CS272 - Theoretical Foundations of Learning, Decisions, and Games

Lecture 16: Stackelberg Games

October 23, 2025

Lecturer: Nika Haghtalab *Readings: Conitzer and Sandholm 2006*
Scribe: Jivat Kaur, Mark Bedaywi

1 Recap

In the lectures up to now, we studied zero-sum and general-sum games. The equilibria we saw so far are simultaneous play notions where both players declare their strategies simultanousely. While the order of play in zero-sum games does not change the value agents receive (as proved in the minmax theorem), in general sum games order of play can be consequential.

oday, we will visit sequential play settings that occur often in the real world. For example, consider our interaction with models like ChatGPT—while we are adjusting our responses when interacting with such tools, the models do not change once released based on our response. Similarly, consider the example of automated resume screening tools. We adjust to the strengths and weaknesses of the tool and take actions based on that; whereas the tool remains static. Today, we will study such settings where there is asymmetry in power of the players: we have leaders and followers, and how the equilibrium solution changes in sequential play. Stackelberg games capture this sequential play nature.

2 Stackelberg games

Consider the following game where the row player can play Up, Down and the column player can play Left, Right. We will look at the Nash equilibrium of this game.

	Left	**Right**
Up	$1, 1$	$3, 0$
Down	$0, 0$	$2, 1$

The dominant strategy for the row player is to play Up and for the column player is to play Left in that case. The only MNE of this game is (Up, Left) with the utility of players being (1, 1).

Now, consider a scenario where the row player can go first and commits to a strategy. If the row player commits to Down, then the column player responds by playing Right and the row player ends with a utility of 2. This shows that it is beneficial to commit! Now, what if the row player commits to a mixed strategy?

		Left	**Right**
$0.5 - \epsilon$	**Up**	$1, 1$	$3, 0$
$0.5 + \epsilon$	**Down**	$0, 0$	$2, 1$

Let the row player commit to a mixed strategy $p = (0.4999, 0.5001)$ i.e., plays Up with probability 0.4999 and Down with 0.5001. Then $U_2(p, \text{Right}) = 0.5001$ and $U_2(p, \text{Left}) = 0.4999$. Thus, column player's response to p is to play Right. This results in the utilities (2.4999, 0.5001). The message from this example game is that if you can commit to your strategy, do it!

(Mixed) Stackelberg Equilibrium: Given the mixed strategy of player 1 $p \in \Delta(A_1)$, the best response of player 2 is:

$$\text{BR}(p) = \arg\max_{a_2 \in A_2} \mathbb{E}_{a \sim p}[u_2(a, a_2)]$$

Stackelberg optimal strategy of player 1 is to

$$p^* \in \arg\max_{p \in \Delta(A_1)} U_1(p, \text{BR}(p))$$

Strong Stackelberg Equilibrium (optimistic). Tie-breaking by player 2 is to the advantage of player 1

$$\max_{p \in \Delta(A_1)} \max_{a_2 \in \text{BR}(p)} u_1(p, a_2)$$

Weak Stackelberg Equilibrium (pessimistic). Tie-breaking by against the benefit of player 1

$$\max_{p \in \Delta(A_1)} \min_{a_2 \in \text{BR}(p)} u_1(p, a_2)$$

Theorem 2.1. *For any 2 player game, utility of the leader in a mixed Strong Stackelberg Equilibrium (SSE) $\geq$ leader's utility in any MNE.*

Proof. Take any MNE (p_1, p_2). Because p_2 is in MNE, we know $p_2 \in \text{BR}(p_1)$. So player 1's utility is:

$$u_1(p_1, p_2) \leq u_1(p_1, \text{BR}(p_1)),$$

since tie-breaking is benefiting player 1.

Now, by definition of SSE:

$$u_1(p^*, \text{BR}(p^*)) \geq u_1(p_1, \text{BR}(p_1))$$

This is because of the way the the leader optimizes which can only improve leader's utility. □

We define the value of commitment as:

$$\text{Value of commitment} = \frac{\text{Leader's utility in SSE}}{\text{Leader's utility in best MNE}} \geq 1$$

Note that we will use Strong Stackelberg Equilibrium notion throughout the semester. Next, we will look at the algorithmic perspective of Stackelberg equilibrium. While they always exist by definition, we will show that we can compute Stackelberg equilibrium efficiently.

Theorem 2.2. *For any game, a strong Stackelberg optimal strategy exists and can be found in* $\text{poly}(|\mathcal{A}_1|, |\mathcal{A}_2|)$*. In fact, it can be computed using* $|\mathcal{A}_2|$ *# calls to a linear program in* $|\mathcal{A}_1|$ *dimensions and* $|\mathcal{A}_2|$ *constraints.*

Consider the leader can play $n = |A_1|$ pure strategies and the follower can play $m = |A_2|$ pure strategies. Any mixed strategy of the leader $p^* = (p_1, \ldots, p_n) \in \Delta_n$ is a probability distribution over a simplex of size n. Before presenting the algorithm for computing the Stackelberg equilibrium, we first present three facts.

Fact 2.3. *Let $j \in [m]$ and define $P_j = \{p \in \Delta_n \mid BR(p) = j\}$ then P_j is a convex polytope.*

Proof. Observe that

$$p \in P_j \Leftrightarrow U_2(p, j) \geq U_2(p, j') \quad \forall j' \in [m]. \tag{1}$$

We can see that this gives us a linear constraint for every j'

$$\sum_{i=1}^{n} p_i \cdot U_2(i, j) \geq \sum_{i=1}^{n} p_i \cdot U_2(i, j') \forall j' \in [m]. \tag{2}$$

Now, from (1) we get that P_j is an intersection of half spaces obtained by the above linear constraints. Thus, P_j is a convex polytope in Δ_n. □

Fact 2.4. *For any $j \in [m]$, $U_1(p, j)$ is a linear function in its first coordinate.*

Proof. We can see that

$$U_1(p, j) = \sum_{i=1}^{n} U_1(i, j).p_i.$$

This is a linear function in n dimensions, which proves the fact. □

If we combine the two facts above, we can see that for any j, $\arg\max_{p \in P_j} \; U_1(p, j)$ can be computed using an LP. This follows as P_j is a convex set from Fact 2.3 and that $U_1(p, j)$ is a linear objective from Fact 2.4.

Fact 2.5. *The Stackelberg equilibrium is* (p^*, j^*).

$$(p^*, j^*) = \arg\max_{(p,j) \in \Gamma} \; U_1(p, j),$$

where $\Gamma = \big\{(p, j) \;\big|\; p = \arg\max_{p \in P_j} U_1(p, j)\big\}$.

With these three facts, we can describe a general algorithm for efficiently computing Stackelberg equilibria in Algorithm 1.

Algorithm 1 Multiple LP Algorithm

```
**Input:** A two player general sum, with leader utility $U_1$ and follower utility $U_2$.
Initialize $\Gamma \leftarrow \emptyset$
**for** $j \in [m]$ **do**
    Solve the following LP to find $(p_j^*, j)$
        $$\arg\max_{p \in \Delta_m} \sum_{i=1}^{n} p_i U_1(i, j)$$
        $$\text{s.t.} \sum_{i=1}^{n} p_i (U_2(i, j) - U_2(i, j')) \geq 0 \qquad \forall j' \in [m]$$
    Update $\Gamma \leftarrow \Gamma \cup \{(p_j^*, j)\}$.
**end for**
Output leader-follower policy pair $(p^*, j^*) = \arg\max_{(p,j) \in \Gamma} U_1(p, j)$.
```

Proof of Theorem 2.2. At a high level, this algorithm loops over each follower action $j \in [m]$, and computes the leader's policy maximizing her utility, conditioned on j being a best response for the follower.

$$\arg\max_{p \in \Delta_m} \sum_{i=1} p_i U_1(i, j)$$
$$\text{s.t. } p \in P_j.$$

For each j, by Theorem 2.4 this optimization problem has a linear objective, and by Theorem 2.3, the space we are optimizing over is convex. Therefore, this can be solved in polynomial time via standard techniques.

Computing the Stackelberg optimal strategy boils down to finding the max of a piecewise linear function with m (number of follower actions) pieces, and in an n (number of leader actions) dimensional space. □

Figure 1 gives us a way to think about this problem. We've partitioned the space of leader policies Δ_n into parts, each of which is convex by Theorem 2.3, and whose optimal solution is on a vertex by Theorem 2.4.

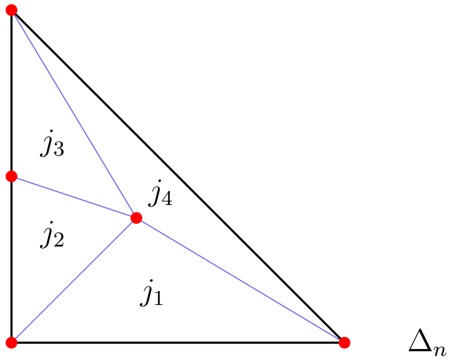

Figure 1: The Leader's strategy space can be divided into convex polytopes, each with a single best follower action. The action $j_\ell \in [m]$ inside each region is the follower's best response for that set of leader policies. To find the optimal action, we only need to check the vertices of these polytopes.

3 Applications

3.1 Security Stackelberg Games

Consider this commonly occurring interaction in nature: a defender would like to protect a set of vulnerable targets from attackers but has limited resources. This could be train operators catching people who skip fares, police protecting a neighborhood, governments catching poachers, etc.

These kinds of interactions can be modeled as Stackelberg games, and algorithms like multiple LP (Algorithm 1) can be used to compute optimal strategies for the defenders. The leader is a defender who must decide on an allocation of resources to defend their infrastructure. The follower is an attacker who then picks which part of the infrastructure to attack. Valid actions for the leader are a distribution over allocations of resources $R_1, \ldots, R_n$ to targets $T_1, \ldots, T_m$. Valid actions for the follower is a distribution over the m targets.

An attack is successful when the target chosen by the attacker is not allocated a resource. The utility of a successful attack for the attacker is $U_A(\text{successful}) \geq U_A(\text{unsuccessful})$, and the utility of a successful defense for the defender is $U_D(\text{unsuccessful}) \geq U_D(\text{successful})$.

Valid assignments of resources to targets are be described by a bipartite graph. When each resource can protect a single target, the leader picks a matching on this graph (Figure 2) as her pure strategy. More generally, when resources can be used to cover subsets of targets, any valid assignment of resources to targets is a pure strategy. The mixed strategy of the leader, induces a *coverage* probability, a vector of probabilities indicate how often each target is protected, which is the only utility relevant piece of information for both users.

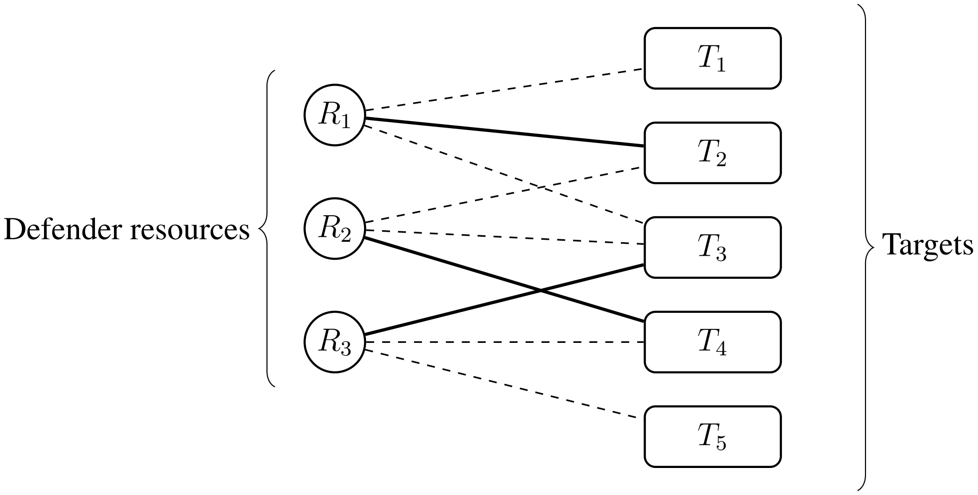

Figure 2: This figure describes a security game. The defender's resources are on the left, the nodes on the right are the potential targets, and the edges describe how resources can be allocated to targets. The filled in edges show one allocation of resources to targets. The defender (leader) commits to a distribution over matchings of resources to targets, and the attacker (follower) picks which target to attack.

3.2 Strategic Classification

In many applications, an automated decision making system is attempting to make decisions using information supplied by agents with incentives. For example, the automated decision making system is attempting to decide who to hire given attributes on a resume supplied by potential applicants who want to be hired. Applicants may misrepresent certain attributes like their qualifications to increase the chance of being hired.

In general, we can imagine a learner (the leader) picking a hypothesis $h^* \in \mathcal{H}$ and a user (the follower) who has attribute x but supplies an attribute z instead. The learner achieves a utility $U_1(h^*, x)$ of 1 when the user is correctly classified and 0 otherwise. The user achieves a utility of $U_{x,z}(h^*, z) = V_2(h^*, z) - \mathrm{cost}(x, z)$ where $V_2(h^*, z)$ is the utility of h^*'s classification of attribute z, and $\mathrm{cost}(x, z)$ is the cost of supplying attribute z when the user's true attribute is x.

This setting can be stated and solved as a Stackelberg game.