Instructor: Dr. Mohammad Hossein Rohban

Lecture 23 Summary

Summarized By: Arshia Gharooni

1 Optimal-Control Interpretation of Demonstrations

Consider an infinite-horizon discounted Markov decision process (MDP)

$$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle, \quad 0 < \gamma < 1, \tag{1.1}$$

with possibly continuous state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$. For any stationary policy π, define the value and action-value functions

$$V_R^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \,\middle|\, s_0 = s \right], \qquad Q_R^\pi(s,a) = R(s,a) + \gamma \mathbb{E}_{s'|s,a} \left[V_R^\pi(s') \right]. \tag{1.2}$$

Optimal control assumes an expert demonstrator follows a policy that is *optimal or near-optimal* for some unknown reward R, i.e.,

$$\pi_E \in \arg\max_\pi V_R^\pi \quad \text{or} \quad V_R^{\pi_E} \geq V_R^\pi - \delta, \ \forall \pi, \tag{1.3}$$

with tolerance $\delta \geq 0$. Recovering such an R therefore provides both a *causal explanation* of observed behaviour and, when re-optimised, a controller that generalises to novel situations.

2 Learning from Demonstrations: Three Paradigms

- **Behavioural cloning** treats the mapping $s \mapsto a$ as a supervised-learning problem.
- **Standard reinforcement learning** presupposes R is known.
- **Inverse reinforcement learning (IRL)** seeks R from trajectories alone and then solves the forward RL problem.

The motivation for IRL is that a single compact reward can induce correct actions in states never visited during demonstration, thereby avoiding covariate-shift error accumulation inherent in pure behavioural cloning.

3 Formal Definition of the IRL Problem

3.1 Demonstrations and Feature Expectations

Let $\mathcal{D} = \{\tau^{(i)}\}_{i=1}^N$ be demonstrations with

$$\tau^{(i)} = \left(s_0^{(i)}, a_0^{(i)}, s_1^{(i)}, a_1^{(i)}, \dots \right), \quad \Phi(\tau) = \sum_{t=0}^{\infty} \gamma^t \, \phi(s_t, a_t), \tag{3.1}$$

where the feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^k$ is fixed. The empirical discounted feature expectation is

$$\widehat{\boldsymbol{\mu}}_E = \frac{1}{N} \sum_{i=1}^{N} \Phi\left(\tau^{(i)}\right). \tag{3.2}$$

A *linear* reward parameterisation is assumed:

$$R_\theta(s,a) = \theta^\top \phi(s,a), \qquad \theta \in \mathbb{R}^k. \tag{3.3}$$

3.2 Ill-posedness Explained

1. **Reward aliasing.** If θ satisfies $\theta^\top \widehat{\boldsymbol{\mu}}_E = 0$, then every demonstrated return equals zero. Scaling θ by any constant keeps returns identical, so infinitely many rewards remain consistent with the data.
2. **Policy non-uniqueness.** The map $\pi \mapsto \boldsymbol{\mu}(\pi) = \mathbb{E}_\pi[\Phi(\tau)]$ need not be injective: different policies can induce the same feature expectation when ϕ is not state-action sufficient.

Additional optimality or regularity principles are therefore required to select a single R.

4 Feature-Matching Inverse RL

4.1 Performance Gap Lemma

Let π, π' be any two policies and θ any parameter vector. Because R_θ is linear,

$$\left|V_\theta^\pi - V_\theta^{\pi'}\right| = \left|\theta^\top \left(\boldsymbol{\mu}(\pi) - \boldsymbol{\mu}(\pi')\right)\right| \le \|\theta\|_\infty \, \|\boldsymbol{\mu}(\pi) - \boldsymbol{\mu}(\pi')\|_1. \tag{4.1}$$

Proof. Substitute (3.3) into the value definitions and apply Hölder's inequality with conjugate norms $\|\theta\|_\infty$ and $\|\cdot\|_1$. ∎

Hence, if the learner attains $\|\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}(\pi)\|_1 \le \varepsilon$, its return under *any* linear reward differs from the expert's by at most $\varepsilon\|\theta\|_\infty$.

4.2 Apprenticeship-Learning Algorithm

Maintain a list $\{\pi^{(j)}\}_{j=1}^m$ and associated $\{\boldsymbol{\mu}^{(j)}\}$. At iteration m, solve the quadratic program:

$$\begin{aligned} \max_{\theta,t} \quad & t \\ \text{s.t.} \quad & \theta^\top \left(\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}^{(j)}\right) \ge t, \quad j = 1, \ldots, m, \\ & \|\theta\|_2 \le 1, \end{aligned} \tag{4.2}$$

yielding a separating hyperplane of margin t. Compute $\pi^{(m+1)} = \arg\max_\pi V_{R_\theta}^\pi$ with any forward RL solver, append $\boldsymbol{\mu}^{(m+1)}$, and repeat until $t \le \varepsilon$.

4.3 Convergence Proof Sketch

Let $D = \max_j \|\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}^{(j)}\|_2$. Each quadratic-program solution delivers a margin

$$t_m \ge \frac{\varepsilon}{D}, \tag{4.3}$$

while the Euclidean projection guarantees that after at most

$$m \le \frac{D^2}{\varepsilon^2} k \tag{4.4}$$

iterations, the hull of learner feature expectations intersects the ε-ball around $\widehat{\boldsymbol{\mu}}_E$. Thus, the algorithm halts in $O(k/\varepsilon^2)$ forward-RL calls.

5 Maximum-Margin Formulation

Imposing an ℓ_1-norm bound $\|\theta\|_1 \leq c$ and slack variables $\xi_j \geq 0$ yields the primal optimisation:

$$\min_{\theta,\xi} \tfrac{1}{2}\|\theta\|_1 + C\sum_j \xi_j \quad \text{s.t.} \quad \theta^\top\big(\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}^{(j)}\big) \geq 1 - \xi_j. \tag{5.1}$$

To obtain the dual, form the Lagrangian with multipliers $\alpha_j \geq 0$:

$$\mathcal{L}(\theta,\xi,\alpha) = \tfrac{1}{2}\|\theta\|_1 + C\sum_j \xi_j - \sum_j \alpha_j\big(\theta^\top\big(\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}^{(j)}\big) - 1 + \xi_j\big). \tag{5.2}$$

Stationarity with respect to ξ_j implies $\alpha_j \leq C$; sub-differential calculus for the ℓ_1-norm then delivers the dual:

$$\min_{0\leq\alpha_j\leq C} \frac{1}{2}\Big\|\sum_j \alpha_j\big(\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}^{(j)}\big)\Big\|_\infty^2 - \sum_j \alpha_j. \tag{5.3}$$

This is identical to the dual of a structured support-vector machine trained to classify expert versus learner feature totals.

6 Latent-Variable Model and Maximum-Entropy Distribution

6.1 Optimality Indicators

Introduce binary latent variables $O_t \in \{0,1\}$ with

$$\Pr(O_t = 1 \mid s_t, a_t; \theta) = \exp R_\theta(s_t,a_t), \quad R_\theta(s_t,a_t) \leq 0, \tag{6.1}$$

so that higher reward implies greater likelihood of optimality.

6.2 Derivation of the MaxEnt Form

The joint log-likelihood of one trajectory and its optimality indicators is

$$\log \Pr(\tau, O_{0:T-1}; \theta) = \sum_{t=0}^{T-1}\Big(\log P(s_{t+1} \mid s_t, a_t) + \log \pi_E(a_t \mid s_t) + O_t R_\theta(s_t,a_t) + \log(1 - e^{R_\theta})^{1-O_t}\Big). \tag{6.2}$$

Maximising the expectation of this log-likelihood under the posterior of O_t with an entropy term for O_t (i.e., an EM iteration with entropy regularisation) yields the optimal posterior:

$$\Pr(O_t = 1 \mid \tau; \theta) = \frac{e^{R_\theta(s_t,a_t)}}{1 + e^{R_\theta(s_t,a_t)}}. \tag{6.3}$$

Substituting back and carrying out the maximisation with respect to θ collapses the terms independent of R_θ, leaving the unconstrained optimisation:

$$\max_\theta \sum_t R_\theta(s_t, a_t) - \log Z(\theta), \quad Z(\theta) = \sum_\tau \exp\Big(\sum_t R_\theta(s_t, a_t)\Big). \tag{6.4}$$

The corresponding trajectory distribution is therefore

$$P_\theta(\tau) = \frac{1}{Z(\theta)} \exp\Big(\sum_t R_\theta(s_t, a_t)\Big), \tag{6.5}$$

which is precisely the *maximum-entropy* distribution subject to reproducing the expert's expected reward.

7 Dynamic-Programming Evaluation of the Partition Function

7.1 Soft Bellman Equations

Define soft value and soft Q-functions recursively:

$$\begin{aligned} Q_\theta(s, a) &= R_\theta(s, a) + \gamma \mathbb{E}_{s'|s,a}\big[V_\theta(s')\big], \\ V_\theta(s) &= \log \sum_{a \in \mathcal{A}} \exp Q_\theta(s, a). \end{aligned} \tag{7.1}$$

Contraction proof. For two value functions V, V',

$$\|(\mathcal{T}V) - (\mathcal{T}V')\|_\infty \le \gamma \|V - V'\|_\infty, \tag{7.2}$$

because the log-sum-exp is 1-Lipschitz and the expectation contracts by γ. Therefore, iterating $V^{(k+1)} = \mathcal{T}V^{(k)}$ converges to the unique fixed point V_θ.

7.2 Connection to the Partition Function

Let s_0 denote the deterministic start state. Because the probability of any path factors into transition probabilities and the policy derived below, one may show inductively that

$$\log Z(\theta) = V_\theta(s_0). \tag{7.3}$$

Hence, soft value iteration computes both the partition function and the optimal *soft* policy:

$$\pi_\theta(a \mid s) = \exp\big(Q_\theta(s,a) - V_\theta(s)\big). \tag{7.4}$$

7.3 Gradient of the Log-Likelihood

For demonstration set $\mathcal{D}$,

$$\nabla_\theta \log P_\theta(\mathcal{D}) = N\big(\widehat{\boldsymbol{\mu}}_E - \boldsymbol{\mu}_{P_\theta}\big), \tag{7.5}$$

where $\boldsymbol{\mu}_{P_\theta} = \mathbb{E}_{P_\theta}[\Phi(\tau)]$. Concavity of $\log P_\theta$ in θ follows because its Hessian equals minus the covariance of Φ under P_θ.

8 Sample-Based IRL with Unknown Dynamics

When P is unknown or continuous, estimate $\boldsymbol{\mu}_{P_\theta}$ via importance sampling. With a proposal policy $\tilde{\pi}$ and roll-outs $\{\tau^{(i)}\}_{i=1}^{M}$,

$$\boldsymbol{\mu}_{P_\theta} = \frac{\displaystyle\sum_{i=1}^{M} w_i \Phi(\tau^{(i)})}{\displaystyle\sum_{i=1}^{M} w_i}, \quad w_i = \frac{\exp R_\theta(\tau^{(i)})}{\prod_t \tilde{\pi}\left(a_t^{(i)} \mid s_t^{(i)}\right)}. \tag{8.1}$$

Since state-transition probabilities cancel between numerator and denominator, no model of P is needed.

9 Variance-Reduction Techniques

9.1 Baseline Subtraction Minimises Variance

For any constant vector b,

$$\mathrm{Var}[w(\Phi - b)] = \mathrm{Var}[w\Phi] - 2b^\top \mathrm{Cov}[w, w\Phi] + b^\top \mathrm{Var}[w] b.$$

Minimising over b gives $b^* = \mathbb{E}_w[\Phi]$. Subtracting this baseline leaves the estimator unbiased but reduces variance.

9.2 Effective Sample Size (ESS)

Define

$$N_{\mathsf{ESS}} = \frac{\left(\sum_i w_i\right)^2}{\sum_i w_i^2}. \tag{9.1}$$

Derivation. The Cauchy–Schwarz inequality implies

$$\left(\sum_i w_i\right)^2 \leq M \sum_i w_i^2,$$

so $N_{\mathsf{ESS}} \leq M$. Under Monte-Carlo central-limit theory, the variance of the self-normalised estimator is approximately $\mathrm{Var}[w\Phi]/\left(\sum_i w_i\right)^2$, hence N_{ESS} behaves as the reciprocal of the variance inflation factor and serves as a diagnostic of sample quality.

9.3 Adaptive Loop

Iteratively:

1. Generate roll-outs with current learner policy $\tilde{\pi}$.
2. Update θ by gradient ascent using the variance-reduced estimate.
3. Improve $\tilde{\pi}$ by any RL method treating R_θ as cost.
4. If N_{ESS} falls below a threshold, resample trajectories.

10 Guided Cost Learning (GCL)

10.1 Objective Functions

Maintain a replay buffer $\mathcal{B}$ containing both expert and learner trajectories. The reward network R_θ maximises

$$\mathcal{L}(\theta) = \sum_{\tau \in \mathcal{D}} R_\theta(\tau) - \log \sum_{\tau \in \mathcal{B}} \exp R_\theta(\tau). \tag{10.1}$$

which is the log-likelihood of a logistic classifier that labels trajectories as expert (positive) or non-expert (negative).

10.2 Policy Update

The learner policy π_ϕ is updated by Trust-Region Policy Optimisation (TRPO) to *minimise* expected cost $J(\phi) = \mathbb{E}_{\pi_\phi}[R_\theta(\tau)]$ under a constraint $D_{\mathsf{KL}}(\pi_\phi \| \pi_{\phi_{\mathsf{old}}}) \leq \delta$. The trust-region ensures the policy distribution remains close enough to its predecessor so that importance weights stay well-behaved.

10.3 Convergence Intuition

At equilibrium, the classifier cannot distinguish learner from expert trajectories; hence the Jensen–Shannon divergence between their distributions is zero and $P_\theta = P_E$. Simultaneously, because the policy optimisation reduces the learned cost, the learner actions approach optimality under the converged reward.