lions@epfl

EE-568: REINFORCEMENT LEARNING
LABORATORY FOR INFORMATION AND INFERENCE SYSTEMS
SPRING 2024

EPFL

INSTRUCTOR: VOLKAN CEVHER

SCRIBE: SEPIDEH MAMOOLER

LINEAR PROGRAMMING

Index terms: Linear Programming, Primal LP, Dual LP, Lagrangian, Saddle Point, Proximal Point Method, Bregman Distance, REPS

1 Introduction to Linear Programming

In the last lecture we talked about dynamic programming approaches in reinforcement learning (RL), like value and policy iteration. Having a policy we evaluate it using the value function corresponding to that policy. Given the value we can take greedy actions to improve the policy. We also talked about model-based and model-free RL algorithms. In model-based you know the probability of which state you land on given the current state and an action. In model-free algorithms this is not the case.

In this lecture, we look at RL from the linear programming perspective. First, let's recall the RL setup:

- Goal: make sequential decision in an unknown environment.
- Markov decision process $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \mu, \gamma)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ denotes the transition probabilities, r is the reward function, μ is the initial state distribution, and γ is the discount factor.
- Stationary stochastic policy $\pi : \mathcal{S} \to \Delta(\mathcal{A}), a_t \sim \pi(.|s_t)$.
- 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: $\max_\pi (1-\gamma) \sum_{s\in\mathcal{S}} \mu(s) V^\pi(s)$.

1.1 Bellman Optimality Condition

Remember that the optimal policy maximizes the value function i.e. $V^* = \max_{\pi\in\Pi} V^\pi(s)$, and V^* satisfies the Bellman optimality equation which can be written as a feasibility problem:

$$\begin{aligned} &\min_V 0 \\ &\text{s.t} \quad V(s) = (\mathcal{T}V(s)) := \max_{a\in\mathcal{A}} \left[r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathbb{P}(s'|s,a)\, V(s') \right], \quad \forall s \in \mathcal{S}, \end{aligned} \tag{1}$$

where $\mathcal{T}$ is the Bellman operator.

Bellman operator has two important properties:

- It is a γ-contraction mapping w.r.t. l_∞-norm:

$$\|\mathcal{T}V' - \mathcal{T}V\|_\infty \le \gamma \|V' - V\|_\infty.$$

- It is monotonic (component-wise):

$$V' \le V \Rightarrow \mathcal{T}V' \le \mathcal{T}V.$$

The Bellman equality constraint (Equation 1) in **non-linear** with respect to V due to the maximisation over $\mathcal{A}$. Now we will turn it into a **linear inequality**.

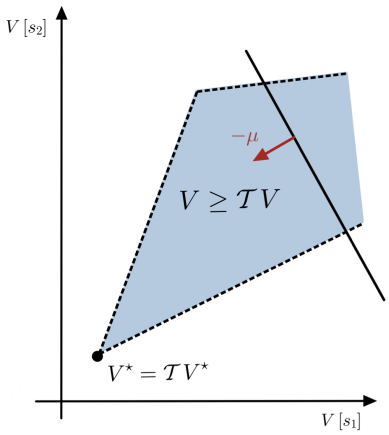

Figure 1: Graphical interpretation of Bellman inequality

1.2 Relaxation of Bellman Optimality Condition

To make the Bellman equality linear we remove the maximization and turn the equality into an inequality that holds for all actions $a \in \mathcal{A}$:

$$V(s) \geq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a)\, V(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}, \tag{2}$$

which is the same as $V(s) \geq \mathcal{T}V(s), \forall s \in \mathcal{S}, a \in \mathcal{A}$. Given the monotonicity property of the Bellman operator, by successively applying the this operator we have:

$$V \geq \mathcal{T}V \geq \mathcal{T}^2 V \geq \ldots \geq \mathcal{T}^\infty V = V^*.$$

As a result, V^* is the value function with the lowest value that satisfies the Bellman inequality (Equation 2). Figure 1 shows a graphical interpretation of the Bellman inequality. The Blue region illustrates the feasibility region i.e. the space of all value functions satisfying the Bellman inequality. V^* is the function with the **lowest value** in this space. This interpretation leads us to the primal formulation of linear programming which we discuss next.

2 Linear Programming: The Primal and Dual Formulation

2.1 Primal LP Formulation

Definition 1 (Primal LP)**.** *Let $\mu(s) > 0, \forall s \in \mathcal{S}$ be the initial distribution. Then, the primal LP is given by:*

$$\begin{aligned} \min_V \quad & (1-\gamma) \sum_{s \in \mathcal{S}} \mu(s) V(s) \\ s.t \quad & V(s) \geq r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a)\, V(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}. \end{aligned} \tag{3}$$

Lemma 2.1 (LP Formulation and V^*)**.** *V^* is the unique optimal solution to the above LP formulation for **any positive** weights $\{\mu(s)\}$.*

Proof. Based on the primal LP (Equation 3), and the Bellman optimality operator (Euqation 1), V^* is in the feasibility region of the Primal LP because:

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

Due to component-wise monotonicity, for any feasible V, we have $V(s) \geq \mathcal{T}V(s), \forall s \in \mathcal{S}, a \in \mathcal{A}$, and as a result $V \geq \mathcal{T}V$. Thus,

$$V \geq \mathcal{T}V \geq \mathcal{T}^2V \geq ... \geq \mathcal{T}^\infty V = V^*,$$

which implies the optimality of V^*.
The uniqueness of V^* is followed by the contractivess of $\mathcal{T}$.

□

Remark.

- The number of decision variables of the primal is $|\mathcal{S}|$, and its number of constraints is $|\mathcal{S}||\mathcal{A}|$.
- Given V^*, we can determine an optimal (deterministic) policy greedily as follows:

$$\pi^* \in \arg\max_{a \in \mathcal{A}} \left[r(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbb{P}(s'|s,a)\, V^*(s) \right]. \tag{4}$$

- The factor $(1-\gamma)$ in the objective will ensure that the dual LP variables, which we will see next, are in the simplex.

2.2 Dual Formulation

Definition 2 (Dual LP)**.** *From linear programming, we know that the dual LP of primal(Equation 3) is given by the following.*

$$\begin{aligned} \max_{\lambda} \quad & \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} r(s,a)\lambda(s,a) \\ s.t \quad & \sum_{a \in \mathcal{A}} \lambda(s,a) = (1-\gamma)\mu(s) + \gamma \sum_{s' \in \mathcal{S}, a' \in \mathcal{A}} \mathbb{P}(s|s',a')\, \lambda(s',a'), \quad \forall s \in \mathcal{S}, \\ & \lambda(s,a) \geq 0, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}. \end{aligned} \tag{5}$$

Remark.

- The number of decision variables is given by $|\mathcal{S}||\mathcal{A}|$. Note that this is bigger than the decision variables of the primal which is $|\mathcal{S}|$.
- The number of constraints is given by $|\mathcal{S}| + |\mathcal{S}||\mathcal{A}|$.
- The constraints imply the decision variables are probabilities: $\lambda \in \Delta([|\mathcal{S}||\mathcal{A}|])$.
- The solution to the dual LP λ^* corresponds to the state-action occupancy of π^* which we define below.

Definition 3 (Occupancy Measure)**.** *The occupancy measure for an initial distribution μ and a policy π is defined as follows:*

$$\lambda_\mu^\pi(s,a) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s, a_t = a | s_0 \sim \mu, \pi), \tag{6}$$

where $\mathbb{P}(.|s_0 \sim \mu, \pi)$ denotes the probability of an event when following policy π starting from $s_0 \sim \mu$.

Based on the definition of the expected value the occupancy measure can be compute by:

$$\lambda_\mu^\pi(s,a) = (1-\gamma)\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t = s, a_t = a | s_0 \sim \mu, \pi) \right], \tag{7}$$

which indicates that for a given policy π and initial distribution μ, the occupancy measure at state-action pair (s,a) is the normalized discounted visitation frequency of the pair (s,a) when π is played. Intuitively, the dual objective tries to find an occupancy of state-action pairs which leads to a high reward.

Figure 2 illustrates the occupancy measure for two different policies for different values of γ. As shown in the first row of this figure, a higher discount factor gives a better view on where the policy is leading us. On the other hand for

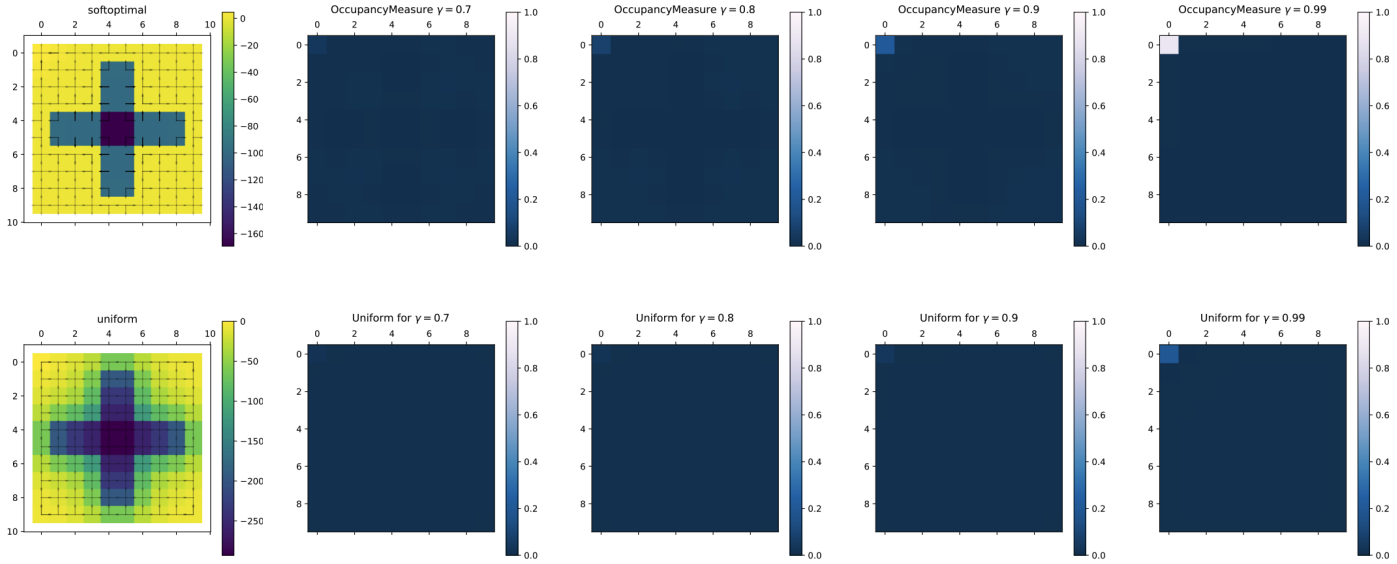

Figure 2: Visualization of policy (leftmost column) and the occupancy measure for different values of γ. The arrows on the leftmost subplots indicate the actions dictated by the policy in each state.

a random policy which does not lead anywhere specific (the second row), the difference between a higher and lower discount factors is less evident.

Using the definition of occupancy measure, we can get from the primal objective, $(1-\gamma)\mathbb{E}_{s\sim\mu}[V^\pi(s)]$, to the dual objective, $\sum_{s\in\mathcal{S},a\in\mathcal{A}}\lambda^\pi(s,a)r(s,a)$, as shown below:

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

Example 1. To intuitively understand the dual objective, consider the following example: you like the food served in cafeteria $\mathcal{X}$, and eating their food makes you happy! If your goal is to be happy (reward), you go there often and your occupancy measure has higher values around that cafeteria. At the same time, you have a budget which does not allow eating cafeteria $\mathcal{X}$'s food every day (environment's constraints). This will add some constraints to your occupancy measure. What the dual LP does is that it maximizes the occupancy measure around the areas with higher reward while satisfying the constraints enforced by the environment.

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

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

In other words, to get to the policy from the occupancy measure, for each state s we normalize the occupancy measures for s to obtain a probability distribution over all actions $a\in\mathcal{A}$. When $\sum_{a\in\mathcal{A}}\lambda(s,a)=0$, we set $\pi_\lambda(,|s)$ arbitrarily. Now given this policy π_λ, if we compute the occupancy measure we obtain λ, meaning $\lambda^{\pi_\lambda}=\lambda$.

2.3 Finding the Optimal Policy

- Primal LP approach: Solve primal LP to obtain for the optimal value function V^*, then construct an (deterministic) optimal policy greedily:

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

- Dual LP approach: Solve the dual LP to obtain an optimal state-action occupancy λ^*, then construct a (randomized) optimal policy by:

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

2.4 Occupancy measure and the Value Function

The relation between the occupancy measure and the value function is given by $(1-\gamma)V^\pi(\mu) = \left\langle \lambda^\pi_\mu, r \right\rangle$.

Derivation:

$$\begin{aligned}
V^\pi(\mu) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) | s_0 \sim \mu, \pi \right] \\
&= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \sum_{s,a} r(s,a)\mathbb{1}(s_t=s, a_t=a) | s_0 \sim \mu, \pi \right] \\
&= \sum_{s,a} r(s,a)\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}(s_t=s, a_t=a) | s_0 \sim \mu, \pi \right] \quad \text{(Linearity of expectation)} \\
&= \sum_{s,a} r(s,a) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t=s, a_t=a | s_0 \sim \mu, \pi) \quad \text{(Dominated convergence theorem)} \\
&= \frac{\sum_{s,a} r(s,a)\lambda^\pi_\mu(s,a)}{1-\gamma} \\
&= \frac{\left\langle \lambda^\pi_\mu, r \right\rangle}{1-\gamma} \\
&\Rightarrow (1-\gamma)V^\pi(\mu) = \langle \lambda_m u^\pi, r \rangle .
\end{aligned}$$

2.5 A More Compact Notation of the Primal and Dual

To obtain a more compact notation of the primal and dual LP we introduce the following notations:

- We write the transitions as a matrix P with dimension $(|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|)$, where each entry in row $(s,a) \in \mathcal{S}\times\mathcal{A}$ and column $s' \in \mathcal{S}$ is the probability of getting to state s' from state s with action a: $\mathbb{P}(s'|s,a)$. P is a block matrix with the $(|\mathcal{S}| \times |\mathcal{S}|)$-matrix P_{a_i} stacked $|\mathcal{A}|$ times:

$$P = \begin{bmatrix} P_{a_1} \\ \vdots \\ P_{a_{|\mathcal{A}|}} \end{bmatrix},$$

where

$$P_{a_i} = \begin{bmatrix} \mathbb{P}(s_1|s_1,a_i) & \dots & \mathbb{P}(s_{|\mathcal{S}|}|s_1,a_i) \\ \vdots & & \vdots \\ \mathbb{P}(s_1|s_{|\mathcal{S}|},a_i) & \dots & \mathbb{P}(s_{|\mathcal{S}|}|s_{|\mathcal{S}|},a_i) \end{bmatrix}.$$

- We consider E a binary matrix of dimension $(|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|)$ defined by $E_{(s,a),s'} = \begin{cases} 1 & \text{if } s = s' \\ 0 & \text{else} \end{cases}$.

E is a block matrix with the $|\mathcal{S}| \times |\mathcal{S}|$ identity matrix stacked $|\mathcal{A}|$ times:

$$E = \begin{bmatrix} I_{|\mathcal{S}|} \\ \vdots \\ I_{|\mathcal{S}|} \end{bmatrix}.$$

- We use $r, \lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ to represent matrices with entries $r(s,a)$ and $\lambda(s,a)$ at index $(s,a) \in \mathcal{S} \times \mathcal{A}$, respectively.
- We use $\mu, V \in \mathbb{R}^{|\mathcal{S}|}$ to represent vectors with entries $\mu(s)$, and $V(s)$ at index $s \in \mathcal{S}$, respectively.

Given the above notation we can write $(EV)(s,a) = V(s)$, and $(PV)(s,a) = \sum_{s'} \mathbb{P}(s'|s,a)\, V(s')$, and define the primal and the dual as follows:

Primal LP:

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

Dual LP:

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} \quad & \langle \lambda, r \rangle \\ \text{s.t.} \quad & E^T \lambda = (1-\gamma)\mu + \gamma P^T \lambda, \quad \lambda \geq 0. \end{aligned}$$

Table 1 summarizes the important properties of the primal and dual LP.

Primal	**Dual**
Over value functions.	Over occupancy measures.
$\lvert\mathcal{S}\rvert$ decision variables.	$\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert$ decision variables.
$\lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert$ constraints.	$\lvert\mathcal{S}\rvert + \lvert\mathcal{S}\rvert\lvert\mathcal{A}\rvert$ constraints.
For all V that is a primal feasible, $V^* \leq V$.	For all policy π, the induced λ^π is dual feasible.
Optimal value function V^* is the optimizer.	For all λ that is dual feasible π^λ has occupancy measure λ.
Optimal policy is the associated greedy policy.	Optimal policy is the associated random policy π^{λ^*}.

Table 1: Overview of important properties of the primal and dual LP.

2.6 Dynamic Programming vs Linear Programming

In dynamic programming we use Value Iteration and Policy Iteration to find optimal value function and policy respectively using the Bellman Operator. In linear programming we solve the primal or dual LP using LP solvers like the Simplex and Interior Point methods, and find the optimal value function and occupancy measure respectively. Table 2 compares these two approaches together.

Dynamic Programming	**Linear Programming**
Solved via simple iterative updates.	Solved using rich library of fast LP solvers.
Polynomial complexity in $\lvert\mathcal{S}\rvert$ and $\lvert\mathcal{A}\rvert$ and $(1-\gamma)^{-1}$.	Polynomial complexity in $\lvert\mathcal{S}\rvert$ and $\lvert\mathcal{A}\rvert$ but not in $(1-\gamma)^{-1}$.
Works better for short horizon problems.	Works better for long horizon problems.

Table 2: Comparison of dynamic and linear programming.

Overall LP has the advantages of being an optimization problem for which many algorithms exists. However, it has a large number of variables which is not desirable.

Next, we will see how the primal and the dual problems can be combined into a single problem.

3 The Lagrangian

3.1 Min-max Optimization

Consider the following constrained problem:

$$\begin{aligned} &\min f(\mathbf{x}) \\ \text{s.t.} \quad &\mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned}$$

We write it as if it did not have a constraints by adding a term to the objective which is ∞ when constraints are violated and zero otherwise:

$$\min_x \{f(\mathbf{x}) + \max_{\mathbf{y}} \langle \mathbf{y}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle\}.$$

As $f(x)$ does not depend on y, we can take the *max* out and obtain a ***min-max*** problem:

$$\min_{\mathbf{x}} \max_{\mathbf{y}} \{f(\mathbf{x}) + \langle \mathbf{y}, \mathbf{A}\mathbf{x} - \mathbf{b} \rangle\}.$$

We can apply this idea to the primal and dual, write the objective of the primal, and put the constraint in the objective as an extra term using λ as the second variable. This is called the *Saddle point* formulation:

$$\min_V \max_{\lambda \geq 0} (1-\gamma) \langle \mu, V \rangle + \langle \lambda, r + \gamma PV - EV \rangle .$$

Now we formalize the concepts described above.

Definition 4 (Bilinear min-max template)**.**

$$\begin{aligned} &\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} \quad f(\mathbf{x}) + \langle \mathbf{A}\mathbf{x}, \mathbf{y} \rangle - h(\mathbf{y}), \\ &\textit{where} \quad \mathcal{X} \subseteq \mathcal{R}^p, \quad \textit{and} \quad \mathcal{Y} \subseteq \mathbb{R}^n, \\ &f : \mathcal{X} \to \mathbb{R}, \quad \textit{and} \quad h : \mathcal{Y} \to \mathbb{R} \quad \textit{are convex.} \end{aligned} \tag{8}$$

Definition 5 (Convex-Concave min-max template)**.** *In a more general form, the above formulation can be written in the following compact form called the* ***Convex-Concave min-max template****:*

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} \quad \phi(\mathbf{x}, \mathbf{y}), \tag{9}$$

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

Given the min-max problem above, we define the gradient vector as $V(\mathbf{z}) = \left[\nabla_{\mathbf{x}} \phi(\mathbf{x}, \mathbf{y}) - \nabla_{\mathbf{y}} \phi(\mathbf{x}, \mathbf{y})\right]$ where $\mathbf{z} = [\mathbf{x}, \mathbf{y}]$.

Now let's consider a simple bilinear problem:

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}\mathbf{y}.$$

The solution to this problem is the point of origin (see Figure 3). The ***gradient descent ascent*** cannot reach this solution because at each point $\mathbf{z}^k$, the gradient $V(\mathbf{z}^k)$ follows a direction perpendicular to the direction of the vector from the point of origin to $\mathbf{z}^k$. As a result following the gradient we move outwards and never reach the point of origin. But what if instead of departing from $\mathbf{z}^k$ at the direction of the gradient at $\mathbf{z}^k$ and landing on $\mathbf{z}^{k+1}$, we land on $\mathbf{z}^{k+1}$ from point $\mathbf{z}^k$ following the gradient of $\mathbf{z}^{k+1}$?

This is the idea behind the ***Proximal Point Method (PPM)*** [Rockafellar, 1976]. It is true that we don't know what the gradient of point $\mathbf{z}^{k+1}$ is before landing on it. As a result, we approximate it using the different between point $\mathbf{z}^k$ and its gradient $V(\mathbf{z}^k)$. This is called the ***Extra-gradient (EG)*** method [Korpelevich, 1976]. We summarize these three methods in Table 3 and illustrate their trajectories in Figure 3.

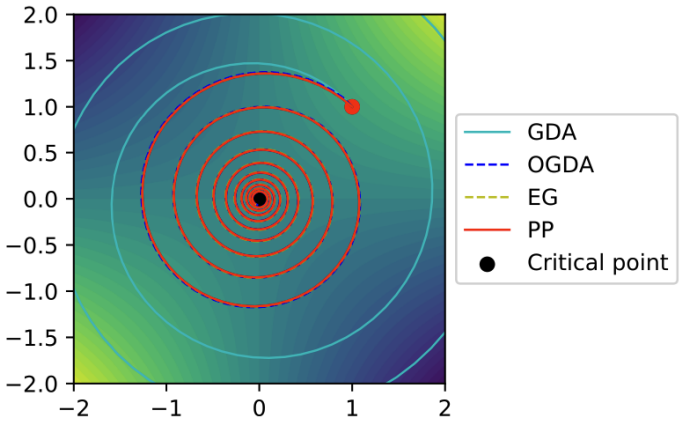

Figure 3: Trajectory of different algorithms for a simple bilinear game: $\min_{\mathbf{x}} \max_{\mathbf{y}} \mathbf{xy}$.

Algorithms	**Update formula**
Gradient Descent Ascent (GDA)	$\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(\mathbf{z}^k)$
Proximal point method (PPM)	$\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(\mathbf{z}^{k+1})$
Extra-gradient (EG)	$\mathbf{z}^{k+1} = \mathbf{z}^k - \eta V(\mathbf{z}^k - \alpha V(\mathbf{z}^k))$

Table 3: Basic algorithms for min-max problem

3.2 Proximal Point Method

The proximal point method (PPM) minimizes the objective + a quadratic minimizer anchored at $\mathbf{x}^k$. This is not necessarily easier than the original objective. But the advantage is that if the original problem is **smooth** and **convex**, adding the regularize makes it **strongly convex**. This improves the convergence rate.

Definition 6 (Proximal Point Method)**.** *Consider the smooth unconstrained optimization problem* $\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x})$*. For a step-size* $\tau > 0$*, Proximal Point Method (PPM) can be written as follows*

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} \{f(\mathbf{x}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{x}^k\|^2\} := \text{prox}_{\tau f}(\mathbf{x}^k). \tag{10}$$

Using the optimality condition (taking the gradient of the objective and setting it to zero) we can find the closed form solution:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau\nabla f(\mathbf{x}^{k+1}). \tag{11}$$

As mentioned before, the downside is that we cannot get the gradient of the next point at the current point.

Remark.

- PPM is an implicit, non-practical algorithm since we need the point $x^k + 1$ for its update.
- Each step of PPM can be as hard as solving the original problem.

By applying the PPM to the min-max problem we get:

$$\min_{\mathbf{x}\in\mathbb{R}^d} \max_{\mathbf{y}\in\mathbb{R}^n} \phi(\mathbf{x}, \mathbf{y}) + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{x}^k\|^2 - \frac{1}{2\tau}\|\mathbf{y} - \mathbf{y}^k\|^2, \tag{12}$$

which by the optimality condition gives us:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau\nabla_{\mathbf{x}}\phi(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}), \quad \mathbf{y}^{k+1} = \mathbf{y}^k + \tau\nabla_{\mathbf{y}}\phi(\mathbf{x}^{k+1}, \mathbf{y}^{k+1}).$$

In the vector notation this can be written as:

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

3.3 Bregman Setup

Given a convex function ω, and two points $\mathbf{z}$ and $\mathbf{z}'$, the linearized version of $\omega(\mathbf{z})$ can be written using the gradient of ω at point $\mathbf{z}'$ as follows:

$$\omega(\mathbf{z}) = \omega(\mathbf{z}') + \nabla\omega(\mathbf{z}')^T(\mathbf{z} - \mathbf{z}').$$

The distance between the actual value of $\omega(\mathbf{z})$ and its linearized version is known as the ***Bregman distance***.

Definition 7 (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 $\|.\|$ on the underlying space and is continuously differentiable. The Bregman distance induced by $\omega(.)$ is given by:*

$$D_\omega(\mathbf{z}, \mathbf{z}') = \omega(\mathbf{z}) - \omega(\mathbf{z}') - \nabla\omega(\mathbf{z}')^T(\mathbf{z} - \mathbf{z}'). \tag{13}$$

Now instead of the quadratic function we can use this distance to regularize the objective function in the PPM:

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

The Bregman distance is visualized in Figure 4.

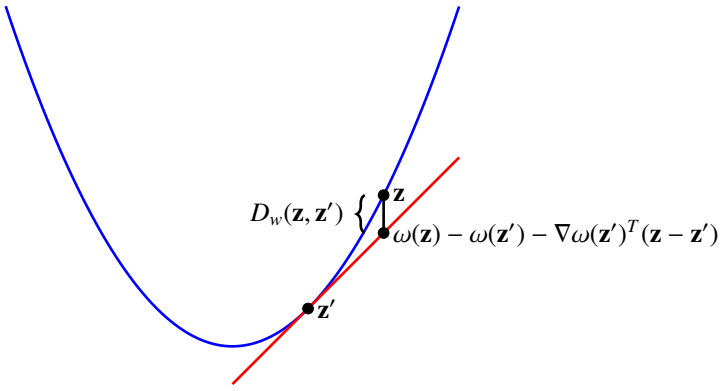

Figure 4: Visualization of the Bregman distance.

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

3.4 Primal-dual Policy Learning Given the Model

We can now put together all we have learned so far in this lecture to solve the Saddle point formulation. Recall the saddle point formulation:

$$\min_V \max_{\lambda\in\Delta_{\mathcal{S}\times\mathcal{A}}} (1-\gamma)\langle \mu, V\rangle + \langle \lambda, r + \gamma PV - EV\rangle .$$

To solve for V, we use the PPM with the quadratic regularizer $\frac{1}{2\eta}\|V - V_K\|_2$. So, we want to minimize the following regularized objective w.r.t V:

$$V_{k+1} = \arg\min_V (1-\gamma)\langle \mu, V\rangle + \langle \lambda, r + \gamma PV - EV\rangle + \frac{1}{2\eta}\|V - V_K\|_2.$$

By taking the derivative of the regularized objective at k^{th} iteration and setting it to zero we have:

$$
\begin{aligned}
&(1-\gamma)\mu + \langle \lambda_k, \gamma P - E\rangle + \frac{1}{\eta}(V_{k+1} - V_k) = 0 \\
&\Rightarrow \frac{1}{\eta}(V_{k+1} - V_k) = -\Big((\gamma P - E)^T \lambda_k + (1-\gamma)\mu\Big) \\
&\Rightarrow V_{k+1} = V_k - \eta\Big((\gamma P - E)^T \lambda_k + (1-\gamma)\mu\Big).
\end{aligned}
$$

For the occupancy measure we use the Bregman distance generated by the function $\omega(\lambda_k) = \langle \lambda_k, \log \lambda_k \rangle$ which is the KL divergence. As a result, we want to maximize the following objective w.r.t λ:

$$
\lambda_{k+1} = \underset{\lambda \in \nabla_{\mathcal{S}\times\mathcal{A}}}{\arg\max} (1-\gamma)\langle \mu, V\rangle + \langle \lambda, r + \gamma PV - EV\rangle - \frac{1}{\eta} KL(\lambda \| \lambda_k),
$$

where $KL(\lambda\|\lambda_k) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right)$ is the Kullback-Leibler divergence.

To solve this problem we introduce an auxiliary problem for any V as follows:

$$
\lambda^V_{k+1} = \text{argmax}_{\lambda\in\nabla}\left(\langle \lambda, r\rangle + \left\langle V, \gamma P^T \lambda - E^T \lambda\right\rangle + (1-\gamma)\langle V, \mu\rangle - \frac{1}{\eta}\left\langle \lambda,\ \log\left(\frac{\lambda}{\lambda_k}\right)\right\rangle\right).
$$

By the optimality condition we get:

$$
\begin{aligned}
&r + \left\langle V, \gamma P^T - E^T\right\rangle - \frac{1}{\eta}\nabla_\lambda\left(\left\langle \lambda^V_{k+1}, \log\left(\frac{\lambda^V_{k+1}}{\lambda_k}\right)\right\rangle\right) = 0 \\
&\Rightarrow r + \left\langle V, \gamma P^T - E^T\right\rangle - \frac{1}{\eta}\left(\log\left(\frac{\lambda^V_{k+1}}{\lambda_k} + 1\right)\right) = 0 \\
&\Rightarrow \eta\,(r + \gamma PV - EV - 1) = \log\left(\frac{\lambda^V_{k+1}}{\lambda_k}\right) \\
&\Rightarrow \lambda^V_{k+1} \propto \lambda_k \exp^{\eta(r+\gamma PV - EV)}
\end{aligned}
$$

By normalizing the above formulation we get the following update for occupancy measure at iteration $k+1$:

$$
\lambda^V_{k+1}(s,a) = \frac{\lambda_k(s,a)\exp^{\eta(r(s,a)+\gamma(PV_{k+1})(s,a)-(EV_{k+1})(s,a))}}{\sum_{s',a'}\lambda_k(s',a')\exp^{\eta(r(s',a')+\gamma(PV)(s',a')-(EV_{k+1})(s',a'))}}.
$$

As a result, the Saddle point formulation can be solved via primal-dual gradient updates:

- $V_{k+1} = V_k - \eta\Big((\gamma P - E)^T \lambda_k + (1-\gamma)\mu\Big)$
- $\lambda_{k+1} \propto \lambda_k \exp^{\eta(r+\gamma PV_{k+1} - EV_{k+1})}$.

4 REPS Algorithm

We finally have all the tools needed to understand the relative entropy policy search (REPS) algorithm [Peters et al., 2010]. REPS is a popular algorithm in robotics. A robot trained with REPS manages to play table tennis!

REPS applies the PPM to the dual LP using the PPM with Bregman distance. We showed in the previous section that using the Bregman distance generated by the function $\omega(\lambda_k) = \langle \lambda_k, \log \lambda_k\rangle$ we get the following objective:

$$
(1-\gamma)\langle \mu, V\rangle + \langle \lambda, r + \gamma PV - EV\rangle - \frac{1}{\eta} KL(\lambda\|\lambda_k), \tag{14}
$$

which led to the following update for the occupancy measure:

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

Now we solve Equation 14 for V:

$$V_{k+1} = \text{argmin}_V \left(\langle \lambda^V_{k+1}, r \rangle + \left\langle V, \gamma P^T \lambda^V_{k+1} - E^T \lambda^V_{k+1} \right\rangle + (1-\gamma) \langle V, \mu \rangle - \frac{1}{\eta} \left\langle \lambda^V_{k+1}, \ \log\left(\frac{\lambda^V_{k+1}}{\lambda_k} \right) \right\rangle \right).$$

We replace λ^V_{k+1} with what we found above. For ease of notation we consider $N = \sum_{s',a'} \lambda_k(s',a') \exp^{\eta(r(s',a')+\gamma(PV)(s',a')-(EV)(s',a'))}$. We have:

$$\begin{aligned}
&\log\left(\frac{\lambda^V_{k+1}}{\lambda_k} \right) = \eta\,(r + \gamma PV - EV) - \log N \\
&\Rightarrow V_{k+1} = \text{argmin}_V \left(\langle \lambda^V_{k+1}, r \rangle + \left\langle V, \gamma P^T \lambda^V_{k+1} - E^T \lambda^V_{k+1} \right\rangle + (1-\gamma) \langle V, \mu \rangle - \frac{1}{\eta} \left\langle \lambda^V_{k+1}, \ \log\left(\frac{\lambda^V_{k+1}}{\lambda_k} \right) \right\rangle \right) \\
&= \text{argmin}_V \left(\langle \lambda^V_{k+1}, r \rangle + \left\langle V, \gamma P^T \lambda^V_{k+1} - E^T \lambda^V_{k+1} \right\rangle + (1-\gamma) \langle V, \mu \rangle - \frac{1}{\eta} \left\langle \lambda^V_{k+1}, \ (\eta\,(r + \gamma PV - EV) - \log N) \right\rangle \right) \\
&= \text{argmin}_V \left((1-\gamma) \langle V, \mu \rangle + \frac{1}{\eta} \left\langle \lambda^V_{k+1}, \ \log N \right\rangle \right) \\
&= \text{argmin}_V \left((1-\gamma) \langle V, \mu \rangle + \frac{1}{\eta} \left\langle \lambda^V_{k+1}, 1 \right\rangle \ \log N \right) \quad \text{(as } \log N \text{ is a scalar)} \\
&= \text{argmin}_V \left((1-\gamma) \langle V, \mu \rangle + \frac{1}{\eta} \ \log N \right) \quad \text{(as } \lambda^V_{k+1} \text{ is normalized and thus } \left\langle \lambda^V_{k+1}, 1 \right\rangle = 1) \\
&= \text{argmin}_V \left((1-\gamma) \langle V, \mu \rangle + \frac{1}{\eta} \ \log \sum_{s',a'} \lambda_k(s',a') \exp^{\eta(r(s',a')+\gamma(PV)(s',a')-(EV)(s',a'))} \right).
\end{aligned}$$

Putting together the obtained formulations for λ_{k+1} and V_{k+1} we arrive to the REPS algorithm.

Algorithm 1 REPS Algorithm

1: **for** each iteration $k = 1, ..., K$ **do**
2: Solve the problem

$$V_{k+1} = \text{argmin}_V \left((1-\gamma) \langle \mu, V \rangle + \frac{1}{\eta} \ \log \sum_{s,a} \lambda_k(s,a) \exp^{\eta(r(s,a)+\gamma(PV)(s,a)-(EV)(s,a))} \right)$$

3: Update the occupancy measure:

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

4: **end for**

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

Table 4: Sample complexity of the REPS algorithm with exact gradient and stochastic bias gradient oracles.

Table 4 summarizes the sample complexity of the REPS algorithm. When the oracle is the exact gradient REPS achieves the best known complexity. The stochastic biased gradient solves the exploration problem assuming $\lambda_k(s,a) \geq \beta > 0$.

5 Conclusion

In this lecture, we studied the Linear Programming approach to Reinforcement Learning. LP allows us to formulate RL as a convex optimization problem. We introduced the primal and dual LP which are equivalent formulations of the RL objective. Combining the primal and dual LP we reached the Saddle point formulation. We also talked about the Proximal Point Method (PPM) which we used to solve the Saddle Point problem. Finally, we applied PPM to the dual LP to obtain the famous REPS algorithm.

References

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

J. Peters, K. Mulling, and Y. Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pages 1607–1612, 2010.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5): 877–898, 1976.