Optimization for CVaR IRL

March 11, 2020

1 Notation

We use the following notation:

- States $\mathcal{S} = \{1, \ldots, S\}$
- Actions $\mathcal{A} = \{1, \ldots, A\}$
- Δ^k probability simplex in k-dimensions.
- Initial distribution: $p_0 \in \Delta^S$
- Rewards: $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Policy: $\pi : \mathcal{S} \to \mathcal{A}$ **[I think all of this should work for stochastic policies, too, right?]**
- Rewards for policy π: $r_\pi(s) = r(s, \pi(s))$
- Expert's policy $\pi_E : \mathcal{S} \to \mathcal{A}$
- Transition probability for a policy π: P_π, treated as a matrix, defined as:

$$P_\pi(s, s') = P(s, \pi(s), s')$$

- Occupancy frequency for a policy: $u_\pi = (I - \gamma P_\pi^\mathsf{T})^{-1} p_0$. This can be derived by noting that the Markov Chain stationary distribution over states u_π satisfies the following equation at equilibrium: $u_\pi = p_0 + \gamma P_\pi^T u_\pi$. Solving for u_π yields the above equation.
- Linear feature matrix with rows as states and columns as features: $\Phi \in \mathbb{R}^{S \cdot A \times k}$, where k is the number of features
- Assume that the rewards are approximated as $r = \Phi w$ for some $w \in \mathbb{R}^k$
- Feature counts: $\mu_\pi = \Phi^\mathsf{T} u_\pi$. Here $\mu_\pi \in \mathbb{R}^k$
- Value function $v_\pi = (I - \gamma P)^{-1} r_\pi$. This can be derived via the bellman equation for values: $v_\pi = r_\pi + \gamma P_\pi v_\pi$ and solving for v_π.
- Return for a specific policy and rewards: $\rho(\pi, r) = p_0^\mathsf{T} v = u_\pi^\mathsf{T} r_\pi$

Now, assume that R is the random variable representing the reward. The posterior can be derived using Bayesian IRL. Let $R_1, R_2, R_3, \ldots$ be samples from the posterior distribution.

2 Value at Risk

When dealing with risk we will assume that lower values are worse (riskier), thus we will want to maximize the Value at Risk or Conditional Value at Risk since tails to the left are bad. We will define α-Value at risk as the $(1-\alpha)$ quantile worst-case outcome. Thus, the α-VaR is such that

$$\alpha\text{-VaR}[X] = \sup\{x : Pr(X \geq x) \geq \alpha\} \tag{1}$$

Given policy π, our AAAI'19 paper [**?**] focused on finding a high-confidence lower bound on:

$$\text{V@R}\left[\rho(\pi, R) - \rho(\pi^*_R, R)\right],$$

This is not quite the same as finding a lower bound on

$$\text{V@R}\left[\rho(\pi, R) - \rho(\pi_E, R)\right],$$

where R is the random variable distributed according to the posterior from the Bayesian IRL. This second form is interesting because we may be able to do better than the expert. We don't want to match the risk of the expert, rather we want to minimize our risk with the expert as the baseline. **Can we do the same thing as our AAAI paper by reusing the BIRL π^* for each policy? I think so. We just adjust the objective so instead of u_E we use u_{π^*}, right?** We will denote the posterior distribution p; and generally assume that p is a probability distribution over a finite number of samples from the posterior distribution, e.g. a uniform distribution over n samples from MCMC. In [**?**] we derive finite-sample bounds in terms of the number of samples from the posterior distribution. Unfortunately, V@R is not convex and thus is hard to optimize.

3 Average Value at Risk

Average Value at Risk (AV@R) is a convex coherent risk measure. It is also commonly referred to as Conditional Value at Risk, expected tail risk, or expected shortfall. It is convex, and is a lower bound on V@R. It can be also preferable because it does not ignore how heavy the tail of the distribution is. V@R only considers the quantile, but ignores any outcome that may be worse than that.

The intuitive (but not entirely correct) definition of AV@R (the same as CVaR) is:

$$\text{AV@R}_\alpha[X] = \mathbb{E}\left[X \;\middle|\; X \leq \text{V@R}_\alpha[X]\right] .$$

This only works for atomless distributions such that no ω has a positive probability (i.e. most continuous distributions). However, we are interested in maximizing AV@R given a finite number of samples from the posterior distribution $P(R|D)$. The correct convex definition of AV@R that works for any distribution (discrete or continuous) is:

$$\max_\sigma \left(\sigma - \frac{1}{1-\alpha} p^\intercal [\sigma \cdot \mathbf{1} - x]_+\right) ,$$

where $[\cdot]_+$ is an element-wise non-negative part of the vector x: $[x]_+ = \max\{x, \mathbf{0}\}$.

A popular way to analyze and use coherent risk measures is to look at their robust representation:

$$\text{AV@R}_\alpha[X] = \min_{q \in \mathcal{Q}} \mathbb{E}_q[X] ,$$

which is the expectation with respect to a worst-case distortion of the nominal probability distribution p. For AV@R the set $\mathcal{Q}$ is defined as:

$$\mathcal{Q} = \left\{q \in \Delta^n \;\middle|\; q \leq \frac{1}{1-\alpha} p\right\} ,$$

where Δ^n is the probability simplex over $\mathbb{R}^n$ and $p \in \Delta^n$.

Lets say that the goal is to find the best policy and we want to minimize AV@R of the "robust baseline regret" [**?**, **?**, **?**]. We called this a robust baseline regret but this is just the standard objective in IRL:

$$\max_{\pi} \text{AV@R}_\alpha \left[\rho(\pi, R) - \rho(\pi_E, R)\right] \tag{2}$$

We can formulate (2) as a linear program following the next steps. Recall the one to one correspondence between randomized policies $\pi : \mathcal{S} \to \Delta^A$ (where A is the number of actions) and the occupancy frequencies u [**?**]. That means that $\max_\pi \rho(\pi, r)$ corresponds to the following linear program [**?**]:

$$\max_{u:\mathcal{S}\times\mathcal{A}\to\mathbb{R}} \left\{ r^\intercal u \;\mid\; \sum_{a\in\mathcal{A}} (\mathbf{I} - \gamma \cdot P_a^\intercal) u_a = p_0, u \geq \mathbf{0} \right\} .$$

I'm currently solving this via SciPy's built in LP solver as follows:

$$\begin{aligned}
\min_{u:\mathcal{S}\times\mathcal{A}\to\mathbb{R}} \quad & -r^\intercal u && (3)\\
\text{s.t.} \quad & \left[(I - \gamma P_{a_1}^\intercal), \ldots, (I - \gamma P_{a_m}^\intercal)\right] \begin{bmatrix} u_{a_1} \\ \vdots \\ u_{a_n} \end{bmatrix} = p_0 && (4)\\
& u \geq \mathbf{0} && (5)
\end{aligned}$$

where $u_a = [u_{(s_1,a)}, u_{(s_2,a)}, \ldots u_{(s_n,a)}]^\intercal$.

Using the same approach as the linear program above, we can formulate (2) as a linear program following these steps. Let R be a matrix $(S \cdot A) \times n$ of all sampled posterior rewards R. That is, each column of R represents one sample of the vector over rewards for each state and action pair. (2) becomes:

$$\max_{u,\sigma} \left\{ \sigma - \frac{1}{1-\alpha} p^\intercal \left[\sigma \cdot \mathbf{1} - R^\intercal u + R^\intercal u_E\right]_+ \;\mid\; \sum_{a\in\mathcal{A}} (\mathbf{I} - \gamma \cdot P_a^\intercal) u_a = p_0, u \geq \mathbf{0} \right\} . \tag{6}$$

This is a linear program which works in the tabular case. This can be written more explicitly as

$$\begin{aligned}
\max_{\sigma,u} \quad & \sigma - \frac{1}{1-\alpha} p^\intercal z && (7)\\
\text{s.t.} \quad & z \geq \sigma \mathbf{1} - R^\intercal (u - u_E) && (8)\\
& \left[(I - \gamma P_{a_1}^\intercal), \ldots, (I - \gamma P_{a_m}^\intercal)\right] \begin{bmatrix} u_{a_1} \\ \vdots \\ u_{a_n} \end{bmatrix} = p_0 && (9)\\
& u \geq \mathbf{0} && (10)\\
& z \geq \mathbf{0} && (11)
\end{aligned}$$

I'm using SciPy to solve this LP in the following form:

$$\begin{aligned}
\min_{\sigma,u} \quad & -\sigma + \frac{1}{1-\alpha} p^\intercal z && (12)\\
\text{s.t.} \quad & -R^\intercal u + \sigma \mathbf{1} - z \leq -R^\intercal u_E && (13)\\
& \left[(I - \gamma P_{a_1}^\intercal), \ldots, (I - \gamma P_{a_m}^\intercal)\right] \begin{bmatrix} u_{a_1} \\ \vdots \\ u_{a_n} \end{bmatrix} = p_0 && (14)\\
& u \geq \mathbf{0} && (15)\\
& z \geq \mathbf{0} && (16)
\end{aligned}$$

The optimal risk-averse IRL policy $\pi^\star$ can be constructed from an optimal $u^\star$ solution to (6) as:

$$\pi^\star(s,a) = \frac{u^\star(s,a)}{\sum_{a' \in \mathcal{A}} u^\star(s,a')} \ . \tag{17}$$

The linear program can be easily extended to linear approximation by assuming that $r = \Phi w$ is which case the return becomes $u^\mathsf{T} r = u^\mathsf{T} \Phi w = \mu^\mathsf{T} w$.

This formulation can be extended to non-linear approximation using a similar approach as GAIL, I think. The key is the dual representation of AV@R naturally maps to an adversarial algorithm.

3.1 Recovering Rewards

The question is how to recover the reward vector the would generate the AVaR return. One possibility is to use the dual representation of AVaR. Let $\pi^\star$ be the optimal solution to (6) as in (17). Let:

$$\mathcal{Q} = \left\{ q \in \Delta^n \ \mid \ q \leq \frac{1}{1-\alpha} p \right\} \ .$$

Then to get the reward, let $\pi^\star$ be the optimal solution to (6). Then one needs to solve:

$$\text{AV@R}_\alpha \left[\rho(\pi^\star, R) - \rho(\pi_E, R) \right] = \max_{q \in \mathcal{Q}} \mathbb{E}_{R \sim q} \left[\rho(\pi^\star, R) - \rho(\pi_E, R) \right] \ . \tag{18}$$

Another way to write the expectation would be as follows:

$$\mathbb{E}_{R \sim q} \left[\rho(\pi^\star, R) - \rho(\pi_E, R) \right] = \sum_{i=1}^{n} q_i \left(\rho(\pi^\star, r_i) - \rho(\pi_E, r_i) \right) \ ,$$

where r_i is the i-th posterior sample.

Let $q^\star$ be the optimal solution to the linear program in (18). Using the fact that ρ is linear in r, we get:

$$\mathbb{E}_{R \sim q^\star} \left[\rho(\pi^\star, R) - \rho(\pi_E, R) \right] = \rho \left(\pi^\star, \mathbb{E}_{R \sim q^\star}[R] \right) - \rho \left(\pi_E, \mathbb{E}_{R \sim q^\star}[R] \right) \ .$$

That means that $\mathbb{E}_{R \sim q^\star}[R]$ is the worst-case reward.