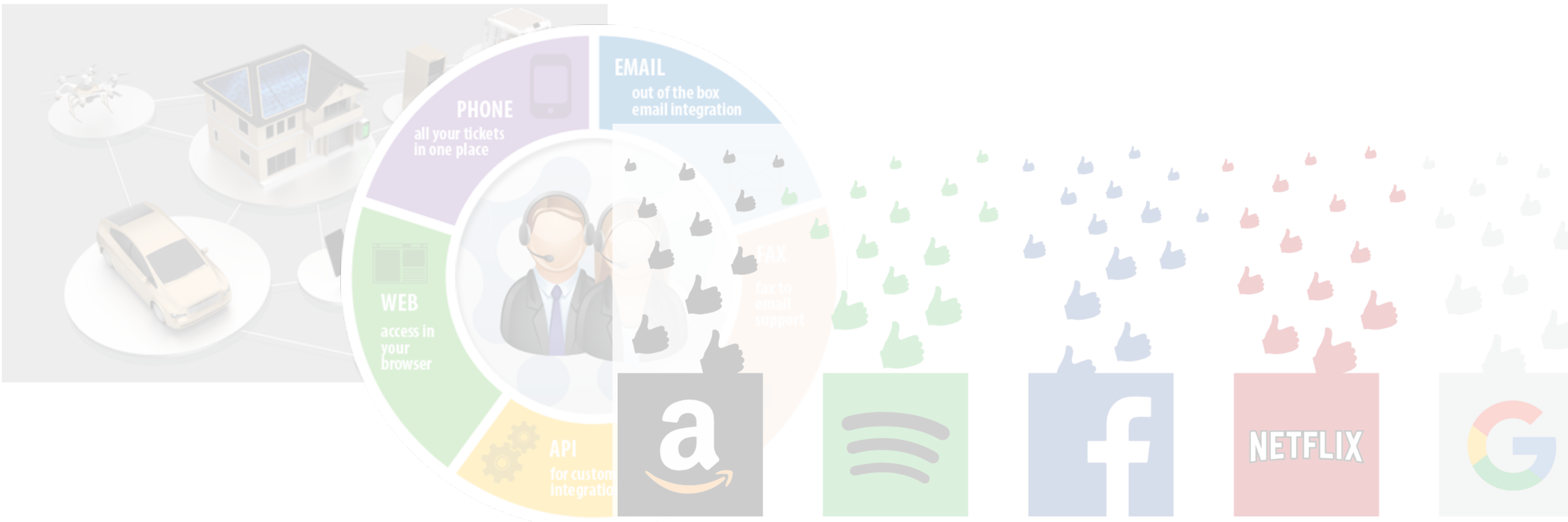
Batch Policy Learning under Constraints

Hoang M. Le **Cameron Voloshin** **Yisong Yue**

California Institute of Technology

Learning from off-line, off-policy data



π_D generates historical (sub-optimal) data

- Learn better policy from data under multiple constraints?
- Learn policy under new constraints?

(Setting: MDP, no exploration)

Given: n tuples data set $\mathrm{D} = \{(\text{state}, \text{action}, \text{next state}, \text{cost})\} \sim \pi_{\mathrm{D}}$

Goal: find π

$$\min_{\pi} \quad C(\pi)$$
$$\text{s.t.} \quad G(\pi) \leq 0$$

m constraints (vector-valued in $\mathbb{R}^m$)

$$C(\pi) = \mathbb{E}\left[\sum c(\text{state}, \text{action})\right]$$

$$G(\pi) = \mathbb{E}\left[\sum g(\text{state}, \text{action})\right] \qquad g = \begin{bmatrix} g_1 & g_2 & \dots & g_m \end{bmatrix}^\top$$

Given: n tuples data set $\mathrm{D} = \left\{ \left(\text{state}, \text{action}, \text{next state}, c, g \right) \right\} \sim \pi_{\mathrm{D}}$

Goal: find π

$$\begin{aligned} \min_{\pi} \quad & C(\pi) \\ \text{s.t.} \quad & G(\pi) \le 0 \end{aligned}$$

m constraints (vector-valued in $\mathbb{R}^m$)

$$C(\pi) = \mathbb{E}\left[\sum c(\text{state}, \text{action}) \right]$$

$$G(\pi) = \mathbb{E}\left[\sum g(\text{state}, \text{action}) \right] \qquad g = \begin{bmatrix} g_1 & g_2 & \dots & g_m \end{bmatrix}^\top$$

Given: n tuples data set $\mathrm{D} = \left\{ \left(\text{state}, \text{action}, \text{next state}, c, g \right) \right\} \sim \pi_{\mathrm{D}}$

Goal: find π

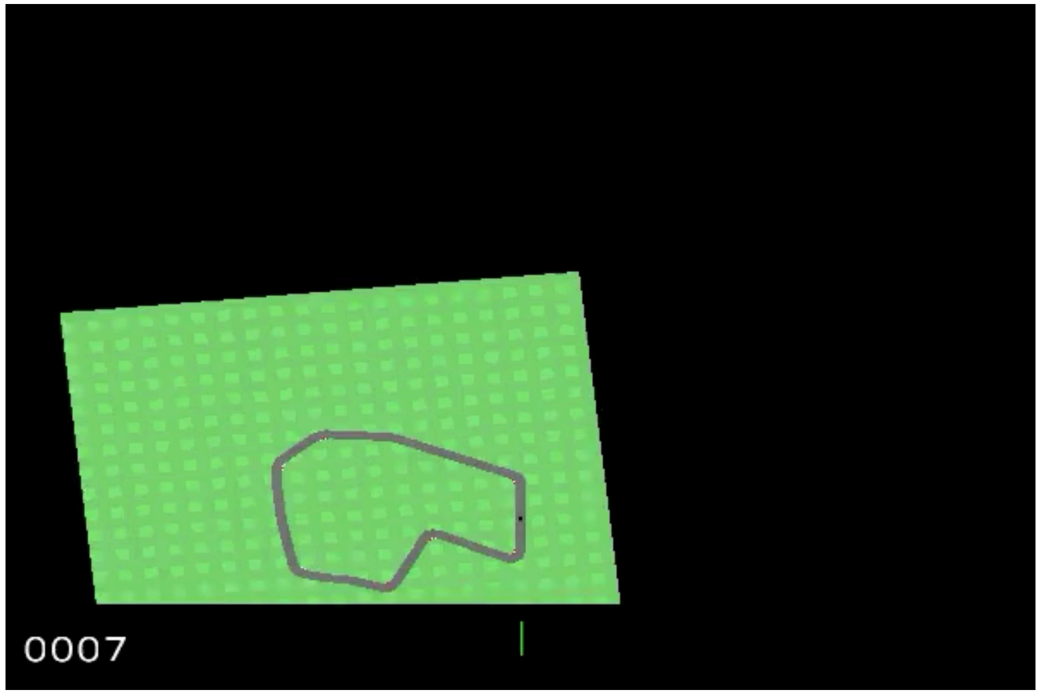
$$\min_{\pi} \quad C(\pi)$$
$$\text{s.t.} \quad G(\pi) \leq 0$$

Examples:

Counterfactual & Safe policy learning $g(x) = \mathbf{1}\left[x = x_{avoid}\right]$

Multi-criteria value-based constraints

$$\min_{\pi} \quad \text{travel time}$$
$$\text{s.t. lane centering}$$
$$\text{smooth driving}$$



Lagrangian $$L(\pi, \lambda) = C(\pi) + \lambda^\top G(\pi)$$

(P) $$\min_{\pi} \max_{\lambda \geq 0} L(\pi, \lambda)$$

(D) $$\max_{\lambda \geq 0} \min_{\pi} L(\pi, \lambda)$$

Proposed Approach:

Multiple reductions to supervised learning and online learning

Lagrangian $$L(\pi, \lambda) = C(\pi) + \lambda^\top G(\pi)$$

(P) $$\min_{\pi} \max_{\lambda \geq 0} L(\pi, \lambda)$$

(D) $$\max_{\lambda \geq 0} \min_{\pi} L(\pi, \lambda)$$

Algorithm (rough sketch)

Iteratively:

1: $\pi \leftarrow$ Best-response(λ) → off-line RL w.r.t. $c + \lambda^\top g$

Lagrangian $L(\pi, \lambda) = C(\pi) + \lambda^{\top} G(\pi)$

(P) $\min_{\pi} \max_{\lambda \geq 0} L(\pi, \lambda)$

(D) $\max_{\lambda \geq 0} \min_{\pi} L(\pi, \lambda)$

Algorithm (rough sketch)

Iteratively:

1: $\pi \leftarrow$ Best-response(λ)

2: L_{max} = evaluate (D) fixing π

3: L_{min} = evaluate (P) fixing λ

Lagrangian $$L(\pi, \lambda) = C(\pi) + \lambda^\top G(\pi)$$

(P) $$\min_{\pi} \max_{\lambda \geq 0} L(\pi, \lambda)$$

(D) $$\max_{\lambda \geq 0} \min_{\pi} L(\pi, \lambda)$$

Algorithm (rough sketch)

Iteratively:

1: $\pi \leftarrow$ Best-response(λ)

2: L_{max} = evaluate (D) fixing π

3: L_{min} = evaluate (P) fixing λ

4: if $L_{max} - L_{min} \leq \omega$:

5: stop

Lagrangian $$L(\pi, \lambda) = C(\pi) + \lambda^\top G(\pi)$$

(P) $$\min_{\pi} \max_{\lambda \geq 0} L(\pi, \lambda)$$

(D) $$\max_{\lambda \geq 0} \min_{\pi} L(\pi, \lambda)$$

Algorithm (rough sketch)

Iteratively:

1: $\pi \leftarrow$ Best-response(λ)

2: L_{max} = evaluate (D) fixing π

3: L_{min} = evaluate (P) fixing λ

4: if $L_{max} - L_{min} \leq \omega$:

5: stop

6: new $\lambda \leftarrow$ Online-algorithm(all previous π)

Regret $= O(\sqrt{T}) \implies$ convergence in $O(\frac{1}{\omega^2})$ iterations

Lagrangian $L(\pi, \lambda) = C(\pi) + \lambda^\top G(\pi)$

(P) $\min_{\pi} \max_{\lambda \geq 0} L(\pi, \lambda)$

(D) $\max_{\lambda \geq 0} \min_{\pi} L(\pi, \lambda)$

Algorithm (rough sketch)

Iteratively:

1: $\pi \leftarrow$ Best-response(λ)

2: L_{max} = evaluate (D) fixing π

3: L_{min} = evaluate (P) fixing λ

4: if $L_{max} - L_{min} \leq \omega$:

5: stop

6: new $\lambda \leftarrow$ Online-algorithm(all previous π)

$\lambda \leftarrow \lambda - \eta \widehat{G}(\pi)$

update λ based on amount of constraint violation

Regret $= O(\sqrt{T}) \implies$ convergence in $O(\frac{1}{\omega^2})$ iterations

Off-policy evaluation

Given $\mathrm{D} = \left\{ \left(\text{state}, \text{action}, \text{next state}, g \right) \right\} \sim \pi_{\mathrm{D}}$ estimate $\widehat{G}(\pi) \approx G(\pi)$

Off-policy evaluation

Given $\mathrm{D} = \left\{ \left(\text{state}, \text{action}, \text{next state}, g \right) \right\} \sim \pi_{\mathrm{D}}$ estimate $\widehat{G}(\pi) \approx G(\pi)$

New approach: model-free function approximation

Fitted Q Evaluation (simplified)

For K iterations:

1: Solve for $Q : (\text{state}, \text{action}) \mapsto y = g + Q_{prev}\left(\text{next state}, \pi(\text{next state})\right)$

2: $Q_{prev} \leftarrow Q$

Return value of Q_K

Off-policy evaluation

Given $\mathrm{D} = \{(\text{state}, \text{action}, \text{next state}, g)\} \sim \pi_{\mathrm{D}}$ estimate $\widehat{G}(\pi) \approx G(\pi)$

New approach: model-free function approximation

Fitted Q Evaluation (simplified)

For K iterations:
1: Solve for $Q : (\text{state}, \text{action}) \mapsto y = g + Q_{prev}(\text{next state}, \pi(\text{next state}))$
2: $Q_{prev} \leftarrow Q$
Return value of Q_K

Guarantee for FQE

For $n = poly(\frac{1}{\epsilon}, \log\frac{1}{\delta}, \log K, \log m, \mathtt{dim}_{\mathrm{F}})$, *with probability* $1-\delta$:

$$|G(\pi) - \widehat{G}(\pi)| \leq O(\sqrt{\beta}\epsilon)$$

distribution shift coefficient of MDP

End-to-end Performance Guarantee

For $n = \mathit{poly}(\frac{1}{\epsilon}, \log\frac{1}{\delta}, \log K, \log m, \mathtt{dim}_{\mathrm{F}})$, *with probability* $1-\delta$:

$$\text{C(returned policy)} - \text{C(optimal)} \leq O(\omega + \sqrt{\beta}\epsilon)$$

and

$$\text{constraint violation} \leq O(\omega + \sqrt{\beta}\epsilon)$$

stopping condition

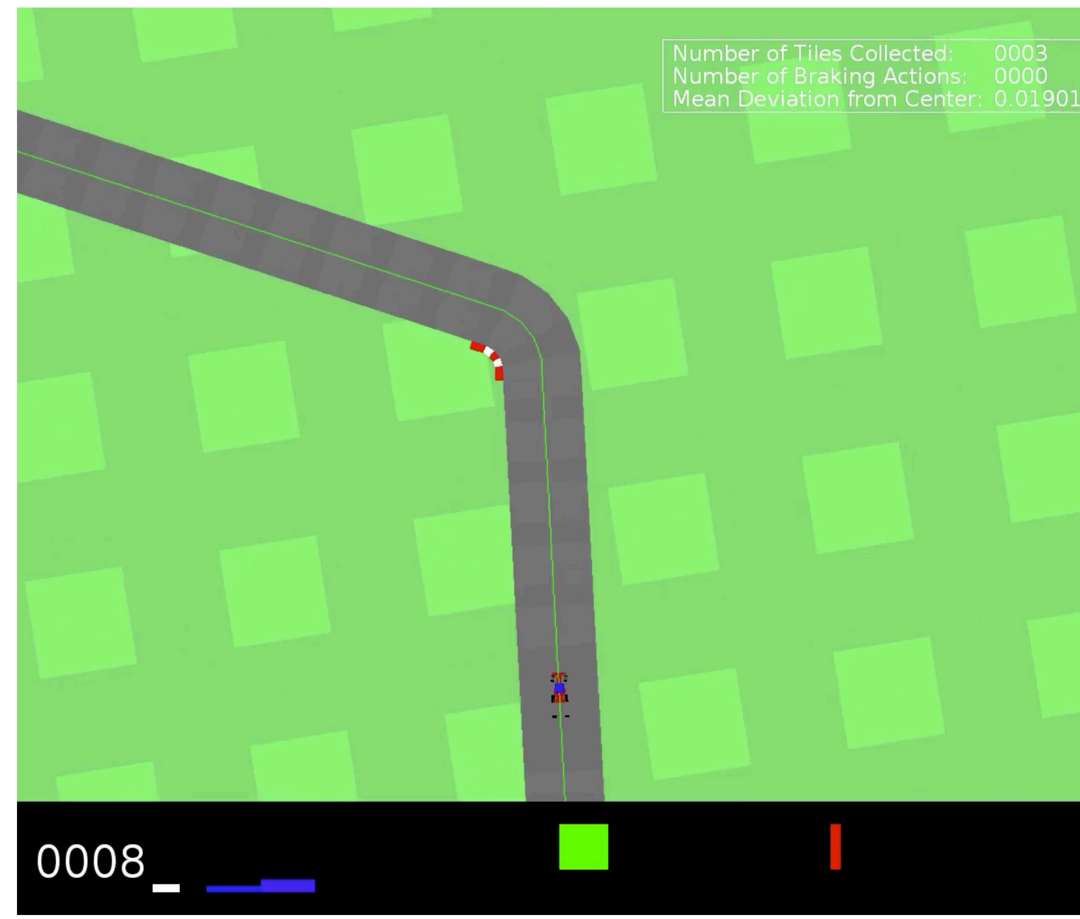
minimize travel time

s.t.

$$\text{smooth driving cost} \leq \frac{1}{2} \text{ online RL optimal (w/o constraint)}$$

$$\text{distance to lane center} \leq \frac{1}{2} \text{ online RL optimal (w/o constraint)}$$

π_{D}



returned policy

Results:

- both constraints satisfied
- travel time still matches online RL optimal

More details in the paper...

- Value-based constraint specification: Flexible to encode domain knowledge

- Data efficiency from off-line policy learning and counterfactual cost function modification