DSC Capstone LaTeX Template

Ryan Batubara rbatubara@ucsd.edu
Ivy Hawks mahawks@ucsd.edu
Darren Ho dah103@ucsd.edu
Ciro Zhang ciz001@ucsd.edu

Shachar Lovett slovett@ucsd.edu

Abstract

TBA

Repository: https://github.com/rybplayer/DSCCapstone

1 Introduction . . . 2
2 Methods . . . 2
3 Background . . . 2
4 Foundational Problems . . . 2
5 Case Study: Pure Nash Equilibrium . . . 3
6 Broad Applications . . . 5
7 Results . . . 5
8 Discussion . . . 5
9 Conclusion . . . 5
10 LaTeX Typesetting Examples . . . 5
References . . . 7
Appendices . . . A1

1 Introduction

2 Methods

3 Background

The field of game theory studies the mathematical models of interactions between rational decision makers. Game theory initially addressed the interactions between two parties in a zero sum game but has grown into an umbrella term on rational decision making.

4 Foundational Problems

4.1 Algorithmic Mechanism Design

Algorithmic mechanism design (AMD) focuses on designing a mechanism where (a) each party behave in a certain way and (b) is computationally efficient. AMD differs from regular mechanism design in that mechanism design assumes infinite compute power. AMD seeks to find a mechanism that performs with certain constraints such as polynomial time.

4.1.1 Vickrey Auction

Vickrey Auction is a sealed-bid second-place auction, where the highest bidder pays the second highest price. This is a classical example of AMD because it is in the best interest of each party to bid truthfully.

Say we have three parties participating in this auction, Alice, Bob, and Charlie. Their sealed bids are:

- Alice: $50
- Bob: $100
- Charlie: $70

In this scenario, Bob wins and pays $70.

Bob cannot improve his results by changing his bid, if he raises it the price he pays does not change, and if he lowers it he risks not winning the bid. Hence in this game, there is a dominant strategy to bid truthfully.

4.2 Nash Equilibrium

Nash Equilibrium are a foundational problem within game theory that involve finding a state within a scenario where no player can improve their outcome by unilaterally changing their

own strategy.

4.2.1 The Prisoner's Dilemma

The prisoner's dilemma is one of the simplest examples of finding a **Nash Equilibrium**.

Two suspects, Prisoner A and Prisoner B, are arrested and interrogated separately. Each prisoner has two possible actions:

- **Cooperate (C)**: remain silent
- **Defect (D)**: betray the other prisoner

The payoff matrix (with payoffs in the form (A's payoff, B's payoff)) is given by:

	B: C	B: D
A: C	$(1,1)$	$(3,0)$
A: D	$(0,3)$	$(2,2)$

The interpretation is:

- If both stay silent (C,C), each receives a light sentence: 1.
- If one defects and the other cooperates, the defector goes free (0) while the cooperator receives the harshest sentence (3).
- If both defect (D,D), they each receive a moderate sentence (2).

Even though mutual cooperation leads to a better outcome than mutual defection, the dominant strategy for each prisoner is to defect, making (D, D) the unique Nash equilibrium.

5 Case Study: Pure Nash Equilibrium

We will be looking to see what the communication complexity of reaching equilibrium is. More specifically, the communication model in which players initially only know their own utility functions. We want to analyze how much information must be transferred between them to jointly compute the equilibrium point. We also assume that each player follows a predetermine protocol, abstracting any other incentives of each player. We will be focusing on the communication complexity of *pure* Nash Equilibrium. A pure Nash Equilibrium being a situation where there is guaranteed to be a unique best input for a all given parties.

5.1 Setting for Pure Action Games

There are $n \geq 2$ players, $i = 1, 2, ..., n$. Each player i has a finite set of actions A_i with $|A_i| \geq 2$. For the analysis of pure action games, we only consider binary action games, that is, for each i, $A_i = \{0, 1\}$. Let the joint action space $A = \prod_{i=1}^{n} A_i$. Each player has a private utility function $u_i : A \longrightarrow \{0, 1\}^n$ which we are assuming are finitely represented for simplicity.

For a joint action $a = (a_1, ..., a_n) \in A$ (for binary action games, a joint action is an n bit binary string), let a^{-1} denote the joint action of all players except player i. A joint action a is a **Pure Nash Equilibrium** if $u_i(a) \geq u_i(b_i, a^{-1})$ for every player i and any action $b_i \in A_i$. That is to say that for all players choosing the alternative option would result in an equal or worse outcome for them.

5.2 Communication Complexity of Pure Nash Equilibrium

Theorem *Any pure Nash equilibrium procedure has communication complexity* $\Omega(2^n)$

Claim If there exists a reduction from the S-disjointness problem to an n-person pure Nash Equilibrium procedure that satisfies reducibility and constructibility properties then any pure Nash Equilibrium procedure has communication complexity of at least $|S|$ bits since it has been proven that $CC(DISJ_S) = |S|$.

We want to prove that there exists a reduction from the set disjointness problem where:

$$S_1 \cap S_2 = 0 \text{ or } S_1 \cap S_2 \neq 0$$

Our goal is to relate the disjointness problem into a game that has a pure Nash equilibrium iff the sets intersect.To do so, we use the *matching pennies reduction*.

Matching Pennies. The matching pennies game is a 2×2 zero-sum game played between two players. Each player chooses either *Heads* (H) or *Tails* (T). If the actions match, player 1 wins; if they mismatch, player 2 wins. The payoff matrix for player 1 is:

	H	T
H	1	-1
T	-1	1

and player 2 receives the negative of this payoff.

A key property of matching pennies is that it has *no pure Nash equilibrium*. This makes it useful as a "destabilizing gadget" in reductions, since attaching a matching-pennies subgame to an action profile guarantees that the profile cannot be a pure equilibrium.

For $n \geq 4$, the matching pennies reduction satisfies the reducibility and constructibility properties.

We first verify the *constructibility* property. Fix $\alpha \in \{1, 2\}$. By the definition of the matching pennies reduction, the payoff function $u_{\alpha,i}(a)$ of every player (α, i) depends only on the following information:

- whether $a \in S_\alpha$, and
- the actions $a_{h,1}$ and $a_{h,2}$ in case $a \notin S_h$

Since S_α is the private input of the agents on side α in the disjointness problem, the event $a \in S_\alpha$ is computable by those agents, and $a_{\alpha,1}, a_{\alpha,2}$ are part of the observed joint action a. Hence each payoff $u_{\alpha,i}(a)$ is computable from (a, S_α, i) alone. This establishes constructibility.

We now prove the *reducibility* property. We must show that

$$S_1 \cap S_2 \neq \emptyset \qquad \iff \qquad G \text{ has a pure Nash equilibrium.}$$

Suppose $a \in S_1 \cap S_2$. Then by construction, every player in T_1 and T_2 receives payoff 2 at a, which is the maximal payoff any player can obtain. Thus no player can improve by deviating, and a is a pure Nash equilibrium.

Suppose $a \notin S_\alpha$ for some $\alpha \in \{1, 2\}$. Then the two designated players $(\alpha, 1)$ and $(\alpha, 2)$ play a matching pennies game at a. The matching pennies game has no pure Nash equilibrium: at every pure action profile of the two players, one of them strictly benefits from unilaterally deviating. Therefore, at a at least one of the players $(\alpha, 1)$ or $(\alpha, 2)$ has a profitable deviation, implying that a cannot be a pure Nash equilibrium of G. Hence no pure Nash equilibrium can lie outside $S_1 \cap S_2$.

Combining the two directions, we conclude that a pure Nash equilibrium exists in G if and only if $S_1 \cap S_2 \neq \emptyset$.

Since we have proven the claim, we know that the communication complexity of any pure Nash Equilibrium is $CC(DISJ_s) = |S|$. Remember that $S = \{0, 1\}^n$ so therefore $|S| = 2^n$ proving the theorem.

6 Broad Applications

The communication complexity of Game Theory has many varied applications and is prevalent in any situation that involves optimizing the output in any multiparty game. Game Theory draws naturally has parallels to other fields such as multi-agent and distributed systems.

7 Results

8 Discussion

9 Conclusion

10 LaTeX Typesetting Examples

This is not a real section; it's just here to show examples of how to format various components. Remove it before submitting!

10.1 LaTeX Basics

Here's **bold** and *italicized* text. Here's `text_that_looks.like(code)`.

- Here's a regular bulleted list item.
- And another.

Here's a hyperlink. If you want to use a numbered list, you can experiment with:

1. This.
2. This.
3. And this.

Here's how you might include a snippet of actual code:

```
# If you want to use syntax highlighting, look into the minted package.
def f(x):
    return 2 * x + 3
```

Here's how you might format a single equation:

$$\int_{-\infty}^{\infty} f_X(x)dx = 1$$

And a chain of equations:

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 &= \frac{1}{n}\sum_{i=1}^{n}(x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - \frac{2}{n}\bar{x}\sum_{i=1}^{n}x_i + \frac{\bar{x}^2}{n}\sum_{i=1}^{n}1 \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - 2\bar{x}^2 + \bar{x}^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}x_i^2 - \bar{x}^2
\end{aligned}$$

10.2 Figure Examples

Here are some example figures. Figure **??** presents a scatter plot.

Figure 1: Yes, put a few words or sentences here explaining what is in the figure.

Figure 2 presents some summaries of the performance of our model. The left panel of Figure 2 presents something. The right panel of Figure 2 presents some other things.

Figure 2: You can put figures side-by-side as well.

10.3 Table Examples

Table **??** presents some summary of the data.

Table 1: Some Table Caption

Table **??** presents some summaries of the performance of our model.

Table 2: Some Other Table Caption

10.4 Equations and Algorithms Examples

Algorithm 1 implements Fuzzy K-means.

Algorithm 2 calculates net activation.

In Variational Autoencoder (VAE), we directly maximize the Evidence Lower Bound (ELBO) using the following Equations 2–4.

$$\mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q_{\phi}(\boldsymbol{z}\mid\boldsymbol{x})}\right] = \mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_{\theta}(\boldsymbol{x}\mid\boldsymbol{z})p(\boldsymbol{z})}{q_{\phi}(\boldsymbol{z}\mid\boldsymbol{x})}\right] \tag{2}$$

$$= \mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\theta}(\boldsymbol{x}\mid\boldsymbol{z})] + \mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{z})}{q_{\phi}(\boldsymbol{z}\mid\boldsymbol{x})}\right] \tag{3}$$

$$= \underbrace{\mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}[\log p_{\theta}(\boldsymbol{x}\mid\boldsymbol{z})]}_{\text{reconstruction term}} - \underbrace{\mathscr{D}_{\mathrm{KL}}(q_{\phi}(\boldsymbol{z}\mid\boldsymbol{x})\,||\,p(\boldsymbol{z}))}_{\text{prior matching term}} \tag{4}$$

10.5 Inline Citation Examples

Citation in text (no parentheses): use `\cite{citekey}`. For example, **?**, **?**.

Citation in parentheses: use `\citep{citekey}`. For example: (**?**), (**?**).

To edit the contents of the "References" section, edit `reference.bib`. Many conference websites format citations in BibTeX that you can copy into `reference.bib` directly; you can also search for the paper on Google Scholar, click "Cite", and then click "BibTeX" (here's an example).

Algorithm 1: Fuzzy K-means clustering algorithm

1. Choose primary centroids v_k
2. Compute the membership degree of all feature vectors in all clusters

$$u_{ki} = \frac{1}{\sum_{j=1}^{K} \left(\frac{D^2(x_i - v_k)}{D^2(x_i - v_j)}\right)^{\frac{2}{m-1}}} \tag{1}$$

Algorithm 2: Computing Net Activation

Input: $x_1, \ldots, x_n, w_1, \ldots, w_n$
Output: y, the net activation
$y \leftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
    $y \leftarrow y + w_i * x_i$;

Appendices

A.1 Training Details A1
A.2 Additional Figures A1
A.3 Additional Tables A1

A.1 Training Details

A.2 Additional Figures

A.3 Additional Tables