Convex-Concave Min-Max Stackelberg Games

Denizalp Goktas, Amy Greenwald
Brown University, Computer Science Department
BROWN

Background: Much progress has been made on **min-max optimization with independent feasible sets:** $\min_{x \in X} \max_{y \in Y} f(x, y)$

But little is known on **min-max optimization with dependent feasible sets** which have applications in deep learning, optimization, and algorithmic game theory:

$$\min_{x \in X} \max_{y \in Y : g(x,y) \geq 0} f(x, y)$$

Assumption: X and Y are compact-convex, f is continuous and convex-concave, g is a vector-valued, continuous, convex-concave function, which gives rise to an *interior feasible point*.

Interpretation: zero-sum sequential, i.e., **min-max Stackelberg game,** between x- and y-players as the order of the min and max matter:

$$\min_{x \in X} \max_{y \in Y : g(x,y) \geq 0} f(x, y) \neq \max_{y \in Y} \min_{x \in X : g(x,y) \geq 0} f(x, y)$$

A solution $(x^*, y^*) \in X \times Y$ to this problem can be modelled as a **Stackelberg equilibrium**:

$$\max_{y \in Y : g(x^*, y) \geq 0} f(x^*, y) \leq f(x^*, y^*) \leq \min_{x \in X} \max_{y \in Y : g(x,y) \geq 0} f(x, y)$$

y-player best responds to x^*.

x-player best-responds to the y-player's best response.

Abstract: We introduce the first polynomial-time algorithm to solve Convex-Concave min-max Stackelberg games.

Tools: We define the value function of the game as:

$$V(x) = \max_{y \in Y : g(x,y) \geq 0} f(x, y)$$

We can then re-express the min-max Stackelberg game as:

$$\min_{x \in X} V(x)$$

Under our assumption V is continuous and convex. **If we can compute a subgradient of V we can then run a subgradient method!**

Theorem (Subdifferential Envelope Theorem)

Let $\mathcal{L}_x(y, \lambda) = f(x, y) + \sum_{k=1}^{K} \lambda_k g_k(x, y)$.

Suppose that $y^*(\hat{x}), \lambda^*(\hat{x}, y^*(\hat{x}))$ is a solution to

Then, $$V(\hat{x}) = \max_{y \in Y} \min_{\lambda \in \mathbb{R}_+^K} \mathcal{L}_{\hat{x}}(y, \lambda)$$

$$\nabla_x \mathcal{L}_x(y^*(\hat{x}), \lambda^*(\hat{x}, y^*(\hat{x})))$$

is a subgradient of V at $\hat{x}$.

Algorithm idea:

1) Run gradient ascent on $f(x^{(t)}, y)$ to obtain the optimal y for $x^{(t)}$.
2) Compute a subgradient V at $x^{(t)}$.
3) Take a gradient descent step on V and obtain $x^{(t+1)}$.

Nested GDA

For $t = 1, \ldots, T^x$:
$$x^{(t+1)} = \Pi_X\left[x^{(t)} - \mathcal{L}_x\left(y^{(t)}, \lambda^*(x^{(t)}, y^{(t)})\right)\right]$$
For $s = 1, \ldots, T^y$:
$$y^{(t+1)} = \Pi_Y[y^{(t)} + \nabla_y f(x^{(t)}, y^{(t)})]$$

Theorem: *The iteration complexities of Nested GDA for min-max Stackelberg games are given as follows. Here, μ_x and μ_y are strong convexity/concavity parameters and ε is the approximation quality of the equilibrium.*

Properties of f	Iteration Complexity
μ_x-Strongly-Convex-μ_y-Strongly-Concave	$\tilde{O}(\varepsilon^{-1})$
μ_x-Strongly-Convex-Concave	$O(\varepsilon^{-2})$
Convex-μ_y-Strongly-Concave	$\tilde{O}(\varepsilon^{-2})$
Convex-Concave	$O(\varepsilon^{-3})$

Experiments: We observe that the computation of equilibria in a large class of markets is a convex-concave min-max Stackelberg game. Experiments suggest how smoothness properties affect the convergence of our algorithms.

Blue/orange: *algorithm's convergence rate* ***Red:*** *Predicted $O(1/\sqrt{T})$ convergence rate.*