Our Q-function:

$$Q(s,a) = r(s,a) + \beta \sum_{s'} P(s'|s,a)V(s'),$$

where

$$V(s) = E\{\max_{a \in A} Q(s,a) + \epsilon_a\}.$$

Thus,

$$\begin{aligned}
V(s) &= \sum_a P(a|s)E[Q(s,a) + \epsilon_a|Y^* = a] \\
&= r^\pi(s) + \sum_a E[\epsilon_a|Y^* = a] + \beta \sum_{s'} P^\pi(s'|s)V(s') \\
&= r^\pi(s) + |A|\gamma - \sum_a \log(\pi(a|s)) + \beta \sum_{s'} P^\pi(s'|s)V(s')
\end{aligned}$$

After normalization, what we really have is:

$$V(s) = r^\pi(s) - \sum_a \log(\pi(a|s)) + \beta \sum_{s'} P^\pi(s'|s)V(s').$$

For any given π, define $H^\pi : V \to V$, $V : S \to R$, we have

$$[H^\pi v](s) = r^\pi(s) - \sum_a \log(\pi(a|s)) + \beta \sum_{s'} P^\pi(s'|s)v(s'),$$

which is a contraction mapping. For each θ in r_θ, IRL inner loop is trying to find π^*, a randomized Markovian policy, such that

$$H^{\pi^*} v^* = v^*$$

$$\pi^*(a|s) \propto e^{r(s,a) + \beta E_{s'|s} v^*(s')}.$$

This may be viewed as a perturbed version of MDP, where we have $\sup_\pi H^\pi v^* = v^*$ in our textbooks. To identify θ, they essentially solved

$$\begin{aligned}
&\min_\theta KL(\pi^*_\theta || \hat{\pi}) \\
&s.t. H^{\pi^*} v^* = v^* \\
&\quad \pi^*(a|s) \propto e^{r(s,a) + \beta E_{s'|s,a} v^*(s')}.
\end{aligned}$$

The inner loop is expensive. Enforce $\pi^*_\theta = \hat{\pi}$, we have

$$H^{\hat{\pi}}_\theta v^{\hat{\pi}}_\theta = v^{\hat{\pi}}_\theta,$$

which is a set of system linear equations depending on θ (note this is not policy iteration; this is only policy evaluation step). Since $[I - \beta P^{\hat{\pi}}]^{-1}$ only needs to be inverted once, $v^{\hat{\pi}}_\theta$ can be obtained efficiently for different θ (a simple matrix * vector). This is why CCP method is much faster than NFXP. Accordingly, $\pi_\theta(a|s) \propto e^{r_\theta(s,a) + \beta E_{s'|s} v^{\hat{\pi}}_\theta(s')}$, you now select θ to maximize log-likelihood. That is,

$$\begin{aligned}
&\max_\theta \sum_{t=1}^T \log \pi_\theta(a_t|s_t) \\
&s.t. H^{\hat{\pi}}_\theta v^{\hat{\pi}}_\theta = v^{\hat{\pi}}_\theta \\
&\pi_\theta(a|s) \propto e^{r_\theta(s,a) + \beta E_{s'|s,a} v^{\hat{\pi}}_\theta(s')}
\end{aligned}$$

Given v, construct π is really easy. The real bottlenecks are (1) taking gradient of $\log \pi_\theta(a_t|s_t)$ and (2) inverting $H^{\hat{\pi}}_\theta v^{\hat{\pi}}_\theta = v^{\hat{\pi}}_\theta$.

For (1), the reason is π_θ being implicit, while for (2) inversion is expensive for large or infinite S. Recall Gaussian elimination is $O(n^3)$ (or maybe 2.6 or 2.8, i forget the exact number). There are some ways for efficiently inverting matrix, which mainly hinge upon utilizing special structure of a matrix, such as sparsity, symmetry, etc. Actor-critic assumes π is in the form of $e^{\omega^T \Phi}$, i.e., parameterize Q function. For (2), two popular ways in the literature, neural network or linear approximation. I am an idiot on neural network, so let's proceed with linear approximation. In fact, this is a main idea of approximate dynamic programming (ADP) and I highly suspect many algorithmic and theoretical results from ADP can be extended to this case with a slightly modified H operator.

It will be great if we know some features of the reward function f. Remember, successful implementation of ADP requires you have a good understanding of your problem's structural results, such as convexity, piecewise linearity, etc. Our belief-based IRL model is indeed convex and we haven't examined how to utilize it.

Build Krylov space $\{f - \sum_a \log(\pi(a|S)), \beta P^{\hat{\pi}}(f - \sum_a \log(\pi(a|S))), ..., \beta^K P^{\hat{\pi}, K-1}(f - \sum_a \log(\pi(a|S)))\} = \{\Phi_1, ..., \Phi_K\}$.

$\forall \theta$, we want to find $\tilde{v}_\theta^{\hat{\pi}} = \omega^T \Phi \approx v_\theta^{\hat{\pi}}$, where $\omega \in R^K, K << |S|$. Namely,

$$\tilde{v}_\theta^{\hat{\pi}} = \arg \max_{u \in Krylov} ||u - v_\theta^{\hat{\pi}}||.$$

Equivalently, we seek ω^* such that

$$\begin{aligned} \omega^* \in & \arg\min_\omega ||\omega^T \Phi(S) - (r_\theta(S) - \sum_a \log(\hat{\pi}(a|S)) + \beta \int_S P^{\hat{\pi}}(s'|s) \omega^T \Phi(s')) ds'||_D \\ & \arg\min_\omega ||\omega^T (\Phi - \beta P^{\hat{\pi}} \Phi) - (r_\theta - \sum_a \log(\hat{\pi}(a|S))||_D \end{aligned}$$

where D has the d_s on its diagonal and $d = (d_s)$ is the stationary distribution of $P^{\hat{\pi}}$.

Note while $|S|$ can be infinite, we only need K equations to uniquely determine ω, and taking the inverse of $(\Phi - \beta P^{\hat{\pi}} \Phi)$ is manageable since K is small and we only need to do it once.

Now we construct

$$\pi_\theta(a|s) = \frac{e^{r_\theta(s,a) + \beta \omega^T P^a \Phi}}{\sum_{a'} e^{r_\theta(s,a') + \beta \omega^T P^{a'} \Phi}}$$

Derivative of softmax has an analytical expression.. So update θ, continue...

If $r_\theta \in Krylov$ space we construct, we will have

$$\pi_\theta(a|s) = \frac{e^{(\theta + \beta \omega^T P^a)\Phi}}{\sum_{a'} e^{(\theta + \beta \omega^T P^{a'})\Phi}},$$

which will allow us to directly use actor-critic to calculate v. We can test which way is faster.