F3

Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's Thesis

Nash Equilibria for Regression Models over Strategic Data

Viacheslav Larionov

May 2023

Supervisor: doc. Ing. Tomáš Kroupa, Ph.D

Acknowledgement / Declaration

I want to thank my supervisor, doc. Tomáš Kroupa, for introducing me to a section of mathematics called Game Theory, showing me a possible application, and being patient because it was difficult work for me. I am grateful to all the teachers who helped me to get through this long, tough path to my thesis. I also want to thank all the students who studied with me and helped me pass the courses. I couldn't have done it without them. I am grateful to the CTU for the excellent organized learning process and the supportive atmosphere. And last but not least, I want to thank my family for the opportunity to get this education and the endless support from entrance exams to the state final exams.

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 22 May 2023

.....

Abstrakt / Abstract

Tato práce se zaměřuje na problém strategického zkreslení dat, ukazuje, kde se může vyskytnout a jaká je motivace k jeho řešení. Na tento problém se dívá prostřednictvím teorie her a vysvětluje motivaci tohoto přístupu. Dále se práce zabývá několika algoritmy pro řešení tohoto druhu problému. Aby potvrdila správnost teoretických konceptů, provádí několik experimentů. Poté se zaměřuje na konkrétní model a zkouší různé způsoby řešení tohoto problému. Nakonec článek rozšiřuje stávající model a provádí s ním experimenty.

Klíčová slova: teorie her, hry s nenulovým součtem, spojité hry, nekonečné hry, potenciální hry, Bayesovské hry, Double Oracle, Iterated Best Response.

This paper focuses on the problem of strategic data distortion, shows where it can occur, and what the motivation for the solving is. It looks at this problem through game theory and explains the motivation behind this approach. The paper then considers several algorithms for solving this kind of problem. In order to confirm the correctness of theoretical concepts, makes several experiments. Then it focuses on a specific model and attempts different ways of solving this problem. In the end, the paper tries to extend the existing model and conducts experiments with it.

Keywords: game theory, nonzero-sum games, continuous games, infinite games, potential games, Bayesian games, Double Oracle, Iterated Best Response.

/ Contents

Tables / Figures

Chapter 1
Introduction

1.1 The Dilemma of Balancing Privacy and Accuracy of Recommendations in Personal Data Disclosure

Data is becoming an increasingly valuable commodity in today's digital age. Companies and organizations collect vast amounts of data about people in order to offer personalized services and recommendations. However, collecting and sharing personal data raises important privacy and accuracy issues.

A major problem is noise distortion present in the data. Naturally, the quality of the data, distorted by such noise, is key to the accuracy of the model. In many cases, however, obtaining high-quality data can come at a cost to the data source.

For example, this occurs when the data are personal. Accordingly, people who care about their privacy tend to distort the data. Or, in extreme cases, they refrain from disclosing it at all.

In cases where high-quality personal data are expensive, it is worth considering the strategic behavior of the participants providing data about themselves. We consider a situation where globally successful data analysis can benefit the people from whom the data is gathered. Open collaboration projects, by their very nature, implicitly imply shared benefits associated with the success of the collaboration. If such benefits outweigh the associated costs of privacy or effort, people may consent to the collection and analysis of high-quality data.

One area where this dilemma is particularly acute is in questionnaire-based recommendation systems. These systems collect personal data from participants to create personalized recommendations. The accuracy of recommendations depends on the honesty of all participants. If all participants tell the truth, the recommendations will be helpful. However, if only one participant lies, they may gain unfairly at the expense of others. This creates a dilemma for participants who are torn between wanting to maintain confidentiality and wanting accurate recommendations.

To better understand this dilemma, it is helpful to look at the two types of participants commonly present in a given game: participants sensitive to privacy (and thus prone to misrepresentation) and participants for whom disclosure of personal data is not a big deal. We can also look at these two types from a different angle. The first type does not care about the survey's outcome (the model's accuracy). They are willing to risk the benefits of participation in exchange for preserving their privacy. The second type, more conservative, is not ready to risk the outcome of the whole project to protect their privacy and tends to disclose data about themselves. The above types are extremes that also involve intermediate options. The dilemma for participants is to find a balance between confidentiality and the benefits derived from the model's accuracy (based on survey data).

Existing literature [1] focuses on the case where all participants are equally aware of each other and respectively know each player's type, based on which they can build

their strategy. We will extend this problem to the Bayesian setting. When participants do not have information about the other agent types, they do not know their attitudes toward privacy and risk.

1.2 Game theory

A dilemma where participants should offer more or less their personal data is a strategic problem because there are several participants who choose between different policies. Moreover, the outcome of each player depends not only on his own choices but also on the choices of the other participants. At the same time, none of the players knows what choice will be made by the other, and each player seeks to maximize his own winnings. This means that each player must consider the possible actions and reactions of the opponent to choose the best strategy.

Game theory provides tools for analyzing strategic games as this. Solving the dilemma with game theory allows us to determine the optimal strategies for each player and find an equilibrium in that system. This helps to better understand and predict the behavior of the participants in general, which is necessary if we are looking at the situation from the perspective of a data analyst.

Approaches that do not involve game theory may not take into account the interaction and interests of other players, which can lead to ineffective decisions and unpredictable outcomes. Also, using other approaches, not related to game theory, may be less effective. Therefore, game theory is the most appropriate tool for analyzing and solving this situation and many other strategic games.

1.3 Applications

This personal information disclosure dilemma can occur in a variety of fields, including medicine, marketing, finance, social media, etc.

For example, in medicine, patients may face the dilemma of disclosing their personal health information. If they don't disclose their information, it can lead to inaccurate diagnosis and treatment. On the other hand, if they disclose their information, it may help research the disease and invent new treatments, but they sacrifice their confidentiality. In marketing and advertising, customers may also face the same dilemma. For instance, they may receive personalized recommendations and offers based on their personal information, but sacrifice their privacy. Another case in point is using social media. If users do not disclose their information, they may miss out on opportunities to connect with friends and colleagues.

1.4 Summary of the thesis

We have studied several types of games. We started with the simplest one and gradually made it more complicated in order to get closer to the final problem that we wanted to solve. We considered what solution is in our games and what strategies agents can use. Then we've considered several ways (algorithms) to solve these games. We made several experiments in order to confirm our theoretical knowledge of algorithms. And after that, we found a solution to our main problem - the dilemma of balancing privacy and accuracy of recommendations in personal data disclosure.

Chapter 2
Basic Notions

2.1 Game theory

Game theory is a mathematical framework used to analyze the behavior of groups and individuals in strategic situations where the outcome of one's decision depends on the decisions made by others. This is a field of applied mathematics that is widely used in fields such as psychology, politics, economics, and computer science.

Game theory deals with the study of decision-making in situations where there are multiple players, each with their own goals and preferences. These players interact with each other, and the outcome of the game depends on the choices they make. In particular, game theory is interested in understanding how players can make rational decisions in such situations, and how the outcome of the game can be predicted based on the players' decisions. In fact, game theory can find its place wherever we are interested in the competition of agents, or the collaboration of agents. On this basis, we can distinguish two areas in game theory: zero-sum and nonzero-sum games. The first type isn't cooperative at all, because as much as one participant wins, the same amount the other loses. In sum, their gain is always zero (or a constant). However, the second type is something between cooperative and non-cooperative. In our work, we focus on nonzero-sum games.

2.1.1 Strategic games

We will describe the main aspect of game theory, which is the strategic form of the game. What do we need to know in order to describe the game and then work with it? Of course, we need players, but it is not enough to describe one player and copy the given rules for all. Each of them may have different strategies and rewards associated with their actions. So everyone may have their own set of strategies and set of outcomes or cost functions. It is important to add that this type of game is not dynamic. All moves occur simultaneously. No players make arbitrary moves, they try to act rationally and they don't have the ability to work together to achieve better results.

The following basic principles and definitions are based on the book [2].

Definition 2.1. A game in strategic form (normal) *is a tuple*

$$G = (N, A, c)$$

where:

- N *is a set of agents;*
- $A = A_1 \times \ldots \times A_n$*, where* A_i *is the set of actions available to player* i*;*
- $c = (c_1, \ldots, c_n)$*, where* $c_i \colon A \to \mathbb{R}$ *is the cost function for player* i*.*

We can distinguish two kinds of games: *finite* and *infinite*. If the set of actions for each player is finite, then it is the first type. If the number of actions is an infinite set (e.g., the interval [0, 1]), then it's the second type. Next, we can distinguish a subtype. *Continuous* game is an infinite game where every strategic set is a subset of Euclidian space and the cost function is continuous.

We have looked at the spaces in which we choose actions, then we look at what can happen in these games. What decisions agents can make. For this purpose, we introduce several definitions relating to the strategies. The most trivial move a player can make is to choose what he thinks is the best action and follow it. We call this *pure strategy*. Each player may select a pure strategy a_i from a nonempty set A_i. Pure strategies of all players about a particular game can be written as an n-tuple $\mathbf{a} = (a_1, \ldots, a_n)$. But this strategy may not be sufficient if the opponents begin to change their choices. In order to minimize the average loss in either outcome, it is better to try something else. As opposed to a pure strategy, there is also a *mixed strategy*. In this case, the player randomly chooses a move with a certain probability distribution over a set of available actions. Here is the definition of mixed strategy.

Definition 2.2. Probability distribution. *Let X is any set, then Π(X) is the set of all probability distributions over X.*

Definition 2.3. Mixed strategy. *Let (N, A, c) be a normal-form game. Then the set of mixed strategies for player i is* $S_i = \prod(A_i)$. Therefore, each player i can choose a mixed strategy $s_i \in S_i$.

Definition 2.4. Mixed-strategy profile. *The set of mixed-strategy profiles is the Cartesian product of the individual mixed-strategy sets,* $S_1 \times \cdots \times S_n$.

But the strategy is only a tool to achieve a goal. The purpose of the players is to minimize the cost function. When an agent knows how the other participants will play, he will try to choose the *best response*. It's the best strategy in response to an opponent's strategies. Depending on the opponent's moves, the best response may be different and it does not have to be the only one. This concept is important to us because we use it often in algorithms (described in section 2.2).

Definition 2.5. Opponents' strategy. $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$ is a strategy profile s without agent i's strategy.

Definition 2.6. Best response. *Player i's best response to the strategy profile* s_{-i} *is a mixed strategy* $s_i^* \in S_i$ *such that* $c_i(s_i^*, s_{-i}) \leq c_i(s_i, s_{-i})$ *for all strategies* $s_i \in S_i$.

We have roughly defined the rules of the game, who plays, and what strategies they can follow. But what is the solution to these games? The basic concept of game theory is Nash equilibrium (NE). It is a set of strategies in which no one can improve their results by changing policy subject to the condition that the other players continue to play their optimal strategies. That is, each agent's choice is the best response and the system (all agents) is in balance. Assuming, the absence of external involvement.

Definition 2.7. Nash equilibrium(abbreviated NE). *A strategy profile $s = (s_1, \ldots, s_n)$ is a Nash equilibrium if, for all agents i, s_i is a best response to s_{-i}.*

If all strategies in the equilibrium are pure, it is called pure NE. According to Nash's theorem, an equilibrium exists in any finite game, but not necessarily a pure one[2]. In continuous games, unfortunately, there may be no equilibrium. It is also worth adding that the Nash equilibrium isn't necessarily unique. It can be several in the same game.

In order to find a Nash equilibrium there are many ways to do it, from simple brute-force(in the case of the finite game) to optimization methods. We discuss some of them in the Algorithms section.

2.1.2 Potential games

Investigating the Nash equilibrium can be very problematic due to the fact that each player has a different cost function. To solve this problem, we can use the concept of *potential games* in which the functions of all players are combined into one. The local optimums of the potential function are a set of pure Nash equilibria. This gives us the ability to solve games using common optimization methods.

Definition 2.8. Potential game. *A game $G = (N, A, c)$ is a potential game if there exists a function $P : A \to \mathbb{R}$ such that, for all $i \in N$, all $a_{-i} \in A_{-i}$ and $a_i, a_i' \in A_i$, $c_i(a_i, a_{-i}) - c_i(a_i', a_{-i}) = P(a_i, a_{-i}) - P(a_i', a_{-i})$.*

Also in (finite) potential games, there is always a pure Nash equilibrium.

Proof. Let $a^* = \arg\max_{a \in A} P(a)$. Clearly for any other action profile a', $P(a^*) \geq P(a')$. Thus by the definition of a potential function, for any agent i who can change the action profile from a^* to a' by changing his own action, $c_i(a^*) \geq c_i(a')$.

For better understanding, we look at an example[1] to see how it works. Let's take the game G $= (N, A, c)$ where:

- $N = \{1, 2\}$,
- $A = \{-1, 1\}^2$,
- $c_1(a_1, a_2) = -2 * a_1 - 3 * a_1 * a_2$ and $c_2(a_1, a_2) = -1 * a_2 + 3 * a_1 * a_2$.

The payoff matrix is in Table 2.1. We can construct potential function $P(a_1, a_2) = -2 * a_1 + a_2 - 3 * a_1 * a_2$ and check if it satisfies the definition of the potential function.

$$c_1(1, a_{-i}) - c_1(-1, a_{-i}) = (-2 - 3 * a_2) - (2 + 3 * a_2) = -4 - 6 * a_2$$
$$P(1, a_{-i}) - P(-1, a_{-i}) = (-2 + a_2 - 3 * a_2) - (2 + a_2 + 3 * a_2) = -4 - 6 * a_2$$

The same is true for the second player cost function. In this potential function, we can find two local minimums. At point (1, 1) the function has a value of -4, and at point (-1, -1) the function has a value of -2. Accordingly, these strategies are Nash equilibria.

	1	-1
1	-5, -2	1, 2
-1	5, 4	-1, -4

Table 2.1. Payoff matrix based on the cost functions.

[1] https://en.wikipedia.org/wiki/Potential_game

	1	-1
1	-4	0
-1	6	-2

Table 2.2. Payoff matrix based on the potential function.

2.1.3 Bayesian games

In strategy games, we always know each player's cost function. In the real world, however, we don't always know who we have to play against. These can be different players, tending to different strategies, and most importantly, with different goals, and sensitivities to the result. For example, you find an unknown person in your home, and you have a gun. You both need to decide simultaneously whether to shoot or not. If it's a thief and you decide to shoot, you might kill a person who wasn't planning to harm your health. And if it's a murderer and you don't shoot, you might just die (this example with cost functions is shown in Table 2.3). In this case, we get two strategic games at once.

	Thief	
	Shoot	Not
Shoot	1, 3	1, 2
Not	2, 1	0, 0

	Murderer	
	Shoot	Not
Shoot	0, 0	1, 2
Not	2, -2	-1, 1

Table 2.3. The householder doesn't know which of these two games he should play.

To solve such a situation, we need to introduce a new concept.

Definition 2.9. Bayesian game. A Bayesian game *is a tuple* (N, A, Θ, p, c) *where:*

- N *is a set of agents;*
- $A = A_1 \times \ldots \times A_n$, *where* A_i *is the set of actions available to player* i*;*
- $\Theta = \Theta_1 \times \ldots \times \Theta_n$, *where* Θ_i *is the type space of player* i*;*
- $p : \Theta \to [0, 1]$ *is a common prior over types; and*
- $c = (c_1, \ldots, c_n)$, *where* $c_i \colon A \times \Theta \to R$ *is the cost function for player* i*.*

A few additions to our normal form of strategy game appear. The first is the types of players, now they can be different from each other. The loss function accordingly now includes not only the choices of players but also their types. A probability distribution function is also added. Using it, we know the probability of each game. And based on this we can calculate our average loss in order to choose the best response. Despite the fact that these additions may not seem important they bring us closer to the simulation of real situations and make the game more realistic.

Since we have changed the space in which the agents play quite a lot. We have to redefine what their strategies are. A (pure) strategy of player $i \in N$ in a Bayesian game is a mapping

$$s_i \colon \Theta_i \to A_i.$$

For each type, it has a specific action. A mixed strategy is the same as in the normal form game. It is a probability distribution set over pure strategies. As before, we denote a mixed strategy for i as $s_i \in S_i$, where S_i is the set of all i's mixed strategies. Later we

use the notation $s_j(a_j|\theta_j)$ to denote the probability under mixed strategy s_j that agent j plays action a_j, given that j's type is θ_j. Next, it is very important to define three concepts in the Bayesian game: *ex post*, *ex interim*, and *ex ante*. The first represents the stage of the game when an agent knows his type and the types of other agents. The second represents the stage when an agent knows only his type. And the last denotes the stage when an agent doesn't even know his type. For each stage, we consider the expected loss. Based on this we further define the Nash Equilibrium in the Bayesian game.

Definition 2.10. Ex post expected loss. *Agent i's ex post expected loss in a Bayesian game (N, A, Θ, p, c), where the agents' strategies are given by s and the agent' types are given by θ, is defined as*

$$EL_i(s, \theta) = \sum_{a \in A} \left(\prod_{j \in N} s_j(a_j|\theta_j) \right) c_i(a, \theta).$$

In this formula, we enumerate all possible combinations of pure strategies, consider the probability that agents choose them, and multiply by the cost function.

Definition 2.11. Ex interim expected loss. *Agent i's ex interim expected loss in a Bayesian game (N, A, Θ, p, c), where i's type is θ_i and where the agents' strategies are given by the mixed-strategy profile s, is defined as*

$$EL_i(s, \theta_i) = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i}|\theta_i) EL_i(s, (\theta_i, \theta_{-i})).$$

In this stage of the game, we don't know the type of the other players, so we have to multiply our expected loss by the probability of each combination type.

Definition 2.12. Ex ante expected loss. *Agent i's ex ante expected loss in a Bayesian game (N, A, Θ, p, c), where the agents' strategies are given by the mixed-strategy profile s, is defined as*

$$EL_i(s) = \sum_{\theta_i \in \Theta_i} p(\theta_i) EL_i(s, \theta_i),$$

or equivalently as

$$EL_i(s) = \sum_{\theta \in \Theta} p(\theta) EL_i(s, \theta).$$

Now we can finally calculate our loss knowing only the strategies of other players. And this allows us to denote the best response, which minimizes our expected loss.

Definition 2.13. Best response in a Bayesian game. *The set of agent i's best responses to mixed-strategy profile s_{-i} are given by*

$$BR_i(s_{-i}) = \arg\min_{s_i' \in S_i} EL_i(s_i', s_{-i}).$$

We recall that the best responses may be several, for this reason, this formula returns set. Concluding this section we introduce the definition of Bayes–Nash equilibrium.

Definition 2.14. Bayes–Nash equilibrium. *A Bayes–Nash equilibrium is a mixed-strategy profile s that satisfies $\forall i \; s_i \in BR_i(s_{-i})$.*

In fact, this definition is no different from the one we gave before. However, we needed to define the best response in the Bayesian game in order for this definition to be correct. Note, it's quite difficult to calculate the best response, and even more to find a Nash equilibrium. Therefore, the solution to this problem can be difficult to find. But there is a way to transform the Bayesian form of the game into the normal form. To illustrate this approach we transform our previous game 2.3.

The table already defines the types of players, their possible actions, and the cost functions, we only need to add the probability of each type. The first player (householder) has only one type (H), so its probability equals 1. The second player can be a Thief (T) or Murderer (M). Let the probability of each type be 0.5. Accordingly $p(T) = 0.5$ and $p(M) = 0.5$. We look at one of the scenarios for this game. Assume that both players choose a pure strategy to shoot (e.g. $s_1 = (S)$ and $s_2 = (SS)$). The result of our expected loss for each player is a new cost function in a normal game.

$$\begin{aligned} c_1(S, SS) &= \sum_{\theta \in \Theta} p(\theta) \cdot 1 \cdot c_1(a, \theta) \\ &= p(H,T) \cdot c_1(S, SS, H, T) + p(H,M) \cdot c_1(S, SS, H, M) \\ &= 0.5 \cdot 1 + 0.5 \cdot 0 = 0.5 \\ c_2(S, SS) &= \sum_{\theta \in \Theta} p(\theta) \cdot 1 \cdot c_2(a, \theta) \\ &= p(H,T) \cdot c_2(S, SS, H, T) + p(H,M) \cdot c_2(S, SS, H, M) \\ &= 0.5 \cdot 3 + 0.5 \cdot 0 = 1.5 \end{aligned}$$

Similarly, we can calculate the expected loss and create new cost functions for other combinations of strategies. And then build a normal game 2.4 2×4 on the basis of this.

	SS	SN	NS	NN
S	0.5, 1.5	1, 2.5	0.5, 1	1, 2
N	2, -0.5	0.5, 1	1, -1	-0.5, 0.5

Table 2.4. Example of the normal game created from the Bayesian.

2.2 Algorithms

The Double Oracle and Iterated Best Response algorithms are two of the most common methods for finding Nash equilibrium in non-cooperative games. They are used to find optimal strategies for all players that reach a Nash equilibrium.

In this section, we look at both algorithms, starting with Iterated Best Response, which is a simpler and easier to understand algorithm (because it only uses pure strategies), and ending with the more complex Double Oracle algorithm (which uses a mixed-strategy space). We study each of the algorithms in detail and describe their features, advantages, and disadvantages. In the next chapter, we use these algorithms for our experiments.

2.2.1 Iterated Best Response

Sometimes calculating Nash equilibrium involves trying all possible variants (brute force). This can be very costly in terms of time/resources. In this case, Iterated Best Response algorithm can help us. The IBR algorithm consists of each player sequentially

choosing the best response strategy to the current strategies of all other players until the maximum specified limit of iterations is reached. The algorithm goes through the following steps:

- each player chooses an initial policy
- each player chooses his best response strategy to the current strategies of all other players
- continue the previous step until the change in the profit of all players is greater than the ϵ

This approximate algorithm usually works fast because it chooses the best policy for only one agent at a time. But since the algorithm is iterative, it converges to a local optimum. In potential games, this converges to Nash equilibrium [3](Theorem 19.12). However, the IBR algorithm does not guarantee a global Nash equilibrium. Depending on the initial profile of strategies and the number of iterations, the algorithm may converge to a local Nash equilibrium. But if the potential function is convex or concave, the local optimum is global as well, and this algorithm always finds a global NE. In addition, compared to the following Double Oracle algorithm, IBR searches for equilibrium on a set of pure strategies, which makes it more efficient.

2.2.2 Double Oracle

Sometimes, even if we have all the time in the world and the most powerful computer, it is impossible to go through all possible variants to find the Nash equilibrium. This happens, for example, in continuous games(or in problems where the equilibrium is a mixed strategy), where the set of moves is infinite. Double Oracle algorithm [4–5] tries to reduce an infinite number of moves to a finite game, and it uses optimization methods to find a new finite game. This approach provides high efficiency for finding a NE.

How does it work? In the first game, each player makes a random move M_1. We can represent each player's strategy as choosing M_1 with a probability of 100%. Then in the following games, everyone does three steps:

- try to find a pure strategy (M_i) based on the previous strategy of the opponents
- add M_i to the subgame G^*
- solve this finite subgame(find mixed strategy)

These iterations are repeated in the cycle N times or until the change in the profit of all players is greater than the ϵ.

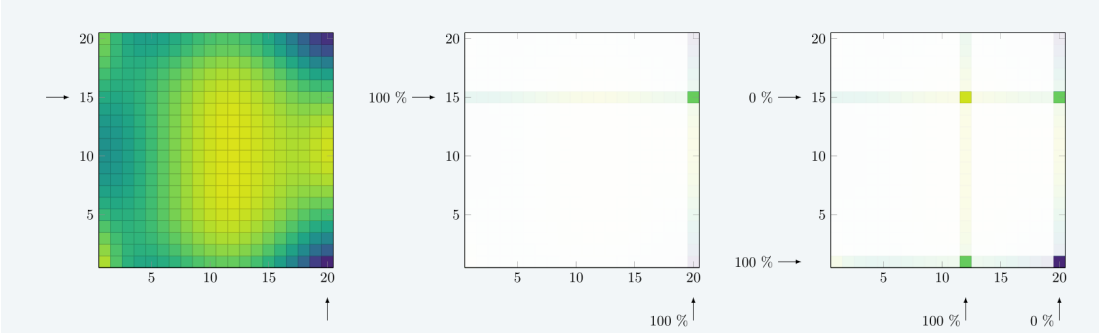

Figure 2.1. [6] Two iterations of Double Oracle.

Although the Double Oracle algorithm is a powerful tool for finding Nash equilibria, it has several drawbacks:

1. Computational complexity: the Double Oracle algorithm requires a lot of computational resources to determine each player's optimal strategies. This can be a problem in games with a large number of players or in games with a large number of possible actions. This is especially noticeable in problems with pure NE, which this algorithm solves through the space of mixed strategies.
2. Initial point sensitivity: the Double Oracle algorithm can be sensitive to the initial point.

Despite these weaknesses, the Double Oracle algorithm is still one of the most efficient and powerful tools for finding Nash equilibria in strategic games with complete information.

Chapter 3
Estimation from strategic Data sources

We try to solve the existing problem [1], verify the experimental results, and later extend it to the Bayesian game. However, we look at it from the other side and try to find an equilibrium using algorithms to find NE.

3.1 Problem definition

3.1.1 Original game

Briefly repeat what the task is. We want to build a model based on data from users, but it contains noise. We want to find this distortion through Nash equilibrium. Therefore, the main task is to study how users behave.

There are N agents ($N = \{1, \dots, n\}$). The text assumes that the agent has public features x_i. The accuracy with which they give their private data is λ_i. This is actually the main variable influenced by the participants. The given noise can obtain values in the range $[0, 1/\sigma^2]$. Each agent $i \in N$, choosing the amount of distortion, tries to minimize the loss function

$$J_i(\lambda_i, \lambda_{-i}) = c_i(\lambda_i) + f(\boldsymbol{\lambda}).$$

Where $c_i : \mathbb{R}_+ \to \mathbb{R}_+$ is the disclosure cost (the price an agent loses by disclosing its data) and $f : \mathbb{R}_+^n \to \mathbb{R}_+$ is the estimation cost (value of loss for inaccurate recommendations received). Therefore, the Nash equilibrium is

$$\lambda_i^* \in \min J_i(\lambda_i, \lambda_{-i}^*), \qquad \text{for all } i \in N.$$

For the reason that the game is potential, there is the potential function $\Phi : [0, 1/\sigma^2]^n \to \mathbb{R}$ such that

$$\Phi(\boldsymbol{\lambda}) = f(\boldsymbol{\lambda}) + \sum_{i \in N} c_i(\lambda_i), \qquad (\boldsymbol{\lambda} \in [0, 1/\sigma^2]^n).$$

3.1.2 Bayesian game

The previous game assumes complete information, that is, agents know the loss function of other participants. We extend this problem to a Bayesian setting by adding type space and common prior probability over types. Now the game is

$$\Gamma = \left\langle N, A, \Theta, p, (J_i)_{i \in N} \right\rangle. \tag{1}$$

- The set of players $N = \{1, \dots, n\}$
- The action set is $A_i = [0, 1]$, and $A = A_1 \times \cdots \times A_n$
- The set of types for each $i \in N$ is $\Theta_i = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$
- The type space $\Theta = \Theta_1 \times \cdots \times \Theta_n$ is endowed with a strictly positive probability distribution p: $\Theta \to [0, 1]$
- $p(\theta) = \frac{1}{|\Theta_i|^n}$
- The loss function J_i: $A \times \Theta \to \mathbb{R}$ is given by

$$J_i\left(\mathbf{a}, \boldsymbol{\theta}\right) = c_i\left(a_i\right) + f(diag(\boldsymbol{\theta}) \cdot \mathbf{a})$$

The first type means that a player is completely indifferent to the accuracy of the model (its argument in function f is four times smaller, so it is only interested in *disclosure cost*), it only worries about its privacy. The last type characterizes a player as conservative, trying not to risk the accuracy of the results, and more disposed to disclose its data. The prior probability is a constant and has a uniform distribution.

We can represent our new function f as $f(Ax + b)$ where $A = diag(\boldsymbol{\theta})$, $b = \vec{0}$ and $x = \mathbf{a}$. Since convexity is invariant under an affine map, $f(diag(\boldsymbol{\theta}) \cdot a)$ is still convex. Therefore J_i is convex as well (because the sum of the convex functions gives the convex function).

Recall that a (pure) strategy of player $i \in N$ in a Bayesian game is a mapping $s_i \colon \Theta_i \to A_i$. Note that each s_i is just a vector in $[0, 1]^4$. Let $s = (s_1, \ldots, s_n)$ be the profile of such strategies. Player's i ex-post loss $l_i(s, \theta)$ is then defined by

$$\begin{aligned} l_i(s, \theta) &= J_i(s(\theta), \theta), \qquad \theta \in \Theta \\ \text{where } s(\theta) &= (s_1(\theta_1), \ldots, s_n(\theta_n)) \in A. \end{aligned}$$

We define the ex-ante loss function L_i of player i by

$$L_i(s) = \sum_{\theta \in \Theta} p(\theta) \cdot l_i(s, \theta)$$

for any strategy profile s. A strategy profile s^* is called a Bayes-Nash equilibrium if

$$L_i(s^*) \leq L_i(s^*_{-i}, s_i)$$

for every $i \in N$ and every strategy s_i.

3.2 Experiments

In this section, we demonstrate how the algorithms work with a few examples. For visualization, we make graphs. Specifically for Double Oracle, we calculate instability

$$c_i(a_i^{j+1}, s^j_{-i}) - c_i(s^j)$$

where j is an iteration number in the algorithm. It is the difference between the result of a pure strategy and a mixed strategy. It provides a simple metric to check the efficiency of the approximation. In other cases, we calculate the difference in the profit of players or precision at equilibrium. For all algorithms, the maximum number of iterations is set to 10.

3.2.1 Environment and implementation

All the codes were run on a laptop with the Ubuntu 20.04 LTS subsystem. The laptop has processor Intel i5-1135G7 and 8 GiB of system memory to perform our experiments. All the algorithms were implemented in the Julia programming language. We use Julia 1.8 and JuMP. We also use optimization methods to find optimums using the library Ipopt [7]. Implementations of all algorithms are available on Git [x] [1]

[1] `https://gitlab.fel.cvut.cz/lariovia/nash-equilibria-for-regression-models-over-strategic-data.git`

3.2.2 Double oracle

First, we run tests to make sure that our algorithm works correctly. Example 1 is the game Cournot oligopoly [8]. Cost function is

$$c_i(a_1, a_2) = \begin{cases} -a_i \cdot (d - a_1 - a_2 - c) & a_1 + a_2 \leq d \\ a_i \cdot c & \text{otherwise} \end{cases}$$

where $d = 10$ and $c = 1$. And strategy set is the interval between 1 and 10. Using 10 steps our algorithm returns $a_1 \approx 3.004$ and $a_2 \approx 3.002$.

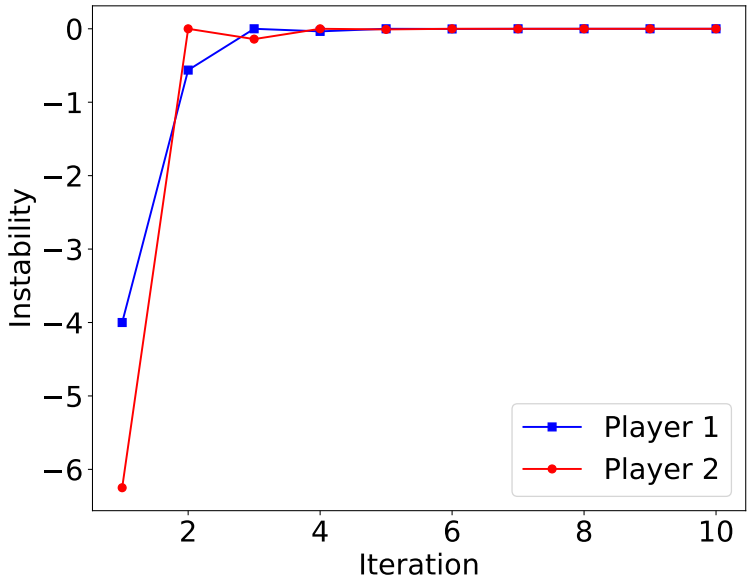

Figure 3.1. Convergence in Cournot Oligopoly.

Another example is the Torus game [9]. It is a two-player game. Each strategy set is the unit circle $S^1 = [-\pi, \pi]$ and the cost functions are

$$c_1(\theta_1, \theta_2) = -\alpha_1 \cos(\theta_1 - \phi_1) + \cos(\theta_1 - \theta_2),$$
$$c_2(\theta_1, \theta_2) = -\alpha_2 \cos(\theta_2 - \phi_2) + \cos(\theta_2 - \theta_1)$$

where $\phi = (0, \pi/8)$ and $\alpha = (1, 1.5)$. Using Ipopt as the best response oracle, our method returns one of two NE: pure strategies $\theta_1 \approx 1.408, \theta_2 \approx -0.325$ or $\theta_1 \approx -1.063, \theta_2 \approx 1.015$.

We have checked on two games that our Double Oracle works correctly and stable, now we can move on to solve our problem.

Example 1. The game is the tuple

$$\Gamma = \left\langle N, [0, 1/\sigma^2]^n, (J_i)_{i \in N} \right\rangle.$$

We consider a 1-dimensional model ($d = 1$) with two agents ($n = 2$). We assume that the disclosure cost of Agent 1 is $c_1(\lambda) = \lambda^{1.01}$ while the disclosure cost of Agent 2 is $c_2(\lambda) = \lambda^{20}$. The scalarization function is the identity, which means that $f(\boldsymbol{\lambda}) = 1/(\lambda_1 + \lambda_2) + \delta/\lambda_1 + \delta/\lambda_2$, where $\delta = 0.01$. We set the maximal precision to $1/\sigma^2 = 1$. Payoff function is

$$J_i(\lambda_i, \lambda_{-i}) = c_i(\lambda_i) + f(\boldsymbol{\lambda}). \tag{2}$$

Using Ipopt as the best response oracle, our method returns pure strategies $\lambda_1 \approx 0.248, \lambda_2 \approx 0.845$.

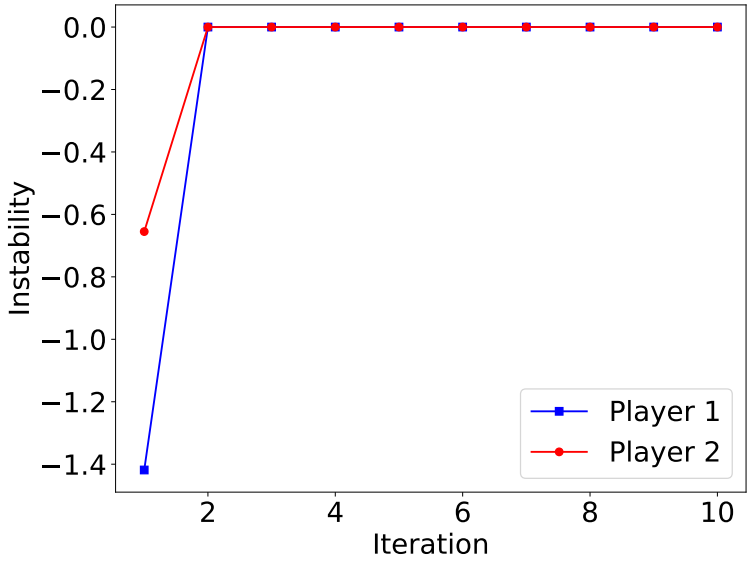

Figure 3.2. Convergence in Torus game.

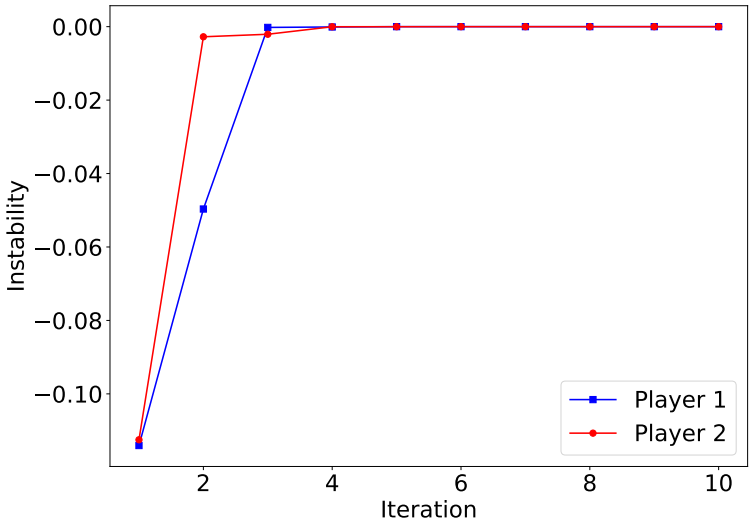

Figure 3.3. Double Oracle convergence.

3.2.3 Iterated Best Response

The first example we start with is the previous one. We try to solve the same game, only using the Iterated Best Response algorithm. The answer is the same ($\lambda_1 \approx 0.248, \lambda_2 \approx 0.845$), but the algorithm does it in fewer iterations on average 3.4.

Example 2. We try to complicate this game by increasing the dimension and the number of players. Our game consists of $d+1$ players and d dimensions. The disclosure cost of the first d agents are $c_i(\lambda) = \lambda^{20}$ (for $i \in \{1, \ldots, d\}$) and the disclosure cost of the last agent is $c_{d+1}(\lambda) = \lambda^{1.5}$. The maximal precision is the same ($1/\sigma^2 = 1$). Note that our original estimation cost function is

$$\begin{aligned} f(\boldsymbol{\lambda}) &= F(V(\boldsymbol{\lambda})), & F(V) &= \operatorname{trace}(V) \\ V(\boldsymbol{\lambda}) &\equiv (X^T \Lambda X)^{-1} + D^T \Lambda^{-1} D, & \Lambda &= \operatorname{diag}(\boldsymbol{\lambda}) \end{aligned}$$

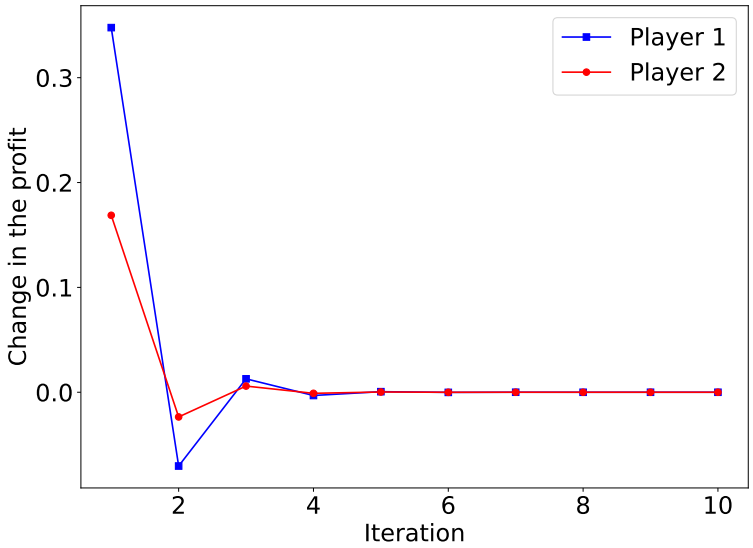

Figure 3.4. IBR convergence.

Therefore in this example, we should define X (the public features) and D (perturbation) matrices:

$$X = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \\ 1/d & \cdots & 1/d \end{bmatrix}, \qquad D = \begin{bmatrix} \sqrt{\delta} & 0 & 0 & \cdots \\ \vdots & 0 & 0 & \cdots \\ \sqrt{\delta} & 0 & 0 & \cdots \\ -d\sqrt{\delta} & 0 & 0 & \cdots \end{bmatrix}.$$

Below there are the Nash equilibria for different dimensions. Despite the increased number of agents and dimensions, the algorithm also converges quickly.

$(d = 2, \delta = 0.001)$ The NE is $\lambda_1 \approx 0.862, \lambda_2 \approx 0.862, \lambda_3 \approx 0.190$

$(d = 5, \delta = 0.00001)$ The NE is $\lambda_{1-5} \approx 0.866, \lambda_6 \approx 0.053$

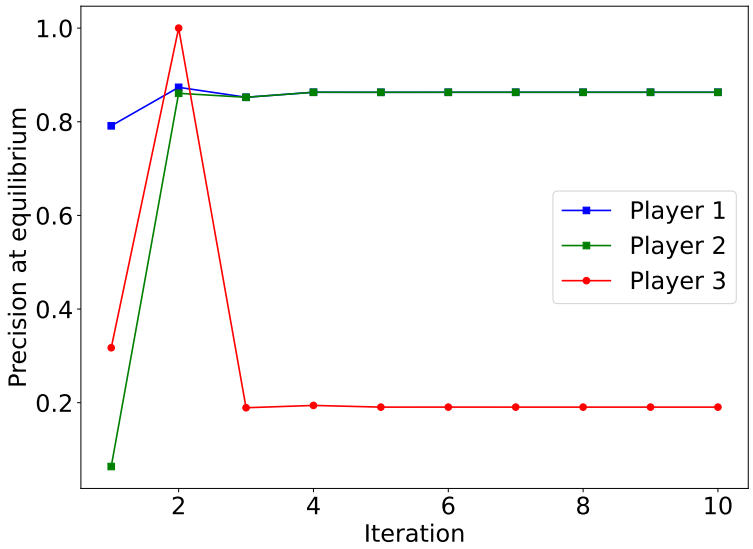

Figure 3.5. IBR convergence ($d = 2$).

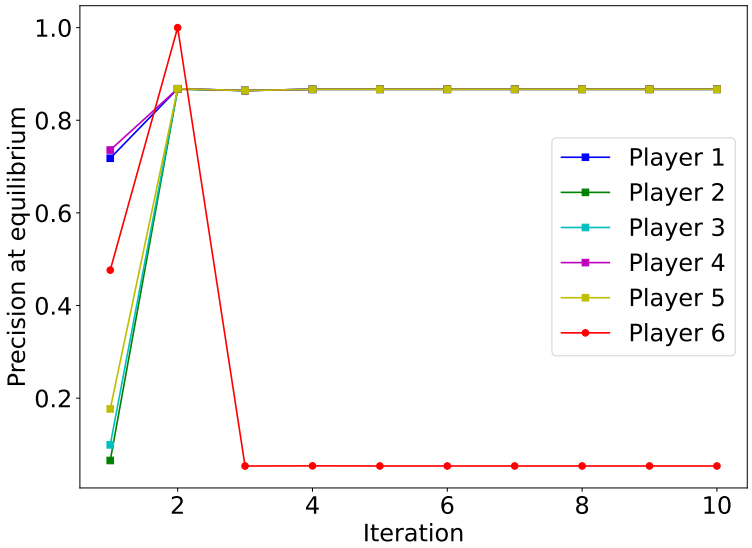

Figure 3.6. IBR convergence ($d = 5$).

3.2.4 Verification of experiments

We verify our results by finding a solution using optimization methods. Simultaneously, we try to repeat the results and experts achieved in [1]. Since our game is potential, we can find a Nash equilibrium using optimization methods. Also, since our loss function is convex, the local optimum is global. Based on this, we can discover equilibria using Ipopt (Nonlinear solver for Julia).

Example 1. Using the same settings the optimum of function (2) is $\lambda_1 \approx 0.248$, $\lambda_2 \approx 0.846$. Then we plot how the optimum and estimation cost behaves depending on perturbation 3.7.

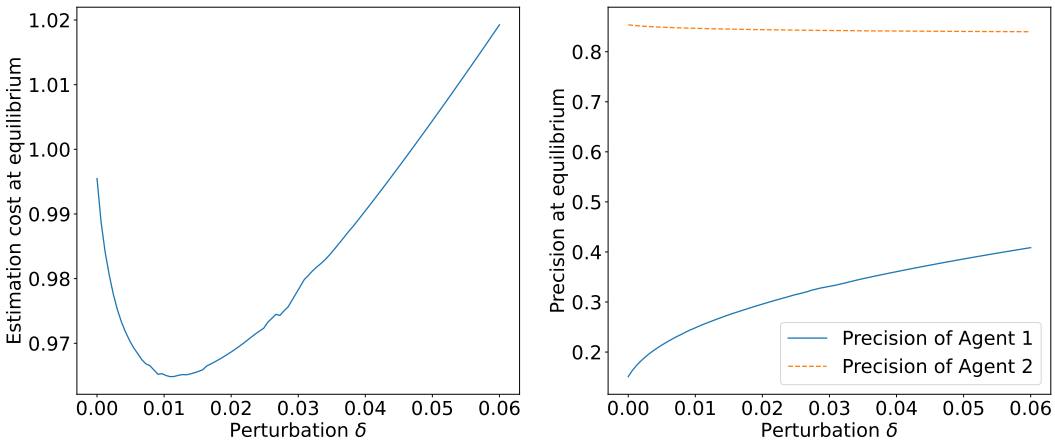

Figure 3.7. Example 1. Estimation cost (left) and precision of agents (right).

Example 2. Similarly, as in the previous example, we find the optimum, compare it, and draw plots.

$$\lambda_1 \approx 0.862, \lambda_2 \approx 0.862, \lambda_2 \approx 0.190 \qquad \text{for } d = 2, \delta = 0.001$$
$$\lambda_{1-5} \approx 0.866, \lambda_6 \approx 0.052 \qquad \text{for } d = 5, \delta = 0.00001$$

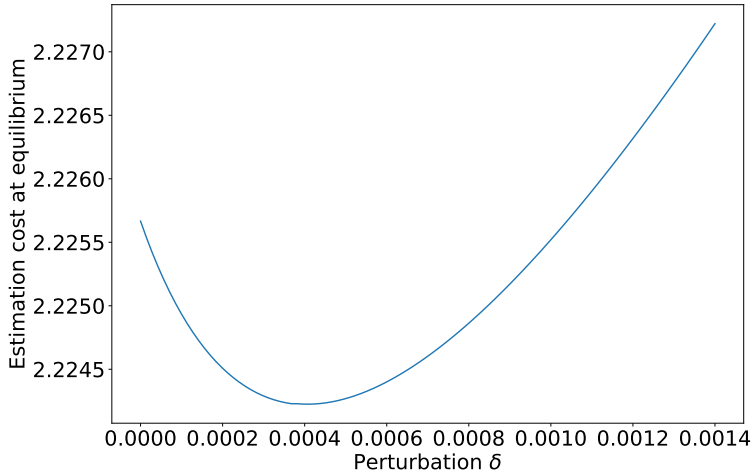

Figure 3.8. Example 2. Estimation cost for d = 2.

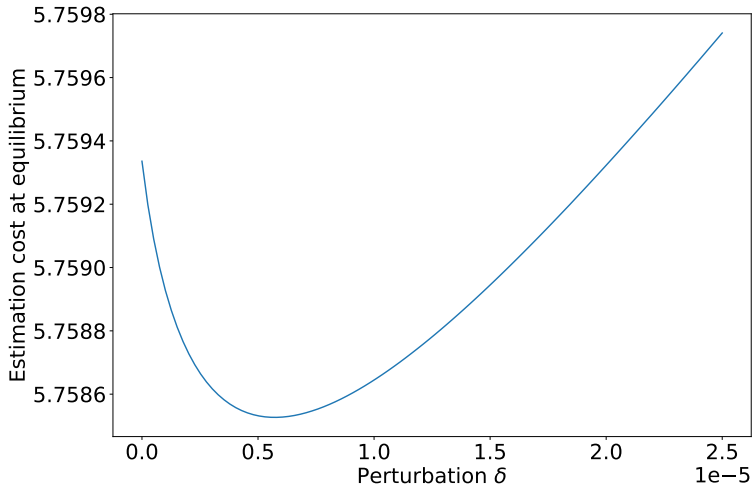

Figure 3.9. Example 2. Estimation cost for d = 5.

We confirmed the experiments obtained in this paper [1]. It also shows that all our algorithms work correctly.

3.2.5 IBR for Bayesian setting

In this part, we consider how agents would behave in the case of a Bayesian game. How much this setup affects their strategy. We find a Nash equilibrium in the game (1) using the algorithm Iterated Best Response.

First, we describe the IBR algorithm for the Bayesian setting. There are minor changes.

- In order to find an equilibrium, start with a randomly generated strategy profile $s^0 = (s_1^0, \ldots, s_n^0) \in [0,1]^{4n}$
- For each player $i \in N$, compute the best response s_i^1 to the strategy s_{-i}^0,

$$s_i^1 \in \arg\min_{s_i} L_i(s_i, s_{-i}^0)$$

- Repeat
- This generates a sequence of profiles $s^0, s^1, s^2, \ldots$

- Its limit should be a Bayesian Nash equilibrium

The graph 3.10 shows that the most selfish agents are now less likely to distort data about themselves. And also, the strategies do not vary greatly depending on the type of player.

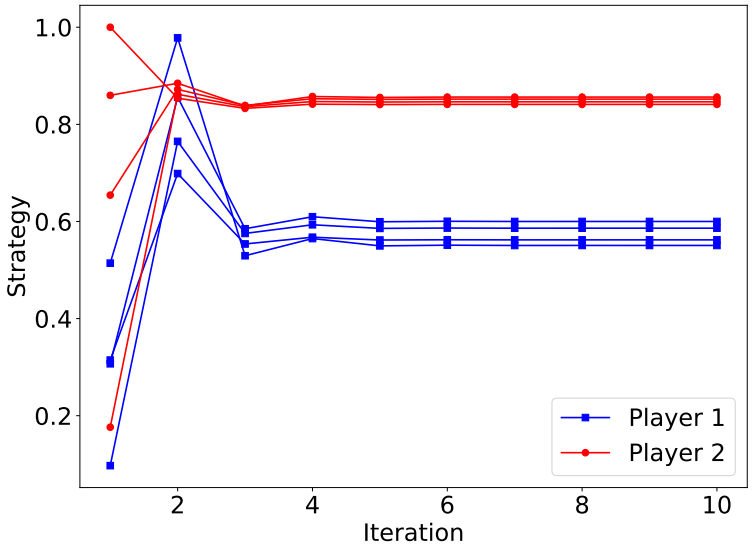

Figure 3.10. Convergence in the Bayesian game.

Chapter 4
Conclusion

4.0.1 Verification

We have implemented several algorithms to find the Nash equilibrium and tested their correctness on games we know the solutions to. Experiments have confirmed what we know theoretically. The Double Oracle and Iterated Best Response algorithms both converge to local optima. Sometimes, it is necessary to change the initialization to find a whole set of solutions. In the case of potential games with a convex or concave function, it is enough to run the algorithm once (since the local optimum is global). In most cases, only a few iterations are enough to find the Nash equilibrium.

After ensuring the algorithms worked correctly, we have repeated the experiments from the paper [1]. We have approached this problem from different sides. We've looked at it as an optimization problem and as a game in normal form (Double Oracle), including using the fact that this is a potential game (IBR). As a result of all the experiments, we have received the same numerical answers.

4.0.2 Expanding the problem

We've extended the problem to the Bayesian settings with incomplete information to bring it closer to the real world and experimented with numerical methods to find equilibria.

References

[1] Nicolas Gast, Stratis Ioannidis, Patrick Loiseau, and Benjamin Roussillon. Linear Regression from Strategic Data Sources. *ACM Transactions on Economics and Computation*. 2020, 8 (2), 1–24. DOI https://doi.org/10.1145/3391436.

[2] Yoav Shoham, and Kevin Leyton-Brown. *Multiagent Systems Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008. ISBN 9780511811654.

[3] Noam Nisan. *Algorithmic game theory*. Cambridge ; New York: Cambridge University Press, 2007. ISBN 9780521872829.

[4] H. Brendan McMahan, Geoffrey J Gordon, and Avrim Blum. *Planning in the Presence of Cost Functions Controlled by an Adversary*. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*. 2003.

[5] Lukáš Adam, Rostislav Horčík, Tomás Kasl, and Tomáš Kroupa. *Double Oracle Algorithm for Computing Equilibria in Continuous Games*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021. 5070–5077.

[6] Tomáš Kroupa. *Pursuit-evasion games I*. `https://cw.fel.cvut.cz/b221/_media/courses/uir/lectures/b4m36uir-lec09-slidestk.pdf`. 2022.

[7] Andreas Wächter, and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*. 2005, 106 (1), 25–57. DOI https://doi.org/10.1007/s10107-004-0559-y.

[8] Julio González-Díaz, Ignacio García-Jurado, and M. Gloria Fiestras-Janeiro. *An Introductory Course on Mathematical Game Theory*. American Mathematical Soc., 2010. ISBN 9780821872024.

[9] Benjamin Chasnov, Lillian Ratliff, Eric Mazumdar, and Samuel Burden. *Convergence Analysis of Gradient-Based Learning in Continuous Games*. `http://proceedings.mlr.press/v115/chasnov20a/chasnov20a.pdf`.

Appendix A
Assignment

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Larionov Viacheslav** Personal ID number: **487350**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Specialisation: **Artificial Intelligence and Computer Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Nash Equilibria for Regression Models over Strategic Data

Bachelor's thesis title in Czech:

Nashova ekvilibria pro regresní modely nad strategickými daty

Guidelines:

The game-theoretic framework [1] extends a standard linear regression model to the setting where data are generated by indviduals who can control (for privacy reasons) the precision of the output to be revelead to a statistician. Specifically, this scenario is modeled as a multiplayer strategic game in which strategy sets are real intervals and convex loss functions express the disclosure and estimation costs associated with the choice of precision by the players. The goal of the bachelor thesis is to verify the experimental results achieved in [1] and, if possible, to extend the existing model beyond current assumptions. In particular, the student will formulate and implement in Julia the tasks a)-c) listed below.
a) Compute the Nash equilibrium of regression game using the fact that it is a potential game [2].
b) Use a multiple oracle algorithm and the best response dynamics [3] to find the same equilibria and evaluate the performance of those numerical methods.
c) (optional!) Generalize the existing model beyond linear regression and/or convex loss functions.

Bibliography / sources:

[1] Gast, N., Ioannidis, S., Loiseau, P. & Roussillon, B. Linear Regression from Strategic Data Sources. ACM Trans. Econ. Comput. 8, 1–24 (2020)
[2] Monderer, D. & Shapley, L. S. Potential Games. Games Econ. Behav. 14, 124–143 (1996)
[3] William H. Sandholm. 2010. Population Games and Evolutionary Dynamics. The MIT Press.

Name and workplace of bachelor's thesis supervisor:

doc. Ing. Tomáš Kroupa, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **23.01.2023** Deadline for bachelor thesis submission: **26.05.2023**

Assignment valid until: **22.09.2024**

doc. Ing. Tomáš Kroupa, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. Date of assignment receipt

Student's signature