A THE PROTOTYPE MODEL

In Section 3.1, we only briefly described the objective function

\[ J(\Theta) = \text{NLL}(\mathcal{D}; \Theta) + \lambda_d R_d(\Theta) + \lambda_c R_c(\Theta) + \lambda_e R_e(\Theta), \]

where \( \text{NLL}(\mathcal{D}; \Theta) \) is the negative log-likelihood, \( R_d(\Theta) \), \( R_c(\Theta) \) and \( R_e(\Theta) \) are regularization terms and \( \lambda_d, \lambda_c \) and \( \lambda_e \) are regularization parameters. \( \Theta \) denotes the set of model parameters, i.e., the parameters of the encoding network \( e \), the weights \( B, c \) and the prototypes \( H \). For a given dataset \( \mathcal{D} = ((h_1, a_1), \ldots, (h_m, a_m)) \), drawn according to a distribution \( p_{\mu} \), the NLL loss of the estimate \( \hat{p}_{\mu} \), parameterized in \( \Theta \), is defined as

\[ \text{NLL}(\mathcal{D}; \Theta) = -\frac{1}{m} \sum_{i=1}^{m} \log \left( \hat{p}_{\mu}(A_t = a_i^t \mid H_t = h_i^t) \right). \]

Furthermore, the regularization terms are defined as follows (see [Ming et al., 2019] for further details):

- The **diversity** regularization
  \[ R_d(\Theta) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \max (0, d_{\text{min}} - d(z_i, z_j))^2, \]
  where \( d(z, z') = \| z - z' \|_2 \), penalizes latent prototypes that are too close to each other. The parameter \( d_{\text{min}} \) is a tunable hyperparameter in our experiments.

- The **clustering** regularization
  \[ R_c(\Theta) = \sum_{h \in \mathcal{D}} \min_{i} d(z_i, e(h))^2 \]
  encourages the encoded histories to approach the most similar latent prototypes, which creates a clustering structure in the latent space.

- The **evidence** regularization
  \[ R_e(\Theta) = \sum_{i=1}^{n} \min_{c \in \Theta} d(z_i, e(h))^2 \]
  encourages the latent prototypes to approach the encodings that are most similar to them.

A.1 PROTOTYPE VALUE

In Section 3.3, we showed how the prototypes can be used to compute the value of a policy \( \pi \) for prototype \( j \) at time \( t \), \( V_{j,t}(\pi) \). Below, we derive a statistical estimator for \( V_{j,t}(\pi) \) using observations under \( \mu \). First, we have

\[ V_{j,t}(\pi) := \mathbb{E}_\pi \left[ \sum_{t' \geq t} R_{t'} \mid J_t = j \right] = \mathbb{E}_\pi \left[ \frac{p(J_t = j \mid H_t)}{p_\pi(J_t = j)} \sum_{t' \geq t} R_{t'} \right]. \]

This equation follows from the fact that \( J_t \) is conditionally independent of all other variables given \( H_t \). Now, with \( W \) importance weights for \( \pi \) and \( \mu \),

\[ V_{j,t}(\pi) = \mathbb{E}_\mu \left[ \frac{p(J_t = j \mid H_t)}{p_\pi(J_t = j)} W \sum_{t' \geq t} R_{t'} \right]. \]

Following standard definitions,

\[ p_\pi(J_t = j) = \mathbb{E}_\pi [p(J_t \mid H_t)], \]

which may be identified using importance sampling,

\[ p_\pi(J_t) = \mathbb{E}_\pi [p(J_t \mid H_t)] = \mathbb{E}_\mu [p(J_t \mid H_t) W_t], \]

with \( W_t = \prod_{t'=0}^{t} p_{\mu}(A_{t'} \mid H_{t'}) / p_\mu(A_{t'} \mid H_{t'}). \) Hence, we may estimate

\[ \hat{p}_\pi(J_t = j) = \frac{1}{m} \sum_{i=1}^{m} p(J_t = j \mid H_t = h_i^t) w_i^t. \]

For trajectories \( i \) which end before \( t \), we let \( \hat{p}(J_t = j \mid H_t = h_i^t) = 0. \)
A.2 IS THERE A GOOD PROTOTYPE MODEL?

In Section 3.1, we asked the question: Assuming that adjusting for the history \( H_t \) is sufficient for unbiased policy evaluation, do there exist prototype histories \( \hat{H} \), an encoding \( e \) and a similarity function \( s \) such that evaluation using the prototype model is accurate? Here follows an example when this is provably the case.

Consider the history at the first time step, \( H_0 = X_0 \in \mathbb{R}^d \). Assume that for each action \( a \), the distribution of histories in which action \( a \) is taken is isotropic Gaussian, \( (H_0 | A_0 = a) \sim \mathcal{N}(\mu_a, \gamma^2) \). Then, the joint distribution of \( (H_0, A_0) \) is a Gaussian mixture model (GMM) with components identified by the actions, \( a = 1, \ldots, k \). We have by the definition of the GMM that

\[
p(A_0 = a | H_0 = h) \propto e^{-\frac{||h - \mu_a||^2}{2\gamma^2}} = s(\mu_a, h),
\]

where \( s \) is defined as in (3). As a result, with the component means \( \hat{X} = [\mu_1, \ldots, \mu_k] \) as prototypes and \( S(\hat{X}, h) = [s(\mu_1, h), \ldots, s(\mu_k, h)]^T \), the behavior policy is given by \( p(A_0 | H_0 = h) = S(\hat{X}, h)/(\sum_k s(\hat{x}_k, h)) \), which matches (4) with \( B = 1/(\sum_k s(\hat{x}_k, h)) \) and \( c = 0 \), up to the application of the softmax function. Furthermore, the prototype assignment probability in (8) is equal to the behavior policy. We state a generalization below.

**Theorem 1.** Assume that there exists a bijective, differentiable encoding function \( e : \mathcal{H} \rightarrow Z \) such that \( \forall t : (e(H_t), A_t) \sim \text{GMM with stationary component means } \{\mu_k\}_{k=1}^K \text{ and variance } \gamma^2 \). Then, with prototypes \( \hat{H} = [e^{-1}(\mu_1), \ldots, e^{-1}(\mu_k)]^T \), and \( S \) as defined above,

\[
\forall t : p(A_t | H_t = h) \propto S(e(\hat{H}), e(h)).
\]

**Proof.** Due to bijectivity, \( p(A_t | H_t = h) = p(A_t | e(H_t) = e(h)) \). The final result follows from the same argument as for the special case of \( H_0 \) above. \( \square \)

Theorem 1 shows that there are indeed problems for which a prototype model that exactly describes the behavior policy exists. This also implies that there exists a prototype estimate of the value \( V(\pi) \) which is unbiased. However, it does not give guarantees for recovering such a model from data, or that the training set contains samples which act well as prototypes. Learning encoding functions \( e \) which satisfy the conditions of Theorem 1 has been studied in the context of normalizing flows [Kong and Chaudhuri, 2020] [Resende and Mohamed, 2015].

B EXPERIMENTAL DETAILS

The prototype approach for off-policy evaluation was evaluated on real-world sepsis data extracted from the MIMIC-III database [Johnson et al., 2016]. In addition, a synthetic environment for sepsis management, provided by [Oberst and Sontag, 2019], was used to study the bias induced by prototypes as a function of the trajectory length. In this section, we give further details about the experiments. To produce the results presented in this paper, we needed about 750 core-hours of computational time. The neural networks were implemented in PyTorch [Paszke et al., 2019] and trained on GPU (Nvidia Tesla T4) using the skorch framework [Tietz et al., 2017]. Other models were implemented using scikit-learn [Pedregosa et al., 2011].

B.1 USING DATA FROM MIMIC-III

We extracted the dataset of sepsis patients from the MIMIC-III database using the code provided by [Komorowski et al., 2018]. This dataset contains the features listed in Supplementary Table 2 in [Komorowski et al., 2018] as well as the total fluid intake and the total urine output for each patient. We also built the AI Clinician using the code provided by [Komorowski et al., 2018]. To evaluate the 500 candidate policies, we used only the MIMIC-III test data and not data from the eCU Research Institute Database.

We used the train-test split associated with the best performing candidate policy in our experiments. We trained and evaluated the estimators of the behavior policy using a subset of the available features: heart rate, systolic blood pressure, diastolic blood pressure, mean blood pressure, shock index, hemoglobin, BUN, creatine, urine output over 4 hours, pH, base excess, bicarbonate, lactate, PaO2/FiO2 ratio, age, Elixhauser index and SOFA score. In addition, we included the treatment dose of vasopressors and IV fluids, respectively, over the previous 4 hours. These values were set to 0 at the initial time steps.

For ProNet and ProSeNet, we selected parameters of the diversity regularization \( (d_{\text{min}}, \lambda_d) \) by performing 3-fold cross-validation over a grid of points in the parameter space \( \{1, 2, 3, 4, 5\} \times \{0.00001, 0.0001, 0.001, 0.01, 0.1\} \). These parameters were optimized for each combination of prototypes \( n \) and prediction prototypes \( q \) in our experiments. The parameters \( \lambda_c \) and \( \lambda_e \) were set to 0.001, and we performed the projection step, see (6), every fifth epoch.

For LR and RF, we searched for optimal models using 3-fold cross-validation, considering the following parameter values:

- LR: regularization: \( \{L_1, L_2\} \); regularization strength: 10 values spaced evenly on a log-scale from \( 1 \times 10^{-4} \) to \( 1 \times 10^4 \);
- RF: maximum tree depth: \( \{5, 10, 15, 20, \text{None}\} \).

To create the model based on post-hoc clustering of en-


F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer,
