Self-sacrificial Behavior in Information Cascades

7.5 ECTS Artficial Intelligence Bachelor Thesis

Universiteit Utrecht

Olav Pelzer	**Dr. Erik Stei**	**Dr. Rachel Boddy**
Author	*Supervisor & Reviewer*	*Second Reviewer*

July 15th, 2020

Abstract

In sequential decision making, information cascades occur when agents base their decisions on the actions of other agents, ignoring their own observations. This can cause rational agents to perform sub-optimally at group level. The problem seems to stem from a disbalance in on one hand providing new information for the group (independent behavior), and other the other hand making use of this publicly available information (dependent behavior). Carried out in extremes, the first type of behavior would ignore potentially useful information, while the latter would lead to possibly incorrect herding-behavior.

In this thesis, I investigate how in-group agents should ideally balance out the use of publicly accessible information, so that they can maximize their result as a group. A solution is provided for a specific example of sequential decision making, known as the *urn problem.* The main finding is that a perfect balance indeed exists, and that the information provided by early actors contributes the most to the optimal group result.
Keywords: *information cascade, sequential decision making, self-sacrificial behavior, rational agent, group behaviour, incomplete information, game theory.*

Contents

1 Introduction **3**

2 Information Cascades **6**
- 2.1 Urn Problem - Informal Analysis 6
- 2.2 Urn Problem - Bayesian Analysis 8
- 2.3 Definitions and Characteristics 10
  - 2.3.1 Cascade Likelihoods 10
  - 2.3.2 Cascade Fragility 12

3 Group Results of Information Cascades **14**
- 3.1 Two Extremes 14
  - 3.1.1 Maximal Independence: *Altruism* 15
  - 3.1.2 Minimal Independence: *Selfishness* 16
- 3.2 Finding a Balance 18
  - 3.2.1 A Recursive Approach 25

4 Discussion & Conclusion **27**

A Recurrence Equation **30**

B Python Code **31**

Chapter 1

Introduction

As technology develops quickly in the fast-paced *information society* we live in, more and more information is gathered and stored both *by* and *for* individuals. As a result, the extent to which we are influenced by others in our decision making increases.

Intuitively, when it comes to sequential decision making, acting after your predecessors should only increase your ability to make the right decisions because you have more information available to base your decisions on. As it turns out, making use of the total information in a group often leads to better results than the results of individuals in that group alone.

For example, Condorcet's Jury Theorem illustrates the likelihood of a group to arrive at a correct decision [1]. More precisely: it illustrates that if individual voters are correct most of the time ($> 50\%$), adding more voters to the group increases the chance of arriving at a majority-correct outcome. But even if individuals are not correct most of the time, we can see a similar increase in performance in the following example. Imagine a simple game where people have to guess the amount of candy in a jar. Assuming the people guess independently and that people are just as likely to overestimate as to underestimate, the average error will eventually balance out and approach zero as the amount of guesses increases to infinity.

In these cases of sequential decision making, the increased correctness of the group is thanks to the *Law of Large Numbers*. However, the superior result was calculated in retrospect and not actually used by anyone in the group itself. This seemingly superior outcome is in this case only accessible by outsiders looking down on the group, who are reasoning from a bird's-eye view.

As Baltag et al. correctly state, the superior combined knowledge of a group that doesn't communicate (like in the above-mentioned examples) is actually a *virtual* type of knowledge [2]. What is meant here by virtual, is that this knowledge is a purely statistical attribute and is not accessible by anyone in the group. But could this virtual knowledge be actualised if the members of the group communicated about their actions? Paradoxically, communicating your actions to your successors in order to transform the virtual group knowledge to actual knowledge may lead to inferior group performances.

Faulty group behavior can sometimes be attributed to psycho-social phenomena that lie at the basis of human group dynamics. Examples being *conformity*, where people deliberately match their behavior to that of the group, or the *Abilene-paradox*, where people make incorrect assumptions about the preferences of others [3].

However, removing irrational human behavior may not always solve the emergence of poor group results. Agents that behave perfectly rational may also produce a surprisingly unsatisfactory group outcome. In the examples mentioned at the beginning of this chapter, the decision makers did not communicate with each other. This independence is one of the concepts that caused the superior group result in the first place. But if the agents were to make use of the actions of others, the independence would be lost, and the superior virtual group knowledge would no longer exist since the Law of Large numbers does not necessarily hold when the samples are not independent [4]. So by trying to pursue the better group result, the virtual knowledge they are trying to actualize will be lost in the process of trying to reach it. It seems that *independence* is key to the formation of the superior group results that only exist *afterwards*, while *dependence* is key to making use of this extra knowledge *during* the problem.

This problematic trade-off between dependence and independence lies at the root of so-called *information cascades*. An information cascade is a situation where it is optimal for an agent, having observed the actions of those that acted earlier, to base his decision on the behavior of his predecessors without regard to his own, private, information [5]. Though not problematic *per se,* in the formation of an information cascade, independent decision making is lost quickly, causing all the decisions to be based only on the information that the first few people (who acted independently) provided. If the information that these initial actors provided was incorrect, all successors would base their decisions on erroneous signals, causing everyone in the group to be incorrect.

Most studies done on information cascades have come from scientific fields that concern themselves with human decision making, examples being Economics and Sociology. However, since information cascades can also occur in computer-like or rational settings, research from other fields such as Computer Science and Network Theory has also investigated them. Multiple efforts have been made to positively influence the outcome of information cascades. For example, T. N. Le et al. analyzed the value of noise in observations made by agents [6], while others researched the effects of signal accuracy on the probability of an incorrect cascade [7, 8].

Despite these efforts and their progress, authors claim that information cascades remain a fundamental challenge to group-rationality and therefore suggest further research [2, 9]. One of the suggestions Baltag et al. make to prevent cascades from happening is to have some in-group agents behave 'irrationally' by disregarding all available evidence except for their own to base decisions on [2]. This way, they would lower their own chances of being correct, but their sacrifice would increase the likelihood of being correct for the others that have yet to act.

In this thesis, I build upon this suggestion. It seems that in most publications surrounding information cascades, the individuals that are being studied had nothing to gain from a superior group outcome; they were merely interested in their own success, because that was where their reward was based upon. This is why Baltag et al. would describe this type of behavior to prevent a cascade as irrational. However, when agents get rewarded based on their performance as a group, this altruistic behavior of self-sacrifice would no longer be intrinsically irrational.

This thesis dives deeper in the scenario where a few can sacrifice for 'the greater good'. There seems to be an optimal balance between on one hand making use of the actions of others, and on the other hand ignoring those actions. If carried out in extremes, the first could potentially lead to catastrophic incorrect herding behavior, while the latter would ignore potentially useful information. The main goal of this thesis is to find out just where the above-mentioned balance lays, and to provide an answer to the question of how a group should optimally take advantage of the collectively available information.

This thesis will be structured as follows:
In chapter 2 I will first exemplify information cascades by providing a textbook example (the so called *urn problem*) together with a formal Bayesian analysis. Moreover, this chapter will provide definitions used to analyze cascades, as well as generalize key concepts and characteristics of information cascades. The main focus will be on individual decision making.
In chapter 3 the attention is shifted to group results. I first examine both extremes by looking at an *all or nothing* approach: one where all agents ignore others so that no cascades emerge, the other where no agent ignores others so that cascades will almost surely appear. Then, I compare these two extremes to a more balanced in-between approach, where the goal is to maximize the amount of correct answers in the group. I show that such an approach indeed exists and leads to very remarkable group outcomes.
In chapter 4 I discuss my main findings and I conclude that an optimal balance exists when trying to maximize the group outcome. These findings confirm that self-sacrificial behavior is not irrational when agents have an incentive to maximize the result of the group instead of maximizing their own likelihood of being correct. In this final chapter I also suggest directions for further research.

Chapter 2

Information Cascades

This chapter starts off with a section that covers an example problem of an information cascade known as the *urn problem*. This problem was created for illustrative purposes and thus gives a very clear insight into the formation mechanism and effects of information cascades [2]. The urn problem described in this section is based on - and very similar to - the ones described in the works of: [ref. 2, 10, 9, 11].

2.1 Urn Problem - Informal Analysis

Consider two urns:

- Urn R, which contains two red balls and one green ball;
- Urn G, which contains two green balls and one red ball.

From the outside, both urns are indistinguishable. An experimenter randomly selects one of these urns and places it in a room. One hundred individuals participating in the experiment have to guess whether the experimenter placed R or G in the room. To do so, they enter the room one at the time and draw one ball from the urn in the room, observe its color, and place it back. The individual then guesses which urn he or she thinks is more likely to be in the room and writes down this guess so that all upcoming individuals can see what guesses were previously made by whom. When everyone has made their guess, the experiment is over and the individuals that voted correctly receive a reward.
We assume that all participants behave rationally and want to maximize their chances of being rewarded. As a tie-breaking rule, whenever an individual has no preference for either R or G, he or she will break the tie by voting for his or her own observation. We also assume that the above-mentioned urn problem and all its rules is common knowledge for all participants.
In this urn problem, the whole issue of partial information and information cascades could easily be solved if the participants also communicated their observations along with their vote. However, despite being an easy option in this example, the point of this setup is to simulate situations where individuals can only observe the actions of others, which is often the case in real life scenarios.
To analyze how this urn problem could play out, and how a cascade could start, I will go over each individual in order to see what vote they would make with regards to the evidence they have at their disposal.

The first individual
When the first participant draws a red ball from the urn, he is justified in believing that the odds of the urn containing two red balls and one green ball is higher than the odds of the urn containing two green balls and one red ball. In this case he will thus pursue his best chances at getting it right, and chooses urn R. The same reasoning would apply were he to draw a green ball, resulting him in choosing urn G. We can see that whatever ball the first person observes, his guess will be equal to his observation. Because upcoming individuals can reason perfectly about the decision making of the first individual, they will be able to infer which ball he observed simply by looking at the vote he made. I will refer to this type of information that is being relayed for upcoming participants as *perfect information*. So, when given this perfect information, nothing will be hidden to those that receive the information. For the sake of analysis, for now we will now assume the first person observed a green ball and voted for G.

The second individual
By looking at the guess made by the first participant, the second participant already has an extra piece of evidence at her disposal besides her own observation. There are two possible observations for the second individual:

- If the second individual draws the same colour ball (a green one) as the first person, she should obviously pick this colour as well since it only provides more evidence for the G urn.

- Suppose she draws a different colour ball (a red one). In this case there is an equal amount of evidence for both options[1]. Consequently, she will apply the tie-breaking rule as described above. Namely, since the total amount of evidence is inconclusive, she will prefer her own signal and vote R.

We can see that the first two individuals both convey perfect information. To continue with the example, let's assume the second participant also draws a green ball and consequently votes for G.

The third individual
Before the third participant draws a ball from the urn, she sees that both individuals prior to her have voted for option G, and since she also knows that their votes relay perfect information, she concludes that both drew a green ball.
Let's again consider both possibilities for the third participant:

- Assume she draws a green ball. This observation obviously only adds to the evidence pointing in the G urn direction. Obviously in this case, she will also vote for G.

- Assume she draws a red ball. Things start to get interesting here. She would reason as follows: two pieces of evidence point towards G, but only one (her own) piece of evidence points towards R. Thus, she concludes that her chances of being correct are higher when she votes for G.

[1] In the next section I show that this intuitive heuristic, where evidence is summed up and cancelled out, is in fact perfectly rational for this specific urn problem.

As we can see, an information cascade has begun: the third participant would, despite her own evidence pointing in other directions, always vote G. In fact, in this case she would not even have to take her own evidence into consideration, since she knows that it could not possibly alter her decision in any way. By voting while being in this cascade, she does not relay any new information to the next individual. In other words, the next individual can not deduce what the third participant observed.

The fourth individual and onward
The fourth individual would reason that the person before him must have acted purely based on the information that was available to her at the moment of arriving in the room. He knows that she voted while being in an information cascade and he reasons that her vote did not relay any new information. Hence, he is in the same epistemic position as the person before him and will act the same: vote G.
The fifth and all other upcoming participants will apply the same reasoning.
As it turns out, all individuals after the first two are in the same epistemic position: *id est* they have the same information available to them. Consequently, in this setup the cascade will continue forever and all upcoming participants will pick their best option, which is joining the cascade and voting for option G.

2.2 Urn Problem - Bayesian Analysis

So far I have shown that the participants based their decisions on the evidence by simply counting it and checking which side is supported by more evidence. This technique of simply summing up all the evidence (despite being a heuristic tool that humans might use in real life situations[2]) is in this case perfectly rational as it leads to the same results as calculating the likelihoods using Bayesian probability theory. The reason for this is because of the symmetry in signal accuracy in our example [6]. In other words, observing a red ball provides just as much evidence in favor of R, as observing a green ball would for G. This allows participants to cancel out opposite observations. However, it should be stressed that this is not generally the case. The composition of the urns could be changed so that the symmetry is lost. In that case this counting evidence heuristic no longer leads to perfect results. In those cases, rational agents would rely on Bayesian probability theory to base their decisions on [12]. For that reason, I provide a Bayesian analysis in this section, which is again very similar to the work of [10].

The Urn Problem
Consider again the urn problem. Let's denote the group of n agents as $A = \{a_1, a_2, ..., a_n\}$. The conditional probabilities are calculated using Bayes Theorem:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where $P(B) \neq 0$.
For each agent, the following prior probabilities are known:

- Both the majority-green urn (G) and the majority-red urn (R) have an equal probability of being placed in the room:
  $P(G) = P(R) = \frac{1}{2}$

- Based on the composition of urn G and R, the probability of drawing a green ball (g) and a red ball (r) respectively, is equal to $\frac{2}{3}$:
  $P(g|G) = P(r|R) = \frac{2}{3}$

Let's assume the experiment plays out the same way:

First Agent
Assume a_1 observes a green ball (receives private signal g). He will calculate the conditional probability of the urn being G:

$$
\begin{aligned}
P(G|g) = \frac{P(g|G)P(G)}{P(g)} &= \frac{P(g|G)P(G)}{P(g|G)P(G) + P(g|R)P(R)} \\
&= \frac{\frac{2}{3} \cdot \frac{1}{2}}{\frac{2}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{2}} \\
&= \frac{2}{3}
\end{aligned}
$$

After receiving private signal g, the conditional probability $P(G|g) = \frac{2}{3}$. Because $P(G|g) > P(R|g)$, logically a_1 votes G.

Second Agent
For a_2, after receiving signal g, he adds this evidence to the evidence he received from a_1 and calculates as follows:

$$
\begin{aligned}
P(G|g,g) = \frac{P(g,g|G)P(G)}{P(g,g)} &= \frac{P(g,g|G)P(G)}{P(g,g|G)P(G) + P(g,g|R)P(R)} \\
&= \frac{\frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{2}}{\frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{2} + \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{1}{2}} \\
&= \frac{4}{5}
\end{aligned}
$$

$P(G|g,g) > P(R|g,g)$, consequently a_2 votes for G.

Third agent
The third agents reasons that if a_2 received signal r, a_2 would have calculated:

$$P(G|g,r) = \frac{P(g,r|G)P(G)}{P(g,r)} = \frac{1}{2}$$

and therefore would have voted R as a result of the tie-breaking rule. Thus, a_3 concludes based on the first two votes that a_1 and a_2 received signal g. After now receiving signal r himself, the total evidence sums up to (g, g, r) and a_3 calculates:

$$\begin{aligned} P(G|g,g,r) &= \frac{P(g,g,r|G)P(G)}{P(g,g,r)} = \frac{P(g,g,r|G)P(G)}{P(g,g,r|G)P(G) + P(g,g,r|R)P(R)} \\ &= \frac{\frac{2}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{1}{2}}{\frac{2}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}\cdot\frac{1}{2}+\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{2}{3}\cdot\frac{1}{2}} \\ &= \frac{2}{3} \end{aligned}$$

The effect of the information cascade becomes clear: a_3 will vote for G despite his own signal being r, since $P(G|g, g, r) > P(R|g, g, r)$. The evidence gathered by observing the agents that acted before him outweighs his own evidence.

Fourth and upcoming agents
Without any interventions, the same reasoning that was carried out by a_3 will be applied by agents $a_4, a_5, ..., a_n$. All of those agents will have the signals of a_1 and a_2 available to them (g, g), with the addition of their own signal $(g \vee r)$. The fact that a_3 voted G does not give the successors any new information, as they can reason that a_3 would have voted G in both of the cases (g, g, r) and (g, g, g). Therefore, from this point on, the amount of information does not grow with each vote.
As a result, all agents after a_3 will vote for G as well.

2.3 Definitions and Characteristics

In this section I will elaborate on the following characteristics of information cascades:

1. The likelihood of information cascades;
2. The fragility of information cascades.

2.3.1 Cascade Likelihoods

In the provided urn example, we have not yet specified which urn was actually in the room. Therefore, we can not yet know whether agents voted correctly or incorrectly. The following terminology will be used to distinguish between the value of certain signals and cascades (which are dependent on the state of the world/the urn in the room):

- *High Signals*
  Signals with a probability higher than $\frac{1}{2}$ will be referred to as *high signals.* In our case $(g|G)$ and $(r|R)$ are high signals, each with a probability of $\frac{2}{3}$;

- *Low Signals*
  Signals with a probability lower than $\frac{1}{2}$ will be referred to as *low signals*. In our case $(r|G)$ and $(g|R)$ are low signals;

- *Incorrect/Correct Cascades*
  Cascades leading to (in)correct results will be named as (in)correct cascades.

To analyze how likely it is for a cascade to appear, let's calculate the likelihoods for all four possibilities after two signals (one cycle), and look at their corresponding outcome (indicated with an arrow):

- *<high signal, high signal>* $\rightarrow$ *correct cascade*
  The chance of receiving two high signals in a row is in our case equal to $P(g, g|G)$ (or $P(r, r|R)$), which is equal to: $\frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}$;

- *<high signal, low signal>* $\rightarrow$ *no cascade: baseline*
  The chance of receiving both a high and a low signal is equal to: $\frac{1}{3} \cdot \frac{2}{3} = \frac{2}{9}$;

- *<low signal, high signal>* $\rightarrow$ *no cascade: baseline*
  The chance of receiving both a low and a high signal is equal to: $\frac{2}{3} \cdot \frac{1}{3} = \frac{2}{9}$;

- *<low signal, low signal>* $\rightarrow$ *incorrect cascade*
  The chance of receiving two consecutive low signals is equal to: $\frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$.

We can conclude that for our instance of the urn problem, the following is true after the first cycle of two agents:
- $P(cascade) = P(high, high) + P(low, low) = \frac{4}{9} + \frac{1}{9} = \frac{5}{9}$;
- $P(\neg cascade) = P(low, high) + P(high, low) = \frac{2}{9} + \frac{2}{9} = \frac{4}{9}$.
We know that for an even amount of votes n there will have been $n/2$ cycles of two votes. In order for a cascade to be present after $n/2$ cycles, a cascade only needs to have happened once in the entire sequence. On the other hand, a baseline situation (no cascade) is only present in the event that a cascade never occurred. The event *cascade* is the complement of $\neg cascade$, therefore they are mutually exclusive and collectively exhaustive (MECE). We can say that a cascade is not present after $n/2$ cycles if and only if the event $\neg cascade$ occurred in every cycle. Therefore we can write that the probability of a cascade after an even amount of votes n is equal to:

$$\begin{aligned} P(cascade) &= 1 - P(\neg cascade)^{n/2} \\ &= 1 - \left(\frac{4}{9}\right)^{n/2} \end{aligned}$$

If the amount of agents approaches to infinity, we can calculate:

$$\begin{aligned} P(cascade) &= \lim_{n \to \infty} \left(1 - \left(\frac{4}{9}\right)^{n/2}\right) \\ &= 1 - 0 \\ &= 1 \end{aligned}$$

Thus we can say that given enough agents, cascades occur almost certainly. The reason is that given enough votes, at some point the evidence difference will reach a value of 2.

2.3.2 Cascade Fragility

A cascade starts when a personal observation can no longer overthrow the prior evidence available to someone. In our case, this happens when the total evidence difference reaches a value of two. In figure 2.1, adapted from Easly & Kleinberg (2010) [10, fig 16.3 on p. 501], I portray the situation as described in the previous section where a cascade started after the first two G votes[2].

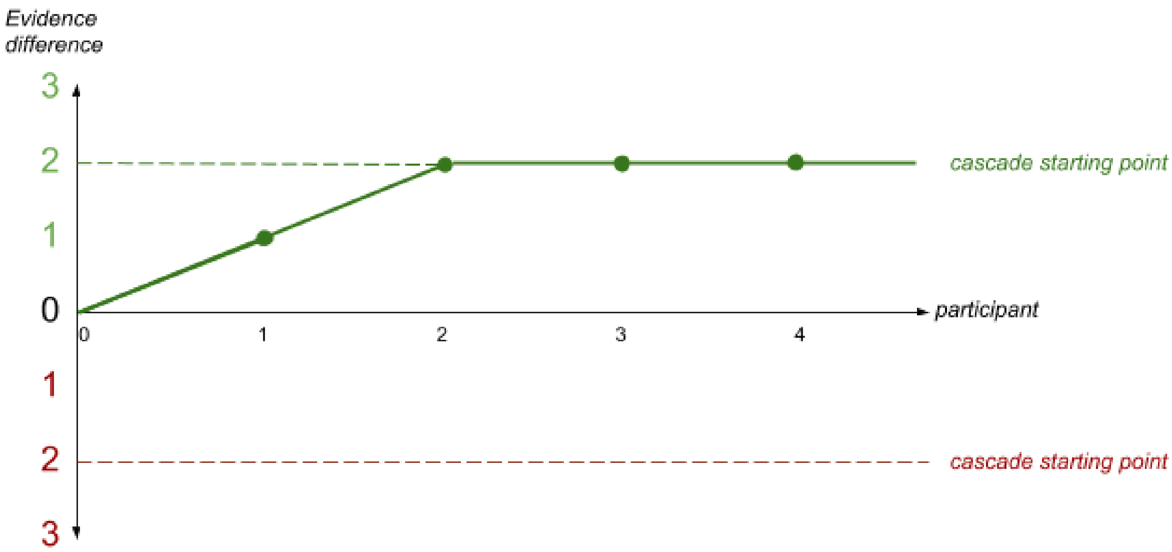

Figure 2.1: A cascade starts after the first two people vote G, making the evidence difference reach a value of 2. The green line will continue to follow the dotted line for all upcoming participants. The colour of the dot (and line reaching that dot) depicts the vote that the corresponding participant makes.

Cascades are fragile because they are based on very little information. As a result, they can easily be 'destroyed' by new information. To illustrate the fragility of cascades, imagine that a certain agent a_m with $m > 2$ decides to publicly announce that he will disregard all other evidence and vote for his own observation. In that case, a_m votes independently and thus conveys new information to a_{m+1}. The result is that upon receiving his own signal, a_{m+1} now has four pieces of evidence at his disposal, namely the evidence of: a_1, a_2, a_m and a_{m+1}.

Coming back to our previous example where a_1 and a_2 voted G, assume now that agent a_m publicly announces that he observed a red ball. Then, a_{m+1}, upon drawing a red ball himself, will calculate:

$$P(G|g, g, r, r) = \frac{P(g, g, r, r|G)P(G)}{P(g, g, r, r)} = \frac{1}{2}$$

[2] I would like to point out that in the original figure, the graph goes beyond the dotted line that indicates the cascade boundary. I deliberately chose not to make the solid line surpass the dotted line, since this would not accurately portray the epistemic position. As reasoned before, all votes made while being in a cascade do not convey any new information. The reason for this subtle distinction will become more obvious after studying fig. 2.2, where I illustrate the fragility of cascades.

Because we can cancel out all the evidence and since there is an equal amount of evidence pointing to G and R, we can immediately see that $P(G) = P(R) = \frac{1}{2}$. Now, having no preference, a_{m+1} will apply the tie-breaking rule and follow his own signal, which is r, and vote R accordingly. The following figure depicts this scenario for $m = 80$ so that a_{80} votes independently:

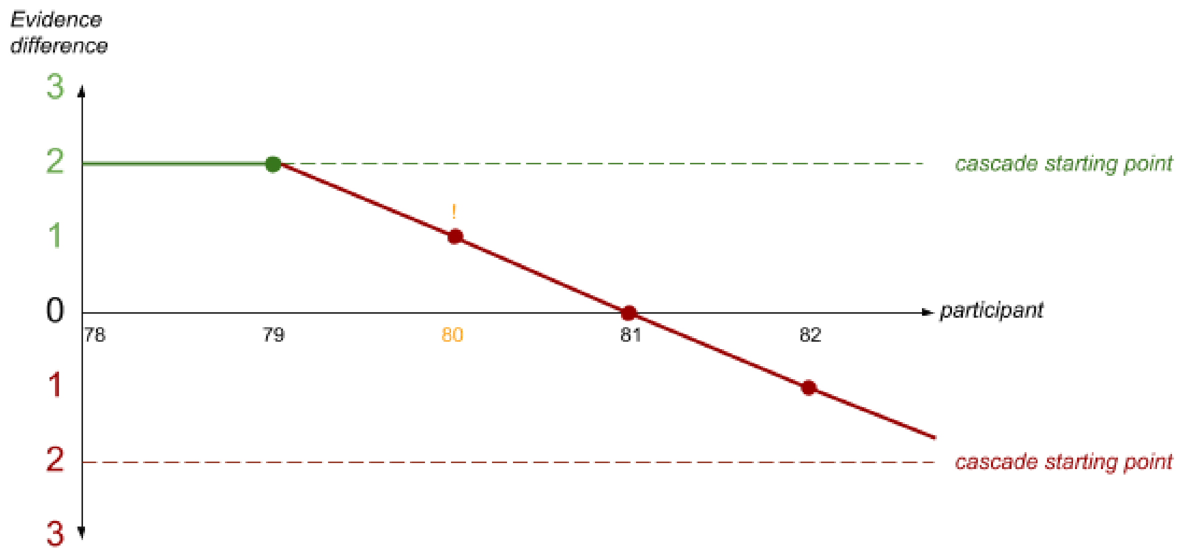

Figure 2.2: A possible continuation after participant 80 conveys new information (highlighted in orange together with an '!'). The cascade no longer exists. Participant 81 will vote according to his own observation. Assuming participants 82 and 83 also receive signal r, a new cascade emerges where every upcoming agent will vote R.

Instead of being destroyed by the new information, the cascade could also have been solidified. Agent a_{80} could have observed g. In that case, the amount of publicly accessible evidence would be (g, g, g) and the solid green line would go up, surpassing the dotted boundary line.
Once a cascade emerges, it remains the same informational state which also happens to be its minimum in order to survive. This is due to the fact that all participant that act on this cascade do not contribute to the total information upon which it is based. As a result, the cascade does not grow stronger as it progresses.

To summarize, we have seen that the following characteristics apply to information cascades in general:

- Cascades occur almost surely, given enough participants;
- Cascades can be incorrect, leading to wrong answers for everyone in the group;
- Cascades are fragile and can therefore easily be destroyed by new information at any point.

Chapter 3

Group Results of Information Cascades

So far, the focus has been on individual decision making for agents who want to maximize their own chance of being correct. If agents are rewarded based purely on their own performance, and if one assumes that rational agents want to maximize their expected pay-off, then this type of selfish behaviour seems perfectly rational.

However, what if agents get rewarded based on their performance as a group? In that case, maximizing your own chances of being correct is not equal to maximizing your expected pay-off. In those situations, the whole problem of information cascades becomes clear: group results may suffer if independence is lost too quickly.

This chapter will address the problem of information cascades directly by shifting the attention to the results of individual decision making on the group outcome. For different types of behavior, the expected group outcome is calculated. To illustrate this, I will again consider the same urn problem as mentioned in chapter 2.

In the first section, we take a look at two extremes. A comparison is made between complete independence (no cascades) and minimal independence. Minimal independence is what we have observed so far in our previous analyses: an agent will make use of available information as soon as possible.

In the second section, we find an optimal balance between the two. The goal is to investigate how many agents should sacrifice themselves in the beginning before other agents can rationally make use of this knowledge.

3.1 Two Extremes

The first extreme behavior I analyze will be referred to as *altruistic behavior.* This is the type of behavior where every agent acts independently and only makes use of his own evidence. Using the word *altruistic* in this context might seem counter-intuitive since the agents are ignoring everything except for their own observations. However, what is meant here, is altruistic in an *epistemological* way. We have seen in the previous chapter that agents who act only on the information of others (agents in a cascade) do not contribute to the publicly available evidence. In contrast, agents who voted only according to their own observations relayed information. Thus, in this case, by acting independently, useful information is provided for upcoming agents. Hence the term altruistic.

One could still argue that altruism, which applies to human behavior, is not the correct word when speaking of rational agents. True altruism is not defined as acting out of self-interest, but acting because you are concerned for others. But a rational agent that is rewarded for the performance of the group, would in this case not share his true information because he cares about others, but because he knows that the group will perform better and therefore his own chance of receiving a high rewarded grows. In that case, a better description of this behavior might be: *One for all, all for one* or *Do ut des*[1]. However, for now I will stick with the terminology of *altruism* in case of agents acting solely based on their own observations.

The other extreme, where agents make use of the available information as soon as possible, will be covered in the *Minimal Independence* subsection. In contrast to the behavior in the other subsection, this behavior could be addressed as *selfish behavior.* Selfish in the sense that these agents will not contribute to the available information unless they are forced to do so.

I will refer to agents that ignore all other evidence except their own as *sacrificial agents.* The following notation will be used to express the expected amount of correct answers (or expected correctness) in the urn problem for a group of n agents where the first s agents act as sacrificial agents:

$$C_s(n) \qquad \text{for } s \in \{0, 1, 2, .., n\}$$

Note that for the altruistic behavior, every agent acts independently, so in that case $s = n$. For the selfish behavior, as seen in chapter 2, the first two agents will always vote for their own observation. As a consequence, even when no agent willingly sacrifices himself, the first two agents will always provide true information. Since a_1 and a_2 relay information no matter whether they sacrifice or not, a similar result will appear when calculating $C_s(n)$ for $s = 0, 1, 2$. This should be kept in mind when calculating $C_s(n)$ for $s = 0$.

3.1.1 Maximal Independence: *Altruism*

Maximal independence would be the best option if one were to reason as an outsider *after* all agents have voted. This way, one would receive the maximum amount of information to base one's decision on. This type of behavior might therefore be optimal in certain observational learning environments where the goal is to find out the signal accuracy.

Expected Correctness
The calculation for the expected amount of correct answers is very straightforward. Since each individual agent has a $\frac{2}{3}$ chance of being correct, we expect that $\frac{2}{3}$ of all agents will be correct. We get:

$$C_n(n) = \frac{2}{3}n$$

[1]A Roman principle of reciprocity. The phrase literally translates to: *I give with the intention that you give.*

To compare the outcome of this strategy to other types, we will calculate the expected correctness for $n = 10, 100, 1000$. In this case, we get:

$$C_{10}(10) \approx 6.67$$
$$C_{100}(100) \approx 66.67$$
$$C_{1000}(1000) \approx 666.67$$

3.1.2 Minimal Independence: *Selfishness*

Agents that act selfishly will use all possible information available to them to increase their own odds of being correct. We might observe this type of behavior in competitive environments.

Expected Correctness
For $n = 0$:
Obviously, a game without any agents cannot have any agents voting correctly:

$$C_0(0) = 0$$

For $n = 2$:
We know from chapter 2 that the MECE-principle applies and that there are two possible events after two votes:

- $P(cascade) = \frac{5}{9}$:
  The chance of an incorrect cascade is $\frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}$ (0 agents vote correctly).
  The chance of a correct cascade is $\frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}$ (2 agents vote correctly).
  So, on average $\frac{4}{5}$ of all cascades are correct cascades.
  This means that $C_0(2|cascade) = \frac{4}{5} \cdot 2 + \frac{1}{5} \cdot 0 = \frac{8}{5}$
- $P(\neg cascade) = \frac{4}{9}$:
  No cascade occurs when a_1 and a_2 vote differently (in that case, exactly one agent is correct).
  This means that $C_0(2|\neg cascade) = 1$.

To calculate the expected value we will use the general formula below, where outcomes are weighted by their probability. Let X be a random variable with outcomes $x_1, x_2, ..x_n$ occurring with probabilities $p_1, p_2, ..p_n$. The expected value of X is then:

$$E(X) = \sum_{i=1}^{n} p_i \cdot x_i \tag{3.1}$$

For our instance, since we have covered all possible events ($\frac{4}{9} + \frac{5}{9} = 1$), the expected correctness can be calculated using:

$$C_s(n) = P(cascade) \cdot C_s(n|cascade) + P(\neg cascade) \cdot C_s(n|\neg cascade)$$

Which in our case is:

$$C_0(2) = \frac{5}{9} \cdot \frac{8}{5} + \frac{4}{9} \cdot 1$$
$$= 1\frac{1}{3}$$

For $n = 4$:
Using the same approach, we have that after the first two votes:

- A cascade appears: chance $= \frac{5}{9}$:
  On average, $\frac{4}{5}$ will be correct cascades in which all 4 agents vote correctly.
  This means that $C_0(4|cascade) = \frac{4}{5} \cdot 4 = 3\frac{1}{5}$
- No cascade appears: chance $= \frac{4}{9}$:
  Exactly one of the two agents is correct in this case. So the expected correctness is 1 + the expected correctness for the remaining two agents. One could calculate again what the expected amount of correct answers is for those remaining agents, but since we are at baseline situation, the problem comes down to our previous calculation: $C_0(2)$.
  This means that $C_0(4|\neg cascade) = 1 + C_0(2) = 1 + 1\frac{1}{3} = 2\frac{1}{3}$

To calculate $C_0(4)$ we use Equation 3.1:

$$\begin{aligned} C_0(4) &= \frac{5}{9} \cdot 3\frac{1}{5} + \frac{4}{9} \cdot 2\frac{1}{3} \\ &\approx 2.96 \end{aligned}$$

As we can see, calculating $C_0(n)$ generally comes down to:

- Calculating $C_0(n)$ for cascades starting after 2 votes;
- Calculating $C_0(n)$ for no cascades after 2 votes, which equals $1 + C_0(n-2)$.

This calculation translates to the recurrence relation below, which describes the relation between n and C_0 for all even n such that $n \geq 2$:

$$\begin{aligned} C_0(n) &= \frac{5}{9} \cdot \frac{4}{5}n + \frac{4}{9} \cdot (1 + C_0(n-2)) \\ &= \frac{4}{9}C_0(n-2) + \frac{4}{9}n + \frac{4}{9} \end{aligned}$$

To see how this strategy compares to the altruistic strategy above, we calculate[2] again for the following values of n:

$$\begin{aligned} C_0(10) &\approx 7.53 \\ C_0(100) &\approx 79.52 \\ C_0(1000) &\approx 799.52 \end{aligned}$$

Unsurprisingly, the expected group performance has increased. This is to be expected, because the agents in this group will make use of more true information. Despite the fact that incorrect cascades will occur in this setting, correct cascades have a higher likelihood of occurring.

If the amount of agents n grows, the likelihood that no cascade occurs shrinks. Proportionally, cascade situations will play a bigger role. Since the likelihood of a cascade being correct is equal to $\frac{4}{5}$, we can expect that the group performance approaches $\frac{4}{5}n$ or 80% correctness for any arbitrary large n. This is also easy to see from the recurrence equation as presented in Appendix A.

[2] The calculation can either be done by solving the recurrence relation (see Appendix A) or by defining a recursive function in Python (see Appendix B).

3.2 Finding a Balance

As mentioned at the beginning of this chapter, even when 0 agents intentionally sacrifice themselves, there is always of minimum of 2 agents that do relay true information. Therefore, we start our analysis for $s = 3$, which actually means that only one agents 'truly' sacrifices himself. Our goal is to calculate:

$$C_3(n)$$

To do so, let's take a look at all possible events with their respective results. I will denote receiving a high/low signal as h and l respectively. For the first s agents, we can make 2^s different combinations. In this case, $2^3 = 8$ combinations:

$$(hhh, hhl, hlh, hll, lhh, lhl, llh, lll)$$

Out of the 2^s possible combinations, only $s + 1$ vary in evidence difference. This is the case because we only have to count the amount of $h's$ and $l's$, as the order in which they occur does (for now) not matter. For a certain value of s it is easy to see that there are indeed $s + 1$ of such combinations that vary in evidence difference: the amount of high (or low) signals in a combination can be equal to $\{0, 1, 2.., s\}$. This set contains $s + 1$ elements.

For $s = 3$, the four combinations that vary in evidence difference are shown below together with how often they occur:

- 1x - 3 h's and 0 l's: $\{hhh\}$
- 3x - 2 h's and 1 l's: $\{hhl, hlh, lhh\}$
- 3x - 1 h's and 2 l's: $\{llh, lhl, hll\}$
- 1x - 0 h's and 3 l's: $\{lll\}$

For our calculations, it is useful to have a short notation for these types of combinations. I will use the following notation: combinations that have n high signals and m low signals (in any order) will be denoted as nHmL.

For higher values of s, checking all combinations and counting how often each different combination occurs becomes very cumbersome. For a certain value of s, the question of how many times a combination with n amount of high signals occurs basically comes down to the question of how many ways can we arrange n high signals amongst a total of s signals. If we denote the amount of high signals as h we can calculate the amount of possible arrangements for a certain combination by taking the binomial coefficient:

$$\binom{s}{h} = \frac{s!}{h!(s-h)!}$$

For example, for $s = 3$: in how many different ways can we arrange 2 high signals? This is easy to see: since there is only one low signal in this case, the low signal can be placed either at the first, second or third place. Thus, the answer is 3. Using the binomial coefficient we get:

$$\binom{3}{2} = \frac{3!}{2!(1)!} = \frac{6}{2} = 3$$

The calculation of the binomial coefficient for increasing values for s starting at 0, and increasing values for h reaching from 0 to and including s, is basically a construction of Pascal's triangle:

$s = 0$	1						
$s = 1$	1	1					
$s = 2$	1	2	1				
$s = 3$	1	3	3	1			
$s = 4$	1	4	6	4	1		
$s = 5$	1	5	10	10	5	1	
$s = 6$	1	6	15	20	15	6	1
$h =$	0	1	2	3	4	5	6

Note that the values in each row sum up to a total of 2^s, which is the total amount of occurrences. The top of the triangle is shifted to the left so that the entries of s and h match to their corresponding binomial coefficient. Note that instead of h, we could have used l since both values logically follow once one of them is know: a combination for a certain s, having h high signals means having $s - h$ low signals.

For example, for $s = 6$, out of all $2^6 = 64$ possible combinations, how many contain exactly 4 low signals? The answers is 15 and can either be found in row $s = 6$ and column $h = 4$ or in row $s = 6$ and column $h = 6 - 4 = 2$.

Now, to calculate $C_3(n)$, we will again make use of Equation 3.1. The probability of a certain combination nHmL (where the order of the signals does not matter) is equal to:

$$Occurrences(nHmL) \cdot P(H)^n \cdot P(L)^m$$

Where $P(H) = 2/3$ and $P(L) = 1/3$.
The following overview includes for $s = 3$ all four (in evidence) different combinations: the amount of occurrences, the probability of the combination and the expected value given that the event (combination) occurs.

1. 3H0L

   - Occurrences: 1;
   - $P(3H0L) = 1 \cdot P(H)^3 \cdot P(L)^0 = (\frac{2}{3})^3 = \frac{8}{27}$;
   - $C_3(n|3H0L) = n$.
     The calculation of the expected value is very straightforward for this combination. Three high signals immediately leads to a correct cascade and all n agents vote correctly in that case.

2. 2H1L

   - Occurrences: 3;
   - $P(2H1L) = 3 \cdot P(H)^2 \cdot P(L)^1 = (\frac{2}{3})^2 \cdot \frac{1}{3} = \frac{12}{27}$;
   - $C_3(n|2H1L)$ breaks down to two possibilities:

- $\frac{2}{3}$ chance that a_4 receives H. To simplify the notation, I will denote this as $a_4 = H$. In this case a correct cascade starts. But since one agent already voted incorrectly, we get that:
  $C_3(n|2H1L \wedge a_4 = H) = n - 1$;
- $\frac{1}{3}$ chance that a_4 receives L, in that case no cascade starts and we return to baseline. Since the upcoming agents will no longer sacrifice, for the remainder of the problem we have that $s = 0$. Since two agents already voted correctly we have that:
  $C_3(n|2H1L \wedge a_4 = L) = 2 + C_0(n - 4)$.

Now combining these two possibilities with their respective probabilities we get:

$$\begin{aligned} C_3(n|2H1L) =& P(a_4 = H|2H1L) \cdot C_3(n|2H1L \wedge a_4 = H) \ + \\ & P(a_4 = L|2H1L) \cdot C_3(n|2H1L \wedge a_4 = L) \\ =& \frac{2}{3} \cdot (n-1) + \frac{1}{3} \cdot (2 + C_0(n-4)) \\ =& \frac{2}{3}n - \frac{2}{3} + \frac{2}{3} + \frac{1}{3}C_0(n-4) \\ =& \frac{1}{3}C_0(n-4) + \frac{2}{3}n \end{aligned}$$

3. 1H2L

- Occurrences: 3;
- $P(1H2L) = 3 \cdot P(H)^1 \cdot P(L)^2 = \frac{2}{3} \cdot (\frac{1}{3})^2 = \frac{6}{27}$;
- $C_3(n|1H2L)$ again breaks down into two possibilities:
  - $\frac{2}{3}$ chance that a_4 receives H, resulting in an equal amount of both signals. The consequence is a baseline situation. Two agents already voted correctly, so we get:
    $C_3(n|1H2L \wedge a_4 = H) = 2 + C_0(n - 4)$;
  - $\frac{1}{3}$ chance that a_4 receives L, in that case there are two more low signals than high signals. We get an incorrect cascade but one agent already voted correctly. We get: $C_3(n|1H2L \wedge a_4 = L) = 1$.

Now combining these two possibilities with their respective probabilities we get:

$$\begin{aligned} C_3(n|1H2L) =& P(a_4 = H|1H2L) \cdot C_3(n|1H2L \wedge a_4 = H) \ + \\ & P(a_4 = L|1H2L) \cdot C_3(n|1H2L \wedge a_4 = L) \\ =& \frac{2}{3} \cdot (2 + C_0(n-4)) + \frac{1}{3} \cdot 1 \\ =& \frac{4}{3} + \frac{2}{3}C_0(n-4) + \frac{1}{3} \\ =& \frac{2}{3}C_0(n-4) + 1\frac{2}{3} \end{aligned}$$

4. 0H3L

   - Occurrences: 1;
   - $P(0H3L) = 1 \cdot P(H)^0 \cdot P(L)^3 = (\frac{1}{3})^3 = \frac{1}{27}$;
   - $C_3(n|0H3L) = 0$.
     If the first three agents vote incorrectly based on a low signal, an incorrect cascade begins and no agent votes correctly.

With the information above we can finally calculate $C_3(n)$ as follows:

$$\begin{aligned}
C_3(n) =& P(3H0L) \cdot C_3(n|3H0L) + P(2H1L) \cdot C_3(n|2H1L) \; + \\
& P(1H2L) \cdot C_3(n|1H2L) + P(0H3L) \cdot C_3(n|0H3L) \\
=& \frac{8}{27}n + \frac{12}{27} \cdot \left(\frac{1}{3}C_0(n-4) + \frac{2}{3}n\right) + \frac{6}{27} \cdot \left(\frac{2}{3}C_0(n-4) + 1\frac{2}{3}\right) + \frac{1}{27} \cdot 0 \\
=& \frac{8}{27}C_0(n-4) + \frac{16}{27}n + \frac{10}{27}
\end{aligned}$$

To see if indeed one extra agent who provides information leads to better results, we calculate and compare:

$$\begin{aligned}
C_3(10) &\approx 7.59 \\
C_3(100) &\approx 82.24 \\
C_3(1000) &\approx 828.90
\end{aligned}$$

For $n = 10, 100, 1000$ we get a higher expected amount of correct answers with $s = 3$ than with $s = 0$. We already benefit from this extra information as soon as $n = 8$:

$$\begin{aligned}
C_3(6) &< C_0(6) \\
C_3(8) &> C_0(8)
\end{aligned}$$

This means that for all $n \geq 8$ it is a better option for the third agent to act on his own information than to base his actions on the first two agents. The reason for this is that his extra information 'steers' the group towards a correct cascade. The downside of the sacrificial behavior becomes clear for $n = 4$ and $n = 6$, where $s = 3$ actually performs worse. The reason is that in those cases the sacrifice of a_3 can not be compensated by having enough successors make use of the extra information. His own (possibly) incorrect answer will then play too big of a role in proportion to the (small) group. Intuitively, the more agents that have to act, the more valuable the information becomes. That is why the expected percentage of correct answers increases as the amount of agents grows.

Our final goal would be to know how many agents s can we sacrifice for an arbitrary value of n so that we can expect the highest amount of correct answers. Before moving to the calculation for $s = 4$ directly, I will touch on the phenomenon of *subcascades*.

For $s = 3$ we were able to treat combinations nHmL with the same signal difference $(n - m)$ as equal, since it did not matter in what order the evidence became publicly

available. For $s \geq 4$ however, it turns out the order does start to matter in some cases.
Imagine for $s = 4$ how a_4 would reason in the case that: a_1, a_2 and a_3 all provide the same signal, but a_4 receives a different signal himself. Now, a_4 would not know if those initial three signals were high (and correct) or low (and incorrect), but he reasons there are two possibilities:

- The first three signals were high signals, his own is low:
  - If a_4 indeed sacrifices by voting for his own signal, he will pass on the following public information to a_5: $< high, high, high, low >$, and the result is a correct cascade, with the total amount of correct answers being equal to $n - 1$.
  - But had a_4 in this case not sacrificed his information, the result for the rest of the group would be the same: a correct cascade appears. The amount of evidence passed on would then be $< high, high, high, high >$. Since a_4 did not vote incorrectly in this situation, the total amount of correct answers will be equal to n.

  To conclude, in this case it would be better for a_4, despite being a sacrificial agent, to ignore his signal and copy the first three agents. The result for the total amount of correct answers would be an improvement of exactly 1.

- The first three signals were low signals, his own is high:
  - In this case, if a_4 indeed sacrifices, the evidence sums up to: $< low, low, low, high >$. An incorrect cascade will start and only a_4 votes correctly. The result is that in total 1 agent votes correctly.
  - But if a_4 does not sacrifice, the sequence becomes $< low, low, low, low >$. Now, because of the incorrect cascade, the result is a total of 0 correct answers.

  To conclude, the best option for a_4 in this case is to sacrifice. The result is again an improvement of 1 correct answer.

Upon receiving three equal signals, a_4 reasons that whatever he votes, he will not be able to avert the group from a cascade. In other words, whatever vote a_4 makes, he can not prevent that all agents $a_5, a_6, .., a_n$ will vote similarly to agents a_1, a_2, a_3. Since his vote can not have any influence on the upcoming agents' decision making, a_4 reasons that his best option is to maximize his own chances of being correct. A problem is, that in-group agents can obviously not know whether a signal is *high* or *low*. However, based on the higher probability of high signals in comparison to low signals, it is more likely that the first three signals were high signals. Thus, by ignoring his sacrificial status and his own signal, and by copying the actions of the previous agents, a_4 increases the expected correctness for himself and the group as a whole.

We could say that a_4 has entered a *subcascade* or a *precascade*: a cascade amongst sacrificial agents in which the agents don't base their actions on their own signals but on the signals of their predecessors, only because they know that their actions can not possibly influence other agents. So, for values of $s > 3$ we should redefine sacrificial agents as agents who share their information by acting independently, *only* in the case that their information could possibly lead to different behavior for any non-sacrificial agent. A subcascade is then nothing more than an extension of a cascade that is inevitably going to take place among the non-sacrificial agents.

We can again use the binomial theorem to calculate for any s, how many of the 2^s combinations will contain subcascades. Before we do so, let's analyze how often a certain signal needs to occur among the sacrificial agents in order for a subcascade to start. We have seen that a subcascade starts when it is no longer possible for the remainder of the sacrificial agents to prevent a cascade from starting at agent a_{s+1}. In order for this to happen, a certain signal should already have occurred in more than half of the sacrificial agents, otherwise the remainder of the sacrificial agents together with a_{s+1} would be able to change the balance in favor of the other signal. So, given s agents, we need the following amount of equal votes at a certain point for a subcascade to start:

$$\lceil s/2 + 1 \rceil$$

However, in order for a subcascade to start, there has to be at least one sacrificial agent left *after* those equal signals have been shared (otherwise, a 'normal' cascade will appear). So, $\lceil s/2 + 1 \rceil$ signals needs to 'fit' in the first $s - 1$ sacrificial agents. As we can see for $s = 3$, we have that $\lceil 3/2 + 1 \rceil = 3$ signals need to fit in the first $3 - 1 = 2$ agents, which is impossible. That is why we were justified in disregarding the order of the signals for $s = 3$, because there weren't any subcascades.

For $s = 4$ we get $\lceil 4/2 + 1 \rceil = 3$ and $4 - 1 = 3$. For obvious reasons, three signals can only be arranged in one way for three agents, but since this can be done with both low and high signals, we need to multiply by 2. This means that for $s = 4$, out of all $2^4 = 16$ combinations, 2 will include a subcascade (which are the cases analyzed previously). More generally, we can calculate the amount of subcascades for a certain s with:

$$Subcascades(s) = 2 \cdot \binom{s-1}{\lceil s/2 + 1 \rceil}$$

Returning to the calculation of $C_4(n)$, we require for all combinations that vary in evidence their probability and their respective expected correctness. Multiplying those properties for all combinations and then adding the sub-results leads to the following calculation:

$$
\begin{aligned}
C_4(n) =& P(4H0L) \cdot C_4(n|4H0L) \ + \\
& \frac{3}{4} \cdot P(3H1L) \cdot C_4(n|3H1L \wedge \neg subcascade) \ + \\
& \frac{1}{4} \cdot P(3H1L) \cdot C_4(n|3H1L \wedge subcascade) \ + \\
& P(2H2L) \cdot C_4(n|2H2L) \ + \\
& \frac{3}{4} \cdot P(1H3L) \cdot C_4(n|1H3L \wedge \neg subcascade) \ + \\
& \frac{1}{4} \cdot P(1H3L) \cdot C_4(n|1H3L \wedge subcascade) \ + \\
& P(0H4L) \cdot C_4(n|0H4L) \\
=& \frac{16}{81} \cdot n \quad + \quad \frac{3}{4} \cdot \frac{32}{81} \cdot (n-1) \quad + \quad \frac{1}{4} \cdot \frac{32}{81} \cdot n \ + \\
& \frac{24}{81} \cdot (2 + C_0(n-4)) \quad + \quad \frac{3}{4} \cdot \frac{8}{81} \cdot 1 \quad + \quad \frac{1}{4} \cdot \frac{8}{81} \cdot 0 \\
=& \frac{24}{81} C_0(n-4) + \frac{24}{81} n + \frac{24}{81}(n-1) + \frac{54}{81} \\
=& \frac{8}{27} C_0(n-4) + \frac{16}{27} n + \frac{10}{27}
\end{aligned}
$$

In case this last expression does not look familiar to the reader: it should. It is equal to the previous calculation for $s = 3$, so that we have:

$$C_4(n) \quad = \quad C_3(n)$$

So why did we not improve on the group outcome by having the fourth agent sacrifice? This is again due to similarity in signal accuracy and the fact that with an even amount of information, we are either in a cascade or at baseline level. Since the first agent to act *after* the s sacrificial agents also has his own observation as evidence, the amount of signals available to that agent is equal to $s + 1$ (an odd s leads to an even amount of evidence).

In our case for $s = 3$, if a_4 was not already in a cascade, he would vote for his own signal. There are only two cases for $s = 3$ in which a_4, would *not* have voted for his own signal, namely after (low, low, low) and $(high, high, high)$. Now, one would initially think the difference for $s = 4$ would be that in those very two instances, a_4 *would* indeed have shared his own information. However, those are the two cases for which we have seen that a_4 would still not vote for his own signal despite being a sacrificial agent (the subcascades situations). Had we not taken into account the emergence of subcascades, the performance for $s = 4$ would actually have been slightly worse than for $s = 3$.

Since the above-mentioned properties apply to higher values of s, we can generally say that for $s = \{1, 3, 5..\}$ we have that:

$$C_s(n) \quad = \quad C_{s+1}(n)$$

As one might expect, with increasing values for n we start to see an improvement[3] with $s = 5$ compared to $s = 3$ and $s = 4$:

$$C_5(10) \approx 7.52$$
$$C_5(100) \approx 84.54$$
$$C_5(1000) \approx 854.91$$

3.2.1 A Recursive Approach

Calculating for higher values of s directly becomes very cumbersome. But by dividing the problem in steps of two and by keeping track of the evidence difference, we can calculate recursively. In essence, this calculation comes down to:

Let n be the amount of agents and let s be the amount of sacrificial agents.
Sacrifice(n,s,EvidenceDifference=0):

- If after two votes it is certain that we have a cascade or all sacrificial agents have at this point sacrificed:
  - Calculate the outcome using one of our earlier methods
- If the outcome of a cascade is not certain after two votes:
  Return a sum of the following recursive calls:
  - 4/9 times (1 + Sacrifice(s - 2, n - 2, EvidenceDifference).
    If two agents both vote differently, one of them is correct. The evidence difference does not change.
  - 4/9 times (2 + Sacrifice(s - 2, n - 2, EvidenceDifference + 2).
    If two agents both vote correctly, both are correct. The evidence difference increases by 2.
  - 1/9 times Sacrifice(s - 2, n - 2, EvidenceDifference - 2).
    If two agents both vote incorrectly, neither are correct. The evidence difference decreases by 2.

[3] The calculation is rather similar to the previous ones. For the interested reader, a python code with the calculation can be found in Appendix B

For a more detailed version of this calculation, please see Appendix B for a Python implementation of this recursion. By plotting the expected correctness for $n = 100$ for different values of s we get:

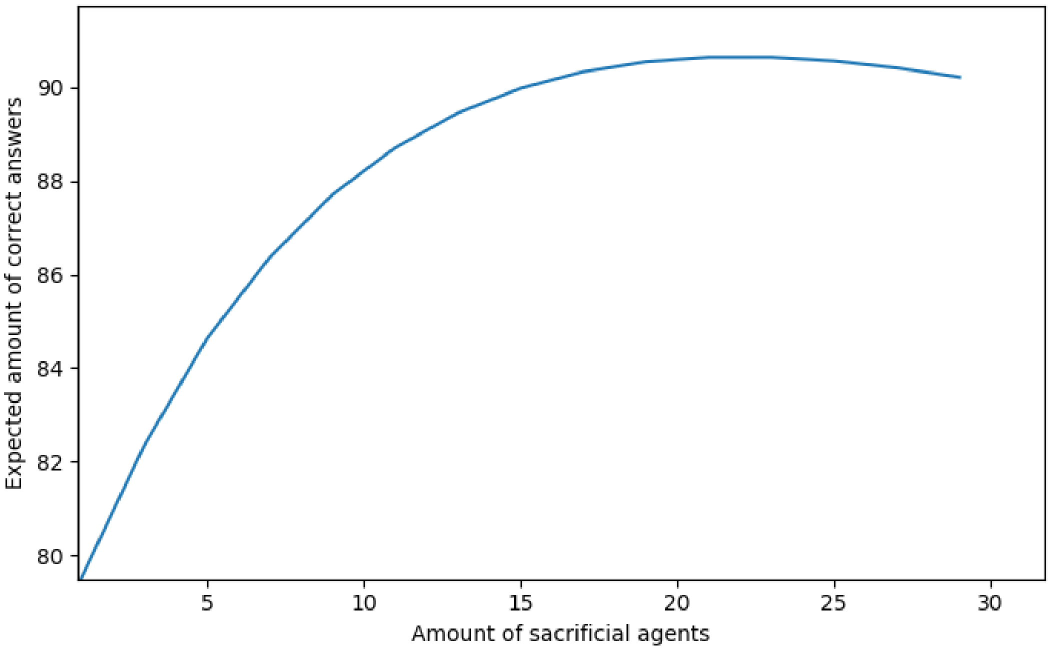

Figure 3.1: The graph is steep in the origin but flattens for higher values of s. This nicely illustrates that the information of the earlier agents is the most valuable. The first agents that sacrifice have the most successors to share their information with. Python tells us that that maximum lies at $s = 23$.

To finally answer our question: for an urn problem with 100 agents, 23 agents should sacrifice to achieve the best expected group outcome. One could say that the cascades that these agents enter are *justified cascades*. Despite being possibly incorrect, following these justified cascade does on average lead to the highest amount of correct answers for the group.

Chapter 4

Discussion & Conclusion

This thesis addresses the problem of poor group results caused by information cascades by analyzing the trade-off between sharing versus making use of information. I conclude that for certain sequential decision making problems there exists an optimal balance between independent and dependent decision making when it comes to group outcomes. My findings also confirm that agents that act earlier in the sequence provide more useful information than agents acting later on. This thesis provides insights by taking a step in a new direction, but has its limits due to the narrow and specific analysis. For more fruitful findings, further research is required.

One direction in which this thesis could be expanded is by combining the approach of self-sacrificial behavior with research from other fields. Being mainly focused on probability theory, this thesis does not focus on the (higher-order) reasoning being carried out by the agents. The key question that remains: could this theoretically optimal approach be actualized by rational agents? The field of Epistemic Logic could shed light on this aspect of the problem, e.g. by incorporating the idea of sacrificial agents into the model used by Baltag et al [2].

While information cascades may occur in human-like settings, this thesis mainly focused on the behavior of rational agents. A comparison was made by looking at group outcomes for both individual and group based reward systems. To more accurately investigate human-like behavior, one could implement a combined reward system where agents get partially rewarded for their own performance and partially rewarded for the group performance. This would mean that agents could take risks by trying to get rewarded immediately, or could play it safe by increasing their odds of at least receiving some reward by sharing their information with the group. Another similar take would be to have agents behave in a stochastic way.

Extra research similar to this thesis could be carried out by increasing the amount of true knowledge in a more practical way. For instance, instead of having a fixed amount of agents sacrifice, one could artificially raise the cascade boundary. This way, a cascade would only start if the evidence difference reaches a certain predefined value, leading to cascades that are based on more information. One could also experiment with a predefined certainty that a group must have of not being in an incorrect cascade before acting upon that very cascade. Requiring high certainty would cause agents to postpone the use of public information.

All in all, self-sacrificial behavior seems to be a promising strategy to prevent the harm caused by information cascades. Further research will indicate the value of this new strategy for practical applications.

Acknowledgements

First of all, I would like to thank my supervisor Erik for his time and patience during this process. Writing a thesis was completely new for me, but luckily Erik was always available to guide me and answer all of my questions. I am also grateful to friends and fellow students for proofreading my thesis, especially Jonah for his thorough analysis. Finally, I would like to express my sincerest appreciation for my long-time friend Bas, who shared some mathematical insights that were of great value to me.

Bibliography

[1] J. Neeman, N. Truong, and S. Troxler. *Lecture: Condorcet's Theorem.* Aug. 2010. URL: https://www.stat.berkeley.edu/~mossel/teach/SocialChoiceNetworks10/ScribeAug31.pdf.

[2] A. Baltag et al. "Logical Models of Informational Cascades". In: *Logic across the University: Foundations and Applications.* Sept. 2013, pp. 405–432. ISBN: 978-1-84890-122-3.

[3] Jerry B. Harvey. "The abilene paradox: The management of agreement". In: *Organizational Dynamics* 3.1 (1974), pp. 63–80. ISSN: 0090-2616. DOI: https://doi.org/10.1016/0090-2616(74)90005-9. URL: http://www.sciencedirect.com/science/article/pii/0090261674900059.

[4] H. Pishro-Nik. "Law of Large Numbers". In: *Introduction to probability, statistics, and random processes.* 2014. Chap. 7.1.1. URL: https://www.probabilitycourse.com.

[5] S. Bikhchandani, D. Hirshleifer, and I. Welch. "A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades". In: *Journal of Political Economy 100* no. 5 (Oct. 1992), pp. 992–1026. DOI: 10.1086/261849.

[6] T. N. Le, V. G. Subramanian, and R. A. Berry. "The value of noise for informational cascades". In: *2014 IEEE International Symposium on Information Theory.* 2014, pp. 1101–1105.

[7] Tuvana Pastine and Ivan Pastine. *Signal accuracy and informational cascades.* Working Papers 200620. School of Economics, University College Dublin, Nov. 2006. URL: https://ideas.repec.org/p/ucn/wpaper/200620.html.

[8] Jiemai Wu. "Helpful laymen in informational cascades". In: *Journal of Economic Behavior & Organization* 116 (2015), pp. 407–415. ISSN: 0167-2681. DOI: https://doi.org/10.1016/j.jebo.2015.05.002. URL: http://www.sciencedirect.com/science/article/pii/S0167268115001377.

[9] A.C. Achimescu. "Games and Logics for Informational Cascades". Master Thesis. Universiteit van Amsterdam, 2014.

[10] David Easley and Jon Kleinberg. "Information Cascades". In: *Networks, Crowds, and Markets.* July 2010. Chap. 16, pp. 483–508. ISBN: 9780521195331.

[11] Dr. Natarajan Meghanathan. *Information Cascades and Diffusion.* 2015. URL: http://www.jsums.edu/nmeghanathan/files/2015/05/CSC434-Fall2014-Module-4-Information-Cascades-Diffusion.pdf?x61976.

[12] William Talbott. *Bayesian Epistemology.* 2016. URL: https://plato.stanford.edu/archives/win2016/entries/epistemology-bayesian/.

Appendix A

Recurrence Equation

We can calculate the expected amount of correct answers with:

$$C_0(n) = 2 \cdot P(high) \cdot P(low) \cdot (1 + C_0(n-2)) + P(high)^2 \cdot n + P(low)^2 \cdot 0$$

Plugging in the following values for $P(high)$ and $P(low)$:

$$P(high) = 2/3$$
$$P(low) = 1/3$$

Gives us the following recurrence relation:

$$C_0(n) = \frac{4}{9}C_0(n-2) + \frac{4}{9}n + \frac{4}{9}$$

Solving this recurrence relation with WolframAlpha[1] gives:

$$C_0(n) = c_1(\frac{2}{9})^n + c_2(-\frac{2}{9})^n + \frac{4}{5}n - \frac{12}{25}$$

Since we are only working with even values of n we can remove parameter c_2 and write:

$$C_0(n) = c(\frac{2}{9})^n + \frac{4}{5}n - \frac{12}{25}$$

We know $C_0(n) = 0$ for $n = 0$. Solving the equation above for $n = 0$ then gives:

$$0 = c(\frac{2}{9})^0 + \frac{4}{5} \cdot 0 - \frac{12}{25}$$
$$c = \frac{12}{25}$$

Finally, we can directly calculate $C_0(n)$ with the equation:

$$C_0(n) = \frac{12}{25} \cdot (\frac{2}{9})^n + \frac{4}{5}n - \frac{12}{25}$$

[1]Available at:
https://www.wolframalpha.com/examples/mathematics/discrete-mathematics/recurrences/

Appendix B

Python Code

```
import math
import matplotlib.pyplot as plt

def Sac0Direct(number): #direct calculation for even numbers, s=0
    return 12/25 * (2/9)**number  + (4/5)*number - 12/25

def Sac0Rec(number): #recursive calculation for even numbers, s=0
    if number == 0:
        return 0
    else:
        return (4/9)*number + (4/9)*(1 + Sac0Rec(number-2))

def Sac3Rec(number): #recursive calculation for even numbers, s=3
    if number <= 3:
        return "Please enter a number greater than 3"
    else:
        return 10/27 + (8/27)*Sac0Rec(number-4) + (16/27)*number

def Sac4Rec(number): #recursive calculation for even numbers, s=4
    if number <= 4:
        return "Please enter a number greater than 4"
    else:
        return (24/81)*number + (24/81)*(Sac0Rec(number-4)) + \
        (54/81)  + (24/81)*(number-1)

def Ft(number): #returns the factorial of the given number
    return math.factorial(number)

def Sac5Rec(number): #recursive calculation for even numbers, s=5
    H = 2/3
    L= 1/3
    ans = 0
    if number <= 5:
        return "Please enter a number greater than 5"
    else:
```

```
        ans += (Ft(5)/(Ft(5)*Ft(0))) * \
        (H**5 * L**0 *(number))
        ans += (Ft(5)/(Ft(4)*Ft(1))) * \
        (H**4 * L**1 *(number-1))*(4/5) #subcascade cases
        ans += (Ft(5)/(Ft(4)*Ft(1))) * \
        (H**4 * L**1 *(number))*(1/5) #subcascade cases
        ans += (Ft(5)/(Ft(3)*Ft(2))) * \
        (H**3 * L**2 *(number-2))*(2/3) #evidence of a6 matters
        ans += (Ft(5)/(Ft(3)*Ft(2))) * \
        (H**3 * L**2 *(3+Sac0Rec(number-6)))*(1/3) #evidence of a6 matters
        ans += (Ft(5)/(Ft(2)*Ft(3))) * \
        (H**2 * L**3 *(3+Sac0Rec(number-6)))*(2/3) #evidence of a6 matters
        ans += (Ft(5)/(Ft(2)*Ft(3))) * \
        (H**2 * L**3 *(2))*(1/3) #evidence of a6 matters
        ans += (Ft(5)/(Ft(1)*Ft(4))) * \
        (H**1 * L**4 *(1))*(4/5) #subcascade cases
        ans += (Ft(5)/(Ft(1)*Ft(4))) * \
        (H**1 * L**4 *(0))*(1/5) #subcascade cases
        ans += (Ft(5)/(Ft(0)*Ft(5))) * \
        (H**0 * L**5 *(0))
        return  ans

#n even agents, s odd sacrifices, ED = Evidence Difference
def SacSrec(n, s, ED = 0):
    if s <= 2 and ED == 0:
        #we know that s=0,1,2 lead to equal outcomes:
        return Sac0Direct(n)

    if s == 1 and ED == 2:
    #if there is 1 sacrificial agent left and ED = 2, then:
        return 2/3 * n + 2/9 * (1 + (n-2)) + 1/9 * Sac0Direct(n-2)
#2/3 H -> correct cascade,  2/9 LH -> correctcascade, 1/9 LL -> baseline

    if s == 1 and ED == -2:
    #if there is 1 sacrificial agent left and ED = -2, then:
        return 1/3 * 0 + 2/9 * (1 + 0) + 4/9 * (2 + Sac0Direct(n-2))
#1/3 L -> incorrect cascade, 2/9 HL -> incorrect cascade, 4/9 HH -> baseline

    if ED - s >= 2:
#cascade will inevitably happen so all agents join the (sub)cascade (high)
        return n

    if -ED - s >= 2:
#cascade will inevitably happen so all agents join the (sub)cascade (low)
        return 0

#if not any of the above, recursive call:
    else:
```

```
#4/9  chance at HL or LH -> 1 correct, ED=same
#4/9 chance at HH -> 2 correct, ED+2
#1/9 chance at LL -> 0 correct, ED-2
        return 4/9 * (1 + SacSrec(n - 2, s - 2, ED)) + 4/9 * \
        (2 + SacSrec(n - 2, s - 2, ED + 2)) + 1/9 * SacSrec(n - 2, s - 2, ED -2)


#Driver Code:

#comparing correctness for n=100:
print(Sac0Rec(100))
print(Sac0Direct(100))
print(Sac3Rec(100))
print(Sac4Rec(100))
print(Sac5Rec(100))

print()
print("Determining optimal s:")
#determine optimal s for n=100 and create graph
Cs = []
Ss = []
OptimalS = 0
BestSoFar = 0
print("Sacrifices:     On avg. correct:")
for s in range(1, 30, 2):
    C = SacSrec(100, s)
    Cs.append(C)
    Ss.append(s)
    print('%f\t%f' % (s, C))
    if C > BestSoFar:
        BestSoFar = C
        OptimalS = s
fig = plt.figure()
plt.plot(Ss, Cs)
plt.xlabel("Amount of sacrificial agents")
plt.ylabel("Expected amount of correct answers")
plt.show()
print()
print("Conclusion:")
print(str(OptimalS) + " Agents should sacrifice to achieve a maximal \
amount of correct answers for n=100.")
```