Evolution Strategies for Approximate Solution of Bayesian Games

Zun Li, Michael P. Wellman
University of Michigan, Ann Arbor
{lizun, wellman}@umich.edu

Symmetric Bayesian Games

- An SBG consists of $(\mathcal{N}, \mathcal{T}, \mathcal{A}, \mathcal{P}, u)$
- $\mathcal{N} = \{1, \ldots, N\}$ the set of agents
- $\mathcal{T} \in \mathbb{R}^T$ the set of types
- $\mathcal{A} \in \mathbb{R}^A$ the set of actions
- Each agent rendered a type $t \sim \mathcal{P}$ i.i.d. before game starts, choose action a
- Payoff symmetry: $u(a_n, a_{\pi(1)}, \ldots, a_{\pi(N)} \mid t_n) = u(a_n, a_1, \ldots, a_N \mid t_n)$ for any permutation π of order $N-1$

Bayesian-Nash Equilibrium

- Pure strategy $s : \mathcal{T} \to \mathcal{A}$
- $\mathcal{S}$ the set of all pure strategies; we represent each as a neural network
- Mixed strategy $\sigma \in \Delta(\mathcal{S})$
- The *deviation payoff* $u(s, \boldsymbol{\sigma})$ is the expected payoff received by one agent choosing pure strategy s while the other $N-1$ choose σ
- $\textsc{Regret}(\sigma) \triangleq \max_{s \in \mathcal{S}} u(s, \boldsymbol{\sigma}) - u(\sigma, \boldsymbol{\sigma})$. If $\textsc{Regret}(\sigma) \leq \epsilon$ and $\sigma \in \mathcal{S}$ (or $\in \Delta(\mathcal{S})$), it is called an ϵ-PBNE (or ϵ-MBNE)

Black-Box Games

- Game represented by a black-box oracle $\mathcal{O} : \mathcal{S}^N \times \Omega \to \mathbb{R}^N$, where Ω is the set of random seeds
- Game analyst is aware of $\mathcal{N}, \mathcal{T}, \mathcal{A}$ and player symmetry, but neither the type distribution $\mathcal{P}$ nor a direct representation of the payoff function u

Natural Evolution Strategies

- Goal: optimize a black-box function $F(\boldsymbol{\theta})$ with respect to network weights $\boldsymbol{\theta}$
- Approach: optimize a Gaussian smoothing function $\mathbb{E}_{\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})}[F(\boldsymbol{\theta} + \nu\boldsymbol{\varepsilon})]$ by constructing finite difference approximation of the gradients

Algorithm 1: NES

Input: Black-box function F, hyperparameters J, α, ν
Output: Approximate maximum $\boldsymbol{\theta}$ of F

Algorithm NES(F, J, α, ν)
  Initialize $\boldsymbol{\theta}$;
  for $i = 1, 2, \ldots$ **do**
    Sample $\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_J \sim \mathcal{N}(0, \boldsymbol{I})$;
    $\forall j, r_{j+} \leftarrow F(\boldsymbol{\theta} + \nu\boldsymbol{\varepsilon}_j), r_{j-} \leftarrow F(\boldsymbol{\theta} - \nu\boldsymbol{\varepsilon}_j); r_j \leftarrow r_{j+} - r_{j-}$;
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \frac{1}{J\nu} \sum_j r_j \boldsymbol{\varepsilon}_j$;
  end
  return $\boldsymbol{\theta}, F(\boldsymbol{\theta})$;

Minimax-NES for PBNE

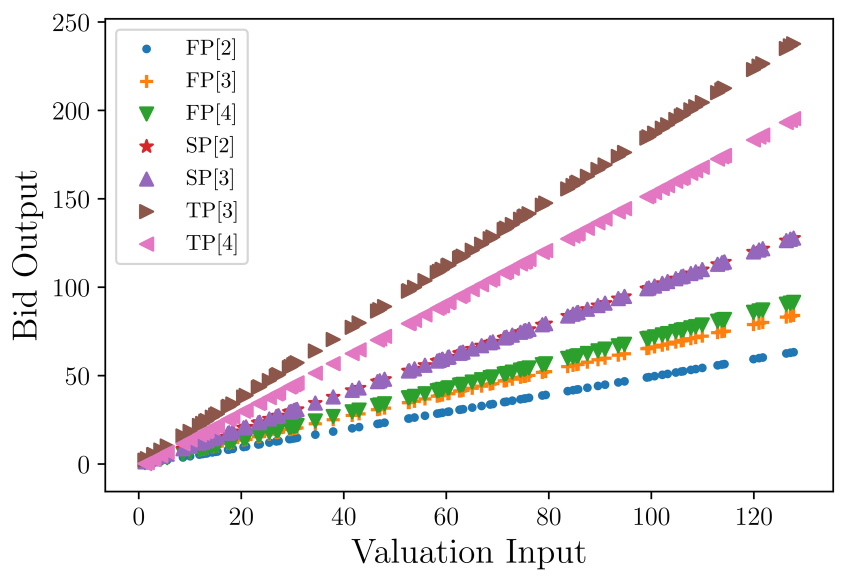
- Minimax formulation: $\min_s \textsc{Regret}(s) = \min_s \max_{s'} u(s', \boldsymbol{s}) - u(s, \boldsymbol{s})$

Algorithm 2: Minimax-NES for PBNE

Input: Payoff Oracle $\mathcal{O}$, hyperparameters $J_1, J_2, \alpha_1, \alpha_2, \nu_1, \nu_2$
Output: Approximate PBNE $s_{\boldsymbol{\theta}}$

Function MinusRegret$(\boldsymbol{\theta})$
  $V \leftarrow \mathcal{O}(s_{\boldsymbol{\theta}}, \boldsymbol{s_\theta})$;
  $\boldsymbol{\theta'}, DEV \leftarrow$ NES$(\mathcal{O}(\cdot, \boldsymbol{s_\theta}), J_1, \alpha_1, \nu_1)$;
  return $V - DEV$;
Algorithm MiniMax()
  return NES(MinusRegret$, J_2, \alpha_2, \nu_2)$

- Canonical solution for N-player single-item first-price auction $FP[N]$ is $s(t) = \frac{N-1}{N} t$.
- Second-price $SP[N] : s(t) = t$.
- Third-price $TP[N] : s(t) = \frac{N-1}{N-2} t$.



Incremental Strategy Generation for MBNE

- ISG discreterizes the strategy space as a finite strategy set S, and iteratively enlarges S via best responses
- two components: a meta-solver and a best response oracle

Algorithm 3: Incremental Strategy Generation

Input: Payoff Oracle $\mathcal{O}$. Meta-solver MS. Hyperparameters J, α, ν ;
Output: A finite strategy set S, a mixed strategy σ over S

Initial strategy set $S = \{s_0\}$, a singleton distribution σ with $\sigma(s_0) = 1$;
for $i = 1, 2, \ldots$ **do**
  $\sigma \leftarrow MS(\mathcal{O}, S, \sigma)$;
  $s', DEV \leftarrow$ NES$(\mathcal{O}(\cdot, \boldsymbol{\sigma}), J, \alpha, \nu)$;
  $S \leftarrow S \cup \{s'\}$;
end

- Given an iteration with restricted strategy set S, a meta-solver outputs a probability mixture over S, which could be:
  - Self-play: all mass on the last pure strategy
  - Fictitious play: uniform mixture on S
  - Replicator dynamics: a Nash equilibrium on the finite game defined by S
- And then NES generates a best-reponse against this mixture into S

Experiments

- Four methods: minimax-NES (MM) and self-play (SP) computing PBNE; fictitious play (FP) and replicator dynamics (RD) solving for MBNE
- Environments: N-player K-good market-based scheduling ($MBS[N, K]$) & homogeneous-good auctions ($HG[N, K]$).
- Both multidimensional types and actions possessing no analytic solutions