Information Cascades on Arbitrary Topologies*

Jun Wan[†1], **Yu Xia**[‡2], **Liang Li**[3], **and Thomas Moscibroda**[4]

1 The Institute for Theoretical Computer Science(ITCS), Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
wanj12@mails.tsinghua.edu.cn

2 The Institute for Theoretical Computer Science(ITCS), Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
xiay12@mails.tsinghua.edu.cn

3 Microsoft Research Asia, Beijing, China
liangl@microsoft.com

4 Microsoft Research Asia, Beijing, China
moscitho@microsoft.com

Abstract

In this paper, we study information cascades on graphs. In this setting, each node in the graph represents a person. One after another, each person has to take a decision based on a private signal as well as the decisions made by earlier neighboring nodes. Such information cascades commonly occur in practice and have been studied in complete graphs where everyone can overhear the decisions of every other player. It is known that information cascades can be fragile and based on very little information, and that they have a high likelihood of being wrong.

Generalizing the problem to arbitrary graphs reveals interesting insights. In particular, we show that in a random graph $G(n, q)$, for the right value of q, the number of nodes making a wrong decision is logarithmic in n. That is, in the limit for large n, the fraction of players that make a wrong decision tends to zero. This is intriguing because it contrasts to the two natural corner cases: empty graph (everyone decides independently based on his private signal) and complete graph (all decisions are heard by all nodes). In both of these cases a constant fraction of nodes make a wrong decision in expectation. Thus, our result shows that while both too little and too much information sharing causes nodes to take wrong decisions, for exactly the right amount of information sharing, asymptotically everyone can be right. We further show that this result in random graphs is asymptotically optimal for any topology, even if nodes follow a globally optimal algorithmic strategy. Based on the analysis of random graphs, we explore how topology impacts global performance and construct an optimal deterministic topology among layer graphs.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases Information Cascades, Herding Effect, Random Graphs

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.64

* The full version of this paper can be found in http://arxiv.org/abs/1604.07166

† This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61361136003.

‡ Part of this work was done when this author visited Microsoft Research Asia.

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 64; pp. 64:1–64:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

An Information Cascade occurs when a person observes the actions of others and then – in spite of possible contradictions to his/her own private information – follows these same actions. A cascade develops when people "abandon their own information in favor of inferences based on earlier people's actions"[12]. Information Cascades frequently occur in everyday life. Commonly cited examples include the choice of restaurants when being in an unknown place people choose the restaurant that already has many guests over a comparatively empty restaurant, or hiring interview loops where interviewers follow earlier interviewer's decisions if they are not sure about the candidate. Notice that information cascades are not irrational behavior; on the contrary, they occur precisely because people rationally decide based on inferences derived from earlier people's actions.

The simple herding experiment by Anderson and Holt illustrates Information Cascades [3, 4](see also Chapter 16 in [12]). In this experiment, an urn contains three marbles, either two red and one blue (majority red), or one red and two blue (majority blue). The players do not know whether the urn is majority red or blue. One by one, the players privately pick one marble from the urn, check its color, return it to the urn, and then publicly announce their guess as to whether the urn is majority red or majority blue. The first and second player will naturally base their guesses on the colors of the marble they picked, thus their guesses reveals their private signals. For any subsequent player however, her rational guess may not reflect her own signal. For example, suppose the first two players both guess *red*. In this case, it is rational for the third player to also guess *majority-red* regardless of the color of the marble she picked. Indeed, the third player makes her decision on a rational inference based on the first two guesses. Since her guess does therefore not reveal any further information about the urn to any subsequent player, *every* subsequent player will guess the urn to be *majority-red*. The example shows that information cascades can be based on very little actual information and thus fragile; and they can be wrong. Indeed, in the above example with urns, it can be shown that with probability $1/5$, a "wrong cascade" occurs, i.e., all players (except from possibly a few at the beginning) will guess wrongly.

The standard model for information cascades studies the process in which players make decisions sequentially based on their own private signals as well as the set of decisions made by earlier players [6, 7, 23]. In this paper, we interpret and generalize the traditional information cascade setting as a game in a graph. Each player is a node, and an edge between two nodes v and w means that w can hear about v's guess (assuming w is after v in the order of decision-making). Thus the traditional information cascade model corresponds to a complete graph (all players hear the decisions of all other players). At the other end of the spectrum, the empty graph means that every player decides independently of all other players, purely based on their own private signal. Casting the information cascade problem in this graph setting allows us to study the range in between the two extreme points of complete and empty graphs.

Studying this range in between reveals fascinating insights. Figure 1 shows the expected number of wrong guesses in the above 3-marble-urn experiment in a random graph $G(n, q)$ topology, for different values of n and q. In the empty graph ($q = 0$), if all nodes take their decisions independently, $1/3$ of the players are wrong. In the complete graph ($q = 1$), $1/5$ of the players are wrong on average as discussed above. However, the interesting thing is that for some values in between these two extremes, the number of wrong decisions is significantly less. Indeed, it seems that for the right value of q and $n \to \infty$, the number of wrong decisions tends to 0.

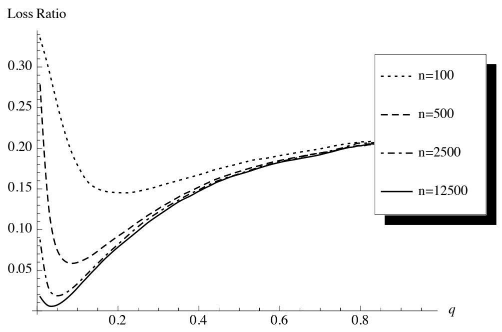

Figure 1 Performance of random graphs for different q and n.

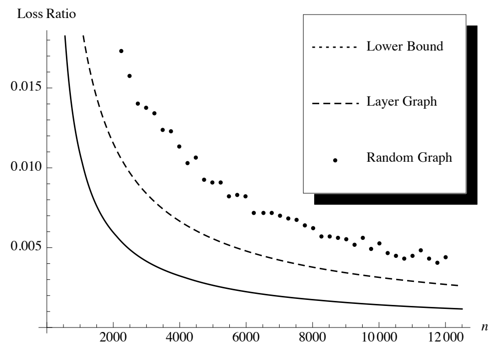

Figure 2 Performance of different topologies and strategies.

These observations are intriguing: It looks like that if people share too much information, a constant fraction of the population is wrong because of bad information cascades occurring. If people share too little information, a larger constant fraction of the population is wrong because the players take their decisions too independently, relying too much on their private signal which has a constant probability of being wrong. But, if exactly the right amount of information is shared, then it seems that in the limit, *all players* (at least asymptotically) take the correct decision.

In this paper, we study this phenomenon. We prove that, indeed, in a random graph the number of wrong nodes is at most $O(\log n)$ for the optimal value of q (Section 3). We then study arbitrary graph topologies and show that $O(\log n)$ wrong nodes is optimal in a strong sense (Section 4). Specifically, even in the best possible topology, there are at least $\Omega(\log n)$ wrong nodes. This result holds even if a global oracle tells each node whether it should (a) base its decision solely on its private signal (thus revealing this signal as additional information to all its neighbors) or (b) base its decision on the majority of private signal and neighboring decisions as in the cascade model above. In other words, even if nodes can "sacrifice" themselves to reveal additional information to their neighbors and even in the best possible topology $\Omega(\log n)$ wrong nodes is a lower bound. Finally, we derive an optimal deterministic topology from among a family of *layer graphs* (Section 5). Detailed proofs and analysis can be found in [22].

2 Related Work

Sequential decision-making has been studied in various areas including politics, economics and computer science[21, 6, 7, 14, 12]. The primary concern on the Bayesian learning model[7, 19, 6, 23, 20, 5, 1] is under what conditions asymptotically correct information cascades occur. For specific graph topologies such as complete graphs and line graphs, conditions on the private signals were addressed to guarantee the correctness of cascades[19, 9]. For arbitrary graph topologies, the approach of Acemoglu et al. [1] is intuitively quite consistent to our k-layer topology(see Section 6) and can be used to explain why our random network and selfishless decision-making algorithm achieve global optimality. While their approach focuses on the asymptotic probability of correct cascades, our result can quantitatively bound the expectation number of incorrect nodes.

There has been research on different sequential decision making models in graphs. For example, Chierichetti et al. [10] study different algorithms for finding appropriate orderings to maximize the fraction making correct decisions and Hajiaghayi et al. [16] and Hajiaghayi et al. [15] generalizes the model and improve related bounds. However, notice that the threshold decision-making processes studied in these works fundamentally differ from the information cascade setting we consider in this work. Indeed, the effect of too little/too much information sharing being bad as shown in Figure 1 is not observed in such threshold models.

There also exists an impressive body of work on sequential and non-sequential decision on arbitrary graphs that however do not capture information cascades as exemplified in Anderson and Holt's herding experiment. Typically, each node updates its opinion through repeated averaging with neighbors. General conditions for convergence to consensus have been developed[2, 11, 13]. Intrigued by the observation that consensus is usually not reached in real world[18], Bindal et al. [8] use a game theoretic approach to study the equilibrium of the dynamical process and measure the cost of disagreement via the Price of Anarchy[17].

3 Preliminaries

We introduce the formal definitions of our model. There are n nodes (numbered $1, 2, \cdots n$) whose neighboring relationship is depicted by a graph $G = ([n], E)$. All nodes make decisions sequentially according to their numbers in order to guess a global ground truth value $b \in \{0, 1\}$. When making its decision, each node can only obtain a random partial information on b. That is, when node i observes b, it can only get a *private signal* s_i which equals b with probability $p > 0.5$ or equals $1 - b$ with probability $1 - p$. The decision-making of a node not only depends on its private signal observed from b, but also on the decisions made by its previous neighbors. Note that the neighboring decisions may or may not be based on those nodes' private signals. More formally, let c_i be the *output decision* or *guess* of node i and c^i be the *decision vector* $(c_1, c_2, \cdots, c_i)$, if $L_i : \{0, 1\}^{i-1} \times \{0, 1\} \to \{0, 1\}$ is the *decision-making algorithm* for node i, we have in general $c_i = L_i(c^{i-1}, s_i)$.

Given the graph G and decision-making algorithms $L_1, L_2, \cdots, L_n$, we use $\mathcal{E}_G(L_1, \cdots, L_n)$ to denote the expected number of nodes that output the wrong value $1 - b$. When it is clear from the context, we may abbreviate this notation to $\mathcal{E}_G$, $\mathcal{E}(L_1, \cdots, L_n)$, or simply $\mathcal{E}$. The global objective of this sequentially decision-making process is to minimize $\mathcal{E}_G(L_1, \cdots, L_n)$, which is equivalent to maximizing the expected number of nodes who guess the ground truth value correctly. We will show that such an optimization task can be achieved by adjusting the graph topology or the decision-making algorithms.

Let $\mathcal{E}_i$ be the *failure probability* that node i outputs $1 - b$. As a node often makes inferences based on others' decisions without knowing their private signals, it is intuitively understandable that a node's probability of correct decision-making can be quantified by the number of private signals it can infer.

In reality, the Majority Algorithm is one of the most popular and practical algorithms for decision-makings. This kind of "following the herd" algorithm can often achieve a locally optimal effect. In this paper, we use Maj_k to denote the Majority Algorithm taking input bits of length k. We just use Maj if k is clear from the context. In Chapter 16 of [12], Easley and Kleinberg shows that the Majority Algorithm is optimal when a node observes multiple independent signals.

▶ **Claim 1.** *For any node i seeking to maximize $\mathcal{E}_i$, when it observes multiple signals(including its own private signal), its optimal algorithm is to output the majority of these observed signals.*

However, Anderson and Holt's experiment shows that if all nodes apply the Majority Algorithm, it is possible that essentially all of the nodes guess incorrectly, leading to an information cascade on the wrong side. In this paper, we address this problem and analyze the impact of topology and algorithms on information cascades.

4 Random Graphs

In this section, we analyze the performance of the Majority Algorithm on random graphs. Conventionally, $G(n,q)$ denotes the random graph model that generates a random graph with n nodes and each pair of nodes are connected by an edge with probability q. Different connection probabilities induces completely different topologies, and thus dramatic changes in $\mathcal{E}_{G(n,q)}$. As introduced in Section 1, when q equals 0 or 1 corresponding to the empty or complete topology, the expected number of wrong output decisions are both $\Theta(n)$.

In this section, we show that there exists a q such that the Majority Algorithm can achieve only $\Theta(\log n)$ expected wrong output decisions:

▶ **Theorem 2.** *There exists a connection probability $q = \Theta(1/\log n)$ such that in $G(n,q)$, when all nodes apply the Majority Algorithm, we have $\mathcal{E}_{G(n,q)} = \Theta(\log n)$.*

We can also demonstrate the optimality of this connection probability by showing a lower bound for the expected number of wrong decisions:

▶ **Theorem 3.** *For any connection probability q, when all nodes apply the Majority Algorithm, the expected number of wrong outputs in $G(n,q)$ is lower bounded by $\Theta(\log n)$, i.e. $\mathcal{E}_{G(n,q)} = \Omega(\log n)$.*

A key ingredient to our proof is to bound the failure probability $\mathcal{E}_i$ for each node. Applying the Chernoff Bound and the Union Bound, we can further bound the overall $\mathcal{E}_{G(n,q)}$. The following are two technical lemmas for bounding the failure probabilities:

▶ **Lemma 4.** *If a constant fraction $f > 0.5$ of the first i nodes are correct (resp. wrong), node $i+1$'s failure (resp. correct) probability $\mathcal{E}_{i+1}$ is upper bounded by $e^{-\Theta(iq)}$.*

▶ **Lemma 5.** *If a constant fraction $f > 0.5$ of the first i nodes are correct, s.t. $\frac{f}{1-f} \geq \sqrt{\frac{q}{1-q}}$, then node $i+1$'s failure probability is upper bounded by p.*

4.1 Proof of Theorem 2

In this subsection, we provide a detailed proof of Theorem 2.

The reason why random graphs behave well for the right value of q is that randomness defers the process of information cascades. The fewer neighbors, the more likely a not will output its private signal, thereby 1) having a high probability of being wrong, but 2) revealing important information to its neighbors. When $q = \Theta(1/\log n)$, with high probability each of the first $\Theta(\log n)$ nodes have at most one neighbor. By definition of Majority Algorithm, any node with only one neighbor will be forced to output its own private signal.

Using Lemma 4, we can prove that an established cascade among the first $\log n/q$ nodes decides the outputs of all later nodes with high probability:

▶ **Lemma 6.** *If among the first $\log n/q$ nodes, only a small constant fraction $f < 0.5$ output wrongly, then for the later $n - (\log n/q)$ nodes, the expected number of wrong outputs is at most $O(1)$.*

Lemma 6 is insufficient to bound the $\Theta(\log n)$ expected failure nodes as required by Theorem 2, in that it only bounds the loss of later nodes in the sequence. It could be the case that the first $\log n/q = \Theta(\log^2 n))$ nodes all fail. To bound the overall $\mathcal{E}$, it is essential to analyze the performance of the first $\log n/q$ nodes. We can use an induction argument to show that for the optimal q, the first $\log n/q$ nodes are majority-correct with high probability:

▶ **Lemma 7.** *Let $\delta = \frac{1}{2}(p + \frac{\sqrt{p}}{\sqrt{p}+\sqrt{1-p}})$. There exists connection probability $q_{opt} = \Theta(1/\log n)$ such that the first $\Theta(\log n/q)$ nodes contains at least δ portion of correct outputs with probability $1 - O(n^{-1}\log n)$.*

Proof. We can prove Lemma 7 by induction. Consider dividing the first $\Theta(\log n/q)$ nodes into $\Theta(\log n)$ segments, where each segment contains $\Theta(1/q) = \Theta(\log n)$ many nodes. We analyze each segment independently and show that

- There exists $q_1 = \Theta(1/\log n)$, such that the first segment contains δ portion of correct outputs with probability at least $1 - O(n^{-1})$.
- If the first i segments contain δ portion of correct outputs, then the $(i+1)^{th}$ segment will also contain δ portion of correct outputs with probability $1 - O(n^{-1})$.

By a single Union Bound, we can combine these two results and show that the first $\Theta(\log n/q)$ nodes contain δ portion of correct outputs with probability $1 - \log n \cdot O(n^{-1})$. ◀

Lemma 7 and 4 together imply a $\Theta(\log n)$ upper bound for the expected number of wrong outputs among the first $\Theta(\log n/q)$ nodes:

▶ **Lemma 8.** *If $q = q_{opt}$, the first $\Theta(\log n/q)$ nodes' expect to have at most $\Theta(\log n)$ many wrong outputs.*

Proof of Theorem 2. With Lemma 6, 7 and 8 proved, the expected number of wrong output decisions under connection probability q_{opt} is bounded by

$$\begin{aligned}\mathcal{E}_{G(n,q)} &= \sum_{i=1}^{n} \mathcal{E}_i = \sum_{i=1}^{\log n/q} \mathcal{E}_i + \sum_{i=\log n/q+1}^{n} \mathcal{E}_i \\ &\le \Theta(\log n) + (1 - O(n^{-1}\log n)) \cdot O(1) + O(n^{-1}\log n) \cdot n = \Theta(\log n).\end{aligned} \tag{1}$$

which completes the proof. ◀

4.2 Proof of Theorem 3

In the previous section, we prove that for the optimal connection probability $q_{opt} = \Theta(1/\log n)$, the expected number of wrong outputs is reduced to $\Theta(\log n)$. However, it remains a problem whether we can move beyond $\Theta(\log n)$. In this section, we prove Theorem 3 which states that the bound in Theorem 2 is asymptotically optimal.

Proof of Theorem 3. We prove this theorem for two separate cases, namely when $q = O(1/\log n)$ and $q = \omega(1/\log n)$.

When $q = O(1/\log n)$, the intuition is that we need at least $\Theta(1/q)$ nodes before accumulating an actual influential cascade. For the i^{th} nodes where $i \le 1/q$, its chance of being isolated is $(1-q)^i + iq(1-q)^{i-1} \ge (1-q)^{1/q} \sim 1/e$. Therefore the node's failure probability is at least $\mathcal{E}_i = (1-p) \cdot \Pr[\text{isolated}] = (1-p)/e$. This lower bounds the expected number of failure nodes by $(1-p)/(eq) = \Theta(1/q)$.

When $q = \omega(1/\log n)$, a wrong cascade occurs with high probability, thus resulting in a significant number of failure nodes. With probability $(1-p)^{\Theta(1/q)}$, all of the first $\Theta(1/q)$ nodes observe a wrong signal and output the wrong guesses. Using Lemma 4, we can show that with high probability, the majority of later nodes follow this wrong cascade. Therefore, the total number of failure nodes is at least $(1-p)^{\Theta(1/q)} \cdot \Theta(n) = n^{1-o(1)} = \Omega(\log n)$. ◀

5 General Lower Bound

In this section, we design a non-constructive scheme that finds the optimal decision-making algorithms for general graphs. Given the graph $G = ([n], E)$, our goal is to find the set of algorithms $\{L_i\}_{i=1}^{n}$ such that $(L_1, \cdots, L_n) = \arg\min_{L'_1,\cdots,L'_n} \mathcal{E}_G(L'_1, \cdots, L'_n)$.

An important use of the non-constructive scheme is to provide a general lower bound for arbitrary topology. For any set of decision-making algorithms in a topology G, we can simulate it on a complete graph by considering only edges in G. Thus the minimal $\mathcal{E}$ for complete graphs is a general lower bound for arbitrary topology:

▶ **Theorem 9.** *The expected number of wrong nodes $\mathcal{E}$ under the optimal decision-making algorithms of complete graphs lower bounds the $\mathcal{E}$ of any algorithms in any topology.*

From our previous discussion on random graphs, we know that the expected number of wrong guesses $\mathcal{E}$ highly depends on the number of nodes revealing their private signals. This inspires us to make the following definitions:

▶ **Definition 10.** Node i reveals **valid** information under c^{i-1} if and only if node i outputs its private signal under c^{i-1}, i.e. $c_i = L_i(c^{i-1}, s_i) = s_i$. Furthermore, we denote Valid($\cdot$) as a function that extracts a vector of valid information out of a decision vector, i.e. c_j is in the vector Valid(c^i) if and only if c_j is valid.

▶ **Definition 11.** A node i's **reveal set** RS_i is the set of c^{i-1} which causes node i to reveal **valid** information.

Note that any valid information is correct with probability p and is independent of other nodes. Using the same Bayesian argument[12], we can prove a similar lemma as Claim 1 in Section 3, which states that a node's guess is beneficial for later nodes if and only if the guess is valid:

▶ **Lemma 12.** *For a node i seeking to minimize its failure probability $\mathcal{E}_i$, the optimal decision-making algorithm is to perform the Majority Algorithm on Valid$(c^{i-1}) \bigcup \{s_i\}$, i.e. $c_i = Maj(Valid(c^{i-1}), s_i)$.*

5.1 A non-constructive optimal algorithm scheme for general graphs

In this section, we provide a general scheme for finding the optimal decision-making algorithms of all nodes in arbitrary topologies. Our scheme is non-constructive in that it neither explicitly specifies what the optimal algorithms are, nor shows how to find them efficiently.

Given the underlying topology, all the nodes decide their algorithms sequentially in a *greedy* way as follows. Node 1 publicly announces L_1, based on its own rationality, then node 2 announces L_2 with the knowledge of L_1, etc(see Algorithm 1). Any node i will base its knowledge on $L_1, \cdots, L_{i-1}$ when deciding L_i. Each node designs its own decision-making algorithm in order to locally minimizes the failure probability. Denote this construction scheme as $GC(\cdot)$, the abbreviation of "greedy construction", then for node i, we have $L_i = GC(L_1, \cdots, L_{i-1})$. We can prove by contradiction that such a locally optimal scheme lead to an overall optimality:

▶ **Theorem 13.** *$L_1, L_2, \ldots, L_n$ constructed as in Algorithm 1 minimizes the expected number of wrong nodes, i.e. $\mathcal{E}(L_1, L_2, \ldots, L_n)$.*

Algorithm 1 A non-constructive optimal algorithm scheme for general graphs

Given $L_1, \cdots, L_{n-1}$, node n constructs L_n that aims at minimizing its own failure probability $\mathcal{E}_n$.

Given $L_1, \cdots, L_{n-2}$, and also the fact that node n is greedy, node $n-1$ constructs L_{n-1} such that the overall loss of him and node n is minimized.

This process continues. Each L_i greedily minimizes the expected number of wrong nodes after among $\{i, \cdots, n\}$ given $L_1, \cdots, L_{i-1}$.

Node 1 knows that all later nodes are "greedy". Their algorithms $L_2, \cdots, L_n$ can all be written as a function of L_1. It then constructs L_1 such that $\mathcal{E}$ is minimized.

Knowing what L_1 is, we can backtrack L_2, and recursively all the output algorithms L_i.

5.2 Optimal algorithms for complete graphs

In this section, we specify the optimal decision-making algorithms for complete graphs and thus provide a general lower bound for our model(by Theorem 17). Several intrinsic properties regarding information cascades in complete graph will also be presented.

We start with a lemma showing that optimal algorithm will either reveal **valid** information or perform Majority Algorithm on all previous guesses.

▶ **Lemma 14.** *In the optimal algorithm, a node either reveals* ***valid*** *information or apply Majority Algorithm on all previous outputs, i.e.*

$$L_i = \begin{cases} s_i & c^{i-1} \in RS_i \\ Maj(c^{i-1}) & c^{i-1} \notin RS_i \end{cases}.$$

It is worth pointing out several non-trivial points of Lemma 14 : (a) the Majority Algorithm is performed on previous guesses only and ignores its own private signal; (b) the Majority Algorithm is performed on *all* previous guesses, not only on the valid guesses. An established result in the proof of Lemma 14 is that the Majority Algorithm will cascade on complete graphs, i.e. if a node performs Majority Algorithm, all later nodes will also perform Majority Algorithm. This implies the existence of a switching point, where all nodes prior to this point reveal their private signals, and all later nodes perform Majority Algorithm based on former nodes' signals. If we can estimate the position of this switching point, then an estimation of $\mathcal{E}$ is achieved.

Lemma 14 specifies a node's action outside the reveal set. However, to get an explicit representation of L_i, an understanding of the reveal set itself is required. We introduce the following lemma that fills this gap.

Denote $\text{diff}(c^{i-1}) = (\#\ 1 \text{ in Valid}(c^{i-1})) - (\#\ 0 \text{ in Valid}(c^{i-1}))$, which serves as a criteria to measure the strength of valid information in previous decision vector $c^{i-1} = (c_1, \cdots, c_{i-1})$.

▶ **Lemma 15.** *The reveal set of a node i can be explicitly expressed with respect to some parameters $\delta_n(\cdot)$, where $RS_i = \{c^{i-1} : |diff(c^{i-1})| \geq \delta_n(i)\}$.*

Proof. This lemma follows from the fact that a node outputs based on the Bayesian probability for the ground truth bit b, which depends solely upon $\text{diff}(c^{i-1})$. Given c^{i-1}, the Bayesian probability for b is

$$\begin{cases} \Pr[b = 0|c^{i-1}] = \frac{1}{1+(\frac{p}{1-p})^{\text{diff}(c^{i-1})}} \\ \Pr[b = 1|c^{i-1}] = \frac{1}{1+(\frac{1-p}{p})^{\text{diff}(c^{i-1})}} \end{cases}. \tag{2}$$

Lemma 14 implies that for each node, (a) if the previous decision vector convinces it that b equals $\text{Maj}(c^{i-1})$ with high probability, it follows the majority of former output decisions; (b) otherwise, it tries to provide more information by revealing its own private signal. As implied by Equation (2), the larger $|\text{diff}(c^{i-1})|$ is, the more likely $b = \text{Maj}(\text{Valid}(c^{i-1}))$. This lead us to conclude the existence of a threshold $\delta_n(i)$ such that L_i applies Majority Algorithm if and only if $|\text{diff}(c^{i-1})| \geq \delta_n(i)$. ◀

Finally, given i and n as input, we show how to efficiently derive $\delta_n(i)$ in average $O(\log n)$ time. Denote $\mathcal{E}(i, d)$ to be the expected number of wrong nodes given that $|\text{diff}(c^{i-1})| = d$. The idea is to use recursion to derive $\mathcal{E}(i, d)$ for all i and d, in the process of which $\{\delta_n(i)|i\}$ may be calculated. If a node k chooses to reveal its private signal, $\mathcal{E}(k, d)$ is updated as

$$\mathcal{E}(k, d) = q_1 \cdot \mathcal{E}(k+1, d+1) + (1 - q_1) \cdot \mathcal{E}(k+1, d-1),$$

where q_1 is the probability that node k's private signal matches the majority of former guesses. Similarly, if node k chooses to do Majority Algorithm, $\mathcal{E}(k, d)$ is updated as

$$\mathcal{E}(k, d) = q_2 \cdot \left(n - \frac{k+d}{2}\right) + (1 - q_2) \cdot \frac{k+d}{2},$$

where q_2 is the probability that the majority of former outputs is correct. Therefore, we can calculate $\{\mathcal{E}(i, d)\}$ in time $O(n^2)$. A further improvement can be made by exploiting the properties of $\delta_n()$: $\delta_n(i+1) - 1 \leq \delta_n(i) \leq \delta_n(i+1) + 1$, Thus to calculate $\{\delta_n(i)\}$, it suffices to calculate $\{\mathcal{E}(i, d) \mid d < \delta_n(i), i \geq n\}$, which requires only $n \cdot \max_i\{\delta_n(i)\} = O(n \log n)$ time complexity.

▶ **Lemma 16.** *Given n as input, we can calculate the set $\{\delta_n(i)\}$ in $O(n \log n)$ time.*

5.3 General lower bound for our model

Finally we analyze the expected number of wrong nodes for the optimal algorithms in complete graphs, and provide a general $\Theta(\log n)$ lower bound for the model.

▶ **Theorem 17.** *The expected number of wrong nodes $\mathcal{E}$ for any topology and any algorithm is at least $\Theta(\log n)$.*

Proof. It suffices to prove that the $\mathcal{E}$ of the optimal algorithms in complete graph is bounded by $\Theta(\log n)$. In the proof of Lemma 14, we develop the concept of a "switching point", where all nodes prior to this point reveal valid information and all nodes afterwards perform Majority Algorithm. Denote m as a random variable of the switching point's position. We prove that at least one of the following happens: (a) $m \geq (\log_{p/(1-p)} n)/2$; or (b) $\mathcal{E}$ is greater than $\sqrt{n}/2$.

If $m < (\log_{p/(1-p)} n)/2$, then from Equation (2), we know that the majority of revealed signals are wrong with probability at least $(1 + (p/(1-p))^m)^{-1} > n^{-0.5}/2$. So the expected number of wrong nodes is lower bounded by $(n - m)/\sqrt{n}$ which is asymptotically greater than $\sqrt{n}/2$. Therefore, for the optimal output algorithms in complete graph, $\mathcal{E}$ is at least

$$\min\left((1-p)\log_{p/(1-p)} n \Big/ 2, \sqrt{n}\Big/2\right) = \Omega(\log n).$$ ◀

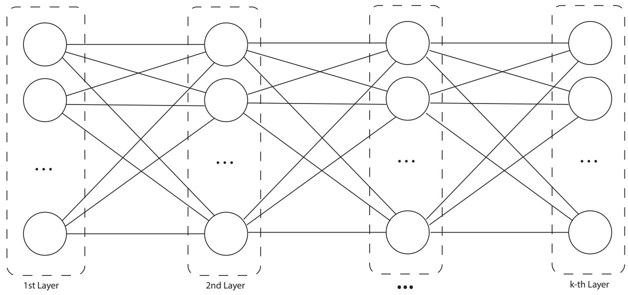

Figure 3 A k-layer graph.

6 Optimal K-Layer Topology

In section 3, we show that the optimal $\mathcal{E}$ for random graphs is $\Theta(\log n)$, which is asymptotically the same as the general lower bound in Theorem 17. Yet the question remains what is the actual optimal topology for the Majority Algorithm.

In this section, we propose a family of layer graphs and search for the optimal topology among this family. We claim, without proof, that the optimal topology of layer graphs is actually the optimal topology for Majority Algorithm.

To find the optimal topology, it helps to first understand the hidden insights behind small overall $\mathcal{E}$. In the optimal algorithms for complete graphs, nodes first judge the strength of the current cascade, and then decide whether to follow the cascade or reveal their own signals to strengthen the cascade. Such a *think-before-acting* way of decision-making guarantees the correctness probability of any established cascade, and thus results in good overall performance. We hope to know whether such *think-before-acting* could make it possible for Majority Algorithm to achieve optimality simply by adjusting the topology. This inspires us the following definition of layer graphs.

▶ **Definition 18** (Definition of layer graphs). A graph is said to have k layers if it can be separated into k disjoint groups, $S_1, \cdots, S_k$, where any node in group S_i is connected to and only to all nodes in S_{i-1}. See Figure 3 for an example.

▶ Algorithm 19. Given a k-layer graph G, we consider how nodes perform in G. First of all, similar to the optimal algorithms in complete graph, we have $|S_1|$ many nodes revealing **valid** information at the very front. If there exists a cascade in S_1 (the number of one choices outmatches another by at least two), then all later nodes follow this cascade. Otherwise, nodes in S_2 reveal their private signals. This process continues until a cascade happens in some layer. In this sense, layer graphs do contains the think-before-acting way of decision-making.

In the following sections, we find the optimal topology among layer graphs, and show that the expected number of wrong nodes $\mathcal{E}$ for such optimal topology is also $\Theta(\log n)$. This optimal $\mathcal{E}$ will be compared to previous bound and results, from which we will be able to glimpse the limit of Majority Algorithm. Throughout this section, if not otherwise mentioned, we will assume that the output algorithm is Majority Algorithm. We denote the expected number of wrong nodes on a k-layer topology $(S_1, \cdots, S_n)$ as $\mathcal{E}(|S_1|, \cdots, |S_n|)$.

6.1 Optimal topology for layer graphs

Remark 19 provides an intuitive way to calculate the $\mathcal{E}$ of any layer graph. Given i independent signals, we denote $p_w(i)$ as the probability that these signals generate a wrong cascade, and $p_n(i)$ as the probability that these signals does not generate cascade in either side. A recursion regarding $\mathcal{E}$ of any layer graph can be shown to be:

$$\mathcal{E}(a_1, \cdots, a_k) = (1-p) \cdot a_1 + p_w(a_1) \cdot (n - a_1) + p_n(a_1) \cdot \mathcal{E}(a_2, \cdots, a_k), \tag{3}$$

where $(1-p)a_1$ is the expected number of wrong nodes among the first layer, $n - a_1$ is the expected number of wrong nodes among later layers under wrong cascade, and $\mathcal{E}(a_2, \cdots, a_k)$ is the expected number of wrong nodes of later layers under no cascade. By extending the recursive term, we can simplify Equation (3) into

$$\mathcal{E}(a_1, \cdots, a_k) = \sum_{i=1}^{k} \Big(\prod_{j=1}^{i-1} p_n(a_j) \cdot \Big((1-p) \cdot a_i + p_w(a_i) \cdot (n - \sum_{j=1}^{i-1} a_j) \Big) \Big). \tag{4}$$

▶ Algorithm 20. Our goal is to estimate $\mathcal{E}$ of the optimal layer graph to the $\Theta(\log n)$ level. Any approximation of $\mathcal{E} + o(\log n)$ would be satisfying. This relaxation releases us from getting an exact optimum and allows us to make proper adjustments that greatly reduce the difficulty of the calculation. For example, in the derivation of Equation (3) and (4), we assume without loss of generality that each layer has an even number of nodes. This will result in $O(1)$ changes in the optimal parameters, which is tolerable.

To find a set of parameters $\{k, a_1, \cdots, a_k\}$ such that $\mathcal{E}(a_1, \cdots, a_k)$ is minimized, we need the following basic steps:

- We first show that in Equation (3), the contribution of $p_n(a_1)\mathcal{E}(a_2, \cdots, a_k)$ is limited and may be discarded without much change to the optimal parameters. Therefore, it suffices to consider the optimization of a_1 in the equation

$$\arg\min_{a_1} f(a_1) = \arg\min_{a_1} \Big((1-p) \cdot a_1 + p_w(a_1) \cdot (n - a_1) \Big). \tag{5}$$

- We then solve the equation $f(x+1) - f(x) = 0$, which has a unique solution. It can be shown that $f(a_1)$ first decreases then increases with respect to a_1. Thus the solution for $f(x+1) - f(x) = 0$ offers an approximation to the optimal a_1 with only $O(1)$ error.
- After solving the optimal size of the first layer, we can apply this method recursively to calculate the optimal size of all layers.

The proof for the above three results are complex and brute-force. First, we present a lemma that addresses result 2. It is worth pointing out that Equation 5 is the expected loss when we have only two layers, with the first layer of size a_1 and the second layer of size $n - a_1$. Therefore, result 2 is equivalent to finding an optimal topology among two layer graphs. For convenience, we denote $s = 1/(4p(1-p))$.

▶ **Lemma 21.** *For the optimal topology among two layer graphs, its first layer has size*

$$\log_s n - \log_s(\log_s n)/2 + O(1).$$

▶ **Theorem 22.** *The optimal layer topology has $k = n/\log_s n + o(n/\log_s n)$ many layers. The first layer has $a_1 = \log_s n$ many nodes. The size of layer i may be written as a recursion of the size of layer $i-1$,*

$$a_i \sim \log_s(s^{a_{i-1}} - a_{i-1}). \tag{6}$$

In other words, the optimal topology satisfies the following structural properties:

- *The sizes of layers gradually decrease, and the number of layers with size $\log_s n - i$ is $(s-1)n/(s^{i+1}(\log_s n - i))$.*

- *The first* $(s-1)n/(s\log_s n)$ *layers have size* $\log_s n$*.*
- *The following* $(s-1)n/(s^2(\log_s n-1))$ *layers have size* $\log_s n - 1$*, and so on.*

We believe that the layer topology provided in Theorem 22 is actually the optimal topology for Majority Algorithm. However, we have not yet come up with any rigorous proof to verify our conjecture. We will leave this as an open problem for future work.

▶ **Conjecture 23.** *The layer topology provided in Theorem* 22 *is the optimal topology for Majority Algorithm.*

6.2 Experiments on the optimal parameters

For layer graphs, Equation (6) can be used to calculate the optimal parameters with high precision. Thus the results in Theorem 22 is a very tight approximation, which works flawlessly if we only seek to analyze the complexity of $\mathcal{E}$. However, in real life, users might wish to achieve the exact optimal parameters. In this case, the constant error in our equations can not be neglected. Here, we introduce an algorithm that searches for the exact optimal topology in average $O(1)$ run time.

Recall that layer graphs satisfy the following properties. For a k-layer structure $(a_1, \cdots, a_k)$, if the first layer cascades, the rest of the nodes follow this cascaded result. Otherwise, the rest of the nodes become equivalent to a $(k-1)$-layer structure, with each layer's size being $(a_2, \cdots, a_k)$ (or $(a_2+1, \cdots, a_k)$ if a_1 is odd). Therefore, for a fixed a_1, the optimal k and $a_2, \cdots, a_k$ should be chosen such that,

- If a_1 is even,

$$(k-1, a_2, \cdots, a_k) = \arg\max_{k', a'_1, \cdots, a'_k} \mathcal{E}\Big(a'_1, \cdots, a'_k \Big| \sum_{i=1}^{k'} a'_i = n - a_1\Big). \tag{7}$$

- If a_1 is odd,

$$(k-1, a_2+1, \cdots, a_k) = \arg\max_{k', a'_1, \cdots, a'_k} \mathcal{E}\Big(a'_1, \cdots, a'_k \Big| \sum_{i=1}^{k'} a'_i = n - a_1 + 1\Big). \tag{8}$$

This implies that given the optimal layer topologies for all $n' < n$, calculating the optimal layer topology for n should be easy. We can simplify the problem into an optimization over a_1, instead of the optimization over many parameters. A further analysis shows that the optimal layer topology for n nodes and $n+1$ nodes cannot differ by too much. More specifically, denote the optimizing parameter for n nodes as $(k_n, a_1^n, \cdots, a_{k_n}^n)$, then $|a_1^n - a_1^{n+1}| \le 1$. Using this, together with the recursive idea in Equation 7 and 8, we can design an algorithm that runs in time $O(n)$, and find the optimal layer topology for all $n' \le n$. The amortized running time of this algorithm is only $O(1)$.

7 Conclusion

In this paper, we discussed information cascades on various network topologies. We provide a non-constructive optimal algorithm scheme for general graphs, solve the scheme for complete graph and achieve a general lower bound for our model. We also studied Majority Algorithm in random graphs and layer graphs, the minimal $\mathcal{E}$ of which was shown to be asymptotically the same with our general lower bound. From the experiment results, a gap between the general lower bound and layer graphs can be observed. We believe this to be a result of the difference in the model setting, i.e. Majority Algorithm is weaker than optimal general algorithms.

Future work in this area may include the study of the following scenarios.

- The nodes' order of decision-making is no longer fixed and given, but instead randomly sampled from all permutations.
- The topology is fixed and we are only able to add or remove a fixed portion of the edges. The goal is to minimize $\mathcal{E}$ under this constraint.
- *Plant nodes* in the network. These nodes could sacrifice themselves to reveal their true private signal. How should a topology designer control and position these plant nodes in the topology?

References

1 Daron Acemoglu, Munther A. Dahleh, Ilan Lobel, and Asuman Ozdaglar. Bayesian Learning in Social Networks. *Review of Economic Studies*, 78(4):1201–1236, 2011. URL: https://ideas.repec.org/a/oup/restud/v78y2011i4p1201-1236.html.

2 Daron Acemoglu and Asuman Ozdaglar. Opinion dynamics and learning in social networks. *Dynamic Games and Applications*, 1(1):3–49, 2010.

3 Lisa R Anderson and Charles A Holt. Classroom games: Information cascades. *The Journal of Economic Perspectives*, 10(4):187–193, 1996.

4 Lisa R Anderson and Charles A Holt. Information cascades in the laboratory. *The American economic review*, pages 847–862, 1997.

5 Abhijit Banerjee and Drew Fudenberg. Word-of-mouth learning. *Games and Economic Behavior*, 46(1):1–22, January 2004.

6 Abhijit V Banerjee. A simple model of herd behavior. *The Quarterly Journal of Economics*, pages 797–817, 1992.

7 Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of political Economy*, pages 992–1026, 1992.

8 David Bindel, Jon Kleinberg, and Sigal Oren. How bad is forming your own opinion. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 57–66, 2011.

9 Boğaçhan Çelena and Shachar Kariv. Observational learning with imperfect information. *Games and Economic Behavior*, 47:72–86, 2004.

10 Flavio Chierichetti, Jon Kleinberg, and Alessandro Panconesi. How to schedule a cascade in an arbitrary graph. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC'12, pages 355–368, New York, NY, USA, 2012. ACM.

11 Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69:118–121, 1974.

12 David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

13 Benjamin Golub and Matthew O. Jackson. Naive learning in social networks: convergence, influence, and the wisdom of crowds, 2010.

14 Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, May 1978.

15 MohammadTaghi Hajiaghayi, Hamid Mahini, and David Malec. The polarizing effect of network influences. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC'14, pages 131–148, New York, NY, USA, 2014. ACM.

16 MohammadTaghi Hajiaghayi, Hamid Mahini, and Anshul Sawant. Scheduling a cascade with opposing influences. In *Algorithmic Game Theory: 6th International Symposium, SAGT 2013, Aachen, Germany, October 21-23, 2013. Proceedings*, pages 195–206, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

17 Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *STACS 99: 16th Annual Symposium on Theoretical Aspects of Computer Science Trier, Germany, March 4–6, 1999 Proceedings*, pages 404–413, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

18 David Krackhardt. A plunge into networks. *Science*, 326(5949):47–48, 2009. doi:10.1126/science.1167367.

19 Lones Smith and Peter Sørensen. Pathological outcomes of observational learning. *ECONOMETRICA*, 68:371–398, 1999.

20 Lones Smith and Peter Sørensen. Rational social learning with random sampling, 2008. URL: http://lonessmith.com/sites/default/files/rational.pdf.

21 Arthur W Brian. Competing technologies, increasing returns, and lock-in by historical events. *The Economic Journal*, 99(394):116–131, 1989.

22 Jun Wan, Yu Xia, Liang Li, and Thomas Moscibroda. Information cascades on arbitrary topologies, 2016. URL: http://arxiv.org/abs/1604.07166.

23 Ivo Welch. Sequential sales, learning, and cascades. *The Journal of finance*, 47(2):695–732, 1992.