Game Theory with Computer Science Applications
Lecture 1: Introduction

Zhenzhe Zheng

Department of Computer Science and Engineering
Shanghai Jiao Tong University

zhengzhenzhe@sjtu.edu.cn

February 22, 2021

Outline

Course Information

- Introduction to fundamentals of game theory and mechanism design.
- Emphasis on solution concepts, basic models, mathematical tools and design techniques in different environments.
- Motivations and applications are from CS, such as
  - wireline and wireless communication networks
  - multi-agent systems
  - machine learning models
  - online platform markets
  - computational advertising
  - ...
- Intended Audience: The course is geared towards CS, EE, or OR students who need to use game theory in their research. The course is also aimed at covering recent advances and open research topics in game theory and related applications from CS.
- Prerequisites: Courses in probability, optimization and algorithm design would be helpful, but is not required.

Course Information

- Grading:
  - 50%: Homeworks
  - 30%: Project
  - 20%: Final
- Homework: There will be 5 homework sets. Homework solutions will be handed out on the day that the homework is due. Late homework will be heavily discounted. Duplicating a solution that someone else has written or providing solutions to be copied is not acceptable.
- Project: You will need to write a project report of 5 to 10 pages in length, and prepare a 10 minutes presentation in the final week of classes. Possible project types include but are not limited to:
  - Read and report on 2-4 papers on a theoretical/application area related to game theory or mechanism design.
  - Theoretical analysis of a game-theoretic model, which we have not covered in class and which has not been fully explored in the literature.
  - Apply the tools from game theory and mechanism design to solve a practical problem from CS applications.

Text and References

- Main Text: I. Menache and A. Ozdaglar. Network Games: Theory, Models and Dynamics, Morgan & Claypool, March 2011.
- Other Useful References:
  - B. Hajek, An Introduction to Game Theory, 2018, UIUC.
  - R. Srikant. Lecture Notes on Game Theory, UIUC.
  - T. Roughgarden. Lecture Notes on Algorithmic Game Theory, Stanford.
  - N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani (Eds.), Algorithmic Game Theory, Cambridge University Press, 2007.
  - Y. Shoham and K Leyton-Brown. Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, 2008.
  - N. Cesa-Bianchi and G. Lugosi, Prediction, Learning, and Games, Cambridge University Press, 2006.
  - M. J. Osbone and A. Rubinstein, A Course in Game Theory, MIT Press, 1994.
  - V. Krishna, Auction Theory, Associated Press, 2002.

Outline

Introduction

- Optimization Theory: Optimize a single objective over a decision variable $x \in \mathbb{R}^n$:

$$\begin{aligned} \text{Maximize} \quad & \sum_i u_i(x), \\ \text{Subject to} \quad & x \in X \subset \mathbb{R}^n. \end{aligned}$$

- Game Theory: Multi-agent decision problems: There are n agents in the system, each of which chooses some $x_i \in \mathbb{R}$ to maximize her own utility $u_i(x), x \in \mathbb{R}^n$, or equivalently,

$$u_i(x_i, x_{-i}) \qquad x_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n).$$

  - How to model the strategic interactions between agents?
  - What is the possible outcome of $\hat{x}^*$? What is the characteristic of $\hat{x}^*$? What is the difference between $\sum_i u_i(\hat{x}^*)$ and $\sum_i u_i(x^*)$?
  - How to calculate $\hat{x}^*$ (computation of equilibrium)?

Example: Single-Item Auction

- Seller has an item to sell.
- n buyers each with a valuation $v_1, v_2, \cdots, v_n$, without loss of generality, $v_1 > v_2 > \cdots > v_n$.
- Allocation solution: $x \in [0,1]^n$; feasible region $\sum x_i = 1$.
- Utility $u_i(x) = v_i \times x_i$.
- Allocation rule: the buyer with the largest valuation wins the item. For multiple these buyers, allocate the item uniformly.
- Optimization Problem: Follow the allocation rule.
- Game Problem:
  - v_i is private and only known to the buyer i.
  - b_i is the bid of buyer i.
  - Strategic interactions: $b_i \neq v_i$, and strategic buyer would claim $b_i = v_1$. Allocate the item uniformly among all users.

	Optimization Problem	Game Problem
Objective	v_1	$\frac{1}{n}\sum_i v_i$
Solution	$(1, 0, \cdots, 0)$	$(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n})$.

Example: Resource Allocation

- One resource with unit capacity, e.g., a single-link network
- Three users share this resource with utility functions:

$$u_0(x_0) = log(x_0),\ u_1(x_1) = 3log(x_1),\ u_2(x_2) = 4log(x_2).$$

- Optimization Problem:

$$\text{Maximize}_{x_0+x_1+x_2\leq 1}\ log(x_0) + 3log(x_1) + 4log(x_2), \tag{1}$$

  which yields allocation

$$x_0^* = 1/8, \quad x_1^* = 3/8 \quad x_2^* = 1/2,$$

  and the overall utility ≈ -3.39.
- Game Problem:
  - We do not even know the utility functions.
  - Reported utility functions $u_0(x) = u_1(x) = u_2(x) = 4log(x)$.

$$\text{Maximize}_{x_0+x_1+x_2\leq 1}\ 4log(x_0) + 4log(x_1) + 4log(x_2), \tag{2}$$

  which yields allocation result $\hat{x}_0^* = 1/3, \quad \hat{x}_1^* = 1/3 \quad \hat{x}_2^* = 1/3$ and the overall (true) utility ≈ -3.82.

Introduction

- Mechanism Design (MD): "Inverse Game Theory": Design the structure of game to "persuade" agents to revise x_i with the goal of maximizing some desirable objective, e.g., maximize $\sum_i u_i(x)$.
  - For example, design right incentives for agents to guarantee that $\sum_i u_i(\hat{x}^*)$ is an good approximation of $\sum_i u_i(x^*)$.
- Algorithmic Game Theory: study of designing efficient mechanism for computational hardness problems with performance guarantee.
  - For example, design mechanism for knapsack problems with good approximation ratio and economic properties.
- In Economics, MD is more about designing the right incentives. In CS, focus is more on the design of efficient and effective algorithms with strategic inputs.

Knapsack Auction

- 0-1 Knapsack Problem:
  - n users, each user i is associated with a Weight w_i and a Value: v_i.
  - Allocation variable: $x_i \in \{0, 1\}$
  - Knapsack Capacity: W.

$$\text{Maximize} \quad \sum_{i=1}^{n} v_i x_i$$

$$\text{Subject to} \quad \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0, 1\}.$$

- 0-1 Knapsack Auction:
  - n strategic users, each user i is associated with a public Weight w_i and a private Value: v_i.

$$\text{Maximize} \quad \sum_{i=1}^{n} b_i x_i$$

$$\text{Subject to} \quad \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0, 1\}.$$

Outline

1. Course Information
2. Introduction to Game Theory
3. Tentative topics

Static Games with Complete Information

- Model for strategic games: matrix games (Prisoner's Dilemma) and continuous games (Cournot Game)
- Solution concepts: equilibrium hierarchy
  - Dominant and Dominated Strategies
  - Pure Nash Equilibrium (NE) and Mixed Nash Equilibrium
  - Correlated Equilibrium
- Existence of NE: a fixed point of best-response correspondences.
  - **For finite games**, Nash's Theorem shows the existence of a mixed NE using fixed-point theorems.
  - **For continuous games**, show the existence of pure NE under convexity assumption.
  - **For general continuous games**, Glicksberg's Theorem shows the existence of mixed NE.
  - **For discontinous games**, existence of a mixed equilibrium established under some assumptions.
- Uniqueness of NE using "strict diagonal concavity"

Achieve Equilibrium Points

- Learning: Best-response dynamics, fictitious play, regret-matching algorithm.
- Evolution: Evolutionarily stable strategies, replicator dynamics.
- Computation of Equilibrium:
  - Zero-sum games.
  - Nonzero-sum games.
  - Algorithms that exploit polyhedral structure, Lemke-Howson algorithm; algorithms for finding fixed-points, Scarf's algorithm; exhaustive "smart" search etc.

Static Games with Special Structure

- Supermodular Game: *when one player takes a higher strategy, the others want to do the same.*
  - Example: Bertrand price competition game.
  - **Nice equilibrium properties**: existence of a pure NE, convergence to pure NE via simple greedy dynamics.
- Potential Game: Games that admit a "potential function" $G(x_i, x_{-i})$ such that maximization with respect to x_i coincide with the maximization problem of each agent. Similar nice equilibrium properties.
  - Example: congestion game in resource allocation.
- Smooth Game: the relation of local utility and the overall utility within the strategic environment can be expressed as an inequality, which can bound Price of Anarchy (POA)

$$POA = \frac{\sum_i u_i(\hat{x}^*)}{\sum_i u_i(x^*)}$$

  - Example: Facility Location Game

Dynamic Game with Complete information

- Repeated Game: Infinitely and finitely repeated games, sustaining desirable/cooperative outcomes. Trigger strategies, folk theorems.
  - Example: Prisoner's Dilemma in repeated setting.
- Multi-Stage Game: backward induction, dynamic programming and **subgame perfect equilibrium**.
  - Example: Bargaining games, Nash bargaining solution.
- Stochastic Game: Markov strategies and Markov perfect equilibrium. Generalization of Markov Decision Processes (MDP) and repeated games.
  - Example: Multi-Agent Reinforcement Learning.

Games with Incomplete Information

- Static games with incomplete information.
  - Each agent has private information (called **"type"**), and know the conditional distribution of types of other agents.
  - Bayesian Nash Equilibrium
- Dynamic games with incomplete information. Solution concept: Perfect Bayesian Equilibrium
- Applications in **auction theory**:
  - Different auction formats (first-price, second-price auctions).
  - **Social welfare** and **revenue** of different auctions.
  - Can we design the "optimal" auction for a given objective?

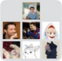
Mechanism Design

- Design of game forms to implement certain desirable outcomes.
- Mechanism as a mapping that maps “signals” from agents into **allocations** and **payments**.
- Revelation principle, strategy-proofness.
- Efficient Mechanism (Vickrey-Clarke-Groves): Design a mechanism to maximize social welfare.
- Optimal Mechanism (Myerson): Design a mechanism to maximize revenue.

Cooperative Games and Applications from CS

- Cooperative Games
  - the concept of coalitions and core
  - Shapley value
  - Example: cost sharing game
- Applications from CS
  - Utility-based resource allocation: congestion control in communication network; Wireless Spectrum Management.
  - Selfish routing. Wardrop and Nash equilibrium.
  - Equilibrium analysis of Generative Adversarial Network (GAN).
  - Positive and negative externalities in social network.
  - Network pricing: Combined pricing and traffic engineering.
  - Competition games in online platform markets.
  - Game Models in Data Marketplace.
  - Auction Design in Computational Advertising.

Thanks!



CS244

Valid until 2/28 and will update upon joining group