Theory and Methods for Reinforcement Learning

Prof. Volkan Cevher
volkan.cevher@epfl.ch

Lecture 4: Linear Programming

Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-618 (Spring 2022)

lions@epfl

HASLERSTIFTUNG SDSC FNSNF Google AI ZEISS
Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation
Office of Naval Research Global Science & Technology
erc EPFL

License Information for Theory and Methods for Reinforcement Learning (EE-618)

Recap - Reinforcement learning objective

- Reinforcement Learning: Sequential decision making in unknown environment
- Markov decision process: $M = (\mathcal{S}, \mathcal{A}, P, r, \mu, \gamma)$
- Stationary stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, $a_t \sim \pi(\cdot|s_t)$
- State-value function: $V^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, \pi\right]$
- Performance objective: $\max_\pi (1-\gamma) \sum_{s \in \mathcal{S}} \mu(s) V^\pi(s)$

Challenges:
- Infer long-term consequences based on limited, noisy short-term feedback.
- Unknown dynamics - Knowledge only through sampled experience.
- Large state and actions spaces.
- Highly nonconvex objective.

Motivation

◦ Approximate dynamic programming

- Attempting to find approximate fixed-point solutions to the (nonlinear) Bellman equation.
- Pros:
  - + Well studied setting for tabular MDPs that comes with theoretical convergence guarantees.
    - See Lectures 2 and 3.
  - + Deep-learning variants (e.g., DQN [18]) are powerful.
- Cons:
  - – Training can oscillate or even diverge under the simplest parameterizations or in offline settings.
    - For divergent examples for TD-learning with nonlinear parameterizations, see e.g., Ex 6.6 and 6.7 in [3].
    - For divergent example for approximate VI with linear parameterizations, see e.g., Ex. 6.11 in [3].
  - – Incompatible with classical machine-learning tools that are rooted in convex optimization.

Motivation (cont.)

◦ The linear programming approach

- Introduces an alternative convex viewpoint that formulates the RL problem as a linear program.
- Overviews recent scalable algorithms with theoretical guarantees rooted in the LP approach.
- Highlights how historical key limitations have been eliminated.

Revisit Bellman optimality equation

- Finding $V^\star$ satisfying Bellman optimality equation can be written as a feasibility problem:

$$\min_{V} \;\; 0$$
$$\text{s.t.} \;\; V(s) = \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V(s') \right], \quad \forall\, s \in \mathcal{S}.$$

- The only feasibile point is $V^\star$.
- The above constraint suggests that $V^\star(\cdot)$ is the "least feasible solution" of $V(\cdot)$ satisfying

$$V(s) \geq \; r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V(s'), \quad \forall\, s \in \mathcal{S},\; a \in \mathcal{A}.$$

- Note that the new constraints above is linear in $V(\cdot)$ $\quad\Longrightarrow\quad$ Linear Programming (LP) .

Solving MDPs with LP - Primal LP formulation

Primal LP

Let $\mu(s) > 0, s \in \mathcal{S}$ be the initial distribution (or any positive weights).

$$\begin{aligned} \min_{V} \quad & (1-\gamma) \sum_{s \in \mathcal{S}} \mu(s) V(s) \\ \text{s.t.} \quad & V(s) \geq \; r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V(s'), \quad \forall\, s \in \mathcal{S},\; a \in \mathcal{A}. \end{aligned} \tag{P}$$

Remarks:

- The optimal value function $V^\star$ is the unique solution to the above LP.
- Number of decision variables: $|\mathcal{S}|$, number of constraints: $|\mathcal{S}| \times |\mathcal{A}|$.
- An optimal (deterministic) policy is the associated greedy policy

$$\pi^\star(s) \in \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\star(s) \right]. \tag{1}$$

- The factor $(1-\gamma)$ in the objective ensures that the dual variables are in the probability simplex.

Solving MDPs with LP - Primal LP formulation (cont.)

Corollary (LP Formulation and $V^\star$)

$V^\star$ is the solution to the above LP formulation for any positive weights $\{\mu(s)\}$.

Proof Sketch

- *First, we establish that $V^\star$ is a feasible solution.*
- *Constraints hold for any $a \in \mathcal{A}$, so choose $a = \pi^\star(s)$, the deterministic stationary optimal policy.*
- *Hence, we have $\left(I - \gamma P^{\pi^\star}\right)V^\star \geq R^{\pi^\star}$.*
- *Then, we need to show that $V^\star$ minimizes the objective.*
- *By the monotonicity property of the Bellman operator, we get that $V \geq V^\star$, for any feasible V.*

Remark:

- When we introduce function approximations, the quality of the approximate minimizer V depends on the choice of the positive weights $\{\mu(s)\}$ (see Slide 19).

A closer look at the primal LP

Recall: Primal LP

Let $\mu(s) > 0, s \in \mathcal{S}$ be the initial distribution (or any positive weights).

$$\begin{aligned} \min_{V} \quad & (1-\gamma)\sum_{s\in\mathcal{S}} \mu(s)V(s) \\ \text{s.t.} \quad & V(s) \geq \; r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V(s'), \quad \forall\, s \in \mathcal{S},\; a \in \mathcal{A}. \end{aligned} \tag{P}$$

Observations:

- Any $V^\star$ is feasible as

  $$V^\star(s) = \mathcal{T}V^\star(s) \geq r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathsf{P}(s'|s,a)V^\star(s'),\ \forall a \in \mathcal{A}.$$

  This implies feasibility.

- For any feasible V, we have $V \geq \mathcal{T}V$. By monotonicity of the Bellman operator $\mathcal{T}$, we have

  $$V \geq \mathcal{T}V \geq \mathcal{T}^2 V \geq \cdots \geq \mathcal{T}^\infty V = V^\star.$$

  This implies optimality.

Solving MDPs with dual LP

Dual LP formulation

$$
\begin{aligned}
\max_{\lambda \geq 0} \quad & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} r(s,a) \lambda(s,a) \\
\text{s.t.} \quad & \sum_{a \in \mathcal{A}} \lambda(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s' \in \mathcal{S}, a' \in \mathcal{A}} \mathsf{P}(s|s',a') \lambda(s',a'), \quad \forall\, s \in \mathcal{S}.
\end{aligned} \tag{D}
$$

Remarks:

- The number of decision variables: $|\mathcal{S}| \times |\mathcal{A}|$.
- The number of constraints: $|\mathcal{S}| + |\mathcal{S}| \times |\mathcal{A}|$.
- The constraints implicitly implies the decision variables are in the probability simplex.
- The solution to the dual LP, $\lambda^\star$, corresponds to the state-action occupancy of $\pi^\star$.

Dual interpretation

- For any policy π and $s_0 \sim \mu$, define the state-action visitation distribution $\lambda^\pi(s,a)$ as

$$\lambda^\pi(s,a) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(s_t = s, a_t = a \,|\, s_0 \sim \mu, \pi)$$

- We can write

$$(1-\gamma)\mathbb{E}_{s\sim\mu}[V^\pi(s)] = (1-\gamma)\,\mathbb{E}\left[\sum\nolimits_{t=0}^{\infty}\gamma^t r(s_t,a_t) \mid s_0 \sim \mu\right] \quad \Rightarrow \text{ primal objective (P)}$$

$$= (1-\gamma)\sum_{s\in\mathcal{S},a\in\mathcal{A}}\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(s_t = s, a_t = a \,|\, s_0 \sim \mu, \pi) r(s,a)$$

$$= \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\lambda^\pi(s,a) r(s,a) \quad \Rightarrow \text{ dual objective (D)}$$

- Easy to verify that $\lambda^\pi(s,a)$ satisfies the constraints in the dual LP.

A closer look at the dual LP

Recall: Dual LP

$$\begin{aligned} \max_{\lambda} \quad & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r(s,a)\lambda(s,a) \\ \text{s.t.} \quad & \sum_{a\in\mathcal{A}} \lambda(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s'\in\mathcal{S},a'\in\mathcal{A}} \mathsf{P}(s|s',a')\lambda(s',a'), \quad \forall\, s\in\mathcal{S}, \\ & \lambda(s,a)\geq 0, \quad \forall\, s\in\mathcal{S}, a\in\mathcal{A}. \end{aligned} \tag{D}$$

Observations:

- For any policy π and $s_0 \sim \mu$, define the state-action visitation distribution

$$\lambda^{\pi}(s,a) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{P}(s_t = s, a_t = a \,|\, \pi, s_0\sim\mu).$$

  By Markov property, we have (see supplementary material for details)

$$\lambda^{\pi}(s,a) = \mu(s)\pi(a|s) + \gamma\sum_{s',a'}\pi(a|s)\mathsf{P}(s|s',a')\lambda^{\pi}(s',a').$$

  Summing over a implies feasibility.

A closer look at the dual LP

Dual LP

$$\max_{\lambda} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} r(s,a)\lambda(s,a)$$
$$\text{s.t.} \sum_{a\in\mathcal{A}} \lambda(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s'\in\mathcal{S},a'\in\mathcal{A}} \mathsf{P}(s|s',a')\lambda(s',a'), \quad \forall\, s\in\mathcal{S},$$
$$\lambda(s,a) \geq 0, \quad \forall\, s\in\mathcal{S}, a\in\mathcal{A}. \tag{D}$$

Observation:

- We can show the objective equivalence of (P) and (D):

$$(1-\gamma)\sum_{s\in\mathcal{S}} \mu(s)V^{\pi}(s) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \,|\, \pi, s_0\sim\mu\right]$$
$$\Updownarrow$$
$$\sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} r(s,a)\lambda^{\pi}(s,a) = (1-\gamma) \sum_{s\in\mathcal{S},a\in\mathcal{A}} \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t=s, a_t=a \,|\, \pi,\, s_0\sim\mu) r(s,a).$$

A closer look at the dual LP (cont.)

Dual LP

$$\begin{aligned} \max_{\lambda} \quad & \sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}} r(s,a)\lambda(s,a) \\ \text{s.t.} \quad & \sum_{a\in\mathcal{A}} \lambda(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s'\in\mathcal{S},a'\in\mathcal{A}} \mathsf{P}(s|s',a')\lambda(s',a'), \quad \forall\, s\in\mathcal{S}, \\ & \lambda(s,a) \geq 0, \quad \forall\, s\in\mathcal{S}, a\in\mathcal{A}. \end{aligned} \tag{D}$$

Observations:

- For any λ feasible to the dual LP, we can define a policy

$$\pi_\lambda(a\,|\,s) = \frac{\lambda(s,a)}{\sum_{a\in\mathcal{A}}\lambda(s,a)}.$$

  It then holds $\lambda^{\pi_\lambda} = \lambda$.

- Note that $\lambda^\star(s,a) = \lambda^{\pi^\star}(s,a)$ and $\pi^\star(a\,|\,s) = \frac{\lambda^\star(s,a)}{\sum_{a\in\mathcal{A}}\lambda^\star(s,a)}$. (self-check)

- Optimal policy does not depend on μ. (LP sensitivity analysis)

Finding the optimal policy

- Primal LP approach:
  - Solve primal LP to obtain for the optimal value function $V^\star$
  - Then construct the optimal policy (deterministic) through the greedy policy

$$\pi^\star(s) \in \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum\nolimits_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a) V^\star(s') \right].$$

- Dual LP approach:
  - Solve the dual LP to obtain the optimal state-action occupancy $\lambda^\star$
  - Then construct the optimal policy (randomized) by

$$\pi^\star(a \,|\, s) = \frac{\lambda^\star(s,a)}{\sum_{a \in \mathcal{A}} \lambda^\star(s,a)}.$$

- Reference: [Puterman 1994] [25] (Section 6.9)

Dynamic programming vs Linear programming (exact solutions)

Algorithm	Component	Output
Value Iteration (VI)	Bellman Optimality Operator $\mathcal{T}$	$V^\star$ (control)
Policy Iteration (PI)	(Multiple) Bellman Operator $\mathcal{T}^\pi$ + Greedy Policy	$\pi^\star$ (control)
Linear Programming (LP)	LP solver (Simplex, Interior Point Method)	$V^\star, \pi^\star$ (control)

Dynamic Programming:

- Simple iterative updates.
- Polynomial complexity in $|\mathcal{S}|$ and $|\mathcal{A}|$.
- Works better for small problems.

Linear Programming:

- Rich library of fast LP solvers.
- Polynomial complexity in $|\mathcal{S}|$ and $|\mathcal{A}|$.
- Works better for large problems.

The LP approach - Pros and Cons

- Why is this useful?
  - Defining optimality is simple: no value functions, no fixed-point equations, just the numerical objective.
  - Easily comprehensible with an optimization background.
  - A disciplined convex optimization template with a rich set of algorithms.

- End User License Agreement:
  - Need to ensure $\sum_{a \in \mathcal{A}} \lambda(s, a) > 0$ to extract a policy.
  - Number of variables is large.
  - Intractable number of constraints.
  - Constraints may be not satisfied when working with function approximators.

Beyond exact solutions - A bit of history of approximate linear programming (ALP)

- [Manne 1960] [17]
  - Formulated the primal LP over value functions and showed equivalence to Bellman equations.
- [Borkar 1988] [4] and [Hérnandez-Lerma & Lasserre 1996, 1999] [11, 12]
  - Studied the LP approach to MDPs with continuous state and action spaces.
  - The corresponding LPs are infinite-dimensional.
- [Schweitzer & Seidman 1982] [29]
  - Proposed linear function approximators to reduce the number of decision variables
  - Proposed a relaxation to reduce the number of constraints.
- [De Farias & Van Roy 2003, 2004] [7, 8]
  - Analyzed the reduction [Schweitzer & Seidman 1982] [29].
  - Inspired some follow-up work in RL [Petrik et al. 2009,2010] [23, 22], [Desai et al. 2012] [9], [Abbasi-Yadkori et al. 2014] [1], [Lakshminarayanan et al. 2018] [15].

Prior works in ALP - Linear function approximation

Large-scale MDPs $\Rightarrow$ Large-scale optimization

- Reduce the number of decision variables by projecting onto a lower-dimensional subspace.
  - Let $\phi_1, \ldots, \phi_k : \mathcal{S} \to \mathbb{R}$ be k basis functions (or features).
  - $\Phi := \begin{bmatrix} \phi_1 & \ldots & \phi_k \end{bmatrix} \in \mathbb{R}^{|\mathcal{S}| \times k}$ is the corresponding feature matrix.
  - The (ALP) is obtained by adding the linear constraint $V = \Phi\theta = \sum_{i=1}^{k} \theta_i \phi_i$ to the original primal LP (P).

Approximate linear program [Schweitzer & Seidman 1982]

$$\begin{aligned} \min_{\theta \in \mathbb{R}^k} \quad & (1-\gamma) \sum_{s \in \mathcal{S}} \mu(s)(\Phi\theta)(s) \\ \text{s.t.} \quad & (\Phi\theta)(s) \geq \; r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathrm{P}(s'|s,a)(\Phi\theta)(s'), \quad \forall \; s \in \mathcal{S}, \; a \in \mathcal{A}. \end{aligned} \tag{ALP}$$

Prior works in ALP - Linear function approximation (cont.)

Assumptions:

- The set $\{\phi_1, \ldots, \phi_k\}$ is linearly independent.
- $\mathbf{1} \in \text{span}\big(\{\phi_1, \ldots, \phi_k\}\big) := \{\Phi\theta \mid \theta \in \mathbb{R}^d\}$. This ensures that (ALP) is feasible [7] .
- The values $\sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s,a)\phi_i(s')$ and $\mu^\top \phi_i$, $i = 1, \ldots, k$, can be accessed in $\mathcal{O}(1)$ time.

Quality of the approximate solution (Th.2 in [De Farias & Van Roy 2003] [7])

$$\|V^\star - V^\star_{\text{ALP}}\|_{1,\mu} \leq \frac{2}{1-\gamma} \underbrace{\min_\theta \|V^\star - \Phi\theta\|_\infty}_{\varepsilon_{\text{approx}}:\ \text{approximation error}}.$$

Notation:

- $\theta^\star_{\text{ALP}}$ is optimal to (ALP) and $V^\star_{\text{ALP}} = \Phi\theta^\star_{\text{ALP}}$ is the approximate value function.
- $\|V\|_{1,\mu} := \sum_{s \in \mathcal{S}} \mu(s)|V(s)|$ is the μ-weighted ℓ_1-norm, where $\mu > 0$.
- $\Phi\theta^\star$ is the $\|\cdot\|_\infty$-norm projection of $V^\star$ to the subspace $V = \Phi\theta$.
- $\varepsilon_{\text{approx}} := \min_\theta \|V^\star - \Phi\theta\|_\infty = \|V^\star - \Phi\theta^\star\|_\infty$ is called the approximation error.

Prior works in ALP - Linear function approximation (cont.)

Quality of the approximate solution

$$\|V^\star - V^\star_{\text{ALP}}\|_{1,\mu} \leq \frac{2}{1-\gamma}\varepsilon_{\text{approx}}.$$

Remarks:

- $\varepsilon_{\text{approx}} = \min_\theta \|V^\star - \Phi\theta\|_\infty$ captures the approximation power of the feature map.
- If $V^\star \in \text{span}\big(\phi_1, \ldots, \phi_k\big)$, then $V^\star = \Phi\theta^\star_{\text{ALP}}$.
- In general, $\|V^\star - V^\star_{\text{ALP}}\|_{1,\mu} = \mathcal{O}(\varepsilon_{\text{approx}})$.
- Focus on finding a good basis, leaving the search of the "right" weights to an LP solver.

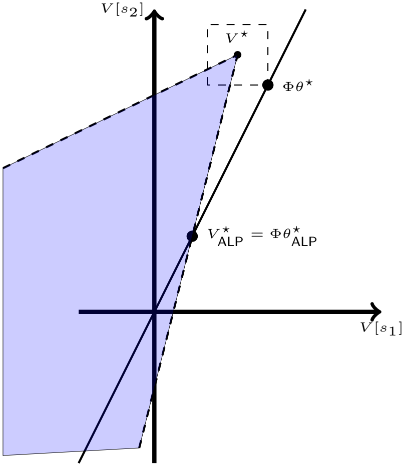


Figure: Graphical interpretation of ALP [7]

Prior works in ALP - Constraint sampling

◦ Reduce the number of constraints by constraint sampling.

- (x, a) is treated as an uncertainty parameter.
- $\mathcal{S} \times \mathcal{A}$ is the uncertainty space.
- $\mathbb{P}$ is a probability distribution on $\mathcal{S} \times \mathcal{A}$.
- $\{(s_i, a_i)\}_{i=1}^N$ i.i.d. samples on $(\mathcal{S} \times \mathcal{A}, \mathbb{P})$.
- $\mathcal{N} \subset \mathbb{R}^k$ is a bounding set.
- The relaxed LP (RLP) is obtained from (ALP) by considering only the N sampled constraints, and restricting $\theta \in \mathcal{N}$.

Relaxed linear program [De Farias & Van Roy 2001] [8]

$$\begin{aligned} \min_{\theta \in \mathcal{N}} \quad & (1-\gamma) \sum_{s \in \mathcal{S}} \mu(s)(\Phi\theta)(s) \\ \text{s.t.} \quad & (\Phi\theta)(s_i) \geq r(s_i, a_i) + \gamma \sum_{s' \in \mathcal{S}} \mathsf{P}(s'|s_i, a_i)(\Phi\theta)(s'), \quad \forall\, i = 1, \ldots, N. \end{aligned} \tag{RLP}$$

Prior works in ALP - Constraint sampling (cont.)

Assumptions:

- The set $\mathcal{N} \subset \mathbb{R}^k$ is compact, i.e., bounded and closed.
- The optimal solution $\theta^\star_{\text{ALP}}$ to (ALP) is in $\mathcal{N}$.
- The sampling probability distribution is $\mathbb{P} = \lambda^{\pi^\star}$, i.e., the state-action visitation distribution induced by an optimal policy $\pi^\star$.

How many samples give a good solution (Th.3.1 in [De Farias & Van Roy 2004] [8])

Let $\varepsilon, \delta \in (0,1)$. If $N \geq \tilde{\mathcal{O}}\Big(\frac{4k\log(\frac{1}{\delta})}{(1-\gamma)\varepsilon}\frac{\sup_{\theta\in\mathcal{N}}\|V^\star - \Phi\theta\|_\infty}{\mu^\top V^\star}\Big)$, then with probability at least $1-\delta$, we have

$$\|V^\star - V^\star_{\text{RLP}}\|_{1,\mu} \leq \|V^\star - V^\star_{\text{ALP}}\|_{1,\mu} + \varepsilon\|V^\star\|_{1,\mu},$$

where the probability is taken over the random sampling of constraints.

Notation:

- $\theta^\star_{\text{RLP}}$ is optimal to (RLP) and $V^\star_{\text{RLP}} = \Phi\theta^\star_{\text{RLP}}$ is the approximate value function.
- $\varepsilon \in (0,1)$ is the desired approximation accuracy.
- $\delta \in (0,1)$ is the desired confidence level.

Prior works in ALP - Constraint sampling (cont.)

Remarks:

- (RLP) is a relaxation of (ALP).
- The constraint $\theta \in \mathcal{N}$ ensures that that the optimal value of (RLP) is bounded.
- The relaxed linear program (RLP) is random.
- $\theta^\star_{\text{RLP}}$ and $V^\star_{\text{RLP}} = \Phi\theta^\star_{\text{RLP}}$ are random variables.
- A lower bound on the number of samples needed to achieve an ε-accurate solution with probability at least $1-\delta$, is called the sample complexity of the problem.
- The sample complexity bound depends on the choice of the bounding set $\mathcal{N}$.
- The sample complexity bound requires access to samples from the optimal state-action visitation distribution (which is not known a priori).

Common theme of all prior ALP works

- Reduce the number of decision variables by projecting on a low-dimensional subspace.
- Reduce the number of constraints (e.g., by constraint sampling).
- Solve the resulted LP with generic solver.
- Analyze the quality of the approximate solution.
- Either scale badly with the size of the state-action spaces or
- Require access to samples from a distribution that depends on the optimal policy.
- Require knowledge of dynamics or access to a simulator.
- Focus mainly on the approximation of the optimal value but not so much on the near optimal policy.

Is this the best we can do?

Some notation: towards an unconstrained problem.

- We will write an equivalent unconstrained problem.
- To simplify the notation we need to introduce a couple of operators:
  - $E : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}}$ such that $(EV)(s,a) = V(s)$.
  - $P : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}}$ such that $(PV)(s,a) = \sum_{s'} \mathsf{P}(s'|s,a)V(s')$.
- Their adjoint are:
  - $E^T : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ such that $(E^T\lambda)(s) = \sum_a \lambda(s,a)$.
  - $P^T : \mathbb{R}^{\mathcal{S}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ such that $(P^T\lambda)(s') = \sum_{s,a} \mathsf{P}(s'|s,a)\lambda(s,a)$.

Linear Programming - Summary

Primal LP:

$$\min_{V \in \mathbb{R}^{|\mathcal{S}|}} \quad (1-\gamma)\langle \mu, V \rangle \qquad \text{s.t.} \quad EV \geq r + \gamma P V. \tag{P}$$

Dual LP

$$\max_{\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \quad \langle \lambda, r \rangle \qquad \text{s.t.} \quad E^\top \lambda = (1-\gamma)\mu + \gamma P^\top \lambda, \quad \lambda \geq 0. \tag{D}$$

- Primal LP over value functions
- $|\mathcal{S}|$ decision variables and $|\mathcal{S}||\mathcal{A}|$ constraints
- $\forall\ V$ primal feasible $\Rightarrow V^\star \leq V$
- Optimal value function $V^\star$ is the optimizer
- Optimal policy is the associated greedy policy

- Dual LP over occupancy measures
- $|\mathcal{S}||\mathcal{A}|$ variables and $|\mathcal{S}| + |\mathcal{S}||\mathcal{A}|$ constraints
- $\forall$ policy π, the induced λ^π is dual feasible
- $\forall$ feasible $\lambda \Rightarrow \pi_\lambda$ has occupancy measure λ
- We have $\lambda^\star = \lambda^{\pi^\star}$ and $\pi^\star = \pi_{\lambda^\star}$

Towards the Lagrangian

- Instead of working solely with the primal or dual LP formulation, we work with an object between them
- Introducing the Lagrangian multipliers vector $\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$, we can write the Lagrangian as follows:

Primal LP:

$$\min_{V \in \mathbb{R}^{|\mathcal{S}|}} \quad (1-\gamma)\langle \mu, V\rangle \qquad \text{s.t.} \quad EV \geq r + \gamma PV. \tag{P}$$

Dual LP

$$\max_{\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \quad \langle \lambda, r\rangle \qquad \text{s.t.} \quad E^\top \lambda = (1-\gamma)\mu + \gamma P^\top \lambda, \quad \lambda \geq 0. \tag{D}$$

$$\Updownarrow$$

Saddle point formulation

$$\min_{V} \max_{\lambda \geq 0} (1-\gamma) \sum_{s \in \mathcal{S}} \langle \mu\,, V\rangle + \langle \lambda\,, r + \gamma PV + EV\rangle. \qquad \text{(Saddle-point problem)}$$

Minimax optimization

Bilinear min-max template

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x}) + \langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle - h(\mathbf{y}),$$

where $\mathcal{X} \subseteq R^p$ and $\mathcal{Y} \subseteq \mathbb{R}^n$.

- $f: \mathcal{X} \to \mathbb{R}$ is convex.
- $h: \mathcal{Y} \to \mathbb{R}$ is convex.

Convex-concave min-max template

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y}), \tag{2}$$

where $\Phi(\mathbf{x}, \mathbf{y})$ is convex in $\mathbf{x}$ and concave in $\mathbf{y}$.

Basic algorithms for minimax

- Given $\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} \Phi(\mathbf{x},\mathbf{y})$, define $V(\mathbf{z}) = [\nabla_{\mathbf{x}}\Phi(\mathbf{x},\mathbf{y}), -\nabla_{\mathbf{y}}\Phi(\mathbf{x},\mathbf{y})]$ with $\mathbf{z} = [\mathbf{x},\mathbf{y}]$.

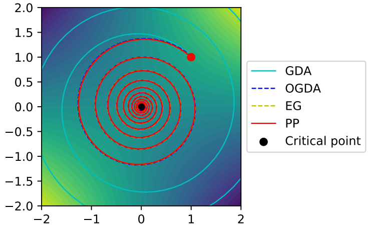


Figure: Trajectory of different algorithms for a simple bilinear game $\min_x \max_y xy$.

- (In)Famous algorithms
  - Gradient Descent Ascent (GDA)
  - Proximal point method (PPM)
  - Extra-gradient (EG)
  - Optimistic Gradient Descent Ascent (OGDA)
  - Reflected-Forward-Backward-Splitting (RFBS)

- EG and OGDA are approximations of the PPM
  - $\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(\mathbf{z}^k)$.
  - $\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(\mathbf{z}^{k+1})$.
  - $\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(\mathbf{z}^k - \alpha V(\mathbf{z}^{k-1}))$
  - $\mathbf{z}^{k+1} = \mathbf{z}^k - \eta [2V(\mathbf{z}^k) - V(\mathbf{z}^{k-1})]$
  - $\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(2\mathbf{z}^k - \mathbf{z}^{k-1})$

Primal-dual π-learning

Saddle point formulation

$$\min_{V} \max_{\lambda \in \Delta_{\mathcal{S}\times\mathcal{A}}} (1-\gamma)\sum_{s\in\mathcal{S}}\langle \mu\,, V\rangle + \langle \lambda\,, r + \gamma P V + E V\rangle. \qquad \text{(Saddle-point problem)}$$

- For know dynamic, it can be solved via primal-dual updates:
  - $V_{k+1} = V_k - \eta\Big((\gamma P - E)^\intercal \lambda_k + \mu\Big)$.
  - $\lambda_{k+1} = \lambda_k \odot e^{\eta(r+\gamma P V_k - E V_k)}$, where $\odot$ denotes entry wise multiplication.
- Gradients are expectations under μ_k and P

$\Rightarrow$ efficient stochastic implementation [Chen et al. 2018] [6], [Jin & Sidford. 2018] [13].

  - State-of-the art sample complexity for solving small MDPs.
  - $\mathcal{O}\left(\frac{|\mathcal{S}||\mathcal{A}|\log(\frac{1}{\delta})}{(1-\gamma)^4\varepsilon^2}\right)$ samples for finding an ε-optimal policy with probability at least $1-\delta$.

Scaling up

Large-scale MDPs $\Rightarrow$ Large-scale optimization

- Parameterize λ and V via linear functions
  - $\lambda_\nu = \Psi\nu$, for some feature matrix $\Psi \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|\times n}$
  - $V_\theta = \Phi\theta$, for some feature matrix $\Phi \in \mathbb{R}^{|\mathcal{S}|\times m}$

Relaxed saddle point formulation

$$\min_{\theta} \max_{\nu\in\Delta_{[n]}} (1-\gamma)\sum_{s\in\mathcal{S}}\langle\mu\,,\,\Phi\theta\rangle + \langle\lambda\,,\,\Psi^\top(r+\gamma P\Phi\theta + E\Phi\theta)\rangle$$

Scaling up (cont.)

Relaxed saddle point formulation

$$\min_{\theta} \max_{\nu \in \Delta_{[n]}} (1-\gamma) \sum_{s \in \mathcal{S}} \langle \mu \, , \, \Phi\theta \rangle + \langle \lambda \, , \, \Psi^\intercal (r + \gamma PV + EV) \rangle$$

◦ Primal-dual updates:

- $\theta_{k+1} = V\theta_k - \eta\Big((\gamma P\Phi - E\Phi)^\intercal \Psi \nu_k + \Phi^\intercal \mu\Big),$
- $\nu_{k+1} = \nu_k \odot e^{\eta \Psi^\intercal (r + \gamma P \Phi \theta_k - E\Phi\theta_k)}.$

◦ Implementable with only sample access to the columns of Ψ and the transition law P [Chen et al. 2018] [6].

- $\mathcal{O}\left(\frac{n\, m \log(\frac{1}{\delta})}{(1-\gamma)^4 \varepsilon^2}\right)$ samples for finding an $\varepsilon + \varepsilon_{\text{approx}}$-optimal policy with probability at least $1-\delta$.
- $\varepsilon_{\text{approx}}$ captures the expressivity of the approximation architecture.

Proximal point method (PPM)

◦ Consider the following smooth unconstrained optimization problem: $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$

Proximal point method for convex minimization.

For a step-size $\tau > 0$, PPM can be written as follows

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{x}^k\|^2 \right\} := \mathrm{prox}_{\tau f}(\mathbf{x}^k) \tag{3}$$

Observations: ◦ The optimality condition of (3) reveals a simpler PPM recursion for smooth f:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau\nabla f(\mathbf{x}^{k+1}).$$

◦ PPM is an **implicit**, non-practical algorithm since we need the point $\mathbf{x}^{k+1}$ for its update.

◦ Each step of PPM can be as hard as solving the original problem.

◦ Convergence properties are well understood due to Rockafellar [28].

PPM and minimax optimization

PPM applied to the minimax template: $\min_{\mathbf{x}\in\mathbb{R}^d}\max_{\mathbf{y}\in\mathbb{R}^n}\Phi(\mathbf{x},\mathbf{y})$

Define $\mathbf{z} = [\mathbf{x},\mathbf{y}]^\top$ and $\mathbf{V}(\mathbf{z}) = [\nabla_{\mathbf{x}}\Phi(\mathbf{x},\mathbf{y}), -\nabla_{\mathbf{y}}\Phi(\mathbf{x},\mathbf{y})]^\top$. PPM iterations with a step-size $\tau > 0$ is given by

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \tau\mathbf{V}(\mathbf{z}^{k+1}).$$

Derivation:

- For $\tau > 0$, $(\mathbf{x}^{k+1},\mathbf{y}^{k+1})$ is the unique solution to the saddle point problem,

$$\min_{\mathbf{x}\in\mathbb{R}^d}\max_{\mathbf{y}\in\mathbb{R}^n}\Phi(\mathbf{x},\mathbf{y}) + \frac{1}{2\tau}\|\mathbf{x}-\mathbf{x}^k\|^2 - \frac{1}{2\tau}\|\mathbf{y}-\mathbf{y}^k\|^2 \tag{4}$$

- Writing the optimality condition of the update in (4)

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau\nabla_{\mathbf{x}}\Phi(\mathbf{x}^{k+1},\mathbf{y}^{k+1}), \qquad \mathbf{y}^{k+1} = \mathbf{y}^k + \tau\nabla_{\mathbf{y}}\Phi(\mathbf{x}^{k+1},\mathbf{y}^{k+1}) \tag{5}$$

Observation:

- **PPM is an implicit algorithm.**
- For the bilinear problem, PPM is implementable!

Proximal point methods in the Bregman setup

Definition: Bregman distance

Let $\omega : \mathcal{X} \to \mathbb{R}$ be a distance generating function where ω is $1-$strongly convex w.r.t. some norm $\|\cdot\|$ on the underlying space and is continuously differentiable. The Bregman distance induced by $\omega(\cdot)$ is given by

$$D_\omega(\mathbf{z}, \mathbf{z}') = \omega(\mathbf{z}) - \omega(\mathbf{z}') - \nabla\omega(\mathbf{z}')^\top(\mathbf{z} - \mathbf{z}').$$

∘ The proximal point method in the Bregman setup reads as follows:

$$\boxed{\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \left\{ f(\mathbf{x}) + \frac{1}{\tau} D_\omega(\mathbf{x}, \mathbf{x}^k) \right\}}$$

Remarks:

- ∘ Choosing the negative entropy as a generating function $\omega(\mathbf{x}) = \langle \mathbf{x}, \log \mathbf{x} \rangle$, we obtain the KL divergence. Such $\omega(\mathbf{x})$ is 1-strongly convex in $\|\cdot\|_1$ norm.
- ∘ This choice will allow to avoid projection in the simplex constraints and it improves the dependence on the domain dimension.
- ∘ Now, we will see PPM in action on the Lagrangian.

REPS: a success story

- REPS is widely popular in the robotics community.
- It is an application of proximal point to the Dual LP.
- A robot trained with REPS manages to play table tennis.

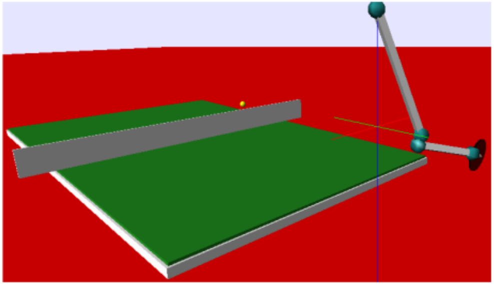


Figure: Source: Relative Entropy Policy Search [21]

Towards REPS: proximal point on the Dual LP

- Recall: Proximal point is an general implicit method.
- However, for a linear objective PPM can be implemented.
- Hence, we can apply proximal point updates on the Lagrangian that is just a bilinear form.

Recall: Dual LP

$$\begin{aligned} \lambda_k \;&=\; \mathrm{argmax}_{\lambda \in \Delta} \langle \lambda, r \rangle \\ &\qquad \text{s.t.} \quad E^T \lambda = \gamma P^T \lambda + (1-\gamma)\mu. \end{aligned}$$

Remarks: ◦ The problem in the current form suffers from $|\mathcal{S}|$ many constraints.

The Lagrangian: towards an unconstrained problem.

- The corresponding Lagrangian is:

$$\max_{\lambda\in\Delta}\min_{V}\ \langle\lambda, r\rangle + \langle V, \gamma P^T\lambda - E^T\lambda\rangle + (1-\gamma)\langle V, \mu\rangle.$$

- Applying proximal point we obtain the following update:

$$\lambda_k = \mathrm{argmax}_{\lambda\in\Delta} \underbrace{\min_{V}\ \langle\lambda, r\rangle + \langle V, \gamma P^T\lambda - E^T\lambda\rangle + (1-\gamma)\langle V, \mu\rangle}_{:=f(\lambda)} - \frac{1}{\eta} D_{KL}(\lambda, \lambda_{k-1}).$$

KKT conditions on the Lagrangian update.

Derivation:

- We notice by convexity of the Bregman divergence that the update is convex in λ.
- We introduce an auxiliary problem for any V as follows:

$$\lambda_k^V \;=\; \text{argmax}_{\lambda\in\Delta}\; \langle \lambda, r\rangle + \langle V, \gamma P^T\lambda - E^T\lambda\rangle + (1-\gamma)\langle V, \mu\rangle - \frac{1}{\eta} D_{KL}(\lambda, \lambda_{k-1}).$$

- By optimality conditions, it must hold

$$r + \gamma PV - EV - \frac{1}{\eta}\nabla_\lambda D_{KL}(\lambda_k^V, \lambda_{k-1}) = 0.$$

- Thus, λ_k^V can be computed in closed form for any V:

$$\lambda_k^V(s,a) = \frac{\lambda_{k-1}(s,a)e^{r(s,a)+\gamma(PV)(s,a)-(EV)(s,a)}}{\sum_{s,a}\lambda_{k-1}(s,a)e^{r(s,a)+\gamma(PV)(s,a)-(EV)(s,a)}}.$$

The unconstrained problem

◦ We can leverage the KKT conditions to write an unconstrained problem where the only decision variable is V:

$$\min_V \langle \lambda_k^V(s,a), r\rangle + \langle V, \gamma P^T \lambda_k^V(s,a) - E^T \lambda_k^V(s,a)\rangle + (1-\gamma)\langle V, \mu\rangle - \frac{1}{\eta} D_{KL}(\lambda_k^V(s,a), \lambda_{k-1}).$$

◦ With some calculus, we have the following compact form.

Unconstrained problem (REPS)

$$V_k = \min_V (1-\gamma)\langle \mu, V\rangle + \frac{1}{\eta}\log \sum_{s,a} \lambda_{k-1}(s,a) e^{r(s,a)+\gamma(PV)(s,a)-(EV)(s,a)}.$$

Remarks:

◦ The decision variable V has dimension $|\mathcal{S}|$.

◦ The objective is convex and smooth, its gradient is Lipschitz continuous.

The REPS algorithm [21]

Algorithm: REPS

Initialize λ_0, for exemple uniform.
for each iteration $k = 1, \ldots, K$ **do**
Solve the problem

$$V_k = \min_V (1-\gamma)\langle \mu, V \rangle + \frac{1}{\eta} \log \sum_{s,a} \lambda_{k-1}(s,a) e^{r(s,a) + \gamma (PV)(s,a) - (EV)(s,a)}$$

Update the occupancy measure:

$$\lambda_k(s,a) \propto \lambda_{k-1}(s,a) e^{r(s,a) + \gamma (PV_k)(s,a) - (EV_k)(s,a)}$$

end for

Sample complexity of REPS [20]

Algorithm	Oracle	Output
REPS	Exact gradient	$\mathcal{O}\left(\frac{\lvert\mathcal{S}\rvert^{3/2}}{(1-\gamma)^2\epsilon^2}\right)$
REPS	Stochastic Biased Gradients	$\mathcal{O}\left(\frac{\lvert\mathcal{S}\rvert^{3/2}}{(1-\gamma)^8\beta^2\epsilon^8}\right)$

Remarks:

- The exact gradient case achieves the best known sample complexity, e.g. comparable to NPG (see Lecture 6)
- The sample complexity with stochastic gradients degrades.
- For the stochastic gradient case, one needs to assume that $\lambda_k(s,a) \geq \beta > 0$. It solves the exploration problem by assumption.

References I

[1] Y. Abbasi-Yadkori, P. L. Bartlett, and A. Malek.
Linear programming for large-scale Markov decision problems.
In *International Conference on Machine Learning (ICML)*, 2014.

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

[3] D. P. Bertsekas and J. Tsitsiklis.
Neuro-Dynamic Programming.
Athena Scientific, 1996.

[4] V. S. Borkar.
A convex analytic approach to Markov decision processes.
Probability Theory and Related Fields, 78(4):583–602, 1988.

[5] Volkan Cevher and Bang Cong Vu.
A reflected forward-backward splitting method for monotone inclusions involving lipschitzian operators.
Set-Valued and Variational Analysis, pages 1–12, 2020.

[6] Y. Chen, L. Li, and M. Wang.
Scalable bilinear π learning using state and action features.
In *International Conference on Machine Learning (ICML)*, 2018.

References II

[7] D. P. De Farias and B. Van Roy.
The linear programming approach to approximate dynamic programming.
Operations Research, 51(6):850–865, 2003.

[8] D. P. De Farias and B. Van Roy.
On constraint sampling in the linear programming approach to approximate dynamic programming.
Mathematics of Operations Research, 29(3):462–478, 2004.

[9] Vijay V. Desai, Vivek F. Farias, and Ciamac C. Moallemi.
Approximate dynamic programming via a smoothed linear program.
Operations Research, 60(3):655–674, 2012.

[10] Alireza Fallah, Aryan Mokhtari, and Asuman E. Ozdaglar.
On the convergence theory of gradient-based model-agnostic meta-learning algorithms.
CoRR, abs/1908.10400, 2019.

[11] O. Hernández-Lerma and J. B. Lasserre.
Discrete-Time Markov Control Processes: Basic Optimality Criteria.
Springer-Verlag New York, 1996.

[12] O. Hernández-Lerma and J. B. Lasserre.
Further Topics on Discrete-Time Markov Control Processes.
Springer-Verlag New York, 1999.

References III

[13] Y. Jin and A. Sidford.
Efficiently solving MDPs with stochastic mirror descent.
In *International Conference on Machine Learning (ICML)*, 2020.

[14] G. M. Korpelevic.
An extragradient method for finding saddle-points and for other problems.
Èkonom. i Mat. Metody., 12(4):747–756, 1976.

[15] C. Lakshminarayanan, S. Bhatnagar, and C. Szepesvári.
A linearly relaxed approximate linear program for Markov decision processes.
IEEE Transactions on Automatic Control, 63(4):1185–1191, 2018.

[16] Yura Malitsky and Matthew K Tam.
A forward-backward splitting method for monotone inclusions without cocoercivity.
SIAM Journal on Optimization, 30(2):1451–1472, 2020.

[17] A. Manne.
Linear programming and sequential decisions.
Management Science, 6(3):259–267, 1960.

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis.
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533, 2015.

References IV

[19] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil.
A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach.
In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1497–1507. PMLR, 26–28 Aug 2020.

[20] Aldo Pacchiano, Jonathan Lee, Peter Bartlett, and Ofir Nachum.
Near optimal policy optimization via REPS.
In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[21] Jan Peters, Katharina Mulling, and Yasemin Altun.
Relative entropy policy search.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

[22] Marek Petrik, Gavin Taylor, Ron Parr, and Shlomo Zilberstein.
Feature selection using regularization in approximate linear programs for markov decision processes.
In *International Conference on International Conference on Machine Learning (ICML)*, 2010.

[23] Marek Petrik and Shlomo Zilberstein.
Constraint relaxation in approximate linear programs.
In *International Conference on Machine Learning (ICML)*, 2009.

[24] Leonid Denisovich Popov.
A modification of the arrow-hurwicz method for search of saddle points.
Mathematical notes of the Academy of Sciences of the USSR, 28(5):845–848, 1980.

References V

[25] M. L. Puterman.
Markov Decision Processes: Discrete Stochastic Dynamic Programming.
John Wiley & Sons, Inc., USA, 1st edition, 1994.

[26] Alexander Rakhlin and Karthik Sridharan.
Optimization, learning, and games with predictable sequences.
arXiv preprint arXiv:1311.1869, 2013.

[27] R Tyrrell Rockafellar.
Conjugate convex functions in optimal control and the calculus of variations.
Journal of Mathematical Analysis and Applications, 32(1):174–222, 1970.

[28] R. Tyrrell Rockafellar.
Monotone operators and the proximal point algorithm.
SIAM Journal on Control and Optimization, 14(5):877–898, 1976.

[29] Paul J Schweitzer and Abraham Seidmann.
Generalized polynomial approximations in markovian decision processes.
Journal of Mathematical Analysis and Applications, 110(2):568–582, 1985.

Supplementary

LP and optimization

Supplementary Material: Bellman Equation for State-action Visitation Distribution

Recall the definition

$$\lambda^\pi(s,a) := \sum_{t=0}^{\infty} \gamma^t P(s_t = s, a_t = a \,|\, \pi, s_0 \sim \mu).$$

Bellman Equation for λ^π

$$\lambda^\pi(s,a) = \mu(s)\pi(a|s) + \gamma \sum_{s',a'} \pi(a|s) P(s|s',a') \lambda^\pi(s',a').$$

Supplementary Material: Bellman Equation for State-action Visitation Distribution

Proof.

$$
\begin{aligned}
&\lambda^\pi(s,a)\\
=&P(s_0=s,a_0=a)+\sum_{t=1}^{\infty}\gamma^t P(s_t=s,a_t=a|\pi,s_0\sim\mu)\\
=&\mu(s)\pi(a|s)+\sum_{t=1}^{\infty}\gamma^t\sum_{s',a'}P(s_t=s,a_t=a|s_{t-1}=s',a_{t-1}=a',\pi,s_0\sim\mu)P(s_{t-1}=s',a_{t-1}=a'|\pi,s_0\sim\mu)\\
=&\mu(s)\pi(a|s)+\gamma\sum_{t=1}^{\infty}P(s_t=s,a_t=a|s_{t-1}=s',a_{t-1}=a')P(s_{t-1}=s',a_{t-1}=a'|\pi,s_0\sim\mu)\\
=&\mu(s)\pi(a|s)+\gamma\sum_{t=1}^{\infty}\pi(a|s)P(s|s',a')\sum_{t=1}^{\infty}\gamma^{t-1}P(s_{t-1}=s',a_{t-1}=a'|\pi,s_0\sim\mu)\\
=&\mu(s)\pi(a|s)+\gamma\sum_{s',a'}\pi(a|s)P(s|s',a')\lambda^\pi(s',a')
\end{aligned}
$$

where the third equality is due to Markov property. □

PPM guarantees for minimax optimization

Theorem (Convergence of PPM [28])

Suppose $(\mathbf{x}^k, \mathbf{y}^k)$ *be the iterates generated by PPM (i.e., (5)), then for the averaged iterates, it holds that*

$$\left| \Phi\left(\frac{1}{K}\sum_{k=1}^{K} \mathbf{x}^k, \frac{1}{K}\sum_{k=1}^{K} \mathbf{y}^k \right) - \Phi(\mathbf{x}^\star, \mathbf{y}^\star) \right| \le \frac{\|\mathbf{x}^0 - \mathbf{x}^\star\|^2 + \|\mathbf{y}^0 - \mathbf{y}^\star\|^2}{\tau K}.$$

Theorem (Linear convergence [28])

Suppose $(\mathbf{x}^k, \mathbf{y}^k)$ *be the iterates generated by (5),* $\Phi(\cdot, \cdot)$ *is* μ_x*−strongly convex in* $\mathbf{x}$ *and* μ_y*−strongly concave in* $\mathbf{y}$*. Let* $\mu = \max\{\mu_x, \mu_y\}$*. Then, for any* $\tau > 0$*,* $(\mathbf{x}^k, \mathbf{y}^k)$ *satisfies the following*

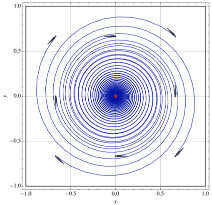
$$r^{k+1} \le \frac{1}{1 + \mu\tau} r^k,$$

where $r^k = \|\mathbf{x}^k - \mathbf{x}^\star\|^2 + \|\mathbf{y}^k - \mathbf{y}^\star\|^2$.

Remark:
- Still need an implementable and convergent algorithm beyond the stylized bilinear case.
- Note what happens when $\tau \to \infty$.

Extra-gradient algorithm (EG) [14]

EG method for saddle point problems
1. Choose $\mathbf{x}^0, \mathbf{y}^0$ and τ.
2. For $k = 0, 1, \cdots,$ perform:
$\tilde{\mathbf{x}}^k := \mathbf{x}^k - \tau \nabla_{\mathbf{x}} \Phi(\mathbf{x}^k, \mathbf{y}^k),$
$\tilde{\mathbf{y}}^k := \mathbf{y}^k + \tau \nabla_{\mathbf{y}} \Phi(\mathbf{x}^k, \mathbf{y}^k).$
$\mathbf{x}^{k+1} := \mathbf{x}^k - \tau \nabla_{\mathbf{x}} \Phi(\tilde{\mathbf{x}}^k, \tilde{\mathbf{y}}^k).$
$\mathbf{y}^{k+1} := \mathbf{y}^k + \tau \nabla_{\mathbf{y}} \Phi(\tilde{\mathbf{x}}^k, \tilde{\mathbf{y}}^k).$



◦ Idea: Predict the gradient at the next point

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \tau \mathbf{V}\big(\underbrace{\mathbf{z}^k - \tau \mathbf{V}(\mathbf{z}^k)}_{\text{prediction of } \mathbf{z}^{k+1}} \big) \tag{EG}$$

Remark: ◦ 1-extra-gradient computation per iteration

Extra-gradient algorithm: Convergence

Theorem (General case [10])

Let $0 < \tau \leq \frac{1}{L}$. It holds that

- *Iterates $(\mathbf{x}^k, \mathbf{y}^k)$ remains bounded in a convex compact set.*
- *Primal-dual gap reduces:* $\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{x}^k, \frac{1}{K}\sum_{k=1}^{K}\mathbf{y}^k\right) \leq \mathcal{O}\left(\frac{1}{K}\right)$.

Theorem (Linear convergence [19])

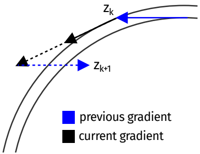
Suppose $(\mathbf{x}^k, \mathbf{y}^k)$ be the iterates generated by Extra-gradient algorithm, $\Phi(\cdot,\cdot)$ is μ_x−strongly convex in $\mathbf{x}$ and μ_y−strongly concave in $\mathbf{y}$. Let $\mu = \max\{\mu_x, \mu_y\}$. Then, for $\tau = \frac{1}{4L}$, $(\mathbf{x}^k, \mathbf{y}^k)$ satisfies,

$$r^{k+1} \leq \left(1 - \frac{1}{c\kappa}\right)^k r^0,$$

where $r^k = \|\mathbf{x}^k - \mathbf{x}^\star\|^2 + \|\mathbf{y}^k - \mathbf{y}^\star\|^2$, $\kappa = \frac{L}{\mu}$ is the condition number of the problem, and c is a constant which is independent of the problem parameters.

Optimistic gradient descent ascent algorithm (OGDA) [26]

OGDA for saddle point problems
1. Choose $\mathbf{x}^0, \mathbf{y}^0, \mathbf{x}^1, \mathbf{y}^1$ and τ.
2. For $k = 1, \cdots,$ perform:
$\mathbf{x}^{k+1} := \mathbf{x}^k - 2\tau\nabla_{\mathbf{x}}\Phi(\mathbf{x}^k, \mathbf{y}^k) + \tau\nabla_{\mathbf{x}}\Phi(\mathbf{x}^{k-1}, \mathbf{y}^{k-1})$.
$\mathbf{y}^{k+1} := \mathbf{y}^k + 2\tau\nabla_{\mathbf{y}}\Phi(\mathbf{x}^k, \mathbf{y}^k) - \tau\nabla_{\mathbf{y}}\Phi(\mathbf{x}^{k-1}, \mathbf{y}^{k-1})$.



◦ Main difference from the GDA: Add a "momentum" or "reflection" term to the updates

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \tau\left[\mathbf{V}(\mathbf{z}^k) + \underbrace{(\mathbf{V}(\mathbf{z}^k) - \mathbf{V}(\mathbf{z}^{k-1}))}_{\text{momentum}}\right]. \qquad \text{(OGDA)}$$

◦ Known as Popov's method [24], it is also a special case of the Forward-Reflected-Backward method [16].

◦ It has ties to the Reflected-Forward-Backward Splitting (RFBS) method [5]:

$$\mathbf{z}^{k+1} = \mathbf{z}^k - \tau\mathbf{V}(2\mathbf{z}^k - \mathbf{z}^{k-1}). \qquad \text{(RFBS)}$$

Remark: ◦ Advanced material at the end: OGDA is an approximation of PPM for bilinear problems.

OGDA: Convergence

Theorem (General case [10])

Let $0 < \tau \leq \frac{1}{2L}$, $\mathbf{x}^1 = \mathbf{x}^0, \mathbf{y}^1 = y^0$. *It holds that*

- *Iterates* $(\mathbf{x}^k, \mathbf{y}^k)$ *remains bounded in a convex compact set.*
- *Primal-dual gap reduces:* $\mathrm{Gap}\left(\frac{1}{K}\sum_{k=1}^K \mathbf{x}^k, \frac{1}{K}\sum_{k=1}^K \mathbf{y}^k\right) \leq \mathcal{O}\left(\frac{1}{K}\right)$.

Theorem (Linear convergence [19])

Suppose $(\mathbf{x}^k, \mathbf{y}^k)$ *be the iterates generated by OGDA,* $\Phi(\cdot,\cdot)$ *is* μ_x-*strongly convex in* $\mathbf{x}$ *and* μ_y-*strongly concave in* $\mathbf{y}$. *Let* $\mu = \max\{\mu_x, \mu_y\}$. *Then, for* $\tau = \frac{1}{4L}$, $(\mathbf{x}^k, \mathbf{y}^k)$ *satisfies,*

$$r^{k+1} \leq \left(1 - \frac{1}{c\kappa}\right)^k r^0,$$

where $r^k = \|\mathbf{x}^k - \mathbf{x}^\star\|^2 + \|\mathbf{y}^k - \mathbf{y}^\star\|^2$, $\kappa = \frac{L}{\mu}$ *is the condition number of the problem, and* c *is a constant which is independent of the problem parameters.*

⋆Bregman divergences

Table: **Bregman functions** $\psi(\mathbf{x})$ & corresponding Bregman divergences/distances $d_\psi(\mathbf{x}, \mathbf{y})^a$.

Name (or Loss)	Domain[b]	$\psi(\mathbf{x})$	$d_\psi(\mathbf{x}, \mathbf{y})$
Squared loss	$\mathbb{R}$	x^2	$(x-y)^2$
Itakura-Saito divergence	$\mathbb{R}_{++}$	$-\log x$	$\frac{x}{y} - \log\left(\frac{x}{y}\right) - 1$
Squared Euclidean distance	$\mathbb{R}^p$	$\\|\mathbf{x}\\|_2^2$	$\\|\mathbf{x}-\mathbf{y}\\|_2^2$
Squared Mahalanobis distance	$\mathbb{R}^p$	$\langle \mathbf{x}, \mathbf{A}\mathbf{x}\rangle$	$\langle (\mathbf{x}-\mathbf{y}), \mathbf{A}(\mathbf{x}-\mathbf{y})\rangle^c$
Entropy distance	p-simplex[d]	$\sum_i x_i \log x_i$	$\sum_i x_i \log\left(\frac{x_i}{y_i}\right)$
Generalized I-divergence	$\mathbb{R}_+^p$	$\sum_i x_i \log x_i$	$\sum_i \left(\log\left(\frac{x_i}{y_i}\right) - (x_i - y_i)\right)$
von Neumann divergence	$\mathbb{S}_+^{p\times p}$	$\mathbf{X}\log\mathbf{X} - \mathbf{X}$	$\mathrm{tr}\left(\mathbf{X}(\log\mathbf{X} - \log\mathbf{Y}) - \mathbf{X} + \mathbf{Y}\right)^e$
logdet divergence	$\mathbb{S}_+^{p\times p}$	$-\log\det\mathbf{X}$	$\mathrm{tr}\left(\mathbf{X}\mathbf{Y}^{-1}\right) - \log\det\left(\mathbf{X}\mathbf{Y}^{-1}\right) - p$

[a] $x, y \in \mathbb{R}$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ and $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{p\times p}$.
[b] $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote non-negative and positive real numbers respectively.
[c] $\mathbf{A} \in \mathbb{S}_+^{p\times p}$, the set of symmetric positive semidefinite matrix.
[d] p-simplex:= $\{\mathbf{x} \in \mathbb{R}^p : \sum_{i=1}^p x_i = 1, x_i \geq 0, i = 1, \ldots, p\}$
[e] $\mathrm{tr}(\mathbf{A})$ is the trace of $\mathbf{A}$.

*Mirror descent [2]

What happens if we use a Bregman distance d_ψ in gradient descent?

Let $\psi : \mathbb{R}^p \to \mathbb{R}$ be a μ-strongly convex and continuously differentiable function and let the associated Bregman distance be $d_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla\psi(\mathbf{y})\rangle$.
Assume that the inverse mapping $\psi^\star$ of ψ is easily computable (i.e., its convex conjugate).

- **Majorize**: Find α_k such that

$$f(\mathbf{x}) \leq f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k\rangle + \frac{1}{\alpha_k} d_\psi(\mathbf{x}, \mathbf{x}^k) := Q^k_\psi(\mathbf{x}, \mathbf{x}^k)$$

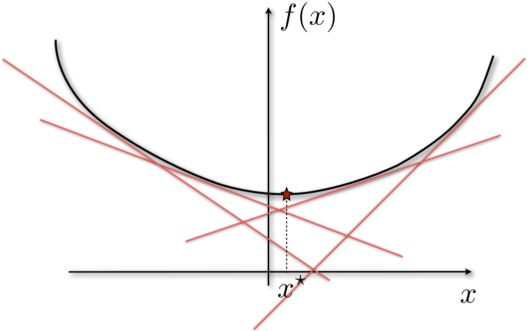
- **Minimize**

$$\begin{aligned}
\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} Q^k_\psi(\mathbf{x}, \mathbf{x}^k) \Rightarrow \nabla f(\mathbf{x}^k) + \frac{1}{\alpha_k}\left(\nabla\psi(\mathbf{x}^{k+1}) - \nabla\psi(\mathbf{x}^k)\right) = 0 \\
\nabla\psi(\mathbf{x}^{k+1}) &= \nabla\psi(\mathbf{x}^k) - \alpha_k \nabla f(\mathbf{x}^k) \\
\mathbf{x}^{k+1} &= \nabla\psi^*(\nabla\psi(\mathbf{x}^k) - \alpha_k \nabla f(\mathbf{x}^k)) \qquad (\nabla\psi(\cdot))^{-1} = \nabla\psi^*(\cdot)[27].
\end{aligned}$$

- Mirror descent is a **generalization** of gradient descent for functions that are Lipschitz-gradient in norms other than the Euclidean.
- MD allows to deal with some **constraints** via a proper choice of ψ.

*What to keep in mind about mirror descent?

- Approximates the optimum by lower bounding the function via hyperplanes at $\mathbf{x}_t$



- The smaller the gradients, the better the approximation!

*Mirror descent example

How can we minimize a convex function over the unit simplex?

$$\min_{\mathbf{x}\in\Delta} f(\mathbf{x}),$$

where

- $\Delta := \{\mathbf{x} \in \mathbb{R}^p \ : \ \sum_{j=1}^p x_j = 1, \mathbf{x} \geq 0\}$ is the **unit simplex**;
- f is convex L_f-Lipschitz continuous with respect to some norm $\|\cdot\|$. (not necessarily *L-Lipschitz gradient*)

Entropy function

- Define the entropy function

$$\psi_e(\mathbf{x}) = \sum_{j=1}^{p} x_j \ln x_j \quad \text{if } \mathbf{x} \in \Delta, \quad +\infty \text{ otherwise.}$$

- ψ_e is 1-strongly convex over $\mathrm{int}\Delta$ with respect to $\|\cdot\|_1$.
- $\psi_e^\star(\mathbf{z}) = \ln \sum_{j=1}^p e^{z_j}$ and $\|\nabla\psi_e(\mathbf{x})\| \to \infty$ as $\mathbf{x} \to \tilde{\mathbf{x}} \in \Delta$.
- Let $\mathbf{x}^0 = p^{-1}\mathbf{1}$, then $d_\psi(\mathbf{x}, \mathbf{x}^0) \leq \ln p$ for all $\mathbf{x} \in \Delta$.

⋆Entropic descent algorithm [2]

Entropic descent algorithm (EDA)

Let $\mathbf{x}^0 = p^{-1}\mathbf{1}$ and generate the following sequence

$$x_j^{k+1} = \frac{x_j^k e^{-t_k f_j'(\mathbf{x}^k)}}{\sum_{j=1}^p x_j^k e^{-t_k f_j'(\mathbf{x}^k)}}, \quad t_k = \frac{\sqrt{2 \ln p}}{L_f} \frac{1}{\sqrt{k}},$$

where $f'(\mathbf{x}) = (f_1(\mathbf{x})', \ldots, f_p(\mathbf{x})')^T \in \partial f(\mathbf{x})$, which is the **subdifferential** of f at $\mathbf{x}$.

- This is an example of **non-smooth** and **constrained** optimization;
- The updates are multiplicative.

⋆Convergence of mirror descent

Problem

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{6}$$

where

- $\mathcal{X}$ *is a closed convex subset of* $\mathbb{R}^p$;
- f *is convex* L_f*-Lipschitz continuous with respect to some norm* $\|\cdot\|$.

Theorem ([2])

Let $\{\mathbf{x}^k\}$ *be the sequence generated by mirror descent with* $\mathbf{x}^0 \in \mathrm{int}\mathcal{X}$.
If the step-sizes are chosen as

$$\alpha_k = \frac{\sqrt{2\mu d_\psi(\mathbf{x}^\star, \mathbf{x}^0)}}{L_f}\frac{1}{\sqrt{k}}$$

the following convergence rate holds

$$\min_{0\leq s\leq k} f(\mathbf{x}^s) - f^\star \leq L_f\sqrt{\frac{2d_\psi(\mathbf{x}^\star, \mathbf{x}^0)}{\mu}}\frac{1}{\sqrt{k}}$$

- *This convergence rate is* **optimal** *for solving* (6) *with a first-order method.*