Inverse Game Theory: An Incenter-Based Approach

Lvye Cui[1,2], **Haoran Yu**[*1], **Pierre Pinson**[3] and **Dario Paccagnan**[2]
[1]School of Computer Science & Technology, Beijing Institute of Technology
[2]Department of Computing, Imperial College London
[3]Dyson School of Design Engineering, Imperial College London
cui_lvye@outlook.com, yhrhawk@gmail.com, p.pinson@imperial.ac.uk, d.paccagnan@imperial.ac.uk

Abstract

Estimating player utilities from observed equilibria is crucial for many applications. Existing approaches to tackle this problem are either limited to specific games or do not scale well with the number of players. Our work addresses these issues by proposing a novel utility estimation method for general multi-player non-cooperative games. Our main idea consists in reformulating the inverse game problem as an inverse variational inequality problem and in selecting among all utility parameters consistent with the data, the so-called *incenter*. We show that the choice of the incenter can produce parameters that are most robust to the observed equilibrium behaviors. However, its computation is challenging, as the number of constraints in the corresponding optimization problem increases with the number of players and the behavior space size. To tackle this challenge, we propose a loss function-based algorithm, making our method scalable to games with many players or a continuous action space. Furthermore, we show that our method can be extended to incorporate prior knowledge of player utilities, and that it can handle inconsistent data, i.e., data where players do not play exact equilibria. Numerical experiments on three game applications demonstrate that our methods outperform the state of the art. The code, datasets, and supplementary material are available at https://github.com/cuilvye/Incenter-Project.

1 Introduction

Game theory analyzes the strategic behaviors of players using equilibrium concepts, while inverse game theory focuses on the inverse problem: *given the observed equilibrium behaviors of players, what are the possible player utilities leading to these behaviors?* Inverse game theory has garnered increasing attention [Waugh *et al.*, 2011; Kuleshov and Schrijvers, 2015; Ling *et al.*, 2019; Noti, 2021; Wu *et al.*, 2022], because understanding player utilities can lead to better-designed mechanisms and economic policies.

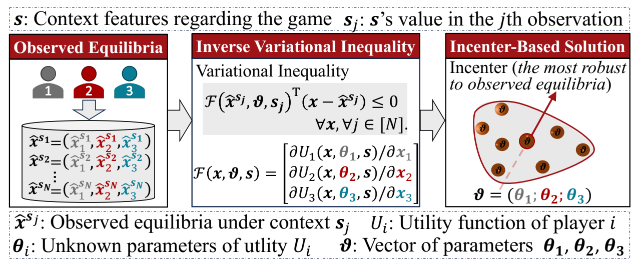


Figure 1: Our Approach to Utility Estimation from Equilibria.

When estimating player utilities, the existing approaches have two main limitations: (i) most methods are tailored to *specific games*, such as matrix games [Noti, 2021; Yu *et al.*, 2022] and attacker-defender security games [Blum *et al.*, 2014]; (ii) many existing methods *do not scale well* with the number of players, often focusing on two-player games [Bertsimas *et al.*, 2015; Tsai *et al.*, 2016; Ling *et al.*, 2018; Ling *et al.*, 2019; Wu *et al.*, 2022].

In this work, we study general multi-player non-cooperative games, and propose utility estimation methods applicable to a broad range of games and practical applications. Figure 1 illustrates our main ideas. The player utility functions are parameterized, and we aim to estimate the vector ϑ of these utility parameters under which the observed player behaviors constitute a Nash equilibrium. Our idea is to reformulate this as an inverse variational inequality problem: *given the observed player behaviors, what vector ϑ ensures that these behaviors satisfy the variational inequalities?*

The crucial challenge is that even with many observations of player behaviors, there may be multiple parameter vectors ϑ solving the inverse variational inequality problem. This raises a fundamental question: *how should we select a single ϑ from the set of ϑ consistent with the observed equilibria?* Existing approaches, such as [Bertsimas *et al.*, 2015], do not address this question but simply single out a ϑ without following a guiding principle. To the contrary, we attempt to solve this problem using a more principled approach.

Our key idea is to select the *incenter* among all consistent ϑ. This concept introduced in inverse optimization [Besbes *et al.*, 2023; Zattoni Scroccaro *et al.*, 2024] describes a point that lies within a given set and is located furthest away from

*Corresponding author.

its boundary. Due to its geometric properties, selecting the incenter as the estimated $\boldsymbol{\vartheta}$ reduces the error in estimating the true parameter vector $\boldsymbol{\vartheta}_{\text{true}}$.

We can efficiently compute the incenter of the set of all consistent $\boldsymbol{\vartheta}$ using convex optimization. When the action space is continuous, the optimization problem of finding the exact incenter is challenging to solve, since it may involve infinitely many constraints. We convert it to an unconstrained loss minimization problem by defining a novel loss function. Then, we propose a first-order algorithm to minimize the loss, and thus estimate the parameter vector $\boldsymbol{\vartheta}$.

Our loss function-based approach of approximating the incenter has the advantage of handling settings where players do not play exact equilibria. When players are bounded-rational, it is possible that no $\boldsymbol{\vartheta}$ satisfies all variational inequalities. In this case, our loss function still guides the search for $\boldsymbol{\vartheta}$ toward minimizing the degree of violating the constraints defined based on variational inequalities.

We further explore a class of games where the player utility functions exhibit a specific structural property (e.g., monotonicity of gradients). We extend our parameter estimation method to incorporate this property by leveraging techniques from *semidefinite programming*.

We summarize our contributions as follows.

- We propose a novel framework for solving inverse equilibrium problems, through integrating techniques from variational inequalities and inverse optimization. We further extend our framework to account for specific structural property of player utility functions.
- Our framework can be broadly applied to many multi-player non-cooperative games and traffic routing games in the transportation field. It imposes no restrictions on the number of players in these game scenarios.
- We conduct extensive numerical experiments on three game applications: Bertrand-Nash price competition [Narahari *et al.*, 2009], aggregative games [Jensen, 2018], and network traffic games [Patriksson, 2015]. Evaluation using different metrics demonstrates that our methods outperform the state of the art.

2 Related Work

Inverse Game Theory [Kuleshov and Schrijvers, 2015] is one of the pioneering studies in this field, and investigated the tractability of estimating player utilities in succinct games. Most subsequent studies addressed the inverse game problem within specific games, such as normal-form games [Ling *et al.*, 2018; Noti, 2021; Yu *et al.*, 2022], two-player zero-sum games [Ling *et al.*, 2019], and attacker-defender security games [Blum *et al.*, 2014; Haghtalab *et al.*, 2016; Wu *et al.*, 2022]. [Ling *et al.*, 2018] proposed an end-to-end parameter estimation framework that applies to two-player normal and extensive form games. Different from these studies, our methods are applicable to a broad range of non-cooperative games and applications, and do not impose restrictions on the number of players.

Our work is related to [Bertsimas *et al.*, 2015], which utilized inverse optimization to estimate utility parameters in Bertrand competition. That approach can handle only a small number of players and behavior observations (e.g., it requires dealing with over $120,000$ constraints when there are 4 players and 500 observations). As will be shown in experiments, our incenter-based method is more accurate and scalable.

Inverse Optimization The goal of inverse optimization [Heuberger, 2004; Chan *et al.*, 2023] is to identify the parameters of an optimization model that render observed decisions approximately or exactly optimal. To estimate the parameters, the literature in this field has proposed various loss functions, such as Predictability Loss [Aswani *et al.*, 2018], Suboptimality Loss [Mohajerin Esfahani *et al.*, 2018; Besbes *et al.*, 2023], Augmented Suboptimality Loss [Zattoni Scroccaro *et al.*, 2024], and Variational Inequality Loss [Bertsimas *et al.*, 2015]. Our work is inspired by [Zattoni Scroccaro *et al.*, 2024]. The main difference is that we focus on game scenarios with multiple decision makers, rather than optimization problems involving a single decision maker. We seek parameters that make the observed data equilibria, rather than the optimal solutions to an optimization problem.

3 Problem Formulation

We represent a multi-player non-cooperative game using a tuple $\mathcal{G} := \{\mathcal{I}, \{\mathcal{X}_i\}_{i\in\mathcal{I}}, \{U_i\}_{i\in\mathcal{I}}\}$. Here, $\mathcal{I} := \{1, \ldots, p\}$ denotes the set of players. Each player $i \in \mathcal{I}$ chooses an action $\boldsymbol{x}_i$ from its feasible action set $\mathcal{X}_i \subseteq \mathbb{R}^{m_i}$, where $m_i \in \mathbb{Z}^+$ denotes the dimension of the action. Let $\boldsymbol{x} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p) \in \mathcal{X}$ be the action profile, where $\mathcal{X} = \prod_{i\in\mathcal{I}} \mathcal{X}_i$ is the global action set. We consider the setting where the game is played in different *contexts*, representing exogenous (but observable) conditions, e.g., variations in GDP and seasonality in Bertrand-Nash games. Given a contextual feature $\boldsymbol{s} \in \mathcal{S}$, we assume each player i seeks to maximize its own utility function $U_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}, \boldsymbol{s}) : \mathcal{X} \times \mathcal{S} \to \mathbb{R}$. The utility depends on its own action $\boldsymbol{x}_i \in \mathcal{X}_i$, other players' actions $\boldsymbol{x}_{-i} \in \mathcal{X}_{-i}$, and the contextual feature $\boldsymbol{s}$. Here, $\mathcal{X}_{-i} := \prod_{j\in\mathcal{I}\setminus\{i\}} \mathcal{X}_j$.

A Nash equilibrium $\boldsymbol{x}^* := (\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_p^*) \in \mathcal{X}$ of the game is defined as a point at which no player can unilaterally increase its utility [Nash, 1950], i.e., $U_i(\boldsymbol{x}_i^*, \boldsymbol{x}_{-i}^*, \boldsymbol{s}) \geqslant U_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}^*, \boldsymbol{s}), \forall \boldsymbol{x}_i \in \mathcal{X}_i, i \in \mathcal{I}$ under the given $s \in \mathcal{S}$.

We make the following standard assumptions: (i) The feasible set $\mathcal{X}_i$ is a nonempty, closed, and convex subset of $\mathbb{R}^{m_i}$ for all $i \in \mathcal{I}$; (ii) Utility function U_i is continuously differentiable and pseudo-concave with respect to $\boldsymbol{x}_i$ for all $i \in \mathcal{I}$; (iii) The contextual feature $\boldsymbol{s}$ is publicly known to all players. Under these assumptions, a Nash equilibrium $\boldsymbol{x}^* \in \mathcal{X}$ exists in the game $\mathcal{G}$ [Nash, 1950].

Given a contextual feature $\hat{\boldsymbol{s}}^j$ of game $\mathcal{G}$, there exists a Nash equilibrium $\hat{\boldsymbol{x}}^j$. Let $\widehat{\mathcal{D}} = \{(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j)\}_{j=1}^N$ denote the observable dataset of all equilibrium-context pairs. Our target is to estimate the utility functions of all players from $\widehat{\mathcal{D}}$.[1]

We consider the case where $U_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}, \boldsymbol{s}; \boldsymbol{\theta}_i)$ can be parameterized by $\boldsymbol{\theta}_i \in \Theta_i$ for each i. In practice, the form of

[1]For example, in a price competition, each firm's utility is known only to itself, and depends on the prices of all firms and the economic condition. By observing the equilibrium prices under different economic conditions, we estimate the utility functions of all firms.

$U_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}, \boldsymbol{s}; \boldsymbol{\theta}_i)$ is often known, but $\boldsymbol{\theta}_i \in \Theta_i$ needs to be estimated from data. Let $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1; \cdots; \boldsymbol{\theta}_p)$ denote the parameter vector, i.e., the vertical concatenation of all $\boldsymbol{\theta}_i$. Our data-driven inverse game problem is as follows.

Problem 1. *Given dataset $\widehat{\mathcal{D}} = \{(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j)\}_{j=1}^N$, estimate parameters $\boldsymbol{\vartheta}$ that reproduce equilibrium $\hat{x}^j$ in context $\hat{s}^j$, $\forall j$.*

4 Incenter-Based Solution

4.1 Problem Reformulation

We begin by exploiting the fact that, under the assumption considered, $\boldsymbol{x}^* \in \mathcal{X}$ is a Nash equilibrium if and only if it solves the *variational inequality* [Harker and Pang, 1990]:

$$\sum_{i=1}^{p} \left(-\nabla_{\boldsymbol{x}_i} U_i(\boldsymbol{x}^*, \boldsymbol{s})\right)^\top (\boldsymbol{x}_i - \boldsymbol{x}_i^*) \geqslant 0, \forall \boldsymbol{x} \in \mathcal{X}. \quad (1)$$

Hence, Problem 1 can be equivalently reformulated as seeking a $\boldsymbol{\vartheta}$ that satisfies the following inequalities for all $j \in [N]$ (j indexes the context-equilibrium pairs):

$$\sum_{i=1}^{p} \left(\nabla_{\boldsymbol{x}_i} U_i\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j; \boldsymbol{\theta}_i\right)\right)^\top \left(\boldsymbol{x}_i^j - \hat{\boldsymbol{x}}_i^j\right) \leqslant 0, \forall \boldsymbol{x}^j \in \mathcal{X}^j. \quad (2)$$

In the following, we consider a general setting where the player action set $\mathcal{X}^j$ can also change with the contextual feature $\hat{\boldsymbol{s}}^j$. Following the existing literature, e.g., [Bertsimas *et al.*, 2015; Peters *et al.*, 2023; Maddux *et al.*, 2023], our work focuses on games where $U_i(\boldsymbol{x}, \boldsymbol{s}; \boldsymbol{\theta}_i)$ can be expressed as a linear combination of parameters $\boldsymbol{\theta}_i$ and functions $\boldsymbol{\varphi}_i(\boldsymbol{x}, \boldsymbol{s})$, i.e., $U_i(\boldsymbol{x}, \boldsymbol{s}; \boldsymbol{\theta}_i) = \langle \boldsymbol{\theta}_i, \boldsymbol{\varphi}_i(\boldsymbol{x}, \boldsymbol{s}) \rangle$. As will be shown in Section 5.1, many classic games fall into this category, including Bertrand-Nash price competition, aggregative games, and traffic games. For ease of presentation, let $\boldsymbol{x}_i \in \mathbb{R}$ and $\nabla_{\boldsymbol{x}_i} \boldsymbol{\varphi}_i(\boldsymbol{x}, \boldsymbol{s}) = \boldsymbol{\phi}_i(\boldsymbol{x}, \boldsymbol{s})$. Then, we can rewrite (2) as

$$\sum_{i=1}^{p} \left({\boldsymbol{\theta}_i}^\top \boldsymbol{\phi}_i\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)\right) \left(\boldsymbol{x}_i^j - \hat{\boldsymbol{x}}_i^j\right) \leqslant 0, \forall \boldsymbol{x}^j \in \mathcal{X}^j. \quad (3)$$

While there may exist (infinitely) many $\boldsymbol{\vartheta}$ satisfying (3), motivated by [Zattoni Scroccaro *et al.*, 2024], we propose to seek the *incenter* of all $\boldsymbol{\vartheta}$ satisfying (3). As will be explained in Section 4.2, this solution is most robust to perturbations of the observable dataset $\widehat{\mathcal{D}}$.

We first formally define the incenter in Section 4.2. We then introduce a novel loss function approach to estimate it and thus learn $\boldsymbol{\vartheta}$ in Sections 4.3 and 4.4. In Section 4.5, we extend our solution to incorporate prior knowledge about $\boldsymbol{\vartheta}$.

4.2 Incenter Parameter Vector

We define the set of parameter vectors consistent with the observed dataset $\widehat{\mathcal{D}}$ as follows.

Definition 1 (Consistent Parameter Vectors)**.** *Given feasible action sets $\{\mathcal{X}^j\}_{j=1}^N$, the dataset $\widehat{\mathcal{D}} = \{(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j)\}_{j=1}^N$, and functions $\{\boldsymbol{\phi}_i(\boldsymbol{x}, \boldsymbol{s})\}_{i \in \mathcal{I}}$, the set of consistent parameter vectors is defined as*

$$\mathbb{C} := \left\{\boldsymbol{\vartheta} : \langle \boldsymbol{\Phi}_{\boldsymbol{\vartheta}}^j, \boldsymbol{x}^j - \hat{\boldsymbol{x}}^j \rangle \leqslant 0, \forall \boldsymbol{x}^j \in \mathcal{X}^j, \forall j \in [N]\right\}. \quad (4)$$

Here, $\boldsymbol{\Phi}_{\boldsymbol{\vartheta}}^j = \left({\boldsymbol{\theta}_1}^\top \boldsymbol{\phi}_1\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right), \cdots, {\boldsymbol{\theta}_p}^\top \boldsymbol{\phi}_p\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)\right)^\top \in \mathbb{R}^p$.

In other words, $\mathbb{C}$ is the set of $\boldsymbol{\vartheta}$ satisfying (3). Geometrically, it forms a convex cone. We focus on searching for the incenter of $\mathbb{C}$, which is defined as follows.

Definition 2 (Incenter of $\mathbb{C}$)**.** *Given a nonempty set $\mathbb{C}$, its incenter $\boldsymbol{\vartheta}^{in}$ is defined as*

$$\boldsymbol{\vartheta}^{in} \in \arg\max_{\boldsymbol{\vartheta} \in \mathbb{C}} \min_{\tilde{\boldsymbol{\vartheta}} \in \overline{int(\mathbb{C})}} a\left(\boldsymbol{\vartheta}, \tilde{\boldsymbol{\vartheta}}\right). \quad (5)$$

Here, $\overline{int(\mathbb{C})}$ is the region excluding the interior of $\mathbb{C}$, and $a\left(\boldsymbol{\vartheta}, \tilde{\boldsymbol{\vartheta}}\right)$ is the angle between $\boldsymbol{\vartheta}$ and $\tilde{\boldsymbol{\vartheta}}$, i.e.,

$$a\left(\boldsymbol{\vartheta}, \tilde{\boldsymbol{\vartheta}}\right) = \arccos\left(\frac{\langle \boldsymbol{\vartheta}, \tilde{\boldsymbol{\vartheta}} \rangle}{\|\boldsymbol{\vartheta}\|_2 \|\tilde{\boldsymbol{\vartheta}}\|_2}\right).$$

Geometrically, an incenter $\boldsymbol{\vartheta}^{\text{in}}$ of $\mathbb{C}$ can be viewed as a vector furthest away from the boundary of $\mathbb{C}$, as measured by the angle. *Since each facet of $\mathbb{C}$ is determined by a $\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ pair in dataset $\widehat{\mathcal{D}}$, incenter $\boldsymbol{\vartheta}^{in}$ can be informally interpreted as the parameter vector that is most robust to perturbations of data $\widehat{\mathcal{D}}$ (i.e., perturbations of the facets of $\mathbb{C}$).* For a formal description of this property, see the supplementary material.

For simplicity, we define a column vector $\boldsymbol{\phi}^j := \left(\boldsymbol{\phi}_1\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right); \cdots; \boldsymbol{\phi}_p\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)\right)$. In the following theorem, we formulate the problem of finding an incenter $\boldsymbol{\vartheta}^{\text{in}}$.

Theorem 1 (Incenter Computation)**.** *When $\mathbb{C}$ is defined as (4) and its interior is nonempty, finding $\boldsymbol{\vartheta}^{in}$ that satisfies (5) is equivalent to solving the following problem:*

$$\begin{aligned} \min_{\boldsymbol{\vartheta}} \ & \|\boldsymbol{\vartheta}\|_2 \\ \text{s.t.} \ & \langle \boldsymbol{\Phi}_{\boldsymbol{\vartheta}}^j, \boldsymbol{x}^j - \hat{\boldsymbol{x}}^j \rangle + \left\|\boldsymbol{\phi}^j \odot \operatorname{vec}\left(\mathbb{1}_n \otimes \left(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\right)\right)\right\|_2 \leqslant 0, \\ & \forall \boldsymbol{x}^j \in \mathcal{X}^j, \forall j \in [N]. \end{aligned} \quad (6)$$

In (6), $\odot$ is the Hadamard product, $\operatorname{vec}(\cdot)$ means stacking all elements into a column vector, and $\otimes$ denotes the Kronecker product. $\mathbb{1}_n$ is an n-dimensional all-one vector, where n is the dimension of $\boldsymbol{\phi}_i\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$. All the proofs are provided in the supplementary material.

Remark: The formulation in (6) converts the problem of finding $\boldsymbol{\vartheta}^{\text{in}}$ that satisfies (5) to a convex optimization problem. Each $\boldsymbol{x}^j \in \mathcal{X}^j$ introduces a constraint to (6). In many games, the cardinality of $\mathcal{X}_j$ is very large or infinite. In this case, it is intractable to enumerate all elements of $\mathcal{X}^j$ for all $j \in [N]$ to check if $\boldsymbol{\vartheta}$ meets the inequality constraints.

4.3 Loss Function Design

We propose a loss function-based method to tackle the case where $|\mathcal{X}_j|$ is large or infinite. Interestingly, we notice that this approach also applies when there may be no $\boldsymbol{\vartheta}$ consistent with $\widehat{\mathcal{D}}$, i.e., $\mathbb{C}/\{\mathbf{0}\} = \varnothing$. This scenario is common, and arises if players are bounded-rational and may only choose actions close to the optimum, leading to an ϵ-Nash equilibrium. Towards obtaining a loss function, we first relax the constraints of (6) by introducing *slack variables* $\beta_1, \ldots, \beta_N$:

$$\begin{aligned}&\min_{\boldsymbol{\vartheta},\boldsymbol{\beta}} \ \frac{1}{N}\sum_{j=1}^{N}\beta_j + \|\boldsymbol{\vartheta}\|_2\\ &\text{s.t. } \langle \boldsymbol{\Phi}^j_{\boldsymbol{\vartheta}}, \boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\rangle + \left\|\boldsymbol{\phi}^j \odot \mathrm{vec}\left(\mathbb{1}_n \otimes \left(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\right)\right)\right\|_2 \leqslant \beta_j,\\ &\quad \forall \boldsymbol{x}^j \in \mathcal{X}^j, \forall j \in [N]. \end{aligned} \tag{7}$$

Then, we define the following loss function.

Definition 3 (Loss Function)**.** *Given a context-action pair* $\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$*, the loss* $\ell_{\boldsymbol{\vartheta}}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ *of a parameter vector* $\boldsymbol{\vartheta}$ *is defined as*

$$\max_{\boldsymbol{x}^j \in \mathcal{X}^j} \left\{\langle \boldsymbol{\Phi}^j_{\boldsymbol{\vartheta}}, \boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\rangle + \left\|\boldsymbol{\phi}^j \odot \mathrm{vec}\left(\mathbb{1}_n \otimes \left(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\right)\right)\right\|_2\right\}.$$

Using the loss function, we reformulate the convex problem (7) as the following regularized empirical loss minimization problem (which remains convex):

$$\min_{\boldsymbol{\vartheta}} \frac{1}{N}\sum_{j=1}^{N} \ell_{\boldsymbol{\vartheta}}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right) + \alpha \mathcal{R}(\boldsymbol{\vartheta}). \tag{8}$$

Here, we replace $\|\boldsymbol{\vartheta}\|_2$ in the objective with a general regularization function $\mathcal{R}(\cdot)$ and use $\alpha \geqslant 0$ as a hyperparameter. In our supplemental material, we use a toy example to illustrate the effectiveness of the reformulation.

Remark 1: In (8), we use the loss function to capture the hard inequality constraints in (7). Since (8) is an unconstrained optimization, we can solve it using a first-order algorithm in Section 4.4 even when $|\mathcal{X}_j|$ is infinite.

Remark 2: If no $\boldsymbol{\vartheta}$ is consistent with $\widehat{\mathcal{D}}$, problem (6) has no feasible solution satisfying all the hard constraints. In this case, our loss function can still guide the search towards a vector that minimizes the degree of constraint violation.

4.4 Estimation Algorithm

In this section, we introduce a first-order algorithm for solving (8). We apply the *mirror descent method* [Beck and Teboulle, 2003], which enables the learning of $\boldsymbol{\vartheta}$ in both Euclidean and non-Euclidean geometries. Specifically, at each iteration t, the mirror descent update is give by

$$\boldsymbol{\vartheta}_{t+1} = \arg\min_{\boldsymbol{\vartheta}} \left\{\eta_t \langle \boldsymbol{g}_t(\boldsymbol{\vartheta}_t), \boldsymbol{\vartheta}\rangle + \mathcal{B}_\omega(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}_t)\right\}. \tag{9}$$

Here, $\eta_t > 0$ is the step-size, $\boldsymbol{g}_t(\boldsymbol{\vartheta}_t)$ is the subgradient of the complete loss function in (8), and function $\mathcal{B}_\omega$ is the Bregman divergence w.r.t. $\omega : \boldsymbol{\Theta} \to \mathbb{R}$ [Bubeck and others, 2015], i.e.,

$$\mathcal{B}_\omega(\boldsymbol{\vartheta}, \boldsymbol{\vartheta}_t) = \omega(\boldsymbol{\vartheta}) - \omega(\boldsymbol{\vartheta}_t) - \langle \nabla\omega(\boldsymbol{\vartheta}_t), \boldsymbol{\vartheta} - \boldsymbol{\vartheta}_t\rangle.$$

In particular, when $\omega(\boldsymbol{\vartheta}) = \frac{1}{2}\|\boldsymbol{\vartheta}\|_2^2$, (9) reduces to $\boldsymbol{\vartheta}_{t+1} = \boldsymbol{\vartheta}_t - \eta_t \boldsymbol{g}_t(\boldsymbol{\vartheta}_t)$, i.e., the subgradient descent update.

To implement the mirror descent method, we first derive subgradient $\boldsymbol{g}_t(\boldsymbol{\vartheta}_t)$ for the loss function in (8). Let $h\left(\boldsymbol{x}^j, \hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right) := \left\|\boldsymbol{\phi}^j \odot \mathrm{vec}\left(\mathbb{1}_n \otimes \left(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\right)\right)\right\|_2$. Using Danskin's Theorem [Bertsekas, 2016], we compute $\partial \ell_{\boldsymbol{\vartheta}}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ (the subdifferential of $\ell_{\boldsymbol{\vartheta}}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ w.r.t. $\boldsymbol{\vartheta}$) as

$$\mathbf{conv}\left\{\boldsymbol{\phi}^j \odot vec\left(\mathbb{1}_n \otimes \left(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\right)\right) \,\middle|\, \boldsymbol{x}^j \in \mathcal{X}^j(\boldsymbol{\vartheta})\right\}. \tag{10}$$

Algorithm 1 Mirror Descent for Problem (8)

1: **function** MIRROR DESCENT($\widehat{\mathcal{D}}, \omega, \mathcal{R}, \{\boldsymbol{\phi}_i\}, \alpha, \{\eta_t\}, \boldsymbol{\vartheta}_0$)
2:   **for** $t \in \{0, \ldots T-1\}$ **do** ▷ Iteration of Estimation
3:     **for** $j \in \{1, \ldots N\}$ **do** ▷ Loop over $\widehat{\mathcal{D}}$
4:       Compute $\tilde{\boldsymbol{x}}^j_t$ according to (11);
5:       Compute $\partial \ell_{\boldsymbol{\vartheta}_t}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ according to (10);
6:     **end for**
7:     Compute subgradient $\boldsymbol{g}_t(\boldsymbol{\vartheta}_t)$ according to (12);
8:     Apply mirror descent updates according to (9);
9:   **end for**
10:   **return** $\{\boldsymbol{\vartheta}_t\}_{t=1}^T$.
11: **end function**

Here, **conv** $\{\cdot\}$ represents the convex hull of the given set, and set $\mathcal{X}^j(\boldsymbol{\vartheta}) = \left\{\tilde{\boldsymbol{x}}^j\right\}$, where

$$\tilde{\boldsymbol{x}}^j = \arg\max_{\boldsymbol{x} \in \mathcal{X}^j} \left\{\langle \boldsymbol{\Phi}^j_{\boldsymbol{\vartheta}}, \boldsymbol{x} - \hat{\boldsymbol{x}}^j\rangle + h\left(\boldsymbol{x}, \hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)\right\}. \tag{11}$$

Based on (10), the subgradient $\boldsymbol{g}_t(\boldsymbol{\vartheta}_t)$ is derived as

$$\boldsymbol{g}_t(\boldsymbol{\vartheta}_t) = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{\phi}^j \odot \mathrm{vec}\left(\mathbb{1}_n \otimes \left(\tilde{\boldsymbol{x}}^j_t - \hat{\boldsymbol{x}}^j\right)\right) + \alpha \nabla \mathcal{R}(\boldsymbol{\vartheta}_t). \tag{12}$$

Algorithm 1 presents the pseudocode of the mirror descent algorithm for solving (8). In line 4 of Algorithm 1, we compute $\tilde{\boldsymbol{x}}^j_t$ for each data point $\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ by solving a maximization problem. In line 5, we use the obtained $\tilde{\boldsymbol{x}}^j_t$ to compute the subdifferential of $\ell_{\boldsymbol{\vartheta}_t}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ based on (10). In lines 7 to 8, we utilize the entire dataset to compute the subgradient $\boldsymbol{g}_t(\boldsymbol{\vartheta}_t)$ of the objective function in (8), and then update current parameters to obtain $\boldsymbol{\vartheta}_{t+1}$ by solving (9). Under certain conditions, the mirror descent method guarantees that the expected objective value at the average of $\{\boldsymbol{\vartheta}_t\}_{t=1}^T$ converges to the global minimum [Zattoni Scroccaro *et al.*, 2024]. We leave the complexity analysis of Algorithm 1 to the supplemental material.

4.5 Incorporation of Priors

In this subsection, we extend our estimation method to incorporate a specific type of prior knowledge about $\boldsymbol{\vartheta}$. We define $F_i(\boldsymbol{x}; \boldsymbol{\theta}_i) := -\partial U_i(\boldsymbol{x}, \boldsymbol{s}; \boldsymbol{\theta}_i)/\partial x_i$ and let $\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta}) := (F_1(\boldsymbol{x}; \boldsymbol{\theta}_i); \ldots; F_p(\boldsymbol{x}; \boldsymbol{\theta}_i))$. In many games [LeBlanc *et al.*, 1975; Jensen, 2018], $\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta})$ is known to be monotone, i.e., to satisfy

$$\left(\boldsymbol{x} - \boldsymbol{x}'\right)^\top \left(\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta}) - \boldsymbol{F}\left(\boldsymbol{x}'; \boldsymbol{\theta}\right)\right) \geqslant 0, \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}. \tag{13}$$

Such monotonicity property is closely related to the equilibrium solution in (1). For instance, if $\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta})$ is strictly monotone, there exists at most one Nash equilibrium [Harker and Pang, 1990]. Moreover, the projection gradient methods for solving (1) exhibit global convergence under this condition [Fukushima, 1992; Korpelevich, 1976; Xiu and Zhang, 2003; Bnouhachem *et al.*, 2015].

We then aim at estimating $\boldsymbol{\vartheta}$ while ensuring that $\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta})$ remains monotone. Towards this goal, we build upon the fact that $\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta})$ is monotone if and only if its Jacobian matrix is positive-semidefinite [Facchinei and Pang, 2003], i.e.,

$\boldsymbol{d}^\top \nabla \boldsymbol{F}(\boldsymbol{x})\, \boldsymbol{d} \geqslant 0, \forall \boldsymbol{x} \in \mathcal{X}, \boldsymbol{d} \in \mathbb{R}^p$. Based on this observation, we incorporate this condition in (8) by using *semidefinite program* [Vandenberghe and Boyd, 1996].

The specific form of $\boldsymbol{F}(\boldsymbol{x}; \boldsymbol{\theta})$ can vary across different game applications, making a unified formulation of the semidefinite program challenging. Hence, we take the Bertrand-Nash competition as an example to introduce our solution. It is a classic game widely studied in literature [Berry, 1994; Berry *et al.*, 1995; Bertsimas *et al.*, 2015]. We include the formulations of Cournot and traffic games in the supplementary material.

For ease of presentation, we consider a two-firm Bertrand competition (i.e., $p = 2$). The analysis of more than two firms is provided in the supplementary material. Each firm $i \in \{1, 2\}$ chooses price x_i to maximize revenue [Bertsimas *et al.*, 2015; Maddux *et al.*, 2023]:

$$U_i(\boldsymbol{x}, s; \boldsymbol{\theta}_i) = x_i(\theta_{i,1}x_1 + \theta_{i,2}x_2 + \theta_{i,3}s + \theta_{i,4}), \qquad (14)$$

where the terms in the parentheses correspond to the demand of firm i. In this case, $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1; \boldsymbol{\theta}_2) \in \mathbb{R}^8$.

In addition to the monotonicity property, there are some other common assumptions on $\boldsymbol{\vartheta}$ in this competition, e.g., U_i is concave w.r.t. x_i, and θ_{21}, θ_{12} can be normalized to 1 without loss of generality [Maddux *et al.*, 2023]. Next, we reformulate (8) to consider all these priors and assumptions.

Lemma 1. *Let* $\sqsupset := -\begin{bmatrix} \theta_{11} & \theta_{21}/2 \\ \theta_{12}/2 & \theta_{22} \end{bmatrix}$, $\tilde{\sqsupset} := \begin{bmatrix} \theta_{13} & \theta_{23} \\ \theta_{14} & \theta_{24} \end{bmatrix}$. *For a two-firm Bertrand competition, formulation (8) can be extended to the following problem:*

$$\begin{aligned} \min_{\sqsupset, \tilde{\sqsupset}} \quad & \frac{1}{N}\sum_{j=1}^{N} \ell_{\sqsupset,\tilde{\sqsupset}}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right) + \alpha \mathcal{R}\left(\sqsupset, \tilde{\sqsupset}\right) \\ \text{s.t.} \quad & \mathrm{Tr}\left(\left(\boldsymbol{e}_i \boldsymbol{e}_i^\top\right)\sqsupset\right) \geqslant 0, \forall i = 1, 2, \\ & \mathrm{Tr}\left(\boldsymbol{A}\sqsupset\right) = -1, \sqsupset \succeq 0, \tilde{\sqsupset} \succeq 0. \end{aligned} \qquad (15)$$

Here, $\boldsymbol{A}$ *equals* $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, *and the vector* $\mathbf{e}_i$ *has a one in the* i*-th position and zeros elsewhere.* $\mathrm{Tr}(\cdot)$ *is the trace operator. Constraint* $\mathrm{Tr}\left(\left(\boldsymbol{e}_i\boldsymbol{e}_i^\top\right)\sqsupset\right) \geqslant 0, \forall i = 1, 2$ *implies* $\theta_{11}, \theta_{22} \leqslant 0$, *ensuring the concavity of each* U_i. $\mathrm{Tr}(\boldsymbol{A}\sqsupset) = -1$ *indicates* $\theta_{21} = \theta_{12} = 1$. $\sqsupset, \tilde{\sqsupset} \succeq 0$ *means that both* $\sqsupset$ *and* $\tilde{\sqsupset}$ *are positive semidefinite. Loss function* $\ell_{\sqsupset,\tilde{\sqsupset}}\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$ *is*

$$\max_{\boldsymbol{x}^j \in \mathcal{X}^j} \left\{-\mathrm{Tr}\left(\Psi_s^j \sqsupset\right) + \mathrm{Tr}\left(\tilde{\Psi}_s^j \tilde{\sqsupset}\right) + h\left(\boldsymbol{x}^j, \hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)\right\}, \qquad (16)$$

where $\Psi_s^j := \begin{bmatrix} v_1^j & v_2^j + v_5^j \\ v_2^j + v_5^j & v_6^j \end{bmatrix}$, $\tilde{\Psi}_s^j := \begin{bmatrix} v_3^j & \frac{v_4^j + v_7^j}{2} \\ \frac{v_4^j + v_7^j}{2} & v_8^j \end{bmatrix}$, *and* $\boldsymbol{v}^j := \boldsymbol{\phi}^j \odot \mathrm{vec}\left(\mathbb{1}_n \otimes \left(\boldsymbol{x}^j - \hat{\boldsymbol{x}}^j\right)\right) \in \mathbb{R}^8$.

To solve the convex problem (15), we apply the *primal-dual interior-point method* [Boyd and Vandenberghe, 2004]. The primal barrier method augments the primal objective with a barrier function to handle inequality constraints, penalizing solutions near the boundary of the feasible set. The primal-dual interior-point method extends this concept to both the primal and dual problems, solving them simultaneously. It is often more efficient than the barrier method.

Algorithm 2 Primal-Dual Interior-Point for Problem (15)

1: **function** PD-IP($\widehat{\mathcal{D}}, \alpha, \epsilon, \sqsupset^0, \tilde{\sqsupset}^0, \Xi^0, \tilde{\Xi}^0, \boldsymbol{\lambda}^0, \nu^0$).
2: $\quad \frac{1}{\mu^0} \leftarrow \frac{\mathrm{Tr}(\sqsupset^0 \Xi^0) + \mathrm{Tr}(\tilde{\sqsupset}^0 \tilde{\Xi}^0)}{4p} + \frac{\sum_{i=1}^{p} \lambda_i^0 \mathrm{Tr}\left(\left(\boldsymbol{e}_i \boldsymbol{e}_i^\top\right)\sqsupset^0\right)}{4}$;
3: $\quad k \leftarrow 0$.
4: $\quad$ **while** $\frac{1}{\mu^k} > \epsilon$ **do**
5: $\quad\quad$ Compute $\Psi_s^j\left(\bar{\boldsymbol{x}}_k^j\right), \tilde{\Psi}_s^j\left(\bar{\boldsymbol{x}}_k^j\right)$ by $\widehat{\mathcal{D}}, \sqsupset^k$ and $\tilde{\sqsupset}^k$;
6: $\quad\quad$ Compute $\left(\Delta\sqsupset^k, \Delta\tilde{\sqsupset}^k, \Delta\Xi^k, \Delta\tilde{\Xi}^k, \Delta\boldsymbol{\lambda}^k, \Delta\nu^k\right)$;
7: $\quad\quad$ Backtracking line search for step-sizes η_p^k, η_d^k;
8: $\quad\quad$ Compute $\left(\sqsupset^{k+1}, \tilde{\sqsupset}^{k+1}, \Xi^{k+1}, \tilde{\Xi}^{k+1}, \boldsymbol{\lambda}^{k+1}, \nu^{k+1}\right)$;
9: $\quad\quad$ Compute $\frac{1}{\mu^{k+1}}$;
10: $\quad\quad k \leftarrow k + 1$;
11: $\quad$ **end while**
12: $\quad$ **return** $\left(\sqsupset^k, \tilde{\sqsupset}^k, \Xi^k, \tilde{\Xi}^k, \boldsymbol{\lambda}^k, \nu^k\right)$.
13: **end function**

To use the primal-dual interior-point method, we derive the perturbed KKT conditions based on the logarithmic barrier [Nesterov and Nemirovskii, 1994] as follows.

Proposition 1. *Let* $(\sqsupset^*, \tilde{\sqsupset}^*)$ *and* $(\Xi^*, \tilde{\Xi}^*, \boldsymbol{\lambda}^*, \nu^*)$ *be the optimal primal and dual solutions, respectively, and* μ *be the barrier parameter. Then, the optimality conditions for the logarithmic barrier centering problem are*

$$\sqsupset^*, \tilde{\sqsupset}^* \succeq 0; \;\; \Xi^*, \tilde{\Xi}^* \succeq 0; \;\; \lambda_i^* \geqslant 0, \;\; \mathrm{Tr}\left(\left(\boldsymbol{e}_i \boldsymbol{e}_i^\top\right)\sqsupset^*\right) \geqslant 0, \forall i;$$

$$\Xi^* \sqsupset^* = \frac{1}{\mu}\boldsymbol{I}; \;\; \tilde{\Xi}^* \tilde{\sqsupset}^* = \frac{1}{\mu}\boldsymbol{I}; \; \lambda_i^* \mathrm{Tr}\left(\left(\boldsymbol{e}_i \boldsymbol{e}_i^\top\right)\sqsupset^*\right) - \frac{1}{\mu} = 0, \forall i;$$

$$\mathrm{Tr}\left(\boldsymbol{A}\sqsupset^*\right) + 1 = 0; \qquad \alpha\tilde{\sqsupset}^* - \tilde{\Xi}^* + \frac{1}{N}\sum_{j=1}^{N} \tilde{\Psi}_s^j\left(\bar{\boldsymbol{x}}^j\right) = \mathbf{0};$$

$$\alpha\sqsupset^* - \sum_{i=1}^{2} \lambda_i^*\left(\boldsymbol{e}_i \boldsymbol{e}_i^\top\right) - \Xi^* + \nu^* \boldsymbol{A} - \frac{1}{N}\sum_{j=1}^{N} \Psi_s^j\left(\bar{\boldsymbol{x}}^j\right) = \mathbf{0},$$

where $\boldsymbol{I}$ *is the identity matrix, and* $\bar{\boldsymbol{x}}^j$ *is defined as* $\arg\max_{\boldsymbol{x}^j \in \mathcal{X}^j} \left\{-\mathrm{Tr}\left(\Psi_s^j \sqsupset^*\right) + \mathrm{Tr}\left(\tilde{\Psi}_s^j \tilde{\sqsupset}^*\right) + h\left(\boldsymbol{x}^j, \hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)\right\}$.

We use Newton's method [Alizadeh *et al.*, 1998] to solve the equations in Proposition 1. We leave the details to our supplementary material.

Algorithm 2 shows the primal-dual interior-point method for solving (15). In line 5, we use current parameters $\sqsupset^k$ and $\tilde{\sqsupset}^k$ to compute $\bar{\boldsymbol{x}}_k^j$ for each data point $\left(\hat{\boldsymbol{x}}^j, \hat{\boldsymbol{s}}^j\right)$, and then obtain $\Psi_s^j(\bar{\boldsymbol{x}}_k^j)$ and $\tilde{\Psi}_s^j(\bar{\boldsymbol{x}}_k^j)$. In line 6, we compute the Newton updates $(\Delta\sqsupset, \Delta\tilde{\sqsupset}, \Delta\Xi, \Delta\tilde{\Xi}, \Delta\boldsymbol{\lambda}, \Delta\nu)$. In lines 7, we utilize the backtracking line search to compute current step sizes η_p^k and η_d^k for the primal and dual variables, respectively. In lines 8 to 9, we update the current primal and dual variables, and compute barrier parameter μ^{k+1} in the next iteration.

5 Numerical Experiments

5.1 Games Applications

Demand Estimation in Bertrand Competition The two-firm Bertrand competition was described in Section 4.5.

The demand function estimation problem is: *Given data* $\{(\hat{x}_1^j, \hat{x}_2^j, \hat{s}^j)\}_{j=1}^N$*, we seek to estimate* $\boldsymbol{\vartheta} = (\boldsymbol{\theta}_1; \boldsymbol{\theta}_2) \in \mathbb{R}^8$ *such that each* $(\hat{x}_1^j, \hat{x}_2^j)$ *is a Nash equilibrium for context* $\hat{s}^j$*.*

Profit Estimation in Aggregative Games In aggregative games [Jensen, 2018], each player i's utility U_i depends on its own action $\boldsymbol{x}_i$ and the aggregation of all players' actions, i.e., $U_i(\boldsymbol{x}) = \tilde{U}_i(\boldsymbol{x}_i, \sum_{k=1}^p \boldsymbol{x}_k)$. We take the Cournot competition as an example. Each firm i decides the quantity x_i of a homogeneous product to supply, aiming to maximize:

$$U_i(\boldsymbol{x}, s; a, b, d, c_i) = x_i \left(b + ds - a \sum_{k=1}^{p} x_k \right) - c_i x_i. \quad (17)$$

The terms in the parentheses represent the inverse demand function, and c_i is the unit cost of production. Following [Risanger *et al.*, 2020], we include s as a demand shock that adjusts the demand intercept b. The problem is as follows: *Given* $\{(\hat{\boldsymbol{x}}^j, \hat{s}^j)\}_{j=1}^N$*, we seek* $\boldsymbol{\vartheta} = (a, b, d, c_1, \ldots, c_p) \in \mathbb{R}^{p+3}$ *such that each* $\hat{\boldsymbol{x}}^j$ *constitutes a Nash equilibrium.*

Cost Estimation in Traffic Game We represent a road network by a tuple $(\mathcal{V}, \mathcal{A}, \mathcal{W}, \boldsymbol{t}(\cdot))$, where $\mathcal{V}$ and $\mathcal{A}$ denote the sets of nodes and links, respectively. $\mathcal{W}$ is the set of all Origin-Destination (OD) pairs. $\boldsymbol{t}(\cdot)$ denotes the cost functions, and its a-th element $t_a(\cdot)$ is the travel latency cost function for link $a \in \mathcal{A}$. A common choice for $t_a(\cdot)$ is the U.S. Bureau of Public Roads (BPR) [Sheffi, 1985] function:

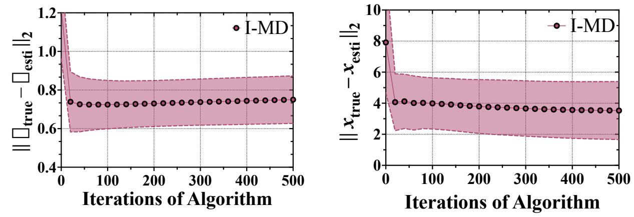
$$t_a(x_a) = \theta_a^0 + \theta_a^1 \left(\frac{x_a}{C_a} \right)^{\gamma}. \quad (18)$$

Here, x_a is the overall traffic on link a, C_a is the capacity of link a, γ is the congestion sensitivity parameter, and θ_a^0, θ_a^1 are cost parameters. In these settings it is standard to assume C_a and γ are common knowledge, while only θ_a^0, θ_a^1 need to be estimated. The cost estimation problem is: *Given link flow data* $\{\hat{\boldsymbol{x}}^j = (\hat{x}_a^j; a \in \mathcal{A})\}_{j=1}^N$*, we seek to estimate cost parameters* $\{\boldsymbol{\theta}_a = (\theta_a^0, \theta_a^1), a \in \mathcal{A}\}$*, such that each* $\hat{\boldsymbol{x}}^j$ *is a Wardrop equilibrium [Sheffi, 1985; Patriksson, 2015].*

5.2 Experimental Settings

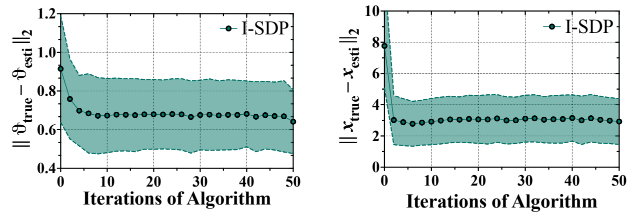
Experimental Data We describe the dataset as follows.

- **Bertrand Competition**: Following [Maddux *et al.*, 2023], we generate $\boldsymbol{\vartheta}$ by randomly sampling its elements from Gaussian distributions: $\theta_{11} \sim \mathcal{N}(-1.2, 0.5^2)$, $\theta_{12} \sim \mathcal{N}(0.5, 0.1^2)$, $\theta_{21} \sim \mathcal{N}(0.3, 0.1^2)$, $\theta_{22} \sim \mathcal{N}(-1, 0.5^2)$, and $\theta_{i3}, \theta_{i4} \sim \mathcal{N}(1, 0.5^2)$ for $i = 1, 2$. We take s to be i.i.d. samples from $\mathcal{N}(5, 1.5^2)$. Given each $\hat{s}^j$, we solve for the equilibrium prices $(\hat{x}_1^j, \hat{x}_2^j)$ using first-order methods. To evaluate different estimation methods, we generate 50 random $\boldsymbol{\vartheta}$. For each $\boldsymbol{\vartheta}$, we create a training dataset $\widehat{\mathcal{D}}_{\text{train}}$ and a test dataset $\widehat{\mathcal{D}}_{\text{test}}$, both with a size of 500.
- **Cournot Competition**: We consider $p = 3$ players, and generate $\boldsymbol{\vartheta}$ by randomly sampling its elements as follows: $a, d \sim \mathcal{U}(5, 10)$, $b \sim \mathcal{N}(50, 5^2)$, $c_i \sim \mathcal{U}(10, 20)$. Given each context $\hat{s}^j \sim \mathcal{N}(6, 2^2)$, we solve for the Nash equilibrium $\hat{\boldsymbol{x}}^j$ using first-order methods. We randomly generate 50 different $\boldsymbol{\vartheta}$ for evaluation. Each $\boldsymbol{\vartheta}$ has a corresponding $\widehat{\mathcal{D}}_{\text{train}}$ and $\widehat{\mathcal{D}}_{\text{test}}$, both with 500 samples.



(a) $\|\boldsymbol{\vartheta}_{\text{true}} - \boldsymbol{\vartheta}_{\text{esti}}\|_2$. (b) $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$.

Figure 2: Convergence of **I-MD** on the Bertrand Dataset.



(a) $\|\boldsymbol{\vartheta}_{\text{true}} - \boldsymbol{\vartheta}_{\text{esti}}\|_2$. (b) $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$.

Figure 3: Convergence of **I-SDP** on the Bertrand Dataset.

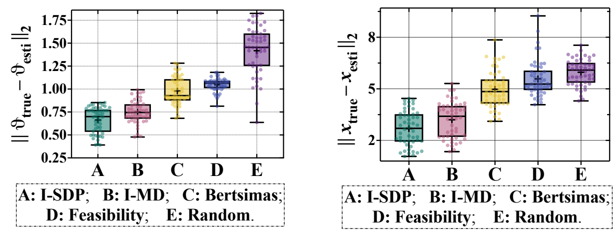
- **Traffic Game**: We consider the Sioux Falls network [LeBlanc *et al.*, 1975], which contains 24 nodes and 76 links. Every two nodes in the network constitute OD pairs. We set $\boldsymbol{\vartheta}$ according to this real network. To generate multiple equilibrium data, following [Bertsimas *et al.*, 2015], we randomly perturb the OD demands 1000 times, with each perturbation drawn from $\mathcal{U}(0, 0.1)$. Using Frank-Wolfe algorithm and true $\boldsymbol{\vartheta}$, we then compute $\hat{x}_a$ for each perturbed OD demand. $\widehat{\mathcal{D}}_{\text{train}}$ and $\widehat{\mathcal{D}}_{\text{test}}$ both contain 500 samples.

Comparison Methods We compare the performance of the following five estimation methods:

- **I-MD** (Incenter with Mirror Descent): We estimate $\boldsymbol{\vartheta}$ using Algorithm 1.
- **I-SDP** (Incenter with SemiDefinite Programming): We use Algorithm 2, which incorporates the monotonicity of $\boldsymbol{F}$ (the versions tailored for the Cournot competition and traffic game are in the supplementary material).
- **Bertsimas** (Data-Driven Estimation in Equilibrium Using Inverse Optimization [Bertsimas *et al.*, 2015]): It estimates $\boldsymbol{\vartheta}$ from observed equilibria based on inverse optimization without using the concept of incenter.
- **Feasibility**: It chooses a parameter vector from the set $\mathbb{C}$ of consistent parameter vectors. To check the feasibility of the parameter vectors w.r.t. (3), it discretizes each $\mathcal{X}^j$.
- **Random**: It randomly chooses a parameter vector.

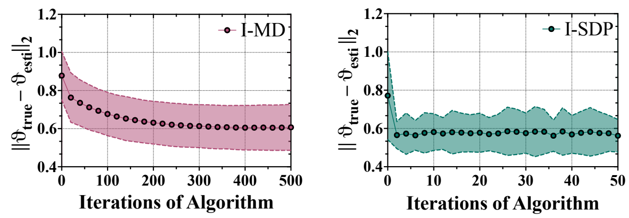
5.3 Experimental Results

We evaluate the estimation errors of the methods using two metrics: (i) The ℓ_2-distance between the estimated parameter vector $\boldsymbol{\vartheta}_{\text{esti}}$ and the true parameter vector $\boldsymbol{\vartheta}_{\text{true}}$, i.e., $\|\boldsymbol{\vartheta}_{\text{true}} - \boldsymbol{\vartheta}_{\text{esti}}\|_2$. To ensure consistent comparisons, all parameter vectors are normalized before evaluation; (ii) The ℓ_2-distance between the equilibrium $\boldsymbol{x}_{\text{esti}}$ computed using $\boldsymbol{\vartheta}_{\text{esti}}$ and the equilibrium $\boldsymbol{x}_{\text{true}}$ computed using the true parameter



(a) $\|\boldsymbol{\vartheta}_{\text{true}} - \boldsymbol{\vartheta}_{\text{esti}}\|_2$. (b) $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$.

Figure 4: Comparison Results on Bertrand Testing Data.



(a) Convergence of **I-MD**. (b) Convergence of **I-SDP**.

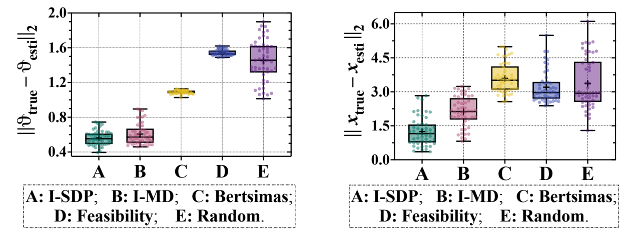
Figure 5: Convergence Results on the Cournot Dataset.

vector $\boldsymbol{\vartheta}_{\text{true}}$, i.e., $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$. The scalability analysis of our methods is provided in the supplementary material.

Bertrand Competition In Figure 2, we depict the convergence of our **I-MD** method on the Bertrand data. Figure 2a illustrates the average error between $\boldsymbol{\vartheta}_{\text{esti}}$ and $\boldsymbol{\vartheta}_{\text{true}}$ over 50 experiments (each with a different $\boldsymbol{\vartheta}_{\text{true}}$). The shaded area represents the standard deviation of these errors. Figure 2b shows the average error between the true equilibrium $\boldsymbol{x}_{\text{true}}$ and the equilibrium $\boldsymbol{x}_{\text{esti}}$ computed using $\boldsymbol{\vartheta}_{\text{esti}}$ on the *testing data*. The convergence of our **I-SDP** method is depicted in Figure 3. Our **I-SDP** converges to lower errors in both metrics compared to **I-MD**. This is attributed to the incorporation of the monotonicity property of $\boldsymbol{F}$. It can also be observed that **I-SDP** converges much faster, thanks to the efficiency of the primal-dual interior-point method.

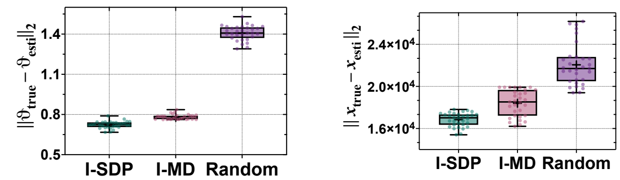
Figure 4 presents the comparative results of different estimation methods on the testing data, evaluated using two metrics. We use boxplots to illustrate the distribution of two error metrics across 50 trials for each method. Each box includes all error points, with the central line representing the median and the "+" sign denoting the mean error. Our **I-SDP** method achieves the lowest median and mean errors with minimal variability across both metrics. Specifically, for the metric $\|\boldsymbol{\vartheta}_{\text{esti}} - \boldsymbol{\vartheta}_{\text{true}}\|_2$, **I-SDP** achieves the lowest mean error (around 0.663), whereas the comparison methods exhibit relatively large errors (with the best among them being 0.979). Similarly, for $\|\boldsymbol{x}_{\text{esti}} - \boldsymbol{x}_{\text{true}}\|_2$, **I-SDP** again performs best, with a mean error around 2.690. Notably, our **I-MD** method outperforms all comparison methods across two metrics.

Cournot Competition In Figure 5, we present the convergence performance of our methods, measured by the gap between $\boldsymbol{\vartheta}_{\text{esti}}$ and $\boldsymbol{\vartheta}_{\text{true}}$. We include the convergence about $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$ in the supplementary material. Comparing



(a) $\|\boldsymbol{\vartheta}_{\text{true}} - \boldsymbol{\vartheta}_{\text{esti}}\|_2$. (b) $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$.

Figure 6: Comparison Results on Cournot Testing Data.



(a) $\|\boldsymbol{\vartheta}_{\text{true}} - \boldsymbol{\vartheta}_{\text{esti}}\|_2$. (b) $\|\boldsymbol{x}_{\text{true}} - \boldsymbol{x}_{\text{esti}}\|_2$.

Figure 7: Comparison Results on Traffic Testing Data.

Figure 5a and Figure 5b, we can see that our **I-SDP** converges to a lower error at a much faster rate compared to **I-MD**.

Figure 6 presents the boxplots concerning two error metrics across different estimation methods on the testing data. It can be observed that our **I-SDP** and **I-MD** methods outperform all the other methods across both metrics. Specifically, our **I-SDP** achieves the lowest errors, with a mean error of 0.562 for $\|\boldsymbol{\vartheta}_{\text{esti}} - \boldsymbol{\vartheta}_{\text{true}}\|_2$ and a mean error of 1.245 for $\|\boldsymbol{x}_{\text{esti}} - \boldsymbol{x}_{\text{true}}\|_2$. In contrast, the lowest errors among the comparison methods are 1.089 and 3.203 for the two metrics.

Traffic Game We leave the convergence performance of our methods with respect to $\|\boldsymbol{\vartheta}_{\text{esti}} - \boldsymbol{\vartheta}_{\text{true}}\|_2$ to our supplementary material. Figure 7 presents a boxplot comparison of our two methods against **Random** on the testing data. **Bertsimas** and **Feasibility** are excluded, because (i) **Bertsimas** becomes intractable, as in the traffic game it requires solving an optimization problem with infinite constraints; (ii) **Feasibility** is impractical due to the vast size of the global action space (e.g., discretizing each flow region with only 2 values leads to a size of 2^{76}). It is evident that both our **I-SDP** and **I-MD** significantly outperform the **Random** method.

6 Conclusion

In this paper, we introduced a novel framework for estimating player utilities from their equilibrium behaviors. Our estimation framework is applicable to a broad range of multiplayer non-cooperative games and practical applications. We also extended it to account for a specific structural property in player utility functions. Experimental results on three game applications demonstrate the superiority of our methods over baselines. The main focus of our work is on efficiently computing the incenter and evaluating its effectiveness. Future directions will include analyzing the sample complexity of our utility estimation method and integrating utility estimation with the prediction of equilibrium behaviors.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (No. 62202050), the Beijing Institute of Technology Research Fund Program for Young Scholars, the BIT Zhumeng Huanyu Scholarship for Innovative Talents, and the EPSRC grant EP/Y001001/1, funded by the International Science Partnerships Fund (ISPF) and UKRI.

References

[Alizadeh *et al.*, 1998] Farid Alizadeh, Jean-Pierre A Haeberly, and Michael L Overton. Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results. *SIAM Journal on Optimization*, 8(3):746–768, 1998.

[Aswani *et al.*, 2018] Anil Aswani, Zuo-Jun Shen, and Auyon Siddiq. Inverse optimization with noisy data. *Operations Research*, 66(3):870–892, 2018.

[Beck and Teboulle, 2003] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

[Berry *et al.*, 1995] Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica: Journal of the Econometric Society*, pages 841–890, 1995.

[Berry, 1994] Steven T Berry. Estimating discrete-choice models of product differentiation. *The RAND Journal of Economics*, pages 242–262, 1994.

[Bertsekas, 2016] Dimitri Bertsekas. *Nonlinear programming*, volume 4. Athena Scientific, 2016.

[Bertsimas *et al.*, 2015] Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153:595–633, 2015.

[Besbes *et al.*, 2023] Omar Besbes, Yuri Fonseca, and Ilan Lobel. Contextual inverse optimization: Offline and online learning. *Operations Research*, 2023.

[Blum *et al.*, 2014] Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. Learning optimal commitment to overcome insecurity. *Advances in Neural Information Processing Systems*, 27, 2014.

[Bnouhachem *et al.*, 2015] Abdellah Bnouhachem, Qamrul Hasan Ansari, and Ching-Feng Wen. A projection descent method for solving variational inequalities. *Journal of Inequalities and Applications*, 2015:1–14, 2015.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[Bubeck and others, 2015] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[Chan *et al.*, 2023] Timothy CY Chan, Rafid Mahmood, and Ian Yihang Zhu. Inverse optimization: Theory and applications. *Operations Research*, 2023.

[Facchinei and Pang, 2003] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

[Fukushima, 1992] Masao Fukushima. Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems. *Mathematical programming*, 53:99–110, 1992.

[Haghtalab *et al.*, 2016] Nika Haghtalab, Fei Fang, Thanh Hong Nguyen, Arunesh Sinha, Ariel D Procaccia, and Milind Tambe. *Three strategies to success: Learning adversary models in security games*. 2016.

[Harker and Pang, 1990] Patrick T Harker and Jong-Shi Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical programming*, 48(1):161–220, 1990.

[Heuberger, 2004] Clemens Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of combinatorial optimization*, 8:329–361, 2004.

[Jensen, 2018] Martin Kaae Jensen. Aggregative games. In *Handbook of Game Theory and Industrial Organization, Volume I*, pages 66–92. Edward Elgar Publishing, 2018.

[Korpelevich, 1976] Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

[Kuleshov and Schrijvers, 2015] Volodymyr Kuleshov and Okke Schrijvers. Inverse game theory: Learning utilities in succinct games. In *Web and Internet Economics: 11th International Conference, WINE 2015, Amsterdam, The Netherlands, December 9-12, 2015, Proceedings 11*, pages 413–427. Springer, 2015.

[LeBlanc *et al.*, 1975] Larry J LeBlanc, Edward K Morlok, and William P Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation research*, 9(5):309–318, 1975.

[Ling *et al.*, 2018] Chun Kai Ling, Fei Fang, and J Zico Kolter. What game are we playing? end-to-end learning in normal and extensive form games. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 396–402, 2018.

[Ling *et al.*, 2019] Chun Kai Ling, Fei Fang, and J Zico Kolter. Large scale learning of agent rationality in two-player zero-sum games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6104–6111, 2019.

[Maddux *et al.*, 2023] Anna M Maddux, Nicolò Pagan, Giuseppe Belgioioso, and Florian Dörfler. Data-driven behaviour estimation in parametric games. *IFAC-PapersOnLine*, 56(2):9330–9335, 2023.

[Mohajerin Esfahani *et al.*, 2018] Peyman Mohajerin Esfahani, Soroosh Shafieezadeh-Abadeh, Grani A Hanasusanto, and Daniel Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167:191–234, 2018.

[Narahari *et al.*, 2009] Yadati Narahari, Dinesh Garg, Ramasuri Narayanam, and Hastagiri Prakash. *Game theoretic problems in network economics and mechanism design solutions*. Springer Science & Business Media, 2009.

[Nash, 1950] John F Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[Nesterov and Nemirovskii, 1994] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

[Noti, 2021] Gali Noti. From behavioral theories to econometrics: Inferring preferences of human agents from data on repeated interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5637–5646, 2021.

[Patriksson, 2015] Michael Patriksson. *The traffic assignment problem: models and methods*. Courier Dover Publications, 2015.

[Peters *et al.*, 2023] Lasse Peters, Vicenç Rubies-Royo, Claire J Tomlin, Laura Ferranti, Javier Alonso-Mora, Cyrill Stachniss, and David Fridovich-Keil. Online and offline learning of player objectives from partial observations in dynamic games. *The International Journal of Robotics Research*, 42(10):917–937, 2023.

[Risanger *et al.*, 2020] Simon Risanger, Stein-Erik Fleten, and Steven A Gabriel. Inverse equilibrium analysis of oligopolistic electricity markets. *IEEE Transactions on Power Systems*, 35(6):4159–4166, 2020.

[Sheffi, 1985] Yosef Sheffi. *Urban transportation networks*, volume 6. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[Tsai *et al.*, 2016] Dorian Tsai, Timothy L Molloy, and Tristan Perez. Inverse two-player zero-sum dynamic games. In *2016 Australian Control Conference (AuCC)*, pages 192–196. IEEE, 2016.

[Vandenberghe and Boyd, 1996] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[Waugh *et al.*, 2011] Kevin Waugh, Brian D Ziebart, and J Andrew Bagnell. Computational rationalization: the inverse equilibrium problem. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1169–1176, 2011.

[Wu *et al.*, 2022] Jibang Wu, Weiran Shen, Fei Fang, and Haifeng Xu. Inverse game theory for stackelberg games: the blessing of bounded rationality. *Advances in Neural Information Processing Systems*, 35:32186–32198, 2022.

[Xiu and Zhang, 2003] Naihua Xiu and Jianzhong Zhang. Some recent advances in projection-type methods for variational inequalities. *Journal of Computational and Applied Mathematics*, 152(1-2):559–585, 2003.

[Yu *et al.*, 2022] Yue Yu, Jonathan Salfity, David Fridovich-Keil, and Ufuk Topcu. Inverse matrix games with unique quantal response equilibrium. *IEEE Control Systems Letters*, 7:643–648, 2022.

[Zattoni Scroccaro *et al.*, 2024] Pedro Zattoni Scroccaro, Bilge Atasoy, and Peyman Mohajerin Esfahani. Learning in inverse optimization: Incenter cost, augmented suboptimality loss, and algorithms. *Operations Research*, 2024.