EFFECTS OF OVERCONFIDENCE ON DECISION MAKING

BY

JIARUI SUN

Senior Thesis in Computer Engineering

University of Illinois at Urbana-Champaign

Advisor: Lav R. Varshney

May 2019

ABSTRACT

This thesis aims to explore the effect of overconfidence on people's decision making. To approach this topic, a standard binary detection problem is considered, and its associated individual decision rule and decision fusion rule are derived. Following an axiomatic and empirical approach, a variant of the Prelec function from cumulative prospect theory is then developed to model the effect of overconfidence as a function of level of training. Next, the probability of detection after decision fusion is derived, and a combinatorial optimization is considered which aims to select a subgroup of people/agents to maximize the overall probability of detection.

Keywords: overconfidence, detection theory, combinatorial optimization

To my parents, for their love and support

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor Professor Lav Varshney. Whenever I encountered a problem on my thesis, he was always glad to help and lead me to the right direction. Instead of directly telling me how to solve the problems, Professor Varshney always encouraged me to figure them out by myself, while providing the right materials for me to read and digest. Without him, this thesis would not be completed.

I would also like to thank Professor Mikhail Lavrov from the math department. While I was struggling on the optimization part of the thesis, he helped me out with his passion, and finally I was able to step on the right track. I am very grateful for his time and valuable instruction.

Finally, I must express my gratitude to my parents and to my girlfriend for providing me with their support and encouragement throughout this last year as an undergraduate student at UIUC. Thank you.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Overconfidence, a bias in which a person's subjective confidence in his or her judgements is greater than the objective accuracy of those judgements, can negatively affect the performance of team decision making. Inspired by this psychological effect [1], here we mathematically model and study the effect of overconfidence on group decision making.

If we want to examine overconfidence's effect in detail—both analytically and numerically, a mathematical model to describe people's decision-making process needs to be established. In this thesis, we established a detection model, which is given in Section 3.1 in detail. The model itself is a parallel fusion network [2], which is described in Section 2.2.

However, having a well-mathematized model is often not enough. In Chapter 4, we formulated an optimization problem, in which a subgroup of people/agents is to be selected to maximize its objective. We also proposed an algorithm for the optimization problem that reduces the running time from exponential to linear.

CHAPTER 2

BACKGROUND MATERIAL

2.1 Overconfidence among beginners

Overconfidence, as mentioned in Chapter 1, is a bias that occurs when one overestimates the chance that one's judgments are accurate or that one's decisions are correct [1]. Sanchez and Dunning conducted six studies to investigate the development of overconfidence among beginners [1]. Their results show that although beginners started out underconfident in their judgments (since they had zero experience), they rapidly surged to a "beginner's bubble" of overconfidence.

More specifically, they considered the relationship between performance and confidence as a function of experience, which is shown in Figure 2.1. In this thesis, we consider settings where such underconfident/overconfident people work together to make decisions, through some designed weighted voting rule that uses both their local decisions and their stated confidence levels.

2.2 Parallel fusion network

Parallel fusion network, described in [2], is a parallel decision-making structure consisting of a number of detectors/agents whose decisions are made locally and are finally transmitted to a decision fusion center for decision combining. This thesis utilizes the parallel fusion network as the underlying mathematical structure for group decision making. In order to analyze the full decision-making process, the decision rule for local detector and fusion center needs to be derived. A typical parallel fusion network architecture is shown in Figure 3.2 on page 7.

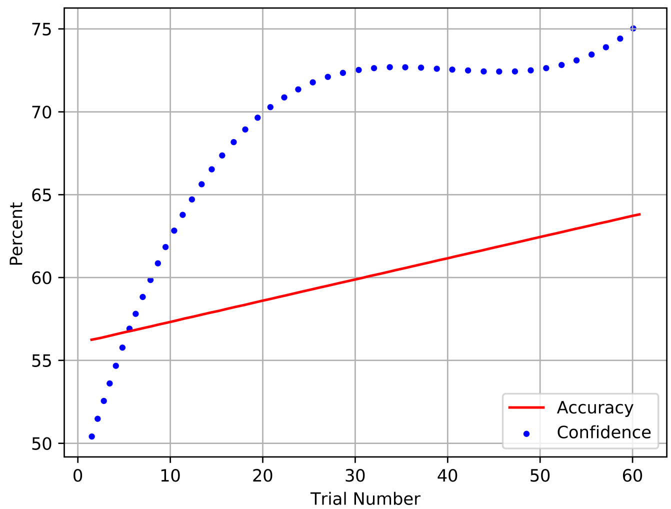

Figure 2.1: Confidence and accuracy trends over 60 trials for one study [1].

2.3 Probability weighting function

Empirical studies have shown that decision makers do not treat probability linearly. While making decisions, people tend to overweight small probabilities and underweight the large ones. Cumulative prospect theory seeks to provide a psychological understanding of such human behaviors [3]. It introduces the notion of probability weighting functions to explain and model the distortions. Among these functions, the Prelec function satisfies the majority of the axiomatic behavior of the cumulative prospect theory [4]. In this thesis, we derive a similar two-parameter probability weighting function to model the underconfidence/overconfidence effect on decision makers' stated confidences, as a function of experience.

2.4 Poisson binomial distribution

The Poisson binomial distribution is the discrete probability distribution of a sum of independent Bernoulli trials that are not necessarily identically distributed [5]. There has been research on obtaining closed-form expression for its probability density function (PDF) and cumulative distribution function (CDF) since the original form is infeasible to compute when the number of trials gets large [6]. The Poisson binomial distribution is essential to this thesis because it is the intermediate result we obtain for calculating the probability of detection after fusion in Section 4.1.

CHAPTER 3

MATHEMATICAL MODEL

We begin by considering a standard team decision-making scenario: a group of people having different expertise/confidence levels governed by the overconfidence phenomenon, simultaneously observing a phenomenon and making their individual decisions without communicating with each other. Then, a final decision is made by aggregating these local decisions through a weighted voting rule.

In Section 2.2, we introduced the parallel fusion network. It will serve as the mathematical model for interpreting the scenario above. In Section 3.1, the problem is introduced. Then, the rules for deciding decisions are designed in Sections 3.2 and 3.3. Overconfidence effect will be take into consideration in Section 3.4.

3.1 Problem description

Consider a binary detection problem. A binary input $H \in \{0, 1\}$ is observed by N independent agents. Additive white Gaussian noise exists for each agent's detection with zero mean and known variance, representing reported (not yet taking the overconfidence effect into account) confidence level of each agent. The raw signal will be encoded to $\pm a$ and decoded back to $\hat{H} \in \{0, 1\}$ at each individual agent. The overall information propagation is shown in Figure 3.1, and the system is shown in Figure 3.2. In terms of notation, we have:

- H: binary signal transmitted
- U: encoded signal as $\pm a$
- V_i: noisy received observation by each detector/agent, $i = 1, ..., N$

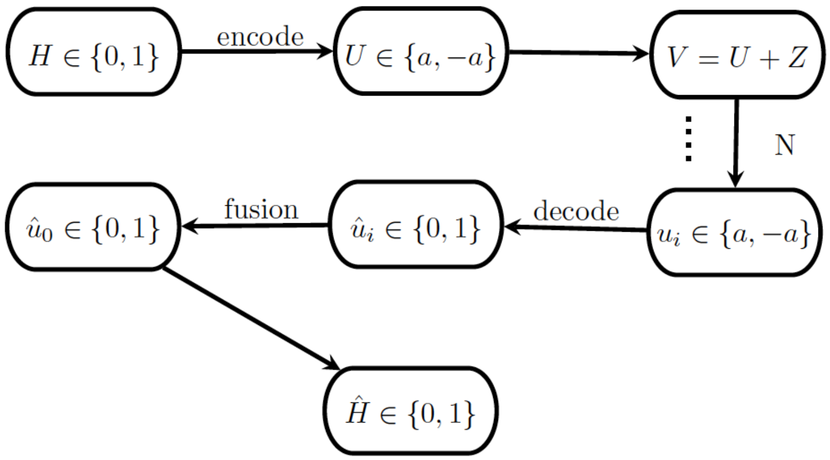

Figure 3.1: Block diagram representing information propagation.

- Z_i: the individual WGN with form $Z \sim N(0, \sigma_i^2)$, $i = 1, ..., N$
- c_i: individual confidence level reported by each detector/agent, $i = 1, ..., N$
- u_i: individual decision by each detector/agent, $i = 1, ..., N$
- u_0: global fused decision
- $\hat{H}$: final output

Our goal is to figure out how the final decision u_0 is made based on observation $V_1, \ldots, V_N$. This requires us to design the local decision rule for each agent and the decision fusion rule for fusion center.

3.2 Local decision rule

In this section, we consider the local decision rule. All agents are considered to be identical, and their operations are considered to be unrelated. Thus, we can simplify the problem to binary detection performed by one single agent.

According to the notation in the previous section, hypothesis 0 (signal absence) is mapped to a, and hypothesis 1 (signal present) is mapped to $-a$. The additive white Gaussian noise, representing uncertainty in observations

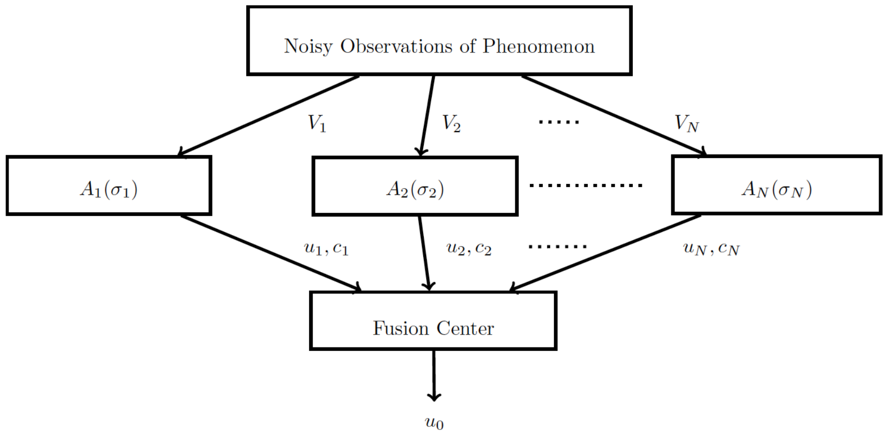

Figure 3.2: Block diagram of a parallel fusion network. A_i stands for agent i.

(and also, for now, the confidence level), is $Z \sim N(0, \sigma_i^2)$ for agent i. The PDF of Z is:

$$f_Z(z) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(z-0)^2}{2\sigma_i^2}} = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[\frac{-z^2}{2\sigma_i^2}\right].$$

Since the observation V is either $a+Z$ or $-a+Z$ depending on the encoded hypothesis U, we have $V \sim N(a, \sigma_i^2)$ conditional on $U = a$ and $V \sim N(-a, \sigma_i^2)$ conditional on $U = -a$. For agent i, explicitly we have:

$$f_{V|U}(v_i|u = +a) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(v_i-(+a))^2}{2\sigma_i^2}} = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[\frac{-(v_i - a)^2}{2\sigma_i^2}\right],$$

$$f_{V|U}(v_i|u = -a) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(v_i-(-a))^2}{2\sigma_i^2}} = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[\frac{-(v_i + a)^2}{2\sigma_i^2}\right].$$

Based on the likelihood ratio test, we have:

$$\begin{aligned}\Lambda(v_i) &= \frac{p(v_i|u = +a)}{p(v_i|u = -a)} \\ &= \frac{f_{V|U}(v_i|u = +a)}{f_{V|U}(v_i|u = -a)} \\ &= \exp\left[\frac{-(v_i - a)^2 + (v_i + a)^2}{2\sigma_i^2}\right] \\ &= \exp\left[\frac{2av_i}{\sigma_i^2}\right].\end{aligned} \tag{3.1}$$

For simplicity, we assume the cost of making correct decision is 0 and the cost of making incorrect decision is 1, and the prior probabilities are the same for both hypotheses, i.e., $p(u = a) = p(u = -a) = 0.5$. Under such simple case, we have:

$$\Lambda(v_i) = \exp\left[\frac{2av_i}{\sigma_i^2}\right] \underset{u_i=-a}{\overset{u_i=a}{\gtrless}} \frac{0.5}{0.5} = 1.$$

By taking the logarithm,

$$\log \Lambda(v_i) = \left[\frac{2av_i}{\sigma_i^2}\right] \underset{u_i=-a}{\overset{u_i=+a}{\gtrless}} \log(1) = 0.$$

Since $2a/\sigma_i^2 > 0$, a more simplified result can be obtained:

$$[v_i] \underset{u_i=-a}{\overset{u_i=+a}{\gtrless}} 0. \tag{3.2}$$

Here, (3.2) explicitly state the local decision rule for the ith agent.

3.3 Decision fusion rule

In this section, we derive the decision fusion rule for obtaining the global decision. We use similar notations as [2]. Let P_{Fi} and P_{Mi} denote the probabilities of false alarm and miss for agent i respectively:

$$P_{Fi} = P(u_i = -a|H = 0) \rightarrow P_{Fi} = P(u_i = -a|u = +a),$$

$$P_{Mi} = P(u_i = +a|H = 1) \rightarrow P_{Mi} = P(u_i = +a|u = -a).$$

We need to calculate P_{Fi} and P_{Mi} first.

The Q-function (tail distribution function of the standard normal distribution) is introduced:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du.$$

Since in our simple case the decision boundary is 0, we define probability of error as $P_{Ei} \triangleq P_{Fi} = P_{Mi}$. We use P_{Fi} for calculation purposes:

$$
\begin{aligned}
P_{Ei} &= P_{Fi} \\
&= \int_{-\infty}^{0} f_{V|U}(v_i|u=+a)dv_i \\
&= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left[\frac{-(v_i-a)^2}{2\sigma_i^2}\right] dv_i \\
&= \frac{1}{\sqrt{2\pi\sigma_i^2}} \int_{-\infty}^{-a} \exp\left[\frac{-v_i^2}{2\sigma_i^2}\right] dv_i \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-a/\sigma_i} \exp\left[\frac{-v_i^2}{2}\right] dv_i \\
&= \frac{1}{\sqrt{2\pi}} \int_{a/\sigma_i}^{\infty} \exp\left[\frac{-v_i^2}{2}\right] dv_i \\
&= Q\left(\frac{a}{\sigma_i}\right).
\end{aligned}
\tag{3.3}
$$

Thus, we obtain $P_{Ei} = P_{Fi} = P_{Mi} = Q(\frac{a}{\sigma_i})$.

Now, we design the optimum fusion rule. According to [2], the rule is given by the following likelihood ratio test (with assumptions that the cost of making correct global decision is 0 and the cost of making incorrect global decision is 1, and the prior probabilities are the same for both hypotheses):

$$
\frac{P(u_1, u_2, ..., u_N|u=-a)}{P(u_1, u_2, ..., u_N|u=+a)} \underset{u_0=+a}{\overset{u_0=-a}{\gtrless}} 1. \tag{3.4}
$$

Since the decisions are independent, the left hand side can be written as:

$$
\begin{aligned}
\frac{P(u_1, u_2, ..., u_N|u=-a)}{P(u_1, u_2, ..., u_N|u=+a)} &= \prod_{i=1}^{N} \frac{P(u_i|u=-a)}{P(u_i|u=+a)} \\
&= \prod_{S_{-a}} \frac{P(u_i=-a|u=-a)}{P(u_i=-a|u=+a)} \prod_{S_{+a}} \frac{P(u_i=+a|u=-a)}{P(u_i=+a|u=+a)} \\
&= \prod_{S_{-a}} \frac{1-P_{Mi}}{P_{Fi}} \prod_{S_{+a}} \frac{P_{Mi}}{1-P_{Fi}} \\
&= \prod_{S_{-a}} \frac{1-Q(\frac{a}{\sigma_i})}{Q(\frac{a}{\sigma_i})} \prod_{S_{+a}} \frac{Q(\frac{a}{\sigma_i})}{1-Q(\frac{a}{\sigma_i})},
\end{aligned}
\tag{3.5}
$$

where S_j is the set of all those local decisions that are equal to j.

Let $\hat{u}_i = 0$ decode $u_i = a$ and $\hat{u}_i = 1$ decode $u_i = -a$; finally, we can express the formula in logarithm space [2]:

$$\begin{aligned}&\sum_{i=1}^{N}\left[\hat{u}_i \log\frac{1-P_{Mi}}{P_{Fi}} + (1-\hat{u}_i)\log\frac{P_{Mi}}{1-P_{Fi}}\right]\\ &= \sum_{i=1}^{N}\left[\hat{u}_i \log\frac{1-Q(\frac{a}{\sigma_i})}{Q(\frac{a}{\sigma_i})} + (1-\hat{u}_i)\log\frac{Q(\frac{a}{\sigma_i})}{1-Q(\frac{a}{\sigma_i})}\right] \underset{u_0=+a}{\overset{u_0=-a}{\gtrless}} \log(1) = 0.\end{aligned} \tag{3.6}$$

At the end, we combine the result from (3.2) with slight modification. We have the local decision rule and decision fusion rule for the standard parallel fusion network:

$$\begin{aligned}&[v_i] \underset{\hat{u}_i=-a}{\overset{\hat{u}_i=+a}{\gtrless}} 0,\\ &\sum_{i=1}^{N}\left[\hat{u}_i \log\frac{1-Q(\frac{a}{\sigma_i})}{Q(\frac{a}{\sigma_i})} + (1-\hat{u}_i)\log\frac{Q(\frac{a}{\sigma_i})}{1-Q(\frac{a}{\sigma_i})}\right] \underset{\hat{H}=0}{\overset{\hat{H}=1}{\gtrless}} 0.\end{aligned} \tag{3.7}$$

It it important to note that v_i is defined as the observation, the channel output detected by the ith agent.

3.4 Incorporating underconfidence/overconfidence

In Sections 3.2 and 3.3, we derived the standard decision-making process of the parallel fusion network. In this section, the overconfidence effect is integrated into our model.

As mentioned in Section 2.3, Prelec derived the probability weighting function as a mathematical description of behavioral experiments in cumulative prospect theory. Here we consider a variant of the same function as a model for overconfidence as a function of experience. The two-parameter Prelec function is:

$$w(p) = \exp(-b((-\ln(p))^a)), 0 < a < 1. \tag{3.8}$$

Function (3.8) has several properties of interest:

- $w : (0,1) \xrightarrow{\text{onto}} (0,1)$ maps a valid probability to another valid probability
- w intersects the identity function $I(x) = x$; it is concave on one interval

and convex on the other one, depending on the values of parameters a and b.

Our interest is to find a variant of the Prelec function that can be applied on P_{Ei} in (3.3) to re-weight the probability of error to model the overconfidence effect. Thus, the function needs to satisfy the properties below:

- $w : (0, 0.5) \xrightarrow{\text{onto}} (0, 0.5)$ The domain and range of the function should not exceed 0.5. In an extreme case of random guessing (agent acquires zero expertise), the probability of error should be 0.5.
- w should intersect the identity function $I(x) = x$; it should be concave on the first interval and convex on the second interval. As mentioned in Section 2.1 and shown in Figure 2.1, people with expertise close to zero tend to have underconfidence, and after a little learning they have overconfidence. Thus, when the declared probability of error is close to zero, the function should map the values to larger ones to model the overconfidence behavior; when the declared probability of error is large, the function should map the values to smaller ones to model the underconfidence behavior.
- The saddle point of w should be closer to 0.5 than 0, since the experimental results described in Section 2.1 show that the shift from underconfidence to overconfidence is rapid as expertise increases.

We obtain the variant of the Prelec function to model the underconfidence/overconfidence effect (Figure 3.3).

$$w(p) = 0.5 \exp(-b((-\ln(2p))^a)). \tag{3.9}$$

Applying $w(p)$ to probability of error, the new rules for the parallel fusion network is:

$$\begin{aligned} &[v_i] \underset{\hat{u}_i=-a}{\overset{\hat{u}_i=+a}{\gtrless}} 0, \\ &\sum_{i=1}^{N} \left[\hat{u}_i \log \frac{1 - w(Q(\frac{a}{\sigma_i}))}{w(Q(\frac{a}{\sigma_i}))} + (1 - \hat{u}_i) \log \frac{w(Q(\frac{a}{\sigma_i}))}{1 - w(Q(\frac{a}{\sigma_i}))} \right] \underset{\hat{H}=0}{\overset{\hat{H}=1}{\gtrless}} 0. \end{aligned} \tag{3.10}$$

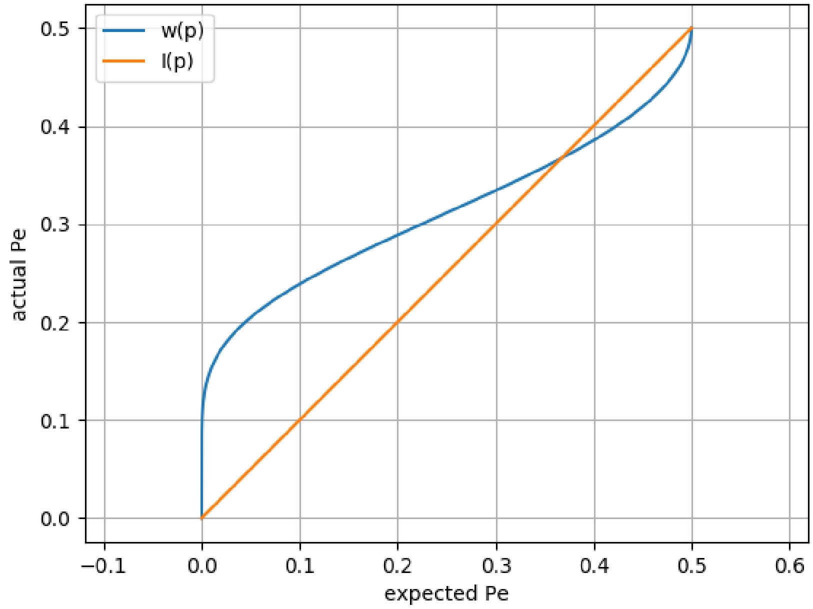

Figure 3.3: Function for modeling overconfidence effect, shown with $a = 0.528; b = 0.575$. Parameters are chosen based on the experimental data in [1]. The intersection $P_e = 0.366865$.

3.5 Simulation

In the previous sections, we derived the local decision rule and decision fusion rule for the proposed parallel fusion network. In this section, we describe a simulation process and provide results.

The simulation process is divided into three stages:

1. Given size of the agent pool, obtain reported confidence level for each agent as c_i. In our simulation, we sample from the truncated normal distribution $C \sim N(\mu = 0.7, \sigma^2 = 0.5^2)$ with lower limit of 0.5, upper limit of 1.0.

2. Using the decision rules (standard rule, re-weighted rule and majority vote rule), construct the information flow of the parallel fusion network, and obtain the final decision made by all simulated agents.

3. Compare the final decision with the previously defined groundtruth. Simulate the process a large number of times to get an average detection

Table 3.1: Detection accuracy for different size of agent pool. ST = standard, STW = re-weighted, MV = majority vote.

# of agents	ST	STW	MV
2	0.8500	0.8602	0.6965
5	0.8773	0.8757	0.8485
10	0.9931	0.9849	0.9522
15	0.9966	0.9949	0.9871
20	0.9994	0.9992	0.9954
30	0.9998	0.9997	0.9923
40	1.0000	0.9999	0.9991
80	1.0000	1.0000	1.0000
100	1.0000	1.0000	1.0000

accuracy. In our simulation, we perform $10,000$ iterations for each agent pool.

Table 3.1 shows the detection accuracy for different number of agents using the standard (ST) rule, re-weighted (STW) rule and majority vote (MV) rule, respectively.

From the results we obtained, we are able to conclude that as the size of agent pool increases, the overall detection accuracy also increases. We can also see that the rule when the overconfidence effect is in play generally performs worse than the optimal standard rule; the majority vote rule gives the worst accuracy among the three. The result based on the re-weighted rule should provide us the actual performance of the team decision making.

However, one may ask whether the majority vote rule could outperform the setting with overconfidence effect in certain settings (when the size of the agent pool is small enough). Thus, we simulate the system 100 times with small agent pools, and count the occurrences when the MV rule outperforms the STW rule.

Table 3.2 shows the results. We can see that no matter what size the agent pool has, the STW rule usually outperforms the MV rule; when the agent pool is large enough ($> \sim 7$), the STW rule is definitely better. We also observe that the STW rule gives better results when the size of the agent pool is even, which is worthy to experiment it on a real-world setting.

Table 3.2: Number of times when MV rule outperforms STW rule, among 100 trials.

# of agents	# of times
2	2
3	27
4	0
5	12
6	0
7	1
8	0
9	0
10	0

CHAPTER 4

OPTIMIZATION

After we have a clear picture about how decision is made by people under underconfidence/overconfidence effect through a parallel fusion network, we may ask ourselves a question: Is it possible for us to select a subgroup of agents, such that we can get the highest detection accuracy at the end of the network?

We show that the answer is "yes" in this chapter. That is, removing some agents may help the global performance due to the distorting effect from underconfidence/overconfidence, even though this reduces the number of independent observations of the phenomenon. In Section 4.1, we derive the analytical form of the final probability of detection. Then, we construct an optimization problem based on the derived formula. In Section 4.3, we show an algorithm which is able to select the optimal subgroup of agents from the agent pool.

4.1 Probability of detection after fusion

We start by finding the analytical form of the probability of detection after decision fusion. We define:

- P_D: the probability of detection after decision fusion
- $\Lambda(\mathbf{u})$: LR test given decisions from all agents
- $P_{Ei} \triangleq P_{Fi} = P_{Mi} = w(Q(\frac{a}{\sigma_i}))$: probability of error for agent i, $i = 1, ..., N$
- $P_{Di} \triangleq 1 - P_{Mi} = 1 - w(Q(\frac{a}{\sigma_i}))$: probability of detection for agent i, $i = 1, ..., N$

Here, (3.4) shows that:

$$\Lambda(\mathbf{u}) = \frac{P(u_1, u_2, ..., u_N|H = H_1)}{P(u_1, u_2, ..., u_N|H = H_0)} \underset{u_0=H_0}{\overset{u_0=H_1}{\gtrless}} 1. \tag{4.1}$$

Thus, we would like to find:

$$P_D = P(\Lambda(\mathbf{u}) > 1|H = H_1),$$

$$P_D = P(\log(\Lambda(\mathbf{u})) > 0|H = H_1). \tag{4.2}$$

Assuming the decisions of all agents are independent, we know:

$$\Lambda(\mathbf{u}) = \prod_{i=1}^{N} \frac{P(u_i|H = H_1)}{P(u_i|H = H_0)},$$

$$\log\left(\Lambda(\mathbf{u})\right) = \sum_{i=1}^{N} \log \frac{P(u_i|H = H_1)}{P(u_i|H = H_0)}. \tag{4.3}$$

Since the addition of independent RVs corresponds to the convolution of their PDFs, we have:

$$P(\log(\Lambda(\mathbf{u}))|H = H_1) = P(\log(\Lambda(u_1))|H = H_1) * \cdots * P(\log(\Lambda(u_N))|H = H_1).$$

Now, we are interested in finding:

$$P(\log(\Lambda(u_i))|H = H_1).$$

From (3.5), after slight modification, we have:

$$\log(\Lambda(u_i)) = u_i \log \frac{P_{Di}}{P_{Fi}} + (1 - u_i) \log \frac{1 - P_{Di}}{1 - P_{Fi}}. \tag{4.4}$$

This equation is essential to our problem. It means that $\log(\Lambda(u_i))$ can take two values given the incoming local agent decision. If $u_i = 0$, it outputs $\log(\frac{1-P_{Di}}{1-P_{Fi}})$ with probability $P_{Mi} = 1 - P_{Di}$ under hypothesis H_1 (missing); if $u_i = 1$, it outputs $\log(\frac{P_{Di}}{P_{Fi}})$ with probability P_{Di} under hypothesis H_1 (detection).

Thus, $\log(\Lambda(u_i))$ is a Bernoulli random variable, with P_{Di} as probability of success (detection), and $P_{Mi} = 1 - P_{Di}$ as probability of failure (missing).

And lastly, we have:

$$\begin{aligned} P_D &= P(\log(\Lambda(\mathbf{u})) > 0 | H = H_1) \\ &= \sum_{\Lambda(\mathbf{u})>1} P(\Lambda(\mathbf{u}) | H = H_1) \\ &= \sum_{\Lambda(\mathbf{u})>1} *(P(\log(\Lambda(u_i)) | H = H_1)), \end{aligned} \tag{4.5}$$

where $*$ stands for sequential convolution.

Since the convolution components are Bernoulli random variable, and the convolution of Bernoulli distributions is Poisson binomial distribution [5], the above expression for P_D can be re-written as:

$$P_D = \sum_{l=\lceil \frac{n}{2} \rceil}^{n} \sum_{A \in F_l} \prod_{i \in A} P_{Di} \prod_{j \in A^c} (1 - P_{Dj}). \tag{4.6}$$

4.2 Approximation

However, with the combinatorial optimization (agent selection) problem we proposed, the Poisson binomial distribution is extremely hard to optimize. A good approximation is required. We introduce the normal approximation of the Poisson binomial distribution based on CLT (central limit theorem).

$$F_X(x) = \mathrm{P}(X \leq x) \approx \Phi\left(\frac{x + 0.5 - \mu}{\sigma}\right), \tag{4.7}$$

with $\mu = \sum_i P_{Di}$ as the mean of Poisson binomial distribution, $\sigma = (\sum_i (1 - P_{Di}) P_{Di})^{\frac{1}{2}}$ as the variance of Poisson binomial distribution, and $\Phi(x)$ as the CDF of the standard normal distribution. Given a sufficiently large agent pool, the normal approximation should approximate the actual Poisson binomial distribution fairly well.

In order to simplify the problem, we limit the number of agents N to only odd cases. Thus, if all agents are selected, using normal approximation, the probability of detection at the fusion center becomes:

$$\begin{aligned} P_D &= \mathrm{P}\left(X \geq x\right) \\ &= \mathrm{P}\left(X \geq \frac{N+1}{2}\right) \\ &= 1 - \mathrm{P}\left(X \leq \frac{N-1}{2}\right) \\ &\approx 1 - \Phi\left(\frac{\frac{N-1}{2} + 0.5 - \mu}{\sigma}\right) \\ &= Q\left(\frac{\frac{N-1}{2} + 0.5 - \mu}{\sigma}\right). \end{aligned}$$

After simplification, we get:

$$P_D = Q\left(\frac{\frac{N}{2} - \sum_i P_{Di}}{(\sum_i (1 - P_{Di})P_{Di})^{\frac{1}{2}}}\right). \tag{4.8}$$

Now, we introduce the agent selection variable:

$$\mathbf{s} = [s_1, s_2, ..., s_N]^T \in \{0, 1\}^N.$$

We want to maximize:

$$P_D = Q\left(\frac{\frac{\sum_i s_i}{2} - \sum_i s_i P_{Di}}{(\sum_i s_i (1 - P_{Di})P_{Di})^{\frac{1}{2}}}\right).$$

Since Q function is a monotonic decreasing function, we essentially want to minimize over $\mathbf{s}$:

$$\frac{\frac{\sum_i s_i}{2} - \sum_i s_i P_{Di}}{(\sum_i s_i (1 - P_{Di})P_{Di})^{\frac{1}{2}}}.$$

In Section 3.4, we know that $0.5 < P_{Di} < 1$; the numerator is non-positive and the denominator is non-negative, thus, the objective can be replaced by minus its square:

$$\begin{aligned} &\min_{\mathbf{s}} - \frac{(\frac{1}{2}\sum_i s_i - \sum_i s_i P_{Di})^2}{\sum_i s_i (1 - P_{Di})P_{Di}}, \\ &\max_{\mathbf{s}} \frac{(\frac{1}{2}\sum_i s_i - \sum_i s_i P_{Di})^2}{\sum_i s_i (1 - P_{Di})P_{Di}}. \end{aligned} \tag{4.9}$$

Constraints:

$$\mathbf{s} = [s_1, s_2, ..., s_N]^T \in \{0,1\}^N,$$

$$\mathbf{P_D} = [P_{D_1}, P_{D_2}, ..., P_{D_N}]^T, 0.5 < P_{Di} < 1.$$

$\mathbf{P_D}$ is given.
Here, (4.9) is the objective function that we want to optimize. In general, one could end with keeping all agents if it is the best option.

4.3 Agent selection algorithm

In this section, we will find the optimal $\mathbf{s}$ selection vector using numerical technique.

Again, the objective is:

$$\max_{\mathbf{s}} \frac{(\frac{1}{2}\sum_i s_i - \sum_i s_i P_{Di})^2}{\sum_i s_i(1-P_{Di})P_{Di}},$$

$$\max_{\mathbf{s}} \frac{(\sum_i (P_{Di} - 0.5)s_i)^2}{\sum_i (1-P_{Di})P_{Di}s_i}. \tag{4.10}$$

We define:

$$a_i = P_{Di} - 0.5 > 0,$$

$$b_i = (1-P_{Di})P_{Di} > 0,$$

and the objective becomes:

$$\max_{\mathbf{s}} \frac{(\sum_i a_i s_i)^2}{\sum_i b_i s_i}. \tag{4.11}$$

Since $\mathbf{s} = [s_1, s_2, ..., s_N]^T \in \{0,1\}^N$, we essentially want to find the same subgroup from $\mathbf{a}$ and $\mathbf{b}$ that maximizes the objective; that is:

$$\text{select } i \text{ from } I \text{ such that} \frac{(\sum_{i\in I} a_i)^2}{\sum_{i\in I} b_i} \text{is maximized.}$$

This is the original problem we proposed.

Now, we observe that, since $0.5 < P_{Di} < 1$:
a_i is increasing with respect to P_{Di},

b_i is decreasing with respect to P_{Di}.

Thus, we have:

Lemma 4.3.1. *Selecting one from two choices of P_D, namely P_{Di} and P_{Dj}, the larger P_D should always be selected to maximize the objective.*

Proof. Suppose $P_{Di} > P_{Dj}$, then $a_i > a_j > 0$, $0 < b_i < b_j$, then:

$$\frac{a_i^2}{b_i} > \frac{a_j^2}{b_j}$$

always holds. □

Lemma 4.3.2. *Selecting a fixed group of size k from a group of size n, the P_Ds should be selected in a descending order until reaching the maximum capacity k to maximize the objective.*

Proof. Suppose a random subset of size k is chosen from the group. Based on Lemma 4.3.1, one should always replace the smaller P_Ds to larger P_Ds until no replacement can occur. The result we finally obtain is clearly selecting the P_Ds in a descending order until reaching the maximum capacity k; thus reaching the optimum. □

Theorem 4.3.3. *Selecting a random size group from a group of size n, the algorithm of selecting P_Ds running time can be reduced from $O(2^n)$ to $O(n)$.*

Proof. Without loss of generality, suppose $\mathbf{P_D}$ is in descending order; that is:

$$\mathbf{P_D} = [P_{D_1}, P_{D_2}, ..., P_{D_N}]^T, 0.5 < P_{D_N} \leq P_{D_{N-1}} \leq P_{D_{N-2}} \leq \cdots \leq P_{D_1} < 1.$$

Based on Lemma 4.3.2, if k is ranging from 1 to N, then only N times of comparison are needed to be performed, since for each case of k, we just directly pick the P_Ds in order. Thus, instead of $O(2^n)$, we have an $O(n)$ algorithm for agent selection; that is, selecting the largest from:

$$\frac{(\sum_{i=1}^{k} a_i)^2}{\sum_{i=1}^{k} b_i}, k = 1, 2, ..., N. \tag{4.12}$$

□

Now, we have a general algorithm for agent selection to maximize the probability of detection after decision fusion.

General Algorithm for Agent Selection:

Algorithm 1 Agent Selection

procedure
  A = encoded value
  $\mathbf{c} = [c_1, c_2, ..., c_N]^T \in [0.5, 1]^N \leftarrow$ reported *confidence/expertise*
  $\mathbf{P_D} = [P_{D_1}, P_{D_2}, ..., P_{D_N}]^T \leftarrow 1 - w(1 - \mathbf{c})$
  $\mathbf{P_D} \leftarrow sortInDescendingOrder(\mathbf{P_D})$
  $\mathbf{a} = [a_1, a_2, ..., a_N]^T \leftarrow \mathbf{P_D} - 0.5$
  $\mathbf{b} = [b_1, b_2, ..., b_N]^T \leftarrow (1 - \mathbf{P_D})\mathbf{P_D}$
  for $k = 1$ to N **do**
    $curr \leftarrow (\sum_{i=1}^{k} a_i)^2 / \sum_{i=1}^{k} b_i$
    keep max curr value, keep max index
  $\mathbf{agent} = [agent_1, agent_2, ..., agent_{maxindex}]^T \subseteq \mathbf{P_D}$

4.4 Simulation

In this section, we simulate the agent selection algorithm to test its correctness (whether the selected group gives the best detection accuracy). We are also interested in the effect from underconfidence/overconfidence. Thus, we designed three agent pools for the algorithm to run on as follows:

- Agent pool of size 100 that contains all underconfident agents. Based on Figure 3.3, underconfident agents should have confidence level $0.5 < P_{Di} < 0.633135$.
- Agent pool of size 100 that contains all overconfident agents. That is, all agents in this pool have confidence level $0.633135 < P_{Di} < 1$.
- Agent pool of size 200 that contains all levels of agents. Agents in this pool have confidence level $0.5 < P_{Di} < 1$. For better comparison purpose, we design it as the combination of the two agent pools above.

All agents are sampled from the given intervals evenly. Intuitively, if we need to pick a subgroup of agents to obtain the best accuracy, we would pick the agents who are close to the intersection point shown in Figure 3.3; that is, pick the agents who are neither too underconfident nor too overconfident (honest people). However, we surprisingly find out that the results we obtain do not match our initial intuition. The results are provided below:

- In total, 92 agents who have higher reported confidence level are selected from the underconfident agent pool.
- In total, all (100) agents are selected from the overconfident agent pool.
- In total, 124 agents who have higher reported confidence level are selected from the combined agent pool, including all overconfident agents and first 24 underconfident agents.

This result shows that, although overconfidence may affect people in team decision making, it does not mean that people who are overconfident are not trustworthy. Since their actual expertise level is still relatively high, they can still contribute a lot in team decision making.

Besides, the result of the mixed agent pool setting shows that we can definitely lower a lot of labor costs if we have a sufficiently large agent pool, which is normal in real-world settings.

CHAPTER 5

CONCLUSION

Overconfidence may affect people in team decision making. By establishing an end-to-end detection model and formulating the overconfidence effect, this thesis provides an analytical picture regarding people's team decision-making process under this particular psychological bias. This thesis also provides an efficient algorithm for selecting the optimal subgroup of people from an agent pool to perform this particular task.

However, there are still future improvements to be made for this topic. For example, the binary detection problem was considered in this thesis for the sake of simplicity. However, in a real-world setting, people often perform M-ary tasks. Thus, the detection model should be extended to more complex cases. Also, Chapter 4 constructs an unconstrained optimization problem. However, in a real-world setting, many constraints need to be considered, such as a given budget limitation. In this case, a more complex constrained optimization problem needs to be analyzed.

REFERENCES

[1] C. Sanchez and D. Dunning, "Overconfidence among beginners: Is a little learning a dangerous thing?" *Journal of Personality and Social Psychology*, vol. 114, no. 1, pp. 10–28, November 2018.

[2] P. K. Varshney, *Distributed Detection and Data Fusion.* Springer, 1997.

[3] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk and Uncertainty*, vol. 5, no. 4, pp. 297–323, October 1992.

[4] D. Prelec, "The probablity weighting function," *Econometrica*, no. 3, pp. 497–527, May 1998.

[5] S. D. Poisson, *Recherches sur la Probabilitè des jugements en matié criminelle et en matiére civile.* Bachelier, Paris, 1837.

[6] M. Fernandez and S. Williams, "Closed-form expression for the Poisson-binomial probability density function," *IEEE Transactions on Aerospace Electronic Systems*, vol. 46, no. 2, pp. 803–817, April 2010.