LOGISTIC Q LEARNING

Gergely Neu
Universitat Pompeu Fabra, Barcelona
Joint work with
Joan Bas-Serrano, Sebastian Curi, Andreas Krause

OUTLINE

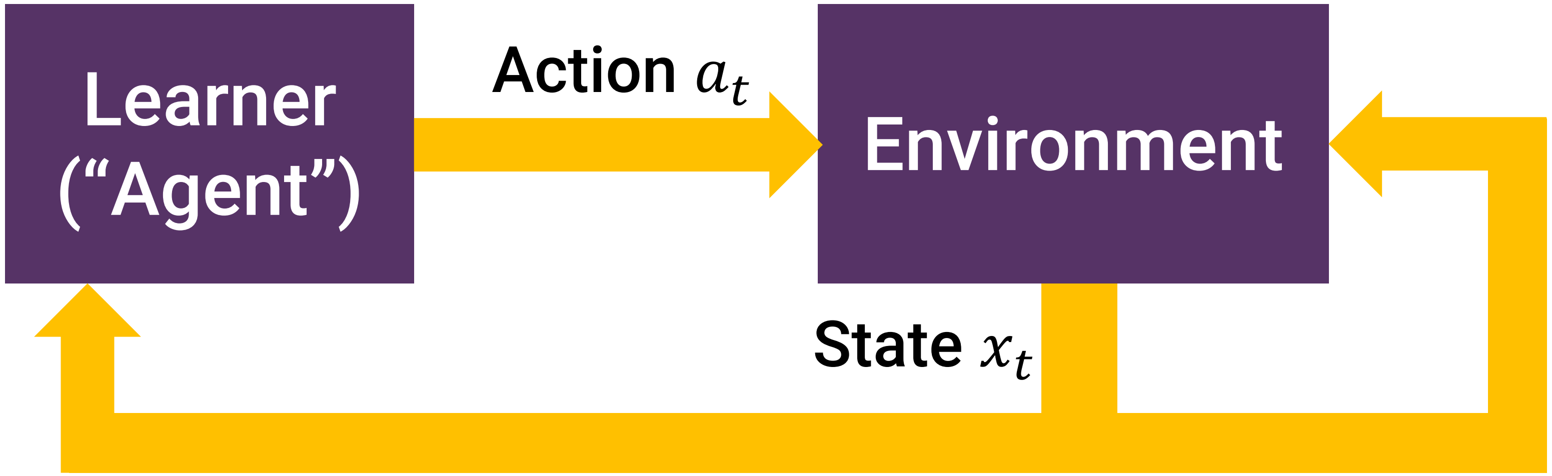
- The problem with modern RL
- Relative Entropy Policy Search
- REPS with Q-functions:

  Q-REPS

- Performance guarantees
- The derivation of Q-REPS
- Parting thoughts

Mainstream RL and REPS

MARKOV DECISION PROCESSES



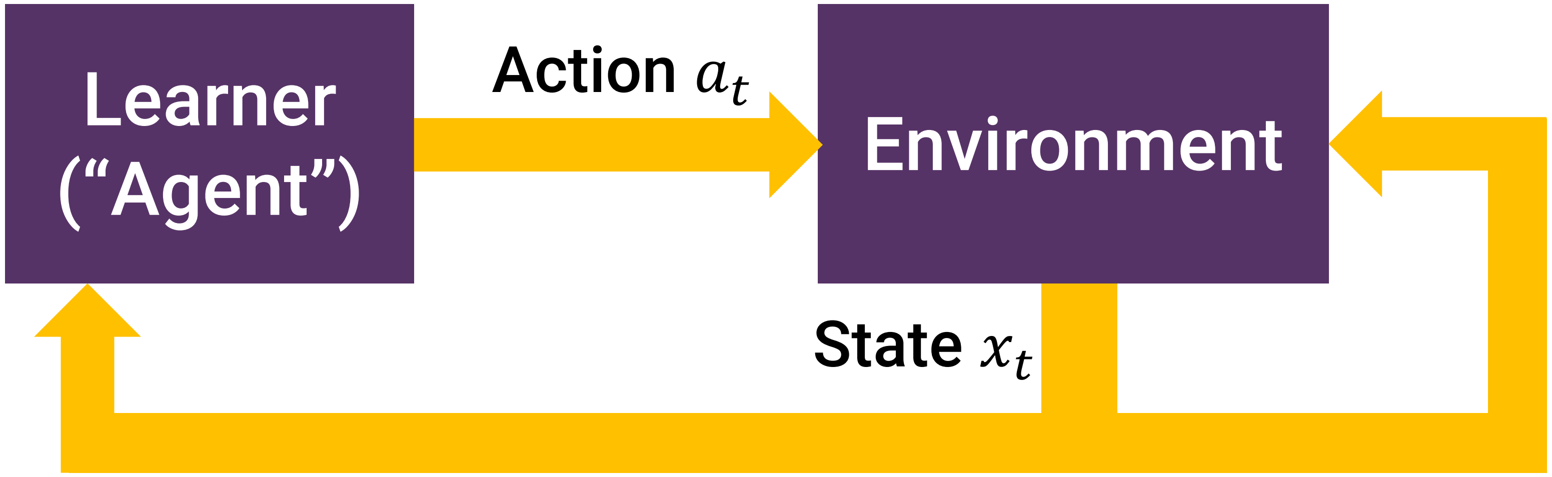
Learner:

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

Environment:

- Generate next state $x_{t+1} \sim P(\cdot \mid x, a)$

MARKOV DECISION PROCESSES



Learner:

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

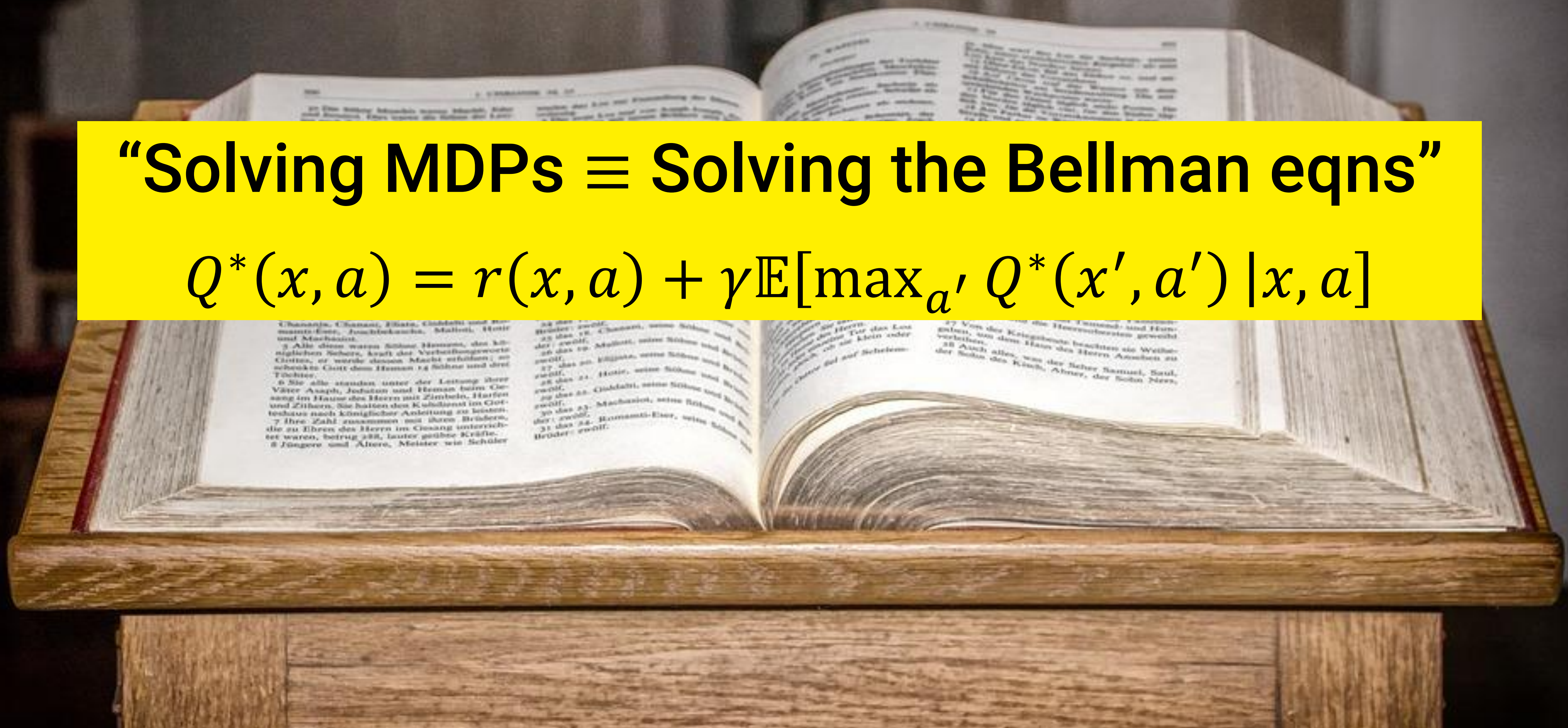
Environment:

- Generate next state $x_{t+1} \sim P(\cdot\,|x, a)$

Goal:
maximize discounted return

$$R = \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t)$$

THE GOSPEL OF MODERN RL



"Solving MDPs ≡ Solving the Bellman eqns"

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

THE GOSPEL OF MODERN RL

"Solving MDPs ≡ Solving the Bellman eqns"

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

Good news:

Optimal Q-function encodes optimal policy:

$$\pi^*(a|x) = \mathbb{I}_{\{a=\mathrm{argmax}_b\, Q^*(x,b)\}}$$

THE GOSPEL OF MODERN RL

"Solving MDPs ≡ Solving the Bellman eqns"

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

Good news:

Optimal Q-function encodes optimal policy:

$$\pi^*(a|x) = \mathbb{I}_{\{a=\operatorname{argmax}_b Q^*(x,b)\}}$$

Bad news:

solving systems of equations is not easy with modern ML tools!

THE SQUARED BELLMAN ERROR

Define the Bellman error

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

and measure the "goodness" of a Q-function with the loss

$$\mathcal{L}(Q) = \mathbb{E}_{(x,a)\sim\mu}\left[\left(\delta_Q(x,a)\right)^2\right]$$

THE SQUARED BELLMAN ERROR

Define the Bellman error

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

and measure the “goodness” of a Q-function with the loss

$$\mathcal{L}(Q) = \mathbb{E}_{(x,a)\sim\mu}\left[\left(\delta_Q(x,a)\right)^2\right]$$

TIME TO DO GRADIENT DESCENT!!!1!!

THE SQUARED BELLMAN ERROR

Define the Bellman error

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

and measure the “goodness” of a Q-function with the loss

$$\mathcal{L}(Q) = \mathbb{E}_{(x,a)\sim\mu}\left[\left(\delta_Q(x,a)\right)^2\right]$$

TIME TO DO GRADIENT DESCENT!!!1!!

Not so fast!

This loss is:

- non-convex, non-smooth & non-Lipschitz
- hard to estimate due to double sampling

THE SBE IS EVERYWHERE!

Patching the SBE:

- Target networks to break non-convexity & double sampling
- Gradient clipping for unbounded gradients
- ...

THE SBE IS EVERYWHERE!

Patching the SBE:

- Target networks to break non-convexity & double sampling
- Gradient clipping for unbounded gradients
- ...

Some version of SBE is used in:

- Deep Q networks
- Policy gradient / Actor-Critic methods
- TRPO / PPO / MPO
- ...

THE SBE IS EVERYWHERE!

Patching the SBE:

- Target networks to break non-convexity & double sampling
- Gradient clipping for unbounded gradients
- ...

Some version of SBE is used in:

- Deep Q networks
- Policy gradient / Actor-Critic methods
- TRPO / PPO / MPO
- ...

One exception: REPS!

SOMETHING DIFFERENT

Relative Entropy Policy Search

Jan Peters, Katharina Mülling, Yasemin Altün
Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076 Tübingen, Germany
{jrpeters,muelling,altun}@tuebingen.mpg.de

- Based on a linear-programming formulation instead of the Bellman equations (Manne, 1960)
- A "mirror descent" algorithm (Nemirovski & Yudin, 1983)
- Key practical novelty: a natural loss function!

RELATIVE ENTROPY POLICY SEARCH

REPS

Parameters: learning rate η, feature map $\psi: \mathcal{X} \to \mathbb{R}^m$

Initialization: policy π_1

For $k = 1,2, \dots, K$

- Let μ_k be the state-action distribution of π_k
- Define loss function:

$$\mathcal{G}_k(\vartheta) = \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta \delta_\vartheta(x,a)}\right] + (1-\gamma)\langle \nu_0, V_\vartheta \rangle$$

- Policy evaluation:

$$\vartheta_k = \arg\min_{\vartheta} \mathcal{G}_k(\vartheta)$$

- Policy update:

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) \exp\left(\eta \delta_{\vartheta_k}(x,a)\right)$$

Definitions

Value-function approximation:

$$V_\vartheta(x) = \langle \vartheta, \psi(x) \rangle$$

Bellman error:

$$\delta_\vartheta(x,a) = r(x,a) + \gamma P_{x,a} V_\vartheta - V_\vartheta(x)$$

RELATIVE ENTROPY POLICY SEARCH

REPS

Parameters: learning rate η, feature map $\psi: \mathcal{X} \to \mathbb{R}^m$

Initialization: policy π_1

For $k = 1,2, \dots, K$

- Let μ_k be the state-action distribution of π_k
- Define loss function:

$$\mathcal{G}_k(\vartheta) = \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta\delta_\vartheta(x,a)}\right] + (1-\gamma)\langle \nu_0, V_\vartheta \rangle$$

- Policy evaluation:

$$\vartheta_k = \arg\min_\vartheta \mathcal{G}_k(\vartheta)$$

- Policy update:

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) \exp\left(\eta \delta_{\vartheta_k}(x,a)\right)$$

Definitions

Value-function approximation:

$$V_\vartheta(x) = \langle \vartheta, \psi(x) \rangle$$

Bellman error:

$$\delta_\vartheta(x,a) = r(x,a) + \gamma P_{x,a} V_\vartheta - V_\vartheta(x)$$

Good news:

convex loss for policy evaluation!

RELATIVE ENTROPY POLICY SEARCH

REPS

Parameters: learning rate η, feature map $\psi: \mathcal{X} \to \mathbb{R}^m$

Initialization: policy π_1

For $k = 1,2, \dots, K$

- Let μ_k be the state-action distribution of π_k
- Define loss function:

$$\mathcal{G}_k(\vartheta) = \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta \delta_\vartheta(x,a)}\right] + (1-\gamma)\langle \nu_0, V_\vartheta \rangle$$

- Policy evaluation:

$$\vartheta_k = \arg\min_{\vartheta} \mathcal{G}_k(\vartheta)$$

- Policy update:

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) \exp\left(\eta \delta_{\vartheta_k}(x,a)\right)$$

Definitions

Value-function approximation:

$$V_\vartheta(x) = \langle \vartheta, \psi(x) \rangle$$

Bellman error:

$$\delta_\vartheta(x,a) = r(x,a) + \gamma P_{x,a} V_\vartheta - V_\vartheta(x)$$

Good news:
convex loss for policy evaluation!

Bad news:
policy update intractable :"(

THE BEST OF BOTH WORLDS?

DQN

Bad news:
no natural loss
function for policy eval

Good news:
policy directly
encoded by Q-function

REPS

Good news:
natural convex loss for
policy evaluation

Bad news:
policy update
intractable

THE BEST OF BOTH WORLDS?

DQN	Q-REPS	REPS
Bad news: no natural loss function for policy eval	**Good news:** natural convex loss for policy evaluation	**Good news:** natural convex loss for policy evaluation
Good news: policy directly encoded by Q-function	**Good news:** policy directly encoded by Q-function	**Bad news:** policy update intractable

THE BEST OF BOTH WORLDS?

DQN	Q-REPS	REPS
Bad news: no natural loss function for policy eval	**Good news:** natural convex loss for policy evaluation	**Good news:** natural convex loss for policy evaluation
Good news: policy directly encoded by Q-function	**Good news:** policy directly encoded by Q-function	**Bad news:** policy update intractable

+ convergence guarantees to optimal policy
+ guarantees on “double sampling” bias
+ practical methods for empirical policy evaluation

Q-REPS

REPS WITH Q-FUNCTIONS

Q-REPS

Parameters: learning rates η, α,
feature map $\varphi: \mathcal{X} \times \mathcal{A} \to \mathbb{R}^m$
Initialization: policy π_1
For $k = 1,2, \dots, K$

- Let μ_k be the state-action distribution of π_k
- Define loss function:

$$\mathcal{G}_k(\theta) = \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta\Delta_\theta(x,a)}\right] + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

- Policy evaluation:

$$\theta_k = \arg\min_\theta \mathcal{G}_k(\theta)$$

- Policy update:

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) \exp\left(\eta Q_{\theta_k}(x,a)\right)$$

Definitions

Q-function approximation:

$$Q_\theta(x,a) = \langle \theta, \varphi(x,a) \rangle$$

Softmax value function

$$V_\theta(x) = \frac{1}{\alpha} \log \mathbb{E}_{a\sim\pi_k(\cdot|x)}\left[e^{\alpha Q_\theta(x,a)}\right]$$

Bellman error:

$$\Delta_\theta(x,a) = r(x,a) + \gamma P_{x,a} V_\theta - Q_\theta(x,a)$$

REPS WITH Q-FUNCTIONS

Q-REPS

Parameters: learning rates η, α,
feature map $\varphi: \mathcal{X} \times \mathcal{A} \to \mathbb{R}^m$
Initialization: policy π_1
For $k = 1,2, \dots, K$

- Let μ_k be the state-action distribution of π_k
- Define loss function:

$$\mathcal{G}_k(\theta) = \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta \Delta_\theta(x,a)}\right] + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

- Policy evaluation:

$$\theta_k = \arg\min_\theta \mathcal{G}_k(\theta)$$

- Policy update:

$$\pi_{k+1}(a|x) \propto \pi_k(a|x) \exp\left(\eta Q_{\theta_k}(x,a)\right)$$

Definitions

Q-function approximation:

$$Q_\theta(x,a) = \langle \theta, \varphi(x,a) \rangle$$

Softmax value function

$$V_\theta(x) = \frac{1}{\alpha} \log \mathbb{E}_{a\sim\pi_k(\cdot|x)}\left[e^{\alpha Q_\theta(x,a)}\right]$$

Bellman error:

$$\Delta_\theta(x,a) = r(x,a) + \gamma P_{x,a} V_\theta - Q_\theta(x,a)$$

Good news:
convex loss for policy evaluation!

Good news:
tractable
policy update :")

THE NEW LOSS FUNCTION

The Logistic Bellman Error (LBE)

$$\mathcal{G}_k(\theta) = \frac{1}{\eta}\log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta\Delta_\theta(x,a)}\right] + (1-\gamma)\langle\nu_0, V_\theta\rangle$$

- Convex and smooth (composition of two monotone convex functions that are smooth)
- 2-Lipschitz w.r.t. ℓ_∞-norm:

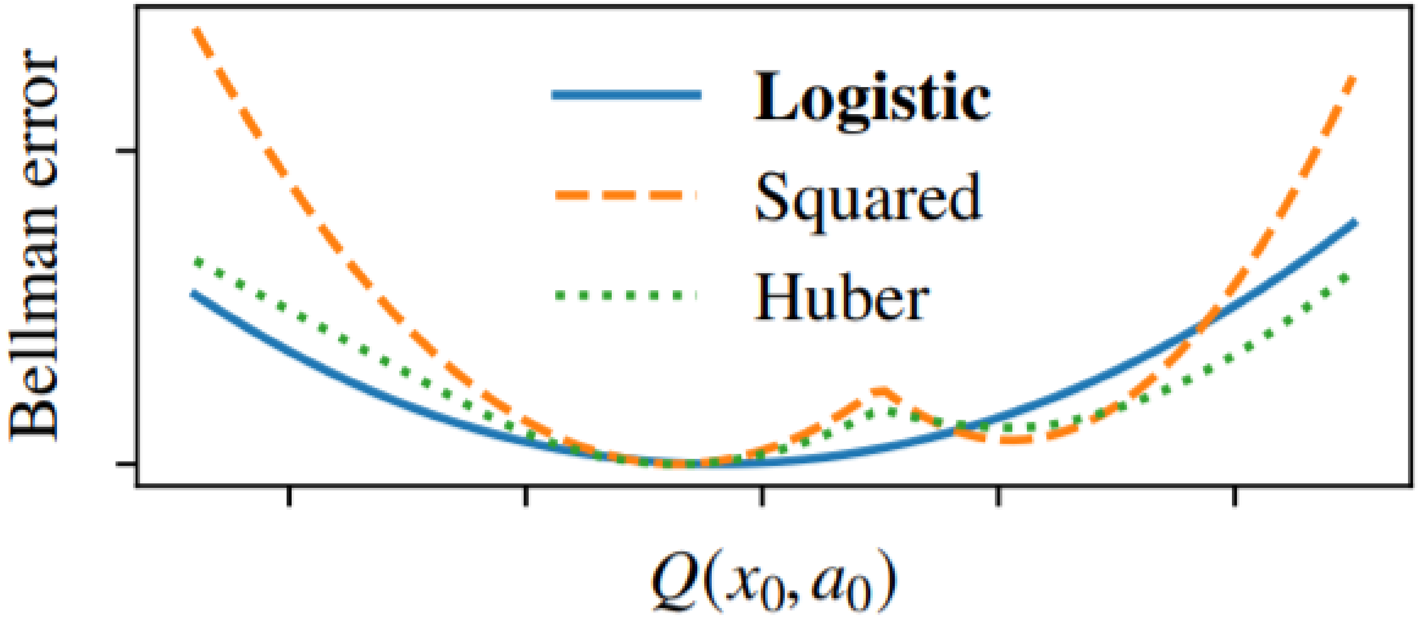
$$\left\|\nabla_Q \mathcal{G}_k(Q)\right\|_1 \leq 2$$

- Easy to estimate reliably using sample transitions

THE NEW LOSS FUNCTION

The Logistic Bellman Error (LBE)

$$\mathcal{G}_k(\theta) = \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_k}\left[e^{\eta \Delta_\theta(x,a)}\right] + (1-\gamma)\langle \nu_0, V_\theta \rangle$$



ESTIMATING THE LBE

- Define TD-error

$$\Delta_\theta(x, a, x') = r(x, a) + \gamma V_\theta(x') - Q_\theta(x, a)$$

- Let $\{(X_n, A_n, X'_n)\}_{n=1}^N$ be sample transitions from μ_k

The empirical LBE (ELBE)

$$\hat{\mathcal{G}}_k(\theta) = \frac{1}{\eta}\log\left(\frac{1}{N}\sum_{n=1}^N e^{\eta\Delta_\theta(X_n, A_n, X'_n)}\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

ESTIMATING THE LBE

- Define TD-error

$$\Delta_\theta(x,a,x') = r(x,a) + \gamma V_\theta(x') - Q_\theta(x,a)$$

- Let $\{(X_n, A_n, X_n')\}_{n=1}^N$ be sample transitions from μ_k

The empirical LBE (ELBE)

$$\hat{\mathcal{G}}_k(\theta) = \frac{1}{\eta}\log\left(\frac{1}{N}\sum_{n=1}^N e^{\eta\Delta_\theta(X_n,A_n,X_n')}\right) + (1-\gamma)\langle \nu_0, V_\theta\rangle$$

Warning!

Subject to “double sampling bias”:

$$\mathbb{E}\left[e^{\eta\Delta(X,A,X')}\right] \neq \mathbb{E}\left[e^{\eta\Delta(X,A)}\right] = \mathbb{E}\left[e^{\eta\mathbb{E}[\Delta(X,A,X')|X,A]}\right]$$

DOUBLE SAMPLING BIAS

- Question: how serious is this bias?

DOUBLE SAMPLING BIAS

- Question: how serious is this bias?
- Answer:

  not too serious!

DOUBLE SAMPLING BIAS

- Question: how serious is this bias?
- Answer:

not too serious!

Theorem

with probability $\geq 1 - \delta$,

$$\left|\mathcal{G}_k(\theta) - \hat{\mathcal{G}}_k(\theta)\right| = O\left(\eta + \sqrt{\frac{\log(1/\delta)}{N}}\right)$$

DOUBLE SAMPLING BIAS

- **Question:** how serious is this bias?
- **Answer:**

not too serious!

Theorem

with probability $\geq 1 - \delta$,

$$\left|\mathcal{G}_k(\theta) - \hat{\mathcal{G}}_k(\theta)\right| = O\left(\eta + \sqrt{\frac{\log(1/\delta)}{N}}\right)$$

Bias is controlled by η!

OPTIMIZATION ERRORS

- Practical implementations will always have optimization errors:

$$\varepsilon_k = \mathcal{G}_k(\theta_k) - \min_{\theta} \mathcal{G}_k(\theta) \geq 0$$

- Question: how do these errors accumulate?

OPTIMIZATION ERRORS

- Practical implementations will always have optimization errors:

$$\varepsilon_k = \mathcal{G}_k(\theta_k) - \min_\theta \mathcal{G}_k(\theta) \geq 0$$

- Question: how do these errors accumulate?
- Answer:

very reasonably!

ERROR PROPAGATION BOUND

Theorem

$$\frac{1}{K}\sum_{k=1}^{K}(R^* - R_k) \leq \frac{D(\mu^*|\mu_0)}{\eta K} + \frac{H(d^*|d_0)}{\alpha K} + \frac{1}{K}\sum_{k=1}^{K}\varepsilon_k + \frac{C_\gamma}{K}\left(\frac{\sqrt{\alpha}}{1-\gamma} + \sqrt{\eta}\right)\sum_{k=1}^{K}\sqrt{\varepsilon_k}$$

ERROR PROPAGATION BOUND

Theorem

$$\frac{1}{K}\sum_{k=1}^{K}(R^* - R_k) \leq \frac{D(\mu^*|\mu_0)}{\eta K} + \frac{H(d^*|d_0)}{\alpha K} + \frac{1}{K}\sum_{k=1}^{K}\varepsilon_k + \frac{C_\gamma}{K}\left(\frac{\sqrt{\alpha}}{1-\gamma} + \sqrt{\eta}\right)\sum_{k=1}^{K}\sqrt{\varepsilon_k}$$

When $\varepsilon_k = 0$, this gives a rate of $O(1/K)$

ERROR PROPAGATION BOUND

Theorem

$$\frac{1}{K}\sum_{k=1}^{K}(R^* - R_k) \leq \frac{D(\mu^*|\mu_0)}{\eta K} + \frac{H(d^*|d\cdots)}{\alpha K} + \frac{1}{K}\sum_{k=1}^{K}\varepsilon_k + \frac{C_\gamma}{K}\left(\frac{\sqrt{\alpha}}{1-\gamma} + \sqrt{\eta}\right)\sum_{k=1}^{K}\sqrt{\varepsilon_k}$$

For large enough N, we can have $\varepsilon_k = O(\eta)$, so setting $\alpha = \eta = 1/\sqrt{K}$ gives a rate of

$$O\left(\frac{1}{\eta K} + \eta\right) = O\left(\frac{1}{\sqrt{K}}\right)$$

When $\varepsilon_k = 0$, this gives a rate of $O(1/K)$

ERROR PROPAGATION BOUND

Theorem

$$\frac{1}{K}\sum_{k=1}^{K}(R^* - R_k) \leq \frac{D(\mu^*|\mu_0)}{\eta K} + \frac{H(d^*|d}{\alpha K}$$

$$+ \frac{1}{K}\sum_{k=1}^{K}\varepsilon_k$$

$$+ \frac{C_\gamma}{K}\left(\frac{\sqrt{\alpha}}{1-\gamma} + \sqrt{\eta}\right)\sum_{k=1}^{K}\sqrt{\varepsilon_k}$$

For large enough N, we can have $\varepsilon_k = O(\eta)$, so setting $\alpha = \eta = 1/\sqrt{K}$ gives a rate of $O\left(\frac{1}{\eta K} + \eta\right) = O\left(\frac{1}{\sqrt{K}}\right)$

When $\varepsilon_k = 0$, this gives a rate of $O(1/K)$

Conditions: the features need to have sufficient representation power (“factored linear MDPs”). This clearly holds for tabular MDPs and the bounds remain meaningful for very large state spaces.

WHY IS THIS A BIG DEAL?

Theorem

$$\left|\mathcal{G}_k(\theta) - \hat{\mathcal{G}}_k(\theta)\right| = O(\eta)$$

No such result possible for squared Bellman error!
(only after severe patching)

Theorem

$$\mathrm{err}_K \leq O\left(\frac{1}{K}\sum_{k=1}^{K}\left(\varepsilon_k + \sqrt{\eta\varepsilon_k}\right)\right)$$

Similar results are known for SBE, but there's no algorithms that can reliably control these errors!
(due to above reason)

MINIMIZING THE ELBE

- Minimizing the LBE can be equivalently written as

$$\min_{\theta} \frac{1}{\eta} \log\left(\frac{1}{N}\sum_{n=1}^{N} e^{\eta \Delta_\theta(X_n, A_n, X_n')}\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

$$= \min_{\theta} \max_{z \in D_N} \sum_{n=1}^{N} z_n \left(\Delta_\theta(X_n, A_n, X_n') - \frac{1}{\eta}\log(N z_n)\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

MINIMIZING THE ELBE

- Minimizing the LBE can be equivalently written as

$$\min_{\theta} \frac{1}{\eta} \log\left(\frac{1}{N}\sum_{n=1}^{N} e^{\eta\Delta_\theta(X_n, A_n, X_n')}\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

$$= \min_{\theta} \max_{z \in D_N} \sum_{n=1}^{N} z_n \left(\Delta_\theta(X_n, A_n, X_n') - \frac{1}{\eta}\log(N z_n)\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

Gradient w.r.t. θ is an expectation $\Rightarrow$ well-suited for SGD!

MINIMIZING THE ELBE

- Minimizing the LBE can be equivalently written as

$$\min_\theta \frac{1}{\eta} \log\left(\frac{1}{N}\sum_{n=1}^{N} e^{\eta \Delta_\theta(X_n, A_n, X_n')}\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

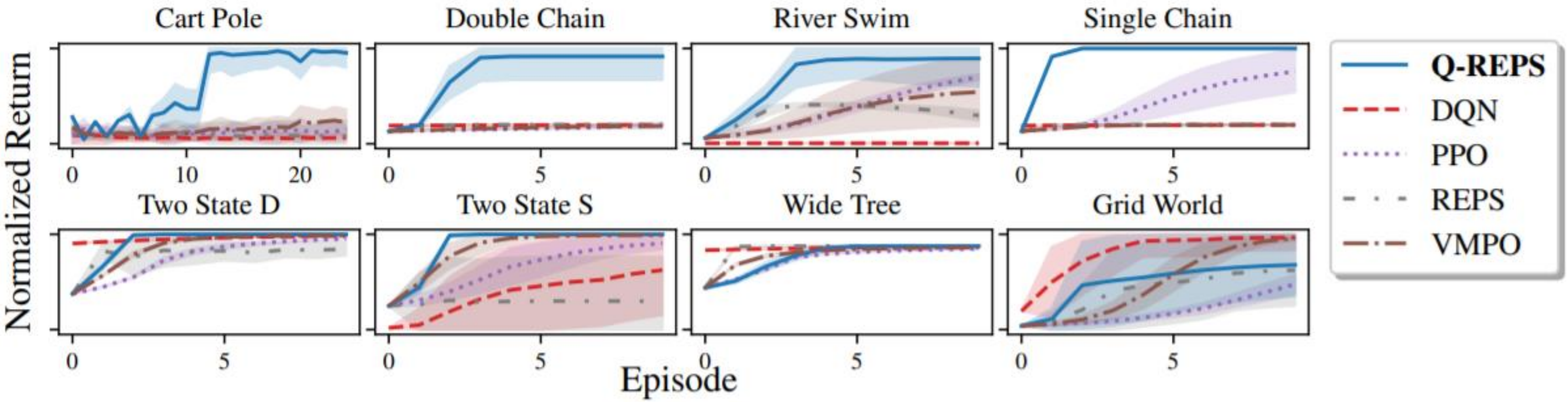
$$= \min_\theta \max_{z \in D_N} \sum_{n=1}^{N} z_n \left(\Delta_\theta(X_n, A_n, X_n') - \frac{1}{\eta}\log(N z_n)\right) + (1-\gamma)\langle \nu_0, V_\theta \rangle$$

Gradient w.r.t. θ is an expectation $\Rightarrow$ well-suited for SGD!

Implementation: two-player game between

- a learner updating θ via SGD
- a sampler updating z via exponentiated GD

AND IT WORKS!!!



Cart Pole
Double Chain
River Swim
Single Chain
Two State D
Two State S
Wide Tree
Grid World
Normalized Return
Episode
0
10
20
5
Q-REPS
DQN
PPO
REPS
VMPO

Derivation of Q-REPS

WHAT'S BEHIND Q-REPS?

- Like REPS, Q-REPS is a mirror descent algorithm:

$$z_{k+1} = \arg\max_{z \in \mathcal{S}} \{\langle z, r \rangle - R(z|z_k)\},$$

with several major differences in how $z, \mathcal{S}, R$ are defined

WHAT'S BEHIND Q-REPS?

- Like REPS, Q-REPS is a mirror descent algorithm:
$$z_{k+1} = \arg\max_{z \in \mathcal{S}} \{\langle z, r \rangle - R(z|z_k)\},$$
with several major differences in how $z, \mathcal{S}, R$ are defined
- Algorithm derived from LP formulation of optimal control in MDPs with 3 tricks:

linear relaxation + regularization + Lagrangian decomposition

WHAT'S BEHIND Q-REPS?

- Like REPS, Q-REPS is a mirror descent algorithm:
$$z_{k+1} = \arg\max_{z \in \mathcal{S}} \{\langle z, r \rangle - R(z|z_k)\},$$
with several major differences in how $z, \mathcal{S}, R$ are defined
- Algorithm derived from LP formulation of optimal control in MDPs with 3 tricks:

  linear relaxation + regularization + Lagrangian decomposition
- Analysis based on:
  - Convex analysis & Lagrangian duality
  - Ideas from the classic mirror-descent analysis
  - A bit of stability analysis for MDPs
  - Exploiting a bunch of properties of the Shannon entropy

LINEAR PROGRAMMING FOR MDPS

- Maximizing discounted return can be written as the LP

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & \sum_a \mu(x,a) = \gamma \sum_{x',a'} P(x|x',a')\mu(x',a') + (1-\gamma)\nu_0(x) \\ & \mu(x,a) \geq 0 \end{aligned}$$

LINEAR PROGRAMMING FOR MDPS

- Maximizing discounted return can be written as the LP

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & \sum_{a} \mu(x,a) = \gamma \sum_{x',a'} P(x|x',a')\mu(x',a') + (1-\gamma)\nu_0(x) \\ & \mu(x,a) \geq 0 \end{aligned}$$

"flow constraint"

LINEAR PROGRAMMING FOR MDPS

- Maximizing discounted return can be written as the LP

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & \sum_a \mu(x,a) = \gamma \sum_{x',a'} P(x|x',a')\mu(x',a') + (1-\gamma)\nu_0(x) \\ & \mu(x,a) \geq 0 \end{aligned}$$

"flow constraint"

Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\mathbb{E}_{x\sim\nu_0}[V(x)] \\ \text{subject to} \quad & V(x) \geq r(x,a) + \gamma \sum_{x'} P(x'|x,a)V(x') \end{aligned}$$

VECTOR NOTATION TO MAKE LIFE EASY

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top \mu = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & \mu \in \Delta_{\mathcal{X} \times \mathcal{A}} \end{aligned}$$

- Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, V \rangle \\ \text{subject to} \quad & EV \geq r + \gamma PV \end{aligned}$$

DERIVATION OF REPS

REPS adds two major components to this LP:

- Linear function-approximation
- Regularization

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top \mu = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, V \rangle \\ \text{subject to} \quad & EV \geq r + \gamma PV \end{aligned}$$

DERIVATION OF REPS

REPS adds two major components to this LP:

- Linear function-approximation
- Regularization

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & \Psi^\top E^\top \mu = \Psi^\top (\gamma P^\top \mu + (1-\gamma)\nu_0) \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, \Psi\vartheta \rangle \\ \text{subject to} \quad & E\Psi\vartheta \geq r + \gamma P \Psi \vartheta \end{aligned}$$

Ψ: feature matrix with rows $\psi(x) \in \mathbb{R}^m$

DERIVATION OF REPS

REPS adds two major components to this LP:

- Linear function-approximation
- Regularization

- Primal convex program:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle - D(\mu|\mu_{\text{ref}})/\eta \\ \text{subject to} \quad & \Psi^\top E^\top \mu = \Psi^\top(\gamma P^\top \mu + (1-\gamma)\nu_0) \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- Dual convex program:

$$\text{minimize} \quad (1-\gamma)\langle \nu_0, \Psi\vartheta \rangle + \frac{1}{\eta}\log \mathbb{E}_{(x,a)\sim\mu_{\text{ref}}}\left[e^{\eta\delta_\vartheta(x,a)}\right]$$

Ψ: feature matrix with rows $\psi(x) \in \mathbb{R}^m$

D: relative entropy

$$D(\mu|\mu') = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\mu'(x,a)}$$

δ_ϑ: Bellman error

$$\delta_\vartheta = r + \gamma P V_\vartheta - E V_\vartheta$$

DERIVATION OF REPS

REPS adds two major components to this LP:

- Linear function-approximation
- Regularization

- Primal convex program:

m[redacted]

su[redacted]

- Dual conve[redacted]

How do we introduce Q-functions?

minimi[redacted]ze $(1-\gamma)\langle \nu_0, \Psi\vartheta\rangle + \frac{[redacted]}{\eta}\log \mathbb{E}_{(x,a)\sim\mu_{\mathrm{ref}}}[e^{[redacted]_\vartheta(x,a)}]$

Ψ: feature matrix with rows $\psi(x) \in \mathbb{R}^m$

D: relative entropy $D(\mu|\mu') = \sum_{x,a} \mu(x,a) \log \frac{\mu(x,a)}{\mu'(x,a)}$

δ_ϑ: Bellman error $\delta_\vartheta = r + \gamma P V_\vartheta - E V_\vartheta$

Q-FUNCTIONS IN THE LP FRAMEWORK

- Lagrangian decomposition: introduce "mirror image" d of μ
- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top \mu = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, V \rangle \\ \text{subject to} \quad & EV \geq r + \gamma PV \end{aligned}$$

Q-FUNCTIONS IN THE LP FRAMEWORK

- Lagrangian decomposition: introduce "mirror image" d of μ
- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top d = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & d = \mu \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, V \rangle \\ \text{subject to} \quad & EV \geq Q \\ & Q = r + \gamma P V \end{aligned}$$

Mehta and Meyn (2009, 2020), Lee and He (2019), Neu and Pike-Burke (2020)

DERIVATION OF Q-REPS

Q-REPS adds two major components to this LP:

- Linear function-approximation
- Regularization

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top d = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & d = \mu \end{aligned}$$

Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, V \rangle \\ \text{subject to} \quad & EV \geq Q \\ & Q = r + \gamma P V \end{aligned}$$

DERIVATION OF Q-REPS

Q-REPS adds two major components to this LP:
- Linear function-approximation
- Regularization

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top d = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & \Phi^\top d = \Phi^\top \mu \end{aligned}$$

Dual LP:

$$\begin{aligned} \text{minimize} \quad & (1-\gamma)\langle \nu_0, V \rangle \\ \text{subject to} \quad & EV \geq \Phi\theta \\ & \Phi\theta \geq r + \gamma P V \end{aligned}$$

Φ: feature matrix with rows $\varphi(x,a) \in \mathbb{R}^m$

DERIVATION OF Q-REPS

Q-REPS adds two major components to this LP:
- Linear function-approximation
- Regularization

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle - D(\mu|\mu_{\text{ref}})/\eta - H(d|d_{\text{ref}})/\alpha \\ \text{subject to} \quad & E^\top d = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & \Phi^\top d = \Phi^\top \mu \end{aligned}$$

Dual LP:

$$\text{minimize}(1-\gamma)\langle \nu_0, V_\theta \rangle + \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_{\text{ref}}}\left[e^{\eta \Delta_\theta(x,a)}\right]$$

with $V_\theta(x) = \frac{1}{\alpha} \log\left(\sum_a \pi_{\text{ref}}(a|x) e^{\alpha Q_\theta(x,a)}\right)$

Φ: feature matrix with rows $\varphi(x,a) \in \mathbb{R}^m$

H: conditional entropy
$H(d|d') = \sum_{x,a} d(x,a) \log \frac{\pi_d(x,a)}{\pi_{d'}(x,a)}$

Δ_θ: Bellman error
$\Delta_\theta = r + \gamma P V_\theta - Q_\vartheta$

DERIVATION OF Q-REPS

Q-REPS adds two major components to this LP:
- Linear function-approximation
- Regularization

- Primal LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle - D(\mu|\mu_{\text{ref}})/\eta - H(d|d_{\text{ref}})/\alpha \\ \text{subject to} \quad & E^\top d = \gamma P^\top \mu + (1-\gamma)\nu_0 \\ & \Phi^\top d = \Phi^\top \mu \end{aligned}$$

Dual LP:

$$\text{minimize}(1-\gamma)\langle \nu_0, V_\theta \rangle + \frac{1}{\eta} \log \mathbb{E}_{(x,a)\sim\mu_{\text{ref}}}\left[e^{\eta \Delta_\theta(x,a)}\right]$$

with $V_\theta(x) = \frac{1}{\alpha} \log\left(\sum_a \pi_{\text{ref}}(a|x) e^{\alpha Q_\theta(x,a)}\right)$

Φ: feature matrix with rows $\varphi(x,a) \in \mathbb{R}^m$

H: conditional entropy
$H(d|d') = \sum_{x,a} d(x,a) \log \frac{\pi_d(x,a)}{\pi_{d'}(x,a)}$

Δ_θ: Bellman error
$\Delta_\theta = r + \gamma P V_\theta - Q_\vartheta$

SOME FAILED IDEAS

- Adding no regularization on d: Q-functions all collapse to V!
- Using $D(d|d_{\text{ref}})$ instead of $H(d|d_{\text{ref}})$: no closed form for V and extra terms in the objective
- Relaxing all primal constraints: leads to parametrization of V which is unnecessary due to closed-form expression
- Replacing penalty by trust-region constraint $D(\mu|\mu_k) \leq \beta$: very sensitive to noise & convergence cannot be guaranteed

SUMMARY

- REPS is awesome:
  - Principled mirror-descent algorithm
  - Convex loss function for policy eval

SUMMARY

- REPS is awesome:
  - Principled mirror-descent algorithm
  - Convex loss function for policy eval
- Q-REPS is even more awesome:
  - Q-function enables tractable policy updates!
  - Guarantees on bias & error propagation (mostly also hold for REPS too)
  - Efficient and robust implementation via two-player game perspective

SUMMARY

- REPS is awesome:
  - Principled mirror-descent algorithm
  - Convex loss function for policy eval
- Q-REPS is even more awesome:
  - Q-function enables tractable policy updates!
  - Guarantees on bias & error propagation (mostly also hold for REPS too)
  - Efficient and robust implementation via two-player game perspective
- Lots of open questions!
  - Improve theory and implementation details
  - Large-scale experiments
  - Adding exploration and dealing with constraints…

SUMMARY

- REPS is awesome:
  - Principled mirror-descent algorithm
  - Convex loss function for policy eval
- Q-REPS is even more awesome:
  - Q-function enables tractable policy updates!
  - Guarantees on bias & error propagation (mostly also hold for REPS too)
  - Efficient and robust implementation via two-player game perspective
- Lots of open questions!
  - Improve theory and implementation details
  - Large-scale experiments
  - Adding exploration and dealing with constraints…

The Logistic Bellman Error is the future!!!

THANKS!!!!

Appendix

FACTORED LINEAR MDPS

- Assume access to a feature map $\varphi: \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^d$
- Reward function can be written as $r(x, a) = \langle \varphi(x, a), \theta_r \rangle$
- Transition function can be written as

$$P(x'|x, a) = \langle \varphi(x, a), m(x') \rangle$$

for some $m(x') \in \mathbb{R}^d$

FACTORED LINEAR MDPS

- Assume access to a feature map $\varphi: \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$
- Reward function can be written as $r(x,a) = \langle \varphi(x,a), \theta_r \rangle$
- Transition function can be written as

$$P(x'|x,a) = \langle \varphi(x,a), m(x') \rangle$$

  for some $m(x') \in \mathbb{R}^d$
- In matrix form:

$$r = \Phi\theta_r, \qquad P = \Phi M,$$

$$\Phi = \begin{bmatrix} \varphi\big((x,a)_1\big) \\ \varphi\big((x,a)_2\big) \\ \vdots \\ \varphi\big((x,a)_N\big) \end{bmatrix} \qquad M = \begin{bmatrix} m(x_1) \\ m(x_2) \\ \vdots \\ m(x_K) \end{bmatrix}$$

SOME USEFUL PROPERTIES

- All action-value functions are expressible by the features:

$$Q^\pi = r + PV^\pi = \Phi\theta_r + \Phi M V^\pi = \Phi(\theta_r + MV^\pi) = \Phi\theta^\pi$$

SOME USEFUL PROPERTIES

- All action-value functions are expressible by the features:

$$Q^\pi = r + PV^\pi = \Phi\theta_r + \Phi M V^\pi = \Phi(\theta_r + MV^\pi) = \Phi\theta^\pi$$

- Plugged into the LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top d = P^\top \mu \\ & \Phi^\top d = \Phi^\top \mu \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

SOME USEFUL PROPERTIES

- All action-value functions are expressible by the features:

$$Q^\pi = r + PV^\pi = \Phi\theta_r + \Phi M V^\pi = \Phi(\theta_r + MV^\pi) = \Phi\theta^\pi$$

- Plugged into the LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top d = P^\top \mu \\ & \Phi^\top d = \Phi^\top \mu \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- If P is linear, all feasible d's are stationary:

$$E^\top d = P^\top \mu = M^\top \Phi^\top \mu = M^\top \Phi^\top d = P^\top d$$

and $\langle d, r \rangle = \langle d, \Phi\theta_r \rangle = \langle \Phi^\top d, \theta_r \rangle = \langle \Phi^\top \mu, \theta_r \rangle = \langle \mu, \Phi\theta_r \rangle = \langle \mu, r \rangle$

SOME USEFUL PROPERTIES

Dual realizability

- All action-value functions are expressible by the features:

$$Q^\pi = r + PV^\pi = \Phi\theta_r + \Phi M V^\pi = \Phi(\theta_r + MV^\pi) = \Phi\theta^\pi$$

- Plugged into the LP:

$$\begin{aligned} \text{maximize} \quad & \langle \mu, r \rangle \\ \text{subject to} \quad & E^\top d = P^\top \mu \\ & \Phi^\top d = \Phi^\top \mu \\ & \mu \in \Delta_{\mathcal{X}\times\mathcal{A}} \end{aligned}$$

- If P is linear, all feasible d's are stationary:

$$E^\top d = P^\top \mu = M^\top \Phi^\top \mu = M^\top \Phi^\top d = P^\top d$$

Primal realizability