

A Proofs for Lemma 4.1, Theorem 4.2

The proof associated with Lemma 4.1 follows.

Proof of Lemma 4.1. We now use the continuity of the likelihood, the finite sample analysis of multivariate kernel density estimators in Wand & Jones (1994) [Section 4.4, Equation 4.16] (Appendix B), which defines the Mean Integrated Square Error (MISE) of the density function, and Theorem 1 of Chacón & Duong (2018) [Section 2.6-2.9] (Appendix C), which asserts that as the sample size increases, the mean of the density estimator converges and variance prevents the mean from exploding. We can use Theorem 1 because we assume that the density function is square-integrable and twice differentiable and that the bandwidth approaches 0 as the dataset size increases. Then, up to a constant, for a given state-action pair (s, a) ,

$$\frac{1}{m} \sum_{j=1}^m e^{-d_s((s,a),(s_j,a_j))^2/(2h)} e^{-d_r(R,R_j)^2/(2h')} \xrightarrow{m \rightarrow \infty} p(s, a, R).$$

The same holds true for d_r , $\frac{1}{m} \sum_{\ell=1}^m e^{-d_r(R,R_\ell)^2/(2h')} \xrightarrow{m \rightarrow \infty} p(R)$. By the Continuous Mapping Theorem (Mann & Wald, 1943), we conclude that

$$\hat{p}_m(s, a|R) = \frac{\frac{1}{m} \sum_{j=1}^m e^{-d_s((s,a),(s_j,a_j))^2/(2h)} e^{-d_r(R,R_j)^2/(2h')}}{\frac{1}{m} \sum_{\ell=1}^m e^{-d_r(R,R_\ell)^2/(2h')}} \xrightarrow{m \rightarrow \infty} \frac{p(s, a, R)}{p(R)} = p(s, a|R).$$

□

Now we prove Theorem 4.2.

Proof of Theorem 4.2. By Lemma 4.1, as $m \rightarrow \infty$, \hat{p}_m converges to the true likelihood, so we can adopt existing tools from Bayesian asymptotic theory.

We first define an equivalence relation on \mathcal{R} , denoted by \simeq :

$$R_1 \simeq R_2 \text{ iff } p(\cdot|R_1) = p(\cdot|R_2), \text{ a.e.}$$

Note that \simeq satisfies reflexivity, symmetry, and transitivity and is, therefore an equivalence relation. We denote the equivalence class by $[\cdot]$, that is, $[R] = \{R' : R' \simeq R\}$, and the quotient space is defined as $\tilde{\mathcal{R}} := \mathcal{R}/\simeq = \{[R] : R \in \mathcal{R}\}$. The corresponding canonical projection is denoted by $\pi : \mathcal{R} \rightarrow \tilde{\mathcal{R}}, R \mapsto [R]$. Then, the projection π induces a prior distribution on $\tilde{\mathcal{R}}$ denoted by $\tilde{\mathcal{P}} : \tilde{\mathcal{P}}(A) := \mathcal{P}(\pi^{-1}(A))$. Moreover, $\tilde{\mathcal{R}}$ admits a metric \tilde{d} :

$$\tilde{d}([R_1], [R_2]) := \|p(\cdot|R_1) - p(\cdot|R_2)\|_{L^1}.$$

Because this metric uses the L^1 norm, it satisfies symmetry and triangular inequality. Additionally, it is true that

$$\tilde{d}([R_1], [R_2]) = 0 \iff p(\cdot|R_1) = p(\cdot|R_2), \text{ a.e.} \iff R_1 \simeq R_2 \iff [R_1] = [R_2],$$

so \tilde{d} fulfills the Identity of Indiscernibles principle. As a result, \tilde{d} is a valid distance metric on $\tilde{\mathcal{R}}$.

Then consider the following Bayesian model:

$$(s, a)|[R] \simeq p(s, a|[R]), [R] \in \tilde{\mathcal{R}}, [R] \simeq \tilde{\mathcal{P}}.$$

This model is well-defined since $p(s, a|[R])$ is independent of the representative of $[R]$ by the definition of the equivalence class. Observe that $\text{KL}(R, R^*) = \text{KL}([R], [R^*])$ by the definition of the equivalence class. Then, let $A = \{[R] : \text{KL}([R], [R^*]) < \epsilon\} \subset \tilde{\mathcal{R}}$. We can define $\pi^{-1}(A) = \{R \in \mathcal{R} : \text{KL}(R, R^*) < \epsilon\} \subset \mathcal{R}$. As a result, $\tilde{\mathcal{P}}(\{[R] : \text{KL}([R], [R^*]) < \epsilon\}) = \mathcal{P}(\{R : \text{KL}(R, R^*) < \epsilon\}) > 0$ for any $\epsilon > 0$, that is, the KL support condition is satisfied. Moreover, the mapping $[R] \rightarrow p(\cdot|R)$ is one-to-one. Because the Bayesian model is parameterized by $[R]$ and we assume that \mathcal{R} is a compact set, by van der Vaart (2000) [Lemma

10.6](Appendix D) there exist consistent tests as required in Schwartz's Theorem (Ghosal & van der Vaart, 2017) [Example 6.19](Appendix E). Then, by Schwartz (1965), the posterior $\hat{\mathcal{P}}_m^n$ on \mathcal{R} is consistent. That is, for any $\epsilon > 0$, $\hat{\mathcal{P}}_m^n(\{[R] : \tilde{d}([R], [R^*]) < \epsilon\}) \xrightarrow[n \rightarrow \infty]{m \rightarrow \infty} 1$. Put in terms of the original parameter space,

$$\mathcal{P}_m^n(\{R : \|p(\cdot|R) - p(\cdot|R^*)\|_{L_1} < \epsilon\}) = \hat{\mathcal{P}}_m^n(\{[R] : \tilde{d}([R], [R^*]) < \epsilon\}) \xrightarrow[n \rightarrow \infty]{m \rightarrow \infty} 1, \forall \epsilon > 0.$$

□

B Wand and Jones, Equation 4.16

The mean integrated squared error (MISE) of a multivariate kernel density estimator is defined as:

$$MISE\{\hat{f}(\cdot; \mathbf{H})\} = n^{-1}(4\pi)^{-\frac{d}{2}} |\mathbf{H}|^{-\frac{1}{2}} + w^\top \{(1 - n^{-1})\Omega_2 - 2\Omega_1 + \Omega_0\}w$$

where \mathbf{H} is a matrix of bandwidth values, n is the size of the dataset, Ω_a denotes the $k \times k$ matrix with (l, l') entry equal to $\phi_a \mathbf{H} + \Sigma_l + \Sigma_{l'}(\mu_l - \mu_{l'})$, ϕ_d is a d -variate Normal kernel, $w = (w_1, \dots, w_k)^\top$ is a vector of positive numbers summing to 1, and for each $l = 1, \dots, k$, μ_l is a $d \times 1$ vector and Σ_l is a $d \times d$ covariance matrix.

C Asymptotic expansion of the mean integrated squared error Theorem 1

Theorem C.1. (i) The integrated squared bias of the kernel density estimator can be expanded as

$$ISB\{\hat{f}(\cdot; \mathbf{H})\} = \frac{1}{4}c_2(K)^2 \text{vec}^\top \mathbf{R}(D^{\otimes 2}f)(\text{vec } \mathbf{H})^{\otimes 2} + o(\|\text{vec } \mathbf{H}\|^2). \quad (10)$$

(ii) The integrated variance of the kernel density estimator can be expanded as

$$IV\{\hat{f}(\cdot; H)\} = m^{-1}|\mathbf{H}|^{-1/2}R(K) + o(m^{-1}|\mathbf{H}|^{-1/2}). \quad (11)$$

Here, \hat{f} is the estimated density function, \mathbf{H} is a matrix of bandwidth values, $c_2(K) = \int_{\mathcal{R}_d} z_i^2 K(z) dz$ for all $i = 1, \dots, d$ is the variance of the kernel function K , and m is the size of the training dataset. In our work, \mathbf{H} is a diagonal matrix where every element on the diagonal is the same bandwidth h_m . In this work, we assume that f is square-integrable and twice differentiable and that the bandwidth matrix $\mathbf{H} \rightarrow 0$ as $m \rightarrow \infty$. Because we use a Gaussian kernel for K , we know that it is square integrable, spherically symmetric, and has a finite second-order moment.

D Asymptotic Statistics, Lemma 10.6

Lemma D.1. Suppose that Θ is σ -compact, $P_\theta \neq P_{\theta'}$ for every pair $\theta \neq \theta'$, and the maps $\theta \rightarrow P_\theta$ are continuous for the total variation norm. Then there exists a sequence of estimators that is uniformly consistent on every compact subset of Θ .

Here, Θ is the space of parameters, P is the probability density function, and $\theta \in \Theta$ is a parameter.

E Schwartz's Theorem

If $p_0 \in KL(\mathcal{P})$ and for every neighborhood \mathcal{U} of p_0 there exist tests ϕ_n such that $P_0^n \phi_n \rightarrow 0$ and $\sup_{p \in \mathcal{U}^c} P^n(1 - \phi_n) \rightarrow 0$, then the posterior distribution $\mathcal{P}_n(\cdot | X_1, \dots, X_n)$ in the model $X_1, \dots, X_n | p \sim^{iid} p$ and $p \sim \mathcal{P}$ is strongly consistent at p_0 .

Algorithm 1 Kernel Density Bayesian IRL

Input: m training task demonstrations, n test task demonstrations, $\#$ sampling iterations c , bandwidth hyperparameters h, h'
for $l = 1, \dots, c$ **do**
 Sample a reward function \hat{R}_l
 Calculate the likelihood \hat{p}_m^n of \hat{R}_l with m training demonstrations and n expert demonstrations. Use can use Equation 5 or Equation 8 if using a featurized reward function.
 Update the posterior using Equation 6 or Equation 9 if using a featurized reward function.
end for
Output: all sampled reward functions $\{R_l\}_{l=1}^c$

F Code and Experiments

Our experiments were run on an internally-hosted cluster using a 320 NVIDIA P100 GPU whose processor core has 16 GB of memory hosted. Our experiments used a total of approximately 150 hours of compute time. Our code uses the MIT License.

To fit KD-BIRL we use Stan (Team, 2011), which uses a Hamiltonian Monte Carlo algorithm. To fit the BIRL and AVRIL posteriors, we first generate the same number of expert demonstration trajectories as used for KD-BIRL. BIRL and AVRIL use an inverse temperature hyperparameter, α ; we set $\alpha = 1$ for all methods. AVRIL uses two additional hyperparameters γ, δ , which we set to 1. Unless otherwise specified, KD-BIRL uses a uniform prior for the reward $r_s \sim \text{Unif}(0, 1)$ for $s = 1, \dots, g \times g$ and Euclidean distance for d_s, d_r . For the 2×2 and 5×5 Gridworld environments, we specify the domain of each of these parameters to be the unit interval. For the 10×10 Gridworld, the domain of w is $[-1, 1]$, and we use a Normal prior with mean 0 and variance 1 for $w^* = [-1, 1]$, and a Normal prior with mean 0.5 and variance 0.5 for $w^* = [1, 1]$.

For the sepsis treatment simulator, we use a VAE to learn ϕ , a function that transforms the original state representation to a lower dimensional feature representation. The VAE uses 4 linear layers, and is trained using a loss function that minimizes reconstruction error and an Adam optimizer (Kingma & Ba, 2017). To train this VAE, we use samples generated randomly from the original simulator. Once ϕ is known, we can choose w^* to generate the required dataset. Then we learn an optimal policy for $R^*(s, a)$ where $R^*(s, a) = \phi(s, a) \times w^*$. The associated expert demonstrations are the rollouts from this policy. We repeat this procedure for several sets of weights w_0, \dots, w_c to generate the training dataset.

G KD-BIRL Algorithm

For clarity, we include an algorithm box that summarizes how to fit a KD-BIRL posterior (Algorithm 1).

H Calculating Expected Value Difference (EVD)

The procedure to calculate EVD varies depending on the method. Recall that EVD is defined as $|V^{\pi^*, R^*} - V^{\pi(r^L), R^*}|$ where R^* is the data-generating reward function and r^L is the learned reward function. Because KD-BIRL and BIRL both use MCMC sampling, we can use the reward samples generated from each iteration of MCMC. However, AVRIL does not use MCMC, so to sample reward functions from the posterior in this setting, we use the AVRIL agent to estimate the variational mean and standard deviation of the reward in each state of the state space. Using these statistics, we then follow the AVRIL assumption, which states that the reward samples arise from a multivariate normal distribution and generate samples according to the mean and standard deviation.

Once we have samples of the reward function, we can then calculate EVD and 95% confidence intervals. For a given method, we calculate standard error across all sampled rewards from the method. To determine the value of the policy optimizing for a particular reward function, we train an optimal agent for that reward function, and generate demonstrations characterizing its behavior; the value of the policy is then the expected

discounted reward across these demonstrations. Finally, we calculate the difference between the value of the policy for R^* and r^L .

In the sepsis environment, because the state space is not discrete, the above approach will not work for AVRIL. To calculate EVD, we used the trained AVRIL agent to generate behavior trajectories using agent-recommended actions starting from an initial state; the EVD here is the difference between the value of these trajectories and the value of trajectories generated using w^* (independent of AVRIL).

I Gridworld environment

The Gridworld environment’s MDP is defined by the grid’s $g \times g$ discrete state space \mathcal{S} where a given state is represented as a one-hot encoded vector in $\mathbb{R}^{g \times g}$, e_i , where the i ’th index is 1 and corresponds to the state in which the agent is in, and g is the size of the grid; the action space contains 5 possible actions $\{\text{NO ACTION}, \text{UP}, \text{RIGHT}, \text{LEFT}, \text{DOWN}\}$, each represented as a one-hot encoded vector. Here, a reward function R is a vector of length $g \times g$, where each cell represents the scalar reward parameter in each possible state. We use three variations of this environment, which are $g = 2, 5, 10$. Unless otherwise specified, we use Euclidean distance for both d_s and d_r .

J Sepsis environment

The sepsis environment models the management of sepsis within simulated patients (Amirhossein Kiani, 2019). There are 24 possible actions, each corresponding to a different combination of drugs and other treatment options. The actions are represented as integers from 0-24. The state features (listed in Table 1) consist of 46 physiological covariates. In the original environment, nonzero reward is only received in the terminal states (+15 if the patient is discharged successfully, and -15 if the patient dies during treatment). To adapt this environment to be more readily used within an IRL setting, we re-parameterize reward as a linear combination of a weight vector and a low-dimensional feature vector. To generate trajectories from this environment, we train an optimal policy for a given weight vector w^* , and then use the policy to recommend actions from an initialized instance of the environment.

Table 1: Features used in sepsis environment

Feature	Description
Albumin	Measured Albumin
Anion Gap	Measured difference between the negatively and positively charged blood electrolytes
Bands	Measuring band neutrophil concentration
Bicarbonate	Measured arterial blood gas
Bilirubin	Measured bilirubin
BUN	Measured Blood Urea Nitrogen
Chloride	Measured chloride
Creatinine	Measured Creatinine
DiasBP	Diastolic blood pressure
Glucose	Administered glucose
Glucose	Measured glucose
Heart Rate	Measured Heart Rate
Hematocrit	Measure of the proportion of red blood cells
Hemoglobin	Measured hemoglobin
INR	International normalized ratio
Lactate	Measured lactate
MeanBP	Mean Blood Pressure
PaCO2	Partial pressure of Carbon Dioxide
Platelet	Measured platelet count
Potassium	Measured potassium
PT	Prothrombin time
PTT	Partial thromboplastin time
RespRate	Respiratory rate
Sodium	Measured sodium
SpO2	Measured oxygen saturation
SysBP	Measured systolic blood pressure
TempC	Temperature in degrees Celsius
WBC	White blood cell count
age	Age in years
is male	Gender, true or false
race	Ethnicity (white, black, hispanic or other)
height	Height in inches
Weight	Weight in kgs
Vent	Patient is on ventilator, true or false
SOFA	Sepsis related organ failure score
LODS	Logistic organ disfunction score
SIRS	Systemic inflammatory response syndrome
qSOFA	Quick SOFA score
qSOFA Sysbp Score	Quick SOFA that incorporates systolic blood pressure measurement
qSOFA GCS Score	Quick SOFA incorporating Glasgow Coma Scale
qSofa Respirate Score	Quick SOFA incorporating respiratory rate
Elixhauser hospital	Hospital uses Elixhauser comorbidity software
Blood culture positive	Bacteria is present in the blood