REINFORCEMENT LEARNING VIA LINEAR PROGRAMMING

Gergely Neu

Universitat Pompeu Fabra
Barcelona
OUTLINE

• Reinforcement Learning
• Markov Decision Processes and the Bellman equations
• Linear Programming for MDPs
• A new breed of RL algorithms
  • Relative entropy policy search
  • Primal-dual methods
Agent in state $x_t$, take action $a_t$

Reward $r_t$, new state $x_{t+1}$

Goal: learn behaviors that maximize reward on the long run
REINFORCEMENT LEARNING

Agent in state $x_t$, take action $a_t$

Environment

reward $r_t$, new state $x_{t+1}$

Why is this interesting?
• Model captures many important real-world problems!

Goal: learn behaviors that maximize reward on the long run
REINFORCEMENT LEARNING

Agent

in state $x_t$, take action $a_t$

Environment

reward $r_t$, new state $x_{t+1}$

Why is this interesting?
• Model captures many important real-world problems!

Why is this challenging?
• Environment dynamics typically unknown
• Actions influence long-term performance

Goal: learn behaviors that maximize reward on the long run
REINFORCEMENT LEARNING VS. SUPERVISED LEARNING

Supervised learning: detect and classify objects
Supervised learning: detect and classify objects

Reinforcement learning: detect and classify objects AND take actions
RL BREAKTHROUGHS

Superhuman performance in
- Atari (Mnih et al., 2013)
- Go (Silver et al., 2016, 2017)
- Starcraft (Silver et al., 2019)

Emerging applications in
- Robotics
- Autonomous driving
- Dialogue management
- Recommendation systems,…
RL BREAKTHROUGHS

Superhuman performance in
• Atari (Mnih et al., 2013)
• Go (Silver et al., 2016, 2017)
• Starcraft (Silver et al., 2019)

Emerging applications in
• Robotics
• Autonomous driving
• Dialogue management
• Recommendation systems,…

This talk:
taking a fresh look at the foundations
MARKOV DECISION PROCESSES
and the Bellman equations
Learner:
• Observe state $x_t$, take action $a_t$
• Obtain reward $r(x_t, a_t)$

Environment:
• Generate next state $x_{t+1} \sim P(\cdot|x_t, a_t)$
**MARKOV DECISION PROCESSES**

**Learner:**
- Observe state $x_t$, take action $a_t$
- Obtain reward $r(x_t, a_t)$

**Environment:**
- Generate next state $x_{t+1} \sim P(\cdot | x_t, a_t)$

**Goal:**
maximize discounted return

$$R = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \right]$$

from initial state $x_0 \sim \nu_0$

$\gamma \in (0,1)$
**BASIC MDP FACTS**

- Markov property: $x_{t+1}$ only depends on $(x_t, a_t)$
- Stationarity: $P(\cdot | x_t, a_t)$ doesn’t depend on $t$

$\Rightarrow \text{enough to consider stationary policies}$

$$\pi(a|x) = \mathbb{P}[a_t = a | x_t = x]$$

- Many other beautiful properties:
  - There is a deterministic optimal policy
  - Simultaneous optimality regardless of $\nu_0$
  - ...
The Bellman optimality equations

\[ Q^*(x, a) = r(x, a) + \gamma \mathbb{E}[\max_{a'} Q^*(x', a') | x, a] \]
The Bellman optimality equations

\[ Q^*(x, a) = r(x, a) + \gamma \mathbb{E}[\max_{a'} Q^*(x', a') | x, a] \]

- value of taking action \( a \) in state \( x \)
- immediate reward
- expected future value
The Bellman optimality equations

\[ Q^*(x, a) = r(x, a) + \gamma \mathbb{E}[\max_{a'} Q^*(x', a') | x, a] \]

value of taking action \( a \) in state \( x \)

immediate reward

expected future value

Optimal policy can be extracted as:

\[ \pi(a|x) = \begin{cases} 1 & \text{if } a = \arg \max_{a'} Q^*(x, a') \\ 0 & \text{otherwise} \end{cases} \]
The Bellman optimality equations

\[ Q^*(x, a) = r(x, a) + \gamma \mathbb{E} \left[ \max_{a'} Q^*(x', a') \mid x, a \right] \]

value of taking action \( a \) in state \( x \)

immediate reward

expected future value

Optimal policy can be extracted as:

\[ \pi(a \mid x) = \begin{cases} 1 & \text{if } a = \arg \max_{a'} Q^*(x, a') \\ 0 & \text{otherwise} \end{cases} \]

Richard Bellman (1954): Solution can be found via “Dynamic Programming”
The Bellman optimality equations

\[ Q^*(x, a) = r(x, a) + \gamma \mathbb{E}[\max_{a'} Q^*(x', a') | x, a] \]

value of taking action \(a\) in state \(x\)  
immediate reward  
expected future value

Challenges for reinforcement learning:

- Expectation over next state \(x'\) cannot be computed explicitly when transition dynamics \(P\) are unknown!
- No hope of finding exact solution when state space is large!
The Recipe for Modern RL Algorithms

• Parametrize a set of $Q$-functions: $Q_\theta: \theta \to \mathbb{R}^{X \times A}$
  (e.g., via neural networks)

• Find a $Q$-function that approximately solves the Bellman equations, e.g.,
  by minimizing the “squared Bellman error”:
  $$\mathcal{L}(Q) = \mathbb{E}_{(x,a) \sim \mu} \left[ (r(x,a) + \gamma \mathbb{E}[\max_{a'} Q(x',a') | x, a] - Q(x,a))^2 \right]$$

• Add lots of heuristics to stabilize training

• Add lots of computational resources and bake on 1000 GPUs until ready
The Recipe for Modern RL Algorithms

• Parametrize a set of Q functions: $Q_\theta : \mathbb{R}^X \times \mathcal{A} \rightarrow \mathbb{R}$ (e.g., via neural networks)

• Find a Q-function that approximately solves the Bellman equations, e.g., by minimizing the "squared Bellman error":

$$\mathcal{L}_Q = \mathbb{E}_{x, a \sim \mu} \left[ r(x, a) + \gamma \mathbb{E} \left[ \max_{a'} Q(x', a') | x, a \right] - Q(x, a) \right]^2$$

• Add lots of heuristics to stabilize training

• Add lots of computational resources and bake on 1000 GPUs until ready

Don’t try this at home!

• This objective is
  • non-convex
  • non-smooth
  • impossible to evaluate

• Does this process converge anywhere at all?

• If it converges, does it lead to a good policy??

$\mathcal{L}(Q) = \mathbb{E}_{(x, a) \sim \mu} \left[ \left( r(x, a) + \gamma \mathbb{E} \left[ \max_{a'} Q(x', a') | x, a \right] - Q(x, a) \right) \right]^2$
Observe: the discounted return of policy $\pi$ is

$$R^\pi_\gamma = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \right]$$
Observe: the discounted return of policy $\pi$ is

$$R^\pi_\gamma = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \right] = \sum_{t=0}^{\infty} \gamma^t E_\pi [r(x_t, a_t)]$$
Observe: the discounted return of policy $\pi$ is

$$R_\gamma^\pi = \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)]$$

$$= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\pi [r(x_t, a_t)]$$

$$= \sum_{t=0}^{\infty} \gamma^t \sum_{x,a} P_\pi [x_t = x, a_t = a] r(x, a)$$

A LINEAR REFORMULATION
A LINEAR REFORMULATION

Observe: the discounted return of policy $\pi$ is

\[
R^\pi_\gamma = E_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \right]
\]

\[
= \sum_{t=0}^{\infty} \gamma^t E_\pi [r(x_t, a_t)]
\]

\[
= \sum_{t=0}^{\infty} \gamma^t \sum_{x,a} P_\pi [x_t = x, a_t = a] r(x, a)
\]

\[
= \sum_{x,a} \sum_{t=0}^{\infty} \gamma^t P_\pi [x_t = x, a_t = a] r(x, a)
\]
A LINEAR REFORMULATION

Observe: the discounted return of policy $\pi$ is

$$R_\gamma^\pi = \mathbb{E}_\pi [\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)]$$

$$= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\pi [r(x_t, a_t)]$$

$$= \sum_{t=0}^{\infty} \gamma^t \sum_{x,a} P_\pi [x_t = x, a_t = a] r(x, a)$$

$$= \sum_{x,a} \sum_{t=0}^{\infty} \gamma^t P_\pi [x_t = x, a_t = a] r(x, a)$$

$$\overset{\text{def}}{=} \mu_\pi(x, a)$$

discounted occupancy measure of $\pi$
A LINEAR REFORMULATION

Observe: the discounted return of policy $\pi$ is

$$R^\pi_\gamma = \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(x_t, a_t)]$$

$$= \sum_{t=0}^\infty \gamma^t \mathbb{E}_\pi[r(x_t, a_t)]$$

$$= \sum_{t=0}^\infty \gamma^t \sum_{x,a} \mathbb{P}_\pi[x_t = x, a_t = a]r(x,a)$$

$$= \sum_{x,a} \sum_{t=0}^\infty \gamma^t \mathbb{P}_\pi[x_t = x, a_t = a]r(x,a)$$

$$\overset{\text{def}}{=} \mu_\pi(x,a)$$

Discounted return is linear in $\mu_\pi$:

$$R^\pi_\gamma = \langle \mu_\pi, r \rangle \overset{\text{def}}{=} \sum_{x,a} \mu_\pi(x,a)r(x,a)$$
EXAMPLE: 2D STATE SPACE

- \( \mu_{\pi_1} \)
- \( \mu_{\pi_2} \)
- \( \mu_{\pi_3} \)
- \( \mu_{\pi_4} \)
- \( \mu_{\pi_5} \)
- \( \mu_{\pi_6} \)

\( \bullet = \text{initial state} \)
EXAMPLE: 2D STATE SPACE

\[ \begin{align*}
\mu_{\pi_1} & \quad \mu_{\pi_2} & \quad \mu_{\pi_3} \\
\mu_{\pi_4} & \quad \mu_{\pi_5} & \quad \mu_{\pi_6}
\end{align*} \]

\[ \times \]

reward function

\( \bullet = \text{initial state} \)
EXAMPLE: 2D STATE SPACE

\[ \begin{bmatrix} \mu_{\pi_1} \\ \mu_{\pi_2} \\ \mu_{\pi_3} \\ \mu_{\pi_4} \\ \mu_{\pi_5} \\ \mu_{\pi_6} \end{bmatrix} \times \text{reward function} \]

\( \bullet \) = initial state
EXAMPLE: 2D STATE SPACE

How can we do this efficiently over the set of all policies?

- $\mu_{\pi_1}$
- $\mu_{\pi_2}$
- $\mu_{\pi_3}$
- $\mu_{\pi_4}$
- $\mu_{\pi_5}$
- $\mu_{\pi_6}$

$\times$

reward function

$\bullet =$ initial state
For any policy $\pi$, the occupancy measure satisfies

$$\sum_a \mu_\pi(x, a) = \nu_0(x) + \gamma (P \mu_\pi)(x)$$
For any policy $\pi$, the occupancy measure satisfies

$$\sum_{a} \mu_{\pi}(x, a) = \nu_{0}(x) + \gamma (P \mu_{\pi})(x)$$

THE SET OF OCCUPANCY MEASURES

occupancy of state $X_t = x$
initial state distribution

occupancy of next state $X_{t+1} = x$
For any policy $\pi$, the occupancy measure satisfies

$$\sum_a \mu_\pi(x,a) = \nu_0(x) + \gamma (P\mu_\pi)(x)$$

**Theorem (Manne 1960)**

$\mu$ is a valid occupancy measure if and only if it satisfies

$$E\mu = \gamma P\mu + \nu_0$$

“Bellman flow constraints”
Linear Programming for MDPs

maximize \( \langle \mu, r \rangle \)
subject to \( E^T \mu = \gamma P^T \mu + v_0 \)
Linear Programming for MDPs

\[
\begin{align*}
\text{maximize} & \quad \langle \mu, r \rangle \\
\text{subject to} & \quad E^T \mu = \gamma P^T \mu + v_0
\end{align*}
\]

- Optimal policy \( \pi^* \) can be extracted from solution \( \mu^* \) as

\[
\pi^*(a|x) = \frac{\mu^*(x,a)}{\sum_{a'} \mu^*(x,a')}
\]

- Basic solutions correspond to deterministic policies
THE LP FORMULATION

**Linear Programming for MDPs**

maximize \( \langle \mu, r \rangle \)
subject to \( E^T \mu = \gamma P^T \mu + v_0 \)

- Optimal policy \( \pi^* \) can be extracted from solution \( \mu^* \) as
  \[
  \pi^*(a|x) = \frac{\mu^*(x,a)}{\sum_a \mu^*(x,a')}
  \]
- Basic solutions correspond to deterministic policies

**Dual Linear Program for MDPs**

minimize \( \langle v_0, V \rangle \)
subject to \( EV \geq r + \gamma PV \)

- Dual solution related to Bellman eqns as
  \[
  Q^* = r + \gamma PV^*
  \]
Why is this useful?

• Defining optimality is very simple: no value functions, no fixed-point equations, no nonlinearity… just a single numerical objective!
• Easily comprehensible with an optimization background
• Powerful tool for developing algorithms
PROS AND CONS

Why is this useful?
• Defining optimality is very simple: no value functions, no fixed-point equations, no nonlinearity… just a single numerical objective!
• Easily comprehensible with an optimization background
  • Powerful tool for developing algorithms

“Why don’t they teach this in school?!?”
• Need to ensure $\mu^*(x, a) > 0$ to extract policy :'(
  • Temporal aspect is a bit abstract
• Number of variables and constraints is large
A BIT OF HISTORY

  - Formulated the primal LP and showed equivalence to Bellman eqns.
- Schweitzer & Seidmann (1982)
  - Proposed a relaxation to reduce the number of constraints
  - (also proposed the squared Bellman error objective!)
- De Farias & Van Roy (2003)
  - Analyzed the reduction of [SS82]
  - Inspired some follow-up work in RL [dFvR05,PZ09,PTPZ10,DFM12,LBS17]
A BIT OF HISTORY

  - Formulated the primal LP and showed equivalence to Bellman eqns.
- Schweitzer & Seidmann (1982)
  - Proposed a relaxation to reduce the number of constraints
  - (also proposed the squared Bellman error objective!)
- De Farias & Van Roy (2003)
  - Analyzed the reduction of [SS82]
  - Inspired some follow-up work in RL [dFvR05, PZ09, PTPZ10, DFM12, LBS17]

Common theme: analyze quality of approximate solution & solve the LP with generic solver
A BIT OF HISTORY

  - Formulated the primal LP and showed equivalence to Bellman equations.
- Schweitzer & Seidman (1982)
  - Proposed a relaxation to reduce the number of constraints
  - (also proposed the squared Bellman error objective!)
- De Farias & Van Roy (2003)
  - Analyzed the reduction of [SS82]
  - Inspired some follow-up work in RL [dFvR05, PZ09, PTPZ10, DFM12, LBS17]

Common theme: analyze quality of approximate solution & solve the LP with generic solver
A NEW BREED OF RL ALGORITHMS
Linear Program for MDPs

maximize $\langle \mu, r \rangle$
subject to $E^T \mu = \gamma P^T \mu + v_0$

• add regularization for tractable solution
• relax constraints like [SS85]
RELATIVE ENTROPY POLICY SEARCH
Peters, Mülling, Altün (2010)

**REPS** (primal form)

maximize \( \langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu|\mu_0) \)

subject to \( \Psi^T E^T \mu = \gamma \Psi^T P^T \mu + \Psi^T \nu_0 \)

- add regularization for tractable solution
- relax constraints like [SS85]
RELATIVE ENTROPY POLICY SEARCH
Peters, Mülling, Altün (2010)

**REPS (primal form)**

\[
\begin{align*}
\text{maximize} & \quad \langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu | \mu_0) \\
\text{subject to} & \quad \Psi^T E^T \mu = \gamma \Psi^T P^T \mu + \Psi^T \nu_0
\end{align*}
\]

- add regularization for tractable solution
- relax constraints like [SS85]

**REPS (dual form)**

\[
\begin{align*}
\theta^* & = \min_{\theta} \frac{1}{\eta} \log \mathbb{E}_{x,a \sim \mu_0} \left[ e^{\eta (r(x,a) + \gamma \mathbb{E} [\Psi \theta(x') | x,a] - \Psi \theta(x))} \right] \\
\mu^* & = \mu_0 \circ e^{\eta (r + \gamma P \Psi \theta^* - E \Psi \theta^*)}
\end{align*}
\]

Lagrangian duality
**RELATIVE ENTROPY POLICY SEARCH**

Peters, Mülling, Altün (2010)

**REPS (primal form)**

\[
\begin{align*}
\text{maximize} & \quad \langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu|\mu_0) \\
\text{subject to} & \quad \Psi^T E^T \mu = \gamma \Psi^T P^T \mu + \Psi^T \nu_0
\end{align*}
\]

- add regularization for tractable solution
- relax constraints like [SS85]

**REPS (dual form)**

\[
\begin{align*}
\theta^* &= \min_{\theta} \frac{1}{\eta} \log \mathbb{E}_{x,a \sim \mu_0} \left[ e^{\eta (r(x,a) + \gamma \mathbb{E}[\Psi \theta(x')]|x,a] - \Psi \theta(x))} \right] \\
\mu^* &= \mu_0 \circ e^{\eta (r + \gamma P \Psi \theta^* - EP \Psi \theta^*)}
\end{align*}
\]
RELATIVE ENTROPY POLICY SEARCH
Peters, Mülling, Altün (2010)

**REPS (primal form)**

\[
\begin{align*}
\text{maximize} & \quad \langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu|\mu_0) \\
\text{subject to} & \quad \Psi^T E^T \mu = \gamma \Psi^T P^T \mu + \Psi^T v_0
\end{align*}
\]

- add regularization for tractable solution
- relax constraints like [SS85]

**REPS (dual form)**

- \( \theta^* = \min_{\theta} \frac{1}{\eta} \log \mathbb{E}_{x,a \sim \mu_0} [e^{\eta (r(x,a) + \gamma \mathbb{E}[\Psi \theta(x')|x,a] - \Psi \theta(x))}] \)
- \( \mu^* = \mu_0 \circ e^{\eta (r + \gamma P \Psi \theta^* - E \Psi \theta^*)} \)

Lagrangian duality
Unconstrained convex optimization problem!
Intractable due to unknown \( P \) in exponent!
RELATIVE ENTROPY POLICY SEARCH
Peters, Mülling, Altün (2010)

REPS (primal form)

maximize $\langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu|\mu_0)$
subject to $\Psi^T E^T \mu = \gamma \Psi^T P^T \mu + \Psi^T \nu_0$

Can we do better?

REPS (dual form)

- $\theta^* = \min_{\theta} \frac{1}{\eta} \log \mathbb{E}_{x,a \sim \mu_0} [e^{\eta (r(x,a) + \gamma \mathbb{E}[\Psi \theta(x')|x,a] - \Psi \theta(x))}]$
- $\mu^* = \mu_0 \circ e^{\eta (r + \gamma P \Psi \theta^* - E \Psi \theta^*)}$

Lagrangian duality

Unconstrained convex optimization problem!

Intractable due to unknown $P$ in exponent!
**Q-REPS** (primal form)

maximize \( \langle \mu, r \rangle - \frac{1}{\eta} KL(\mu | \mu_0) - \frac{1}{\alpha} H(u | u_0) \)

subject to \( E^T \mu = \gamma P^T u + \nu_0 \)
\( \Phi^T \mu = \Phi^T u \)

- Lagrangian decomposition to introduce “Q”
- Composite regularization
LOGISTIC Q-LEARNING
Bas-Serrano, Curi, Krause & Neu (2021)

\[ \text{Q-REPS (primal form)} \]

maximize \[ \langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu|\mu_0) - \frac{1}{\alpha} H(u|u_0) \]
subject to \[ E^T \mu = \gamma P^T u + v_0 \]
\[ \Phi^T \mu = \Phi^T u \]

- Lagrangian decomposition to introduce “Q”
- Composite regularization

\[ \text{Q-REPS (dual form)} \]

\[ \theta^* = \min_{\theta} \frac{1}{\eta} \log \mathbb{E}_{x,a \sim \mu_0} \left[ e^{\eta (r(x,a) + \gamma \mathbb{E}[V_\theta(x')] | x,a)} - \Phi \theta(x,a) \right] \]
\[ \pi^* = \pi_0 \circ e^{\eta (\Phi \theta^* - V_{\theta^*})} \]
**LOGISTIC Q-LEARNING**
Bas-Serrano, Curi, Krause & Neu (2021)

---

**Q-REPS (primal form)**

maximize \( \langle \mu, r \rangle - \frac{1}{\eta} \text{KL}(\mu | \mu_0) - \frac{1}{\alpha} H(u | u_0) \)

subject to \( E^T \mu = \gamma P^T u + \nu_0 \)

\( \Phi^T \mu = \Phi^T u \)

---

- Lagrangian decomposition to introduce “Q”
- Composite regularization

---

**Q-REPS (dual form)**

- \( \theta^* = \min_{\theta} \frac{1}{\eta} \log \mathbb{E}_{x,a \sim \mu_0} \left[ e^{\eta (r(x,a) + \gamma \mathbb{E}[V_{\theta}(x') | x,a] - \Phi \theta(x,a))} \right] \)
- \( \pi^* = \pi_0 \circ e^{\eta (\Phi \theta^* - V_{\theta^*})} \)

---

Lagrangian duality

Unconstrained convex optimization problem!

Explicit, tractable policy update!!
A PRINCIPLED LOSS FUNCTION
Bas-Serrano, Curi, Krause & Neu (2021)

The Logistic Bellman Error (LBE)

\[
\mathcal{G}(Q) = \frac{1}{\eta} \log \mathbb{E}_{(x,a) \sim \mu_0} \left[ e^{\eta (r(x,a) + \gamma \mathbb{E}[V_Q(x')|x,a] - Q(x,a))} \right]
\]

• Convex and smooth (composition of two monotone convex functions that are smooth)
• 2-Lipschitz w.r.t. \( \ell_\infty \)-norm:
  \[
  \| \nabla_Q \mathcal{G}_k(Q) \|_1 \leq 2
  \]
• Easy to estimate reliably using sample transitions
The Logistic Bellman Error (LBE)

\[ G(Q) = \frac{1}{\eta} \log \mathbb{E}_{(x,a) \sim \mu_0} \left[ e^{\eta (r(x,a) + \gamma \mathbb{E}[V_Q(x')|x,a] - Q(x,a))} \right] \]
STRONG GUARANTEES!
Bas-Serrano, Curi, Krause & Neu (2021)

“Theorem”
\[ |G_k(\theta) - \hat{G}_k(\theta)| = O(\eta) \]

“LBE can be estimated with small bias”
Impossible for squared BE!

“Theorem”
\[ \text{err}_K \leq O\left(\frac{1}{K} \sum_{k=1}^{K} (\varepsilon_k + \sqrt{\eta \varepsilon_k})\right) \]

“Optimization errors \( \varepsilon_k \) have moderate long-term impact”
Comparable with best results for SBE!
AND IT WORKS!!!
• Primal-dual methods:
  • consider equivalent saddle-point problem
    \[
    \max_{\mu} \min_{V} \left\langle \mu, r + \gamma PV - EV \right\rangle + \langle v_0, V \rangle
    \]
  • solve with primal-dual gradient descent
  • scale up by parametrizing \( \mu = U\lambda \) and \( V = \Psi\theta \)
OTHER LP-BASED METHODS

• Primal-dual methods:
  • consider equivalent saddle-point problem
    \[ \max_{\mu} \min_{V} \langle \mu, r + \gamma PV - EV \rangle + \langle v_0, V \rangle \]
  • solve with primal-dual gradient descent
  • scale up by parametrizing \( \mu = U\lambda \) and \( V = \Psi\theta \)

• Implementable with only sample access to \( P \)
• State of the art method for small MDPs
• When features \( \Phi \) and \( \Psi \) are chosen well:
  • guaranteed convergence to optimum
  • excellent empirical performance

• Off-policy RL: fixed data set sampled from $\mu_0$
• “DualDICE” reparametrization of primal variables:
  $$\xi(x, a) = \frac{\mu(x, a)}{\mu_0(x, a)}$$
• Leads to new primal-dual and REPS-like algorithms
• Off-policy RL: fixed data set sampled from $\mu_0$
• “DualDICE” reparametrization of primal variables:
  \[ \xi(x, a) = \frac{\mu(x, a)}{\mu_0(x, a)} \]
• Leads to new primal-dual and REPS-like algorithms

• Incredibly practical methods for off-policy value estimation!
• Even works without knowledge of $\mu_0$!!

SUMMARY

• LP formulation is currently obscure but holds huge potential!
• Solid alternative to fixed-point computation
• Historical limitations are mostly due to rigid interpretation
• Useful for deriving new algorithms & analyzing existing ones
• Lots of work left to do!
  • Room for improvement both in theory & practice
  • Existing toolbox not as well-developed as for other RL approaches
SUMMARY

• LP formulation is currently obscure but holds huge potential!
• Solid alternative to fixed-point computation
• Historical limitations are mostly due to rigid interpretation
• Useful for deriving new algorithms & analyzing existing ones
• Lots of work left to do!
  • Room for improvement both in theory & practice
  • Existing toolbox not as well-developed as for other RL approaches

THANKS!!
PRIMAL-DUAL METHODS

Primal LP for MDPs

\begin{align*}
\text{maximize} & \quad \langle \mu, r \rangle \\
\text{subject to} & \quad E^T \mu = \gamma P^T \mu + v_0
\end{align*}

Dual LP for MDPs

\begin{align*}
\text{minimize} & \quad \langle v_0, V \rangle \\
\text{subject to} & \quad EV \geq r + \gamma PV
\end{align*}
**PRIMAL-DUAL METHODS**

Primal LP for MDPs

- **maximize** \( \langle \mu, r \rangle \)
- **subject to** \( E^T \mu = \gamma P^T \mu + \nu_0 \)

Dual LP for MDPs

- **minimize** \( \langle \nu_0, V \rangle \)
- **subject to** \( EV \geq r + \gamma PV \)

Equivalent via Lagrangian duality

Primal-dual formulation for MDPs

- \( \max_{\mu} \min_{V} \langle \mu, r + \gamma PV - EV \rangle + \langle \nu_0, V \rangle \)
Primal-dual formulation for MDPs
\[
\max_\mu \min_V \langle \mu, r + \gamma PV - EV \rangle + \langle \nu_0, V \rangle
\]

Can be solved via iterative updates:
- \( V_{k+1} = V_k - \eta \left( (\gamma P - E)^T \mu_k + \nu_0 \right) \)
- \( \mu_{k+1} = \mu_k \circ e^{\eta (r + \gamma PV_k - EV_k)} \)
SADDLE-POINT OPTIMIZATION

Primal-dual formulation for MDPs
\[
\max_{\mu} \min_{V} \langle \mu, r + \gamma PV - EV \rangle + \langle v_0, V \rangle
\]

Can be solved via iterative updates:

- \( V_{k+1} = V_k - \eta \left( (\gamma P - E)^\top \mu_k + v_0 \right) \)
- \( \mu_{k+1} = \mu_k \circ e^{\eta (r + \gamma PV_k - EV_k)} \)

- Gradients are expectations under \( \mu_k \)
  \( \implies \text{efficient stochastic implementation} \)
SADDLE-POINT OPTIMIZATION

Primal-dual formulation for MDPs
\[
\max_{\mu} \min_V \langle \mu, r + \gamma PV - EV \rangle + \langle \nu_0, V \rangle
\]

Can be solved via iterative updates:
- \( V_{k+1} = V_k - \eta((\gamma P - E)^T \mu_k + \nu_0) \)
- \( \mu_{k+1} = \mu_k \circ e^{\eta(r + \gamma PV_k - EV_k)} \)

- Gradients are expectations under \( \mu_k \)
  \( \Rightarrow \) efficient stochastic implementation

State of the art sample complexity for solving “small” MDPs!
(Wang 2017)
SCALING UP

• **Problem:** intractable for large state spaces due to large number of constraints & variables!

• **Idea:** parametrize $\mu$ and $V$ via linear functions!
  - $\mu_\lambda = \Psi \lambda$ for some feature matrix $\Psi \in \mathbb{R}^{(X \times \mathcal{A}) \times n}$
  - $V_{\theta} = \Phi \theta$ for some feature matrix $\Phi \in \mathbb{R}^{X \times m}$

Primal-dual formulation for MDPs

\[
\max_{\mu} \min_{V} \langle \mu, r + \gamma PV - EV \rangle + \langle v_0, V \rangle
\]
Problem: intractable for large state spaces due to large number of constraints & variables!

Idea: parametrize $\mu$ and $V$ via linear functions!

- $\mu_\lambda = \Psi \lambda$ for some feature matrix $\Psi \in \mathbb{R}^{(X \times A) \times n}$
- $V_\theta = \Phi \theta$ for some feature matrix $\Phi \in \mathbb{R}^{X \times m}$

**Relaxed primal-dual formulation for MDPs**

$$\max_{\lambda} \min_{\theta} \langle \lambda, \Phi^T (r + \gamma P \Psi \theta - E \Psi \theta) \rangle + \langle v_0, \Psi \theta \rangle$$
SCALING UP

• **Problem:** intractable for large state spaces due to large number of constraints & variables!

• **Idea:** parametrize $\mu$ and $V$ via linear functions!
  
  $\mu_\lambda = \Psi \lambda$ for some feature matrix $\Psi \in \mathbb{R}^{(X \times A) \times n}$
  
  $V_\theta = \Phi \theta$ for some feature matrix $\Phi \in \mathbb{R}^{X \times m}$

**Relaxed primal-dual formulation for MDPs**

$$\max_{\lambda} \min_{\theta} \langle \lambda, \Phi^T (r + \gamma P \Psi \theta - E \Psi \theta) \rangle + \langle \nu_0, \Psi \theta \rangle$$

- $\theta_{k+1} = \theta_k - \eta \left( (\gamma P \Psi - E \Psi)^T \Phi \lambda_k + \Psi^T \nu_0 \right)$
- $\lambda_{k+1} = \lambda_k \circ e^{\eta \Phi^T (r + \gamma P \Psi \theta_k - E \Psi \theta_k)}$
SCALING UP

- Implementable with only sample access to transition function $P$
- When features $\Phi$ and $\Psi$ are chosen well:
  - guaranteed convergence to optimum
  - excellent empirical performance


\[
\begin{align*}
\theta_{k+1} &= \theta_k - \eta \left( (\gamma P \Psi - E \Psi) \Phi \lambda_k + \Psi^T \nu_0 \right) \\
\lambda_{k+1} &= \lambda_k \circ e^{\eta \Phi^T (r + \gamma P \Psi \theta_k - E \Psi \theta_k)}
\end{align*}
\]
OFF-POLICY LEARNING

• What if we can’t sample from $\mu_k$?
• Off-policy RL: fixed data set sampled from $\mu_0$
• What if we can’t sample from $\mu_k$?
• Off-policy RL: fixed data set sampled from $\mu_0$
• DualDICE reparametrization (Nachum & Dai, 2020):
  rewrite primal variables as $\xi(x, a) = \mu(x, a)/\mu_0(x, a)$
OFF-POLICY LEARNING

• What if we can’t sample from $\mu_k$?
• Off-policy RL: fixed data set sampled from $\mu_0$
• DualDICE reparametrization (Nachum & Dai, 2020): rewrite primal variables as $\xi(x, a) = \frac{\mu(x, a)}{\mu_0(x, a)}$

Primal-dual formulation for MDPs
\[
\max_{\mu} \min_{V} \langle \mu, r + \gamma PV - EV \rangle + \langle v_0, V \rangle
\]
OFF-POLICY LEARNING

• What if we can’t sample from $\mu_k$?
• Off-policy RL: fixed data set sampled from $\mu_0$
• DualDICE reparametrization (Nachum & Dai, 2020):
  rewrite primal variables as $\xi(x, a) = \mu(x, a)/\mu_0(x, a)$

DualDICE formulation for MDPs
$$\max_{\xi} \min_{V} \left\langle \xi, \mu_0 \circ (r + \gamma PV - EV) \right\rangle + \langle \nu_0, V \rangle$$
OFF-POLICY LEARNING

• What if we can’t sample from $\mu_k$?
• Off-policy RL: fixed data set sampled from $\mu_0$
• DualDICE reparametrization (Nachum & Dai, 2020): rewrite primal variables as $\xi(x, a) = \mu(x, a)/\mu_0(x, a)$

$$\begin{align*}
\max_{\xi} \min_V \langle \xi, \mu_0 \circ (r + \gamma PV - EV) \rangle + \langle \nu_0, V \rangle
\end{align*}$$

Incredibly practical methods for off-policy value estimation!
Even works without knowledge of $\mu_0$!!