Robot Learning

Imitation learning

Inverse reinforcement learning

Last time…

$$\underset{\pi}{\text{maximize}} \quad \mathbb{E}_{\boldsymbol{w}}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

$$\text{subject to} \quad s_t = f(s_t, a_t, w_t)$$

$$a_t = \pi(s_t)$$

✨ Reinforcement Learning ✨

Model-based

Model-free

Approximate DP

Direct Policy Search

Last time…

$$\underset{\pi}{\text{maximize}} \quad \mathbb{E}_{\boldsymbol{w}}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$$

Where does this come from?

$$\text{subject to } s_t = f(s_t, a_t, w_t)$$

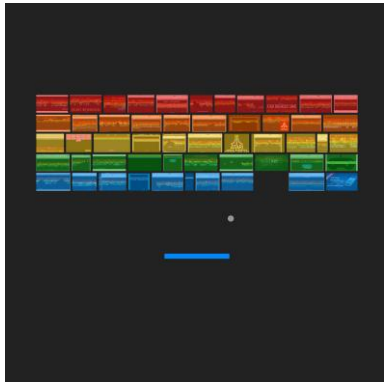
$$a_t = \pi(s_t)$$

This is the world model.

This is our optimization variable.

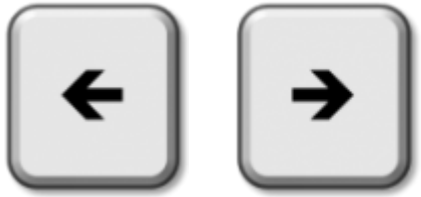
Sometimes it is given…

Goal: Achieve a high score in the Atari game "Breakout"



States: Image of the current screen (?)

Actions: Left and right actions



Reward: Change in the score of the game

Sometimes we (try to) design it…

What is a good reward function for an autonomous car?

Proposal:

- Negative reward for crashing
- Positive reward for high speed
- Negative reward for too high speed

Video from: Mat Kelcey (Edge Impulse)

Sometimes we (try to) design it…

What is a good reward function for a robot vacuum?

Proposal:

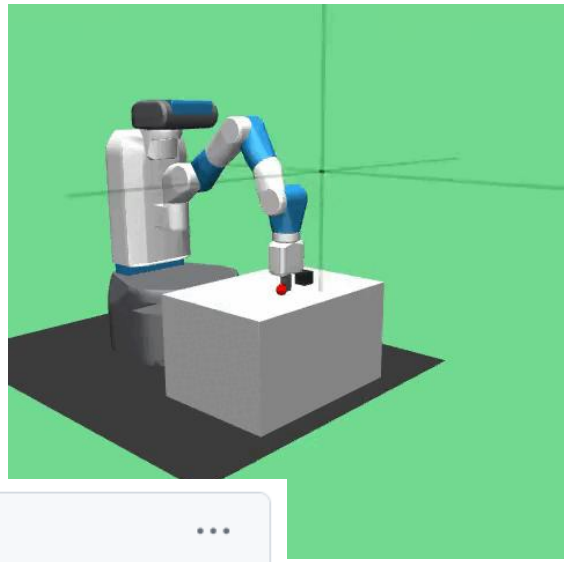
- Positive for vacuuming dirt

> preceding section. We might propose to measure performance by the amount of dirt cleaned up in a single eight-hour shift. With a rational agent, of course, what you ask for is what you get. A rational agent can maximize this performance measure by cleaning up the dirt, then dumping it all on the floor, then cleaning it up again, and so on.

Artificial Intelligence: A Modern Approach
Russell and Norvig

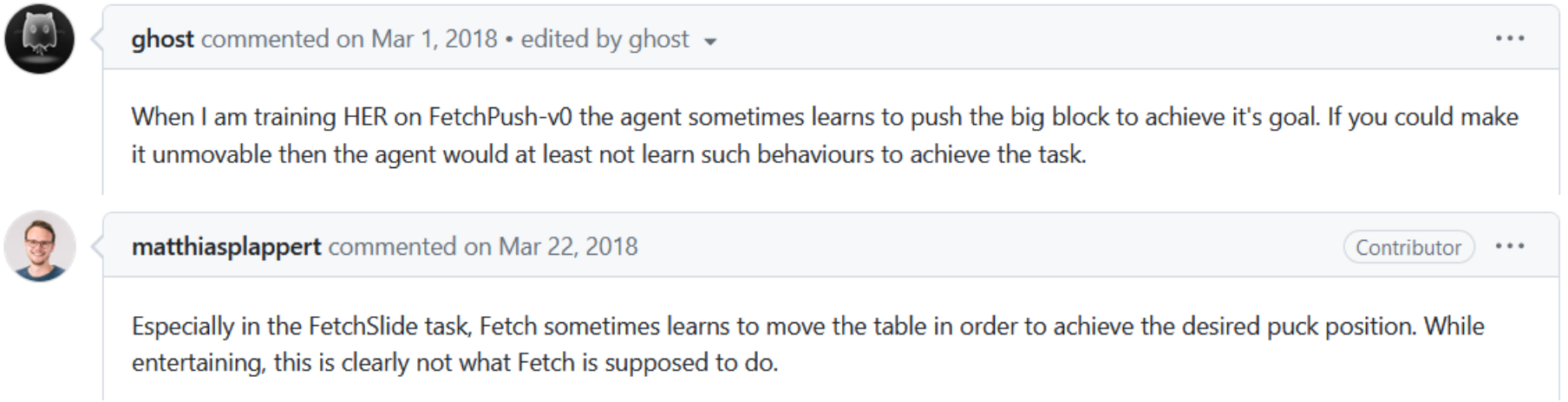
Sometimes we (try to) design it…

What is a good reward function for *FetchPush*?



Proposal:

- Negative for error distance



ghost commented on Mar 1, 2018 • edited by ghost

When I am training HER on FetchPush-v0 the agent sometimes learns to push the big block to achieve it's goal. If you could make it unmovable then the agent would at least not learn such behaviours to achieve the task.

matthiasplappert commented on Mar 22, 2018 Contributor

Especially in the FetchSlide task, Fetch sometimes learns to move the table in order to achieve the desired puck position. While entertaining, this is clearly not what Fetch is supposed to do.

OpenAI Gym
Brockman et al.

Sometimes we (try to) design it…

Goal: Make an RC helicopter fly and perform some maneuvers

States: Sensory input of the helicopter

Actions: Control inputs

Reward: Positive for the maneuvers, negative for crashing

Autonomous helicopter aerobatics through apprenticeship learning
Abbeel et al., IJRR 2010

Not that easy!

Goal: Make an RC helicopter fly and perform some maneuvers

States: Sensory input of the helicopter

Actions: Control inputs

Reward: Positive for the maneuvers, negative for crashing

This is a very naïve reward function. They instead learned the reward from expert demonstrations.

Today…

- Imitation learning

- Inverse reinforcement learning (IRL)

Imitation learning vs. IRL

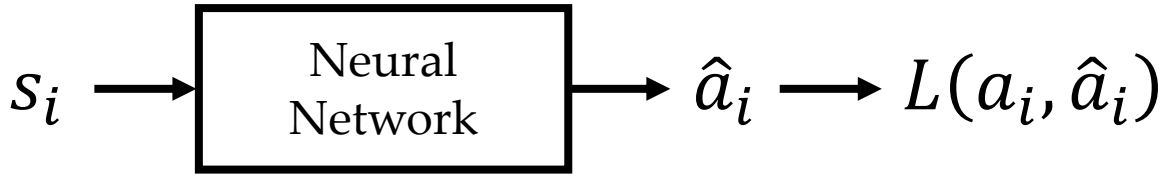
People sometimes use them interchangeably.
We will use the most common definitions.

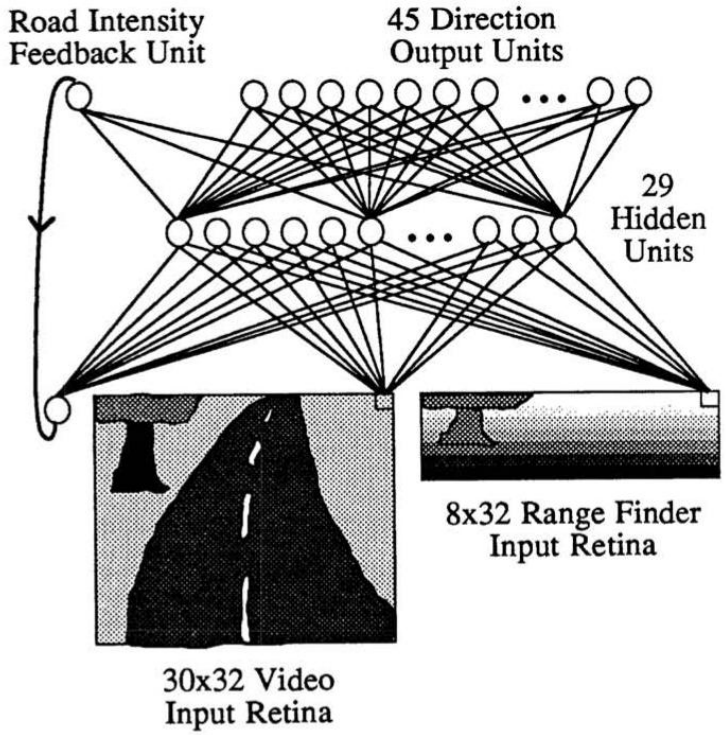
Given some expert data $(s_0, a_0, s_1, a_1, \ldots)$, …

Imitation learning	**Inverse reinforcement learning**
directly learns a policy that imitates the expert.	learns a reward function which, when optimized, performs the task.
simple, not ambiguous, fast	interpretable, generalizable

Behavioral cloning

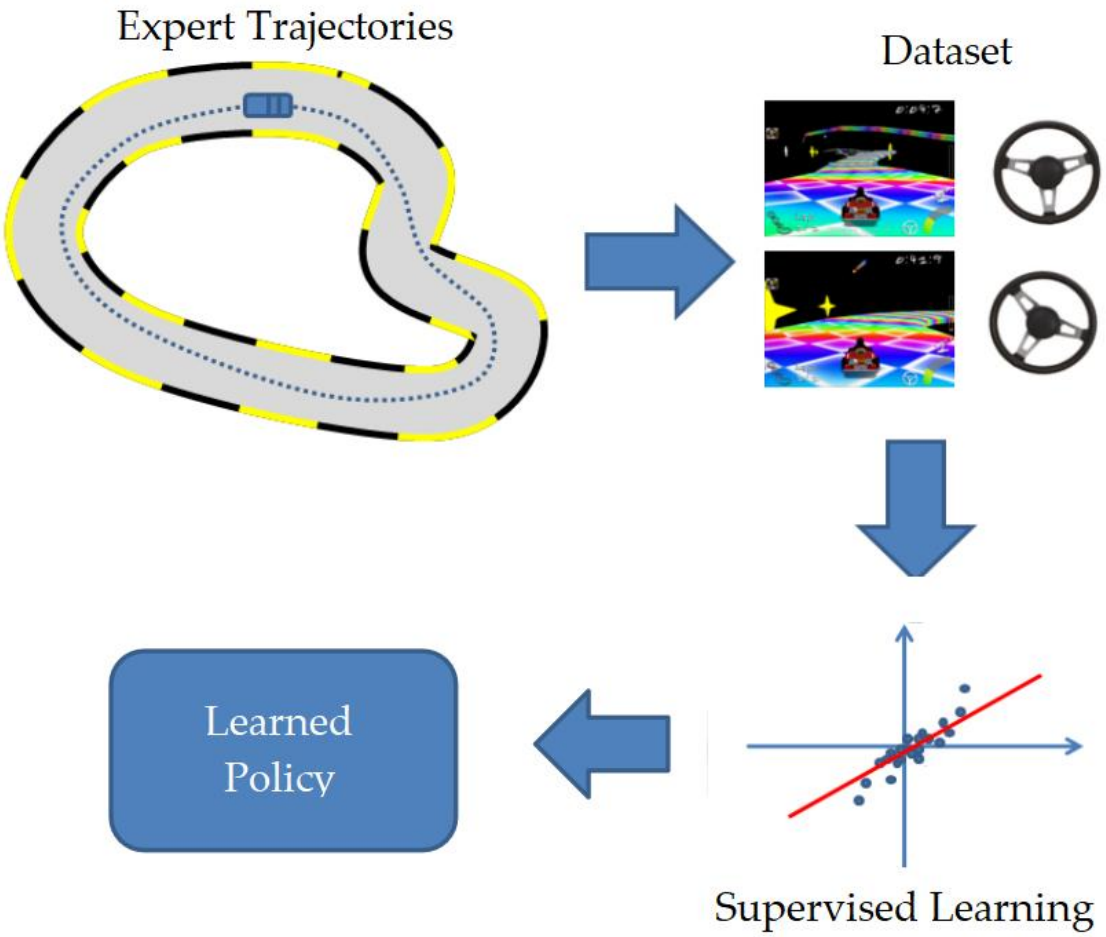
Train a neural network to map states into expert actions.





ALVINN: An Autonomous Land Vehicle in a Neural Network
Pomerleau, NeurIPS 1988

Behavioral cloning

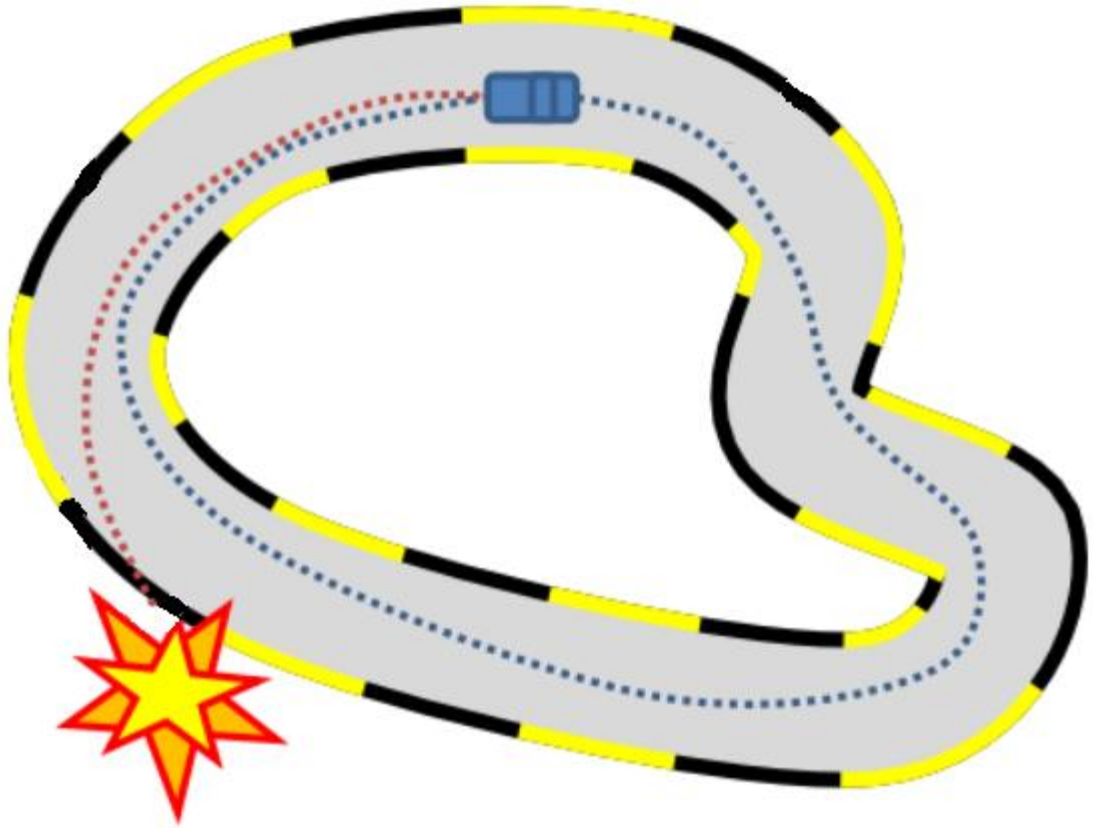


From: Cornell CS4789

Compounding errors

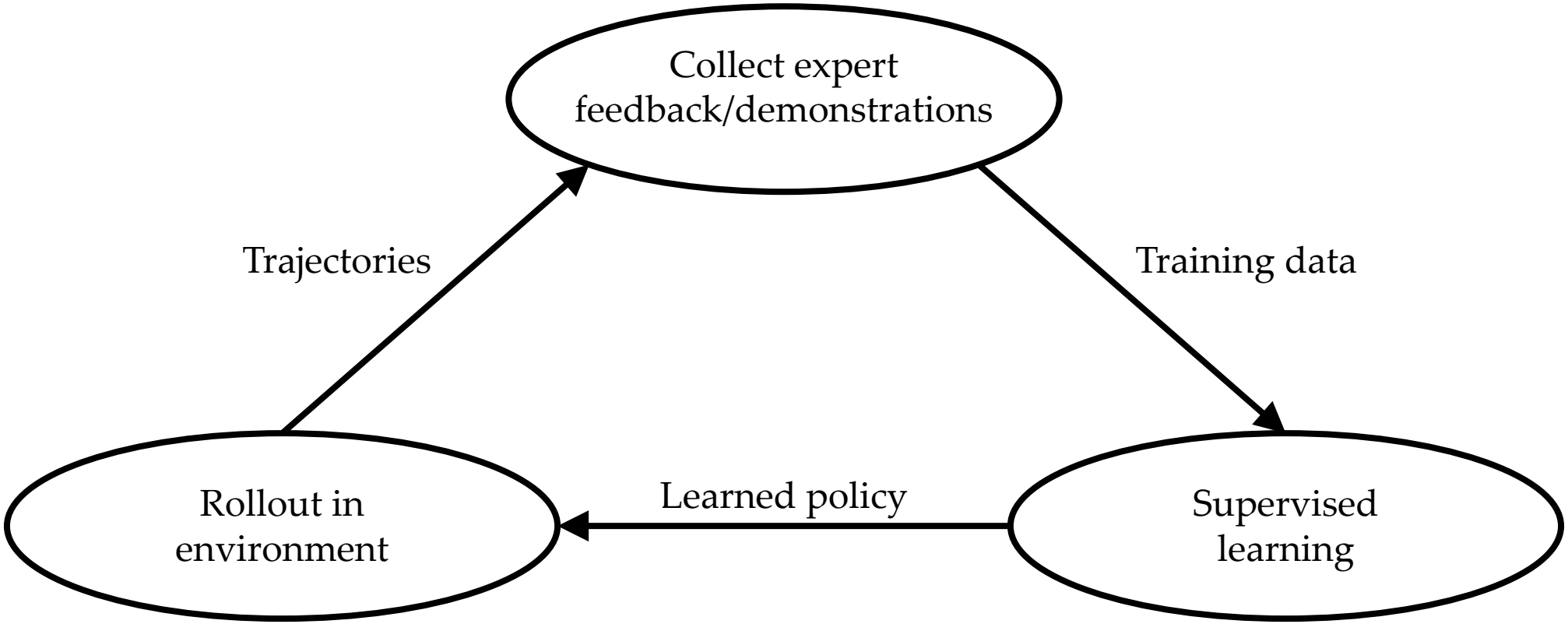
Small errors in the actions taken will slightly deviate the trajectory from the expert.

These new states will lead to larger errors.



From: Cornell CS4789

Direct policy learning



Direct policy learning

More on this next week!

- Ross et al., A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning (2011).
- Ho and Ermon, Generative Adversarial Imitation Learning (2016).
- Florence et al., Implicit Behavioral Cloning (2021).
- Shafiullah et al., Behavior Transformers: Cloning k modes with one stone (2022).
- Jain et al., Vid2Robot: End-to-end Video-conditioned Policy Learning with Cross-Attention Transformers (2024).
- Fu et al., In-Context Imitation Learning via Next-Token Prediction (2024).

Today…

- Imitation learning

- Inverse reinforcement learning (IRL)

Inverse reinforcement learning (IRL)

- Kalman, 1964: Inverse optimal control for 1D problems
- Boyd et al., 1994: Linear matrix inequality (LMI) for LQ setting
- Ng, Russell, 2000: First MDP formulation and reward ambiguity
- Abbeel, Ng, 2004: Apprenticeship learning (feature matching)
- Ratliff et al., 2006: Max margin planning (MMP)
- Ziebart et al., 2008: Max-Ent IRL

From: Stanford CS237B

Today…

- Imitation learning

- Inverse reinforcement learning (IRL)
  - Apprenticeship learning
  - Maximum margin planning
  - Max-Ent IRL

General IRL formulation

We assume a feature function ϕ such that $R(s,a) = w^\top \phi(s,a)$.

We only need to learn w.

$$V^\pi(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t R(s_t, \pi(s_t)) \mid s_0 = s\right]$$

$$= w^\top \mathbb{E}\left[\sum_{t\geq 0} \gamma^t \phi(s_t, \pi(s_t)) \mid s_0 = s\right]$$

$$= w^\top \phi(\pi, s)$$

General IRL formulation

We know, for any policy π and $s \in \boldsymbol{S}$,

$$V^{\pi^*}(s) \geq V^{\pi}(s)$$

which we now write as

$$w^\top \phi(\pi^*, s) \geq w^\top \phi(\pi, s)$$

This is the only condition w must satisfy.

We just solved IRL: it turns out $w = 0$, yay! 🎉

Reward ambiguity

$$w^\top \phi(\pi^*, s) \geq w^\top \phi(\pi, s)$$

More generally, if a w^* satisfies this condition, cw^* will also satisfy for any $c \geq 0$.

Note: Reward ambiguity is not just this. Many w vectors satisfy the condition even if we constrain $\|w\|_2$ to a constant.

Apprenticeship learning

This is how the helicopter flew!

An attempt to alleviate reward ambiguity. First, assume $\|w\|_2 \leq 1$.

Observation:

$$\|\phi(\pi^*, s) - \phi(\pi, s)\|_2 \leq \epsilon \;\Rightarrow\; |w^\top \phi(\pi^*, s) - w^\top \phi(\pi, s)| \leq \epsilon$$

Even if we cannot find the true w^*, we will get expert-level performance if we match the features.

Apprenticeship learning

In most cases, only a subset of the state space can be initial states.

This means we need to match $\phi(\pi^*, s)$ only at $s_0 \sim P(s_0)$:

$$\phi(\pi) = \mathbb{E}_{s_0}\left[\sum_{t \geq 0} \gamma^t \phi(s_t, \pi(s_t))\right]$$

Apprenticeship learning

We iteratively improve the learned w and policy.

Compute the optimal features $\phi(\pi^*)$

Initialize a policy π_0

Loop $i = 0,1, \ldots$**:**

Find w_i that best separates π^* from π_i

Assuming w_i is true weights, learn π_{i+1} optimizing the reward

Apprenticeship learning

Compute $\phi(\pi^*)$ using expert data

Initialize a policy π_0

for $i = 0,1, \ldots$ **do:**

$$w_i, t_i = \arg\max_{w,t} \; t$$

$$\text{subject to} \quad w^\top \phi(\pi^*) \geq w^\top \phi(\pi_j) + t, \forall j \in \{0,1, \ldots, i\}$$

$$\|w\|_2 \leq 1$$

if $t_i \leq \epsilon$ **then: return** the best feature-matching policy from $\{\pi_0, \pi_1, \ldots, \pi_i\}$

else: $\pi_{i+1} \leftarrow \arg\max_{\pi} w_i^\top \phi(\pi)$

We are solving an RL problem in each iteration!

Apprenticeship learning

Compute $\phi(\pi^*)$ using expert data

Initialize a policy π_0

for $i = 0,1, \ldots$ **do:**

What if the expert is suboptimal?

$$w_i, t_i = \arg\max_{w,t} \; t$$

$$\text{subject to} \quad w^\top \phi(\pi^*) \geq w^\top \phi(\pi_j) + t, \; \forall j \in \{0,1, \ldots, i\}$$

$$\|w\|_2 \leq 1$$

if $t_i \leq \epsilon$ **then: return** the best feature-matching policy from $\{\pi_0, \pi_1, \ldots, \pi_i\}$

else: $\pi_{i+1} \leftarrow \arg\max_{\pi} w_i^\top \phi(\pi)$

Today…

- Imitation learning

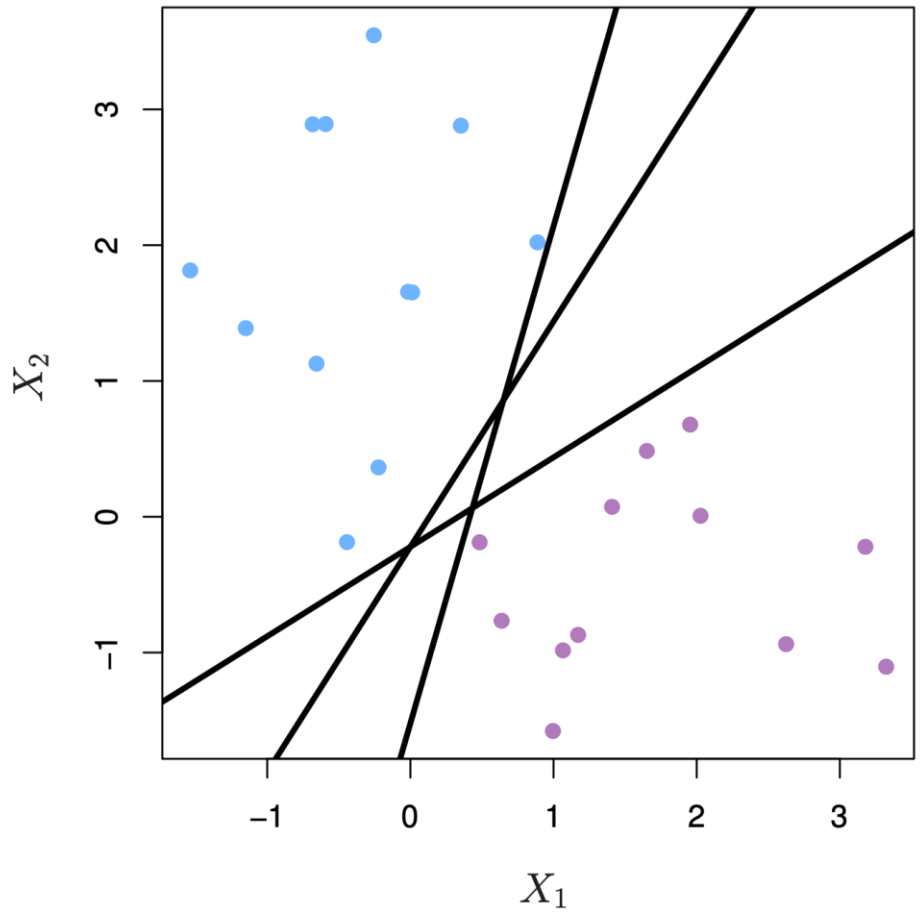
- Inverse reinforcement learning (IRL)
  - Apprenticeship learning
  - Maximum margin planning
  - Max-Ent IRL

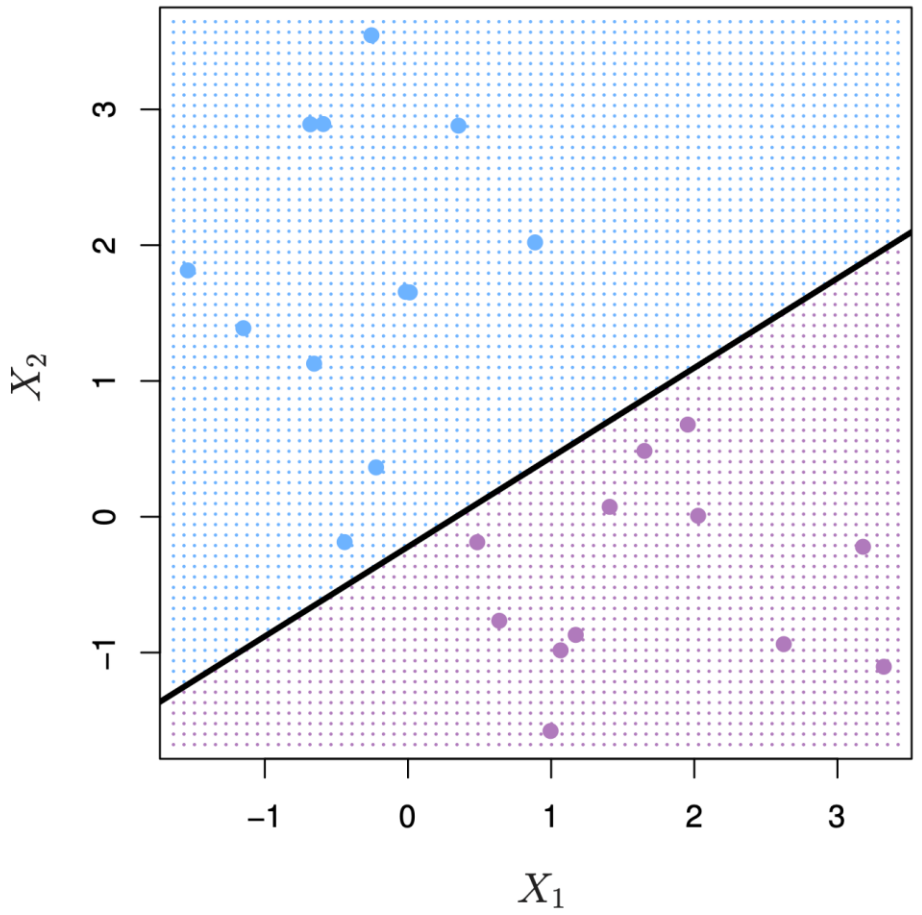
Maximum margin planning (MMP)

MMP has a similar formulation, but helps with suboptimal experts.

First let's go over *maximal margin classifiers*.

Maximal margin classifiers

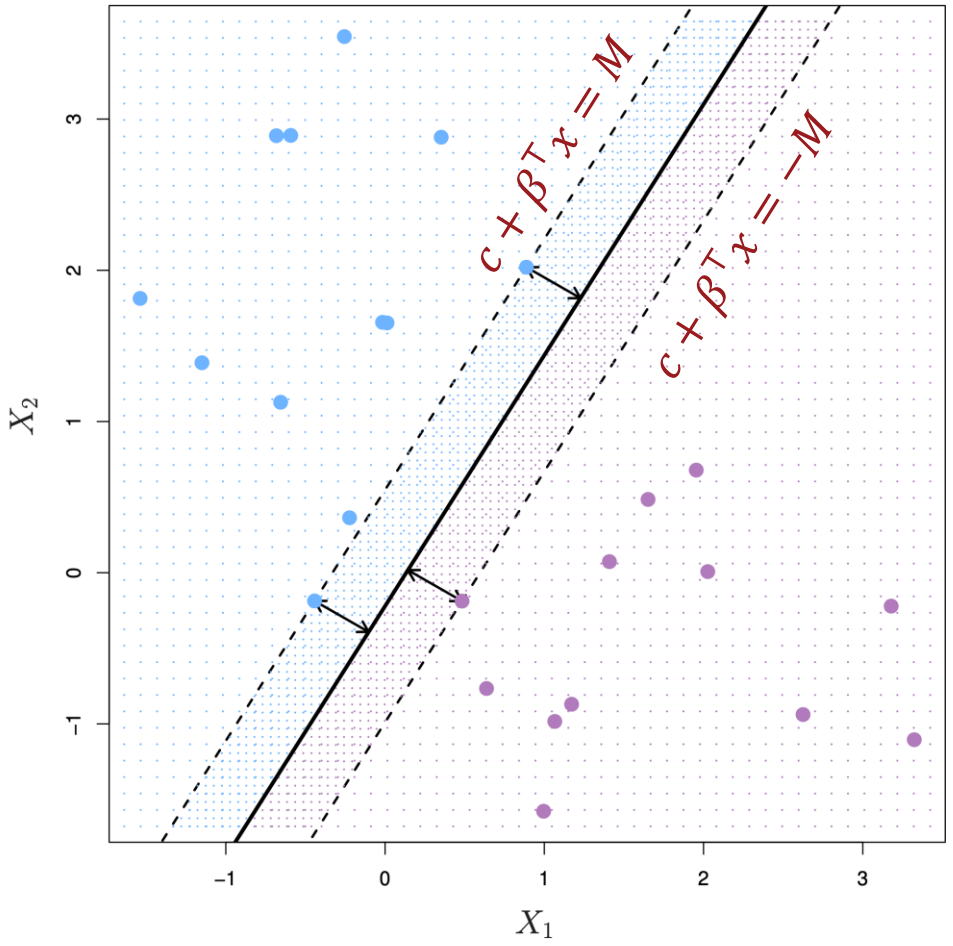




From: Stanford STATS202

Maximal margin classifiers

WLOG, assume the separating hyperplane has distance M to the closest points.



From: Stanford STATS202

Maximal margin classifiers

WLOG, assume the separating hyperplane has distance M to the closest points.

$$\underset{\boldsymbol{\beta},c}{\text{maximize}}\ M$$

$$\text{subject to } \|\boldsymbol{\beta}\|_2 \leq 1$$

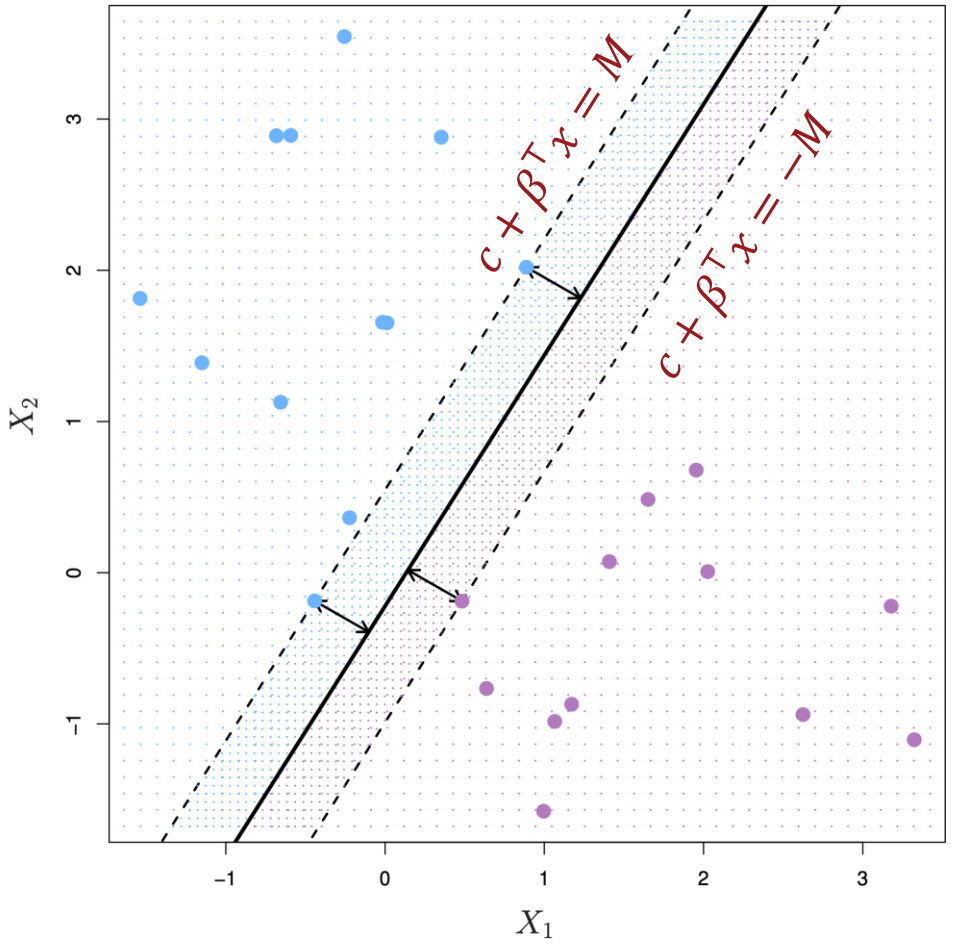
Is it possible that the optimal β has $\|\beta\|_2 < 1$?

$$c + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots \geq M$$

for all positive samples i

$$c + \beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \cdots \leq -M$$

for all negative samples j



Maximal margin classifiers

WLOG, assume the separating hyperplane has distance M to the closest points.

$$\underset{\boldsymbol{\beta},c}{\text{maximize}}\ M$$

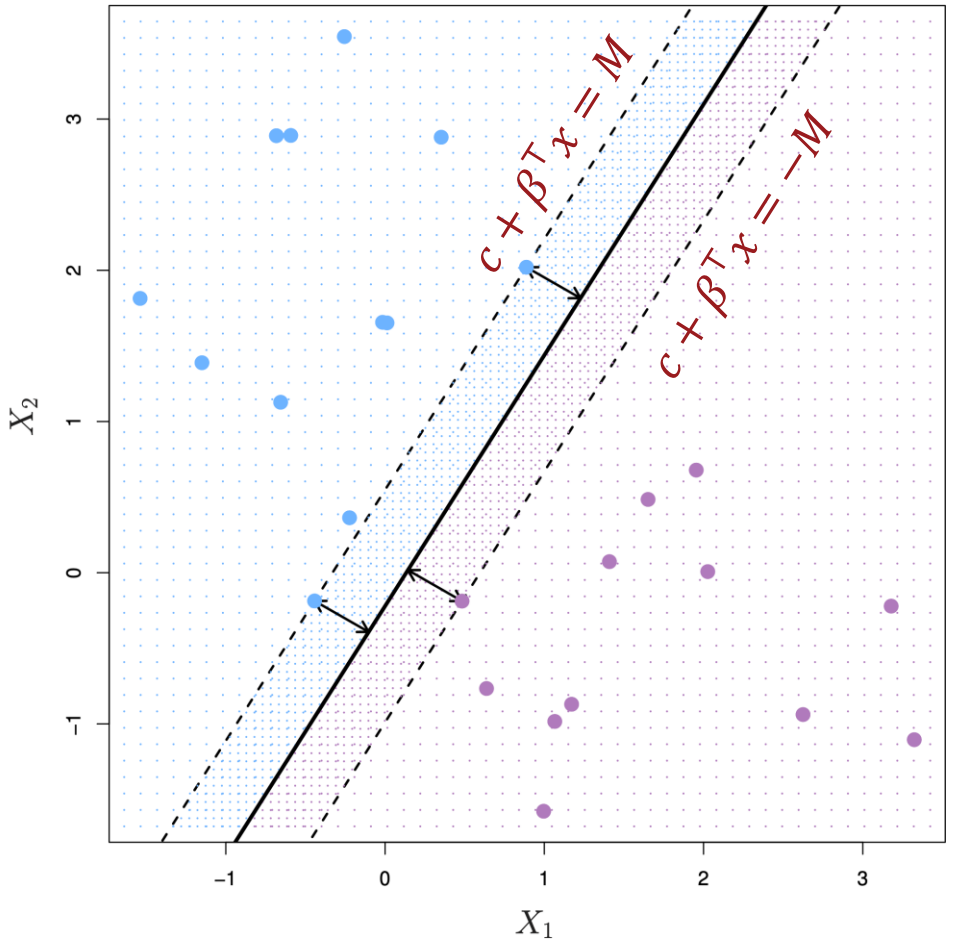
$$\text{subject to } \|\boldsymbol{\beta}\|_2 = 1$$

$$c + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots \geq M$$

for all positive samples i

$$c + \beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \cdots \leq -M$$

for all negative samples j



From: Stanford STATS202

Maximal margin classifiers

WLOG, assume the separating hyperplane has distance M to the closest points.

$$\underset{\boldsymbol{\beta},c}{\text{maximize}}\ M$$

$$\text{subject to } \|\boldsymbol{\beta}\|_2 = 1$$

$$c + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots \geq M$$

for all positive samples i

$$c + \beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \cdots \leq -M$$

for all negative samples j

Let $w = [\beta_1, \beta_2, \ldots]/2M$

Note $\|w\|_2 = \frac{\|\beta\|_2}{2M} = \frac{1}{2M}$

From: Stanford STATS202

Maximal margin classifiers

$$\underset{w,c}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to } \|\boldsymbol{\beta}\|_2 = 1$$

$$c + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots \geq M$$

for all positive samples i

$$c + \beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \cdots \leq -M$$

for all negative samples j

Let $w = [\beta_1, \beta_2, \ldots]/2M$

Note $\|w\|_2 = \frac{\|\beta\|_2}{2M} = \frac{1}{2M}$

From: Stanford STATS202

Maximal margin classifiers

$$\underset{w,c}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to } \|w\|_2 = 1/2M$$

$$c + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots \geq M$$

for all positive samples i

$$c + \beta_1 x_1^{(j)} + \beta_2 x_2^{(j)} + \cdots \leq -M$$

for all negative samples j

Let $w = [\beta_1, \beta_2, \ldots]/2M$

Note $\|w\|_2 = \frac{\|\beta\|_2}{2M} = \frac{1}{2M}$

Maximal margin classifiers

Let $w = [\beta_1, \beta_2, \ldots]/2M$

Note $\|w\|_2 = \frac{\|\beta\|_2}{2M} = \frac{1}{2M}$

$$\underset{w,c}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to } \|w\|_2 = 1/2M$$

$$c/2M + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots \geq 1/2$$

for all positive samples i

$$c/2M + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \cdots \leq -1/2$$

for all negative samples j

From: Stanford STATS202

Maximal margin classifiers

$$\underset{w,c}{\text{minimize}} \quad \|w\|_2$$

subject to

$$c\|w\|_2 + w_1 x_1^{(i)} + w_2 x_2^{(i)} + \cdots \geq 1/2$$

for all positive samples i

$$c\|w\|_2 + w_1 x_1^{(j)} + w_2 x_2^{(j)} + \cdots \leq -1/2$$

for all negative samples j

These constraints just mean

$$w^\top x^{(i)} - w^\top x^{(j)} \geq 1$$

for all positive samples i and negative samples j.

From: Stanford STATS202

Maximal margin classifiers

$$\underset{w,c}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to} \quad w^\top x^{(i)} - w^\top x^{(j)} \geq 1$$

for all positive samples i and negative samples j

From: Stanford STATS202

Maximal margin classifiers

$$\underset{w}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to} \quad w^\top x^{(i)} - w^\top x^{(j)} \geq 1$$

for all positive samples i and negative samples j

Back to MMP

If the expert is optimal, there exists a separating hyperplane $w^\top \phi = c$ such that $w^\top \phi(\pi^*) \geq c$ and $w^\top \phi(\pi) \leq c$ for all $\pi \neq \pi^*$.

So we can use a *maximal marginal classifier* with only one positive sample!

$$\underset{w}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to} \quad w^\top \phi(\pi^*) - w^\top \phi(\pi) \geq 1 \qquad \text{for all } \pi \neq \pi^*$$

Maximum margin planning (MMP)

Let's allow the expert to be suboptimal by adding a slack variable.

$$\underset{w}{\text{minimize}} \quad \|w\|_2$$

$$\text{subject to} \quad w^\top \phi(\pi^*) - w^\top \phi(\pi) \geq 1 \qquad \text{for all } \pi \neq \pi^*$$

Maximum margin planning (MMP)

Let's allow the expert to be suboptimal by adding a slack variable.

$$\underset{w,v}{\text{minimize}} \quad \|w\|_2 + Cv$$

$$\text{subject to} \quad w^\top \phi(\pi^*) - w^\top \phi(\pi) \geq 1 - v \qquad \text{for all } \pi \neq \pi^*$$

Maximum margin planning (MMP)

Let's allow the expert to be suboptimal by adding a slack variable.

We could also be more tolerant to the policies that are similar to π^*.

$$\underset{w,v}{\text{minimize}} \quad \|w\|_2 + Cv$$

$$\text{subject to} \quad w^\top \phi(\pi^*) - w^\top \phi(\pi) \geq 1 - v + d(\pi^*, \pi) \quad \text{for all } \pi \neq \pi^*$$

Today…

- Imitation learning

- Inverse reinforcement learning (IRL)
  - Apprenticeship learning
  - Maximum margin planning
  - Max-Ent IRL

Max-Ent IRL

Assumption: Experts are noisily optimal, i.e., the probability that they demonstrate trajectory ξ is:

$$P(\xi \mid w) = \frac{\exp\left(w^\top \phi(\xi)\right)}{\int \exp\left(w^\top \phi(\xi')\right) d\xi'}$$

where $\phi(\xi)$ is the cumulative discounted features of trajectory ξ.

Max-Ent IRL

Key insight: Find a probability distribution P^* over trajectories such that the feature expectation matches the expert features, i.e.,

$$\mathbb{E}_{\xi \sim P^*(\xi)}[\phi(\xi)] = \phi(\pi^*)$$

But which distribution?

Principle of maximum entropy

"When estimating the probability distribution, you should select that distribution which leaves you *the largest remaining uncertainty* consistent with your constraints. That way you have not introduced any additional assumptions or biases."

From: MIT 6.050J/2.110J

Max-Ent IRL

$$\max_{P} \quad -\int P(\xi) \log P(\xi)\, d\xi$$

$$\text{subject to} \quad \int P(\xi)\phi(\xi) d\xi = \phi(\pi^*)$$

$$\int P(\xi) d\xi = 1$$

$$P(\xi) \geq 0, \qquad \forall \xi$$

Ignore the inequality constraints for now. Later, we will show the solution already satisfies them.

Max-Ent IRL

$$\max_{P} \quad -\int P(\xi) \log P(\xi)\, d\xi$$

$$\text{subject to} \quad \int P(\xi)\phi(\xi)d\xi = \phi(\pi^*)$$

$$\int P(\xi)d\xi = 1$$

Write the Lagrangian using multipliers λ and ν:

$$L(P, \lambda, \nu) = -\int P(\xi) \log P(\xi)\, d\xi + \lambda^\top \left(\int P(\xi)\phi(\xi)d\xi - \phi(\pi^*) \right) + \nu \left(\int P(\xi)d\xi - 1 \right)$$

We now need to solve $\min_{\lambda,\nu} \max_{P} L(P, \lambda, \nu)$.

Solve for P^*

$$L(P,\lambda,\nu) = -\int P(\xi)\log P(\xi)\,d\xi + \lambda^\top\left(\int P(\xi)\phi(\xi)d\xi - \phi(\pi^*)\right) + \nu\left(\int P(\xi)d\xi - 1\right)$$

$$L(P,\lambda,\nu) = \int \underbrace{\left(-P(\xi)\log P(\xi) + \lambda^\top P(\xi)\phi(\xi) + \nu P(\xi)\right)}_{F\left(\xi, P(\xi), \dot{P}(\xi)\right)} d\xi \underbrace{- \lambda^\top\phi(\pi^*) - \nu}_{\text{doesn't depend on P}}$$

Euler-Lagrange Equation

P is a local optimum of $\int F\left(\xi, P(\xi), \dot{P}(\xi)\right)d\xi$ if and only if:

$$\frac{\partial F}{\partial P}\left(\xi, P(\xi), \dot{P}(\xi)\right) = \frac{d}{d\xi}\boxed{\frac{\partial F}{\partial \dot{P}}\left(\xi, P(\xi), \dot{P}(\xi)\right)}$$

This is zero!

Solve for P^*

$$\frac{\partial F}{\partial P}\left(\xi, P(\xi), \dot{P}(\xi)\right) = 0$$

$$\frac{\partial}{\partial P}\left(-P(\xi)\log P(\xi) + \lambda^\top P(\xi)\phi(\xi) + \nu P(\xi)\right) = 0$$

$$\log P^*(\xi) = -1 + \lambda^\top \phi(\xi) + \nu$$

$$P^*(\xi) = e^{\lambda^\top \phi(\xi) + \nu - 1}$$

$$P^*(\xi) = e^{\lambda^\top \phi(\xi)+\nu-1}$$

Back to Lagrangian

$$L(P^*, \lambda, \nu) = \int \left(-P^*(\xi) \log P^*(\xi) + \lambda^\top P^*(\xi)\phi(\xi) + \nu P^*(\xi)\right) d\xi - \lambda^\top \phi(\pi^*) - \nu$$

$$= \int \left(-e^{\lambda^\top \phi(\xi)+\nu-1}(\lambda^\top \phi(\xi) + \nu - 1) + \lambda^\top e^{\lambda^\top \phi(\xi)+\nu-1}\phi(\xi)\right.$$

$$\left. + \nu e^{\lambda^\top \phi(\xi)+\nu-1}\right) d\xi - \lambda^\top \phi(\pi^*) - \nu$$

$$= \int e^{\lambda^\top \phi(\xi)+\nu-1} d\xi - \lambda^\top \phi(\pi^*) - \nu$$

$$P^*(\xi) = e^{\lambda^\top \phi(\xi) + \nu - 1}$$

Solve for ν^*

Having solved for P^*, we now need to solve $\min_{\lambda,\nu} L(P^*, \lambda, \nu)$.

$$L(P^*, \lambda, \nu) = \int e^{\lambda^\top \phi(\xi) + \nu - 1} d\xi - \lambda^\top \phi(\pi^*) - \nu$$

$$\frac{\partial L}{\partial \nu}(P^*, \lambda, \nu) = 0 \;\; \Rightarrow e^{\nu^*} \int e^{\lambda^\top \phi(\xi) - 1} d\xi - 1 = 0$$

$$e^{-\nu^*} = \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

$$\nu^* = -\log \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

Back to P^*

$$P^*(\xi) = e^{\lambda^\top \phi(\xi) + \nu - 1}$$

$$\nu^* = -\log \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

$$P^*(\xi) = e^{\lambda^\top \phi(\xi) + \nu^* - 1}$$

$$P^*(\xi) = e^{\lambda^\top \phi(\xi) - \log \int e^{\lambda^\top \phi(\xi) - 1} d\xi \; - 1}$$

$$P^*(\xi) = \frac{e^{\lambda^\top \phi(\xi)}}{e^{\log \int e^{\lambda^\top \phi(\xi) - 1} d\xi \; + 1}}$$

$$P^*(\xi) = \frac{e^{\lambda^\top \phi(\xi)}}{\int e^{\lambda^\top \phi(\xi)} d\xi}$$

Remember this?
It turns out $w^* = \lambda^*$.

Back to Lagrangian

$$P^*(\xi) = e^{\lambda^\top \phi(\xi) + \nu - 1}$$

$$\nu^* = -\log \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

$$L(P^*, \lambda, \nu^*) = \int e^{\lambda^\top \phi(\xi) + \nu^* - 1} d\xi - \lambda^\top \phi(\pi^*) - \nu^*$$

$$L(P^*, \lambda, \nu^*) = \int e^{\lambda^\top \phi(\xi) - \log \int e^{\lambda^\top \phi(\xi') - 1} d\xi' \, - 1} d\xi - \lambda^\top \phi(\pi^*) + \log \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

$$L(P^*, \lambda, \nu^*) = \int \frac{e^{\lambda^\top \phi(\xi)}}{\int e^{\lambda^\top \phi(\xi')} d\xi'} d\xi - \lambda^\top \phi(\pi^*) + \log \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

$$L(P^*, \lambda, \nu^*) = 1 - \lambda^\top \phi(\pi^*) + \log \int e^{\lambda^\top \phi(\xi) - 1} d\xi$$

Solve for $\lambda^* = w^*$

We want to minimize $L(P^*, \lambda, \nu^*)$.

$$\frac{dL}{d\lambda}(P^*, \lambda, \nu^*) = \frac{d}{d\lambda}\left(1 - \lambda^\top \phi(\pi^*) + \log \int e^{\lambda^\top \phi(\xi) - 1} d\xi\right)$$

$$= \frac{d}{d\lambda} \log \int e^{\lambda^\top \phi(\xi) - 1} d\xi \; - \phi(\pi^*)$$

$$= \frac{\frac{d}{d\lambda} \int e^{\lambda^\top \phi(\xi) - 1} d\xi}{\int e^{\lambda^\top \phi(\xi) - 1} d\xi} - \phi(\pi^*)$$

Solve for $\lambda^* = w^*$

$$\frac{dL}{d\lambda}(P^*, \lambda, \nu^*) = \frac{\frac{d}{d\lambda}\int e^{\lambda^\top \phi(\xi)-1} d\xi}{\int e^{\lambda^\top \phi(\xi)-1} d\xi} - \phi(\pi^*)$$

$$= \frac{\int \frac{d}{d\lambda} e^{\lambda^\top \phi(\xi)} d\xi}{\int e^{\lambda^\top \phi(\xi)} d\xi} - \phi(\pi^*)$$

$$= \frac{\int \phi(\xi) e^{\lambda^\top \phi(\xi)} d\xi}{\int e^{\lambda^\top \phi(\xi)} d\xi} - \phi(\pi^*)$$

This is just $P(\xi \mid w)$

Solve for $\lambda^* = w^*$

$$\frac{dL}{dw}(P^*, w, v^*) = \int \phi(\xi) P(\xi \mid w) d\xi - \phi(\pi^*)$$
$$= \mathbb{E}_{\xi \sim P(\xi|w)}[\phi(\xi)] - \phi(\pi^*)$$

This gives an algorithm:

1. Initialize w
2. Perform RL to learn a policy that optimizes the reward with w
3. Roll out the learned policy to compute:
$$w \leftarrow w - \left(\mathbb{E}_{\xi \sim P(\xi|w)}[\phi(\xi)] - \phi(\pi^*)\right)$$
4. Repeat from step 2

Today…

- Imitation learning

- Inverse reinforcement learning (IRL)
  - Apprenticeship learning
  - Maximum margin planning
  - Max-Ent IRL

Next time…

- Learning from human feedback
  - Suboptimal demonstrations
  - Pairwise comparisons
  - …