Maximum Margin Planning

Nathan Ratliff, Drew Bagnell and Martin Zinkevich

Presenters:

Ashesh Jain, Michael Hu

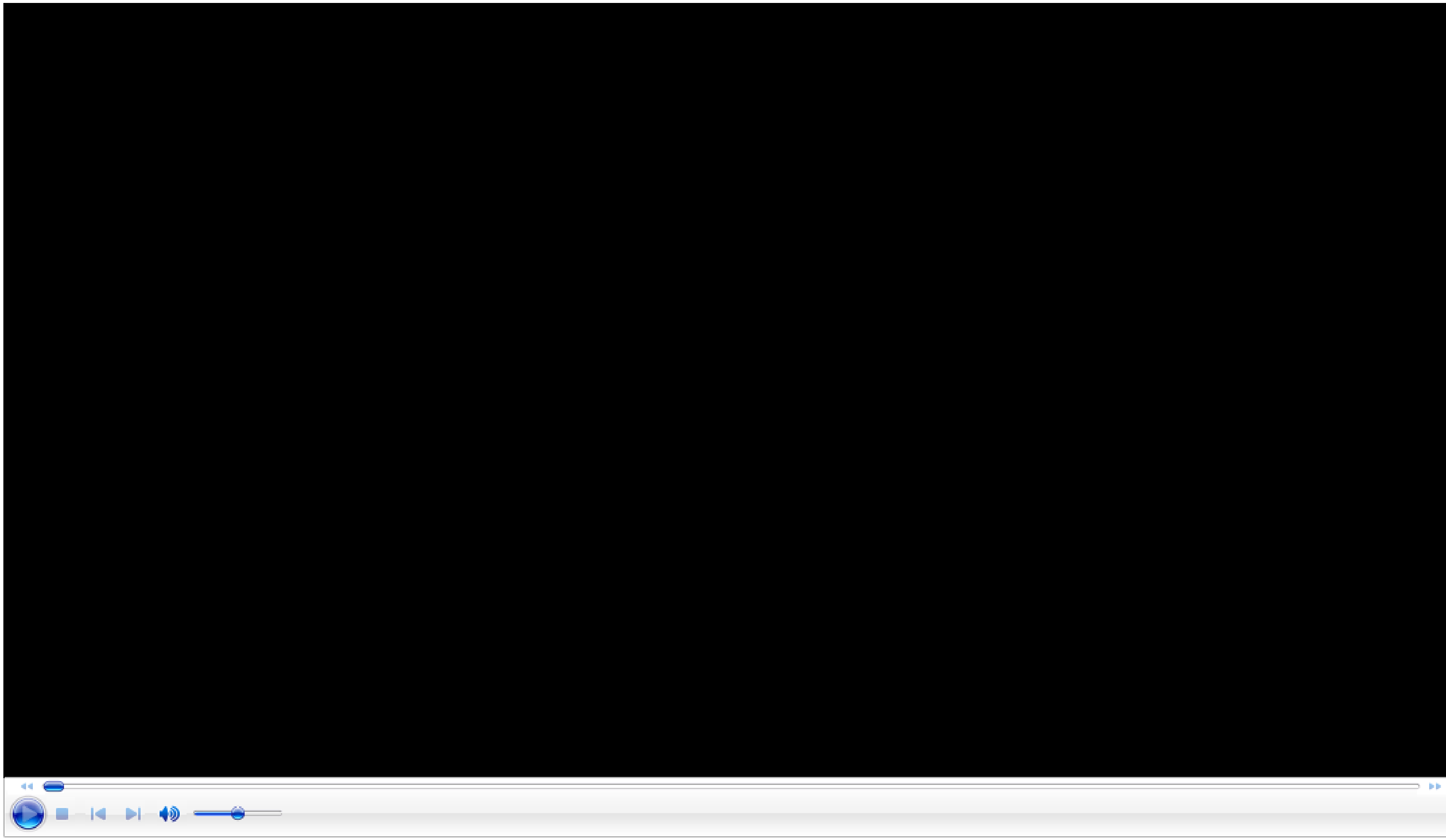
CS6784 Class Presentation

Theme

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning
    - Reward based learning

Inverse Reinforcement Learning

Motivating Example (Abbeel et. al)



Outline

Basics of Reinforcement Learning

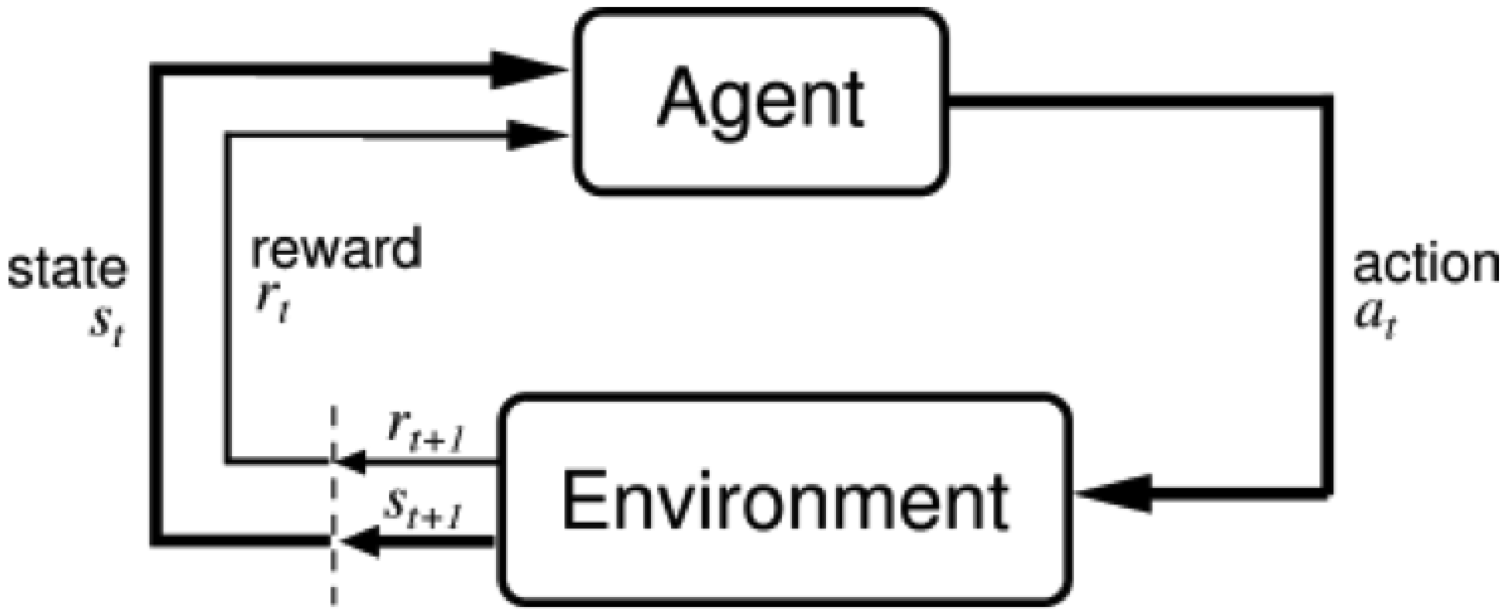
– State and action space
– Activity

Introducing Inverse RL

– Max-margin formulation
– Optimization algorithms
– Activity

Conclusion

Reward-based Learning



Credit: Sutton and Barto, Reinforcement Learning: An Introduction, 1998

Markov Decision Process (MDP)

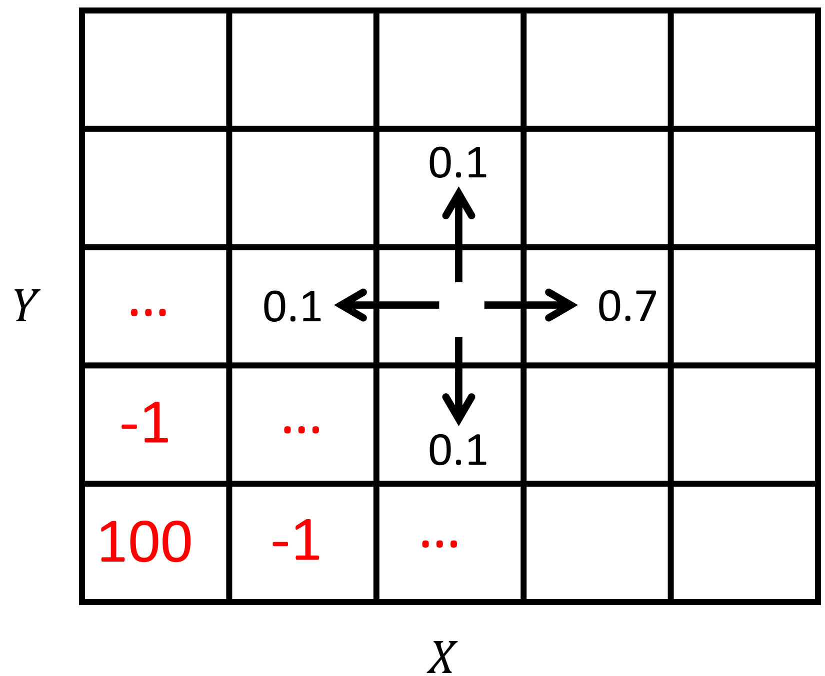
MDP is defined by $(\mathcal{X}, \mathcal{A}, p, R)$

$\mathcal{X}$ State space

$\mathcal{A}$ Action space

p Dynamics model $p(x'|x, a)$

R Reward function $R(x)$ or $R(x, a)$

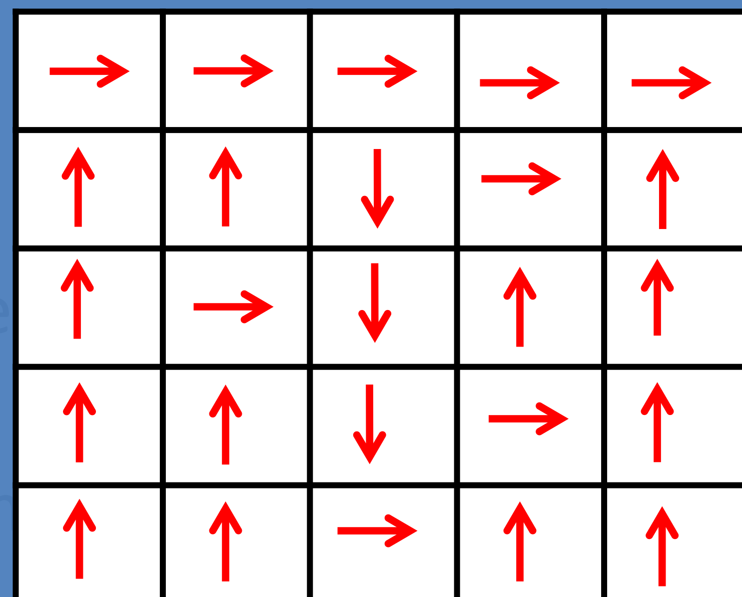


Jain, Hu

Markov Decision Process (MDP)

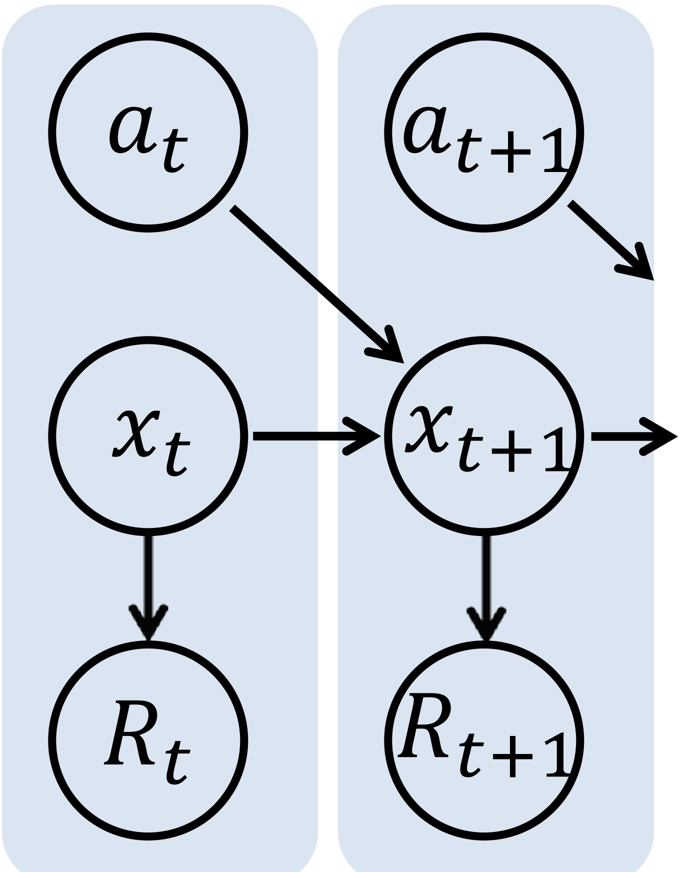
Policy

$$\pi: \mathcal{X} \rightarrow \mathcal{A}$$



Example policy

Graphical representation of MDP



Goal: Learn an optimal policy

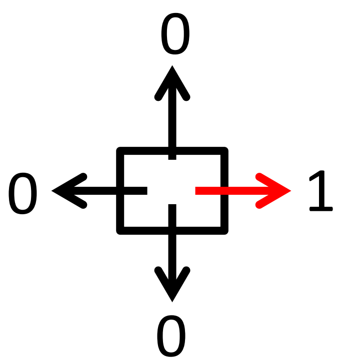
$$\pi: \mathcal{X} \rightarrow \mathcal{A}$$

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t R_t(x_t, \pi(x_t))\right]$$

$$(x_t, \pi(x_t)) \rightarrow (x_{t+1}, \pi(x_{t+1})) \rightarrow \cdots$$

Activity – Very Easy!!!

Deterministic Dynamics



-1	-1	-1	-1	100
-1	-1	-1	-1	-1
-1	-1	-50	-1	-1
-1	-1	-1	-1	-80
-1	-1	-1	-1	-1

Draw the optimal Policy!

Activity – Very Easy!!!

-1	-1	-1	-1	 100
-1	-1	-1	-1	-1
-1	-1	 -50	-1	-1
-1	-1	-1	-1	 -80
-1	-1	-1	-1	-1

Activity – Solution

→ -1	→ -1	→ -1	→ -1	 100
↑ -1	↑ -1	↑ -1	↑ -1	↑ -1
↑ -1	↑ -1	 -50	↑ -1	↑ -1
↑ -1	→ -1	→ -1	↑ -1	 -80
↑ -1	→ -1	→ -1	↑ -1	← -1

Outline

Basics of Reinforcement Learning

- State and action space
- Activity

Introducing Inverse RL

- Max-margin formulation
- Optimization algorithms
- Activities

Conclusion

RL to Inverse-RL (IRL)

- State space: $\mathcal{X}$
- Action space: $\mathcal{A}$

Optimal Policy π^*

$\pi : x \rightarrow a$

Reward Function

$\mathrm{R}(x, a)$

Inverse RL Algorithm

Dynamics Model

$p(x' | x, a)$

Reward Function

$\mathrm{R}(x, a)$

Optimal Policy π^*

$\pi : x \rightarrow a$

Inverse problem: Given optimal policy π^* or samples drawn from it, recover the reward $\mathrm{R}(x, a)$

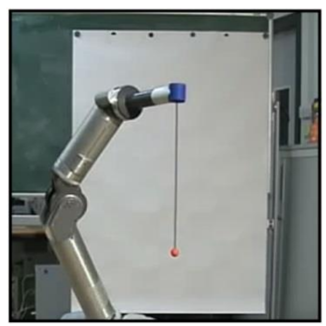
Why Inverse-RL?

- State space: $\mathcal{X}$
- Action space: $\mathcal{A}$

Reward Function $\mathrm{R}(x,a)$ → RL Algorithm

Specifying $\mathrm{R}(x,a)$ is hard, but samples from π^* are easily available

Ratliff et. al.



Kober et. al.

Abbeel et. al.

IRL framework

$\pi^*: x \to a$

Expert

Interacts

$$y^* = (x_1, a_1) \to (x_2, a_2) \to (x_3, a_3) \to \cdots \to (x_n, a_n)$$

Demonstration

$$w^T f(y^*) = w^T \quad + w^T \quad + w^T \quad + \cdots\cdots + w^T$$

Reward

Jain, Hu

Paper contributions

1. Formulated IRL as structural prediction
   - Maximum margin approach

2. Two optimization algorithms
   - Problem specific solver
   - Sub-gradient method

3. Robotic application: 2D navigation etc.

Outline

Basics of Reinforcement Learning

- State and action space
- Activity

Introducing Inverse RL

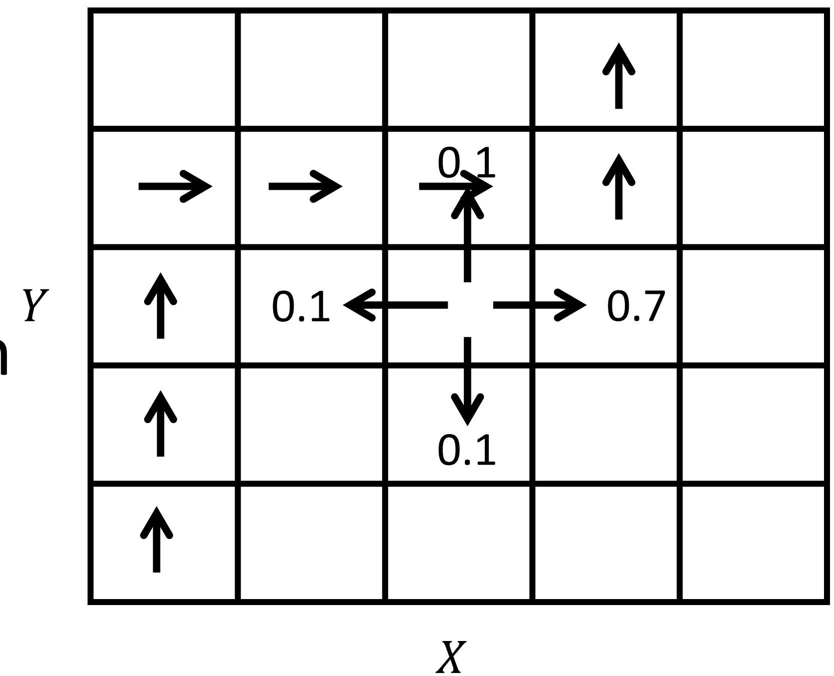
- Max-margin formulation
- Optimization algorithms
- Activities

Conclusion

Formulation

Input $\{(\mathcal{X}_i, \mathcal{A}_i, p_i, f_i, y_i, \mathcal{L}_i)\}_{i=1}^{n}$

$\mathcal{X}_i$	State space
$\mathcal{A}_i$	Action space
p_i	Dynamics model
y_i	Expert demonstration
$f_i(\cdot)$	Feature function
$\mathcal{L}_i(\cdot, y_i)$	Loss function



Max-margin formulation

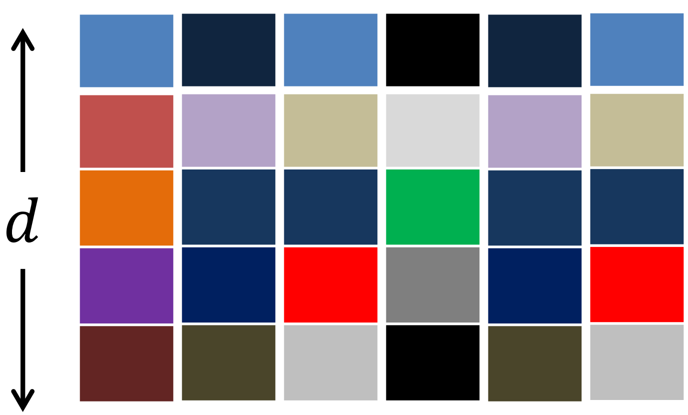
$$\min_{w,\xi_i} \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_i \beta_i \xi_i$$

$$s.t.\ \forall i \quad w^T f_i(y_i) \geq \max_{y \in \mathcal{Y}_i} w^T f_i(y) + \mathcal{L}(y, y_i) - \xi_i$$

Features linearly decompose over path $f_i(y) = F_i \mu$

$F_i = d$

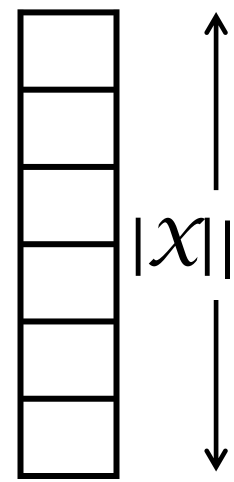
Feature map



$|\mathcal{X}||\mathcal{A}|$

$\mu \quad =$

Frequency counts



$|\mathcal{X}||\mathcal{A}|$

Jain, Hu

Max-margin formulation

$$\min_{w,\xi_i} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_i \beta_i \xi_i$$

$$s.t.\ \forall i \quad w^T F_i \mu_i \geq \max_{\mu \in \mathcal{G}_i} w^T F_i \mu + l_i^T \mu - \xi_i$$

μ satisfies Bellman-flow constraints

$$\forall x' \quad \underbrace{\sum_{x,a} \mu^{x,a} p_i(x'|x,a) + s_i^{x'}}_{\text{Inflow}} = \underbrace{\sum_a \mu^{x',a}}_{\text{Outflow}}$$

Outline

- Basics of Reinforcement Learning
  - State and action space
  - Activity
- Introducing Inverse RL
  - Max-margin formulation
  - Optimization algorithms
    - Problem specific QP-formulation
    - Sub-gradient method
  - Activities
- Conclusion

Problem specific QP-formulation

$$\min_{w,\xi_i} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_i \beta_i \xi_i$$

$$s.t.\ \forall i \quad w^T F_i \mu_i \geq \max_{\mu \in \mathcal{G}_i} w^T F_i \mu + l_i^T \mu - \xi_i$$

$$\forall i, x' \quad \underbrace{\sum_{x,a} p_i(x'|x,a)\,\mu_i^{x,a} + s_i^{x'}}_{\text{Inflow}} = \underbrace{\sum_a \mu_i^{x',a}}_{\text{Outflow}}$$

Can be optimized using off-the-shelf QP solvers

Outline

Basics of Reinforcement Learning

- State and action space
- Activity

Introducing Inverse RL

- Max-margin formulation
- Optimization algorithms
  - Problem specific QP-formulation
  - Sub-gradient method
- Activities

Conclusion

Optimizing via Sub-gradient

$$\min_{w,\xi_i} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_i \beta_i \xi_i$$

$$s.t.\ \forall i \quad w^T F_i \mu_i \geq \max_{\mu \in \mathcal{G}_i} w^T F_i \mu + l_i^T \mu - \xi_i$$

Re-writing constraint $\qquad \xi_i \geq \max_{\mu \in \mathcal{G}_i}\left(w^T F_i \mu + l_i^T \mu\right) - w^T F_i \mu_i \geq 0$

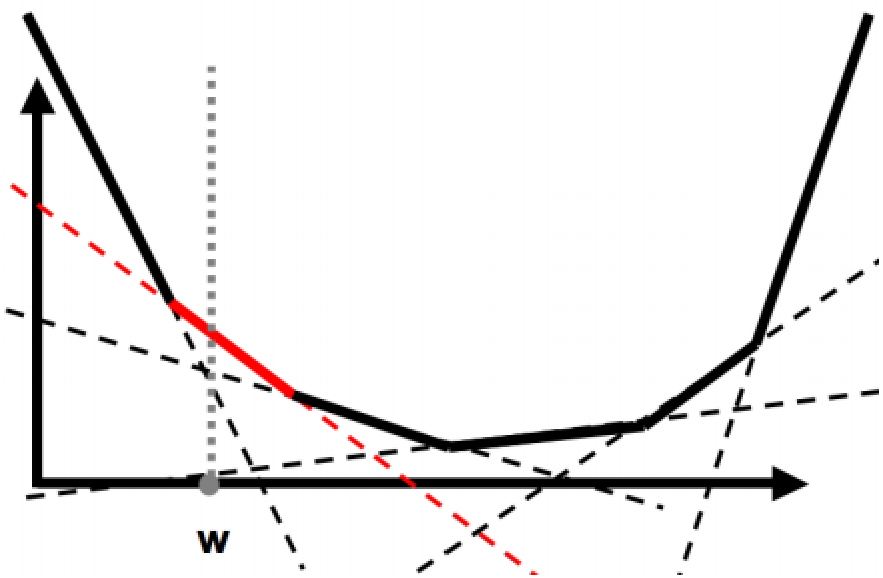
Its tight at optimality $\qquad \xi_i = \max_{\mu \in \mathcal{G}_i}\left(w^T F_i \mu + l_i^T \mu\right) - w^T F_i \mu_i$

μ satisfies Bellman-flow constraints

Jain, Hu

Optimizing via Sub-gradient

$$c(w) = \min_{w,\xi_i} \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_i \beta_i \left[\max_{\mu \in \mathcal{G}_i}\left(w^T F_i \mu + l_i^T \mu\right) - w^T F_i \mu_i\right]$$



Ratliff, PhD Thesis

Weight update via Sub-gradient

Standard gradient descent update

$$w_{t+1} \leftarrow w_t - \alpha \boxed{\nabla_{w_t} c(w)}$$

$$\min_{w,\xi_i} \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_i \beta_i \left[\max_{\mu \in \mathcal{G}_i} \left(w^T F_i \mu + l_i^T \mu \right) - w^T F_i \mu_i \right]$$

$$\mu^* = \arg\max_{\mu \in \mathcal{G}_i} \left(\underbrace{w_t^T F_i + l_i^T}_{\text{Loss-augmented reward}} \right) \mu$$

$$g_{w_t} = \lambda w_t + \frac{1}{n} \sum_i \beta_i F_i (\mu^* - \mu_i)$$

Convergence summary

$$c(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} r_i(w)$$

$$w \leftarrow w - \alpha g_w$$

How to chose α?

If $\forall w \;\; \|g_w\| \leq G$ then a constant step size $0 < \alpha \leq \frac{1}{\lambda}$ gives linear convergence to a region around the minimum

$$\|w_{t+1} - w^*\|^2 \leq \kappa^{t+1}\|w_0 - w^*\|^2 + \frac{\alpha G^2}{\lambda}$$

Optimization error

$$0 < \kappa < 1$$

Outline

Basics of Reinforcement Learning

- State and action space
- Activity

Introducing Inverse RL

- Max-margin formulation
- Optimization algorithms
- Activity

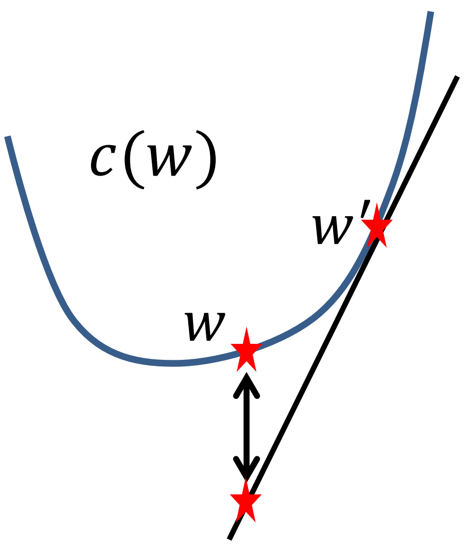
Conclusion

Convergence proof

We know $\forall w \;\; \|g_w\| \leq G$

$$\|w_{t+1} - w^*\|^2 = \|w_t - \alpha g_t - w^*\|^2$$

$$= \|w_t - w^*\|^2 + \alpha^2 \|g_t\|^2 - \boxed{?}$$



$$c(w) \geq c(w') + g_{w'}^T (w - w') + \underline{\frac{\lambda}{2} \|w - w'\|^2}$$

Substitute $w \leftarrow w_t \quad w' \leftarrow w^*$

$$\frac{\lambda}{2} \|w_t - w^*\|^2 \leq g_t^T (w_t - w^*)$$

Use $c(w^*) \leq c(w_t)$

$$\leq (1 - \alpha\lambda) \|w_t - w^*\|^2 + \alpha^2 G^2$$

Outline

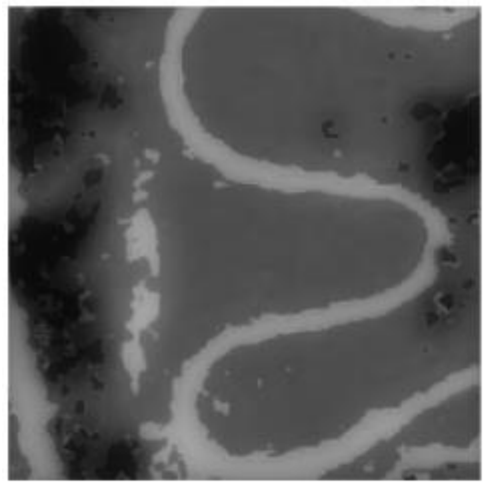
Basics of Reinforcement Learning
- State and action space
- Activity

Introducing Inverse RL
- Max-margin formulation
- Optimization algorithms
- Activity

Conclusion

2D path planning

Expert's
Demonstration

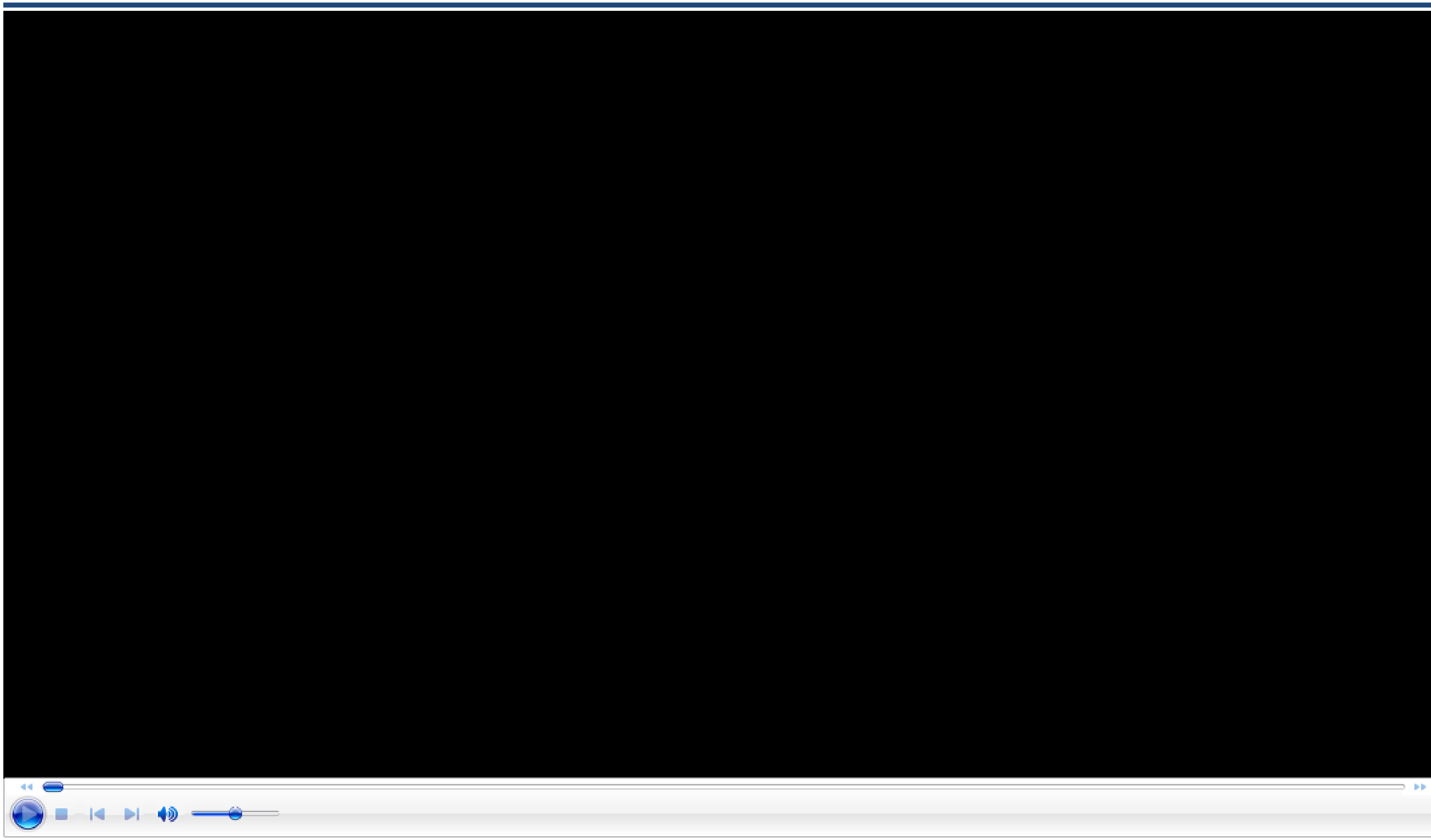


Learned
Reward

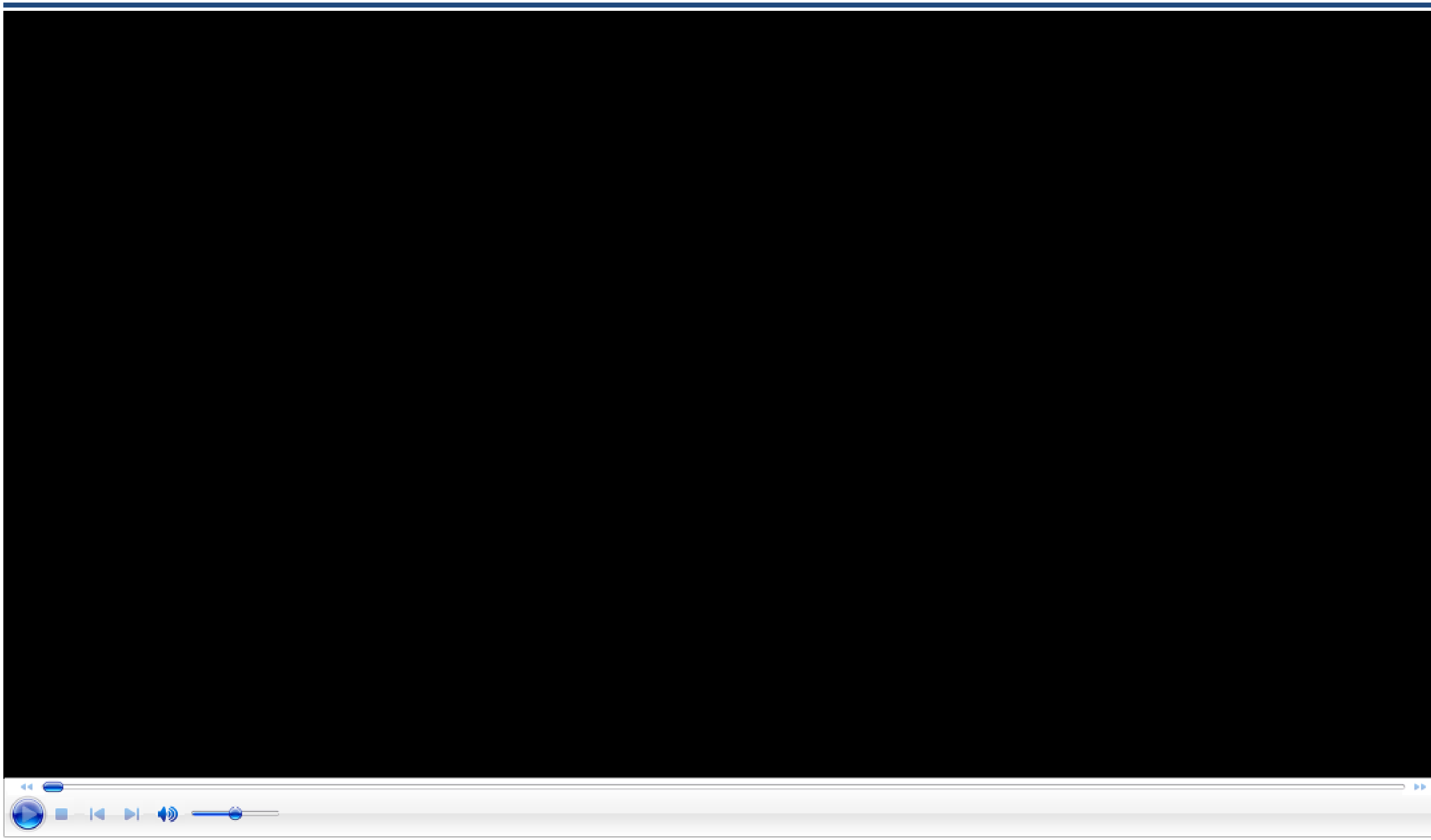
Planning in a new
Environment

Jain, Hu

Car driving (Abbeel et. al)



Bad driver (Abbeel et. al)



Recent works

1. N. Ratliff, D. Bradley, D. Bagnell, J. Chestnutt. *Boosting Structures Prediction for Imitation Learning*. In NIPS 2007
2. B. Ziebart, A. Mass, D. Bagnell, A. K. Dey. *Maximum Entropy Inverse Reinforcement Learning*. In AAAI 2010
3. P. Abbeel, A. Coats, A. Ng. *Autonomous helicopter aerobatics through apprenticeship learning*. In IJRR 2010
4. S. Ross, G. Gordon, D. Bagnell. *A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning*. In AISTATS 2011
5. B. Akgun, M. Cakmak, K. Jiang, A. Thomaz. *Keyframe-based leaning from demonstrations*. In IJSR 2012
6. A. Jain, B. Wojcik, T. Joachims, A. Saxena. *Learning Trajectory Preferences for Manipulators via Iterative Improvement*. In NIPS 2013

Questions

- Summary of key points:
  - Reward-based learning can be modeled by the MDP framework.
  - Inverse Reinforcement Learning takes perception features as input and outputs a reward function.
  - The max-margin planning problem can be formulated as a tractable QP.
  - Sub-gradient descent solves an equivalent problem with provable convergence bound.