POLITECNICO MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Algorithmic Bayesian Persuasion under Uncertainty

Tesi di Laurea Magistrale in Computer Science Engineering - Ingegneria Informatica

Author: **Luca Tombesi**

Student ID: 10865393
Advisor: Prof. Alberto Marchesi
Co-advisors: Martino Bernasconi, Matteo Castiglioni

Academic Year: 2022-23

Abstract

This work explores the Bayesian persuasion framework, where a player, called *Sender*, by having an informational advantage towards another player, called *Receiver*, influences her by transmitting strategic signals. This framework, initially studied in its basic form, by Kamenica [26], has gained significant interest for its possible applications in a wide range of fields. Indeed there are present various studies, which analyze distinct variants of the original problem. Consequently, this work focus its attention on a different variant of the original problem. More precisely in this thesis we relax the constraint of the original problem for which the Sender knows exactly the Receiver's payoff. This factor of uncertainty brings the Receiver from the Sender perspective to have an unpredictable behaviour. The core of this study lies in finding a signaling scheme for the Sender that is robust enough to overcome the Receiver unpredictability. We propose an algorithm able to solve exactly this problem. The final phase of our research involves a comprehensive evaluation of this algorithm. We rigorously test it in various simulated environments to assess its practical performance. Key performance metrics include the algorithm's execution time and its efficiency in terms of achieving expected utility under different scenarios. This multifaceted analysis not only benchmarks the algorithm's effectiveness but also provides insights into potential areas for further refinement and application in real-world settings.

Keywords: Bayesian Persuasion, Uncertainty, Signaling Scheme.

Abstract in lingua italiana

Questo lavoro esplora il framework della persuasione bayesiana, in cui un giocatore, chiamato *Mittente*, avendo un vantaggio informativo nei confronti di un altro giocatore, chiamato *Ricevente*, lo influenza trasmettendo segnali strategici. Questo framework, inizialmente studiato nella sua forma base, da Kamenica [26], ha suscitato notevole interesse per le sue possibili applicazioni in una vasta gamma di settori. Infatti, sono presenti vari studi che analizzano diverse varianti del problema originale. Di conseguenza, questo lavoro focalizza la sua attenzione su una variante diversa del problema originale. Più precisamente, in questa tesi rilassiamo il vincolo del problema originale per cui il Mittente conosce esattamente il payoff del Ricevente. Questo fattore di incertezza porta il Ricevente, dal punto di vista del Mittente, a comportarsi in modo imprevedibile. Il nucleo di questo studio consiste nel trovare uno schema di segnalazione per il Mittente che sia abbastanza robusto da superare l'imprevedibilità del Ricevente. Proponiamo un algoritmo in grado di risolvere questo problema in modo esatto. La fase finale della nostra ricerca coinvolge una valutazione approfondita di questo algoritmo. Lo testiamo rigorosamente in vari ambienti simulati per valutarne le prestazioni pratiche. Le metriche chiave delle prestazioni includono il tempo di esecuzione dell'algoritmo e la sua efficienza nel raggiungere l'utilità attesa in diverse situazioni. Questa analisi sfaccettata non solo misura l'efficacia dell'algoritmo, ma fornisce anche spunti per eventuali ulteriori perfezionamenti e applicazioni in contesti reali.

Parole chiave: Persuasione Bayesiana, Incertezza, Schema di Segnali.

Contents

List of Figures

List of Tables

1 | Introduction

1.1. General overview

In recent years, the value of information, and consequently of data, has grown exponentially. Indeed not only companies but also people in general became aware of the incredible potential of the value that there is behind owning personal data and information. Owned data can be utilized in numerous ways to enhance wealth. For instance, in a business context, data are crucial for evaluating performance and understanding what strategies are effective. In other contexts having an *informational advantage*, namely having access to information that others lack, can be a significant asset. Consider the example of an investor and a financial adviser in the economic realm, such as the stock market. Here, informational advantage is often leveraged to persuade and maximize potential gains. *Persuasion* is the ability of inducing another player to perform an action that is the better for the one who is persuading. This specific form of persuasion, which capitalizes on an *informational advantage*, is known as *Bayesian persuasion*, and it is rising a lot in popularity.

The Bayesian Persuasion model is a game of two players, the first is called *Sender*, who wishes to persuade the other player called *Receiver*. Both players have a common knowledge called prior. However, the Sender has an informational edge, which is strategically utilized to influence the Receiver's decisions through signaling. The Receiver processes these signals and updates their beliefs according to the Bayesian rule, leading to actions that are intended to benefit both parties. Dughmi and Xu [18] affirm that since the first model of *Bayesian persuasion* was published by Kamenica [26] persuasion as a share of economic activity appears to be growing $-^{-}$ a more recent estimate places the figure at 30 percent. Consequently it is of real interest to study this category of problem in different scenarios by varying some aspects of the basic model. In this thesis we study the Bayesian Persuasion problem in the context where the Sender does not know the actual utilities of the Receiver but has an uncertainty (δ-*knowledge*) of them.

In this thesis we define a functioning model able to face this scenario using the *worst-*

case analysis approach. This approach avoid explicitly modelling the Receiver's irrational behavior and instead considers the worst possible Receiver's behavior for the Sender. This model is particularly relevant in cases where the Sender has only a rough estimate of the Receiver's utilities and, therefore, only a general idea of the actions the Receiver is likely to prefer. Consequently, our model aims to find a solution that is more robust to the Receiver's behavior, which may be less than optimal.

1.2. Related works

Various works have explored the Bayesian Persuasion framework. The first work is the one of Kamenica et al. [25], that defined this kind of problem and created a model to address it. After this work, the Bayesian Persuasion framework has been studied in its application in many different areas, such as online advertising [4, 20, 31], voting [1, 13, 15, 16], traffic routing [11, 15, 35], recommendation systems [30], security [34, 37], and product marketing [2, 12]. However one of the main lines of research emerged has been the one aiming at the construction of models able to guarantee the robustness of persuasion, when the Sender is unaware of the Receiver's goals. Our work follows this research field and consequently is related to various works like [3, 17, 24], which have, as a point of contact, the search for robust solutions in this type of framework or similar ones.

For example the work done by Dworczak [19] focuses in finding a robust signaling scheme. Indeed it challenges the case in which the Sender is uncertain about the exogenous sources of information of the Receiver, which means that the Sender may be concerned that his beliefs are wrong. For this reason they focus on finding a policy that is not optimal under the Sender conjecture but that protects him well in the case the conjectures are wrong. This scenario is modeled with the state of nature that may send an additional signal to the Receiver. The main result shown in this work identifies states that can not appear together in the support of any of the posterior beliefs induced by a robust solution. Indeed to obtain worst-case optimality such separation of states is both necessary and sufficient.

Another noteworthy work is the one done by Zu et al. [40]. In their work it is addressed the case in which neither the Sender nor the Receiver knows the distribution of the payoff relevant state. Indeed the Sender does not know the prior distribution and consequently he learns this distribution over time by observing the state realizations. Subsequently, the problem of learning the prior in Bayesian Persuasion was studied in [7] extending it to sequential games [6, 10, 21]. Persuasion was studied also in Markov Decision Processes in [22, 36] for myopic receiver and by Bernasconi et al. [9] for farsighted receivers.

This type of problem can find its immediate application in the recommender systems field, where the recommender, which is the Sender, has to suggest to the user, the Receiver, some type of item and does not know if it will be liked or not by the user. The solution proposed consists in an algorithm, which proceeds by mantaining at each time a set of candidate priors, based on the observed state realization in the past. The proposed solution includes an algorithm that maintains a set of candidate priors based on past state realizations. This algorithm aims to offset the ignorance of the prior distribution by focusing on robustness, achieving a sub-linear regret of $O(\sqrt{T \log T})$ under certain conditions.

In the Bayesian persuasion research field another work, which is related to this thesis is the work done by Castiglioni et al. [14] and subsequently extended by Bernasconi et al. [8]. In their work they relax the assumption of the Sender, who must know exactly the Receiver's utility function to compute an optimal signaling scheme. In order to perform such task, they study a repeated Bayesian persuasion problem where, at each round, the Receiver's type is adversarially chosen from a finite set of types. The goal of the study is to find a model able to reccomend a signaling scheme at each round, guaranteeing an expected utility to the Sender. The study is done in the *full information* scenario, where the Sender selects a signaling scheme and later observes the type of the best-responding Receiver, and the *partial information* scenario, where the Sender only observes the actions taken by the Receiver. They show the impossibility of finding a *no-regret* algorithm with polynomial per-round running time. However, by relaxing the running time constraint, they find that is possible to achieve a regret polynomial in the size of the problem instance and sub-linear in the number of rounds T under both full with a complexity of $\mathrm{O}(T^{-1/2})$ and in the partial feedback with a complexity of $\mathrm{O}(T^{-1/5})$).

Another work that discuss a similar problem as the one described in this thesis is done by Gan et al. [23] and is related to [27, 33, 39]. This work studies the Stackelberg equilibrium framework, where we have two players a *Leader* and a *Follower*. In this problem setting the Leader, who has a similar role of the Sender, has incomplete information on the utilities of the Follower, who is similar to the Receiver. Consequently the Follower, acts in a sub-optimal manner in respect to the Leader knowledge which brings to the Leader unpredictability of the Follower actions. Similarly to our case, in their work they analyze the problem using a worst-case scenario approach. Indeed they suppose that the Follower not only plays sub-optimally, but also picks the worst possible action for the Leader. So the Leader commits to a mixed strategy, to which the Follower responds by choosing an action that maximizes his expected utility. The solution found to their problem is NP-Hard. Indeed assuming P $\neq$ NP, it is not possible to find an approximate solution that is polynomial (FPTAS). However, they present a quasi-polynomial approximation scheme

(QPTAS).

1.3. Original Contributions

In this work, we begin with an in-depth analysis of the problem structure, adopting a worst-case scenario approach. The primary objective is to develop a robust solution, that effectively withstands scenarios where the Receiver engages in sub-optimal decision-making.

We start by presenting a negative result. Specifically, the Best Response set that is induced by the posteriors is not a convex set. This lack of convexity means that it's not feasible to find the optimal solution within polynomial time, thereby classifying the problem as NP-Hard. To address this, we define a convex set starting from the Best Response with the aim of finding a viable solution. Consequently, by using this newly defined set, we propose a solution and create an algorithm capable of identifying the optimal one for our problem setting. This algorithm works by determining all potential robust solutions and selecting the most advantageous one for the Sender.

Then we proceed to explain in depth how the algorithm operates, including its implementation and the functionalities of the utilized libraries. Additionally, we present the results of our experiments, which were designed to fully comprehend the algorithm's behavior. Indeed we analyzed the execution time to better comprehend how the NP-Hardness influences the algorithm and so its usability by evaluating all the execution times of the algorithm parts. Furthermore, we examine the impact of the δ parameter which represents the uncertainty of the Sender, in order to evaluate the robustness of the algorithm solution.

1.4. Thesis structure

The thesis is structured as follows:

- In chapter *2* we show theoretical basics in order to frame better and give context to our work. In particular we discuss key concepts as: The Bayesian Game, the Bayesian Persuasion Game, Computational complexity and Convexity;
- In chapter *3* we introduce and propose the problem of finding an optimal signaling scheme with the presence of uncertainty;
- In chapter *4* we proceed to explain in detail the functioning of our algorithm;

- In chapter *5* we show our experimentation done to test the algorithm, discussing the results obtained, regarding the time experimentation and the uncertainty impact in the problem;
- In chapter *6*, we draw the conclusions of this work and we propose new research direction that could enrich this type of setting.

2 | Preliminaries

In this section we will define all the components and models that are useful to understand the context of this work and so to facilitate its comprehension.

2.1. Bayesian games

In game theory a Bayesian game is a generalization of a complete-information game. In addition to the common knowledge of the classical complete-information game, in this setting the players could have private information. In such case is present some uncertainty for one or more players, which is captured by the notion of an epistemic type. An epistemic type describes the player's knowledge. An example of this uncertainty coming from private information could be a player, who does not know exactly the utilities of the opponents. Then the Bayesian game could be represented in the epistemic-form as a tuple(N,A,Θ,Ω,U) where:

- N={1,2,...,n} is the set of players;
- A={$A_1, A_2, ..., A_m$} is the set of actions of all the players and A_i={$a_1, a_2, ..., a_m$} is the set of player i's actions;
- $\Theta = \Theta_1 \times \Theta_2 \times ... \times \Theta_n$ is the set of all the players and $\Theta_i = \{\theta_{i,1}, \theta_{i,2}, .., \theta_{i,n}\}$ is the set of the types of player i;
- $\Omega : \Theta \longrightarrow \Delta(\Theta)$ returns the probability associated with each $(\theta_1, \theta_2, .., \theta_n)$ where $\theta_i \in \Theta_i$;
- U={$U_1, U_2, ..., U_n$} is the set of the utility functions of all the players $U_i : A_1 \times A_2 \times ... \times A_n \times \Theta \longrightarrow R$ is the utility function of player i.

In a Bayesian game the uncertainty about the opponent's utilities is captured by the concept of the type. Specifically, if a player *p* is uncertain about the payoffs of her opponent *p'*, we model it as if player *p* is uncertain over the different types of the other player, each with different payoffs values. Then the private information affects only the utilities. Every player maintain their beliefs in the utilities in the form of a probability distribution over the types, which is common knowledge and is called *prior*. The players are rational and so they maximize their utilities. This type of game is called Bayesian game, because after the players receive private information, they update their knowledge using the Bayes rule. Similar to complete-information games, it is presupposed that all the information mentioned is shared knowledge among the players [28]. Moreover, the assumption is made that each player p eventually becomes aware of their own type. This implies that an agent's strategy is a function of its type and so:

$$\theta_i \longrightarrow A_i \ \ , \ \ \forall \, p \; i, s_i$$

The Bayesian game can be divided in three phases :

- **ex-ante** phase
- **interim** phase
- **ex-post** phase

In the first phase, called ex-ante phase, each player is unaware of their own type or the types of other players. When analyzing a strategy during this phase, we employ the following expected utility function:

$$E_{\theta \sim F}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})]$$

In the second phase, called interim phase, each player is aware of their own type, but do not know the types of the other players. To find a strategy in this phase, we adopt the following expected utility function:

$$E_{\theta_{-i} \sim F_{-i}|\theta_i}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})]$$

Finally in the ex-post phase, each player is aware of their type and of the types of every other player. Then to find a strategy in this kind of phase we adopt the following utility function:

$$u_i(s_i, s_{-i}; \theta_i, \theta_{-i})$$

Then because there are three possible phases, we have three possible types of equilibria in a Bayesian game and consequently three types of strategies. The first type of equilibrium is called ex-ante Bayes-Nash equilibrium.

Definition 2.1. *A strategy profile $s = (s_i, s_{-i}) \in \ S$ is an ex-ante Bayes-Nash equilibrium if no player can increase their ex-ante expected utility by unilaterally changing their* *strategy:*

$$E_{\theta\sim F}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})] \geq E_{\theta\sim F}[u_i(s_i', s_{-i}; \theta_i, \theta_{-i})], \quad \forall i \in [n], \forall s_i' \in S_i$$

The second type of equilibrium is called interim Bayes-Nash equilibrium.

Definition 2.2. *A strategy profile $s = (s_i, s_{-i}) \in S$ is an interim Bayes-Nash equilibrium if no player can increase their interim expected utility by unilaterally changing their strategy:*

$$E_{\theta_{-i}\sim F_{-i}|\theta_i}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})] \geq E_{\theta_{-i}\sim F_{-i}|\theta_i}[u_i(s_i', s_{-i}; \theta_i, \theta_{-i})], \quad \forall i \in [n], \forall s_i' \in S_i$$

Consequently the third type of equilibrium is called ex-post Nash equilibrium and is defined as follows:

Definition 2.3. *A strategy profile $s = (s_i, s_{-i}) \in S$ is an ex-post Nash equilibrium if no player can increase their* ex-post *expected utility by unilaterally changing their strategy:*

$$u_i(s_i, s_{-i}; \theta_i, \theta_{-i}) \geq u_i(s_i', s_{-i}; \theta_i, \theta_{-i}), \quad \forall i \in [n], \forall \theta \in \Theta, \forall s_i' \in S_i$$

However *ex-ante* and *interim* equilibria are equivalent and so they are referred as Bayes-Nash equilibria. Obviously an interim equilibrium is always an ex-ante equilibrium, because the interim equilibrium condition is stronger than the ex-ante one. However an ex-ante equilibrium is an interim one because if we take the expected utility function **??** of player i and we restated it as:

$$s_i* \in \max_{s_i \in S_i} E_{\theta_i\sim F}[E_{\theta_{-i}\sim F_{-i}|\theta_i}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})]]$$

then by using the Jensen's inequality we obtain:

$$\max_{s_i \in S_i} E_{\theta_i\sim F}[E_{\theta_{-i}\sim F_{-i}|\theta_i}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})]] \leq E_{\theta_i\sim F}[\max_{s_i \in S_i} E_{\theta_{-i}\sim F_{-i}|\theta_i}[u_i(s_i, s_{-i}; \theta_i, \theta_{-i})]]$$

This implies that whenever the player i is optimizing in the ex-ante phase for all strategies $s_i' \in S_i$, the player i is also interim optimizing. Finally because the Bayes-Nash equilibria are Nash equilibria, this implies that Nash's theorem guarantees their existence.

2.2. Bayesian Battle of Sexes

We show an example to clarify the working of this kind of game. As an example we propose the so-called *Battle of Sexes*. This game was introduced in 1957 by Luce [29]. In this book the game is described as follows: Consider a scenario where a man and a woman intend to meet in the evening but face a dilemma between two events: a prize fight and a ballet. The man leans towards attending the prize fight, while the woman prefers the ballet. Both would rather attend the same event than different ones. In the absence of communication how they decide where to go?

From the classical setting we extend the game, inspired by [38], to study the Bayesian case of the Battle of Sexes case. Then we consider the scenario were the players can have two possible types. The first type indicates that the player is interested(I) in the other player, while the second type indicates that the player is not interested(U) in the other. Consequently by adding this kind of typification we obtain a Bayesian version of the Battle of Sexes game. Indeed we consider in our example the case were the woman has only one type, which is the first type, while the man can be of both types with equally probability. Consequently in the epistemic form we could have such a setting:

- N= {1,2};
- A= $\{A_1, A_2\}$ with A_1={B,S},A_2={B,S};
- Θ= $\{\Theta_1, \Theta_2\}$ with Θ_1=$\{\theta_{1,1}\}$,Θ_2=$\{\theta_{2,1}, \theta_{2,2}\}$;
- Ω= $\{\Omega_1, \Omega_2\}$

Ω_1={1 for $\theta_{1,1}$} and Ω_2={0.5 for $\theta_{2,1}$, 0.5 for $\theta_{2,2}$}. Then the utility functions are represented with the following bi-matrices:

$\theta_{2,1}$

$\theta_{1,1}$		S	B
	S	10,5	0.0
	B	0.0	5,10

$\theta_{2,2}$

$\theta_{1,1}$		S	B
	S	10,0	0,0
	B	0,0	5,10

Table 2.1: The payoff matrices based on the types of the two players

The first table represent the payoff values where the man is interested and this happens with a $pr(I) = \frac{1}{2}$, while the other table represent the payoff values where the man is uninterested, which also happens with a $pr(U) = \frac{1}{2}$. By using the ex-ante equation **??** defined in the previous section we can represent the game in the normal form, by computing all the possible expected utilities. For example the woman expected utility is:

$$E[u_w(S, SB)] = \sum_{\theta \in \Theta} pr(\theta) u_w(S(\theta_1), SB(\theta_2); \theta) = \frac{1}{2}(10) + \frac{1}{2}(0) = 5$$

While the man expected utility would be:

$$E[u_m(SB, S)] = \sum_{\theta \in \Theta} pr(\theta) u_m(SB(\theta_1), S(\theta_2); \theta) = \frac{1}{2}(5) + \frac{1}{2}(10) = \frac{15}{2}$$

Consequently by computing all the ex-ante expected utilities we obtain the following representation in the normal form game of the Bayesian Battle of Sexes:

	SS	**SB**	**BS**	**BB**
S	10, 5/2	5, 15/2	5, 0	0, 5
B	0, 5/2	5/2, 0	5/2, 15/2	5, 0

Table 2.2: The expected payoffs in Bayesian Battle of the Sexes

By analyzing the table representing the expected utilities of the game, it can be noted that there are no dominant strategies in this game, but there is still one pure-strategy Nash equilibrium identified by (S,SB), which gives to the women a payoff of 5 and to the man a payoff of $\frac{15}{2}$. However, except the pure-strategy equilibrium, we can find more equilibria, which are called mixed-strategy Nash equilibria. Indeed by using such kind of strategies, we consider the probability of our opponent selecting one strategy over the other and balance our payoff accordingly. Then let p be the probability that the woman plays S, let q_i be the probability that the man plays S, if the man is interested in the woman and let q_u be the probability that the man plays S, if the man is uninterested in the woman. If the woman plays the mixed strategy p the man will respond depending on its type. So if the man is interested in the woman, his payoff for playing S is:

$$5p + 0(1 - p) = 5p$$

While if he plays B is:

$$0p + 10(1 - p) = 10 - 10p$$

Given such payoffs we can find the strategy of the woman that makes the men indifferent between the two actions. Such probability p is $\frac{2}{3}$. Therefore we obtain the mixed strategy $(\frac{2}{3}, \frac{1}{3})$. While if the man is not interested on the woman we have the payoff of the man playing S that is:

$$0p + 5(1 - p) = 5(1 - p)$$

While if he plays B is:

$$10p + 0(1-p) = 10p$$

In the scenario were the man is uninterested, and the woman uses the previous strategy, he obtains a payoff of $\frac{5}{3}$ for playing S and of $\frac{20}{3}$ for playing B. Therefore the man payoff for playing B is strictly greater than the other when he his of the uninterested type and so $q_u = 0$. For this reason the woman payoff, when $q_u = 0$ is:

$$pr(I)[10q_i + 0(1-q_i)] + pr(U)[10q_u + 0(1-q_u)] = 5q_i$$

While her payoff for playing B, when $q_u = 0$ is:

$$pr(I)[0q_i + 5(1-q_i)] + pr(U)[0q_u + 5(1-q_u)] = \frac{5}{2}q_i + \frac{5}{2}$$

Then the value of q_i, which makes the woman indifferent between the two actions is $q_i = \frac{2}{3}$. Finally we obtain the following results:

- The woman plays S with probability $p = \frac{2}{3}$ and B with probability $1-p = \frac{1}{3}$
- The man when has the type $\theta_{2.1}$=I, plays S with probability $q_i = \frac{2}{3}$, and B with probability $1-q_i = \frac{1}{3}$
- The man when has the type $\theta_{2.2}$=U, plays S with probability $q_u = 0$, and B with probability $1-q_u = 1$

So at this equilibrium we have that the woman utility is $\frac{30}{9}$, while the man utility is $\frac{35}{9}$. Finally we can describe by the following table the joint distribution over the woman and the man strategies at this equilibrium:

	SS	**SB**	**BS**	**BB**
S	0	4/9	0	2/9
B	0	2/9	0	1/9

Table 2.3: Joint probabilities at the mixed strategy Nash equilibrium.

As we can see the Bayesian game is a dynamic game with Nature, represented by all the possible combinations of types, in which the first player to move is Nature. The standard case of the Bayesian game presented before, does not include any form of communication between players. Indeed every player acts without exchanging any information with the opponents. This type of games is also known as simultaneous-moves game, in which every player cannot receive any information before an outcome is reached. In the following sections we will extend the standard case by taking in analysis the Bayesian persuasion framework, which extend the standard Bayesian game by adding the possibility to one player, called *Sender*, to exchange some type of information called signal.

2.3. The Bayesian Persuasion framework

In this section we start by analyzing the Bayesian Persuasion framework. It was first introduced by Kamenica [26] and it is a game of two players, the first is called **Sender**, who wishes to persuade the other player called **Receiver**. The Receiver is considered, like in all sorts of Bayesian Game frameworks, as a rational Bayesian player.

In the simplest model of this type of framework we have that the Receiver has to choose one between multiple actions, which are associated with an a-priori unknown payoff for both players. The payoffs do not depend only on the action that is chosen but also on the realization of the state of nature belonging to a finite set of possible states of nature. Both players have the same prior probability distribution of the next realization of the state of nature.

The Sender however, unlike the Receiver, knows the realization of the next state of nature and after it is observed, the Sender sends a **signal** to the Receiver, who given the prior and the signal, updates her beliefs using the Bayesian rule computing the posterior, and chooses an action. So the Sender, by committing to a policy and so by sending a signal, tries to persuade the Receiver to take the most favorable action for him. This policy is called **signaling scheme** and it is a randomized mapping from states of nature to signals being sent to the Receiver.

In this part we discuss thoroughly the problem defining the model in a rigorous way by mathematical means, which is inspired by [18],[5]. The Receiver has to choose an action a from $[n] = 1, ...n$, with an a-priori-unknown payoff to each of the Sender and Receiver. The payoffs depend on an unknown state of nature θ, that is drawn from a set of potential states of nature Θ. We assume that the set Θ is finite and has a dimension of m. So the Sender and Receiver payoffs are functions $u_r, u_s : \Theta \times [n] \longrightarrow R$. Then $u_s(\theta) \in R^n$ denotes the Sender's payoff vector as a function of the state of nature and consequently $u_s(\theta, a)$ indicates the payoff of the Sender given the state of nature and the action that the Receiver chose. Consequently $u_r(\theta) \in R^n$ indicates the Receiver's payoff vector as a function of the state of nature and so $u_r(\theta, a)$ denotes the payoff of the Receiver given the state of nature and the action chosen by her. The state of nature, that is a-priori unknown to the Receiver, is drawn from a prior distribution μ supported on Θ.

The Sender commits to a signaling scheme called $\phi \in \Phi^m$, where Φ^m is the space of all the signaling schemes, mapping randomly states of nature Θ to a family of signals Σ, so formally we will have $\phi : \theta \longrightarrow \Delta_\Sigma$, where Σ is a finite set of signals. Given $\theta \in \Theta$, we use $\phi(\theta)$ to signify the random signal selected when the state of nature is θ. Indeed, we use $\phi(\theta, \sigma)$ to indicate the probability of selecting the signal σ given a state of nature θ. So we can say that an algorithm implements a signaling scheme ϕ, if it takes as input a state of nature θ, and samples the random variable $\phi(\theta)$ (See Figure 2.1 for an illustration).

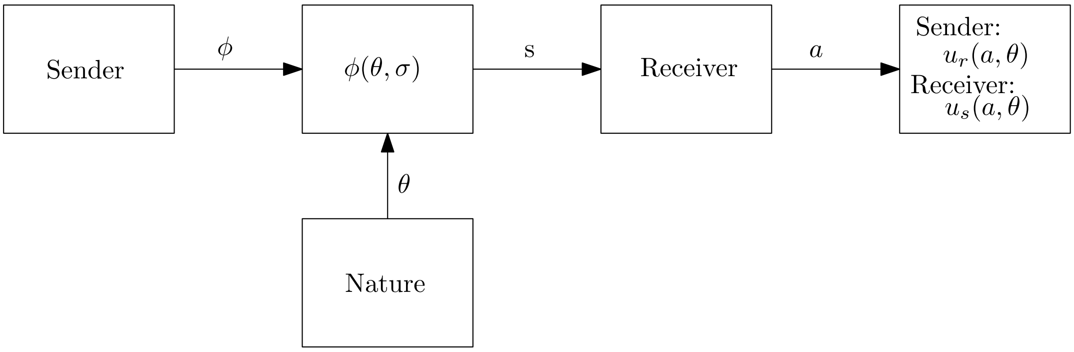


Figure 2.1: The Bayesian persuasion framework

Given a signaling scheme ϕ with signals Σ, each signal $\sigma \in \Sigma$ is realized with probability:

$$x_s = \sum_{\theta \in \Theta} \mu_\theta \phi(\theta, \sigma).$$

Then, given the signal σ the expected payoff of the Receiver are described by the vector $u_r(\sigma) = \frac{1}{x_r} \sum_{\theta \in \Theta} \mu_\theta \phi_{\theta,\sigma} u_r(\theta)$. Consequently for the Sender we will have the expected payoffs described by the vector $u_s(\sigma) = \frac{1}{x_s} \sum_{\theta \in \Theta} \mu_\theta \phi_{\theta,\sigma} u_s(\theta)$.

After the signal σ is sent by the Sender, the Receiver observes it and performs a rational update of her beliefs using the Bayes rule. We define $\Xi := \Delta_\Theta$ as the set of Receiver's posterior beliefs over the states of nature Θ, then the Bayesian update infers a posterior belief $\xi \in \Xi$ over the states of nature such that the component of ξ corresponding to state of nature $\theta \in \Theta$ is:

$$\xi_\theta = \frac{\mu_\theta \phi(\theta, \sigma)}{\sum_{\theta' \in \Theta} \mu_{\theta'} \phi_{\theta'}} \tag{2.1}$$

The Receiver, after computing ξ, selects the action a_i such that $a_i(\sigma) \in \max u_{r,i}(\sigma)$, which will induce to the Receiver an utility of $\max_i u_{r,i}(\sigma)$, which will induce to the Sender the utility $u_{s,i}(\sigma)$.

The Sender objective is to compute a signaling scheme ϕ such that will maximize his expected utility obtaining then an optimal signaling scheme ϕ^*. In the classical setting we can adopt a simple revelation-principle style argument that shows that an optimal signaling scheme need not use more than n signals, with one recommending each action. Such a direct scheme ϕ has signals $\Sigma = \sigma_1, ..., \sigma_n$, and satisfies

$$u_{r,i}(\sigma_i) \geq u_{r,j}(\sigma_i), \ \ \forall \ \ i, j \in [n].$$

The signals in order to be followed by the Receiver, need to be persuasive. Consequently We think of σ_i as a signal recommending action i, and we introduce the requirement $u_{r,i}(\sigma_i) \geq \max_j u_{r,j}(\sigma_i)$ as an incentive-compatibility (IC) constraint on our signaling scheme. More formally this constraint is defined as:

$$\sum_{\theta\in\Theta} \mu_\theta \phi(\theta, \sigma_i) u_{r,i} \geq \sum_{\theta\in\Theta} \mu_\theta \phi(\theta, \sigma_i) u_{r,j}, \ \ for \ \ i, j \in [n] \tag{2.2}$$

2.4. Defining the standard problem

2.5. Analyzing the problem in the posterior space

In this section we will focus in the relationship between the signaling scheme and the posterior space, representing the signaling schemes as convex combinations of posterior beliefs, inspired by [14].

Given a signaling scheme ϕ each signal $\sigma \in \Sigma$ induces a posterior probability $\xi_\sigma \in \Xi$, whose components are defined in equation (2.1). Because every signal σ has a probability $\phi(\theta, \sigma)$ of being drawn when the next realization is θ, we have that the signaling scheme ϕ induces a probability distribution over the posteriors beliefs. This posterior distribution is denoted by $x \in \Delta_x$. We say that a signaling scheme $\phi : \theta \longrightarrow \Delta_\Sigma$ induces $x \in \Delta_x$ if, for every $\xi_\sigma \in \Xi$, the component of x corresponding to ξ is defined as:

$$x_\xi := \sum_{\sigma\in S:\xi_\sigma=\xi} \sum_{\theta\in\Theta} \mu_\theta \phi(\theta, \sigma)$$

Then if the signaling scheme ϕ induces x, x_ξ represent the probability that ϕ induces the posterior probability $\xi \in \Xi$. We define the set of posteriors $supp(x) := \xi \in \Xi, x_\xi > 0$ that can be induced by ϕ with strictly positive probability x_ξ. We can affirm that the probability distribution over the set of posteriors is consistent if:

$$\sum_{\xi\in supp(x)} x_\xi \xi_\theta = \mu_\theta, \ \forall \ \ \theta \in \Theta$$

Finally given a feasible probability distribution over the signaling scheme x, the set of posteriors $supp(x)$ and the prior distribution μ we have that:

$$\phi(\theta, \sigma) = \frac{x_\xi \xi_\theta}{\mu_\theta}, \ \forall \ \ \theta \in \Theta$$

Consequently, given the posterior ξ induced by the Sender signal σ, the Receiver will chose the action, that maximizes the expected utility. Then we can define the *best response set* denoted as $BRset$. This set can be interpreted as the function $BR := \xi \longrightarrow [n]$ and can be defined as:

$$BR(\xi) = \max_{a \in A} \sum_{\theta \in \Theta} \xi_\theta u_r(a, \theta) \tag{2.3}$$

When given a posterior ξ if there are multiple actions that maximize the Receiver expected utility, we assume that the Receiver breaks ties in favor of the Sender and so chooses the action $a \in \max_{a \in [n]} \sum_{\theta \in \Theta} \xi_\theta u_r(a, \theta)$.

2.6. The judge and persecutor example

To better understand how the Bayesian persuasion framework works we consider a simple example inspired by Kamenica [26]. In their work they bring as an example the case in which the Sender is a prosecutor, the Receiver is a judge, and the state of nature is the guilt or innocence of a defendant.

The Receiver (judge) has two actions, conviction and acquittal, and wishes to maximize the probability of rendering the correct verdict. On the other hand, the Sender (prosecutor) is interested in maximizing the probability of conviction. There are two states of the world: the defendant is either guilty or innocent. The judge gets utility 1 for choosing the correct action (convict when guilty and acquit when innocent) and utility 0 for choosing the wrong action (convict when innocent and acquit when guilty). While the prosecutor gets utility 1 if the judge convicts the defendant and 0 otherwise. The prosecutor and the judge share a prior belief of the next state being guilt of $\mu_g = 0.3$ and consequently a prior belief of the next state being innocent of $\mu_i = 0.7$. The prosecutor being the Sender gets to observe the realized state of nature, so gets to observe if the defendant is guilty or innocent. Giving this informational advantage the prosecutor can exploit it by sending to the judge a signal from the set $S = \{s_1, s_2\}$.

The problem setting can be then summarized by the following tables:

As they demonstrate, it is possible to create scenarios in which the optimal signaling scheme for the Sender provides noisy partial information regarding the guilt or innocence

		State G ($\mu_G = .3$)		State I ($\mu_G = .7$)	
$\mathcal{A}$	A	0	0	0	1
	C	1	1	1	0

(a)

		Realized state State G	State I
$\mathcal{S}$	s_1	0	4/7
	s_2	1	3/7

(b)

		State of nature State G	State I	w*
supp(w*)	ξ_1	0	1	2/5
	ξ_2	1/2	1/2	3/5

(c)

Table 2.4: (a) u_s and u_r payoff matrices; (b) Optimal signaling scheme ϕ^*; (c) Posteriors generated by the optimal signaling scheme ϕ^*.

of the defendant. For instance, if the defendant is guilty with a probability of 0.3, the prosecutor's best strategy is to declare 'guilt' whenever the defendant is truly guilty, and also to claim 'guilt' just under half the time when the defendant is innocent. With this signaling scheme represented in the tables 2.4 it will induce the posterior ξ_1 and ξ_2 with a probability of $x_1 = 2/5$ and $x_2 = 3/5$. Then the Sender expected utility will be:

$$E_{u_s} = \sum_{\xi \in supp(x)} x_\xi u_s(\xi) = \frac{2}{5}(\sum_{\theta \in \Theta} \xi_1 u_s(\theta)) + \frac{3}{5}(\sum_{\theta \in \Theta} \xi_2 u_s(\theta)) = \frac{2}{5}(0) + \frac{3}{5}(1) = \frac{3}{5} = 0.6$$

Consequently, the defendant will be convicted whenever the prosecutor asserts 'guilt' (with a probability just under 0.7), assuming that the judge is fully aware of the prosecutor's signaling scheme. It's worth noting that it's not in the prosecutor's best interest to unconditionally claim 'guilt,' as a rational judge who is aware of such a policy would assign no significance to such a signal. Such a judge would render a verdict based solely on prior beliefs, which, in this case, would consistently result in acquittal.

2.7. Technical preliminaries

In this section we show the mathematical preliminaries that can be useful to better understand the study done in this work. Indeed we disclose the notion of convexity, because it can be handy to better understand the analysis that has been done of the problem and also the computational complexity, which can be useful to understand better the meaning of the result found in this thesis.

2.7.1. Convexity

In this part we show the various definitions regarding the property of convexity regarding a set. Indeed a set can be:

- **Convex**;
- **Strictly convex**;
- **Non convex**.

Definition 2.4. *A set of points S is convex if, $\forall\ x, y \in S$ and $\forall\ \alpha \in [0,1]$, the point z originated as the convex combination $z = \alpha x + (1-\alpha)y$ belongs also to S.*

Then a set of points S is defined as convex if for every pair of points within the set S, the line segment connecting those points is also entirely within the set S. So a convex set or region is a subset that, when intersected with any line, forms a single line segment. The boundary of a convex set is always a convex curve. The intersection of all the convex sets that contain a given subset A of Euclidean space is called the convex hull of A. It is the smallest convex set containing A. A more close type of convex set can is the strictly convex set, which has the following definition:

Definition 2.5. *A set S is strictly convex if every point on the line segment connecting x and y other than the endpoints is inside the* topological interior *of S.*

The *topological interior* of a subset C of a topological space X is the union of all subsets of S that are open in X. A point that is in the interior of C is an interior point of X. Consequently a closed convex subset is strictly convex if and only if every one of its boundary points is an extreme point. Finally, if a set does not satisfy the first two definitions, is called a non convex set. Often a polygon originated by a non convex set of points can be called concave.

The importance of a set being convex is the fact that it can be implemented the convex optimization. The convex optimization is the procedure that permits to maximize or minimize a convex function over a convex set. Indeed many classes of convex optimization admits polynomial-time algorithm solutions. Consequently if the set that need to be optimize and so if the set where the solution lies on is convex ti means that the problem can be solved in polynomial time and so it is not an NP-Hard problem.

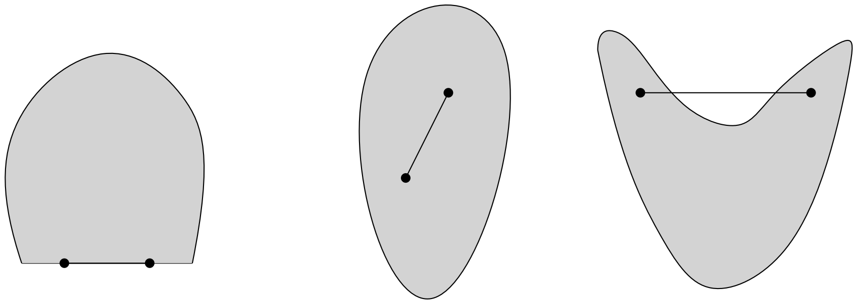


Figure 2.2: Convexity of a set of points.

2.7.2. Computational complexity

In this part we present some theoretical basis, inspired by [32], in order to better understand how computational complexity works, its role and its meaning.

Computational complexity theory is a computer science field that focuses on understanding the inherent difficulty of solving computational problems. It involves classifying problems based on the resources required to solve them, such as time, memory, or other computational resources.

Computational complexity of a problem is measured within two possible dimensions; The **space** complexity and the **time** complexity. The space complexity measures the amount of space that is allocated by the memory to solve the problem, while the time complexity measures the number of computational steps needed to solve the problem. The complexity of an algorithm is often defined with a dependency over the input size. Indeed often the input size determines the number of computational steps that the algorithm needs to perform. Consequently, in order to obtain a valid esteem of the computational complexity of a problem we rely on an asymptotic evaluation giving an upper bound measure of the time complexity.

Definition 2.6. *The function $f(n)$ is asymptotically upper-bounded by $g(n)$, we write it as $f(n) = O(g(n))$, if there exist two positive constants c and $n0$ such that:*

$$0 \leq f(n) \leq cg(n), \quad \forall n \geq n0$$

We can distinguish between the two main complexity classes: the **polynomial** time class and the **exponential** time class. The polynomial time class can be defined as:

Definition 2.7. *An algorithm belongs to the polynomial time class if for some $k > 0$, its* *running time on inputs of size n is $O(n^k)$.*

While we can define the exponential time class as:

Definition 2.8. *An algorithm belongs to the exponential time class if its running time on inputs of size n is of $O(2^n)$.*

Indeed any problem can be classified based on its time complexity, which classifies the problem to specific classes. Problems that are usually analyzed in such manner are **decision** problems and **functional** problem. A decision problem is a computational problem where the answer is either "yes" or "no" based on a set of inputs. In other words, the task is to decide whether a particular property holds for the given inputs. While a functional problem involves computing a function rather than making a yes/no decision. Indeed instead of asking whether a certain property holds, functional problems ask for the value of a particular function given certain inputs.

For what regards the decision problem category we can identify three major classes:

- **P** class, which identifies the class of decision problems that can be solved by a polynomial time algorithm.
- **NP** class, which identifies all the decision problems such that for every "yes"-instance there is a proof which allows to verify, in polynomial time, that the instance really admits a "yes" answer.
- **Co-NP** class, which identifies all the decision problems whose complements are in NP.

While the functional problem class has only two major classes. This is because it is a more straightforward type of problem where given an input inevitably you obtain an output. The classes are:

- **FP** class, which identifies the class of functional problems that can be solved in polynomial time.
- **FNP** class, which identifies the class of functional problems that admit a polynomial time algorithm able to verify given a problem instance I and a solution y, whether y is a solution of I.

For what regards the NP type problems there are some distinctions to be made. Indeed there can be identified principally in **NP-Hard** problems and **NP-complete** problems. In order to do that we have to first introduce the concept of reduction of a problem.

Definition 2.9. *A problem π can be reduced in polynomial time to another problem π', written as $\pi \leq p\pi'$, if there exists a transformation that allows to build an instance I' of π' from an instance I of π in polynomial time, and such that a solution of I can be derived, in polynomial time, from one of I'.*

Then for the reduction property if a problem π is reducible to another problem π' in polynomial time, then π' is at least as complex as π. Given the definition of reduction we can introduce the notion of NP-hard problems and NP-complete problems.

Definition 2.10. *A problem π is NP-Hard if every problem $\pi' \in NP$ is such that $\pi' \leq P\pi$.*

Definition 2.11. *A problem π is NP-Complete if it is in NP and it is NP-Hard.*

Finally another fundamental aspect in the computational complexity theory is the **P vs NP** problem. Indeed in the computational complexity field there is still the open question whether $P = NP$ or $P \neq NP$. If the first scenario would be verified then it would mean that every problem, where a potential solution can be verified quickly, could also be solved quickly. Essentially, every "easy to check" problem would be also "easy to solve". However, if the second scenario would be verified then it would mean that there are problems, where even if a potential solution could be verified quickly, the process of finding the solution itself would be inherently more difficult. In this case, it would exist the case where "easy to check" problems are not "easy to solve." In order to demonstrate that $P = NP$, it would be necessary to prove that there exists a polynomial algorithm for a NP-Complete problem. The solution to this dilemma still exists because it is still not proven that $P \neq NP$. This because it is complex to demonstrate that it does not exist a polynomial algorithm for a NP-Complete problem and so that $P \subset NP$ and then that some problem in NP cannot be solved in polynomial time.

3 | The delta Bayesian persuasion problem

In this thesis we address the Bayesian persuasion problem from a different perspective. Starting from the standard case, we take in analysis the problem of finding an optimal signaling scheme ϕ^* where the Sender does not know perfectly the u_r payoff matrix. Consequently from the Sender perspective the Receiver plays with an unpredictable manner. From the ideal situation, where the Receiver picks always an optimal response without any error, here we have that the Receiver may pick sub-optimal responses.

In order to represent this concept, we introduce a parameter δ, which indicates the uncertainty on the knowledge of the Sender. Consequently the Receiver could play committing a small error $\delta > 0$ from the Sender perspective. For this reason we have to define a new best response set of the Receiver, when the Receiver decides to play with δ-*optimality*. Thus δ-*optimality* is the case where the Receiver plays the action that from the Sender observation is not the optimal one but it is sub-optimal in a quantity of δ. The new Best Response set that we define is an extension of the concept of BR set, and so given a δ value and a posterior probability ξ, we obtain a subset of [n], called $BR_\delta(\xi)$ that is defined as follows:

$$BR_\delta\left(\xi\right) = \left\{ a' \in [n] : \quad \sum_{\theta\in\Theta} \xi_\theta u_r\left(\theta, a'\right) > \max_{a\in[n]} \sum_{\theta\in\Theta} \xi_\theta u_r\left(\theta, a\right) - \delta \right\} \tag{3.1}$$

This new set includes not only the action that we perceive as the one with maximum utility for the Receiver, but also it includes all the actions which have a Receiver utility value which differs at most from the maximum utility of a quantity of δ.

Our analysis aims at minimizing the Receiver sub-optimality by finding a robust solution. In order to do that we define a new set that starting from the $BR_\delta(\xi)$ set finds the action that is worst for the Sender. Namely, by using this new set we want to find the action a that minimizes the Sender utility in order to find the minimum utility that the Sender will receive if the Receiver plays with δ-optimality. Then given the $BR_\delta\left(\xi\right)$ set, we define the function $B_\delta(\xi)$, that given as input the posterior probability returns the action a, that corresponds to the action from the $BR_\delta(\xi)$ set with the minimum utility for the Sender:

$$B_\delta(\xi) = \min_{a \in BR_\delta} \sum_{\theta \in \Theta} \xi(\theta) u_s(a, \theta) \tag{3.2}$$

Consequently by the above equation we obtain the inverse function $\xi = B_\delta^{-1}(a)$. This function is used to associate to the posterior $\xi \in \Xi$ the action returned by the B_δ set. This function will be used to study the convexity of the set which include the solution that we want to find.

In this problem setting the Sender objective is to find a robust signaling scheme ϕ. The signaling scheme needs to induce the BR_δ set, which maximizes the minimum sender utility. Indeed this solution guarantees to the Sender the maximum expected utility achievable when the Receiver plays in the worst possible manner for him.

3.1. Analysis on the convexity of the $BR_\delta(\xi)$ set

In this section we present our negative result. We prove that the $BR_\delta(\xi)$ set is not convex. Then we provide an alternative set, starting from the original one, that assures the convexity. Namely we show that the function $B_\delta(\xi)$ is not convex as a consequence of the non convexity of the $BR_\delta(\xi)$ set. Then after showing that the standard definition of the $BR_\delta(\xi)$ set bring to a non convex set, we introduce the concept of a^* regions to re-define the $BR_\delta(\xi)$ set around them.

Theorem 3.1. *The function $\xi \mapsto B_\delta(\xi)$ is not convex.*

Proof. To prove the statement, we have to demonstrate that $\forall a$, $\forall \xi', \xi'' \in B_\delta^{-1}(a)$ and $\forall \xi$ with $\xi = \alpha\xi' + (1-\alpha)\xi''$, where $\alpha \in (0,1)$, then $B_\delta(\xi) \neq a$. We start by analyzing the $BR_\delta(\xi)$ set given the $BR_\delta(\xi')$ and the $BR_\delta(\xi'')$ sets. In fact two posteriors can generate two different BR_δ sets while satisfying the condition that both include the action a which is the one with the minimum utility for the Sender. In the case in which $BR_\delta(\xi') \neq BR_\delta(\xi'')$ we can assert that the convexity property does not hold. This happens for example if we take a setting where we have two actions a_1 and a_2, a $\delta = 1$ and two possibles state of natures θ_1 and θ_2, with the Sender's and Receiver's payoffs shown in the Table(3.1). If we take as posteriors $\xi' = (0.7, 0.3)$ and $\xi'' = (0.2, 0.8)$ we obtain the following BR_δ sets: $BR_\delta(\xi_1) = (a_2, a_1)$ and $BR_\delta(\xi'') = (a_1)$. Both the posteriors induce as the action with minimum utility a_1. We choose as our $\alpha = 0.2$ to construct the posterior ξ as linear combination of the former two, which produces the following set $BR_\delta(\xi) = (a_2, a_1)$. In

	θ_1	θ_2
a_1	5,3	2,1
a_2	3,2	4,4

Table 3.1: Example of non convexity of $B_\delta(\xi)$

this case given the utilities for the Sender, the action with the minimum utility is a_2 and not a_1 as for ξ' and ξ''.

The following theorem prove that the $BR_\delta(\xi)$ set is not convex.

Theorem 3.2. *The set $BR_\delta(\xi)$ is not convex.*

Proof. Similarly to Theorem(3.1) we want to demonstrate that $\forall \xi', \xi''$ where $BR_\delta(\xi') = BR_\delta(\xi'')$, $\forall \xi$ with $\xi = \alpha\xi' + (1-\alpha)\xi''$, where $\alpha \in (0,1)$, then $BR_\delta(\xi) \neq BR_\delta(\xi') = BR_\delta(\xi'')$. This can be proven by a different example. We consider the case in which we have: $\delta = 1$; Three states of nature θ_1, θ_2 and θ_3; Three actions a_1, a_2 and a_3; Two posteriors ξ' and ξ". ξ' is (0.4,0.3,0.3) and ξ" is (0.3,0.4,0.3). The utilities for the Sender and the Receiver are written in the Table(3.2).

	θ_1	θ_2	θ_3
a_1	5,3	5,3	0,1
a_2	3,2	5,3	0,4
a_3	0,0	0,0	5,0

Table 3.2: Example of non convexity of the $BR_\delta(\xi)$ set

In this case we have $BR_\delta(\xi') = (a_1, a_2)$ and $BR_\delta(\xi'') = (a_1, a_2)$. By picking $\alpha = 0.5$ we have that $\xi = \alpha\xi' + (1-\alpha)\xi" = (0.35, 0.35, 0.3)$ and so the $BR_\delta(\xi)$ set is (a_1,a_2,a_3), which causes the non convexity.

3.2. An algorithm for the optimal signaling scheme ϕ^*

In this section we show that always exists an optimal signaling (ϕ^*) scheme known a value of $\delta > 0$, a prior μ and the utilities of the Sender and the Receiver. As said before the signaling scheme induces a probability distribution of the posteriors ξ. Then the optimal signaling scheme ϕ^* induces the optimal posterior ξ^*. For this reason we define the $BR_\delta(\phi)$ set, which represents the set of actions that are δ-optimal given a signaling scheme ϕ:

$$BR_\delta(\phi) = \left\{ a' \in [n] : \quad \sum_{\theta \in \Theta} \mu_\theta \phi_\theta u_r(\theta, a') > \max_{a \in [n]} \sum_{\theta \in \Theta} \mu_\theta \phi_\theta u_r(\theta, a) - \delta \right\}$$

Consequently the Sender objective function can be expressed by the following equation, where $u_s(\phi, a)$ indicates the sender utility induced by the signaling scheme ϕ and the action a:

$$\phi^* = \max_{\phi \in \Phi} \min_{a \in BR_\delta(\phi)} u_s(\phi, a)$$

Theorem 3.3. *An optimal signaling scheme ϕ^* exists in every game for any $\delta > 0$ and can be computed in $O(2^n poly(m,n))$*

Proof. Given that the set $BR_\delta(\xi)$ is not convex, by the demonstration 3.2, we create a convex set, starting from the original definition of the $BR_\delta(\xi)$ set, that satisfy specific properties. Because we are search the optimal signaling scheme ϕ^* we define this convex set depending on the signaling scheme variable ϕ. This convex set is defined as the intersection of convex sets. Namely we start from the Sender strategy space and we divide it in convex sub-regions. Each sub-region, called $a*$ region is characterized by a set of actions A and the action with the maximum Receiver utility called $a^* \in A$. Then for any possible set of action A$\in 2^n$ and any $a^* \in A$:

$$\chi_{(A,a^*)} = \{\phi \in \Phi^m | \ BR_\delta(\phi) = A, \ u_r(\phi, a^*) > u_r(\phi, a) \ \forall a \in A\} \tag{3.3}$$

We have that $U_{a^* \in A, A \in 2^n} \chi_{(A,a^*)} = \Phi^m$. Indeed any $\phi \in \Phi^m$ must induce some Receiver δ-optimal response set A and an optimal Receiver response a^*. Then we can affirm that $\chi_{(A,a^*)}$ is a convex set, because it is a polytope. Indeed it is described by linear constraints, which are linear in the variable ϕ, where $\phi = (\phi_1, ..., \phi_m)$. The constraints are the following:

(C1) $\sum_\theta \mu_\theta \phi_\theta u_\theta(a^*) \geq \sum_\theta \mu_\theta \phi_\theta u_\theta(a') + \delta$, $a^* \in A$, $\forall a' \notin A$

(C2) $\sum_\theta \mu_\theta \phi_\theta u_\theta(a^*) \geq \sum_\theta \mu_\theta \phi_\theta u_\theta(a)$, $\forall a \in A$, $a^* \in A$

(C3) $\sum_\theta \mu_\theta \phi_\theta u_\theta(a^*) \leq \sum_\theta \mu_\theta \phi_\theta u_\theta(a) - \delta$, $\forall a \in A$

The constraints C1 and C2 derive directly from the definition of $BR_\delta(\phi) = A$, while constraint C3 is derived by the other part of the definition of $\chi_{(A,a^*)}$. The same convex polytope can be defined in the posterior space. The linear constraints (3.2) become:

(P1) $\sum_\theta \xi_\theta u_\theta(a^*) \geq \sum_\theta \xi_\theta u_\theta(a') + \delta$, $a^* \in A$, $\forall a' \notin A$

(P2) $\sum_{\theta} \xi_{\theta} u_{\theta}(a*) \geq \sum_{\theta} \xi_{\theta} u_{\theta}(a)$, $\forall a \in A$, $a^* \in A$

(P3) $\sum_{\theta} \xi_{\theta} u_{\theta}(a*) \leq \sum_{\theta} \xi_{\theta} u_{\theta}(a) + \delta$, $\forall a \in A$

The constraints are linear in the variable $\xi \in \Delta^m$, where $\xi = (\xi_1, \ldots, \xi_m)$. To find the optimal Sender signaling scheme ϕ^*, we use Algorithm 3.1 to solve multiple linear programs. The algorithm correctness relies on the fact that it always exists starting from any signaling scheme $\phi \in \Phi^m$ a set A and an action a^*. The algorithm for any possible sub-region identified by the tuple (A, a^*) enumerates all possible vertices of the polytope in the posterior space, breaking ties in favour of the Sender. Then it finds the optimal signaling scheme for the subset of actions A called ϕ_A. This is done by using a Linear Program, which finds a feasible probability distribution, that represents the probability of the signaling scheme to induce the posteriors, which maximizes the Sender utility in the vertices. Finally the algorithm picks the optimal signaling scheme ϕ^* as the one that brings the maximum Sender utility between all the signaling schemes found. Algorithm (3.1) always outputs a valid signaling scheme ϕ^*. Now it is important to understand if the output signaling scheme ϕ^* brings to the Sender at least the utility of the action a of the B_δ set(3.2). Since $U_{a*\in A, A\in 2^n} \chi_{(A,a^*)} = \Phi^m$ and the LP finds the feasible signaling scheme, that maximizes the minimum utility of the Sender, then the utility is at least the one of the action a of the B_δ set. Also the Sender utility, derived by the optimal signaling scheme ϕ^*, is at least the optimal objective of LP$(A*, a^*)$. This is true because, $a* \in A*$ and the LP maximizes the Leader utility for the sub-regions identified by the subset $A*$.

3.3. Algorithm complexity

The algorithm (3.1) to find the optimal solution ϕ^* has to compute all the possible robust signaling schemes and then has to pick the optimal one. In order to find a feasible robust signaling scheme, given a possible $BR_\delta(\xi)$ set called in the proof A, the algorithm has to iterate for every action $a \in A$. In each iteration the algorithm computes all the vertices creating the sub-regions such that the action a is the maximum. The algorithm computes all the vertices with a time complexity of $O(n^3)$. After the algorithm finds all the possible vertices it solves an LP in order to find a feasible probability distribution, which maximizes the Sender utility. The LP is solved in $O(mn)$. If a feasible distribution is found the algorithm computes the signaling scheme ϕ_A, where A indicates the signaling scheme of the partition A. Then given a partition A, the computation of the signaling scheme ϕ_A has a time complexity of $O(mn^4)$, which is polynomial. However, the algorithm has to iterate this process for every possible partition derived by the set of actions $[n]$. Consequently the algorithm iterates for every possible subset of A. This process implies a

Algorithm 3.1 Compute the optimal signaling scheme for the Sender

Input: Receiver utilities u_r, Sender utilities u_s, parameter $\delta > 0$ and the prior distribution μ.
Output: optimal signaling scheme ϕ^* for the Sender.
$phiTot$=[]
for any non-empty $A \subseteq [n]$ **do**
  V=[]
  $uSender$=[]
  for $a \in A$ **do**
    a^*=a
    $csiTemp$=[]
    Append all the vertices of the polytope derived by the following constraints to the vertices set in $csiTemp$:
    $\sum_{\theta\in\Theta} \xi_\theta u_{r,\theta}(a^*) \geq \sum_{\theta\in\Theta} \xi_\theta u_{r,\theta}(a') + \delta,\ a' \notin A$
    $\sum_{\theta\in\Theta} \xi_\theta u_{r,\theta}(a^*) \leq \sum_{\theta\in\Theta} \xi_\theta u_{r,\theta}(a) - \delta,\ a \in A$
    $\sum_{\theta\in\Theta} \xi_\theta u_{r,\theta}(a^*) \geq \sum_{\theta\in\Theta} \xi_\theta u_{r,\theta}(a),\ a' \in A$
    $\xi \in \Delta^m$
    for ξ in $csiTemp$ **do**
      V.append(ξ)
      $uSender$.append(argmin$_{a\in A}(\xi u_s(a))$)
  vertices=set(V)
  uFinal=[len(vertices)]
  for i in range(vertices) **do**
    uFinal[i]=-inf
    for v in range(V) **do**
      if (V[v]==vertices[i] and uFinal[i] < uSender[v]) **then**
        uFinal[i] = uSender[v]
  Solve the following LP, to obtain the signaling scheme of A.
  maximize: $utot = \sum_{k\in vertices} x_k uFinal(k)$
  subject to:
  $\sum_{k\in vertices} x_A(k) v_k(\xi_\theta) = \mu_\theta,\ \theta \in \Theta$
  $x_k \geq 0$
  $\sum_{k\in vertices} x_k = 1$
  if (LP.status=="Optimal") **then**
    $\phi_A(\theta, s) = \frac{x_k v_k(\theta)}{\mu_\theta}$, for all θ in Θ
    $phiTot$.append($\phi_A, utot$)

$\phi^* = max_{utot}(phiTot)$
return(ϕ^*)

complexity of $O(2^n)$, which is exponential. Finally the time complexity of the algorithm is O($2^n poly(m, n)$), which confirms the statement of the theorem 3.3.

4 | Analysis of the algorithm

In this chapter we focus in the algorithm proposed in the proof of the theorem (3.3) and we discuss in detail its functioning. Then we proceed to describe its implementation giving all the details.

4.1. Algorithm functioning

The algorithm described in (3.1) is used to find the optimum signaling scheme for the Sender, in this particular case of the Bayesian Persuasion framework. We can divide the algorithm functioning in three phases, where the first two are performed for every possible partition and the last one is performed as the final stage of the algorithm. The phases are the following:

1. **Enumeration** of all the vertices
2. **Finding** a feasible signaling scheme
3. **Computing** the optimal signaling scheme

The algorithm receives as input: both the utility matrices of the Sender and the Receiver, respectively u_s and u_r, the uncertainty value δ and the prior distribution μ. The first thing that is done by the algorithm is to find all the possible partitions given the number of possible actions. A possible partition is a binary partition between the set of actions, which originates two sets, where the first one represents the actions belonging to the BR_δ set and the second one represents all the actions not belonging to the BR_δ set. Consequently all the possible partitions are all the possible BR_δ sets which can be induced by some posterior $\xi \in \Xi$. After this step is done it performs the first two phases for every partition and in the end it performs the final phase.

4.1.1. Enumeration of the vertices

The first phase is the enumeration of all the vertices of a partition. This process is done multiple times, one for each action belonging to the BR_δ set of the partition considered.

Consequently each time, one of the actions of the BR_δ set is defined as the a^* action. The vertices are enumerated in the polytope originated by the inequalities described in the proof of correctness of the theorem (3.3). Then, after the algorithm finds, for all the actions (considered one at a time the a^* action) belonging to the BR_δ set, all the possible vertices, it performs an analysis considering all the vertices found. The algorithm for each vertex found, which represents a feasible posterior, computes the Sender utility for each action belonging to the BR_δ set, keeping only the one with minimum utility. This is done because we are creating a robust algorithm, which can guarantee to the Sender the best utility for him in the worst case scenario.

After the algorithm associates each vertex to an utility value, it proceeds to eliminate all the duplicate vertices, maintaining between all the possible duplicates only the vertex with the most utility for the Sender. Indeed when the posterior is associated with multiple Sender utilities we have to consider only the utility with the most value for the Sender by breaking ties in favour of him. This is done to guarantee the existence of the equilibrium. Consequently after such steps are performed we obtain for the partition, and so for the BR_δ set taken in examination, the posterior probabilities that we would induce via our signaling scheme. To find the signaling scheme that could generate such posteriors, the algorithm enters in the second phase.

4.1.2. Finding a feasible signaling scheme

In such phase the algorithm perform a LP optimization problem, where there is only a variable to be optimize. This variable is a probability distribution called x, which represents the probability of inducing a specific posterior. So for every vertex $v_k \in V$, where V represents the vertices set, we will have a probability $x_k \in x$ associated to it. The objective function to maximize is the product between the utility associated to the vertices and x, consequently we will have $\max \sum_{k \in K} x_k v_k$. For what regards the constraints, we need to be assured that the variable x represents a feasible probability distribution implying that all the components x_k need to be non-negative and have to add up to 1. Another constraint is used to guarantee that the posterior generated is feasible in respect to the prior distribution μ. In order to be assured of this aspect, we introduced the following constraint:

$$\sum_{k \in vertices} x_k(\theta) v_k(\theta) = \mu_\theta, \ \forall \ \ \theta \in \Theta.$$

Then the algorithm solves the LP and if a solution is found, it stores the resulting probability distribution, the vertices set and the objective function value.

4.1.3. Computing the optimal signaling scheme

After the algorithm performs the previous two phases for every partition, we enter in its final phase. In this phase the algorithm selects, between all the feasible solutions found, the one with the most utility for the Sender. Indeed the solution with the most utility is the optimal one. Then the algorithm ends by computing the optimal signaling scheme ϕ^*, where $\phi^*(\theta, k) = \frac{x_k(\theta)v_k(\theta)}{\mu_\theta}$.

4.2. Algorithm implementation

In this section we describe in which environment the algorithm was implemented describing in detail the libraries used.

We decided to adopt python as our programming language. This decision permitted us to use some specific libraries able to perform precise and vital tasks in the algorithm process. The environment we chose for the implementation was Google Colab. Other than the classic python libraries that were used to manage data such as Pandas and Numpy, we used the Pypoman and PuLP library. The first one was used to find the vertices, while the other one was used to perform the linear programming maximization. Pypoman is a library that was created in order to allow common operations over convex polyhedra such as polytope projection and vertex enumeration. The libray requires a python version of at least 3.7. For what regards PuLP , it is used to model Linear Programs and can solve them using various type of optimizers such as GLPK, COIN-OR CLP/CBC, CPLEX, GUROBI, MOSEK, XPRESS, CHOCO, MIPCL, SCIP optimizers. For our implementation, because the linear programs that the algorithm needed to solve were simple, we opted for the default optimizer of the library which is CPLEX.

4.3. A running example of the algorithm

In this section we show how the implementation of the algorithm 3.1 solves a problem instance of the Bayesian persuasion framework when a δ value is added to the u_r payoff matrix.

We consider the following **setting**:

- the number of states of nature is 3
- the number of actions is 3
- the prior is defined as $\mu = (0.19, 0.25, 0.56)$

- the δ value is 0.5

We define the payoffs for the Sender and the Receiver described in the following tables. The right table shows the payoffs of the Sender, while the left one shows the payoffs of the Receiver.

	a_1	a_2	a_3
θ_1	0.92	0.85	0.82
θ_2	0.21	0.21	0.19
θ_3	0.51	0.85	0.64

	a_1	a_2	a_3
θ_1	0.23	0.43	0.97
θ_2	0.01	0.92	0.90
θ_3	0.17	0.92	0.78

Table 4.1: Payoff matrices of the Sender and the Receiver in function of the state of nature and action

The algorithm starts by creating all the possible action combinations of the BR_δ set and then identifies all the possible vertices of the polytope. By observing the Receiver utilities and because the δ is 0.5, the algorithm finds a set of vertices for two possible BR_δ sets; The first is composed only by the action a_3, while the second is the set of actions (a_2, a_3). While for the first set the LP does not find a feasible distribution x able to generate the posterior found, for the second BR_δ set the LP finds a feasible solution, which is also the optimal signaling scheme ϕ^*. The signals found by the algorithm are expressed in the following table:

	θ_1	θ_2	θ_3
s_1	0.16	0	0
s_2	0.05	1	0
s_3	0.79	0	1

	θ_1	θ_2	θ_3
ξ_1	1	0	0
ξ_2	0.04	0.96	0
ξ_3	0.21	0	0.79

Table 4.2: Optimal signaling scheme ϕ^* and the posterior induced

This signaling scheme guarantees the Sender in this scenario of uncertainty an utility of 0.56. The utility value and the incidence of the δ value in the problem instance depends deeply from the utility matrices.

5 | Experimental results

In this chapter, we describe how the algorithm was tested, detailing the specific aspects considered to gain insights into its workings and effectiveness. To achieve this, we conducted two types of experiments. The first focused on identifying and analyzing disparities in execution time and assessing the feasibility across various settings. The second experiment concentrated specifically on the δ parameter, aiming to illustrate how this parameter impacts the expected utility for the Sender.

5.1. Execution time

In this section, we present the methodology of the first type of experiment, which was designed to analyze the execution time of the algorithm. From now on we name the execution time as CPU time.

Test suite and parameters. To accomplish this, we computed the CPU time across multiple settings. The settings were based on two parameters: the number of states of nature m and the number of actions n, with both parameters capped at a maximum value of 7. We then generated a range of problem settings, encompassing every possible combination of actions and states of nature, varying from 2 to 7. This resulted in the analysis of 36 distinct problem settings. For each setting, in order to find a good approximation of the CPU time, we created 20 different samples. Each sample contained all necessary inputs for the algorithm's operation. Specifically we set the δ value at 0.1 and randomly generated the prior and utility matrices for both the Sender and the Receiver. The utility value for a specific state of nature and a specific action was fixed in a range from 0 to 1. In each setting we measured:

- Average CPU time of a single partition;
- Average CPU time to compute the vertices;
- Average CPU time of the LP process;
- Total CPU time for all samples.

All the time measures were done using the python library *time* and the code was executed using the *Google colab* CPU. The average CPU time of every component of the setting was computed by summing up all the registered times and then by dividing by the number of measures registered. We decided to keep track of these particular metrics in order to monitor the performance of the algorithm in the most crucial and most computational intensive parts. As previously mentioned, the algorithm is divided into several phases. Therefore, we measured the CPU time of each phase to identify the algorithm's key component.

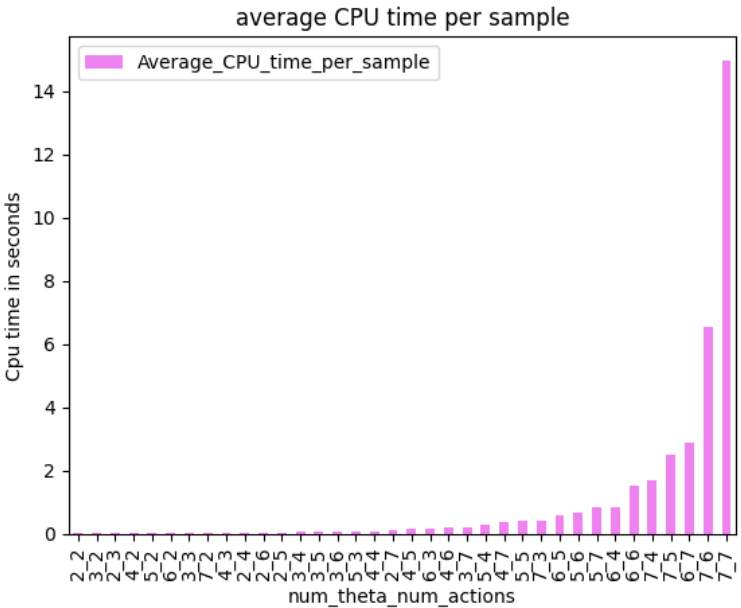


Figure 5.1: The relationship between the average CPU time of a sample and the number of states of nature and actions

Results. The results of this type of experiment are the following: The chart 5.1 describes the relationship between the CPU time and the dimension of the problem. To represent at best this relationship, we used a box plot. Indeed the box plot describes how much time on average the algorithm needs to solve a single sample. As we can see, the time needed grows in an exponential manner. Notably, the required time increases exponentially, aligning with our theoretical expectations. Interestingly, solving a problem with seven states of nature and seven actions takes, on average, only 14 seconds.

We further analyzed the CPU time of individual phases of the algorithm to understand the source of the exponential increase in execution time. The figure 5.2, a line plot, displays the averages CPU times for the algorithm's main parts. The red line represents the average time for vertices calculation, the green for the linear programming (LP) calculation, and the orange for the execution of a partition. As the problem's complexity

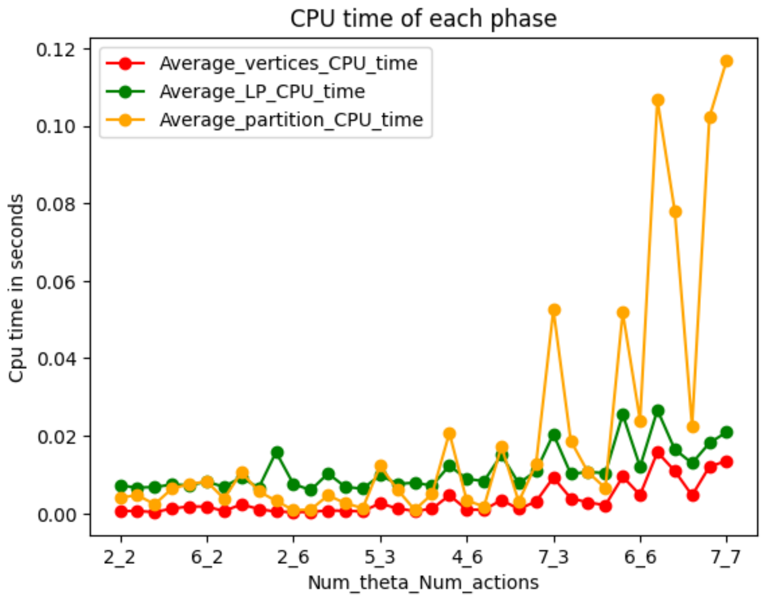
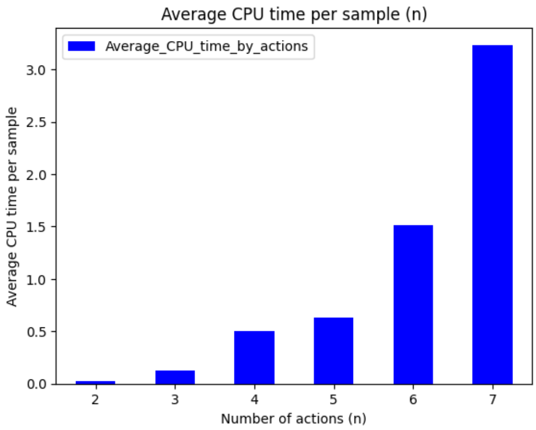


Figure 5.2: CPU time of each phase

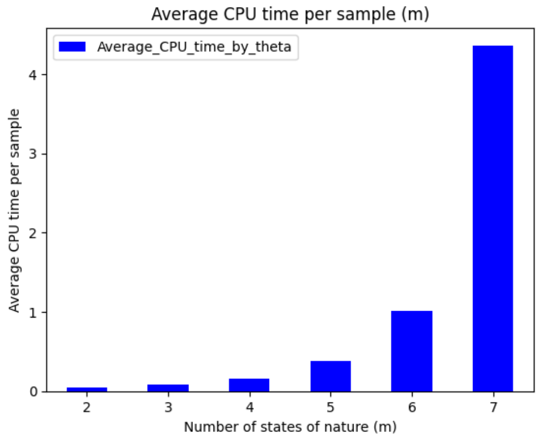
increases, all three metrics rise. However, the first two exhibit polynomial growth, while the exponential increase is primarily due to the partition execution, represented by the orange line. This growth is attributed to the computation of all possible vertices in each partition and the subsequent LP solution to find a feasible signaling scheme. As the number of actions and states of nature increases, the frequency of vertex calculations grows exponentially, thereby increasing the LP CPU time. This empirical finding corroborates the theoretical discussion about the algorithm's complexity, pinpointing the analysis of all possible partitions as the cause of its exponential nature.

A final analysis was done to understand better which of the two parameters impacted more in the CPU time. To find this out, we computed the average CPU time per sample grouped by the number of actions (n) and the average CPU time per sample grouped by the number of states of nature (m).

From Figure **??** we can observe the trend of the average CPU time in relationship with the number of actions and the average CPU time in relationship with the number of states of nature. It can be observed that the actions chart has a trend, which is dominated by the states of nature chart. Indeed we have that, from point to point, the increase in values is bigger for the parameter m rather than for the parameter n and so the number of states of nature have a bigger impact on the CPU time rather than the number of actions.



(a) Relationship between the average execution time and n



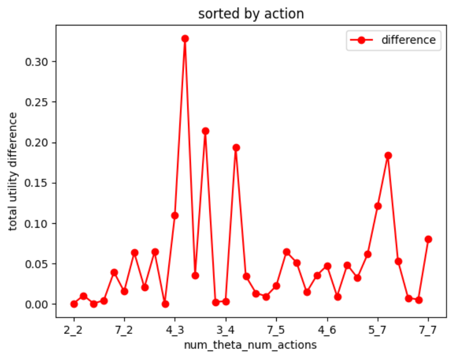
(b) Relationship between the average execution time and n

5.2. Relationship between utility and uncertainty

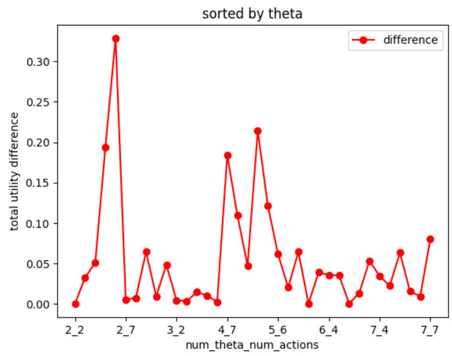
In this section, we show the second type of experiment conducted to assess the expected utility of the Sender when the δ parameter varies. The influence of the δ parameter is strongly contingent to the problem setting. Indeed the u_r payoff matrix plays a big role in this aspect. When uncertainty is high, relative to payoff values, the δ value has a more significant impact on the Sender's strategy than in scenarios with a lower δ value. A key observation is that the presence of uncertainty, rather than its magnitude, primarily affects the Sender's utility. Indeed if there is uncertainty the Sender has to find a robust signaling scheme rather than a signaling scheme that simply maximizes his expected utility.

Therefore, our analysis focuses on cases where the utility values in the u_r payoff matrix is such that the utility values are relatively uniform, allowing the uncertainty factor to introduce greater variability into the Sender's signaling scheme and, consequently, his expected utility. Indeed in this scenario the degree of uncertainty is a key component. We hypothesize that a smaller δ parameter parameter will yield a higher utility value compared to scenarios with a larger δ value.

Test suite and parameters. In order to better understand the relationship between the δ value and the problem itself, we structured the experiment as follows: Like the first experiment, we set the maximum values for the number of states of nature m and actions n to 7. Then we proceeded to create a setting for all the possible combinations of states of nature and actions, generating 17 samples for each scenario. However, the creation of samples was done differently. Unlike the first experiment, we standardized the prior distribution and utility matrices across all settings, with utility values ranging from



(a) Difference in expected utility as the δ varies ordered by the number of actions



(b) Difference in expected utility as a function of δ and number of states of nature

Figure 5.4: The relationship between the difference in the Sender's expected utility and the number of states of nature and actions

0 (minimum) to 0.5 (maximum). While, for the δ value, we selected a different value for every sample, ranging from 0.1 to 0.9.

The aim of this experiment was to discern if there was a relationship between the δ value and the setting and also to determine if, for each type of setting the increase of the δ value comported in a non increase of the Sender expected utility. For this purposes we decided to measure the expected utility of each sample and then analyze the difference between the Sender expected utility with a δ of 0.1 and a δ of 0.9.

Finally we did the following plots:

- A plot describing the relationship between the δ and the number of actions;
- A plot depicting the correlation between the δ parameter and the states of nature;
- A plot showcasing the relationship between the Sender's expected utility and the δ parameter.

Results. As previously mentioned, the amount of uncertainty present may not always significantly impact the Sender's expected utility. However, when the δ is comparable in value respect to the Receiver payoffs, then the difference in the Sender expected utility between the setting with 0.1 δ and the one with 0.9 δ is often present. Moreover, as expected, a bigger uncertainty does not bring a better utility value for the Sender, but at most it results in an equal utility value. We used the difference metric in order to evaluate how the δ value impacted in the result to understand how much is the distance between the maximum possible utility value and the minimum one in a robust context. The following two plots, where the first one describes the relationship between the δ and

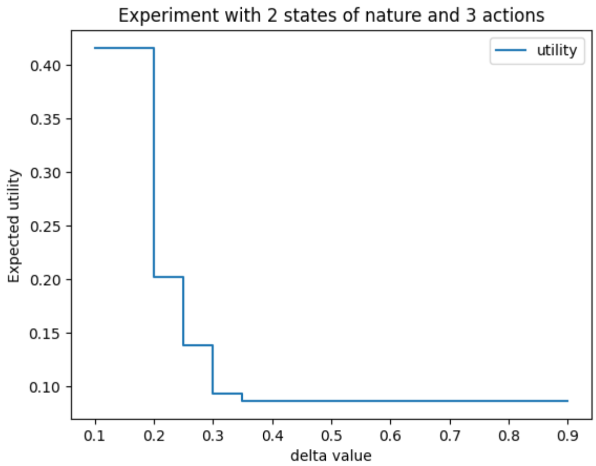


Figure 5.5: Sender expected utility based on the quantity of uncertainty

the number of actions and the second one describes the relationship between the δ and the number of state of nature.

As it can be noted by the Charts 5.4, there is no direct correlation between changes in the Sender's expected utility and the complexity of the problem setting. Indeed an increase of the number of actions or the number of states of nature does not correspondingly increase the variation in the Sender's expected utility between δ values of 0.1 and 0.9. However, when the uncertainty value is close to the utility value of the Receiver, we have that almost every time the Sender's expected utility changes based on the δ value. Also as the uncertainty increases there is not a solution that turns out to be better in terms of utility than the solution found when the uncertainty is minimum.

Subsequently, we direct attention to a specific experiment, such as the one conducted for a setting with 2 states of nature and 3 actions, to examine how the Sender's expected utility shifts with varying δ values. From the plot described in the Image 5.5, it can be seen the typical decreasing trend. Indeed at lower uncertainty levels, the Sender's expected utility peaks, but it diminishes as the δ value rises. For instance, in this particular scenario, we have a big change in the Sender expected utility, which starts with a big value and decreases drastically to an almost minimum utility.

6 | Conclusions and future works

In this thesis we studied the Bayesian persuasion framework. We extended the existing knowledge regarding this type of problem, by exploring scenarios where the Sender faces uncertainty about the Receiver's behavior. To address this unpredictability we decided to face the problem in a robust manner to safeguard the sender's expected utility.

Our approach, centered around robustness, was pivotal in navigating the unpredictability inherent in the sender-Receiver dynamic. We modeled the problem with a focus on the worst-case scenario, where the Receiver invariably opts for the action most detrimental to the Sender. This is despite the fact that, from the sender's perspective, the Receiver's choice is sub-optimal. This approach led to a novel definition of the Best Response set, and we demonstrated that this set is not convex. This finding resulted in the problem not being able to be solved in polynomial time. Indeed to find the solution it is mandatory to divide the not convex set in all the possible convex partitions, which lead to an exponential time solution, unless $P = NP$.

We then proceeded to analyze the usability of the algorithm through an in depth experimentation. The time experimentation lead to the conclusion that, even though the problem is NP-Hard, it can be a useful resource in vast majority of cases where the number of states of nature and of actions, which come in play are limited. While the experimentation on the uncertainty, lead to the conclusion that the main discrimen for the Sender expected utility is the presence or absence of uncertainty, because it would lead to a completely different approach to the problem. However, when the uncertainty factor is present and the Receiver payoffs esteems are close to the δ value, then the Sender expected utility could suffer from greater variability depending on the δ value.

Our research primarily concentrated on scenarios involving a single Receiver. Then as a new possible research direction could be interesting the case where there are multiple Receivers and each one has a different δ value. Indeed could be interesting to find out the different approach of the Sender in the case of a public signaling scheme and in the case of a private signaling scheme. Although the Bayesian persuasion has already been studied in contexts with multiple Receivers both when the Sender adopts a public or a private signaling scheme, we think that the presence of uncertainty for the Sender could lead him to adopt a completely different approach to the problem.

Also one key assumption of our case study is that the degree of uncertainty expressed by the δ value is already known and never changes. Could be interesting to investigate scenarios where at the beginning there is absence of uncertainty and only in at a later time the uncertainty element appears. This would lead the Sender to learn in an online fashion the presence of the δ value. Even though the Bayesian persuasion problem has already being discussed in the online occurrence, this variation of the standard problem could result in the problem to be viewed with a different perspective. Consequently another scenario could be represented by the setting where the uncertainty changes in value overtime. While in this scenario the Sender may create a test to evaluate if the δ value changed.

Lastly, it could be relaxed the assumption of the Receiver getting signals only by the Sender. Indeed allowing for multiple sources of information for the Receiver could significantly complicate the sender's task of persuasion, especially in situations where the sender does not fully understand the Receiver's utilities, thus posing a greater challenge in enhancing his expected utility.

Bibliography

[1] R. Alonso and O. Câmara. Persuading voters. *American Economic Review*, 106(11): 3590–3605, 2016.

[2] Y. Babichenko and S. Barman. Algorithmic aspects of private bayesian persuasion. In C. H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 34:1–34:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi: 10.4230/LIPICS.ITCS.2017.34. URL `https://doi.org/10.4230/LIPIcs.ITCS.2017.34`.

[3] Y. Babichenko, I. Talgam-Cohen, H. Xu, and K. Zabarnyi. Regret-minimizing bayesian persuasion. *Games and Economic Behavior*, 136:226–248, 2022.

[4] X. Badanidiyuru, Bhawalkar. Targeting and signaling in ad auctions. In *Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2545–2563. SIAM, 2018.

[5] D. Bergemann and S. Morris. Information design: A unified perspective. *Journal of Economic Literature*, 57(1):44–95, 2019.

[6] M. Bernasconi, F. Cacciamani, M. Castiglioni, A. Marchesi, N. Gatti, and F. Trovò. Safe learning in tree-form sequential decision making: handling hard and soft constraints. In *International Conference on Machine Learning*, pages 1854–1873. PMLR, 2022.

[7] M. Bernasconi, M. Castiglioni, A. Marchesi, N. Gatti, and F. Trovò. Sequential information design: Learning to persuade in the dark. *Advances in Neural Information Processing Systems*, 35:15917–15928, 2022.

[8] M. Bernasconi, M. Castiglioni, A. Celli, A. Marchesi, F. Trovò, and N. Gatti. Optimal rates and efficient algorithms for online bayesian persuasion. In *International Conference on Machine Learning*, pages 2164–2183. PMLR, 2023.

[9] M. Bernasconi, M. Castiglioni, A. Marchesi, and M. Mutti. Persuading farsighted receivers in mdps: the power of honesty. In *NeurIPS*, 2023.

[10] M. Bernasconi-de Luca, F. Cacciamani, S. Fioravanti, N. Gatti, A. Marchesi, and F. Trovò. Exploiting opponents under utility constraints in sequential games. *Advances in Neural Information Processing Systems*, 34:13177–13188, 2021.

[11] U. Bhaskar, Y. Cheng, Y. K. Ko, and C. Swamy. Hardness results for signaling in bayesian zero-sum and network routing games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 479–496, 2016.

[12] O. Candogan. Persuasion in networks: Public signals and k-cores. In A. Karlin, N. Immorlica, and R. Johari, editors, *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 133–134. ACM, 2019. doi: 10.1145/3328526.3329618. URL `https://doi.org/10.1145/3328526.3329618`.

[13] M. Castiglioni, A. Celli, and N. Gatti. Persuading voters: It's easy to whisper, it's hard to speak loud. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1870–1877. AAAI Press, 2020. doi: 10.1609/AAAI.V34I02.5555. URL `https://doi.org/10.1609/aaai.v34i02.5555`.

[14] M. Castiglioni, A. Celli, A. Marchesi, and N. Gatti. Online bayesian persuasion. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16188–16198. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/ba5451d3c91a0f982f103cdbe249bc78-Paper.pdf`.

[15] M. Castiglioni, A. Celli, A. Marchesi, and N. Gatti. Signaling in bayesian network congestion games: the subtle power of symmetry. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 5252–5259. AAAI Press, 2021. doi: 10.1609/AAAI.V35I6.16663. URL `https://doi.org/10.1609/aaai.v35i6.16663`.

[16] Y. Cheng, H. Y. Cheung, S. Dughmi, E. Emamjomeh-Zadeh, L. Han, and S. Teng.

Mixture selection, mechanism design, and signaling. *CoRR*, abs/1508.03679, 2015. URL `http://arxiv.org/abs/1508.03679`.

[17] G. de Clippel and X. Zhang. Non-bayesian persuasion. *Journal of Political Economy*, 130(10):2594–2642, 2022.

[18] S. Dughmi and H. Xu. Algorithmic bayesian persuasion. In D. Wichs and Y. Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 412–425. ACM, 2016. doi: 10.1145/2897518.2897583. URL `https://doi.org/10.1145/2897518.2897583`.

[19] P. Dworczak. Preparing for the worst but hoping for the best: Robust (bayesian) persuasion. *SSRN*, 2020. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3523114`.

[20] Y. Emek, M. Feldman, I. Gamzu, R. P. Leme, and M. Tennenholtz. Signaling schemes for revenue maximization. *ACM Trans. Economics and Comput.*, 2(2):5:1–5:19, 2014. doi: 10.1145/2594564. URL `https://doi.org/10.1145/2594564`.

[21] G. Farina, C. Kroer, and T. Sandholm. Stochastic regret minimization in extensive-form games. In *International Conference on Machine Learning*, pages 3018–3028. PMLR, 2020.

[22] J. Gan, R. Majumdar, G. Radanovic, and A. Singla. Bayesian persuasion in sequential decision-making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5025–5033, 2022.

[23] J. Gan, M. Han, J. Wu, and H. Xu. Robust stackelberg equilibria. In K. Leyton-Brown, J. D. Hartline, and L. Samuelson, editors, *Proceedings of the 24th ACM Conference on Economics and Computation, EC 2023, London, United Kingdom, July 9-12, 2023*, page 735. ACM, 2023. doi: 10.1145/3580507.3597680. URL `https://doi.org/10.1145/3580507.3597680`.

[24] J. Hu and X. Weng. Robust persuasion of a privately informed receiver. *Economic Theory*, 72:909–953, 2021.

[25] E. Kamenica, K. Kim, and A. Zapechelnyuk. Bayesian persuasion and information design: perspectives and open issues. *Economic Theory*, 72:701–704, 2021. doi: 10.1007/s00199-021-01383-4. URL `https://doi.org/10.1007/s00199-021-01383-4`.

[26] G. Kamenica. Bayesian persuasion. *American Economic association*, 2011. URL `https://www.aeaweb.org/articles?id=10.1257/aer.101.6.2590`.

[27] C. Kroer, G. Farina, and T. Sandholm. Robust stackelberg equilibria in extensive-form games and extension to limited lookahead. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[28] S. Lasaulce and H. Tembine. *Game theory and learning for wireless networks: fundamentals and applications*. Academic Press, 2011.

[29] R. Luce. *Games and Decisions: Introduction and Critical Survey (Dover Books on Mathematics)*. Paperback, 1989.

[30] Y. Mansour, A. Slivkins, V. Syrgkanis, and Z. S. Wu. Bayesian exploration: Incentivizing exploration in bayesian games. *arXiv preprint arXiv:1602.07570*, 2016.

[31] P. B. Miltersen and O. Sheffet. Send mixed signals – earn more, work less. *CoRR*, abs/1202.1483, 2012. URL `http://arxiv.org/abs/1202.1483`.

[32] C. H. Papadimitriou. Computational complexity. In *Encyclopedia of computer science*, pages 260–265. 2003.

[33] J. Pita, M. Jain, M. Tambe, F. Ordónez, and S. Kraus. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.

[34] Z. Rabinovich, A. X. Jiang, M. Jain, and H. Xu. Information disclosure as a means to security. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 645–653, 2015.

[35] S. Vasserman, M. Feldman, and A. Hassidim. Implementing the wisdom of waze. In Q. Yang and M. J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 660–666. AAAI Press, 2015. URL `http://ijcai.org/Abstract/15/099`.

[36] J. Wu, Z. Zhang, Z. Feng, Z. Wang, Z. Yang, M. I. Jordan, and H. Xu. Sequential information design: Markov persuasion process and its efficient reinforcement learning. *arXiv preprint arXiv:2202.10678*, 2022.

[37] H. Xu, R. Freeman, V. Conitzer, S. Dughmi, and M. Tambe. Signaling in bayesian stackelberg games. In C. M. Jonker, S. Marsella, J. Thangarajah, and K. Tuyls, editors, *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 150–158. ACM, 2016. URL `http://dl.acm.org/citation.cfm?id=2936950`.

[38] C. Zhao and J. Li. Equilibrium selection under the bayes-based strategy updating rules. *Symmetry*, 12(5):739, 2020.

[39] K. Zhu, E. Hossain, and A. Anpalagan. Downlink power control in two-tier cellular ofdma networks under uncertainties: A robust stackelberg game. *IEEE Transactions on Communications*, 63(2):520–535, 2014.

[40] Y. Zu, K. Iyer, and H. Xu. Learning to persuade on the fly: Robustness against ignorance. In P. Biró, S. Chawla, and F. Echenique, editors, *EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021*, pages 927–928. ACM, 2021. doi: 10.1145/3465456.3467593. URL https://doi.org/10.1145/3465456.3467593.