Learning models in social networks

Ho, Jack

2019

Ho, J. (2019). Learning models in social networks. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/85449>

<https://doi.org/10.32657/10356/85449>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

LEARNING MODELS IN SOCIAL NETWORKS

JACK HO

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2018

LEARNING MODELS IN SOCIAL NETWORKS

JACK HO

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2018

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

29/11/2018

.....
Date

.....
Jack Ho

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

29/11/2018

.....
Date

.....
Tay Wee Peng

Authorship Attribution Statement

This thesis contains material from 2 papers published in the following peer-reviewed journal where I was the first and/or corresponding author.

Chapters 2 and 3 are published as J. Ho, W. P. Tay and T. Q. S. Quek, "Robust detection and social learning in tandem networks," *2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, 2014, pp. 5457-5461.

The contributions of the co-authors are as follows:

- Prof. Tay provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof. Quek.
- I came up with the theoretical proofs in the paper and ran the simulations. I also analyzed the data.

Chapters 2 and 3 are published as J. Ho, W. P. Tay, T. Q. S. Quek and E. K. P. Chong, "Robust Decentralized Detection and Social Learning in Tandem Networks," in *IEEE Trans. on Sig. Process.*, vol. 63, no. 19, pp. 5019-5032, Oct.1, 2015.

The contributions of the co-authors are as follows:

- Prof. Tay provided the initial project direction and edited the manuscript drafts.
- I prepared the manuscript drafts. The manuscript was revised by Prof. Quek.
- Prof. Chong gave some helpful advice and also made some final revisions of the manuscript.
- I came up with the theoretical proofs in the paper and ran the simulations. I also analyzed the data.

29/11/2018

.....
Date

.....
Jack Ho

Learning Models in Social Networks

by

Jack Ho

Submitted to the Nanyang Technological University
on Nov 30, 2018, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Electrical and Electronic Engineering

Abstract

This thesis studies the problem of modeling learning through a set of agents connected via a social network. We study how learning can occur in a social network and propose several models, each with their own practical application. Just like in many practical applications, there is a subtheme of imperfect information running through our models, where the information that each decision maker receives is either noisy or incomplete.

We first analyze a robust detection problem in a tandem network where agents sequentially receive private signals about the state of the world, as well as the decision of their predecessors, and then attempt to make a decision about the state of the world. The knowledge they have regarding their predecessors' private signals, however, is incomplete, and we propose a policy to allow agents to minimize the worst-case error probability over all possible distributions of their predecessors' private signals.

We also characterize the conditions where asymptotic learning occurs, that is, when the error probability of the agents converge to zero as the number of agents increases. We study a scenario where agents are working collectively to minimize the asymptotic error probability, as well as the case where agents act selfishly to minimize their personal error probability. We analyze these two scenarios under two different possibilities, one where the level of uncertainty towards each other agent's private signal is constant throughout and another where it is allowed to vary with time. We also examine the situation where agents do not know their exact position in the tandem network.

We then consider the problem of online advertising, where an online retailer wishes to learn more about users in a social network. We model the problem in a variety of different ways. One of these models is optimizing a string-submodular function with incomplete information about the step-wise gains of each user. With complete information, the retailer can adopt a greedy strategy by choosing the agent with the highest step-wise gain at each time step. We provide a pseudo-greedy method for the retailer to choose which user to target, as well as several strategies to decide which user's step-wise gains to observe. We then derive a lower bound for each strategy's performance with respect to the optimal string's performance. We also

discuss conditions where each strategy outperforms the others.

Finally, we model the problem of online advertising again, this time as a multi-armed bandit problem in a social network setting. In fact, we use two different multi-armed bandit models, one where targeting a user allows the retailer to observe a biased version of the rewards of his immediate neighbors in the social network, and one where the retailer can pay an additional cost to explore further than the immediate neighbors of the targeted user. We propose an algorithm for the retailer to select the targeted user for each of these models and compare their performance to that of existing algorithms.

Thesis Supervisor: Tay Wee Peng
Title: Associate Professor

Thesis Supervisor: Tony Quek
Title: Associate Professor

Acknowledgments

It is hard for me to express my extreme gratitude to my two thesis supervisors, Prof. Tay Wee Peng and Prof. Tony Quek. Without Prof Tay's guidance, or his infinite patience and kindness in dealing with my shortcomings, there is no way that I would have completed this thesis. It has truly been a blessing to have him as my supervisor. I would like to thank Prof. Quek too for offering me advice when I needed it and looking out for me. I have truly learned a lot from them during my time in NTU.

Besides my two supervisors, the friends that I have made at NTU have also made this journey enjoyable. I would like to thank Wang Yuan, Sun Meng, Tze Siong, Jielong, Wenchang and Qiyu. I would also like to thank my friends outside NTU, Alvin, Belinda, Edmund, Jia Min, Kelvin, Perry and Xinying, who have offered me support all the way.

Lastly, I would like to dedicate this thesis to my family, who has been with me through both good times and bad.

Contents

1	Introduction	17
1.1	Learning in a social network	17
1.2	Literature review	22
1.3	Contributions	26
2	Robust Detection	29
2.1	Model	29
2.2	Minimax Decentralized Detection and Social Learning	30
2.3	Uncertainty Classes	32
2.3.1	2-alternating capacities	32
3	Asymptotic Learning Under Robust Detection	37
3.1	Robust Learning in a Tandem Network	37
3.2	Asymptotic Detection and Social Learning	41
3.2.1	Identical Uncertainty Classes	42
3.2.2	Varying uncertainty classes for social learning	46
3.2.3	Unknown agent positions	51
3.3	Numerical Results	59
3.4	Summary	62
4	Learning as a String Submodular Problem	65
4.1	Set submodular functions	66
4.2	String submodular functions	67

4.3	Problem formulation	68
4.4	Strategies	71
4.5	Results and Discussion	72
4.6	Numerical results	80
4.7	Summary	83
5	Learning as a Multi-armed Bandit Problem	85
5.1	Multi-armed bandits	86
5.2	Learning from biased neighbor observations	88
5.3	The UCB-R algorithm	91
5.4	Learning from larger neighborhoods	101
5.5	The UCB-V algorithm	103
5.6	Numerical results	108
5.7	Summary	111
6	Conclusion and Future Work	113
6.1	Conclusion	113
6.2	Future Work	114

List of Figures

2-1	Hypothesis testing in a tandem network.	30
4-1	Ratio against optimal string for each strategy	81
4-2	Ratio against optimal string for each strategy	82
5-1	Comparison of UCB-R algorithm with UCB1 and UCB-N	109
5-2	Comparison of UCB-V algorithm with UCB1 (1000 time steps)	110
5-3	Comparison of UCB-V algorithm with UCB1 (10 time steps)	111

List of Tables

3.1	Conditions for asymptotic learning	63
-----	--	----

Chapter 1

Introduction

1.1 Learning in a social network

Social networks have become a ubiquitous part of our everyday lives. From connecting with friends and colleagues to reading news reports, we have come to rely on social networks to provide us information about the world around us. However, communication through social networks have their limits. In some cases, there may be too much information to sift through in order to draw the right conclusions. For instance, a news aggregator may need to choose which sources to cull information from as there are simply too many news sources out there. On the other end of the spectrum, an individual user in a social network may only be limited to information provided by that user's immediate neighborhood of friends or followers. In some cases, it may be possible to obtain information from other users in the network, but that may come at a cost. Furthermore, some of this data may be conflicting, with ignorant or malicious users in a network possibly providing misleading or inaccurate information. Clearly, learning in social networks can come in many forms, each with its own different set of constraints. In this thesis, we model some of these forms and propose efficient algorithms to aid learning under each of these restrictions.

In this thesis, we will examine three different models of learning in a social network. The first is based on the individual, where a large number of individual users are attempting to determine the underlying state of the world. This state of the world

could be anything from the current traffic conditions to the quality of a product or service being offered or the truthfulness of certain news articles. Each individual will make this determination based on a noisy private signal as well as the decision of its neighbor in a social network.

Many papers have studied the problem of learning in a social network, and it has been shown that under certain reasonable assumptions, it is possible for individuals to learn the true underlying state of the world in a social network setting. That is, the error probability of each individual regarding the state of the world will converge towards zero as information flows through the network. However, in many of the models presented in these papers, an important assumption is that each individual is completely aware of how informative the signals received by every other individual is. In this thesis, we challenge that assumption, and add a constraint that each individual only has partial information about the statistics of the signals received by other individuals in the network. In this way, any learning algorithm used will have to be robust with regard to this incomplete information, and hence we will be referring to this as *robust detection*.

We first consider robust decentralized detection and social learning in tandem networks, which are very simple in structure. Social networks and Internet of Things (IoT) networks are much more complex in practice. Therefore, our results are limited, and can only be applied *heuristically* to more practical networks. Our analysis forms the foundation for studying more complex networks like trees and general loopy graphs, and provides insights into designing optimal decision rules for such networks. For example, although using likelihood ratio tests based on the least favorable distributions (LFDs) at each agent is known not to be optimal for loopy graphs, we expect this to produce reasonable results in practice. Under a tandem network, individuals in a social network, henceforth known as *agents* or *users*, sequentially choose between the two sides of a binary hypothesis. To use the examples above, this could be whether or not there is heavy traffic on a certain road, or if a certain restaurant provides value for money. To aid each agent in coming to a decision, every agent, except the first, will receive the decision of the agent before it, as well as some *private*

signal.

To determine how effective a learning algorithm is, we define the *asymptotic error probability* of a social network as the limit of the error probability of agent decisions as the network grows larger. If the asymptotic error probability is 0, then we say that *asymptotic learning* has occurred.

To find out if asymptotic learning can occur, it is critical to look at the *likelihood ratio* of each private signal. If each private signal's likelihood ratio is bounded away from 0 and infinity, then we say that the likelihood ratios are *bounded*. Otherwise, if there are signals with likelihood ratios that can be arbitrarily low and high, we say that the likelihood ratios are *unbounded*.

In general, if the structure of each agent's private signal is completely known to every other agent, then the agents' decisions will eventually converge towards the true state of the world if and only if the likelihood ratios are unbounded. If they are bounded, there will come a point where each individual will ignore its own private signal and just copy the previous agent's decision. This will lead to an asymptotic error probability above 0. In this thesis, we assume that there is some doubt about the structure of every agent's private signal, with each one coming from a known *uncertainty class* for each state of the world. We then use the *least favorable distributions* of the uncertainty classes to devise a learning algorithm so as to minimize the maximum error probability of the agents' decisions over all possible private signal structures.

In previous literature, there is also another common assumption that agents in a social network are collectively working towards making the best eventual decision. In practice, this is not often true. Most users of social networks generally aim to propagate the decision that they feel is the correct one, without taking the network's eventual decision into account. This may lead to an increase in error probability of the final decision. The optimal strategy for the network often involves some agents propagating a decision that they believe is incorrect. Hence, we attempt to take this into consideration and examine both cases. The first is when agents collectively work towards reducing the asymptotic error probability of the network as much as

possible, which we denote as *decentralized detection*. The other is when agents work to minimizing their personal error probability, which we call *social learning*.

Furthermore, individual users in a social network may not know the general topology of the network. In particular, for the tandem network, they may not know which position they are in. Under this further constraint, we investigate how to minimize the maximum error probability of the network and if asymptotic learning can occur.

Another possible model for learning in a social network involves an entity outside the network attempting to gauge certain properties of users in the network. For example, a company might wish to learn the preferences of users of its products, so that it can better gauge which customers to advertise to. It may be possible to conduct market research to learn this. However, the preferences of customers continually change, and it is impractical to be conducting market research on each group of customers all of the time. This unique set of constraints lends itself to be modeled via the optimization of a *string submodular function*.

String submodular functions have applications in many areas such as sensor management or online advertising. There are two concepts key to the structure of string submodular functions. The first is that of *monotonicity*, where the objective value of a string is no less than that of its substrings. The second is the *diminishing return* property, where the net gain in choosing an action decreases as the string length increases. These properties are ideal to apply to our problem of advertising in a social network. In our problem, each group of customers can be represented by an agent, and the company would like to choose a group of customers to advertise to at each time step in order to maximize its returns. The monotonicity principle is satisfied as advertising to more groups of customers is certain to increase the revenue of the company, while the diminishing returns property is satisfied as the first wave of advertising is likely to generate the most revenue, with a decrease in revenue in further waves as time goes on.

Efficiently finding the optimal string of a string submodular function can be found using dynamic programming, but is generally regarded as intractable [12]. This is because the computational complexity of finding an optimal solution is exponential

both with respect to the length of the string and the size of the set of possible actions. One alternative is to use the greedy algorithm where the step-wise gain at each time step is maximized. However, in reality, it is impractical for the company to know the step-wise gain at each time step for every group of customers. Hence, we have constrained ourselves to only being able to discover the step-wise gains of a subset of agents at each time step in our model. However, because of this constraint we have imposed on ourselves, we have to impose another constraint on the model. We assume that there is a limit to the decrease in the incremental reward, which is known by the decision maker.

Under this additional constraint, we propose a *pseudo-greedy* algorithm which chooses the action with the highest known step-wise gain at each time step. Because we are only allowed to observe the step-wise gains of some agents at each time step, deciding which agents to observe is of importance. We propose a few strategies on how to select which agents' step-wise gains to observe at each time step. We then analyze the performance of each of our proposed strategies under the pseudo-greedy algorithm, comparing it to that of the optimal string's.

A third way of modeling learning in a social network would be to model it as a *multi-armed bandit* problem. In the canonical multi-armed bandit problem, actions are chosen sequentially so as to maximize the cumulative reward. The reward of each action at each time step is a stochastic random variable that has a fixed but unknown mean. The classical tradeoff is between that of *exploitation* by choosing a well-explored action that has a high probability of yielding a high reward, and that of *exploration* by choosing actions which one knows less about, which may potentially yield a higher or lower reward.

Multi-armed bandit models can be a good way of modeling learning in a social network, as the decision maker in a multi-armed bandit problem is essentially trying to learn the mean reward of each of the possible actions. This translates well to a social network setting as each agent can be represented by an action. One possible weakness of using the classical multi-armed bandit model, though, is that it is assumed that the mean reward of each action remains constant throughout. This may not be true

in practical situations. There are more advanced multi-armed bandit models which deal with changing reward structures, but for simplicity of analysis we have restricted ourselves to constant mean rewards.

A common way to measure the performance of a multi-armed bandit algorithm is through the *cumulative regret* of the algorithm. This is the difference between the expected cumulative reward of the algorithm and the expected cumulative reward of choosing only the action with the highest expected reward. An optimal solution for the classical multi-armed bandit problem is the upper confidence bound algorithm (UCB1), which achieves a logarithmic order of regret in the worst case scenario.

In the traditional multi-armed bandit problem, choosing one action gains no information about the expected rewards of other actions. However, in a social network setting, if we represent each action as an agent in a network, choosing an agent may gain us information about its neighboring agents. A possible application could be the example above where a company is choosing which customers to market its products too. By advertising a product to a customer, the company might be able to exploit the social network structure in order to gauge the reaction of the customer's neighbors towards its product.

We model the information gain of the neighbors' rewards in two different ways. In the first, we assume that when an agent is chosen, the company gain its associated reward as well as knowledge about its neighbors in the form of noisy signals based on their rewards. In the second, the company gains the chosen agent's associated reward and then may optionally pay a cost to discover the exact reward value of its neighbors. Using these two models, we propose algorithms to choose agents and compare them to the UCB1 algorithm.

1.2 Literature review

The literature on learning in social networks and decentralized detection is vast. While not exactly in the form of a social network, the work in [14], [10] and [11] studies individuals making decisions by observing the past actions of others. The

idea of bounded and unbounded likelihood ratios was introduced in [54] and this will play a critical role in our thesis. The concept of asymptotic learning can be found in [3], where necessary and sufficient conditions for asymptotic learning to occur in social networks are derived. However, in [3], it is assumed that the private signal structures of each agent in the network is completely known. In [23], the learning model is studied in a modified tandem network. Instead of observing only the previous agent's decision, each agent gets to observe the previous K agents' decisions. Here, it was shown that learning can occur even with bounded likelihood ratios if $K > 1$. A similar problem is examined in [42], where instead of the previous K agents' decisions, the current agent gets to observe the decision of a random subset of all the previous agents. In [62], learning in a tandem network is explored using a Neyman-Pearson formulation.

Decentralized detection has been widely studied as well in [7, 15, 21, 28, 57–60, 63]. In particular, decentralized detection policies for tandem networks are also considered in [51], and conditions for the error probability approaching zero as the number of agents grows are derived. This is a network where each agent after the first receives exactly one decision from its predecessor. The authors also study a sub-optimal scheme where each sensor selfishly tries to minimize its own error, as opposed to the error of the root agent.

The work in [3] and [51] is fairly similar to our own. However, a key difference is the knowledge of the private signal structure of agents. In [3] and [51], agents know the private signal structure of each other agent under either state of the world. This may not always be a realistic assumption to make. For instance, an example of a private signal when determining the quality of food in a restaurant could be the agent's opinion of the food. As different people have different preferences, it would be unreasonable to expect that each agent knows exactly the preferences of each agent in its neighborhood. We relax this assumption of complete knowledge, and allow some doubt in each agent's knowledge through the use of uncertainty classes.

Uncertainty classes were introduced in the robust detection framework first proposed by [31], where a single agent is used to model the case where the observation

distributions are not known exactly. A survey of results in this area can be found in [34]. The underlying probability distributions governing the agent observations are assumed to belong to different uncertainty classes under different hypotheses, and it is shown that under a minimax error criterion, the optimal decision rule for the agent is a likelihood ratio test based on the pair of least favorable distributions (LFDs). Subsequently, the work [61] investigates robust detection in a finite parallel configuration, with and without a fusion center. The problem of robust sequential detection is studied in [28]. Robust social learning however has not been addressed in these works. In addition, robust detection and learning have not been studied for the tandem network.

There has been some work done on submodular functions. Set submodular functions, which are special cases of string submodular functions, have been studied extensively. Some of the work done on this topic include [5, 6, 25, 35, 36, 41, 43, 46, 48, 49, 55, 70, 71]. Both [35] and [46] show the application of set submodular functions to influence in social networks. There are other applications as well like sensor scheduling [32, 39, 40], online advertising [5, 29, 45], set cover problems [6, 24, 47] and control problems [43].

In [48], it was shown that a greedy approach achieves at least an $(1 - e^{-1})$ -approximation of the optimal solution for a uniform matroid. This bound was improved in [66]. Some papers have also considered the set submodular problem with the assumption that the order of actions matters [5, 55]. This results in a problem that is similar to the string submodular problem that we are studying, except that each action can only be selected once. It was shown in [5] that a greedy approach achieves the $(1 - e^{-1})$ -approximation of the optimal solution as well under these assumptions.

The concept of string submodularity was introduced in [41]. It was shown in [69] and [70] that a greedy strategy achieves at least a $(1 - e^{-1})$ of the optimal string. These results are further refined in [72]. In [56], string submodularity is applied to a set cover problem and this bound is improved using a gradient based algorithm. Our thesis extends this result in another direction, considering a scenario where it is not feasible to always be able to know which action the greedy action is.

There is a wealth of literature on the multi-armed bandit problem. In the seminal work of [37], it was shown that the expected regret of the decision maker will grow at least logarithmically with the number of time steps. An algorithm that achieves this lower bound was also proposed in [37]. Later, in [8], this algorithm was expanded and is now known as the UCB1 algorithm, the most common algorithm to solve the stochastic multi-armed bandit problem.

Besides the standard multi-armed bandit problem, there are also many other variants of bandit problems. Some of them are close to our model. In [50], the problem of dependent bandits is studied, where actions are embedded in a metric space, and those which are close to each other have similar reward probability distributions.

Contextual bandits are studied in [13, 17, 20, 27, 38, 52, 53, 64, 65, 67], where at each time step, a context is chosen from a context set $\mathcal{S}$, and the reward distributions for each action are dependent on the context chosen. In [67], contextual bandits are put in a social network setting. Each time step, the decision maker would be able to observe the context and payoff of actions adjacent to the chosen one. This is similar to our work, except that we have chosen the observation to be a biased reward instead of using a contextual bandits setting. The work in [27] uses a model that is not exactly a social network, but fairly similar. Actions are divided into clusters, where actions in the same cluster have similar reward distributions given the context.

In [20], a different way of the multi-armed bandit problem in a social network setting is presented. Here, it is not the actions that are embedded in a social network structure. Instead, the problem studied is one where there are multiple decision makers in a social network is considered, where adjacent decision makers are allowed to share information (contexts and rewards) of the actions they have chosen.

The variant which is closest to our models is the one used in [18, 19, 44]. These papers also examine the multi-armed bandit in a social network setting in a way that is similar to our first model. However, we have chosen to add a constraint where the observations of the chosen neighbors' rewards are biased, as opposed to being able to observe them accurately, as assumed in these papers. The UCB-N algorithm is introduced in [19], and we will be comparing our UCB-R algorithm to it.

1.3 Contributions

In this section, we describe the main results of our thesis. In Chapter 2, we formulate our minimax robust learning problem and introduce the concept of uncertainty classes. In Chapter 3, we expand on the result that a likelihood ratio test based on the pair of least favorable distributions of the uncertainty classes will minimize the maximum false alarm probability and miss detection probability no matter what the actual signal structure of each agent's private signals is. This is because such a likelihood ratio test is optimal when the marginal distributions of the agents' information are exactly the least favorable distributions. However, the test also attains the maximum error probability when that condition is true. We generalize the result to the tandem network.

Because of this property, we can then show that to solve the minimax decentralized detection problem when each agent's private signal is from a pair of known uncertainty classes, it suffices to consider the problem when each agent's private signal structure is made up of the respective pair of least favorable distributions. We show that this is true for the social learning problem too.

We also show the necessary and sufficient conditions for asymptotic learning to occur in the decentralized detection and social learning problems respectively, when the uncertainty classes are identical for all agents. In the decentralized detection problem, asymptotic learning will occur when the likelihood ratios of the private signals are unbounded, and there is no uncertainty over the private signals for every agent under one of the states of the world. For the social learning problem, having no uncertainty over the private signals for every agent under both states of the world is necessary for asymptotic learning to occur.

We also consider the situation where the uncertainty classes are allowed to vary between agents, and provide a necessary and sufficient condition for asymptotic learning to occur in the social learning problem under this assumption.

We show that for the decentralized detection problem, we should consider the problem when each agent's private signal structure is made up of the respective pair

of least favorable distributions even when agents do not know their position in the social network. We demonstrate this for the social learning problem as well. We also provide numerical results to show that even though the exact threshold of the optimal likelihood ratio test is challenging to determine analytically, it is easily estimated numerically.

In Chapter 4, we proceed to the model where learning is represented by trying to optimize a string submodular function. We propose strategies to select which agent to choose for its reward and which subset of agents to observe. We then provide a lower bound of the ratio of the functional value of the string generated by our strategy to the optimal string's value, when no other agents are allowed to be observed other than the one chosen at each time step. We also extend this result to derive bounds for each of our strategies to select which subset of agents to observe, then show some numerical results to compare them with other baseline strategies.

In Chapter 5, we move on to our next model, where we model learning in a social network as a multi-armed bandit problem. We do this in two different ways, each with its own unique assumptions. In the first, we assume that the decision maker can observe not just the reward of the chosen agent, but also that of that agent's immediate neighbors. However, the observations from the neighbors are biased. We propose the Upper Confidence Bound - Robust (UCB-R) algorithm to allow the decision maker to estimate the value of the biases, and to determine which agent to select at each time step.

In the second multi-armed bandit model, we assume that the decision maker can pay a cost to observe the rewards of agents other than the chosen agent, where the cost scales with the distance between the observed agents and the chosen agent. Here, we propose the Upper Confidence Bound - Value (UCB-V) algorithm to aid the decision maker in not only choosing an agent at each time step, but also how far away from the chosen agent he should explore. We then present some numerical results to compare the performance of the UCB-R and UCB-V algorithms with other commonly used algorithms.

Finally, in Chapter 6, we close with a summary of our results and a discussion of

possible future extensions.

Chapter 2

Robust Detection

2.1 Model

In this chapter, we describe our tandem network model as well as outline a few important concepts relating to robust detection. We consider a tandem network consisting of N agents, with agent 1 being the first agent and N being the last (see Figure 2-1). Each agent's task is to determine the state of the world. To simplify our calculations, we assume that the decisions that each agent makes, as well as the underlying state of the world, is binary. Agents sequentially make their decision about the state of the world and pass this decision on to the next agent in the network.

We consider a binary hypothesis testing problem in which the true hypothesis H is H_i with prior probability $\pi_i \in (0, 1)$, for $i = 0, 1$. The state of the world H_i is unknown, and for $k = 1, \dots, N$, and conditioned on $H = H_i$, each agent k in the network makes an observation Y_k , with distribution $P_{i,k}$. In various applications like inference in a social network, the distribution $P_{i,k}$ may be unknown a priori. Therefore, in contrast to much of the previous literature on learning in social networks, we assume that $P_{i,k}$ belongs to an uncertainty class $\mathcal{P}_{i,k}$ of distributions, and so each agent only knows that the previous agents' information follows a distribution that is an element of the uncertainty class. The uncertainty classes, however, are known to each agent. We assume that $\mathcal{P}_{0,k}$ and $\mathcal{P}_{1,k}$ are disjoint and that all distributions in $\mathcal{P}_{0,k}$ and $\mathcal{P}_{1,k}$ are absolutely continuous with respect to one another. This makes it

Figure 2-1: Hypothesis testing in a tandem network.

so that no one signal can immediately reveal the state of the world. Furthermore, we assume that conditioned on the true hypothesis, the observations of each agent are all mutually independent. In the following, we first describe the decision making process of the agents under the decentralized detection and social learning formulations, and then the uncertainty class models we adopt in this thesis.

2.2 Minimax Decentralized Detection and Social Learning

Each agent k makes a decision $U_k = \phi_k(Y_k, U_{k-1}) \in \{0, 1\}$ about the hypothesis H , where ϕ_k is an agent decision rule whose decision i corresponds to deciding in favor of H_i , and $U_0 \equiv 0$. For $i = 0, 1$, let $P_i^{(N)} = P_{i,1} \times P_{i,2} \times \dots \times P_{i,N}$. Similarly, define $\mathcal{P}_i^{(N)} = \mathcal{P}_{i,1} \times \mathcal{P}_{i,2} \times \dots \times \mathcal{P}_{i,N}$. In the decentralized detection problem, our aim is to find a sequence of decision rules $\phi^{(N)} = (\phi_1, \phi_2, \dots, \phi_N)$ to minimize the maximum probability of error given by

$$P_N^{\text{DD}}(\phi^{(N)}) = \sup_{(P_0^{(N)}, P_1^{(N)}) \in \mathcal{P}_0^{(N)} \times \mathcal{P}_1^{(N)}} P_{e,N}(\phi^{(N)}, P_0^{(N)}, P_1^{(N)}), \quad (2.1)$$

where

$$P_{e,N}(\phi^{(N)}, P_0^{(N)}, P_1^{(N)}) = \pi_0 P_{F,N}(\phi^{(N)}, P_0^{(N)}) + \pi_1 P_{M,N}(\phi^{(N)}, P_1^{(N)}). \quad (2.2)$$

In (2.2), $P_{F,N}$ and $P_{M,N}$ are the false alarm and missed detection probabilities of agent N respectively, given the decision rules $\phi^{(N)}$ and the agents' observation distributions $P_i^{(N)}$, $i = 0, 1$. When there is only one distribution in each of the uncertainty classes, we recover the classical Bayesian hypothesis testing problem.

In the decentralized detection problem, agents are working together to minimize the error probability of the final agent. This may involve choosing a decision that does not minimize its own local error probability.

In the social learning problem, the first agent chooses its decision rule ϕ_1 to minimize

$$\sup_{(P_{0,1}, P_{1,1}) \in \mathcal{P}_{0,1} \times \mathcal{P}_{1,1}} P_{e,1}(\phi_1, P_{0,1}, P_{1,1}).$$

Each other agent k , given the decision rules $\phi^{(k-1)}$ of the previous agents $1, \dots, k-1$, is able to derive the false alarm and miss detection probabilities of agent $k-1$. In practice, agents far down the tandem will not know a lot about the distribution of the observations of agents near the start of the tandem. Hence, we use the uncertainty classes to model the lack of complete knowledge of this distribution. Each agent then seeks to find ϕ_k to minimize

$$P_k^{\text{SL}}(\phi_k \mid \phi^{(k-1)}) = \sup_{(P_0^{(k)}, P_1^{(k)}) \in \mathcal{P}_0^{(k)} \times \mathcal{P}_1^{(k)}} P_{e,k}(\phi^{(k)}, P_0^{(k)}, P_1^{(k)}). \quad (2.3)$$

In contrast to the decentralized detection problem in (2.1), each agent *myopically* seeks to minimize its local maximum probability of error. Both problems have their own applications. The decentralized detection problem can be used when there is a final decision maker at the end of the network whose decision is more important than each other agent's. The social learning problem is more applicable in many practical cases where users of a social network do not know how far the information they transmit will propagate, and will generally seek to maximize their own credibility via minimizing their local error probability.

To see an example where there is a difference in decision rules between the decentralized detection framework and the social learning one, consider the following network where agents have their information identically distributed as follows:

Under H_0 , we have

$$Y_i = \begin{cases} 0 & \text{with probability } 1 \\ 1 & \text{with probability } 0. \end{cases}$$

Under H_1 , we have

$$Y_i = \begin{cases} 0 & \text{with probability } 1 - \epsilon \\ 1 & \text{with probability } \epsilon. \end{cases}$$

Assume that $\pi_0/\pi_1 < 1 - \epsilon$. Then, under social learning, each agent will have $U_i = 1$, which leads to an error probability bounded away from zero as the number of agents grows. Under decentralized detection, the first agent will have $U_1 = 0$, and each other agents will have a decision rule $U_i = 0$ if and only if $U_{i-1} = 0$ and $Y_i = 0$. This leads to an error probability converging to zero as the number of agents approaches infinity.

2.3 Uncertainty Classes

To signify the incomplete information that each agent has of the other agents' private signal structure, we make the use of uncertainty classes. For each uncertainty class $\mathcal{P}_{i,k}$, for $i = 0, 1$, let $(\mathcal{Y}, \mathcal{A})$ be a common measurable space for the observation of each agent k in the network, where $\mathcal{Y}$ is a Polish metric space. We use the concept of 2-alternating capacities introduced by [30] to characterize the uncertainty classes for which LFDs exist for minimax tests between two hypotheses.

2.3.1 2-alternating capacities

We let Σ be a completely separable metric space, with $\mathcal{A}$ its Borel- σ -algebra and $\mathcal{M}$ the set of all probability measures on Σ . For each subset $\mathcal{P} \subset \mathcal{M}$, we can define an upper probability

$$v(A) = \sup \{P(A) | P \in \mathcal{P}\},$$

where $A \in \mathcal{A}$.

A 2-alternating capacity is a set function v which satisfies the following four prop-

erties.

$$v(\phi) = 0, v(\Sigma) = 1 \quad (2.4)$$

$$A \subset B \implies v(A) \leq v(B) \quad (2.5)$$

$$\text{If } A_n \text{ converges towards } A, \text{ then } v(A_n) \text{ converges towards } v(A) \quad (2.6)$$

$$v(A \cup B) + v(A \cap B) \leq v(A) + v(B) \quad (2.7)$$

For this thesis, it suffices to note that several natural uncertainty classes like the ϵ -contamination uncertainty class, as shown below, are special cases of the uncertainty classes defined by 2-alternating capacities.

For each hypothesis H_i and agent k , let $v_{i,k}$ be a 2-alternating capacity. Define the uncertainty class associated with H_i at agent k to be

$$\mathcal{P}_{i,k} = \{Q \in \mathcal{R} \mid Q(A) \leq v_{i,k}(A), \forall A \in \mathcal{A}\}, \quad (2.8)$$

where $\mathcal{R}$ is the set of all probability measures on $(\mathcal{Y}, \mathcal{A})$. An example of an uncertainty class is the ϵ -contamination uncertainty class [30]. For each hypothesis H_i , let $P_i^* \in \mathcal{R}$ be a nominal probability distribution. A natural uncertainty class is a set of distributions that are within a certain “distance” (in some sense) of the nominal distribution P_i^* . Let $v_{i,k} = (1 - \epsilon_{i,k})P_i^* + \epsilon_{i,k}$, where $\epsilon_{i,k} \in [0, 1)$ is called the contamination value. It is easy to see that (2.8) reduces to

$$\mathcal{P}_{i,k} = \{Q : Q = (1 - \epsilon_{i,k})P_i^* + \epsilon_{i,k}R, R \in \mathcal{R}\}, \quad (2.9)$$

which we call an ϵ -contamination uncertainty class. Another example of an uncertainty class of distributions within a “distance” to a given nominal distribution $P_{i,k}^*$ is the total variation uncertainty class [30]. In this case, we let $v_{i,k} = \min\{P_{i,k}^* + \epsilon_{i,k}, 1\}$,

and

$$\mathcal{P}_{i,k} = \left\{ Q : \sup_{A \in \mathcal{A}} |Q(A) - P_{i,k}^*(A)| \leq \epsilon_{i,k} \right\},$$

is the class of distributions within total variation distance $\epsilon_{i,k}$ of P_i^* . Our subsequent results hold for all uncertainty classes defined by 2-alternating capacities under various technical conditions. However, for illustration purposes, we will use ϵ -contamination uncertainty classes throughout this thesis as examples to interpret our theorems.

For each agent k , let $p_{i,k}$ be the density (with respect to some common measure) of $P_{i,k}$. The least favorable distributions (LFDs) for two given uncertainty classes $\mathcal{P}_{0,k}$ and $\mathcal{P}_{1,k}$ is defined in [30] to be the pair of distributions $(Q_{0,k}, Q_{1,k})$ with densities $(q_{0,k}, q_{1,k})$ such that

$$P_{0,k}(l_q(Y_k) > t) \leq Q_{0,k}(l^*(Y_k) > t)$$

and

$$Q_{1,k}(l_q(Y_k) > t) \leq P_{1,k}(l^*(Y_k) > t),$$

for all $t \geq 0$, where $l^*(Y_k)$ is the likelihood ratio $q_{1,k}(Y_k)/q_{0,k}(Y_k)$.

For the example of the ϵ -contamination uncertainty classes in (2.9), we have

$$q_{0,k}(y) = \begin{cases} (1 - \epsilon_{0,k})p_0^*(y) & \text{for } p_1^*(y)/p_0^*(y) < c'' \\ (1 - \epsilon_{0,k})p_1^*(y)/c'' & \text{for } p_1^*(y)/p_0^*(y) \geq c'' \end{cases}$$

$$q_{1,k}(y) = \begin{cases} (1 - \epsilon_{1,k})p_1^*(y) & \text{for } p_1^*(y)/p_0^*(y) > c' \\ c'(1 - \epsilon_{1,k})p_0^*(y) & \text{for } p_1^*(y)/p_0^*(y) \leq c', \end{cases}$$

where $0 \leq c' < c'' \leq \infty$ are determined such that $q_{0,k}$ and $q_{1,k}$ are probability densities. Note that $c' = 0$ if and only if $\epsilon_{1,k} = 0$, and $c'' = \infty$ if and only if $\epsilon_{0,k} = 0$.

Let $b_k = (1 - \epsilon_{1,k})/(1 - \epsilon_{0,k})$. We then have

$$\frac{q_{1,k}(y)}{q_{0,k}(y)} = \begin{cases} b_k c' & \text{for } p_1^*(y)/p_0^*(y) \leq c' \\ b_k \cdot \frac{p_1^*(y)}{p_0^*(y)} & \text{for } c' < \frac{p_1^*(y)}{p_0^*(y)} < c'' \\ b_k c'' & \text{for } p_1^*(y)/p_0^*(y) \geq c''. \end{cases} \quad (2.10)$$

In many of our theorems, we require the presence of LFDs between the uncertainty classes. The following lemma which proves the existence of LFDs for uncertainty classes derived from 2-alternating capacities was shown in [30].

Lemma 2.1. *Let $v_{0,k}$ and $v_{1,k}$ be 2-alternating capacities. Then for $i = 0, 1$, there exist probability measures $Q_{i,k} \in \mathcal{P}_{i,k}$ such that for all $t > 0$,*

$$\begin{aligned} Q_{0,k}(l^*(Y_k) \geq t) &= v_{0,k}(l^*(Y_k) \geq t) \\ Q_{1,k}(l^*(Y_k) < t) &= v_{1,k}(l^*(Y_k) < t). \end{aligned}$$

In the rest of this thesis, for any random variable Y with distributions drawn from a given pair of uncertainty classes, we let $l^*(Y)$ be the likelihood ratio $q_1(Y)/q_0(Y)$, where q_0 and q_1 are the respective densities of the LFDs of the given uncertainty classes. In addition, we use $l^*(Y = y)$ to denote the realization of $l^*(Y)$ when $Y = y$. A sequence $x_1, x_2, \dots, x_n$ is denoted as $(x_i)_{i=1}^n$. Using the foundational ideas introduced in this chapter, we can now present our results regarding learning in an infinite tandem network under robust detection in the next chapter.

Chapter 3

Asymptotic Learning Under Robust Detection

3.1 Robust Learning in a Tandem Network

When there is only a single agent, the minimax error $\inf_{\phi} P_1^{\text{SL}}(\phi)$ is achieved by setting ϕ to be a likelihood ratio test using the LFDs $(Q_{0,1}, Q_{1,1})$ [30]. A similar result is proven in [61] for a parallel network configuration. In this section, we show the same result for the tandem network. We do this by first showing that for any sequence of agent decision rules that consists of likelihood ratio tests between LFDs, the error probabilities (2.1) and (2.3) are maximized when all the private agent observations are drawn from the corresponding LFDs. We then show that when agents' private observations are drawn from the respective LFDs, then likelihood ratio tests using LFDs minimize the error probabilities.

We can now present our first result. For all $k \geq 1$, let $(Q_{0,k}, Q_{1,k})$ be the LFDs for $(\mathcal{P}_{0,k}, \mathcal{P}_{1,k})$, and $Q_i^{(N)} = Q_{i,1} \times Q_{i,2} \times \dots \times Q_{i,N}$ for $i = 0, 1$. We make use of the following two lemmas in our proof, the second of which is shown in [61].

Lemma 3.1. *For any N , $Q_1^{(N)}(U_N = 1) \geq Q_0^{(N)}(U_N = 1)$ and $Q_1^{(N)}(U_N = 0) \leq Q_0^{(N)}(U_N = 0)$.*

Proof. We will prove this lemma using mathematical induction on N . The likelihood

ratio test for agent $k \geq 1$ is of the form

$$U_i = \begin{cases} 1 & \text{if } l^*(Y_k, U_{k-1}) > t_k, \\ 0 & \text{otherwise,} \end{cases}$$

where t_k is some chosen threshold. From the definition of the LFDs, we have

$$\begin{aligned} Q_1^{(1)}(U_1 = 1) &= Q_1^{(1)}(l^*(Y_1) \geq t_1) \\ &\geq Q_0^{(1)}(l^*(Y_1) \geq t_1) \\ &= Q_0^{(1)}(U_1 = 1), \end{aligned}$$

so that the lemma holds for $N = 1$.

Assume that the lemma is true for $N = k - 1$. Each agent's observation is independent of the previous agent's decision. Hence, we have

$$l^*(Y_k, U_{k-1}) = l^*(Y_k)l^*(U_{k-1}),$$

and the likelihood ratio test can be rewritten in the form

$$U_k = \begin{cases} 1 & \text{if } l^*(Y_k) \geq t_k^0, \\ U_{k-1} & \text{if } t_k^1 \leq l^*(Y_k) < t_k^0, \\ 0 & \text{if } l^*(Y_k) < t_k^1, \end{cases}$$

where $t_k^i = t_k/l^*(U_{k-1} = i)$. We obtain

$$\begin{aligned} Q_1^{(k)}(U_k = 1) &= Q_{1,k}(U_k = 1 \mid U_{k-1} = 1)Q_1^{(k-1)}(U_{k-1} = 1) \\ &\quad + Q_{1,k}(U_k = 1 \mid U_{k-1} = 0)Q_1^{(k-1)}(U_{k-1} = 0) \\ &= Q_{1,k}(l^*(Y_k) \geq t_k^1)Q_1^{(k-1)}(U_{k-1} = 1) \\ &\quad + Q_{1,k}(l^*(Y_k) \geq t_k^0)(1 - Q_1^{(k-1)}(U_{k-1} = 1)). \end{aligned}$$

From the induction hypothesis, we then have

$$\begin{aligned} &Q_{1,k}(l^*(Y_k) \geq t_k^1)Q_1^{(k-1)}(U_{k-1} = 1) + Q_{1,k}(l^*(Y_k) \geq t_k^0)(1 - Q_1^{(k-1)}(U_{k-1} = 1)) \\ &\geq Q_{1,k}(l^*(Y_k) \geq t_k^1)Q_0^{(k-1)}(U_{k-1} = 1) + Q_{1,k}(l^*(Y_k) \geq t_k^0)(1 - Q_0^{(k-1)}(U_{k-1} = 1)) \\ &\geq Q_{0,k}(l^*(Y_k) \geq t_k^1)Q_0^{(k-1)}(U_{k-1} = 1) + Q_{0,k}(l^*(Y_k) \geq t_k^0)(1 - Q_0^{(k-1)}(U_{k-1} = 1)) \\ &= Q_0^{(k)}(U_k = 1), \end{aligned}$$

where the second inequality follows from the definition of the LFDs. The proof of the lemma is now complete. $\square$

Lemma 3.2. *Let Z_1 and Z_2 be non-negative, independent random variables. If for $k = 1, 2$, we have distributions F and G such that*

$$F(Z_k > t) \geq G(Z_k > t), \quad \forall t \geq 0, \quad (3.1)$$

then

$$F(Z_1 Z_2 > t) \geq G(Z_1 Z_2 > t), \quad \forall t \geq 0.$$

If (3.1) holds, we say that Z_k is stochastically larger under F than G .

Theorem 3.1. *Let $\phi^{(N)}$ be any sequence of likelihood ratio tests based on the LFDs $(Q_0^{(N)}, Q_1^{(N)})$ for a tandem network. Then for all $(P_0^{(N)}, P_1^{(N)}) \in \mathcal{P}_0^{(N)} \times \mathcal{P}_1^{(N)}$, we have*

$$P_{F,N}(\phi^{(N)}, Q_0^{(N)}) \geq P_{F,N}(\phi^{(N)}, P_0^{(N)}), \quad (3.2)$$

and

$$P_{M,N}(\phi^{(N)}, Q_1^{(N)}) \geq P_{M,N}(\phi^{(N)}, P_1^{(N)}). \quad (3.3)$$

Proof. We proceed by mathematical induction on N . From the definition of the LFDs, the theorem holds for $N = 1$. We now assume that it holds for $N < i$.

Since the observation of agent k is independent from the decision it receives, for $i = 0, 1$ we have

$$Q_i^{(k)}(l^*(Y_k) > t) = Q_{i,k}(l^*(Y_k) > t).$$

From the definition of the LFDs, $l^*(Y_k)$ is stochastically larger under $Q_0^{(k)}$ than under any other distribution $P_0^{(k)} \in \mathcal{P}_0^k$. To show the same for $l^*(U_{k-1})$, we obtain from Lemma 3.1 that

$$\begin{aligned} l^*(U_{k-1} = 1) &= \frac{Q_1^{(k-1)}(U_{k-1} = 1)}{Q_0^{(k-1)}(U_{k-1} = 1)} \\ &\geq 1 \\ &\geq \frac{1 - Q_1^{(k-1)}(U_{k-1} = 1)}{1 - Q_0^{(k-1)}(U_{k-1} = 1)} \\ &\geq l^*(U_{k-1} = 0). \end{aligned}$$

For any $l^*(U_{k-1} = 0) < t < l^*(U_{k-1} = 1)$,

$$\begin{aligned} Q_0^{(k)}(l^*(U_{k-1}) > t) &= Q_0^{(k)}(U_{k-1} = 1) \\ &\geq P_0^{(k)}(U_{k-1} = 1) \\ &= P_0^{(k)}(l^*(U_{k-1}) > t), \end{aligned}$$

where the inequality follows from the induction hypothesis. Note that $l^*(U_{k-1})$ only takes the two values $l^*(U_{k-1} = 0)$ and $l^*(U_{k-1} = 1)$, so for any t not in between these values, the equality $Q_0^{(k)}(l^*(U_{k-1}) > t) = P_0^{(k)}(l^*(U_{k-1}) > t)$ is trivially true. Therefore, $l^*(U_{k-1})$ is stochastically larger under $Q_0^{(k)}$.

From Lemma 3.2, the product of $l^*(Y_k)$ and $l^*(U_{k-1})$ is stochastically larger under $Q_0^{(k)}$ than under any other distribution $P_0^{(k)}$ as well. Therefore, we have

$$\begin{aligned} Q_0^{(k)}(U_k = 1) &= Q_0^{(k)}(l^*(U_{k-1}, Y_k) > t_k) \\ &\geq P_0^{(k)}(l^*(U_{k-1}, Y_k) > t_k) \\ &= P_0(U_k = 1). \end{aligned}$$

The proof for the missed detection probability inequality (3.3) is similar, and the induction is complete. The theorem is now proved. $\square$

Theorem 3.2. *Let $\phi_*^{(N)}$ be an optimal sequence of decision rules such that*

$$\phi_*^{(N)} = \arg \min_{\phi^{(N)}} P_{e,N}(\phi^{(N)}, Q_0^{(N)}, Q_1^{(N)}).$$

Then, $\phi_^{(N)}$ minimizes $P_N^{DD}(\cdot)$ in (2.1). Similarly, for each $k \geq 1$, define $\psi_{*,k}$ recursively as*

$$\psi_{*,k} = \arg \min_{\psi_k} P_{e,k}((\psi_*^{(k-1)}, \psi_k), Q_0^{(k)}, Q_1^{(k)}),$$

where $\psi_{,0}$ is ignored. Then, $\psi_{*,k}$ minimizes $P_k^{SL}(\cdot | \psi_*^{(k-1)})$ in (2.3) for all $k \geq 1$.*

Proof. In [58], it was shown that $\phi_*^{(N)}$ is a sequence of likelihood ratio tests based on $(Q_0^{(N)}, Q_1^{(N)})$. Hence, for any sequence of decision rules $\phi^{(N)}$, we have from Theorem 3.1,

$$\begin{aligned} \sup_{(P_0^{(N)}, P_1^{(N)}) \in \mathcal{P}_0^{(N)} \times \mathcal{P}_1^{(N)}} P_{e,N}(\phi_*^{(N)}, P_0^{(N)}, P_1^{(N)}) &= P_{e,N}(\phi_*^{(N)}, Q_0^{(N)}, Q_1^{(N)}) \\ &\leq P_{e,N}(\phi^{(N)}, Q_0^{(N)}, Q_1^{(N)}) \\ &\leq \sup_{(P_0^{(N)}, P_1^{(N)}) \in \mathcal{P}_0^{(N)} \times \mathcal{P}_1^{(N)}} P_{e,N}(\phi^{(N)}, P_0^{(N)}, P_1^{(N)}) \end{aligned} \quad (3.4)$$

where the inequality in (3.4) follows from the theorem assumption. Therefore, the minimax error in the decentralized detection problem is equal to the minimum error when all the distributions of the private observations are equal to the LFDs. A similar argument holds for the social learning problem in the second part of this theorem. The proof is now complete. $\square$

We remark that Theorem 3.2 can be extended to general tree topologies where the last agent is at the root of the tree, and agent decisions are passed towards the root. This is because in such a topology, the decisions received by each agent are mutually independent. Hence,

$$l^*(Y_k, U_{k_1}, \dots, U_{k_m}) = l^*(Y_k) \prod_{j=1}^m l^*(U_{k_j}),$$

where $k_1, \dots, k_m$ are the agents that agent k is receiving decisions from. The rest of the proof then proceeds by showing that Theorem 3.1 holds for all agents in the tree topology by induction on the distance from the root agent. We can then use this result to prove a variant of Theorem 3.2 for general tree topologies.

3.2 Asymptotic Detection and Social Learning

Theorem 3.2 shows that there is no loss in optimality in both the decentralized detection and social learning problems if agents in a tandem network are restricted to likelihood ratio tests based on the LFDs. It however does not tell us the smallest minimax error probability that is achievable at each agent. In this section, we study the minimax error probability in long tandems under various technical assumptions in order to simplify the problem. In particular, we investigate the conditions under which the minimax error probability converges to zero as the number of agents increases. We first consider the case where every agent has identical uncertainty classes, and provide necessary and sufficient conditions for the minimax error probability to approach zero. We will then proceed to relax our assumptions in two different ways. In the first, we analyze social learning in long tandems where the uncertainty classes can differ. In the second way, we keep the assumption of identical uncertainty classes for each agent, but study the achievable asymptotic minimax error probability when agents do not know their own positions in the tandem. In each of the above scenarios, we examine what happens when agents work together to minimize the asymptotic minimax error probability (decentralized detection) as well as what happens when agent myopically choose to minimize their local error probability (social learning).

3.2.1 Identical Uncertainty Classes

In this subsection, we make the following assumption that the uncertainty classes of every agent are identical.

Assumption 3.1. *For all $k \geq 1$, $\mathcal{P}_{0,k} = \mathcal{P}_0$, $\mathcal{P}_{1,k} = \mathcal{P}_1$, and the LFDs of each agent's uncertainty classes are (Q_0, Q_1) .*

The following two results give necessary and sufficient conditions for the minimax error probability to approach zero under the decentralized detection and social learning frameworks, respectively. To do so, we make use of the following lemma.

Lemma 3.2.1. *For any probability distribution $Q_1 \in \mathcal{R}$,*

$$\lim_{t \rightarrow \infty} \frac{\log(1 - Q_1(l^*(Y) \geq t))}{\log(1 - \frac{Q_1(l^*(Y) \geq t)}{t})} = \infty.$$

Proof. Let $g(x) = \log(1 - x)$, a concave function. For $0 < x < 1$, we have $g(x) < 0$.

For any fixed $t > 0$, by Jensen's Inequality,

$$\frac{1}{t}g(x) + \frac{t-1}{t}g(0) \leq g\left(\frac{x}{t}\right).$$

As $g(0) = 0$, we have

$$\frac{g(x)}{g\left(\frac{x}{t}\right)} \geq t.$$

Letting $x = Q_1(l^*(Y) \geq t)$,

$$\frac{\log(1 - Q_1(l^*(Y) \geq t))}{\log(1 - \frac{Q_1(l^*(Y) \geq t)}{t})} \geq t.$$

Hence,

$$\lim_{t \rightarrow \infty} \frac{\log(1 - Q_1(l^*(Y) \geq t))}{\log(1 - \frac{Q_1(l^*(Y) \geq t)}{t})} = \infty.$$

□

Now, we can state our theorem.

Theorem 3.3 (Decentralized detection). *Suppose that Assumption 3.1 holds, and that the decision rules for every agent are chosen so as to minimize $P_N^{DD}(\phi^{(N)})$ in (2.1). Then $P_N^{DD}(\phi^{(N)}) \rightarrow 0$ as $N \rightarrow \infty$ if and only if the log-likelihood ratio of Q_0 versus Q_1 is almost surely unbounded from either above or below.*

Proof. Without loss of generality, we assume that the log-likelihood ratio of Q_0 and Q_1 is almost surely unbounded from above.

Using the scheme proposed in [51], we can show that we can make the maximum error arbitrarily small as the number of agents tends to infinity, thus showing sufficiency. We denote this scheme as Φ_δ . The decision rules of Φ_δ are as follows:

For a given N^* and a threshold t ,

$$U_1 = \begin{cases} 0 & \text{for } l^*(Y_1) < t \\ 1 & \text{for } l^*(Y_1) \geq t. \end{cases}$$

For $1 < k < N^*$,

$$U_k = \begin{cases} 0 & \text{for } l^*(Y_k) < t \text{ and } U_{k-1} = 0 \\ 1 & \text{otherwise.} \end{cases}$$

For $k \geq N^*$,

$$U_k = U_{k-1}.$$

Note that Φ_δ is a sequence of likelihood ratio tests between Q_1 and Q_0 . It was shown in [51] that by choosing a suitable t and N^* , we can get an arbitrarily small error rate if $P_1^{(N)} = Q_1^N$. To see this, consider a point on the receiver operating characteristic (ROC) curve of an agent using the likelihood ratio test between Q_0 and Q_1 , $(Q_0(l^*(Y) \geq t), Q_1(l^*(Y) \geq t))$. Since the log-likelihood ratio of Q_0 and Q_1 is unbounded from above, we have the initial slope of the ROC curve of the agent being infinity. Because of this and the concavity of the ROC curve, for sufficiently large t we have

$$Q_0(l^*(Y) \geq t) < \frac{Q_1(l^*(Y) \geq t)}{t}.$$

The asymptotic miss detection probability using the decision rules outlined above is thus

$$\begin{aligned} \lim_{N \rightarrow \infty} P_{M,N}(\Phi_\delta, Q_1^N) &= Q_1(U_{N^*} = 0) \\ &= (1 - Q_1(l^*(Y) \geq t))^{N^*}. \end{aligned}$$

Similarly, the asymptotic false alarm probability is

$$\begin{aligned} \lim_{N \rightarrow \infty} P_{F,N}(\Phi_\delta, Q_0^N) &= 1 - (1 - Q_0(l^*(Y) \geq t))^{N^*} \\ &< 1 - \left(1 - \frac{Q_1(l^*(Y) \geq t)}{t}\right)^{N^*}. \end{aligned}$$

Choose an arbitrary $\delta > 0$. To get the asymptotic miss detection probability smaller than δ , we have

$$(1 - Q_1(l^*(Y) \geq t))^{N^*} < \delta$$

and so

$$N^* > \frac{\log(\delta)}{\log(1 - Q_1(l^*(Y) \geq t))}. \quad (3.5)$$

Similarly, to get the asymptotic false alarm probability smaller than δ , we have

$$\frac{\log(1 - \delta)}{\log(1 - \frac{Q_1(l^*(Y) \geq t)}{t})} > N^*. \quad (3.6)$$

From Lemma 3.2.1, by choosing a large enough t , we have

$$\frac{\log(1 - Q_1(l^*(Y) \geq t))}{\log(1 - \frac{Q_1(l^*(Y) \geq t)}{t})} > 2 \frac{\log(\delta)}{\log(1 - \delta)}$$

and so

$$\frac{\log(1 - \delta)}{\log(1 - \frac{Q_1(l^*(Y) \geq t)}{t})} > 2 \frac{\log(\delta)}{1 - \log(Q_1(l^*(Y) \geq t))}.$$

For a sufficiently large t ,

$$\frac{\log(\delta)}{1 - \log(Q_1(l^*(Y) \geq t))} > 1.$$

Hence we can find an integer N^* that lies between the bounds in (3.5) and (3.6).

Then,

$$\begin{aligned} \lim_{N \rightarrow \infty} \inf_{\phi^{(N)}} P_N^{\text{DD}}(\phi^{(N)}) &\leq \lim_{N \rightarrow \infty} P_N^{\text{DD}}(\Phi_\delta) \\ &= \lim_{N \rightarrow \infty} \sup_{P_0^{(N)} \in \mathcal{P}_0^{(N)}} \pi_0 P_{F,N}(\Phi_\delta, P_0^{(N)}) + \pi_1 \sup_{P_1^{(N)} \in \mathcal{P}_1^{(N)}} P_{M,N}(\Phi_\delta, P_1^{(N)}) \\ &= \lim_{N \rightarrow \infty} \pi_0 P_{F,N}(\Phi_\delta, Q_0^{(N)}) + \pi_1 P_{M,N}(\Phi_\delta, Q_1^{(N)}) \\ &< \pi_0 \delta + \pi_1 \delta \\ &= \delta, \end{aligned} \quad (3.7)$$

where (3.7) is due to Theorem 3.1. Since δ was arbitrarily chosen, we can make the maximum error arbitrarily small.

To show necessity, assume that the log-likelihood of Q_1 and Q_0 is bounded. Then

$$\lim_{N \rightarrow \infty} \inf_{\phi^{(N)}} \sup_{(P_0^{(N)}, P_1^{(N)})} P_{e,N}(\phi^{(N)}, P_0^{(N)}, P_1^{(N)}) \geq \lim_{N \rightarrow \infty} \inf_{\phi^{(N)}} P_{e,N}(\phi^{(N)}, Q_0^N, Q_1^N),$$

which is known to be bounded above zero as $N \rightarrow \infty$ as the log-likelihood of Q_1 and Q_0 is bounded from both above and below.

□

If all agents are rational, it is very unlikely that the log-likelihood of the uncertainty classes are unbounded. In practice, there will be some agents who will place a very high weight on their own personal observations and may discount the decision of the previous agent. For these agents, the log-likelihood of the uncertainty classes are effectively unbounded. If no such agent exists, at some point far enough down the chain, each agent will deem the decisions of all previous agents to carry more weight than their personal observations, and will just relay the previous agent's decision to the next one.

Theorem 3.4 (Social learning). *Suppose that Assumption 3.1 holds, and that the decision rule for each agent is chosen sequentially so as to minimize $P_k^{SL}(\phi_k | \phi^{(k-1)})$ in (2.3). Then $P_k^{SL}(\phi_k | \phi^{(k-1)}) \rightarrow 0$ as $k \rightarrow \infty$ if and only if the log-likelihood ratio of Q_0 versus Q_1 is almost surely unbounded from both above and below.*

Proof. It was demonstrated in Proposition 3 of [51] that using social learning decision rules, the error probability in a tandem network where all agents have the same observation distributions will converge to zero if and only if the log-likelihood ratios between the two probability distributions is unbounded from both above and below. The sufficiency of the given condition then follows immediately. To show that it is also necessary, observe that if the log-likelihood ratio of Q_1 versus Q_0 is bounded from either above or below respectively, and Theorem 3.2 and Proposition 3 of [51] implies that $P_N^{DD}(\phi^{(N)})$ is bounded away from zero. The proof is now complete. $\square$

For the ϵ -contamination uncertainty classes defined in (2.9) with $\epsilon_{i,k} = \epsilon_i$ for $i = 0, 1$ and all $k \geq 1$, Theorem 3.3 implies that learning can only occur under decentralized detection if at least one of the following is true:

- 1) $\epsilon_0 = 0$ and the log-likelihood ratio of P_1^* versus P_0^* is almost surely unbounded from above,
- 2) $\epsilon_1 = 0$ and the log-likelihood ratio of P_1^* versus P_0^* is almost surely unbounded from below.

Similarly, Theorem 3.4 implies that social learning can only occur if $\epsilon_0 = \epsilon_1 = 0$ and the log-likelihood ratio of P_1^* versus P_0^* is almost surely unbounded. This means that in many practical cases, since there is often some doubt about each agent's private signal under either state of the world, asymptotic learning cannot occur when the uncertainty classes are identical.

3.2.2 Varying uncertainty classes for social learning

In this subsection, we relax the assumption of identical uncertainty classes for all agents in the previous subsection, and study the effect of varying uncertainty classes on the asymptotic error probability in the social learning framework. We make the following assumption.

Assumption 3.2. *Each agent $k \geq 1$ knows its own uncertainty class, as well as those of its predecessors, and its position in the tandem network.*

We will show that under these assumption, learning the true hypothesis can occur only if there exist infinite subsequences $(k_n)_{n \geq 1}$ and $(j_n)_{n \geq 1}$ (which may be distinct) such that $\log(l^*(Y_{k_n}))$ is almost surely unbounded from above and $\log(l^*(Y_{j_n}))$ is almost surely unbounded from below. Furthermore, we will show that learning will occur if in addition, both uncertainty classes $\mathcal{P}_{0,k_n}$ and $\mathcal{P}_{1,j_n}$ converge to singletons.

Note that if only some, but not all of the agents know their position in the network, then under the decentralized detection framework, it is identical to all agents knowing their positions, as those which do not know can simply pass on the decision from the previous agent without taking into account their own private observations. However, under the social learning framework, it is difficult to determine a good decision rule for agents who do not know their position in the network, as they will not have enough information regarding what the decision of the previous agent means. We will examine the case where all agents do not know their position in the network later in the chapter.

We first observe that under the social learning framework, agents minimize their local maximum error probability. This implies that the minimax error probability of

each agent k is non-increasing in k . This is because any agent can simply pass on the decision of the previous agent if no other decision rule leads to a decrease in minimax error probability. Note that this means if only some, but not all of the agents know their position in the network, then under the decentralized detection framework, it is identical to all agents knowing their positions.

For ease of notation, we let $Q_{F,k} = P_{F,k}(\phi^{(k)}, Q_0^{(k)})$ and $Q_{M,k} = P_{M,k}(\phi^{(k)}, Q_0^{(k)})$ be the LFD false alarm and miss detection probabilities of agent k respectively, where $\phi^{(k)}$ is the sequence of optimal social learning decision rules that minimizes $P_k^{\text{SL}}(\phi_k | \phi^{(k-1)})$ in (2.3). From [51], it can be shown that the decision rule used by agent k is of the form

$$u_k = \begin{cases} 0 & \text{if } l^*(Y_k) < \frac{\pi_0 Q_{F,k-1}}{\pi_1(1-Q_{M,k-1})} \\ 1 & \text{if } l^*(Y_k) \geq \frac{\pi_0(1-Q_{F,k-1})}{\pi_1 Q_{M,k-1}} \\ u_{k-1} & \text{if } \frac{\pi_0 Q_{F,k-1}}{\pi_1(1-Q_{M,k-1})} \leq l^*(Y_k) < \frac{\pi_0(1-Q_{F,k-1})}{\pi_1 Q_{M,k-1}}. \end{cases} \quad (3.8)$$

We now have the following two theorems which derive necessary and sufficient conditions for asymptotic learning to occur under social learning. To do so, we first prove the following two lemmas.

Lemma 3.2.2. *Suppose that the LFDs for a pair of uncertainty classes $(\mathcal{P}_0, \mathcal{P}_1)$ are (Q_0, Q_1) . Then, for any random variable Y with distributions belonging to these uncertainty classes, we have for all $t > 0$,*

$$Q_1(l^*(Y) \leq t) \leq tQ_0(l^*(Y) \leq t) - \frac{t}{2}Q_0(l^*(Y) \leq t/2),$$

and

$$Q_0(l^*(Y) \geq t) \leq \frac{1}{t}Q_1(l^*(Y) \geq t) - \frac{1}{2t}Q_1(l^*(Y) \geq 2t).$$

Furthermore, the above two inequalities hold when $l^*(Y) \leq t$ and $l^*(Y) \geq t$ are replaced by $l^*(Y) < t$ and $l^*(Y) > t$, respectively throughout.

Proof. To show the first inequality, we observe that

$$\begin{aligned}
Q_1(l^*(Y) \leq t) &= tQ_0(l^*(Y) \leq t) - \int_{x=0}^t (t-x) dQ_0(l^*(Y) = x) \\
&\leq tQ_0(l^*(Y) \leq t) - \int_{x=0}^{\frac{t}{2}} (t-x) dQ_0(l^*(Y) = x) \\
&\leq tQ_0(l^*(Y) \leq t) - \int_{x=0}^{\frac{t}{2}} (t - \frac{t}{2}) dQ_0(l^*(Y) = x) \\
&= tQ_0(l^*(Y) \leq t) - \frac{t}{2}Q_0(l^*(Y) \leq t/2).
\end{aligned}$$

The second inequality follows by interchanging Q_0 and Q_1 , and replacing t by $1/t$. The proof is now complete. $\square$

Lemma 3.2.3. *Suppose that $Q_{F,k} + Q_{M,k}$ is bounded away from zero, and that $\log l^*(Y_k)$ is almost surely unbounded from either above or below. Then for any integer n , there exists some $k' \geq n$ such that*

$$\begin{aligned}
\pi_0 Q_{F,k'} + \pi_1 Q_{M,k'} &\leq \pi_0 Q_{F,k'-1} + \pi_1 Q_{M,k'-1} \\
&\quad - \frac{\pi_1 Q_{M,k'-1}}{2} \cdot Q_{1,k'} \left(l^*(Y_{k'}) \geq \frac{2\pi_0(1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) \\
&\quad - \frac{\pi_0 Q_{F,k'-1}}{2} \cdot Q_{0,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 Q_{F,k'-1}}{2\pi_1(1 - Q_{M,k'-1})} \right).
\end{aligned}$$

Proof. For any n , choose $k' \geq n$ such that $Q_{1,k'} \left(l^*(Y_{k'}) \geq \frac{2\pi_0(1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) > 0$ or $Q_{0,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 Q_{F,k'-1}}{2\pi_1(1 - Q_{M,k'-1})} \right) > 0$. This is possible as we assume either $Q_{F,k}$ or $Q_{M,k}$ is bounded away from 0, and from our lemma assumption, the lower or upper bound

on $l^*(Y_k)$ converges to 0 or ∞ , respectively (cf. (2.10)). Then, we have

$$\begin{aligned}
\pi_0 Q_{F,k'} + \pi_1 Q_{M,k'} &= \pi_0 Q_{F,k'-1} \cdot Q_{0,k'} \left(l^*(Y_{k'}) \geq \frac{\pi_0 Q_{F,k'-1}}{\pi_1 (1 - Q_{M,k'-1})} \right) \\
&\quad + \pi_0 (1 - Q_{F,k'-1}) \cdot Q_{0,k'} \left(l^*(Y_{k'}) \geq \frac{\pi_0 (1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) \\
&\quad + \pi_1 Q_{M,k'-1} \cdot Q_{1,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 (1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) \\
&\quad + \pi_1 (1 - Q_{M,k'-1}) \cdot Q_{1,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 Q_{F,k'-1}}{\pi_1 (1 - Q_{M,k'-1})} \right) \\
&\leq \pi_0 Q_{F,k'-1} \cdot Q_{0,k'} \left(l^*(Y_{k'}) \geq \frac{\pi_0 Q_{F,k'-1}}{\pi_1 (1 - Q_{M,k'-1})} \right) \\
&\quad + \pi_1 Q_{M,k'-1} \cdot Q_{1,k'} \left(l^*(Y_{k'}) \geq \frac{\pi_0 (1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) \\
&\quad - \frac{\pi_1 Q_{M,k'-1}}{2} \cdot Q_{1,k'} \left(l^*(Y_{k'}) \geq \frac{2\pi_0 (1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) \\
&\quad + \pi_1 Q_{M,k'-1} \cdot Q_{1,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 (1 - Q_{F,k'-1})}{\pi_1 Q_{M,k'-1}} \right) \\
&\quad + \pi_0 Q_{F,k'-1} \cdot Q_{0,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 Q_{F,k'-1}}{\pi_1 (1 - Q_{M,k'-1})} \right) \\
&\quad - \frac{\pi_0 Q_{F,k'-1}}{2} \cdot Q_{0,k'} \left(l^*(Y_{k'}) < \frac{\pi_0 Q_{F,k'-1}}{2\pi_1 (1 - Q_{M,k'-1})} \right)
\end{aligned}$$

where the inequality follows from Lemma 3.2.2. The lemma is now proved. $\square$

We can now state our theorem.

Theorem 3.5. *Suppose that Assumption 3.2 holds, and that each agent k in a tandem network chooses decision rule ϕ_k to minimize $P_k^{SL}(\phi_k \mid \phi^{(k-1)})$ in (2.3). Then $P_k^{SL}(\phi_k \mid \phi^{(k-1)}) \rightarrow 0$ as $k \rightarrow \infty$ only if there exist infinite subsequences $(k_n)_{n \geq 1}$ and $(j_n)_{n \geq 1}$ (which may be distinct) such that $\log(l^*(Y_{k_n}))$ is almost surely unbounded from above and $\log(l^*(Y_{j_n}))$ is almost surely unbounded from below. In addition, if $\mathcal{P}_{0,k_n}$ and $\mathcal{P}_{1,j_n}$ each converges to a singleton as $n \rightarrow \infty$, then $P_k^{SL}(\phi_k \mid \phi^{(k-1)}) \rightarrow 0$ as $k \rightarrow \infty$.*

Proof. We will begin by proving that the necessary condition is required for asymptotic learning to occur. Assume that such subsequences do not exist. Without loss of generality, assume that $\log l^*(Y_k)$ is bounded above zero. Let $\inf_{k,y} l^*(Y_k = y) = \delta > 0$.

Assume now that $\pi_0 Q_{F,k} + \pi_1 Q_{M,k} \rightarrow 0$ as $k \rightarrow 0$. Then both $Q_{F,k} \rightarrow 0$ and $Q_{M,k} \rightarrow 0$. Choose j such that for all $k \geq j$, $Q_{F,k} + Q_{M,k} \leq 1$ and $\pi_0 Q_{F,k} + \delta \pi_1 Q_{M,k} \leq \delta \pi_1$. We have from (3.8),

$$\begin{aligned} Q_{F,j+1} &= Q_{F,j} \cdot Q_{0,j+1} \left(l^*(Y_{j+1}) \geq \frac{\pi_0 Q_{F,j}}{\pi_1(1 - Q_{M,j})} \right) \\ &\quad + (1 - Q_{F,j}) \cdot Q_{0,j+1} \left(l^*(Y_{j+1}) \geq \frac{\pi_0(1 - Q_{F,j})}{\pi_1 Q_{M,j}} \right) \\ &\geq Q_{F,j} \cdot Q_{0,j+1} (l^*(Y_{j+1}) \geq \delta) \\ &= Q_{F,j}. \end{aligned}$$

By induction, $Q_{F,k} \geq Q_{F,j}$ for all $k \geq j$, a contradiction. Hence, the necessary condition is required for asymptotic learning to occur.

We will now show the sufficient condition. If $Q_{F,k} + Q_{M,k} \rightarrow 0$, the theorem holds trivially. Therefore we assume otherwise. Since $P_k^{\text{SL}}(\phi_k \mid \phi^{(k-1)})$ is bounded and non-increasing, it converges. Suppose that $\pi_0 Q_{F,k} + \pi_1 Q_{M,k} \rightarrow C$, for some $C > 0$. Either $\limsup_{k_n} (Q_{F,k_n}) > 0$ or $\limsup_{k_n} (Q_{M,k_n}) > 0$. Without loss of generality, let $\limsup_{k_n} (Q_{F,k_n}) = C' > 0$. Then there exists a subsequence of agents $(k_{n_\alpha})_{\alpha \geq 1}$ such that $Q_{F,k_{n_\alpha}} \rightarrow C'$. Choose N such that $Q_{F,k_{n_{\alpha'}}} > \frac{C'}{2}$ for all $\alpha' \geq \alpha$. Then, from Lemma 3.2.2 and Lemma 3.2.3, we have

$$\begin{aligned} P_{k_{n_{\alpha}+1}}^{\text{SL}}(\phi_{k_{n_{\alpha}+1}} \mid \phi^{(k_{n_{\alpha}+1}-1)}) &\leq P_{k_{n_{\alpha}}+1}^{\text{SL}}(\phi_{k_{n_{\alpha}}+1} \mid \phi^{(k_{n_{\alpha}})}) \\ &\leq P_{k_{n_{\alpha}}}^{\text{SL}}(\phi_{k_{n_{\alpha}}} \mid \phi^{(k_{n_{\alpha}}-1)}) - \frac{\pi_0 Q_{F,k_{n_{\alpha}}-1}}{2} \\ &\quad \cdot Q_{0,k_{n_{\alpha}}} \left(l^*(Y_{k_{n_{\alpha}}}) < \frac{\pi_0 Q_{F,k_{n_{\alpha}}-1}}{2\pi_1(1 - Q_{M,k_{n_{\alpha}}-1})} \right) \\ &\leq P_{k_{n_{\alpha}}}^{\text{SL}}(\phi_{k_{n_{\alpha}}} \mid \phi^{(k_{n_{\alpha}}-1)}) - \frac{\pi_0 C'}{4} \cdot Q_{0,k_{n_{\alpha}}} \left(l^*(Y_{k_{n_{\alpha}}}) < \frac{C'}{4\pi_1} \right). \end{aligned}$$

From our theorem assumption that $\mathcal{P}_{0,k_n}$ converges to a singleton, by letting $\alpha \rightarrow \infty$ on both sides of the above inequality, we obtain a contradiction. The proof of the theorem is now complete. $\square$

Theorem 3.5 provides separate necessary and sufficient conditions for asymptotic social learning to occur. For asymptotic learning to occur, one would expect that the observations of the agents become more and more informative as k increases.

This is why the log-likelihood ratio of the LFDs must be almost surely unbounded in the two subsequences in Theorem 3.5. However, without any other assumptions on the uncertainty classes, it is possible to construct uncertainty classes for each agent such that the LFD log-likelihood ratio becomes unbounded with k , but the observation becomes less and less informative. If we add the assumption that the uncertainty classes converge to singletons, then the above construction is not possible, and asymptotic learning is guaranteed.

For the ϵ -contamination uncertainty classes in (2.9), the above necessary and sufficient conditions are identical.

Corollary 3.1. *If the uncertainty classes in Theorem 3.5 are ϵ -contamination uncertainty classes as defined in (2.9), then $P_k^{SL}(\phi_k | \phi^{(k-1)}) \rightarrow 0$ as $k \rightarrow \infty$ if and only if the log-likelihood ratio of P_0^* and P_1^* is almost surely unbounded from both above and below, and there exist infinite subsequences $(k_n)_{n \geq 1}$ and $(j_n)_{n \geq 1}$ such that $\epsilon_{0,k_n} \rightarrow 0$ and $\epsilon_{1,j_n} \rightarrow 0$ as $n \rightarrow \infty$.*

Proof. From (2.10), a subsequence of agents with $\log l^*(Y_{j_n})$ almost surely unbounded from below would require the log-likelihood ratio of P_1^* versus P_0^* being unbounded, as well as $c' \rightarrow 0$, which is equivalent to $\epsilon_{1,j_n} \rightarrow 0$. Similarly, we also need another subsequence where $\epsilon_{0,k_n} \rightarrow 0$. Since this implies that both the uncertainty classes converge to the distributions P_1^* and P_0^* respectively, this is also a sufficient condition. $\square$

We observe that agents in a tandem network in the social learning framework have total error probabilities at least that of agents adopting decentralized detection rules. Hence, Theorem 3.5 also provides a sufficient condition for the minimax error probability (2.1) under a decentralized detection framework to converge to zero.

3.2.3 Unknown agent positions

In a social network, users have to make their decisions not knowing how many hops information has been propagated from a source node. We model this in a tandem

network by assuming that each agent has no knowledge of its position in the network. We make the following assumption, in addition to Assumption 3.1, in this subsection.

Assumption 3.3. *Every agent $k > 1$ uses the same decision rule.*

Except for agent 1 (which knows its position in the tandem because it does not receive any preceding messages), every other agent in the tandem does not know its own position, and has access to exactly the same information when it comes to choosing a decision rule, which is based solely on the uncertainty class of the agent. Therefore, it is natural to make Assumption 3.3.

Because of Assumption 3.3, any sequence of k agent decision rules has the form $\phi^{(k)} = (\phi_1, \phi^{k-1})$, where ϕ_1 is the decision rule used by the first agent, and ϕ is the decision rule used by every other agent with $\phi^{k-1} = (\phi, \dots, \phi)$ consisting of $k - 1$ copies of ϕ . For simplicity, and by abusing notation, we replace $\phi^{(k)}$ in our notations by (ϕ_1, ϕ) in the sequel. In the following, we consider two different scenarios.

Minimizing asymptotic error

We consider the case where agents are collaborating to minimize the asymptotic error. This might occur when there is a chain of agents trying to relay some information to a fusion center, but each agent is unsure of how many other agents there are between itself and the fusion center. For a given decision rule ϕ , the asymptotic maximum error probability is given by

$$P_\infty^{\text{DD}}(\phi) = \lim_{k \rightarrow \infty} P_k^{\text{DD}}(\phi_1, \phi), \quad (3.9)$$

where we have implicitly assumed that P_∞^{DD} does not depend on ϕ_1 . This assumption is valid, as shown in the next theorem, which also provides a characterization for the optimal ϕ that achieves the asymptotic minimax error probability, defined as

$$\inf_{\phi} P_\infty^{\text{DD}}(\phi).$$

In the previous sections, we had no need to consider randomized decision rules. This is because under the decentralized detection framework, the final error probability under a randomized sequence of decision rules is no less than the minimum final

error probability of each of the respective deterministic sequences of decision rules. Similarly, under the social learning framework, the error probability of an agent using a randomized decision rule is no less than the minimum error probability under each of the deterministic rules. However, this property does not hold for the asymptotic maximum error probability. A randomized version of two deterministic decision rules may yield a lower asymptotic maximum error probability than either of the two deterministic rules. Thus, we introduce the randomized likelihood ratio test:

$$u_k = \begin{cases} 0 & \text{if } l^*(Y_k) < t_1 \\ A & \text{if } l^*(Y_k) = t_1 \\ u_{k-1} & \text{if } t_1 < l^*(Y_k) < t_0 \\ B & \text{if } l^*(Y_k) = t_0 \\ 1 & \text{if } l^*(Y_k) > t_0, \end{cases} \quad (3.10)$$

where A is equal to 0 with probability p and equal to u_{k-1} with probability $1 - p$, B is equal to u_{k-1} with probability q and equal to 1 with probability q , and t_1 , t_0 , p and q are constants to be determined.

We now show how to obtain the minimax asymptotic error probability. To do so, we start with the following two lemmas.

Lemma 3.3. *Suppose that Assumptions 3.1 and 3.3 hold. Then,*

$$\begin{aligned} \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi), Q_0^k, Q_1^k) &= \frac{\pi_0 Q_0(\phi(Y_1, 0) = 1)}{Q_0(\phi(Y_1, 0) = 1) + Q_0(\phi(Y_1, 1) = 0)} \\ &\quad + \frac{\pi_1 Q_1(\phi(Y_1, 1) = 0)}{Q_1(\phi(Y_1, 1) = 0) + Q_1(\phi(Y_1, 0) = 1)}. \end{aligned}$$

Proof. We have the two following recurrence relations:

$$\begin{aligned} P_{F,k}((\phi_1, \phi), Q_0^k) &= Q_0^k(U_k = 1) \\ &= Q_0^{k-1}(U_{k-1} = 0) \cdot Q_0(U_k = 1 \mid U_{k-1} = 0) \\ &\quad + Q_0^{k-1}(U_{k-1} = 1) \cdot Q_0(U_k = 1 \mid U_{k-1} = 0) \\ &= (1 - Q_0^{k-1}(U_{k-1} = 1)) \cdot Q_0(\phi(Y_k, 0) = 1) \\ &\quad + Q_0^{k-1}(U_{k-1} = 1) \cdot Q_0(\phi(Y_k, 1) = 1) \\ &= P_{F,k-1}((\phi_1, \phi), Q_0^{k-1}) \cdot [Q_0(\phi(Y_1, 1) = 1) - Q_0(\phi(Y_1, 0) = 1)] \\ &\quad + Q_0(\phi(Y_1, 0) = 1), \end{aligned} \quad (3.11)$$

and

$$\begin{aligned}
P_{M,k}((\phi_1, \phi), Q_0^k) &= Q_1^k(U_k = 0) \\
&= Q_1^{k-1}(U_{k-1} = 0) \cdot Q_1(U_k = 0 \mid U_{k-1} = 0) \\
&\quad + Q_1^{k-1}(U_{k-1} = 1) \cdot Q_1(U_k = 0 \mid U_{k-1} = 1) \\
&= Q_1^{k-1}(U_{k-1} = 0) \cdot Q_1(\phi(Y_k, 0) = 0) \\
&\quad + (1 - Q_1^{k-1}(U_{k-1} = 0)) \cdot Q_0(\phi(Y_k, 1) = 0) \\
&= P_{M,k-1}((\phi_1, \phi), Q_1^{k-1}) \cdot [Q_1(\phi(Y_1, 1) = 1) - Q_1(\phi(Y_1, 0) = 1)] \\
&\quad + Q_1(\phi(Y_1, 0) = 0),
\end{aligned}$$

The first recurrence relation converges linearly to

$$\frac{Q_0(\phi(Y_1, 0) = 1)}{Q_0(\phi(Y_1, 0) = 1) + Q_0(\phi(Y_1, 1) = 0)}, \quad (3.13)$$

and the second recurrence relation converges linearly to

$$\frac{\pi_1 Q_1(\phi(Y_1, 1) = 0)}{Q_1(\phi(Y_1, 1) = 0) + Q_1(\phi(Y_1, 0) = 1)}. \quad (3.14)$$

Hence, the proof is complete. $\square$

Next, we show that if the observation distributions are all drawn from the LFDs, then the decision rule that minimizes that asymptotic error probability is a randomized likelihood ratio test between Q_0 and Q_1 .

Lemma 3.4. *Suppose that Assumptions 3.1 and 3.3 hold. For any ϕ_1 , let*

$$\phi_* = \arg \min_{\phi} \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi), Q_0^k, Q_1^k).$$

Then, there is no loss in optimality if ϕ_ is restricted to be a randomized likelihood ratio test between Q_0 and Q_1 .*

Proof. Assume that ϕ_* is not a randomized likelihood ratio test between Q_0 and Q_1 . For any agent $k \geq 2$, consider a randomized likelihood ratio test ϕ' in the form presented in (3.10), where t_0, t_1, p, q are chosen such that

$$\begin{aligned}
Q_0(\phi'(Y_k, 0) = 1) &= Q_0(l^*(Y_k) \geq t_0) \\
&= Q_0(\phi_*(Y_k, 0) = 1)
\end{aligned}$$

$$\begin{aligned}
Q_1(\phi'(Y_k, 1) = 0) &= Q_1(l^*(Y_k) < t_1) \\
&= Q_1(\phi_*(Y_k, 1) = 0).
\end{aligned}$$

From the Neyman-Pearson lemma, we then have

$$\begin{aligned} Q_1(\phi'(Y_k, 0) = 0) &= Q_1(l^*(Y_k) < t_0) \\ &\leq Q_1(\phi_*(Y_k, 0) = 0) \end{aligned}$$

$$\begin{aligned} Q_0(\phi'(Y_k, 1) = 1) &= Q_0(l^*(Y_k) < t_1) \\ &\leq Q_0(\phi_*(Y_k, 1) = 1). \end{aligned}$$

Hence, from (3.11) and (3.12), it is clear that for any ϕ_1 and any $k \geq 1$, we have

$$P_{F,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) \geq P_{F,k}((\phi_1, \phi'), Q_0^k, Q_1^k),$$

and

$$P_{M,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) \geq P_{M,k}((\phi_1, \phi'), Q_0^k, Q_1^k),$$

and hence

$$P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) \geq P_{e,k}((\phi_1, \phi'), Q_0^k, Q_1^k).$$

By choosing suitable values of t_0, t_1, p, q , we can set $Q_0(\phi(Y_k, 0) = 1)$ and $Q_0(\phi(Y_k, 1) = 1)$ to any value between 0 and 1. Similarly, we can also set $Q_1(\phi(Y_k, 1) = 0)$ and $Q_1(\phi(Y_k, 0) = 0)$ to any value between 0 and 1. Hence,

$$\min_{\phi} \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi), Q_0^k, Q_1^k)$$

is attainable, and the decision rule used to attain it can be assumed to be in the form of a randomized likelihood ratio test. $\square$

The proof of Lemma 3.4 shows that the asymptotic minimax error probability is attainable even when the likelihood ratio of Y_k is not continuous. To avoid cumbersome notation, for the rest of the thesis, we assume that the likelihood ratio of Y_k is continuous. It is easy to extend our results if this is not the case.

We can now prove the following theorem, which provides an expression for the minimax asymptotic error.

Theorem 3.6. *Suppose that Assumptions 3.1 and 3.3 hold, and let ϕ_* be the randomized likelihood ratio test such that*

$$\phi_* = \arg \min_{\phi} \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi), Q_0^k, Q_1^k).$$

Then,

$$\inf_{\phi} P_{\infty}^{DD}(\phi) = \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k).$$

Proof. From Theorem 3.1, we have

$$\begin{aligned}\lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) &= \lim_{k \rightarrow \infty} \sup_{(P_0^{(k)}, P_1^{(k)}) \in \mathcal{P}_0^k \times \mathcal{P}_1^k} P_{e,k}((\phi_1, \phi_*), P_0^{(k)}, P_1^{(k)}) \\ &= P_\infty^{\text{DD}}(\phi_*).\end{aligned}$$

By the theorem assumption, for any decision rule ϕ we have

$$\lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) \leq \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi), Q_0^k, Q_1^k).$$

Hence, for any decision rule ϕ , we have

$$\begin{aligned}P_\infty^{\text{DD}}(\phi) &= \lim_{k \rightarrow \infty} \sup_{P_0^{(k)} \in \mathcal{P}_0^k, P_1^{(k)} \in \mathcal{P}_1^k} P_{e,k}((\phi_1, \phi), P_0^{(k)}, P_1^{(k)}) \\ &\geq \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi), Q_0^k, Q_1^k) \\ &\geq \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) \\ &= P_\infty^{\text{DD}}(\phi_*).\end{aligned}$$

The proof is now complete. $\square$

Theorem 3.6 states that each agent should find the decision rule to optimize the asymptotic minimax error as if its observations were distributed according to the LFDs of the uncertainty class. This is consistent with our results when agents do know their positions (Theorem 3.1). However, in general, the exact threshold values for ϕ_* are difficult to find analytically, but can be computed using numerical methods. Together with Lemma 3.3, since the uncertainty classes $\mathcal{P}_{0,k}$ and $\mathcal{P}_{1,k}$ are disjoint for all k , Theorem 3.6 shows that asymptotically learning the true hypothesis is impossible when agents do not know their own positions and also provides an expression for the asymptotic minimax error.

Minimizing error of current agent

We now assume that every agent is acting to minimize its local minimax error probability, and that each agent, except the first, does not know which position it is in. This is true in general for most social networks, where it is difficult to find the root of any information spread. Hence, users typically would not know how many hops information has been propagated from its source.

We find the decision rule to minimize the maximum error probability by allowing each agent to consider the maximum error probability for every possible position it might be in, then finding the decision rule that minimizes this value. Like in the previous subsection, we also assume that Assumptions 3.1 and 3.3 are in effect. More specifically, for each agent $k \geq 2$, we wish to find ϕ that minimizes

$$P_{\max}^{\text{SL}}(\phi) = \sup_{k \geq 2} P_k^{\text{SL}}(\phi \mid (\phi_1, \phi)), \quad (3.15)$$

where ϕ_1 is the optimal decision rule that minimizes $P_1^{\text{SL}}(\cdot)$. We will show that for the optimal decision rule ϕ , the maximum error probability occurs either in the second position or at the asymptotic limit, as defined in (3.9). We make use of the following lemma.

Lemma 3.2.4. *Suppose that Assumptions 3.1 and 3.3 hold. There is no loss of optimality if the optimal decision rule ϕ_* for all agents $k \geq 2$ is restricted to be a randomized likelihood ratio test between Q_0 and Q_1 . Furthermore, we have*

$$P_{\max}^{\text{SL}}(\phi_*) = \sup_{k \geq 2} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k).$$

Proof. The proof is similar to that of Lemma 3.4. From the proof of Lemma 3.4 it is shown that if ϕ_* is not a randomized likelihood ratio test between Q_0 and Q_1 , there is some decision rule ϕ'_* which is a randomized likelihood ratio test between Q_0 and Q_1 such that for any $k \geq 2$, $P_0^{(k)}, P_1^{(k)}$,

$$P_{e,k}((\phi_1, \phi_*), P_0^{(k)}, P_1^{(k)}) \geq P_{e,k}((\phi_1, \phi'_*), P_0^{(k)}, P_1^{(k)}),$$

and so there is no loss in optimality in assuming that ϕ_* is a likelihood ratio test between Q_0 and Q_1 .

From Theorem 3.1, if ϕ_* is a randomized likelihood ratio test between Q_0 and Q_1 , then for any $k \geq 2$, $P_0^{(k)}, P_1^{(k)}$, we have

$$P_{e,k}((\phi_1, \phi_*), P_0^{(k)}, P_1^{(k)}) \leq P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k).$$

Hence, we obtain

$$\begin{aligned} P_{\max}^{\text{SL}}(\phi_*) &= \sup_{k \geq 2} \sup_{P_0 \in \mathcal{P}_0, P_1 \in \mathcal{P}_1} P_{e,k}((\phi_1, \phi_*), P_0^{(k)}, P_1^{(k)}) \\ &= \sup_{k \geq 2} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k), \end{aligned}$$

and the lemma is proved. $\square$

We can now state our theorem.

Theorem 3.7. *Suppose that Assumptions 3.1 and 3.3 hold. Then, $\phi_* = \arg \min_{\phi} P_{\max}^{SL}(\phi)$ is a randomized likelihood ratio test between the LFDs Q_0 and Q_1 , and*

$$P_{\max}^{SL}(\phi_*) = \max \{P_2^{SL}(\phi_* | \phi_1), P_{\infty}^{DD}(\phi_*)\},$$

where $P_{\infty}^{DD}(\phi_*)$ is as defined in (3.9).

Proof. The decision rule of agent 1, ϕ_1 , is of the form

$$u_1 = \begin{cases} 0 & \text{if } l^*(Y_1) < \frac{\pi_0}{\pi_1} \\ 1 & \text{if } l^*(Y_1) \geq \frac{\pi_0}{\pi_1}. \end{cases}$$

Hence, the minimax error probability for the first agent is $\pi_0 Q_0(l^*(Y_1) \geq \frac{\pi_0}{\pi_1}) + \pi_1 Q_1(l^*(Y_1) \geq \frac{\pi_0}{\pi_1})$, which is at least $\inf_{\phi} P_{\max}^{SL}(\phi)$. This is because using the trivial rule of ignoring one's own private observation and simply passing on the decision of the previous agent, we get a constant maximum error probability of $\pi_0 Q_0(l^*(Y_1) \geq \frac{\pi_0}{\pi_1}) + \pi_1 Q_1(l^*(Y_1) \geq \frac{\pi_0}{\pi_1})$. Hence, we have

$$\inf_{\phi} \sup_{k \geq 2} P_k^{SL}(\phi | (\phi_1, \phi)) \leq \pi_0 Q_0(l^*(Y_1) \geq \frac{\pi_0}{\pi_1}) + \pi_1 Q_1(l^*(Y_1) \geq \frac{\pi_0}{\pi_1}). \quad (3.16)$$

From (3.16), we have

$$\begin{aligned} P_2^{SL}(\phi_* | \phi_1) &\leq \sup_{k \geq 2} P_k^{SL}(\phi | (\phi_1, \phi_*)) \\ &\leq \pi_0 Q_0(l^*(Y_1) \geq \frac{\pi_0}{\pi_1}) + \pi_1 Q_1(l^*(Y_1) < \frac{\pi_0}{\pi_1}). \end{aligned} \quad (3.17)$$

Let

$$\begin{aligned} \alpha &= P_{F,2}((\phi_1, \phi_*), Q_0^2) - \lim_{k \rightarrow \infty} P_{F,k}((\phi_1, \phi_*), Q_0^k), \\ \beta &= P_{M,2}((\phi_1, \phi_*), Q_1^2) - \lim_{k \rightarrow \infty} P_{M,k}((\phi_1, \phi_*), Q_1^k). \end{aligned}$$

and define the function

$$\begin{aligned} g(x) &= \lim_{k \rightarrow \infty} P_{e,k}((\phi_1, \phi_*), Q_0^k, Q_1^k) + \pi_0 \alpha (Q_0(\phi_*(Y_1, 1) = 1) - Q_0(\phi_*(Y_1, 0) = 1))^{x-1} \\ &\quad + \pi_1 \beta (Q_1(\phi_*(Y_1, 0) = 0) - Q_1(\phi_*(Y_1, 1) = 0))^{x-1}. \end{aligned}$$

From the recurrence relation in Lemma 3.3, we see that for integer values of x , this function gives $P_{e,x}((\phi_1, \phi_*), Q_0^x, Q_1^x)$. Differentiating with respect to x , we observe that

$g(x)$ has at most one stationary point for $x > 0$. From (3.17), we have $g(2) \geq g(1)$. Hence, the stationary point, if any exists, must be a minimum. This means that for $x \geq 2$, the maximum value of $g(x)$ must be either at $x = 2$ or at its asymptotic limit. Thus, the theorem is now proven. $\square$

Finding an analytical form for ϕ_* in Theorem 3.7 is difficult. However, as ϕ_* is known to be a randomized likelihood ratio test of Q_0 and Q_1 , this can be done numerically by minimizing $\max\{P_2^{\text{SL}}(\phi_* | \phi_1), P_\infty^{\text{DD}}(\phi_*)\}$ with respect to the thresholds for ϕ_* .

3.3 Numerical Results

In this section, we provide numerical results to illustrate part of our theoretical contributions in Section 3.2.3 for the case where the uncertainty classes of each agent are ϵ -contamination uncertainty classes in (2.9). Our numerical results show that even if an agent k 's position is unknown, where $k \geq 2$, its optimal decision rule ϕ_* can be chosen to be a randomized likelihood ratio test between the LFDs Q_0 and Q_1 . This is true whether the agent is collaborating with others to minimize the asymptotic error (decentralized detection) or is trying to minimize its own error probability (social learning). We have shown this in Theorems 3.6 and 3.7 respectively. The form of this randomized likelihood ratio test ϕ_* is given in (3.10). Now, we can rewrite this in terms of an optimization problem.

First, given the nominal distributions P_0^* and P_1^* , as well as the contamination values $\epsilon_{0,k} = \epsilon_0$ and $\epsilon_{1,k} = \epsilon_1$ for all $k \geq 1$, we can use a binary search to compute c' and c'' in (2.10). From the definition of the LFDs, we know that the range of thresholds we have to optimize over is bounded between bc' and bc'' , where $b = \frac{1 - \epsilon_1}{1 - \epsilon_0}$. Then, we can derive the LFDs Q_0 and Q_1 and obtain

$$Q_0(\phi_*(Y, 1) = 1) = Q_0(l^*(Y) > t_1) + (1 - p)Q_0(l^*(Y) = t_1)$$

and

$$Q_0(\phi_*(Y, 0) = 1) = Q_0(l^*(Y) > t_0) + (1 - q)Q_0(l^*(Y) = t_0).$$

Similarly, we have

$$Q_1(\phi_*(Y, 1) = 0) = Q_1(l^*(Y) < t_1) + pQ_1(l^*(Y) = t_1)$$

and

$$Q_1(\phi_*(Y, 0) = 0) = Q_1(l^*(Y) < t_0) + qQ_1(l^*(Y) = t_0).$$

In the case where agents collaborate to minimize the asymptotic maximum error P_∞^{DD} , we minimize the expression in Lemma 3.3 to obtain the optimal decision rule ϕ_* . From Theorem 3.6, since ϕ_* is a randomized likelihood ratio test, we can perform the optimization over t_1 , t_0 , p and q in (3.10).

Similarly, in the case where agents minimize their local maximum error probability $P_{\max}^{\text{SL}}$, we minimize the expression in Theorem 3.7 over t_1 , t_0 , p and q .

We now present a numerical example using the exponential distribution. First, we fix P_0^* as an exponential distribution with mean 1 and let P_1^* be an exponential distribution with mean 2. We set $\epsilon_0 = \epsilon_1 = 0.01$, and $\pi_0 = \pi_1 = 0.5$. Then, the optimal decision rule ϕ_1 for agent 1 has the form:

$$u_1 = \begin{cases} 0 & \text{if } l^*(Y_1) < 1 \\ 1 & \text{if } l^*(Y_1) \geq 1. \end{cases}$$

Note that for this case, if $t_1 \neq c'$ and $t_0 \neq c''$, then it does not matter what p and q are. This is because when the nominal distributions are both continuous, $Q_0(l^*(Y_k) = x)$ and $Q_1(l^*(Y_k) = x)$ are both zero unless $x = c'$ or $x = c''$. We plot $P_k^{\text{SL}}(\phi \mid (\phi_1, \phi))$ against k when $\phi = \phi_A$ and ϕ_B , which are randomized likelihood ratio tests of the form (3.10). Both rules have $t_1 = c'$ and $p = 1$, but ϕ_A has $t_0 = 5$ and ϕ_B has $t_0 = 1.1$.

Figure 3.3 shows that the total error probability is decreasing over k for ϕ_A . Hence, the maximum error probability occurs when $k = 2$. For ϕ_B , the maximum occurs at the asymptotic limit $k \rightarrow \infty$. This is in line with our conclusion in Theorem 3.7.

Next, we let P_0^* be an exponential distribution with mean 1 and P_1^* be an exponential distribution with variable mean. We let $\epsilon_0 = \epsilon_1 = 0.01$. We denote

$$\phi_*^{\text{SL}} = \arg \min_{\phi} P_{\max}^{\text{SL}}(\phi)$$

and

$$\phi_*^{\text{DD}} = \arg \min_{\phi} P_{\infty}^{\text{DD}}(\phi),$$

Figure 3-1: Comparison of decision rules.

where $P_{\max}^{SL}(\phi)$ and $P_{\infty}^{DD}(\phi)$ are defined as in (3.9) and (3.15), respectively.

Figure 3.3 shows that as the mean of P_1^* increases, both $P_{\max}^{SL}(\phi_*^{SL})$ and $P_{\infty}^{DD}(\phi_*^{DD})$ decrease. This is intuitive as the Kullback-Leibler divergence of P_1^* from P_0^* increases as the mean of P_1^* increases. As the nominal distributions become easier to differentiate, the asymptotic error probability decreases.

Figure 3-2: Error probability versus mean of P_1^* .

Figure 3.3 shows that both $P_{\max}^{SL}(\phi_*^{SL})$ and $P_{\infty}^{DD}(\phi_*^{DD})$ increases as $\epsilon_0 = \epsilon_1$ increases. For this graph, P_0^* and P_1^* are kept constant as exponential distributions with means 1 and 2 respectively. As expected, as the contamination increases, so does the asymptotic error. When the contamination is large enough, both $P_{\max}^{SL}(\phi_*^{SL})$ and $P_{\infty}^{DD}(\phi_*^{DD})$

Figure 3-3: Error probability versus contamination value.

converge towards 0.5. Furthermore, as contamination increases, the gap between $P_{\max}^{\text{SL}}(\phi_*^{\text{SL}})$ and $P_{\infty}^{\text{DD}}(\phi_*^{\text{DD}})$ decreases. In a network with a lot of uncertainty in the observation distributions, there is not much incentive in trying to get agents to collaborate, as agents selfishly trying to minimize their own error probability leads to similar error performance.

3.4 Summary

In this chapter, we have shown that with some uncertainty in the observation distributions of a tandem network, the minimax error probability is obtained by assuming that each observation is distributed according to the LFDs of the uncertainty class. For a tandem topology with constant contamination in the uncertainty classes under both hypotheses, the asymptotic minimax error probability is bounded above zero. However, the social learning minimax error probability converges to zero if the amount of contamination decays to zero, and the log-likelihood ratio of the nominal distributions are unbounded. The above conclusions hold only if agents know their positions in the tandem. In many social learning scenarios, agents have no knowledge of their positions. We have provided error bounds for this case where the uncertainty classes are identical and agents do not know their positions in the network, and shown that if the identical uncertainty classes' contamination converges to zero, the minimax error probability also converges to this bound.

Lastly, we provide a table summarizing the conditions for asymptotic learning.

	Decentralized Detection	Social Learn- ing
Known position, $\epsilon_0 = \epsilon_1 = 0$	Y	Y
Known position, $\epsilon_0 = 0, \epsilon_1 > 0$	Y	N
Known position, $\epsilon_0, \epsilon_1 > 0$	N	N
Known position, $\epsilon_0, \epsilon_1 \rightarrow 0$	Y	Y
Unknown position	N	N

Table 3.1: Conditions for asymptotic learning

Chapter 4

Learning as a String Submodular Problem

The world of social media is fast and ever-changing. Opinions of users in a social network may not be the same from one day to the next. Hence, it is a challenge to try to keep up with the opinions of the many users in a social network. However, there are many practical applications where getting accurate information from the users as soon as possible is important.

The motivating example we will use for this thesis is in the realm of online advertising. A retailer may be interested in increasing its sales of a new product through launching an advertising campaign, where one of various groups of consumers is targeted at a time over many months. We shall see that this can be modeled as a problem of maximizing the objective value of a string submodular function.

However, maximizing the objective value of a string submodular problem is a difficult task. Instead, it is common to choose actions in a greedy manner. Yet in many practical situations, it is not possible to perfectly predict the rewards of all the possible actions at every single moment. We propose a model where we only are able to choose to observe some of the actions' rewards in the string submodular function, as well as provide algorithms both to choose which actions to observe, as well as to choose which actions to take. We then derive some performance bounds of our algorithms with respect to the optimal string's performance. In the next section,

we will first describe on set submodular functions, which are a special case of string submodular functions.

4.1 Set submodular functions

Let V be a finite set of elements and 2^V be its power set. A set function $f : 2^V \rightarrow \mathbb{R}$ is set submodular if for every $X, Y \subseteq V$ with $X \subseteq Y$ and every $x \in V \setminus Y$, we have

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y) \quad (4.1)$$

We will be looking at monotone set submodular functions, where for every $X \subset Y \subset V$, we have $f(X) \leq f(Y)$. A set submodular function is a set function which has the property that the value of the function increases as more elements are added to the input set, but the difference in the incremental value decreases as the size of the input set increases.

There has been a large amount of work done on the topic of set submodular functions [5, 6, 25, 35, 36, 43, 46, 48, 49, 55]. Set submodular functions have applications including social networks [35, 46], online advertising [5, 29, 45], set cover problems [6, 24] and control problems [43].

Optimizing a set submodular function involves finding an N -element subset $A \subset V$ which maximizes $f(A)$. It is extremely difficult to find an optimal solution for the set submodularity function problem. In practice, it is relatively much easier to use a greedy algorithm, where elements are chosen sequentially so as to maximize the gain in functional objective value, to approximate a good solution. In [48], it was shown that a greedy approach achieves at least an $(1 - e^{-1})$ -approximation of the optimal solution for a uniform matroid.

Two key features of set submodular functions is that, being a function on elements on a set, no element can be chosen more than once to be part of the input set, and that the order that the elements are chosen generally does not matter. Some papers have also considered the set submodular problem with the assumption that the order

of actions matter [5, 55]. It was shown in [5] that a greedy approach achieves the $(1 - e^{-1})$ -approximation of the optimal solution as well under these assumptions.

These two key features make set submodular functions an unsuitable choice to represent the problem about a retailer using online advertising that we are trying to solve. Since the retailer would possibly want to spend more time and resources advertising to one group of customers over another, we would ideally like to use a function where the same element can be chosen more than once to indicate the extra resources spent. Furthermore, the interest levels of different groups of customers may decrease at different rates. Because of this, the order of the groups of customers that the retailer chooses to advertise to may be crucial. These two features are not present in string submodular functions, which are a general form of set submodular functions. Hence, this makes using a string submodular function to represent our problem a good choice.

4.2 String submodular functions

A string submodular function very similar to a set submodular function, except that the function's argument is a string of elements rather than a set. This allows for duplication of the elements and for the order of elements in the string to matter. More formally, Let $\mathbb{A}$ be the set of all possible actions, with $|\mathbb{A}| = K$. At each stage i , we choose some action $a_i \in \mathbb{A}$. We define a string of actions $A = (a_1, a_2, \dots, a_T)$ as the actions taken up to time T . We define the set of all possible strings of actions as

$$\mathbb{A}^* = \{(a_1, \dots, a_T) \mid T = 0, 1, \dots \text{ and } a_i \in \mathbb{A} \text{ for all } i\}.$$

If $M = (a_1^m, \dots, a_{T_1}^m)$ and $N = (a_1^n, \dots, a_{T_2}^n)$ are two strings in $\mathbb{A}^*$, then we define string concatenation as

$$M \oplus N = (a_1^m, \dots, a_{T_1}^m, a_1^n, \dots, a_{T_2}^n).$$

We say M is a prefix of N if $N = M \oplus L$ for some $L \in \mathbb{A}^*$. We denote this as $M \preceq N$.

We define a string submodular function $f : \mathbb{A}^* \rightarrow \mathbb{R}$ with the following properties:

1. $f(M \oplus a) \geq f(M)$, for all $M \in \mathbb{A}^*$ and all $a \in \mathbb{A}$.
2. $f(a \oplus M) \geq f(M)$, for all $M \in \mathbb{A}^*$ and all $a \in \mathbb{A}$.
3. If $M, N \in \mathbb{A}^*$ and $M \preceq N$, then $\frac{f(N \oplus a) - f(N)}{f(M \oplus a) - f(M)} \leq 1$ for all $a \in \mathbb{A}$.

The first two properties indicate that the functional value of a string cannot be made worse by adding additional actions to either end of the string. This is known as prefix-monotonicity and suffix-monotonicity. The third property implies that any action taken has diminishing returns, where the gain in functional value from adding any action will be generally worse if the string is longer. This is analagous to (4.1), the main definition of a set submodular function.

The concept of string submodularity was introduced in [69]. Next, we define a greedy string. A greedy string $G_T = (a_1^*, a_2^*, \dots, a_T^*)$ is one such that

$$a_i^* = \arg \max_{a_i \in \mathbb{A}} f(a_1^*, \dots, a_{i-1}^*, a_i) - f(a_1^*, \dots, a_{i-1}^*),$$

for $1 \leq i \leq T$.

We generate a greedy string by choosing the action with the highest step-wise gain at each time step. It was shown in [70] that a greedy strategy achieves at least a $(1 - e^{-1})$ of the optimal string. These results are further refined in [72]. Our thesis extends this result in another direction, considering a scenario where it is not feasible to always be able to know which action the greedy action is.

4.3 Problem formulation

We model the problem of a retailer choosing which set of customers to advertise to using a string submodular function. We denote each set of customers as an action in $\mathbb{A}$, and the retailer wishes to sequentially advertise its product to various groups of customers over time to maximize its gain. The gain of the retailer can then be

denoted as $f(M)$, where M is the string of groups of customers that the retailer advertises to over time.

We can immediately see that f is monotone, as more advertising to one extra group of customers is going to lead to a greater gain overall. Furthermore, it is reasonable to assume that the diminishing return property holds as well. A group of customers being marketed to near the product launch date is likely to purchase more than the same group of customers being advertised to later.

In practice, the above assumption is not always true. Too much advertising may turn a customer off. However, we have chosen not to introduce a budget constraint as this would not add much to the results of this chapter, which are largely about choosing which set of customers to observe. Furthermore, in many practical situations, it is unlikely that the advertiser will advertise to any group of customers enough to turn them off, due to the diminishing returns property. Thus, we can see that using a string submodular function to model our problem is feasible.

As discussed above, it is generally intractable to find the optimal set to maximize a set submodular function efficiently. This holds true for string submodular functions as well. Using a greedy algorithm is a good alternative. However, to generate a greedy string, at each step we choose an action such that the step-wise gain is maximized. To do so, we have to know the step-wise gains of all actions. In our example, this could be done through market research, allowing the retailer to know how effective the advertising is going to be before proceeding. However, even the largest companies have finite resources. It is impractical to assume that the retailer is going to conduct market research on every group of customers at every time step. Hence, we make the following assumption.

Assumption 4.1. *At each time step t , before choosing each action, we observe the step-wise gains of n actions, where $0 \leq n \leq K$. We denote the set comprising these actions as H_t .*

This assumption states that the retailer will only be able to see the gains of advertising to some groups of customers. Because of the diminishing return property,

the step-wise gain of each action will decrease as time passes. However, if there is no further assumption on the rate of decay, then the step-wise gain of an unobserved action might decay greatly in a single time step, making analysis impossible. Hence, we impose a further assumption on the step-wise gains.

Assumption 4.2. For $M, N \in \mathbb{A}^*$ and $M \preceq N$,

$$\frac{f(N \oplus a_i) - f(N)}{f(M \oplus a_i) - f(M)} \geq \alpha_i(|N| - |M|),$$

where $0 < \alpha_i < 1$ are known constants.

This assumption states that each action's step-wise gain has a maximum rate of decay dependent on α_i . The smaller α_i is, the more volatile the action's step-wise gain is going to be. In practice, it is likely that most of the α_i are going to be close to 1. This assumption is a reasonable one to make, as the gain of advertising to a group of customers is unlikely change by a lot if only one time step has passed.

Assumption 4.3. $f(G'_t) - f(G'_{t-1})$ is revealed at the end of time step t , where G'_t is the string made up of actions chosen in the first t time steps.

We also assume that the step-wise gain of the action chosen of each step is revealed at the end of each time step. For instance, each time step could represent an advertising season, and a retailer could find out what the reward of each action taken by keeping track of the revenue gain each season.

Our last two assumptions are made to simplify the analysis.

Assumption 4.4. $f(\phi) = 0$, where ϕ is the empty string.

Assumption 4.5. All K step-wise gains are known initially.

Under these assumptions, we propose a pseudo-greedy strategy for choosing which groups of customers to advertise to at each time step. At each time step, the retailer also has to decide which groups of customers to conduct market research on (which actions' step-wise gains to observe). Hence, we also propose strategies on how to select which actions' step-wise gains to observe at each time step. We then analyze

each of these under the pseudo-greedy strategy, and derive a lower bound of their performance with respect to that of the optimal string. We then compare each of these strategies to one another and two baseline strategies. that of observing actions randomly and that of observing alternating actions. We will see that which of these strategies to use will heavily depend on the α_i values.

4.4 Strategies

We use a selection strategy similar to the greedy strategy. At each time step, we attempt to choose the action which generates the highest step-wise gain. However, since we cannot observe the step-wise gains of all actions, we instead choose the action which has the highest previously observed step-wise gain, leading to a pseudo-greedy string G'_t .

For $1 \leq i \leq K$, define $b_{i,1} = f(a_i)$. We denote G'_0 as the empty string. These represent the previously observed step-wise gain for each action. After observing the step-wise gain of each action in H_t , we update our observed values:

$$b_{i,t} = \begin{cases} f(G'_{t-1} \oplus a_i) - f(G'_{t-1}) & \text{for } a_i \in H_t \\ b_{i,t-1} & \text{otherwise.} \end{cases}$$

If $(t-1)^* \notin H_t$ then

$$b_{(t-1)^*,t} = f(G'_{t-1}) - f(G'_{t-2}).$$

This is because since we assume that the functional value of our pseudo-greedy string is known, we will be able to indirectly observe the step-wise gain of the chosen action in the previous time step.

We then choose the action which has the highest previously observed step-wise gain and update our string.

$$G'_t = G'_{t-1} \oplus a_{t^*},$$

where

$$t^* = \arg \max_{1 \leq i \leq K} b_{i,t}.$$

If $n \geq 1$, besides choosing which action to take at each time step, we also have to select n actions at each time step to observe their step-wise gains. We outline three such strategies.

Grouping Strategy:

We divide the K actions into groups of n actions, with the last group possibly having fewer than n actions. There will be $\lceil \frac{K}{n} \rceil$ such groups, denoted as $B_1, \dots, B_{\lceil \frac{K}{n} \rceil}$. If action a_j is taken at time t , then the actions in the group containing a_j will be observed at time $t + 1$.

Constant Strategy:

We observe the step-wise gains of the same n actions at every time step, regardless of which action is taken at each time step. We denote the set of observed actions as C .

Hybrid Strategy:

We observe the step-wise gains of the same n' actions at every time step, regardless of which action is taken at each time step, where $n > n'$. We divide the remaining actions into groups of $n - n'$ actions. We denote the groups as $D_1, \dots, D_{\lceil \frac{K-n'}{n-n'} \rceil}$. If action a_i is taken at time t , and $a_i \in D_j$ for some j , then D_j will also be observed at the next time step. Otherwise, the next group observed will be chosen at random.

4.5 Results and Discussion

We first start with the case where $n = 0$. That is, no action's step-wise gain can be observed other than the one taken in the previous time step. Define $\hat{\alpha} = \prod_{i=1}^K \alpha_i$. We use the following lemma and the Arithmetic Mean-Geometric Mean (AM-GM) inequality, which we will state below without proof. We now show the following

lemma.

Lemma 4.1. *For all l , $\sum_{i=0}^{l-1} \frac{c_{l-i}}{T} \prod_{j=1}^i (1 - \frac{c_{l+1-j}}{T}) = 1 - \prod_{i=1}^l (1 - \frac{c_i}{T})$.*

Proof. We prove this lemma using induction. We define $S_l = \sum_{i=0}^{l-1} \frac{c_{l-i}}{T} \prod_{j=1}^i (1 - \frac{c_{l+1-j}}{T})$. Observe that the lemma is true for $l = 1$. Assume that it is true for $l = m$. Then,

$$\begin{aligned} S_{m+1} &= \sum_{i=0}^m \frac{c_{m+1-i}}{T} \prod_{j=1}^i (1 - \frac{c_{m+2-j}}{T}) \\ &= \frac{c_{m+1}}{T} + (1 - \frac{c_{m+1}}{T}) S_m \\ &= \frac{c_{m+1}}{T} + (1 - \frac{c_{m+1}}{T}) (1 - \prod_{i=1}^m (1 - \frac{c_i}{T})) \\ &= 1 - \prod_{i=1}^{m+1} (1 - \frac{c_i}{T}). \end{aligned}$$

Hence the lemma is proven. $\square$

Lemma 4.2. *For any n non-negative real numbers $x_1, \dots, x_n$, we have*

$$\frac{\sum_{i=1}^n x_i}{n} \geq (\prod_{i=1}^n x_i)^{\frac{1}{n}}.$$

We can now state our theorem.

Theorem 4.1. *Let f be a string submodular function, with O_T being an optimal string of length T . Assume that at each time step, we are only able to observe the step-wise gain of the action chosen in the previous time step. Then, using the strategy outlined in Section 4.4, we have*

$$f(G'_T) \geq (1 - e^{-\hat{\alpha}}) f(O_T).$$

We define c_t as the ratio between the step-wise gain of the action chosen at time

t against that of the greedy action at time t . That is,

$$c_t = \frac{f(G'_{t-1} \oplus a_{t^*}) - f(G'_{t-1})}{\max_{1 \leq i \leq K} f(G'_{t-1} \oplus a_i) - f(G'_{t-1})}. \quad (4.2)$$

If $n = K$, the greedy action will always be chosen at each time step and hence $c_t = 1$. At all time steps, we have $c_t \leq 1$. Note that $c_1 = 1$ as it is assumed that we know the functional values of all single action strings.

We first try to bound the cumulative product of c_t . Note that each previously observed step-wise gain of an action will always be greater or equal to the current actual step-wise gain of that action. Hence,

$$\begin{aligned} b_{t^*,t} &= \max_{1 \leq i \leq K} b_{i,t} \\ &\geq \max_{1 \leq i \leq K} f(G'_{t-1} \oplus a_i) - f(G'_{t-1}). \end{aligned}$$

We then have

$$\begin{aligned} c_t &= \frac{f(G'_{t-1} \oplus a_{t^*}) - f(G'_{t-1})}{\max_{1 \leq i \leq K} f(G'_{t-1} \oplus a_i) - f(G'_{t-1})} \\ &\geq \frac{f(G'_t \oplus a_{t^*}) - f(G'_{t-1})}{b_{t^*,t}} \\ &= \frac{f(G'_t) - f(G'_{t-1})}{b_{t^*,t}} \\ &= \frac{b_{t^*,t+1}}{b_{t^*,t}}. \end{aligned}$$

Hence,

$$\begin{aligned} \prod_{t=1}^T c_t &\geq \prod_{t=1}^T \frac{b_{t^*,t+1}}{b_{t^*,t}} \\ &\geq \prod_{t=1}^T \frac{b_{t^*,T}}{b_{t^*,0}} \\ &\geq \prod_{i=1}^K \frac{b_{i,T}}{f(a_i)} \end{aligned}$$

$$\geq \hat{\alpha}^T. \quad (4.3)$$

Now, we use a result shown in [70]. Let $M, N \in A^*$, $N = (a_1, \dots, a_{|N|})$. We then have

$$\begin{aligned} & f(M \oplus N) - f(M) \\ &= \sum_{i=1}^{|N|} f(M \oplus (a_1, \dots, a_i)) - f(M \oplus (a_1, \dots, a_{i-1})). \end{aligned}$$

Hence, there exists some action a_j such that

$$\begin{aligned} & f(M \oplus (a_1, \dots, a_j)) - f(M \oplus (a_1, \dots, a_{j-1})) \\ & \geq \frac{1}{|N|} f(M \oplus N) - f(M). \end{aligned}$$

We then have

$$\begin{aligned} & f(M \oplus a_j) - f(M) \\ & \geq f(M \oplus (a_1, \dots, a_j)) - f(M \oplus (a_1, \dots, a_{j-1})) \\ & \geq \frac{1}{|N|} f(M \oplus N) - f(M). \end{aligned}$$

Now substituting $M = G'_{t-1}$ and $N = O_T$, we have

$$\begin{aligned} & f(G'_t) - f(G'_{t-1}) \\ &= f(G'_{t-1} \oplus a_{t*}) - f(G'_{t-1}) \\ & \geq c_t \cdot \frac{1}{T} (f(G'_{t-1} \oplus O_T) - f(G'_{t-1})) \end{aligned}$$

Then,

$$\begin{aligned} f(G'_T) & \geq \frac{c_T}{T} (f(O_T) + (1 - \frac{c_T}{T}) f(G'_{T-1})) \\ & \geq f(O_T) \sum_{i=0}^{T-1} \frac{c_{T-i}}{T} \prod_{j=1}^i (1 - \frac{c_{T+1-j}}{T}). \end{aligned} \quad (4.4)$$

From Lemma 4.1, substituting into (4.4) we get

$$f(G'_T) \geq (1 - \prod_{i=1}^T (1 - \frac{c_i}{T})) f(O_T).$$

Now, we have

$$\begin{aligned} \frac{\prod_{i=1}^T (1 - \frac{c_i}{T})}{(1 - \frac{\hat{\alpha}}{T})^T} &= \prod_{i=1}^T \frac{1 - \frac{c_i}{T}}{1 - \frac{\hat{\alpha}}{T}} \\ &\leq (\frac{1}{T} \sum_{i=1}^T \frac{1 - \frac{c_i}{T}}{1 - \frac{\hat{\alpha}}{T}})^T \end{aligned} \tag{4.5}$$

$$\begin{aligned} &= (\frac{T - \frac{1}{T} \sum_{i=1}^T c_i}{T - \hat{\alpha}})^T \\ &\leq (\frac{T - (\prod_{i=1}^T c_i)^{\frac{1}{T}}}{T - \hat{\alpha}})^T \end{aligned} \tag{4.6}$$

$$\begin{aligned} &\leq (\frac{T - \hat{\alpha}}{T - \hat{\alpha}})^T \\ &= 1. \end{aligned} \tag{4.7}$$

Here, inequalities (4.5) and (4.6) are due to Lemma 4.2, and (4.7) is due to (4.3).

Now, we have

$$\begin{aligned} f(G'_T) &\geq (1 - \prod_{i=1}^{m+1} (1 - \frac{c_i}{T})) f(O_T) \\ &\geq (1 - (1 - \frac{\hat{\alpha}}{T})^T) f(O_T) \\ &\geq (1 - e^{-\hat{\alpha}}) f(O_T), \end{aligned}$$

and hence the theorem is proven.

Note that this bound becomes rather poor if the number of actions is large. However, we can improve the bound further if we are able to observe the step-wise gains of several actions before making a choice at each time step. We will now consider this scenario and determine lower bounds for the functional value of the pseudo-greedy string using each of the three strategies for selecting which actions to observe. We will show that none of these three strategies are strictly superior to the other two,

and which strategy to select depends greatly on the α_i values.

Proposition 4.1. *Let f be a string submodular function, with O_T being an optimal string of length T . Assume that at each time step, we are able to observe the step-wise gain of the action taken in the previous time step, as well as that of n other actions. Then, using the strategy outlined in Section 4.4 to select which action to take, and the grouping strategy to select which actions' step-wise gains to observe, we have*

$$f(G'_T) \geq (1 - e^{-\beta})f(O_T),$$

where

$$\beta = \prod_{i=1}^{\lceil \frac{K}{n} \rceil} \min_{j|a_j \in B_i} \alpha_j.$$

Proof. Since we always observe the step-wise gains of each action in a group at the same time step, we can view each group as a single action and reduce the problem to the case where $n = 0$. In this case, we have $\lceil \frac{K}{n} \rceil$ actions, and the maximum stepwise gain of each group B_i is reduced by at most $\min_{j|a_j \in B_i} \alpha_j$ at each time step. Hence, the desired bound is obtained. $\square$

From Proposition 4.1, we see that to obtain a maximum lower bound, β should be as large as possible. Thus, we should divide the groups such that the n actions with the lowest α_i values are in one group, followed by always putting the next n actions with the lowest α_i in the next group. We now derive a lower bound for the constant strategy.

Proposition 4.2. *Let f be a string submodular function, with O_T being an optimal string of length T . Assume that at each time step, we are able to observe the step-wise gain of the action taken in the previous time step, as well as that of n other actions. Then, using the strategy outlined in Section 4.4 to select which action to take, and*

the constant strategy to select which actions' step-wise gains to observe, we have

$$f(G'_T) \geq (1 - e^{-\gamma})f(O_T),$$

where

$$\gamma = \prod_{i \notin C} \alpha_i.$$

Proof. Whenever an action in C is taken, $c_t = 1$. Hence, the lower bound for the functional value of the pseudo-greedy string is achieved when no actions in C are taken at any time step. This reduces to the problem with $n = 0$ using the remaining actions, and the desired lower bound is obtained. $\square$

From Proposition 4.2, to obtain a maximum lower bound, we wish to maximize γ . Thus, C should be chosen such that the n actions with the lowest α_i values belong in it. Finally, we now derive a lower bound for the hybrid strategy.

Proposition 4.3. *Let f be a string submodular function, with O_T being an optimal string of length T . Assume that at each time step, we are able to observe the step-wise gain of the action taken in the previous time step, as well as that of n other actions. Then, using the strategy outlined in Section 4.4 to select which action to take, and the hybrid strategy to select which actions' step-wise gains to observe, we have*

$$f(G'_T) \geq (1 - e^{-\delta})f(O_T),$$

where

$$\delta = \prod_{i=1}^{\lceil \frac{K-n'}{n-n'} \rceil} \min_{j|a_j \in D_i} \alpha_j.$$

From Proposition 4.3, it can be seen that the lower bound will change as n' changes. However, for a given n' , we should choose the n' actions that are always to be observed using the same way as in the constant strategy, that is, the n' actions

with the lowest α_i values. The remaining actions should then be divided into groups in a similar manner as the grouping strategy, such that the next $n - n'$ actions with the lowest α_i values form a group etc. The choice of n' is interesting and heavily depends on what the α_i values are.

Each strategy gives a different lower bound. Hence, the optimal strategy on choosing which actions' step-wise gains to observe is heavily dependent on the α_i values. Of the three strategies presented in our thesis, the grouping strategy works best when all of the α_i values are not too far apart from one another. Using the advertising analogy, this may occur if the advertiser has little information about the customer base, and thus expects that each group of consumers may react in approximately the same way towards advertising.

The constant strategy is best used when there are n actions with α_i values far less than the other actions. In this case, it would make sense to observe these actions much more frequently than the other actions as taking one of these actions without it being observed recently may be very costly. This may occur when the advertiser can identify groups of long-term, loyal customers. These customers' reactions to advertising will likely change very little over time, compared to newer customers who may only be following a fad. In these cases, the advertiser would likely want to pay more attention to the sales generated by the newer customers.

In many other scenarios, where there is a more even mix of α_i values, the hybrid strategy may be best. The best n' still has to be determined, but in general it is advisable to choose n' such that the α_i values of the remaining actions are roughly even.

In general, even if the α_i values are high, observing only a few actions at each time step leads to a very poor bound if K is large. However, the bound improves greatly if the number of actions observed is a proportion of K . For instance, if each $\alpha_i = 0.9$ and we are able to observe $K/2$ actions at each time step, then we can achieve a bound of at least $(1 - e^{0.9})$ of the optimal functional value, which is not too far off from the bound of $(1 - e^{-1})$ that the greedy strategy attains. This is a small price to pay for being able to cut by half the cost of observing how much actions add to the

functional value.

Note, however, that our results only provide a theoretical lower bound on the performance of the three strategies. The bounds can be attained if at each time step, an adversary chooses the step-wise gain of each action following the submodularity assumptions before we observe the actions' step-wise gains. In practice, the grouping strategy will perform significantly better than the other two methods. This is because in many scenarios, the change in step-wise gains between each time step is stochastic, and the grouping strategy often observes more actions than the other two strategies over a similar time horizon. This generally leads to a better average performance, but an inferior worse-case performance in some scenarios. We can see the effect of this in the next section.

4.6 Numerical results

In this section, we compare the performances of the grouping, constant and hybrid observation strategies against that of two other observation strategies. The first is the random strategy, where besides the chosen action, n other actions are chosen at random to observe. The second is the round-robin strategy, where similar to the grouping strategy, actions are divided into groups of n actions. However, here the observed group rotates every time step such that each group is observed with approximately equal frequency. This is in contrast to the grouping strategy, where the group observed is the same one as the action taken.

We performed a simulation using 30 actions, each with an initial known reward uniformly distributed between 0.9 and 1. We assumed that at each time step, 3 actions would be observable, including the action taken in the previous time step. We also assume that at the end of each time step, we randomly choose a set of 15 actions H_t and each of the actions in H_t will have its step-wise gains decay by the maximum amount, with the others not changing. That is, $\forall a_i \in H_t$,

Figure 4-1: Ratio against optimal string for each strategy

$$f(M_t \oplus a_i) - f(M_t) = \alpha_i \{(M_{t-1} \oplus a_i) - f(M_{t-1})\},$$

where M_t is the string at time step t . The other step-wise gains will stay the same. We choose the α -values for each action uniformly between 0.8 and 1. We run each trial 100 times, with each trial being 100 time steps, and then plot the lowest ratio of the reward for the strings produced by each strategy against the reward produced by the optimal string at each time step over all trials. We choose to use the lowest value of the ratio over all trials as we wish to look at the worst-case performance for each of the strategies. The result is shown in Figure 4-1.

Figure 4-1 shows us that there is a large drop in the ratio between the first and second time steps. This is due to not being able to observe all the actions' step-wise gains between each time step. The chance of picking the optimal action at the second time step is thus the lowest. As more and more actions are observed, the performance of each strategy becomes better. The group strategy outperforms all the other strategies. This is because the group strategy is able to observe the step-wise gains of a wide variety of actions, yet focus more on the actions that are more likely

Figure 4-2: Ratio against optimal string for each strategy

to be chosen at each time step. In contrast, the constant strategy performs poorly as the actions that it chooses at each time step are less likely to be the ones that it observes.

In Figure 4-2, we perform the same 100 trials. However, here we make a small tweak to the α -values, setting the α -values of the first two actions to be uniformly chosen between 0 and 0.1, as opposed to uniformly chosen between 0.8 and 1. The constant strategy still performs the most poorly in the long run under these assumptions. However, we see that at time step 2, it performs significantly better than the other strategies. This is because the strategy is designed to maximize the ratio when the actions with the greatest potential changes are likely to be chosen. Any time this is true, the constant strategy will outperform the other strategies. However, under our assumptions, this is likely to be true only in the first few time steps. Note that the hybrid strategy is a good mix between the constant and group strategy. Its minimum ratios in the early time steps are better than that of the group strategy, and its overall performance in the long run is significantly better than the constant strategy, being only slightly worse than the random strategy.

4.7 Summary

In this chapter, we have provided a strategy on how to choose actions to maximize the value of a string submodular function if the step-wise gain of some or all actions are not known. We have also derived some bounds on the value of the string generated by this strategy, as compared to that of the optimal string. We have also provided three different strategies to select which actions' step-wise gains to observe, for cases at least one other action's step-wise gain can be observed in each time step. Furthermore, we have provided bounds using each of the strategies described and presented scenarios where each one is optimal.

Chapter 5

Learning as a Multi-armed Bandit Problem

In the previous chapter, we saw how learning in a social network can be modeled as trying to optimize a string-submodular function. In this chapter, we present another way to model learning in a social network. In many practical applications, interacting with users in a social network will allow one to indirectly interact with the neighbors of that user. For example, if a user posts on a social network, his friends or followers will be able to directly comment on the post. In a similar manner, by interacting with a user in a social network to learn more about that user, we will be able to learn about his neighbors as well. Applications of this phenomenon include customer profiling and reputation systems, where a user that is trusted by other trustworthy users is likely to have a good reputation.

Again, we will use the example of a retailer trying to decide how to advertise its products to the users of a social network. The key concept is that learning the response of a user will be able to give us some information about the response of that user's neighbors as well. We will see that we can model this as a variant of the classical multi-armed bandit problem in a social network setting. We will first discuss the multi-armed bandit problem and some algorithms to solve it.

5.1 Multi-armed bandits

The multi-armed bandit problem is one where a gambler walks into a casino and is faced with N different slot machines, each with a different payout. The gambler does not know the mean payout of each machine, and yet wishes to maximize his expected winnings. By trying each machine several times, the gambler will gradually form an idea of the expected payout of each machine. A large part of solving the multi-armed bandit problem is then balancing between selecting the machine that the gambler expects to have the largest payout, or exploitation, and selecting other machines in order to learn more about their payout, or exploration.

More formally, we sequentially choose between N actions at each time step T . Each time step, we get a random reward $X_{i,T}$ for choosing action i , where $X_{i,T}$ is drawn from some unknown probability distribution P_i . Our aim is then to maximize our cumulative expected reward.

In this thesis, we concern ourselves with the stochastic multi-armed bandit problem, where each action's reward is generated randomly at each time step. Another common form of the multi-armed bandit problem is that of adversarial bandits, where there is an adversary which chooses the rewards of each action at each time step.

We can see that a multi-armed bandit framework works well in modeling our problem of online advertising. Ideally, a retailer would want to generate discussion about its online advertisements. Hence, the reward for each user can be measured by how much the user is engaged with the advertisement, like whether or not it is shared with the user's neighbors. The aim of the retailer is to maximize its reward by marketing its product to users which are more interested.

The performance of a multi-armed bandit algorithm is typically measured by its regret, which is the expected difference between the reward generated by the algorithm and that of the optimal strategy of always choosing the action which gives the highest expected reward. For the stochastic problem, it was shown in [37] that the expected regret will grow at least logarithmically with the number of time steps.

There has been a large amount of work done on the topic of multi-armed bandit

problems [4, 8, 9, 13, 16, 33, 37, 68]. In [37], an algorithm was introduced which decided which action to take based on upper confidence bounds (UCBs). This algorithm was shown to asymptotically achieve the logarithmic lower bound derived in the same paper. Since then, much work has been done on extending and developing the UCB algorithm. In particular, in [8], the UCB1 algorithm was introduced, which was shown to achieve the logarithmic lower bound uniformly over time, rather than asymptotically. There have also been many variants of the traditional multi-armed bandit problem being studied. Here are some of them that are more closely related to the problem that we are trying to model.

In the classical multi-armed bandit problem, the rewards of each action at each time step are generated independent of one another. In [50], the problem of dependent bandits is studied. Here, the number of possible actions is very large, and to mitigate this, each action is represented as part of a metric space, with actions close to each other having a similar reward probability distribution. This allows choosing an action to gain us information about the reward distributions of nearby actions, and is one possible way to model the multi-armed bandit problem in a social network setting. However, we have chosen to take a different tack. In our model, the rewards of each action is independently generated. Nevertheless, taking one action will allow us to gain information about the reward distributions of its neighbors.

Another similar multi-armed bandit model is that of linear bandits [1, 2, 22, 26]. Here, at each time step, each action x is a point in a finite dimensional vector space. The mean reward of each action is $Y \cdot x$, where Y is an unknown vector. Again, we see that information about nearby actions' reward distributions can be derived by taking an action. This extends into the problem of contextual bandits [13, 17, 38, 52, 53, 64, 65], where at each time step, a context is chosen from a context set $\mathcal{S}$, and the reward distributions for each action are dependent on the context chosen. This is yet another take on modeling learning in a social network. One application would be in news article recommendation, where a website wishes to recommend articles to its viewers. Each viewer's response to each article would be different, but each viewer has some side information associated with him. This could be address, age, or

another demographic information. Using these features, contexts are created, where it is assumed that viewers with similar demographic information would like similar articles.

Lastly, the models used in [18, 19, 44] are fairly similar to our own. Here, actions are connected to one another through some known graph topology, and taking an action allows one to observe the reward of all its neighbors for that time step. In [19], the UCB-1 algorithm is generalized to the UCB-N algorithm which takes the side observations from the social network structure into account, which is a significant improvement over the UCB1 algorithm in dense networks. The work in [44] performs similar analysis, except taking the adversarial multi-armed bandit problem into account, rather than the stochastic one.

One key feature about the above model is that when an action is taken, all the rewards of all its neighbors for that time step are observed directly. In our work, we propose two different extensions of this model.

In the first, the rewards of all the neighbors of the chosen action are not observed directly. Instead, a biased version of the reward is observed, where the bias is unknown to the decision maker. Besides trying to learn which action produces the best expected reward, the decision maker is now also tasked with learning the bias between each neighboring pair of actions.

Our second model gives the decision maker a choice to pay an additional cost to observe a larger neighborhood of the chosen action. In both of these models, we propose algorithms to allow the decision maker to choose which action to take at each time step. We then compare our algorithm’s performance to the UCB1 and UCB-N algorithms.

5.2 Learning from biased neighbor observations

In this section we formally define the multi-armed bandit problem in the context of our first model. We let $\mathcal{N} = \{1, \dots, N\}$ be defined as the set of agents in a social network. At each time step T , rewards $X_{i,T}$ are independently generated from

unknown distributions $\mathcal{P}_i$ for $1 \leq i \leq N$. We assume that each $\mathcal{P}_i$ is normally distributed with mean μ_i and variance σ^2 . We denote $\mu = \{\mu_i\}_{i=1}^N$. Now, we define the optimal agent as

$$i^* = \arg \max \mu_i,$$

the agent which provides the decision maker highest expected reward. At each time step T , the decision maker chooses an agent, which we denote as I_T . The decision maker then gains the agent's associated reward $X_{I_T, T}$. We now can define the regret of the decision maker at time T , which is

$$\sum_{t=1}^T X_{i^*, t} - \sum_{t=1}^T X_{I_t, t}.$$

The aim of the decision maker is to minimize the expected regret of the policy it uses.

We have chosen to use equal variance Gaussian distributions for each agent's reward because of ease of analysis. It should be noted that other distributions can be used, which will lead to different equations for the mean and bias estimates later in this chapter. However, the core idea of the algorithms will still work for different distributions, as long as these equations can be derived. If they cannot, it is possible that numerical methods like gradient descent will still help us to find these estimates.

In the standard multi-armed bandit problem, the decision maker is assumed to have access to the full history of rewards of agents that it had chosen in the previous time steps. This will allow the decision maker to form an estimate of the mean rewards of each agent and to determine which agent to select next. However, the problem that we are interested in modeling is in a social networks setting, people are connected to one another, and learning information about one person is likely to give us information about his nearby neighbors in the social network as well.

This translates naturally into defining an undirected graph G over the set of agents, with each agent representing a vertex in G . We connect two agents i, j via

an edge if they are connected in the social network, for instance if they are friends on Facebook. We now define $N_1(i)$, the immediate neighborhood of an agent i , as consisting of agent i and its immediate neighbors in G . We now apply the following assumptions to extend the standard multi-armed bandit problem.

Assumption 5.1. *When choosing agent I_T at time T , the decision maker also observes $Y_{j,I_T,T} = X_{j,T} + B_{I_T,j}$ for all $j \in N_1(I_T)$. For all $i \notin N_1(I_T)$ or all $k \neq I_T$, we also have $Y_{i,k,T} = 0$.*

This assumption allows the decision maker to gain information about an action's reward without taking that action. We note that this assumption is quite a strict one. In practice, the decision maker may not be able to observe the biased rewards of all neighbors of the chosen agent. However, we have chosen to use this assumption to simplify our analysis. The algorithm will still function if only a random subset of the neighbors is observed. The equations used to estimate the means and biases will be different though.

Returning to our example, the online retailer may invite certain users in a social network to share information about their product for a discount with their friends, and attempt to gauge their reaction to their products. The more enthusiastic a user is, the greater the reward of the retailer. After the user shares the news with his friends, the retailer is then also able to gauge the opinion of the user's friends towards the products. However, not all connections in a social network are equal. A close friend of the user is more likely to engage with his posts even if his interest in the retailer's products is relatively low. Hence, the retailer will not be able to accurately gauge the opinion of the user's friends. We model this by adding a bias term to the observed reward by the decision maker. We also make the following two assumptions on the bias term.

Assumption 5.2. $B_{i,i} = 0$ for all $1 \leq i \leq N$.

Assumption 5.3. $B_{i,j} = B_{j,i}$ for all $1 \leq i, j \leq N$.

The first assumption states that an agent's bias term with itself is zero. By targeting the chosen user at each time step, the retailer will be able to directly gauge

the user’s response, and hence directly observe the exact reward for that agent. Hence, the bias is zero for the chosen agent at each time step.

The second assumption states that the bias term is symmetric for each pair of users. This bias term can be either positive or negative. This is a reasonable assumption to make, as it is not unrealistic that a pair of friends that are close to each other are likely to engage with each other’s posts fairly equally, whereas a pair of friends that are more distant will not show the same level of engagement either way. We can represent a high level of engagement with a positive number and a low level with a negative one. Lastly, we denote the set of biases as $B = \{B_{i,j}\}$ for $1 \leq i, j \leq N$.

An agent’s reward can be observed directly if that agent is chosen by the decision maker. It can also be observed indirectly if the decision maker chooses one of its neighbors. We denote the number of times up to time step T that the agent i ’s reward has been directly observed (and hence how many times agent i has been chosen by the decision maker) as $O_{i,T}$.

5.3 The UCB-R algorithm

In this section, we will propose an algorithm to tackle the multi-armed bandit problem generated by our model. Because we wish to model learning in a social network and have it robust to biased observations of the neighbors, we called our algorithm the UCB-R algorithm. We first start by briefly describing the UCB1 algorithm, which minimizes the upper bound of the regret of the standard multi-armed bandit problem.

The UCB1 algorithm balances exploration and exploitation by creating an index known as the Upper Confidence Bound (UCB) for each action, and then choosing the action with the highest UCB. It does so by adding a bias term to the sample mean of the reward of each action. For action i at time step t , we have

$$UCB_{i,T} = \bar{X}_{i,T} + \sqrt{\frac{2 \log T}{O_{i,T}}},$$

where $\bar{X}_{i,T}$ is the sample mean of the observed rewards of action i up to time T .

The UCB1 algorithm strikes a good balance between exploitation and exploration, as in general actions with a higher expected mean reward will be chosen due to the sample mean in the UCB (exploitation), but actions which have not been chosen many times previously may also get chosen periodically because of the bias term in the UCB (exploration).

In [8], it was shown that the UCB1 algorithm achieves an upper bound of regret growth on the order of $\log T$. To generalize the UCB1 algorithm to our model, we have to account for the additional information provided by choosing an agent's neighbors. We will need an estimate for the mean of the reward at time step T , which we denote as $\hat{\mu}_{i,T}$ for agent i . Furthermore, each agent will be observed whenever one of its neighbors or itself is chosen by the decision maker. Hence, we can define an index

$$UCB'_{i,T} = \hat{\mu}_{i,T} + \sqrt{\frac{2 \log T}{\sum_{j \in N_1(i)} O_{j,T}}}.$$

To make a decision on which agent to choose at time T , the decision maker has to have some sort of estimate $\mu_{i,T}$ for each agent i . In the standard multi-armed bandit problem, this is easy. For each agent i , we can use the sample mean

$$\frac{\sum_{t=1}^{T-1} Y_{i,i,t}}{O_{i,T-1}}.$$

However, in our model, we have extra observations from choosing agent i 's neighbors. Furthermore, we have to contend with the biases as well. Hence, at each time step t , to determine a good estimate for the means of the agent rewards, the decision maker will need estimates for biases, which we denote as $\hat{B}_{i,j,T}$ for $1 \leq i, j \leq N$ and $i \neq j$ respectively. We denote the set of rewards observed up to time T as

$$Y_T = \{Y_{i,j,t}\},$$

for $1 \leq i, j \leq N$ and $1 \leq t \leq T$.

Note that $Y_{j,I_T,T} - B_{I_T,j} - \mu_j$ is a Gaussian random variable with mean 0 and variance σ^2 for all $j \in N_1(I_T)$. Hence, the likelihood function relating the observed set of rewards and the parameters we want to estimate is

$$\begin{aligned} L(\mu, B; Y_T) &= p(Y_T | \mu, B) \\ &= \prod_{t=1}^T \left\{ \prod_{j \in N_1(I_t)} f(Y_{j,I_t,t} - B_{j,I_t} - \mu_j) \right\}, \end{aligned} \quad (5.1)$$

where f is the probability distribution function of a $\mathcal{N}(0, \sigma^2)$ random variable.

The decision maker would like to find the set of μ and B values that will maximize the likelihood function. That is,

$$\arg \max_{\mu, B} \prod_{t=1}^T \left\{ \prod_{j \in N_1(I_t)} f(Y_{j,I_t,t} - B_{j,I_t} - \mu_j) \right\}.$$

To do so, we could also try to maximize the log-likelihood function

$$\log(L(\mu, B; Y_T)) = \sum_{t=1}^T \left\{ \sum_{j \in N_1(I_t)} \log \{f(Y_{j,I_t,t} - B_{j,I_t} - \mu_j)\} \right\}.$$

This turns out to be quite challenging to find, so we will attempt to find an estimate numerically. One important property about the likelihood function is that if either of μ or B is known, then finding the value of the other parameter which maximizes the likelihood function is relatively easy. We shall show that in the two propositions below.

Proposition 5.1. *Let $L(\mu, B; Y_T)$ be defined as in (5.1). Assume μ is fixed. Then, the estimates of $B_{i,j}$ which maximize $L(\mu, B; Y_T)$ are given by*

$$\hat{B}_{ij,T} = \frac{\sum_{t=1}^T (Y_{i,j,t} + Y_{j,i,t}) - (O_{i,T} \cdot \mu_j + O_{j,T} \cdot \mu_i)}{O_{i,T} + O_{j,T}},$$

for all $i \in N_1(j), i \neq j$.

Proof. Note that since μ is fixed, the terms in the likelihood function can be partitioned such that each term

$$\log f(Y_{i,j,t} - B_{i,j} - \mu_i)$$

in the function can be assigned to a pair of agents i and j . Hence, we can maximize each set of terms assigned to different pairs of agents separately. For the rest of the proof, we will look at how to maximize the terms assigned to the pair of agents i and j , that is

$$\sum_{t=1}^T \{ \mathbb{1}_{I_t=i} \cdot \log f(Y_{j,i,t} - B_{i,j} - \mu_j) + \mathbb{1}_{I_t=j} \cdot \log f(Y_{i,j,t} - B_{i,j} - \mu_i) \},$$

where $\mathbb{1}$ is the indicator function.

Note that each time $I_t = j$, we have

$$(Y_{i,j,t} - \mu_i) \sim \mathcal{N}(B_{i,j}, \sigma^2).$$

We denote $(Y_{i,j,t} - \mu_i)$ as $Z_{i,j,t}$. Similarly, each time $I_t = i$, we have

$$\begin{aligned} Z_{j,i,t} &= (Y_{j,i,t} - \mu_j) \\ &\sim \mathcal{N}(B_{j,i}, \sigma^2) \\ &= \mathcal{N}(B_{i,j}, \sigma^2). \end{aligned}$$

As we assume that Y_T and μ are both known, we have $O_{i,T}$ observations of $Z_{j,i,t}$ and $O_{j,T}$ observations of $Z_{i,j,t}$, each being an independent realization of $\mathcal{N}(B_{i,j}, \sigma^2)$. Hence, to find an estimate of $B_{i,j}$ that maximizes the likelihood ratio, we can take the sample mean of these $(O_{i,T} + O_{j,T})$ realizations, which is

$$\frac{\sum_{t=1}^T (Y_{i,j,t} + Y_{j,i,t}) - (O_{i,T} \cdot \mu_j + O_{j,T} \cdot \mu_i)}{O_{i,T} + O_{j,T}}.$$

Applying this to all connected pairs i and j completes our proof. $\square$

Now that we have seen how we can find a good estimate of B if μ is known, we shall show the converse. In the next proposition, we will see how to find the values of μ which maximize the likelihood function if B is known.

Proposition 5.2. *Let $L(\mu, B; Y_T)$ be defined as in (5.1). Assume B is fixed. Then, the values of μ_i which maximize $L(\mu, B; Y_T)$ are given by*

$$\mu_{i,T} = \frac{\left\{ \sum_{t=1}^T \sum_{j \in N_1(i)} Y_{i,j,t} \right\} - \sum_{j \in N_1(i)} O_{j,T} \cdot B_{i,j}}{\sum_{j \in N_1(i)} O_{j,T}},$$

for all $1 \leq i \leq N$.

Proof. Similar to the proof of Proposition 5.1, since B is fixed, the terms in the likelihood function can be partitioned such that each term

$$\log f(Y_{i,j,t} - B_{i,j} - \mu_i)$$

in the function can be assigned to a pair of agent i , and we can maximize each set of terms assigned to each agent separately. For the rest of the proof, we will look at how to maximize the terms assigned to the pair of agent i , that is

$$\sum_{t=1}^T \left\{ \sum_{j \in N_1(i)} \mathbb{1}_{I_t=j} \cdot \log f(Y_{i,j,t} - B_{i,j} - \mu_i) \right\}.$$

Each time $j \in N_1(I_t)$, we have

$$(Y_{i,I_t,t} - B_{i,I_t}) \sim \mathcal{N}(\mu_i, \sigma^2).$$

We denote $(Y_{i,I_t,t} - \mu_i)$ as $Z'_{i,I_t,t}$. As we assume that Y_T and B are both known, we have $\sum_{j \in N_1(i)} O_{j,T}$ observations of $Z_{i,I_t,t}$, each being an independent realization of

$\mathcal{N}(\mu_i, \sigma^2)$. Hence, to find an estimate of μ_i that maximizes the likelihood ratio, we can take the sample mean of these $\sum_{j \in N_1(i)} O_{j,T}$ realizations, which is

$$\frac{\left\{ \sum_{t=1}^T \sum_{j \in N_1(i)} Y_{i,j,t} \right\} - \sum_{j \in N_1(i)} O_{j,T} \cdot B_{i,j}}{\sum_{j \in N_1(i)} O_{j,T}}.$$

Applying this to all agents i completes our proof. $\square$

Propositions 5.1 and 5.2 provide us with a way of getting a good estimate of either μ and B provided that the other is known. Provided we have an initial estimate of μ and B , we can continually refine these estimates using the above propositions. At time step T , We denote our initial estimates of $\hat{\mu}_{i,T}$ and $\hat{B}_{ij,T}$ for all $1 \leq i, j \leq N$ as $\hat{\mu}_{i,T}^{(1)}$ and $\hat{B}_{i,j,T}^{(1)}$ respectively. Using our initial estimates of $\hat{\mu}_{i,T}$, we use Proposition 5.1 to form a second estimate of the $\hat{B}_{i,j,T}$ values. We then use those values and Proposition 5.2 to form our next estimate of the $\hat{\mu}_{i,T}$ values. This is repeated until the values converge.

More formally, we have

$$\hat{B}_{i,j,T}^{(k)} = \frac{\sum_{t=1}^T (Y_{i,j,t} + Y_{j,i,t}) - (O_{i,T} \cdot \hat{\mu}_{j,T}^{(k-1)} + O_{j,T} \cdot \hat{\mu}_{i,T}^{(k-1)})}{O_{i,T} + O_{j,T}}, \quad (5.2)$$

for $1 \leq i \leq N, i \neq j$. We also have

$$\hat{\mu}_{i,T}^{(k)} = \frac{\left\{ \sum_{t=1}^T \sum_{j \in N_1(i)} Y_{i,j,t} \right\} - \sum_{j \in N_1(i)} O_{j,T} \cdot \hat{B}_{i,j,T}^{(k)}}{\sum_{j \in N_1(i)} O_{j,T}}, \quad (5.3)$$

for $1 \leq i \leq N$.

By alternating between (5.2) and (5.3), we can find a good estimate of B and μ

when the values converge. That is,

$$\sum_{1 \leq i \leq N} \sum_{j \in N_1(i)} (\hat{B}_{i,j,T}^{(k)} - \hat{B}_{i,j,T}^{(k-1)})^2 < \epsilon_B$$

and

$$\sum_{1 \leq i \leq N} (\hat{\mu}_{i,T}^{(k)} - \hat{\mu}_{i,T}^{(k-1)})^2 < \epsilon_\mu,$$

for some pre-chosen ϵ_B and ϵ_μ .

We will now show that alternating between (5.2) and (5.3) guarantees convergence.

Proposition 5.3. *Let $\hat{B}_{i,j,T}^{(k)}$ and $\hat{\mu}_{i,T}^{(k)}$ be defined as in (5.2) and (5.3) respectively. Then, both $\hat{B}_{i,j,T}^{(k)}$ and $\hat{\mu}_{i,T}^{(k)}$ converge as $k \rightarrow \infty$.*

Proof. Define $C_k = \max_{1 \leq i \leq N, j \in N_1(i)} |\hat{B}_{i,j,T}^{(k)} - \hat{B}_{i,j,T}^{(k-1)}|$. Similarly, define $D_k = \max_{1 \leq i \leq N} |\hat{\mu}_{i,T}^{(k)} - \hat{\mu}_{i,T}^{(k-1)}|$. Then, we have

$$\begin{aligned} D_{k+1} &= \max_{1 \leq i \leq N} |\hat{\mu}_{i,T}^{(k+1)} - \hat{\mu}_{i,T}^{(k)}| \\ &= \left| \frac{\sum_{j \in N_1(i)} O_{j,T} \cdot \hat{B}_{i,j,T}^{(k-1)} - \sum_{j \in N_1(i)} O_{j,T} \cdot \hat{B}_{i,j,T}^{(k)}}{\sum_{j \in N_1(i)} O_{j,T}} \right| \\ &= \left| \frac{\sum_{j \in N_1(i)} O_{j,T} \cdot (\hat{B}_{i,j,T}^{(k-1)} - \hat{B}_{i,j,T}^{(k)})}{\sum_{j \in N_1(i)} O_{j,T}} \right| \\ &\leq \left| \frac{\sum_{j \in N_1(i)} O_{j,T} - O_{i,T} \cdot C_k}{\sum_{j \in N_1(i)} O_{j,T}} \right| \end{aligned}$$

The final inequality is because $B_{i,i} = 0$ for all i .

In a similar manner, we can also show that $C_k \leq D_k$. Hence, we have

$$D_{k+1} \leq \left| \frac{\sum_{j \in N_1(i)} O_{j,T} - O_{i,T} \cdot D_k}{\sum_{j \in N_1(i)} O_{j,T}} \right|,$$

and hence $\hat{\mu}_{i,T}^{(k)}$ converges. This also implies that $\hat{B}_{i,j,T}^{(k)}$ converges and the proof is complete. $\square$

Using the values of $\hat{\mu}_{i,T}^{(k)}$ and $\hat{B}_{i,j,T}^{(k)}$, we can form estimates for $\hat{\mu}_{i,T}^{(k+1)}$ and $\hat{B}_{i,j,T}^{(k+1)}$. However, we have yet to define what our initial estimates $\hat{\mu}_{i,T}^{(1)}$ and $\hat{B}_{i,j,T}^{(1)}$ should be. For the first N time steps, the decision maker will gather information by selecting each of the N agents in turn. Then, we will have some information to get an initial estimate. Since each agent will be directly observed once, the initial reward mean estimate for each agent can be equal to the reward when that agent is selected. That is,

$$\hat{\mu}_{i,N+1}^{(1)} = \sum_{t=1}^N Y_{i,i,t},$$

for all $1 \leq i \leq N$.

Because we have little information about the biases, we can initialize $\hat{B}_{i,j,N+1}^{(1)} = 0$ for all $1 \leq i, j \leq N$. After the first N time steps, since the changes to the estimates are likely to be relatively small between time steps, we will use the estimates in the previous time steps as the initial estimates. That is, if $T > N + 1$, we have

$$\begin{aligned} \hat{\mu}_{i,T}^{(1)} &= \hat{\mu}_{i,T-1} \\ \hat{B}_{i,j,T}^{(1)} &= \hat{B}_{i,j,T-1}. \end{aligned}$$

Using the converged values of $\hat{\mu}_{i,T}^{(k)}$, the decision maker will select the agent

$$\arg \max \hat{\mu}_{i,T} + \sqrt{\frac{2 \log T}{\sum_{j \in N_1(i)} O_{j,T}}}$$

at time step T .

Presented on the next page is a summary of our algorithm.

```

for  $1 \leq T \leq N$  do
    Choose agent  $T$ 
    Receive observations  $\{Y_{i,T}\}_{i \in N_1(T)}$ 
end

 $\hat{\mu}_{i,N+1}^{(1)} \leftarrow \sum_{t=1}^N Y_{i,i,t}$ 
 $\hat{B}_{i,j,N+1}^{(1)} \leftarrow 0$ 
 $O_{i,N} \leftarrow 1$ 

for  $T > N$  do
     $k \leftarrow 1$ 
    while  $\sum_{1 \leq i \leq N} (\hat{\mu}_{i,T}^{(k)} - \hat{\mu}_{i,T}^{(k-1)})^2 > \epsilon_\mu, \sum_{1 \leq i \leq N} \sum_{j \in N_1(i)} (\hat{B}_{i,j,T}^{(k)} - \hat{B}_{i,j,T}^{(k-1)})^2 > \epsilon_B$  do
        
$$\hat{B}_{i,j,T}^{(k)} \leftarrow \frac{\sum_{t=1}^T (Y_{i,j,t} + Y_{j,i,t}) - (O_{i,T} \cdot \hat{\mu}_{j,T}^{(k-1)} + O_{j,T} \cdot \hat{\mu}_{i,T}^{(k-1)})}{O_{i,T} + O_{j,T}}$$


$$\hat{\mu}_{i,T}^{(k)} \leftarrow \frac{\left\{ \sum_{t=1}^T \sum_{j \in N_1(i)} Y_{i,j,t} \right\} - \sum_{j \in N_1(i)} O_{j,T} \cdot \hat{B}_{i,j,T}^{(k)}}{\sum_{j \in N_1(i)} O_{j,T}}$$

         $k \leftarrow k + 1$ 
    end

     $\hat{\mu}_{i,T} \leftarrow \hat{\mu}_{i,T}^{(k-1)}$ 
     $\hat{B}_{i,j,T}^{(1)} \leftarrow \hat{B}_{i,j,T}^{(k-1)}$ 
    
$$I_T \leftarrow \arg \max \hat{\mu}_{i,T} + \sqrt{\frac{2 \log T}{\sum_{j \in N_1(i)} O_{j,T}}}$$

    Choose agent  $I_T$ 
    Receive observations  $\{Y_{i,I_T,T}\}_{i \in N_1(M)}$ 
     $O_{I_T,T+1} < -O_{I_T,T} + 1$ 
     $\hat{\mu}_{i,T+1}^{(1)} \leftarrow \hat{\mu}_{i,T}$ 
     $\hat{B}_{i,j,N+1}^{(1)} \leftarrow \hat{B}_{i,j,T}$ 
end

```

Algorithm 1: The UCB-R algorithm

5.4 Learning from larger neighborhoods

In this section, we explore a variant of the multi-armed bandit problem in a social network context. We have seen that as an advertiser, it is possible to gauge the opinion of a targeted user in a social network, as well as the opinions of all of his neighbors in the network. However, a user in a social network is part of a much greater whole. Besides the user's immediate neighbors, there are larger neighborhoods of the user to be considered. The retailer may be able to estimate the reactions of people outside the targeted user's immediate neighborhood, as neighbors of the targeted user may share the advertisement with their neighbors, who may in turn share the advertisement to more people. Our aim is then to model learning in a social network under these conditions.

As in the previous model, we let $\mathcal{N} = \{1, \dots, N\}$ be defined as the set of agents in a social network. At each time step T , rewards $X_{i,T}$ are independently generated from unknown Gaussian distributions $\mathcal{P}_i$ for $1 \leq i \leq N$ with mean μ_i and variance σ^2 , and $\mu = \{\mu_i\}_{i=1}^N$.

The optimal agent i^* is still the agent which provides the decision maker highest expected reward, and the decision maker chooses an agent I_T at each time step T , receiving reward $X_{I_T,T}$. However, in this model, we allow the decision maker to explore a greater neighborhood of the chosen agent. We now introduce a new definition, the k -neighborhood of an agent.

A k -neighborhood of an agent i is the set of agents

$$N_k(i) \{j | \text{dist}(i, j) \leq k\},$$

where $\text{dist}(i, j)$ indicates the hop-distance in the network between agents i and j . In particular, 0-neighborhood of agent i is just i itself, and the 1-neighborhood of i is itself with its immediate neighbors in the network.

In our model, at each time step T , other than choosing an agent I_T whose reward to take, the decision maker has to also decide on a depth $d_T > 0$. Then, the decision maker receives reward $X_{I_T,T}$. In addition, the decision maker also gets to observe

the rewards of all the agents in the d_T -neighborhood of I_T . That is, we make the following assumption.

Assumption 5.4. *When choosing agent I_T at time T , the decision maker also observes $Y_{j,T} = X_{j,T}$ for all $j \in N_{d_T}(I_T)$, and $Y_{j,T} = 0$ otherwise.*

This is clearly very beneficial, as it allows the decision maker to explore the sub-optimal arms without actually choosing them. However, it comes with a cost. To explore a greater neighborhood of the targeted user, some resources have to be expended. Returning to our example about online advertising, there is some chance that the retailer’s advertisement will spread to large neighborhoods of the targeted user, but the larger the neighborhood, the less likely it is going to be. The retailer can attempt to boost this chance by paying a cost. For instance, the retailer can offer an incentive to the targeted user to promote the advertisement to his friends, increasing the chance that the advertisement will be able to be spread deeper within the social network, and allowing the retailer to gauge the reactions of more users. To simplify our analysis, we assume that this cost subtracts from the reward. In our example, for instance, the reward could be the revenue gain from advertising, while the cost paid could be the marketing fee to conduct more in-depth marketing research.

In our model, we model this cost as a fixed known cost dependent on the depth chosen by the decision maker, but not dependent on which agent was chosen at each time step. Hence, at each time step, the decision maker will instead receive the reward

$$X'_{I_T,T} = X_{I_T,T} - C_{d_T},$$

where C_{d_T} is the cost paid by the retailer.

Instead of having the cost paid tied to the distance from the chosen agent, we can define a cost function between any pair of agents, and have the decision maker specifically select which extra agents to explore. The reason we have chosen to use the distance instead is to simplify the computation. It should be noted that if the cost function is defined differently, our proposed algorithm will still work with a slight change to the equations used.

It can be seen that compared to our previous model, the rewards observed by the decision maker are exact and not biased. This can be justified because it should be possible to garner more accurate reactions of the targeted user's neighbors if more resources are expended by the retailer, which is taken into account by the cost that the retailer pays to explore larger neighborhoods of the user. We will now propose a variant of the UCB1 algorithm which allows the decision maker to determine which agent to choose at each time step, as well as how large a neighborhood the decision maker should explore.

5.5 The UCB-V algorithm

In this section, we introduce our algorithm, the UCB-V algorithm. In our algorithm, the decision maker determines which agent to choose at each time step based on the UCB1 algorithm, modified slightly to accommodate the social network setting. Then, it assigns a value to each agent in the network, and then determines the depth by comparing the sum of values of each agent in various neighborhoods of the chosen agent to the cost of exploring that neighborhood.

Recall that in the UCB1 algorithm, we have

$$UCB_{i,T} = \bar{X}_{i,T} + \sqrt{\frac{2 \log T}{O_{i,T}}},$$

where $\bar{X}_{i,T}$ is the sample mean of the observed rewards of action i up to time T .

To adapt this to the social network setting, we have to factor in the additional observations of rewards by observing an agent's neighbors. Hence, we denote $B_{i,T}$ for each agent as the total number of times that its reward has been observed by time step T . This can be done by the decision maker choosing the agent or the decision maker choosing one of the agents at most d_t away. Hence, the sample mean of each

agent i 's reward at time step T is

$$\bar{\mu}_{i,T} = \frac{\sum_{t=1}^T Y_{i,t}}{B_{i,T}}$$

Then, we can modify the UCB index to get

$$UCBV_{i,t} = \bar{\mu}_{i,T} + \sqrt{\frac{2 \log T}{B_{i,T}}},$$

for $1 \leq i \leq N$.

Now, just like in the UCB1 algorithm, our algorithm picks at time step T the agent corresponding to

$$I_T = \arg \max_{1 \leq i \leq N} \bar{\mu}_{i,T} + \sqrt{\frac{2 \log T}{B_{i,T}}}.$$

However, our decision maker still has to decide what d_T should be. To do so, we will assign a value $V_{i,T}$ to each agent i . This value is based on how important it is that agent i is observed indirectly. The higher $V_{i,T}$ is, the more desirable it is to observe agent i indirectly, as opposed to choosing it. We will now discuss a few properties that a suitable $V_{i,T}$ should possess.

Firstly, the lower $\bar{\mu}_{i,T}$ is, the higher $V_{i,T}$ should be. This is because we want to choose the agent with the highest mean reward most of the time, and hence leave the suboptimal agents to be explored while choosing the optimal agent as much as possible. The lower the expected reward of the agent, the less likely we would want to choose it and hence the more likely we would want to observe its rewards indirectly. Thus, an agent with a low sample reward mean should have a high $V_{i,T}$. We now define

$$\Delta_{i,T} = \max_{1 \leq j \leq N} \bar{\mu}_{j,T} - \bar{\mu}_{i,T}.$$

This is the difference between the sample mean reward of agent i and that of the agent

with the highest sample mean reward. The higher $\Delta_{i,T}$ is, the higher $V_{i,T}$ should be.

The other thing we need to factor in is the bias term of the UCBV index. In general, the higher the bias term of agent i , the more desirable it is to observe the reward of agent i . Hence, the higher the bias term, the higher $V_{i,T}$ should be. However, that does not tell us the full story. Consider a case of two agents A and B in two separate instances of the multi-armed bandit problem. Let us assume that they have the same values for their sample reward mean and bias term. However, the time step for agent A , T_A , is much smaller than that of agent B , T_B .

If we were only to consider the two agents' bias terms, since they are of the same value, we might conclude that exploring them is of equal importance. However, this is not the case. It is actually much more urgent to explore agent A through indirect observation, as each successive time step that passes without agent A being chosen increases its UCBV index much more than agent B 's. Hence, what is truly important is the number of time steps needed for the UCBV index of agent i to catch up with the current maximum UCBV index. We can define this as $H_{i,T}$, where it is such that

$$\sqrt{\frac{2 \log(T + H_{i,T})}{B_{i,T}}} - \sqrt{\frac{2 \log T}{B_{i,T}}} = \delta_{i,T}, \quad (5.4)$$

and

$$\delta_{i,T} = UCBV_{I_T,T} - UCBV_{i,T}, \quad (5.5)$$

the difference between the UCBV index of agent i and that of the chosen agent at time T . Note that we are only approximating the number of time steps needed, so $H_{i,T}$ may not be an integer. Indeed, we can achieve greater accuracy if we do not restrict ourselves to the integers. The larger $H_{i,T}$ is, the smaller $V_{i,T}$ should be.

Keeping in mind the two properties above, we can now define

$$V_{i,T} = \frac{\Delta_{i,T}}{H_{i,T}},$$

for all agents $1 \leq i \leq N$.

To compute $V_{i,T}$, we need to compute the values of both $\Delta_{i,T}$ and $H_{i,T}$. $\Delta_{i,T}$ is relatively easy to compute.

Proposition 5.4. *Let $H_{i,T}$ and $\delta_{i,T}$ be defined as in (5.4) and (5.5). Then,*

$$H_{i,T} = T \cdot \left(e^{\delta_{i,T} \sqrt{2B_{i,T} \log T} + \frac{\delta_{i,T}^2 B_{i,T}}{2}} - 1 \right).$$

Proof. Define y such that

$$\sqrt{\frac{2 \log(yT)}{B_{i,T}}} - \sqrt{\frac{2 \log T}{B_{i,T}}} = \delta_{i,T}.$$

Then,

$$\begin{aligned} \sqrt{\frac{2 \log(yT)}{B_{i,T}}} - \sqrt{\frac{2 \log T}{B_{i,T}}} = \delta_{i,T} &\iff \sqrt{\log yT} - \sqrt{\log T} = \frac{\delta_{i,T} \sqrt{B_{i,T}}}{\sqrt{2}} \\ &\iff \sqrt{\log y + \log T} - \sqrt{\log T} = \frac{\delta_{i,T} \sqrt{B_{i,T}}}{\sqrt{2}}. \end{aligned}$$

We also have

$$\begin{aligned} \sqrt{\alpha + x} - \sqrt{\alpha} = \beta &\iff x + 2\alpha - 2\sqrt{x\alpha + \alpha^2} = \beta^2 \\ &\iff 2\sqrt{x\alpha + \alpha^2} = x + 2\alpha - \beta^2 \\ &\iff 4x\alpha + 4\alpha^2 = x^2 + 4\alpha^2 + 4x\alpha + \beta^4 - 2x\beta^2 - 4\alpha\beta^2 \\ &\iff x^2 - 2x\beta^2 + \beta^4 = 4\alpha\beta^2 \\ &\iff (x - \beta^2)^2 = 4\alpha\beta^2 \\ &\iff x - \beta^2 = 2\beta\sqrt{\alpha} \\ &\iff x = 2\beta\sqrt{\alpha} + \beta^2. \end{aligned} \tag{5.6}$$

Using (5.6) and substituting $\alpha = \log T$ and $\beta = \frac{\delta_{i,T} \sqrt{B_{i,T}}}{\sqrt{2}}$, we then have

$$\log y = \delta_{i,T} \sqrt{2B_{i,T} \log T} + \frac{\delta_{i,T}^2 B_{i,T}}{2}.$$

and

$$y = e^{\delta_{i,T} \sqrt{2B_{i,T} \log T} + \frac{\delta_{i,T}^2 B_{i,T}}{2}}.$$

Since $yT = T + H_{i,T}$, we then have

$$\begin{aligned} H_{i,T} &= (y - 1) \cdot T \\ &= T \cdot \left(e^{\delta_{i,T} \sqrt{2B_{i,T} \log T} + \frac{\delta_{i,T}^2 B_{i,T}}{2}} - 1 \right). \end{aligned}$$

□

We can now determine what d_T the UCB-V algorithm will choose. It will choose

$$d_T = \arg \max_{k \geq 0} \left(\sum_{j \in N_k(I_T)} (V_{j,T}) - C_k \right).$$

The intuition is as follows. When a non-optimal agent i is chosen at time step T , the expected regret is $\Delta_{i,T}$. If it can be observed indirectly, that will delay having to choose agent i . If T is large enough, the sample mean and bias terms are not likely to change much in each time step. Hence, if agent i is not observed indirectly, it will be approximately $H_{i,T}$ time steps before it will have to be chosen, incurring a regret of $\Delta_{i,T}$. Therefore, if we observe agent i indirectly at time step T , we will save approximately $\Delta_{i,T}$ regret over $H_{i,T}$ time steps, implying an average of $V_{i,T}$ per time step. We can estimate the total regret saved per time step per k -neighborhood of I_T by summing the $V_{i,T}$ values in that neighborhood. Once we subtract the cost for exploration, we can approximate the regret saved by exploring further.

Lastly, we have to determine the first agent that the algorithm chooses. To get as many observations as possible in the first time step, our algorithm will choose the Jordan center of the underlying graph G as its agent for the first time step, with d_1 equal to the radius of G .

We summarize the UCB-V algorithm below.

```

 $I_1 \leftarrow$  Jordan center of  $G$ 
 $d_1 \leftarrow$  radius of  $G$ 
Choose agent  $I_1$ 
Choose depth  $d_1$ 
Receive observations  $\{X_{i,1}\}_{1 \leq i \leq N}$ 
for  $T \geq 2$  do
     $I_T \leftarrow \arg \max_{1 \leq i \leq N} \bar{\mu}_{i,T} + \sqrt{\frac{2 \log T}{B_{i,T}}}$ 
    for  $1 \leq i \leq N$  do
         $V_{i,T} \leftarrow \frac{\Delta_{i,T}}{H_{i,T}}$ 
    end
     $d_T \leftarrow \arg \max_{k \geq 0} \left( \sum_{j \in N_k(I_T)} (V_{j,T}) - C_k \right)$ 
    Choose agent  $I_T$ 
    Choose depth  $d_T$ 
    Receive observations  $\{X_{i,T}\}_{i \in N_{d_T}(I_T)}$ 
end

```

Algorithm 2: The UCB-V algorithm

5.6 Numerical results

In this section, we compare the performances of the UCB-R algorithm against that of the UCB1 algorithm and the UCB-N algorithm. We also compare the performance of the UCB-V algorithm against that of the UCB1 algorithm. The UCB1 algorithm has been described in the previous section. We assume that the UCB1 algorithm is not using any information from the underlying graph G , and is just treating the problem as a standard multi-armed bandit problem.

The UCB-N algorithm, introduced in [19], is an algorithm made for solving the multi-armed bandit problem in a social network setting. However, the main difference between the UCB-N algorithm and ours is that the UCB-N algorithm assumes that we will be receiving unbiased observations from the neighbors of the chosen agent. In the simulation, we will assume that this is the case for the UCB-N algorithm.

We performed a simulation using 100 trials of 1000 time steps each. Each trial

Figure 5-1: Comparison of UCB-R algorithm with UCB1 and UCB-N

uses a random Erdos-Renyi graph with 100 nodes and edge probability 0.2. The means of each agent's reward was uniformly chosen between 0 and 1. For the UCB-R algorithm, we also chose ϵ_μ and ϵ_B values of 0.05. We then take the mean of the cumulative reward of each algorithm over all the trials, and plot it against the time step T . The result is shown in Figure 5-1.

As expected, both the UCB-R algorithm and the UCB-N algorithm perform significantly better than the UCB1 algorithm. In fact, the UCB-R algorithm almost matches the performance of the UCB-N algorithm despite the presence of biased observations. This indicates that the UCB-R algorithm converges to an accurate estimate of the biases very quickly. Both the UCB-R and UCB-N algorithms outperform the UCB1 algorithm largely because to explore suboptimal agents, they do not have to choose them, and can instead explore them indirectly via their neighbors who may provide a better reward.

We also compare the performance of the UCB-V algorithm against that of the UCB1 algorithm. The conditions for this simulation are similar to those that we used for the UCB-N algorithm comparison, except that we have imposed a cost of $0.1k$ for the decision maker to explore the k =neighborhood of the chosen agent. We then take the mean of the cumulative reward of both algorithms over all the trials, and plot it

Figure 5-2: Comparison of UCB-V algorithm with UCB1 (1000 time steps)

against the time step T . The result is shown in Figure 5-2.

As can be seen, the UCB-V algorithm vastly outperforms the UCB1 algorithm. This is to be expected, as just like the UCB-N and UCB-R algorithms, the UCB-V algorithm has the huge advantage of not needing to choose agents with a very low expected reward in order to observe their rewards. This means that it can choose agents with a high expected reward, and occasionally pay the cost to explore the 1-neighborhood or 2-neighborhood around the chosen agents in order to observe the rewards of the other agents. Some agents do not have to be chosen at all. The UCB1 algorithm has no such luxury and is forced to eventually choose each agent an unbounded number of times, given enough time steps.

However, though it may seem that the UCB-V algorithm outperforms the UCB1 algorithm at every time step, this is not the case. Figure 5-3 shows the performance of both algorithms for the first 10 time steps. As can be seen, the UCB-V algorithm does worse than the UCB1 algorithm for the first couple of time steps. This is when the UCB-V algorithm is paying a cost to explore more of the network. The early costs are certainly worth it, though, as this early exploration phase is what allows the UCB-V algorithm to skip choosing agents with a low expected reward.

Note that the network that we used in our simulation is relatively small with a low

Figure 5-3: Comparison of UCB-V algorithm with UCB1 (10 time steps)

radius. Hence, the decision maker does not have to spend much in the initial exploration phase. This also makes it probable that each agent with a low expected reward is neighboring one with a high one, and hence the decision maker will rarely have to explore greater than depth 1. In larger or sparser networks, this may not be the case, and it is likely that the UCB-V algorithm will do worse than the UCB1 algorithm for a greater number of time steps. Still, the UCB-V algorithm will outperform the UCB1 algorithm in the long run.

5.7 Summary

In this chapter, we proposed two different ways of modeling learning in a social network as a variant of the multi-armed bandit problem. In the first, we assumed that learning an agent's reward also gave the decision maker a biased reading of its neighbors' rewards. We then introduced a method of finding estimates for the means of each agent's reward and biases between each agent and its neighbors. We used these estimates in the UCB-R algorithm we proposed in order to select which agent's reward to learn at each time step, and then compared its performance to other multi-armed bandit algorithms.

In the second, we assumed that the decision maker does not observe the rewards of the neighbors of the chosen agent by default, but can instead pay an additional cost to observe the reward of neighbors within a selected radius of the chosen agent, with the cost increasing as the radius increases. We introduced the UCB-V algorithm to determine which agent the decision maker should choose at each time step as well as which radius should be chosen. As with the first model, we also compared our algorithm against other multi-armed bandit algorithms.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we studied the problem of modeling learning in a social network. We modeled that in a tandem network, with agents making a binary decision about the state of the world based on a private signal that they receive and the decision of the previous agent. We proposed a decision rule to allow agents to minimize the worst-case asymptotic error probability, even if there is some uncertainty about the distribution of the previous agents' private signals. We also proposed a rule to allow agents to minimize their personal error probability.

Under these conditions, we showed that if agents are collectively working towards minimizing the asymptotic error probability, and if the level of uncertainty towards each other agent's private signal is the same, then asymptotic learning cannot occur unless there is no uncertainty over the private signals of the previous agent's under at least one of the two states of the world. If agents are selfishly minimizing their own error probability, then no uncertainty over the private signals under both states of the world is required for asymptotic learning to occur.

We then derived conditions for asymptotic learning to occur if the level of uncertainty towards each other agent's private signal can vary, as well as examined the scenario where agents do not know their positions in the network. In that case, asymptotic learning cannot occur, but we proposed a decision rule in order to minimize the

minimax asymptotic error probability.

We next turned our attention to the problem of modeling online advertising in a social network, where a retailer is attempting to gauge its customers' reactions to its products. We modeled this in the form of optimizing a string-submodular function, where because of limited resources, only some of the customers' step-wise gains can be observed at any one time. Under these constraints, we proposed a method for the retailer to decide which group of customers to target at each time step, as well as several strategies on how to pick which groups of customers' step-wise gains to observe for the next time step. We then derived a lower bound for the performance of each of these strategies against that of the optimal string.

Finally, we modeled online advertising again, this time in the guise of a multi-armed bandit problem. We proposed two different model. In one of them, targeting a user will allow the retailer to also gauge the reactions of that user's neighbors in the social network, albeit with some bias involved. In the other, the retailer is given the choice to pay an additional cost in order to gauge the reactions of larger neighborhoods of the targeted user. Under both models, we proposed algorithms to solve the multi-armed bandit problem, and compared their performance to similar algorithms.

6.2 Future Work

There is a lot of potential for future work. Firstly, these are only some of the possible models for modeling learning in a social network. There can be many other models as well.

For the robust detection problem, we have only studied the tandem network. Other network topologies like the star topology can be considered as well. We could also revise the assumptions so that agents have more knowledge of the distributions of the observations of other nearby agents than that of those further away. Lastly, we could consider a situation where some, but not all of the agents know their positions exactly in the network.

For our model on learning in a social network with biased observations, we have only considered the case where the biases stay the same throughout. It may be interesting to consider the case where the biases can change as time passes.

Finally, in our model on learning in a social network with larger neighborhoods, we examined the scenario where the cost is a function of the depth chosen by the decision maker. There can be a lot of variability on the cost function. For instance, the cost could instead be based on the number of neighbors observed. Furthermore, we do not have to limit ourselves to observing fixed neighborhoods of the targeted user. For instance, it might be possible to study the case where any of the agents can be observed with an additional cost based on the distance from the targeted user. We could extend our method of estimating means and biases to distributions which are non-Gaussian. Finally, we could introduce an additional assumption where the decision maker observes a noisy version of the neighbors' rewards, with the noise variance increasing as the distance from the chosen

List Of Publications

JOURNAL PAPERS

- 2015** J. Ho, W. P. Tay, T. Q. S. Quek and E. K. P. Chong, “Robust Decentralized Detection and Social Learning in Tandem Networks”, In IEEE Trans. on Sig. Process., vol. 63, no. 19, pp. 5019-5032, 2015.

CONFERENCE PAPERS

- 2014** J. Ho, W. P. Tay and T. Q. S. Quek, “Robust detection and social learning in tandem networks”, In 2014 IEEE Int. Conf. on Acoustics, Speech and Sig. Process. (ICASSP), vol. , no. , pp. 5457-5461, 2014.

Bibliography

- [1] Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvari. Improved algorithms for linear stochastic bandits. In *Proc. of the 24th Int. Conf. on Neural Information Processing Systems*, NIPS'11, pages 2312–2320, USA, 2011.
- [2] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proc. of the 21st Annual Conf. on Learning Theory (COLT)*, 2008.
- [3] Daron Acemoglu, Munther A. Dahleh, Ilan Lobel, and Asuman Ozdaglar. Bayesian learning in social networks. Working Paper 14040, National Bureau of Economic Research, May 2008.
- [4] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. Taming the monster: A fast and simple algorithm for contextual bandits. In *Proc. of the 31st Int. Conf. on Mach. Learn.*, volume 32 of *Proceedings of Machine Learning Research*, pages 1638–1646, Beijing, China, 22–24 Jun 2014. PMLR.
- [5] Saeed Alaei and Azarakhsh Malekian. Maximizing sequence-submodular functions and its application to online advertising. *CoRR*, abs/1009.4153, 2010.
- [6] Noga Alon, Baruch Awerbuch, and Yossi Azar. The online set cover problem. In *Proc. of the 35th Annual ACM Symposium on Theory of Comp.*, STOC '03, pages 100–105, 2003.
- [7] S. Appadwedula, V. V. Veeravalli, and D. L. Jones. Decentralized detection with censoring sensors. *IEEE Trans. on Sig. Process.*, 56(4):1362–1373, April 2008.
- [8] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Mach. Learn.*, 47(2-3):235–256, May 2002.
- [9] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, January 2003.
- [10] A. Banerjee and D. Fudenberg. Word-of-mouth learning. *Games and Economic Behaviour*, 46:1–22, 2004.

- [11] Abhijit V Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, 107(3):797–817, August 1992.
- [12] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd edition, 2000.
- [13] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *AISTATS*, volume 15 of *JMLR Proc.*, pages 19–26, 2011.
- [14] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100(5):992–1026, 1992.
- [15] Rick Blum, Saleem Kassam, and H. Vincent Poor. Distributed detection with multiple sensors: Part ii-advanced topics. *Proc. of the IEEE*, 85:64 – 79, 02 1997.
- [16] Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR*, abs/1204.5721, 2012.
- [17] Sebastien Bubeck, Gilles Stoltz, Csaba Szepesvari, and Remi Munos. Online optimization in x-armed bandits. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 201–208. Curran Associates, Inc., 2009.
- [18] Swapna Buccapatnam, Atila Eryilmaz, and Ness B. Shroff. Stochastic bandits with side observations on networks. In *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '14, pages 289–300, New York, NY, USA, 2014. ACM.
- [19] Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. In *Proc. of the Twenty-Eighth Conf. on Uncertainty in Art. Int.*, UAI'12, pages 142–151, Arlington, Virginia, United States, 2012. AUAI Press.
- [20] Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. *CoRR*, abs/1306.0811, 2013.
- [21] J.-F. Chamberland and V.V. Veeravalli. Decentralized detection in sensor networks. *Trans. Sig. Proc.*, 51(2):407–416, February 2003.
- [22] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proc. of the 21st Annual Conf. on Learning Theory (COLT)*, 2008.
- [23] K. Drakopoulos, A. Ozdaglar, and J.N. Tsitsiklis. On learning with finite memory. *IEEE Trans. Inf. Theory*, 59(10):6859–6872, 2013.

- [24] Uriel Feige, Laszlo Lovasz, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.
- [25] Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, July 2011.
- [26] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 586–594. Curran Associates, Inc., 2010.
- [27] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, number 2 in Proceedings of Machine Learning Research, pages 757–765, Beijing, China, 22–24 Jun 2014. PMLR.
- [28] E. Geraniotis and Y. A. Chau. Robust data fusion for multisensor detection systems. *IEEE Trans. on Information Theory*, 36(6):1265–1279, Nov 1990.
- [29] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *Proc. of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 982–991, 2008.
- [30] Peter J. Huber and Volker Strassen. Minimax tests and the Neyman-Pearson lemma for capacities. *Ann. Statist.*, 1(2):251–263, March 1973.
- [31] PJ Huber. A robust version of the probability ratio test. In *Ann. Math. Statist.*, volume 36, pages 1753–1758, 1965.
- [32] F. Iannello and O. Simeone. On the optimal scheduling of independent, symmetric and time-sensitive tasks. *IEEE Trans. on Autom. Cont.*, 58(9):2421–2425, Sept 2013.
- [33] Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Efficient bandit algorithms for online multiclass prediction. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 440–447, New York, NY, USA, 2008. ACM.
- [34] S.A. Kassam and H.V. Poor. Robust techniques for signal processing: A survey. *Proc. of the IEEE*, 73(3):433–481, March 1985.
- [35] David Kempe, Jon Kleinberg, and Eva Tardos. Influential nodes in a diffusion model for social networks. In *Proc. of the 32nd International Conference on Automata, Languages and Programming*, ICALP'05, pages 1127–1138, 2005.
- [36] Andreas Krause and Daniel Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, February 2014.

- [37] T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. Appl. Math.*, 6(1):4–22, March 1985.
- [38] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in Neural Information Processing Systems 20*, pages 817–824. 2008.
- [39] C. Li and N. Elia. A sub-optimal sensor scheduling strategy using convex optimization. In *Proc. of the 2011 American Cont. Conf.*, pages 3603–3608, June 2011.
- [40] C. Li and N. Elia. Stochastic sensor scheduling via distributed convex optimization. *ArXiv e-prints*, March 2014.
- [41] Y. Liu, E. K. P. Chong, and A. Pezeshki. Bounding the greedy strategy in finite-horizon string optimization. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 3900–3905, Dec 2015.
- [42] Julian Lorenz, Martin Marciniszyn, and Angelika Steger. Observational learning in random networks. In Nader H. Bshouty and Claudio Gentile, editors, *Learning Theory*, pages 574–588, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [43] Jun Lu and T. Suda. Coverage-aware self-scheduling in sensor networks. In *2002 14th International Conference on Ion Implantation Technology Proceedings (IEEE Cat. No.02EX505)*, pages 117–123, Oct 2003.
- [44] Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Proc. of the 24th Int. Conf. on Neural Inf. Proc. Systems*, NIPS’11, pages 684–692, USA, 2011.
- [45] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5), October 2007.
- [46] Elchanan Mossel and Sebastien Roch. On the submodularity of influence in social networks. In *Proc. of the 39th Annual ACM Symposium on Theory of Comp.*, STOC ’07, pages 128–134, 2007.
- [47] Kamesh Munagala, Shivnath Babu, Rajeev Motwani, and Jennifer Widom. The pipelined set cover problem. In *Proc. of the 10th Int. Conf. on Database Theory*, ICDT’05, pages 83–98, Berlin, Heidelberg, 2005. Springer-Verlag.
- [48] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions I. In *Mathematical Programming*, volume 14, pages 265–294, 1978.
- [49] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions II. In *Mathematical Programming*, volume 8, pages 73–87, 1978.

- [50] Sandeep Pandey, Deepayan Chakrabarti, and Deepak Agarwal. Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, pages 721–728, New York, NY, USA, 2007. ACM.
- [51] D Papastavrou and Michael Athans. Distributed detection by a large team of sensors in tandem. In *IEEE Trans. Aerosp. Electron. Syst.*, volume 28, pages 639–653, 1992.
- [52] Yevgeny Seldin, Peter Auer, John S. Shawe-taylor, Ronald Ortner, and François Laviolette. PAC-bayesian analysis of contextual bandits. In *Advances in Neural Information Processing Systems 24*, pages 1683–1691. 2011.
- [53] Aleksandrs Slivkins. Contextual bandits with similarity information. *J. Mach. Learn. Res.*, 15(1):2533–2568, January 2014.
- [54] Lones Smith and Peter Sorensen. Pathological outcomes of observational learning. *Econometrica*, 68(2):371–398, 2000.
- [55] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. *Advances in Neural Information Processing Systems 21*, pages 1577–1584, 2008.
- [56] Xinmiao Sun, Christos G. Cassandras, and Xiangyu Meng. A submodularity-based approach for multi-agent optimal coverage problems. In *56th IEEE Annual Conference on Decision and Control, CDC 2017, Melbourne, Australia, December 12-15, 2017*, pages 4082–4087, 2017.
- [57] Wee Peng Tay, John N. Tsitsiklis, and Moe Z. Win. On the sub-exponential decay of detection error probabilities in long tandems. *IEEE Trans. Inf. Theory*, pages 4767–4771, 2008.
- [58] John N. Tsitsiklis. Decentralized detection. In *Advances in Statistical Signal Processing*, pages 297–344. JAI Press, 1993.
- [59] P.K. Varshney. *Distributed Detection and Data Fusion*. Springer-Verlag, New York, 1997.
- [60] V. V. Veeravalli, T. Basar, and H. V. Poor. Decentralized sequential detection with a fusion center performing the sequential test. *IEEE Trans. on Information Theory*, 39(2):433–442, March 1993.
- [61] VV Veeravalli, T Basar, and HV Poor. Minimax robust decentralized detection. In *IEEE Trans. Inf. Theory*, volume 40, pages 35–40, 1994.
- [62] R. Viswanathan, S. C A Thomopoulos, and R. Tumuluri. Optimal serial distributed decision fusion. *IEEE Trans. Aerosp. Electron. Syst.*, 24(4):366–376, 1988.

- [63] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors part I. fundamentals. *Proc. of the IEEE*, 85(1):54–63, Jan 1997.
- [64] Chih-Chun Wang, S. R. Kulkarni, and H. V. Poor. Bandit problems with side observations. *IEEE Trans. on Autom. Cont.*, 50(3):338–355, March 2005.
- [65] Chih-Chun Wang, Sanjeev R. Kulkarni, and H. Vincent Poor. Arbitrary side observations in bandit problems. *Advances in Applied Mathematics*, 34(4):903 – 938, 2005.
- [66] Zengfu Wang, Bill Moran, Xuezhi Wang, and Quan Pan. Approximation for maximizing monotone non-decreasing set functions with a greedy method. *Journal of Combinatorial Optimization*, 31(1):29–43, Jan 2016.
- [67] Qingyun Wu, Huazheng Wang, Quanquan Gu, and Hongning Wang. Contextual bandits in a collaborative environment. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR ’16, pages 529–538, New York, NY, USA, 2016. ACM.
- [68] Yuhong Yang and Dan Zhu. Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates. *Ann. Math. Stat.*, 30(1):100–121, 02 2002.
- [69] Z. Zhang, E. K. P. Chong, A. Pezeshki, W. Moran, and S. D. Howard. Submodularity and optimality of fusion rules in balanced binary relay trees. In *2012 IEEE 51st IEEE Conf. on Decision and Control (CDC)*, pages 3802–3807, Dec 2012.
- [70] Z. Zhang, Z. Wang, E. K. P. Chong, A. Pezeshki, and W. Moran. Near optimality of greedy strategies for string submodular functions with forward and backward curvature constraints. In *52nd IEEE Conf. on Decision and Control*, pages 5156–5161, Dec 2013.
- [71] Zhenliang Zhang, Ali Pezeshki, William Moran, Stephen D Howard, and Edwin KP Chong. Error probability bounds for binary relay trees. In *IEEE Trans. Inf. Theory*, volume 58, pages 3548–3563, 2012.
- [72] ZL Zhang, EKP Chong, A Pezeshki, W Moran, et al. String submodular functions with curvature constraints. *IEEE Trans. on Autom. Cont.*, 61(3):601–616, 2016.