salat, and LA Debreczeni. Analysis of the cough sound frequency in adults and children with bronchial asthma. *Acta Physiologica Hungarica*, 90(1):27–34, 2003.

[22] Eric C Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N Patel. Accurate and privacy preserving cough sensing using a low-cost microphone. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 375–384. ACM, 2011.

[23] Xiaoyan Li, Alexander Doroshenko, and Nathaniel D Osgood. Applying particle filtering in both aggregated and age-structured population compartmental models of pre-vaccination measles. *PloS one*, 13(11):e0206529, 2018.

[24] M. Nasehi M. Moosazadeh, N. Khanjani and A. Bahrampour. Predicting the incidence of smear positive tuberculosis cases in iran using time series analysis. *Iranian Journal of Public Health*, 44(11):1526–1534, 2015.

[25] Birring S. S. Pavord I. D. & Evans D. H Matos, S. An automated system for 24-h monitoring of cough frequency: the leicester cough monitor. *IEEE transactions on bio-medical engineering*, 54:1472–1479.

[26] Sergio Matos, Surinder S. Birring, Ian D. Pavord, and David Evans. Detection of cough signals in continuous audio recordings using hidden markov models. *IEEE transactions on bio-medical engineering*, 53:1078–83, 07 2006.

[27] Sara McPhee-Knowles. *The complex problem of food safety : Applying agent-based modeling to the policy process*. PhD thesis, University of Saskatchewan, 2015.

[28] Osgood N. McPhee-Knowles S. *Agent-based Models and Health Oriented Mobile Technologies; Chapter in Kaplan G.A., Diez Roux A., Galea S., Simon C.P., Editors, Growing Inequality: Bridging Complex Systems, Health Disparities, and Population Health. Oxford University.* Westphalia Press, an Imprint of the Policy Studies Organization, 2017, pp. 275–296., 2016.

[29] Paul S Mead, Laurence Slutsker, Vance Dietz, Linda F McCaig, Joseph S Bresee, Craig Shapiro, Patricia M Griffin, and Robert V Tauxe. Food-related illness and death in the united states. *Emerging infectious diseases*, 5(5):607, 1999.

[30] AH Morice, GA Fontana, MG Belvisi, SS Birring, KF Chung, Peter Vytautas Dicpinigaitis, JA Kastelik, LP McGarvey, JA Smith, M Tatar, et al. Ers guidelines on the assessment of cough. *European respiratory journal*, 29(6):1256–1276, 2007.

[31] Akira Murata, Yasuyuki Taniguchi, Yasushi Hashimoto, Yasuyuki Kaneko, Yuji Takasaki, and Shoji Kudoh. Discrimination of productive and non-productive cough by sound analysis. *Internal Medicine*, 37(9):732–735, 1998.

[32] Jared O'Connell and Søren Højsgaard. Hidden semi markov models for multiple observation sequences: The mhsmm package for R. *Journal of Statistical Software*, 39(4):1–22, 2011.

[33] Government of Canada. Foodborne disease infographics global. `http://www.who.int/foodsafety/areas_work/foodborne-diseases/infographics_global_en.pdf?ua=1`. [Online; accessed May-2017].

[34] Weicheng Qian, Nathaniel D Osgood, and Kevin G Stanley. Integrating epidemiological modeling and surveillance data feeds: a kalman filter based approach. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, pages 145–152. Springer, 2014.

[35] Elizabeth C Redmond, Christopher J Griffith, Jenny Slader, and Tom J Humphrey. Microbiological and observational analysis of cross contamination risks during domestic food preparation. *British Food Journal*, 106(8):581–597, 2004.

[36] D.H. David R.M. Kaplan, M.L. Spittel. *Population Health: Behavioral and Social Science Insights*. Agency for Healthcare Research and Quality, Rockville, MD., 2015.

[37] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12(1):77, Mar 2011.

[38] Anahita Safarishahrbijari and Nathaniel D Osgood. Social media surveillance for outbreak projection via transmission models: Longitudinal observational study. *JMIR public health and surveillance*, 5(2):e11615, 2019.

[39] Patrick Seitzinger. Addressing limitations in foodborne outbreak investigation: Recall bias and the feasibility of new surveillance strategies, 2017.

[40] Patrick Seitzinger, Nathaniel Osgood, Wanda Martin, Joanne Tataryn, and Cheryl Waldner. Compliance rates, advantages, and drawbacks of a smartphone-based method of collecting food history and foodborne illness data. *Journal of food protection*, 82(6):1061–1070, 2019.

[41] Patrick J Seitzinger, Joanne Tataryn, Nathaniel Osgood, and Cheryl Waldner. Foodborne outbreak investigation: Effect of recall inaccuracies on food histories. *Journal of food protection*, 82(6):931–939, 2019.

[42] Vinayak Swarnkar, Udantha R. Abeyratne, Anne B. Chang, Yusuf A. Amrulloh, Amalia Setyati, and Rina Triasih. Automatic identification of wet and dry cough in pediatric patients with respiratory diseases. *Annals of Biomedical Engineering*, 41(5):1016–1028, May 2013.

[43] Aydin Teyhouee and Nathaniel D. Osgood. Cough detection using hidden markov models. *CoRR*, abs/1904.12354, 2019.

[44] Waldner C. Osgood N. Teyhouee A., McPhee-Knowles S. Prospective detection of foodborne illness outbreaks using machine learning approaches. 10354, 2017.

[45] M Kate Thomas, Regan Murray, Logan Flockhart, Katarina Pintar, Frank Pollari, Aamir Fazil, Andrea Nesbitt, and Barbara Marshall. Estimates of the burden of foodborne illness in canada for 30 specified pathogens and unspecified agents, circa 2006. *Foodborne pathogens and disease*, 10(7):639–648, 2013.

[46] CW Thorpe, LJ Toop, and KP Dawson. Towards a quantitative description of asthmatic cough sounds. *European Respiratory Journal*, 5(6):685–692, 1992.

[47] Yuan Tian and Nathaniel Osgood. Comparison between individual-based and aggregate models in the context of tuberculosis transmission. In *Proceedings, the 29th International Conference of the System Dynamics Society*, pages 1–29, 2011.

[48] LL Toop, KK Dawson, and CW Thorpe. A portable system for the spectral analysis of cough sounds in asthma. *Journal of Asthma*, 27(6):393–397, 1990.

[49] G.L. Wallstrom X. Jiang. A bayesian network for outbreak detection and prediction. pages 1155–1160, 2006.

[50] W. J. Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.

Appendix A

A.1 Collected answers from Illness Reporting Survey in JSON format

Q/A Set 1:

```
{
  "order_id": 1,
  "q_type": "single_choice",
  "q_content": "Did you consult a health-care professional\\
  regarding this illness?",
  "resp": [
    {
      "resp_time": "2016-01-15 21:22:56.073+0000",
      "answer_content": "No",
      "answer_id": 2
    }
  ]
}
```

Q/A Set 2:

```
{
  "order_id": 2,
  "q_type": "single_choice",
  "q_content": "Did you suspect your illness might be related to \\
  consumption of alcoholic beverages?",
  "resp": [
    {
      "resp_time": "2016-01-15 21:22:57.907+0000",
      "answer_content": "Yes",
      "answer_id": 1
    }
  ]
}
```

Q/A Set 3:

```
{
  "order_id": 3,
  "q_type": "multiple_choice",
  "q_content": "I've got ...",
  "resp": [
    {
      "resp_time": "2016-01-15 21:23:00.640+0000",
      "answer_content": "Nausea",
      "answer_id": 1
    },
    {
      "resp_time": "2016-01-15 21:23:01.039+0000",
```

```
      "answer_content": "Vomiting",
      "answer_id": 2
    },
    {
      "resp_time": "2016-01-15 21:23:01.439+0000",
      "answer_content": "Diarrhea",
      "answer_id": 3
    },
    {
      "resp_time": "2016-01-15 21:23:01.871+0000",
      "answer_content": "Abdominal pain and cramps",
      "answer_id": 4
    },
    {
      "resp_time": "2016-01-15 21:23:02.269+0000",
      "answer_content": "Fever",
      "answer_id": 5
    },
    {
      "resp_time": "2016-01-15 21:23:02.640+0000",
      "answer_content": "Other",
      "answer_id": 6
    }
  ]
}
```


A.2 Connecting to Cassandra

Before we start answering the above questions, we first set up our Spark environment and connect to our database. To do so, we need first to stop the default Spark Context and setup a new one on top a new Spark Configuration, which requires new elements, such as the Cassandra Connector and its required credentials, enabling us to connect to the host of our study database.

```
sc.stop()
val conf = new SparkConf(true).set("spark.cassandra.connection.host"
  ,"SERVER_ADDRESS").set("spark.cassandra.auth.username", "U_NAME").
  set("spark.cassandra.auth.password", "PASS").set("spark.executor.
  memory","4g")
val sc = new SparkContext("local", "test", conf)
```

A.3 Aggregation and Filtration

Food Consumption survey responses from the two databases, corresponding to each of those study cohorts, were merged into a single dataframe *dfEatingSurvey* for the ease of computations. Same methodology was applied for aggregating the Illness Report surveys. By using the term "filter($/"duration" > -1)", survey responses which has expired (i.e., the participant has not filled the survey in a predefined time window) or canceled (participant has actively canceled the survey) are being filtered out.

```
val dfSurveyResp1 = sqlContext.read.format("org.apache.spark.sql.
  cassandra").options(Map("keyspace"-> "ethica_study_xx", "table" ->
  "survey_responses")).load()
val dfSurveyResp2 = sqlContext.read.format("org.apache.spark.sql.
  cassandra").options(Map("keyspace"-> "ethica_study_yy", "table" ->
  "survey_responses")).load()
val dfEatingSurvey = dfSurveyResp1.union(dfSurveyResp2).filter($
  "survey_id"====** || $"survey_id"====**).filter($"duration">-1)
```

A.4 Food Source Diversity

If we have a look to *dfEatingSurvey* we will notice that our desired information is in the *resp* column.

```
resps
---------+-----------+-------------+--------------------------------+--
[1 -> {"order_id": 1, "q_type": "single_choice", "q_content": "Where is
the source of this food?",
"resp": [{"resp_time": "2016-01-17 17:38:54.321+0000",
"answer_content": "Eating food prepared at home", "answer_id": 4}]},
2 -> {"order_id": 2, "q_type": "image",
"q_content": "Please take a photo of the food you are eating.",
"resp": [{"answer_url": "resp_files/530/image/20160117115302751-
83b7044351f9b19e-378-2.jpg",
"resp_time": "2016-01-17 17:53:02.726+0000"}]},
3 -> {"order_id": 3, "q_type": "audio_text",
"q_content": "Would you like to explain what is this food?", "resp":
[{"resp_time": "2016-01-17 17:53:42.797+0000", "answer_content":
"Sunny side up egg, slice of bread  and Indian tea with milk"}]}]
```

At our first attempt, we need to define a function which takes an integer value (i.e., a question number) as the key of a map data structure and when applied on *resp* column, will return the value stored in the map for any single element in that column. These new values which are in valid **JSON** format will be appended to our dataframe as a new column named "q1Info".

```
val mapValueExtractor = ((Q_N:Int) => udf((m:Map[Int,String])
  => m.getOrElse(Q_N,"")))
```

We need to dig a bit further to access the answer to the food source question. For this, we need to import functionalities of a JSON library.

[nobreak=true, nobreak=true]

```
import net.liftweb.json._
implicit val formats = DefaultFormats
```

We want to parse the data using these two case classes.

```
case class Survey_Q_Resp(resp_time: String, answer_content:String,
  answer_id:Long)
case class SurveyResp(q_content: String, resp:Array[Survey_Q_Resp],
  q_type:String)
```

and we need to define a function (UDF: User Defined Function) which takes a text in JSON and then, by means of the above case classes, extracts our desired answer from the text:

```
val parseJsonContentUDF = udf((JsonText:String) => {implicit val
  formats = DefaultFormats;parse(JsonText).extract[SurveyResp].
  resp(0).answer_content})
```

Using a UDF, we will use abbreviations as follows for the acquired answers: "CampusF" for "Food purchased on campus", "RestaurantF" for "Eating at restaurant", "R2EF_PurchOffCamp" for "Ready-to-eat food purchased off campus" and "HomeF" for "Eating food prepared at home".

A.4.1 Food Type Fraction Per User

To find the consumption frequency for each of these per-user food categories, we need to count the total number of reports and number of each food types per user. By applying the following aggregation operation on our dataframe, we will obtain the result.

```
val w = Window.partitionBy($"user\_id")
val result = (dfEatingSurveyCategorizedFood.groupBy("user\_id",
  "foodType").agg(count($"foodType").as("foodType_cnt")).
  withColumn("food_report_cnt", sum($"foodType_cnt").over(w)).
  sort(desc("user\_id"))).withColumn("foodDiversity", round((
  $"foodType_cnt"/$"food_report_cnt"),2))
```

A.4.2 Daily Food Type Consumption Frequency Per Participant

The final result will be saved in the *foodCategoryFreqDF* dataframe.

```
val reporting_duration = dfEatingSurveyCategorizedFood.withColumn(
  "record_time_secs_food",$"record_time".cast("long")).agg(min(
  "record_time_secs_food").alias("min"),max("record_time_secs_food")
  .alias("max")).withColumn("study_length",round(($"max"-$"min")/
  (3600*24))).first().getDouble(2)
val foodCategoryFreqDF =  result.withColumn("freqPerDay",
  $"foodType_cnt"/reporting_duration).select("user\_id","foodType",
  "foodDiversity","freqPerDay")
```

A.4.3 Food Types Daily Frequency Per User

the following lines of code dumps the food type daily frequency per user for all the participants of the study in JSON format.

```
val foodCategoryFreqJSON = foodCategoryFreqDF.withColumn("struct",
  struct("foodType","foodDiversity","freqPerDay")).groupBy(
  "user\_id").agg(collect_list(col("struct").as("foodInfo")).
  orderBy("user\_id").toJSON.collect
sc.parallelize(foodCategoryFreqJSON).saveAsTextFile("/result")
```

and the following shows the exported results:

```
{"user\_id":591,"C":{"food_category":"CampusF",
"FoodCategCnt/FoodReportCnt(%)":4.0,"FoodCategFreq(perDay)":0.0125}}
{"userID":591,"C":{"food_category":"R2EF_PurchOffCamp",
"FoodCategCnt/FoodReportCnt(%)":25.0,"FoodCategFreq(perDay)":0.075}}
{"userID":591,"C":{"food_category":"HomeF",
"FoodCategCnt/FoodReportCnt(%)":71.0,"FoodCategFreq(perDay)":0.2125}}
```

```
{"userID":585,"C":{"food_category":"RestAurantF",
"FoodCategCnt/FoodReportCnt(%)":4.0,"FoodCategFreq(perDay)":0.05}}
{"userID":585,"C":{"food_category":"R2EF_PurchOffCamp",
"FoodCategCnt/FoodReportCnt(%)":11.0,"FoodCategFreq(perDay)":0.15}}
{"userID":585,"C":{"food_category":"CampusF",
"FoodCategCnt/FoodReportCnt(%)":3.0,"FoodCategFreq(perDay)":0.0375}}
{"userID":585,"C":{"food_category":"HomeF",
"FoodCategCnt/FoodReportCnt(%)":83.0,"FoodCategFreq(perDay)":1.175}}
{"userID":584,"C":{"food_category":"R2EF_PurchOffCamp",
"FoodCategCnt/FoodReportCnt(%)":23.0,"FoodCategFreq(perDay)":1.0}}
{"userID":584,"C":{"food_category":"CampusF",
"FoodCategCnt/FoodReportCnt(%)":4.0,"FoodCategFreq(perDay)":0.175}}
{"userID":584,"C":{"food_category":"RestAurantF",
"FoodCategCnt/FoodReportCnt(%)":5.0,"FoodCategFreq(perDay)":0.2}}
{"userID":584,"C":{"food_category":"HomeF",
"FoodCategCnt/FoodReportCnt(%)":69.0,"FoodCategFreq(perDay)":3.05}}
```


A.5 Clinical Presentation Frequency

A.5.1 Finding Time Lag Between Report

```
val w = Window.partitionBy("user\_id").orderBy("record_time_secs")
val previousEnd = lag($"record_time_secs", 1).over(w)
val fbiReportIntervalDF = dfFBI_withRecordTimeInSec.withColumn(
  "prev_record_time_secs", previousEnd).withColumn("timeLag",
  ($"record_time_secs"-$"prev_record_time_secs"))
```

A.6 To Avoid Eating Out

A.6.1

To make our life easier, we need to add two columns to the illness reports dataframe – record_time in seconds and the time 24 hours after the illness report, again in seconds. We also change the record_time column in food report dataframe into seconds.

```
val record_time_inSecs = col("record_time").cast("long")
val oneDayLater_inSecs = col("record_time").cast("long") + 86400L
```

```
val joinedEatingWithIllnessReport = eatingSurveyDF.join(dfFBI,
  eatingSurveyDF("user_id") <=> dfFBI("user_id") &&
  eatingSurveyDF("record_time").between(dfFBI("record_time"),
  dfFBI("oneDayAfter_record_time"))
  )
```

A.6.2

```
val riskyFoodCategorizerUDF = udf((foodType:String) => (foodType match {
  case "CampusF" => 1; case "RestaurantF" => 1;
  case "R2EF_PurchOffCamp" => 0; case "HomeF" => 0; }))
val eatingReportIn24H_afterIllnessReport=joinedEatingWithIllnessReport.
  withColumn("riskyFood", riskyFoodCategorizerUDF($"foodType")).
  groupBy("user_id", "record_time_illness").agg(sum($"riskyFood").
  as("riskyFoodConsumptionCount"), count($"riskyFood").
  as("FoodConsumptionCount")).filter($"FoodConsumptionCount"!== 0)
```

Post-Illness Avoidance of Recent Restaurants

```
val exprs = Array("lat", "lon").map(_ -> "mean").toMap
val dfGPS_oneMinuteWindow = dfGPS.groupBy($"user_id", $"record_time").
  agg(exprs).withColumn("lat", round($"avg(lat)", 4)).
  withColumn("lon", round($"avg(lon)", 4))

val geoTaggedRestaurantFoodReport =
    dfGPS_oneMinuteWindow.join(onlyEatingAtRest,
    dfGPS_oneMinuteWindow("user_id_gps") <=> onlyEatingAtRest("user_id")
    && dfGPS_oneMinuteWindow("record_time_min_gps") <=> onlyEatingAtRest
    ("record_time_min_eating"))
```

```
val geoTaggedEatingJoinedWithIllness_before =
  geoTaggedRestaurantFoodReport.join(dfFBI_TodayAndOneWeekBeforeAfter,
  geoTaggedRestaurantFoodReport("user_id") <=>
  dfFBI_TodayAndOneWeekBeforeAfter("user_id")
  && geoTaggedRestaurantFoodReport("record_time").
  between(dfFBI_TodayAndOneWeekBeforeAfter("oneWeekBefore_record_time"),
  dfFBI_TodayAndOneWeekBeforeAndAfter("record_time")))
```

Appendix B

B.1 Principle behind Transition and Emission Matrices Extraction

Table 6.1 shows a simplified and shortened sample of aggregated model run outputs. For the sake of simplicity in exposition, within this discussion, we limit the number of restaurants in the model to three and for the sake of brevity, "GNO", "Rest0", "Rest1" and "Rest2" are denote the hidden state names of "Global No_Outbreak", "Restaurant[0]_Contaminated", "Restaurant[1]_Contaminated", and "Restaurant[2]_Contaminated", respectively. A schematic of the HMM for our simplified example is shown in Figure B.1.

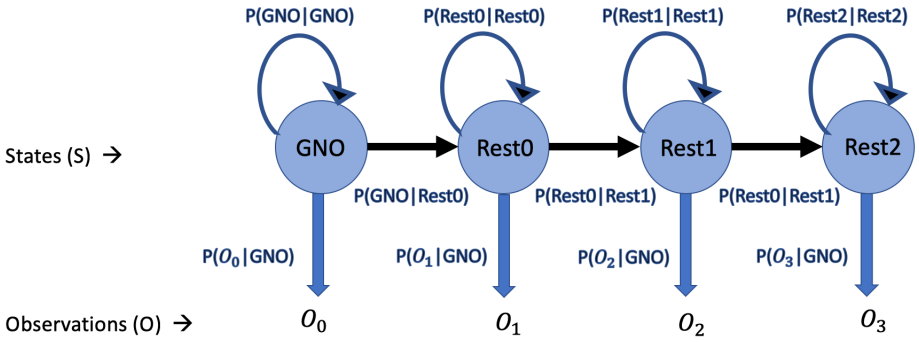


Figure B.1: Hidden Markov Model For Simplified Example

At any given time-step (each day in our example), our Markovian multistate system can be in one of the four hidden states of "GNO", "Rest0", "Rest1" or "Rest2", with transitions as shown in Figure B.1. This entails a Transition matrix as shown in Table B.2. Please note that the probability of transition from any contaminated restaurant directly to another contaminated restaurant is taken as zero, meaning that – as shown in Figure B.1 – the HMM assumes that the system must return to the GNO state for at least a minimum of one day following resolution of restaurant contamination before a new restaurant is contaminated.

To calculate the probability of transition from a current state to any of the probable states, we first need to find the probability of leaving a given state, which is equivalent to one over the mean number of time-steps that the system has spent in that state. By finding the portion of transitions from a given state to any other state and multiplying it by the probability of leaving that given state, one can calculate the per-time-step transition probability P_{xy}, where x is the current state and y is the next state of the system. We can summarize our findings as follows:

$$P_{GNO,GNO} = 1 - \frac{1}{\Delta T_{GNO}} = 1 - \frac{1}{8} = 0.875$$

$$P_{GNO,Rest0} = \frac{1}{\Delta T_{GNO}} \times \frac{GNO\_to\_Rest0\_Transition\_Count}{GNO\_Exit\_count} = \frac{1}{8} \times \frac{1}{3} = 0.042$$

$$P_{GNO,Rest1} = \frac{1}{\Delta T_{GNO}} \times \frac{GNO\_to\_Rest1\_Transition\_Count}{GNO\_Exit\_count} = \frac{1}{8} \times \frac{1}{3} = 0.042$$

Table B.1: Training Data Sample

Day	True_State	Rest0_Obs_Count	Rest1_Obs_Count	Rest2_Obs_Count
0	GNO	0	0	0
1	GNO	0	0	0
2	Rest1	0	0	0
3	Rest1	0	3	0
4	Rest1	0	5	0
5	GNO	0	8	0
6	GNO	0	0	0
7	GNO	0	0	0
8	Rest0	2	0	0
9	Rest0	6	0	0
10	GNO	0	0	0
11	Rest2	0	0	4
12	Rest2	0	0	7
13	Rest2	0	0	8
14	GNO	0	0	0
15	GNO	0	0	0

Table B.2: Transition Table for Sample Data

	GNO	Rest0	Rest1	Rest2
GNO	$P_{GNO\|GNO}$	$P_{GNO\|Rest0}$	$P_{GNO\|Rest1}$	$P_{GNO\|Rest2}$
Rest0	$P_{Rest0\|GNO}$	$P_{Rest0\|Rest0}$	0.0	0.0
Rest1	$P_{Rest1\|GNO}$	0.0	$P_{Rest1\|Rest1}$	0.0
Rest2	$P_{Rest2\|GNO}$	0.0	0.0	$P_{Rest2\|Rest2}$

$$P_{GNO,Rest2} = \frac{1}{\Delta T_{GNO}} \times \frac{GNO\_to\_Rest2\_Transition\_Count}{GNO\_Exit\_count} = \frac{1}{8} \times \frac{1}{3} = 0.042$$

$$P_{Rest0,GNO} = \frac{1}{\Delta T_{Rest0}} = 0.5$$

$$P_{Rest0,Rest0} = 1 - \frac{1}{\Delta T_{Rest0}} = 1 - \frac{1}{2} = 0.5$$

$$P_{Rest1,GNO} = \frac{1}{\Delta T_{Rest1}} = \frac{1}{3} = 0.33$$

$$P_{Rest1,Rest1} = 1 - \frac{1}{\Delta T_{Rest1}} = 1 - \frac{1}{3} = 0.67$$

$$P_{Rest2,GNO} == \frac{1}{\Delta T_{Rest2}} = \frac{1}{3} = 0.33$$

$$alignP_{Rest2,Rest2} = 1 - \frac{1}{\Delta T_{Rest2}} = 1 - \frac{1}{3} = 0.67$$

If we assume that the number of cases reporting a restaurant as being in contaminated state follows a Poisson distribution, then upon triggering the emergence of contamination within in a restaurant, that

restaurant will transition from a lower (non-outbreak) lambda to a higher lambda. With the HMM having four hidden states, for each restaurant we can define four Poisson distributions. Table B.3 shows a matrix which holds the Poisson probability density function corresponding to each of the states for each restaurant. The density function at row and column one, for instance, is corresponding to a Poisson distribution fitted over the data when only observations regarding Rest[0] and the state of "GNO" are considered.

Table B.3: Emission Matrix holding Poisson probability density functions for Sample Data

	Rest[0]	Rest[1]	Rest[2]
GNO	$f(O_{Rest[0]}\|\lambda_{GNO})$	$f(O_{Rest[1]}\|\lambda_{GNO})$	$f(O_{Rest[2]}\|\lambda_{GNO})$
Rest0	$f(O_{Rest[0]}\|\lambda_{Rest0})$	$f(O_{Rest[1]}\|\lambda_{Rest0})$	$f(O_{Rest[2]}\|\lambda_{Rest0})$
Rest1	$f(O_{Rest[0]}\|\lambda_{Rest1})$	$f(O_{Rest[1]}\|\lambda_{Rest1})$	$f(O_{Rest[2]}\|\lambda_{Rest1})$
Rest2	$f(O_{Rest[0]}\|\lambda_{Rest2})$	$f(O_{Rest[1]}\|\lambda_{Rest2})$	$f(O_{Rest[2]}\|\lambda_{Rest2})$

B.2 Emission and Transition Portions Inference During Simulation Model Run

```
Previous_Step_Probability = [1*N]; //N: number of states
Transition_Portion = [1*N]
for (State s in States){
    Probability_T_Portion = 0;
    for(i in number_of_states){
        //P(i-->s): Probability of transition from i to s
        P(i-->s) = Previous_Step_Probability[i] *
        Transition_Probability(i-->s);
        Probability_T_Portion += P(i-->s);
    }
    Transition_Portion[s] = Probability_T_Portion;
}
```

```
Emission_Portion = [1*N]
for (State s in States){
    //if state is No_outbreak
    if (s == NGO){
        Emission_Probability = 1.0;
        //iterate over the restaurants
        for(Restaurant r in restaurants){
            - get the distribution corresponding to lower occurrences of illness
            for each restaurant r, (where r is not contaminated)
            and calculate p the probability of observing o,
            given the daily report count for r.
            - //multiply the achieved probabilities:
            Emission_Probability *= p;
        }
    } else {
        for(Restaurant r in restaurants){
            // restaurant name r is same as state s
            if(r == s ){
                - get the distribution corresponding to higher occurrences of illness
                (i.e this restaurant is contaminated) and calculate p_targeted,
                the probability of observing o,
                given the daily report count for r;
                Emission_Probability *= p_targeted
            } else {
                - get the distribution corresponding to lower occurrences of
                illness for each restaurant r, (where r is not contaminated)
                and calculate p the probability of observing o,
                given the daily report count for r;
                //multiply the achieved probabilities:
                Emission_Probability *= p;
            }
        }
    }
    Emission_Portion[s] = Emission_Probability;
}
```