Evaluating Policies for Sepsis Management: Decomposing Value Estimates Using Prototypes

Anton Matsson[1], **Fredrik D. Johansson**[1]

[1]Chalmers University of Technology
antmats@chalmers.se, fredrik.johansson@chalmers.se

Abstract

Importance sampling (IS) is often used to perform off-policy policy evaluation but is prone to several issues—especially when the behavior policy is unknown and must be estimated from data. Significant differences between the target and behavior policies can result in uncertain value estimates due to, for example, high variance and non-evaluated actions. If the behavior policy is estimated using black-box models, it can be hard to diagnose potential problems and to determine for which inputs the policies differ in their suggested actions and resulting values. To address this, we propose estimating the behavior policy for IS using prototype learning. We apply this approach in the evaluation of policies for sepsis treatment, demonstrating how the prototypes give a condensed summary of differences between the target and behavior policies. We also describe estimated values in terms of the prototypes to better understand which parts of the target policies have the most impact on the estimates.

Introduction

Historical data on decisions and outcomes provide opportunities for evaluating policies for future decision-making. For example, the prospect of using patient records to evaluate new policies for medication dosing in sepsis management has attracted recent attention (Komorowski et al. 2018; Gottesman et al. 2019). An example of off-policy policy evaluation (OPPE), this amounts to estimating the value of a target policy based on data gathered under a different so-called behavior policy (see e.g., Thomas (2015) for an overview).

Importance sampling (IS) methods (Precup 2000) perform OPPE for decision-making by weighting observed outcomes by the density ratio of the target policy and the behavior policy. IS methods are often preferred over alternatives, which rely on modeling outcomes or covariate transitions, due to their simplicity and that behavior policies often are controllable or human made. However, in practice, it is often difficult to assess the quality of an IS value estimate; see Figure 1 for an example. When the behavior policy is unknown and must be estimated from data, conditions which guarantee good estimates are hard to meet and rely on untestable assumptions (Rosenbaum et al. 2010; Namkoong et al. 2020). Standard practices of inspecting weights (Li, Thomas, and Li 2019) or removing outliers (Crump et al. 2009) give only aggregate or per-sample perspectives on potential issues, which may be insufficient for a domain expert to reason about the result's validity. An additional overview of related work is provided in Appendix E.

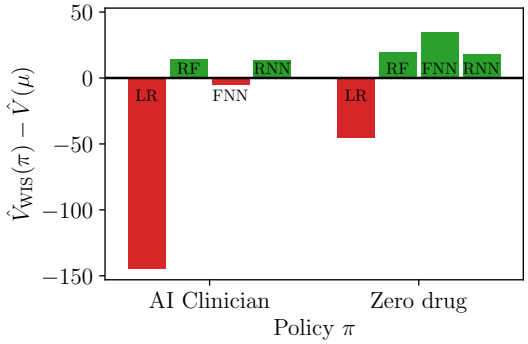

Figure 1: Off-policy policy evaluation of two target policies π for sepsis management: the so-called AI Clinician (Komorowski et al. 2018) and a zero-drug policy. Weighted importance sampling of observations from MIMIC-III (Johnson et al. 2016) was used to estimate the value of each policy based on several models of the unknown behavior policy μ, followed by physicians in data: a logistic regression classifer (LR), a random forest classifier (RF), a feedforward neural network (FNN) and a recurrent neural network (RNN). For each estimator, we plot the difference between the value estimates of behavior and target policies. The results suggest that both target policies may be superior to the behavior policy, especially the zero-drug policy. However, never treating patients with sepsis clearly goes against intuition. With this in mind, can we trust the estimates of the policies' values?

In this work, we study OPPE of sequential decision-making policies using IS with an unknown behavior policy. To enable diagnosis of potential issues in cases where the input space is large or sequential, we propose estimating the behavior policy using prototype learning (Li et al. 2018; Ming et al. 2019). Prototypes are cases from the input data, selected by the learning algorithm and representative of the state-action space. The prototypes should be readily

interpretable by a domain expert, giving transparency to the model. In healthcare applications, prototypes correspond to trajectories of former patients, and prototype-based policies to how physicians use experience from previous patients to treat new ones.

To illustrate our method, we study the management of sepsis, for which several AI-based policies have been proposed (Peng et al. 2018; Komorowski et al. 2018). Examining the target policy of Komorowski et al. (2018), we show that a prototype-based estimate of the behavior policy allows us to more easily describe differences between behavior and target policies, spot violations of overlap and explain which patients contribute to differences in estimated values.

Off-Policy Policy Evaluation

Policy evaluation refers to estimating the *value* $V(\pi)$ of a *target policy* $\pi \in \Pi$. We focus on the sequential case, where a policy is used to select an *action* $A \in \mathcal{A} = \{1, \dots, k\}$ after a *history* $H \in \mathcal{H}$, comprising a sequence of previous actions and *contexts* $X \in \mathcal{X}$. The history until time t is defined as $H_t := (X_0, A_0, X_1, A_1, \dots, X_t)$, with $H_0 = X_0$. A policy $\pi : \mathcal{H} \to \Delta_{\mathcal{A}}$ is a map from a history to a distribution over $\mathcal{A}$. In a medical example, a context X contains information about a patient's state, an action A is a medical intervention, and the target policy π corresponds to new clinical guidelines.

The value of a policy π is defined as the expectation of a *reward* or *outcome* $R \in \mathbb{R}$, accumulated after acting according to π. Here, we study the special case where a single reward is awarded at the end of the sequence, $R = R_T$, but our results generalize to the case where rewards are given after every action. Under the distribution $p_\pi(X_0, A_0, ..., X_T, A_T, R) = p_\pi(H_T, A_T, R)$, induced by the policy π, the value is $V(\pi) := \mathbb{E}_\pi[R]$.

Estimating $V(\pi)$ is trivial given a large enough number of samples from p_π. In off-policy policy evaluation (OPPE), we have access to no such samples, but must estimate $V(\pi)$ using an observational dataset of m samples $\mathcal{D} = ((h^1_{t_1}, a^1_{t_1}, r^1), \dots, (h^m_{t_m}, a^m_{t_m}, r^m))$, drawn according to a distribution $p_\mu(H_T, A_T, R)$, controlled by a *behavior policy* $\mu \in \Pi$. In the medical example, the behavior policy represents current clinical practice. In this work, the behavior policy μ is unknown and an estimate $\hat{p}_\mu(A \mid H)$ is learned from the samples $\mathcal{D}$.

A common method for OPPE is *importance sampling* (IS). The (standard) importance sampling estimator uses an estimate $\hat{p}_\mu$ in a weighted average over the samples $\mathcal{D}$ (Hanna, Niekum, and Stone 2019):

$$\hat{V}_{\mathrm{IS}}(\pi; \hat{\mu}) := \frac{1}{m} \sum_{i=1}^{m} w_i r^i, \tag{1}$$

$$\text{with} \qquad w_i := \prod_{t=0}^{t_i} \frac{p_\pi(A_t = a^i_t \mid H_t = h^i_t)}{\hat{p}_\mu(A_t = a^i_t \mid H_t = h^i_t)}. \tag{2}$$

Under appropriate assumptions, the estimator $\hat{V}_{\mathrm{IS}}(\pi; \hat{\mu})$ is an unbiased estimator of $V(\pi)$ but suffers from high variance when p_μ and p_π differ significantly. The weighted importance sampling estimator (Rubinstein and Kroese 2016), $\hat{V}_{\mathrm{WIS}}(\pi; \hat{\mu}) := \frac{1}{\sum_{i=1}^m w_i} \sum_{i=1}^m w_i r^i$ introduces bias, but often has less variance. Under the Markov assumption, i.e., that context (or "state") transitions, actions and rewards depend only on the most recent context-action pair, the history H_t in (2) can be replaced with the context X_t. We leave out the subscript t where clear.

Can We Trust an IS Estimate?

Sufficient conditions for unbiased value estimation include (sequential) ignorability and overlap (Rosenbaum and Rubin 1983; Robins 1986). As ignorability is untestable, we focus on overlap here. Overlap is satisfied for a history-action pair (h, a) if it being observable under π implies that it is observable under μ, that is, for all t,

$$p_\pi(A_t = a \mid H_t = h) > 0 \Rightarrow p_\mu(A_t = a \mid H_t = h) > 0.$$

A fundamental problem is that the extent of overlap is unknown if μ is unknown. As a result, assessing the quality of an estimate $\hat{V}_{\mathrm{IS}}(\pi; \hat{\mu})$ *inherently relies on evaluation by a domain expert*. It is critical that the OPPE method is transparent enough to enable such evaluation. Standard practices such as inspecting importance weights $\{w_i\}_{i=1}^m$ or estimated action propensities $\hat{p}_\mu(A_t \mid H_t)$ give a per-sample and an average view of potential issues. However, several questions remain unanswered:

A. **Which observations contribute to the IS estimate?** What *signifies* the trajectories and history-action pairs that are assigned high importance weights, $\{i : w_i \gg 1\}$?

B. **In which situations is overlap violated?** When does π suggest actions a that would never be observed under μ? Can we describe the set $\{(h, a) : p_\pi(a \mid h) > 0, \hat{p}_\mu(a \mid h) \approx 0\}$? This is not detectable in $\{w_i\}$ since observing a sample from this set has probability ≈ 0.

C. **If $\hat{V}(\pi) > \hat{V}(\mu)$, what gives π the edge?** In *which* situations does acting according to π result in higher rewards than acting according to μ?

Off-Policy Policy Evaluation With Prototypes

To answer questions A–C, we propose performing OPPE using *prototype learning* (Li et al. 2018; Ming et al. 2019). Let $\tilde{H} = [\tilde{h}^1, \dots, \tilde{h}^n]^\intercal$ be a list of n prototype histories.[1] Each prototype is a *subsequence* of an observed history, $\tilde{h}^j = h^i_t$ for $h^i \in \mathcal{D}$ and $t \leq t_i$. The behavior policy $p_\mu(A_t \mid H_t = h_t)$ is approximated based on the similarity between an observation h_t and the prototypes in a learned representation. The prototypes are themselves selected by the learning algorithm.

To learn $\tilde{H}$, we follow Li et al. (2018); Ming et al. (2019) by first learning a set of latent prototypes as free parameters $\tilde{Z} = [\tilde{z}_1, \dots, \tilde{z}_n]^\intercal$ in an encoding space $\mathcal{Z}$. Given an encoder $e : \mathcal{H} \to \mathcal{Z}$, for an arbitrary history h_t, let

$$S(\tilde{Z}, e(h_t)) = [s(\tilde{z}_1, e(h_t)), \dots, s(\tilde{z}_n, e(h_t))]^\intercal$$

[1] From now, we refer to these as "prototypes".

be the *similarity vector* for the encoding of h_t comparing $e(h_t)$ to $\tilde{Z}$ using a fixed function $s : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}_+$. We use an RBF-kernel with unit bandwidth, $s(\tilde{z}, e(h)) \coloneqq \exp(-d(\tilde{z}, e(h))^2)$, where $d(z, z') = \|z - z'\|_2$, which takes values between 0 (no similarity) and 1 (full similarity). With $B \in \mathbb{R}^{k \times n}$, we estimate the behavior policy μ through logistic regression in the space induced by S,

$$\hat{p}_\mu(A_t \mid H_t = h) = \sigma(BS(\tilde{Z}, e(h)) + c), \tag{3}$$

where σ denotes the softmax function over rows and $c \in \mathbb{R}^k$ is a bias term.

We train the model by minimizing a regularized negative log-likelihood, see Appendix A, using stochastic gradient descent. To make sure that prototypes represent real cases, i.e., to select $\tilde{H}$, latent prototypes are projected onto encodings of training samples at regular intervals between descent steps,

$$\tilde{h}^j \leftarrow \arg\min_{h_t^i \in \overline{\mathcal{D}}} s(\tilde{z}_j, e(h_t^i)) \quad \text{and} \quad \tilde{z}_j \leftarrow e(\tilde{h}^j), \tag{4}$$

with $\overline{\mathcal{D}}$ the set of all subsequences of trajectories in $\mathcal{D}$.

Inspecting Prototype-Based Estimates

The prototypes $\tilde{H}$, their probability coefficients B_j, and estimated action probabilities

$$\hat{p}_j(a) = \hat{p}_\mu(A = a \mid H = \tilde{h}^j) \tag{5}$$

give an overview of the behavior policy estimate. It allows for describing context regions where the probability of certain actions is low (**Questions A and B**) and for describing differences between $p_\mu(A \mid H)$ and $p_\pi(A \mid H)$. As we show below, using prototypes also allows for stratifying the value function $V(\pi)$ to describe which histories contribute to differences in estimated value between two policies π and μ.

We define J_t to be a random variable with values in $\{1, ..., n\}$ representing an assignment of a history H_t to a prototype at time t. We let J_t be distributed according to a normalization of the similarity s,

$$p(J_t = j \mid H_t = h) = \frac{s(\tilde{z}_j, e(h))}{\sum_{k=1}^{n} s(\tilde{z}_k, e(h))}.$$

Furthermore, we define the value $V_{j,t}(\pi)$ of prototype j at time t, obtained under a policy π, as the expected future reward under π given the assignment $J_t = j$,

$$V_{j,t}(\pi) \coloneqq \mathbb{E}_\pi \left[R_T \mid J_t = j \right], \tag{6}$$

which we estimate from finite samples, see Appendix A. With $p(J_t = j)$ the marginal probability of being assigned to prototype j at time t, the product $V_{j,t}(\pi)p(J_t = j)$ represents the contribution to the overall value $V(\pi)$ from histories which are similar to prototype j at time t, effectively stratifying the value by types of situations (**Question C**).

Experiments

We illustrate our method by further examining the example of sepsis management introduced in Figure 1. To produce the results, we replicate the AI Clinician (Komorowski et al. 2018) using patient data from the MIMIC-III database (Johnson et al. 2016). The data are coded as multidimensional time series with a discrete time step of 4 hours. The treatment doses of intravenous (IV) fluids (f) and vasopressors (v) are discretized into 25 possible actions $(f, v) \in \{0, 1, 2, 3, 4\}^2$. We use rewards $r^i = \pm 100$ based on the survival of the patients.

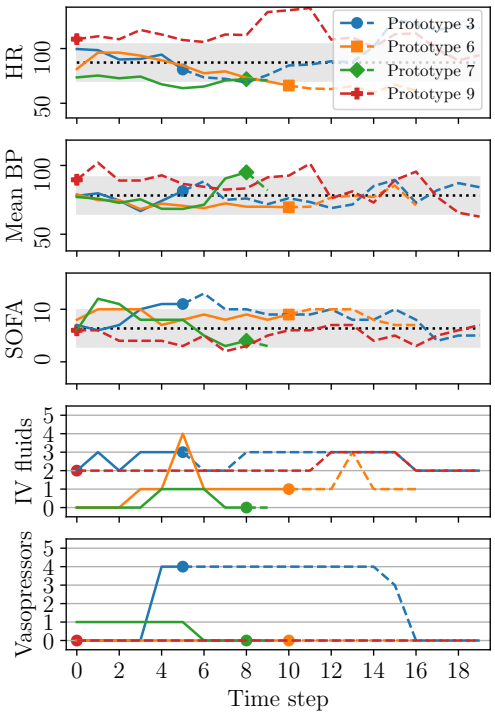

Figure 2: Vital signs and SOFA score plotted against time for four different prototype patients. The lower two panels show the actions taken by the physicians.

As a complement to the models in Figure 1, we learn a prototype model with 10 prototypes and an RNN encoder. When computing $\hat{p}_\mu(A_t = a_t \mid H_t = h_t)$ with the trained model, we ignore all but the two prototypes that are most similar to h_t in the encoding space. Further implementation details are provided in Appendix C and we refer to Appendix D for a discussion of the trade-off between accuracy and transparency for the prototype model. With the prototype-based estimate of the behavior policy, we obtain WIS estimates of the target policies that are comparable to the results in Figure 1 (see Appendix B). That is, both the AI Clinician and the zero-drug policy appear better than the behavior policy followed by physicians. However, leaving all patients untreated is clearly a bad idea—what is going wrong?

While black-box models, e.g., the deep neural networks in Figure 1, make it hard to diagnose potential problems, we can utilize the learned prototypes of the prototype model to reason about the validity of the results. To interpret the prototypes, we can study the trajectories of the corresponding patients. In Figure 2, we plot three key features—heart rate

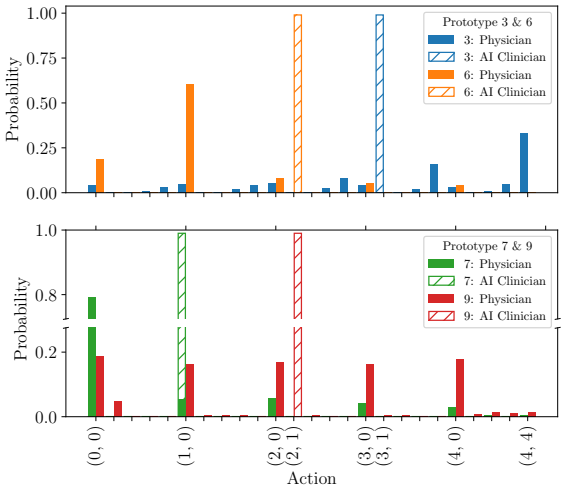

Figure 3: The distribution of actions suggested by the physicians (modeled with the prototype model) and the AI Clinician for prototypes 3, 6, 7 and 9.

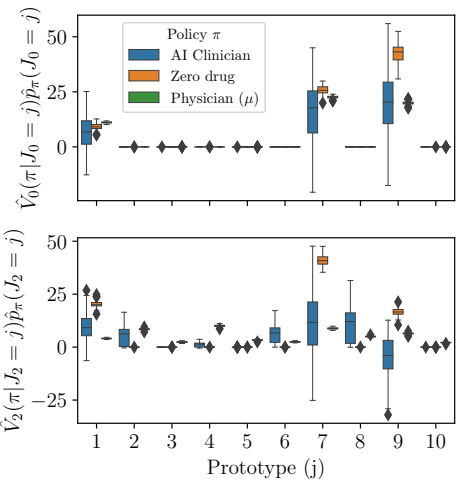

Figure 4: Prototype-based contributions to the overall value $V(\pi)$ at time $t = 0$ (upper panel) and $t = 2$ (lower panel), respectively. We include both target policies and the behavior policy followed by the clinicians. The boxes are always ordered as in the legend.

(HR), mean blood pressure (BP) and SOFA score[2]—as well as the treatment variables against time for four of the prototype patients. Note that the time index of the prototypes are marked with filled markers; for example, prototype 3 is the subsequence ending at time 5 of a patient history. We see for instance that prototype 7 corresponds to a patient who received low doses of IV fluids and vasopressors throughout, and that the prototype 3 patient was treated more aggressively.

By evaluating the target and behavior policies for each prototype, see (5), we quickly get an overview of differences between the policies; see Figure 3 where we compare the AI Clinician with the behavior policy for the selected prototypes. Note that the zero-drug policy always suggests action (0, 0) with probability 1. While this action is likely under the behavior policy for prototype 7, it has low probability for prototype 3. From Figure 2 we may suspect that the corresponding patient, who received aggressive treatment, likely suffered from severe sepsis. It is reasonable to assume that patients represented by prototype 3 are at greater risk of dying and hence contribute with a negative reward if they are included in the value estimate. For the naive evaluation of the zero-drug policy, however, most of these samples are ignored due to lack of overlap, and the estimated value is probably inflated.

In Figure 3, we note a difference between the AI Clinician and the behavior policy for prototype 7, which represents relatively healthy and likely untreated patients. The AI Clinician suggests increasing the dose of fluids for these patients—a rare action under the behavior policy. Such observations are consequently assigned high importance weights. Furthermore, for prototypes 6 and 9, the action (2, 1) of moderate fluids and low vasopressors suggested by the AI Clinician is never observed under the behavior policy—an example of complete lack of overlap. Thus, the estimated value does not reflect this choice of action.

We have now shown how we can use the prototypes to answer Questions A and B stated above. It remains to answer Question C: If $\hat{V}(\pi) > \hat{V}(\mu)$, what gives π the edge? To do this, we can divide the estimated value into each of the prototypes. In Figure 4, we estimate $V_{j,t}(\pi)p(J_t = j)$ for each prototype of the prototype model, for $t = 0$ and $t = 2$. At the initial time step, we see that only prototypes 1, 7 and 9 contribute to the value estimates. This is reasonable since these prototypes correspond to relatively healthy patients (see Figure 6 in Appendix B), and we expect most patients to be relatively healthy from the beginning. Interestingly, both the AI Clinician and the zero-drug policy have higher value than the behavior policy for prototype 9. It should be stressed that this result only applies to the first time step. Already at $t = 2$, we see that the value of AI Clinician for prototype 9 is lower than that for μ. For this time step, we observe a slight positive effect of following the AI Clinician for prototype 6. However, given the lack of overlap between the AI Clinician and μ, see Figure 3 for this prototype, the data may not support evaluating AI Clinician for this prototype.

Conclusion

We have studied off-policy policy evaluation (OPPE) using importance sampling (IS) in the case where the behavior policy μ is unknown and must be estimated from data. We motivated why IS can be difficult and identified three questions about the IS estimate that are likely to remain unanswered when the behavior policy is estimated using black-box models. We proposed performing OPPE using prototype learning to better answer these questions and we illustrated our method for the management of sepsis.

[2]SOFA: Sequential Organ Failure Assessment.

A The Prototype Model

The parameters $\Theta = (e, B, c, \tilde{H})$ of the prototype model, comprising the parameters of the encoder e, coefficients B, c and the set of prototypes $\tilde{H}$, are all unknown and must be learned from data. Following Ming et al. (2019), we do this by minimizing the regularized negative log-likelihood (NLL) of the observed data $\mathcal{D}$ under the model in (3):

$$J(\Theta) = \mathrm{NLL}(\mathcal{D}; \Theta) + \lambda_d R_d(\Theta) + \lambda_c R_c(\Theta) + \lambda_e R_e(\Theta).$$

Here, $R_d(\Theta)$, $R_c(\Theta)$ and $R_e(\Theta)$ are regularization terms and λ_d, λ_c and λ_e are regularization parameters. The regularization terms are defined as follows:

- The **diversity** regularization $R_d(\Theta) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \max\left(0, d_{\min} - d(\tilde{z}_i, \tilde{z}_j)\right)^2$, where $d(z, z') = \|z - z'\|_2$, penalizes latent prototypes that are too close to each other. The parameter $d_{\min}$ is a tunable hyperparameter in our experiments.
- The **clustering** regularization $R_c(\Theta) = \sum_{h \in \mathcal{D}} \min_i \; d(\tilde{z}_i, e(h))^2$ encourages the encoded histories to approach the most similar latent prototypes, which creates a clustering structure in the latent space.
- The **evidence** regularization $R_e(\Theta) = \sum_{i=1}^{n} \min_{h \in \mathcal{D}} \; d(\tilde{z}_i, e(h))^2$ encourages the latent prototypes to approach the encodings that are most similar.

For a given dataset $\mathcal{D} = ((h^1_{t_1}, a^1_{t_1}), \ldots, (h^m_{t_m}, a^m_{t_m}))$, drawn according to a distribution p_μ, the NLL loss of the estimate $\hat{p}_\mu$, parameterized in Θ, is defined as

$$\mathrm{NLL}(\mathcal{D}; \Theta) = -\frac{1}{m} \sum_{i=1}^{m} \log\left(\hat{p}_\mu(A_t = a^i_{t_i} \mid H_t = h^i_{t_i})\right).$$

Predicting With Prototypes

When computing the estimated behavior policy for a history h, the similarity vector $S(e(\tilde{H}), e(h))$ determines how similar each of the n prototypes are to h. The number n is a hyperparameter. The more prototypes are used, the greater the flexibility of the model, but a large n may result in S consisting of multiple elements close to 1, making predictions more difficult to interpret. For example, if $s(e(\tilde{h}_j), e(h)) \approx 1$ for more than 10 prototypes j, it may be difficult to reason about the policy decision after all.

To mitigate this problem, we use only a limited number of $q \leq n$ prototypes—so-called prediction prototypes—when making predictions with the trained model. Let $s_q(h)$ be the similarity between $e(h)$ and its qth most similar latent prototype. For $j = 1, \ldots, n$, we truncate the similarity vector according to

$$s(\tilde{z}_j, e(h)) \leftarrow \begin{cases} s(\tilde{z}_j, e(h)) & \text{if } s(\tilde{z}_j, e(h)) \geq s_q(h), \\ 0 & \text{otherwise.} \end{cases}$$

We perform this step independently for all contexts h.

Prototype Value

Below, we derive a statistical estimand for the value of policy π for prototype j at time t using observations under μ. First, we have

$$\begin{aligned} V_{j,t}(\pi) &:= \mathbb{E}_\pi\left[\sum_{t' \geq t} R_{t'} \;\middle|\; J_t = j\right] \\ &= \mathbb{E}_\pi\left[\frac{p(J_t = j \mid H_t)}{p_\pi(J_t = j)} \sum_{t' \geq t} R_{t'}\right]. \end{aligned}$$

The equation follows from the fact that J_t is conditionally independent of all other variables given H_t. Now, with W importance weights for π and μ,

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

By the law of total expectation, $V(\pi) = \sum_{j=1}^{n} V_{j,t}(\pi) p(J_t = j)$ for any t. 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 $W_t = \prod_{t'=0}^{t} \frac{p_\pi(A_{t'} | H_{t'})}{p_\mu(A_{t'} | H_{t'})}$. 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 $\hat{p}(J_t = j \mid H_t = h^i_t) = 0$.

B Complementary Results

The results from Figure 1 are reproduced with greater detail in Figure 5. Here, we show 100 repeated WIS estimates of the value of the AI Clinician and the zero-drug policy using different estimators of the behavior policy ($r^i = \pm 100$ based on the survival of the patients). The estimated value of μ, $\hat{V}(\mu)$, is included as a reference. The baselines estimators of $p(A \mid H)$, i.e., a logistic regression classifier (LR), a random forest classifier (RF), a feed-forward neural network (FNN) and a recurrent neural network (RNN), are the same as those in Figure 1. The prototype model is denoted "Ours".

As an overview of the relationship between the prototypes of the prototype mpdel, a PCA plot of encoded training data, including the latent prototypes (numbered 1–10) is shown in Figure 6. The colors indicate treatment chosen by μ.

C Experimental Details

To extract the dataset of patients suffering from sepsis from the MIMIC-III database, we used the code provided by Komorowski et al. (2018).[3] This dataset contains the features listed in Supplementary Table 2 in Komorowski et al. (2018)

[3]The code is available at https://github.com/matthieukomorowski/AI_Clinician.

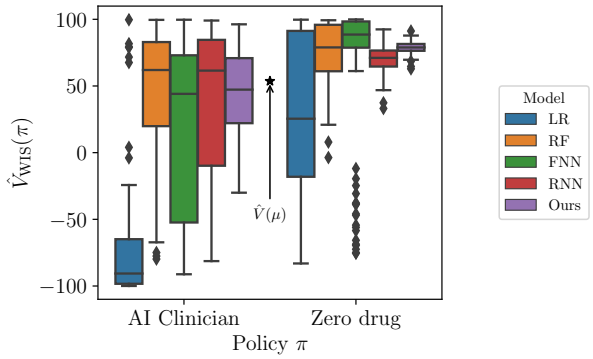

Figure 5: Estimated values of AI Clinician and a zero-drug policy on the test data. The estimated value of the behavior policy μ is included as a reference.

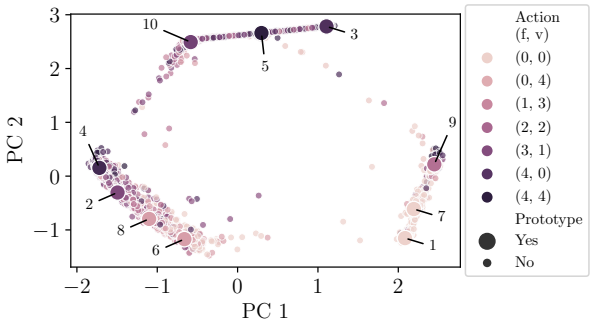

Figure 6: A PCA plot of encoded training samples, colored w.r.t. the action taken by the physicians. The prototypes are numbered 1–10.

as well as the total fluid intake and the total urine output. We learned the target policy, the so-called AI Clinician, using the Matlab code provided by Komorowski et al. (2018). In short, this means clustering the data into 750 states, discretizing the combinations of IV fluids and vasopressors into 25 possible actions, and solving the corresponding Markov decision process using value iteration with rewards based on the survival of the patients. The process was repeated 500 times, each time with a new train-test split (80/20), and the best performing policy on the test set was taken as the target policy, π. To evaluate the 500 candidate policies, we used only the MIMIC test data and not data from the eICU Research Institute Database. We used the train-test split associated with the best performing candidate in our experiments.

We trained 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, $\text{PaO}_2/\text{FiO}_2$ ratio, age, elixhauser score and SOFA score. In addition, we included the treatment doses (vasopressors and IV fluids) over the previous 4 hours; at the first time step, these values were set to 0. For all models except the RNN baseline and the prototype model, we made the Markov assumption, and modeled $p(A \mid H)$ using only the last context-action pair of the history.

Model Parameters

The prototype model was fitted to the training data using $n = 10$ prototypes and an RNN encoder consisting of two layers (each of size 64). The hyperbolic tangent (tanh) function was used as activation function. For the neural network baselines, FNN and RNN, we used similar architectures. The FFN baseline had two layers (each of size 64) and ReLu activation function. The RNN baseline had 2 layers (each of size 64) and tanh activation function. A linear layer was added to the baseline models to obtain predictions of the desired shape.

We trained all neural networks over 400 epochs, using a batch size of 64 for RNN and the prototype model and 1024 for FNN. For optimization, the Adam algorithm was used with default parameters, learning rate 0.001 and weight decay 0.001. For the prototype model, we selected parameters of the diversity regularization $(d_{\min}, \lambda_d)$ by performing 3-fold cross-validation over a grid of points in the parameter space $\{1, 2, 3, 4, 5\} \times \{0.00001, 0.0001, 0.001, 0.01, 0.1\}$. The parameters λ_c and λ_e were set to 0.001, and we performed the projection step, see Equation (4), every fifth epoch.

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

- LR: regularization: $\{\text{L1}, \text{L2}\}$, regularization strength: 10 numbers spaced evenly on a log-scale from 1×10^{-4} to 1×10^{4}
- RF: maximum tree depth: $\{5, 10, 15, 20, \text{None}\}$.

All models were calibrated using sigmoid calibration on a held-out validation set (25 % of the training data).

D Accuracy vs Transparency

We have argued that we can use prototype learning to obtain a transparent estimate of the behavior policy. We can then inspect the prototypes to reason about how well the data supports evaluation of various target policies. However, while introducing transparency, the use of prototypes imposes restrictions on the model, possibly increasing the approximation error. What can be said about the bias induced by the protoypes?

In Figure 7, we compare the accuracy of two types of prototype models—ProNet and ProSeNet—in approximating $p_\mu(A \mid H)$ on the sepsis data for a varying number of prototypes n and prediction prototypes q (see Appendix A). ProNet uses an FNN encoder whereas ProSeNet uses an RNN encoder. For ProNet, we make the Markov assumption and model $p(A \mid H)$ using only the last context-action pair of the history. Overall, the sequential model, making use of the entire history H, performs best, especially for $q = 1$ and $q \geq 4$. Interestingly, the effect of increasing the number of prototypes from 10 to 50 or even 100 is rather small. Using only two prediction prototypes works well for this dataset.

In Table 1, we compare the prototype models with $n = 10$ and $q = 2$ to the baseline models in approximating $p_\mu(A \mid$

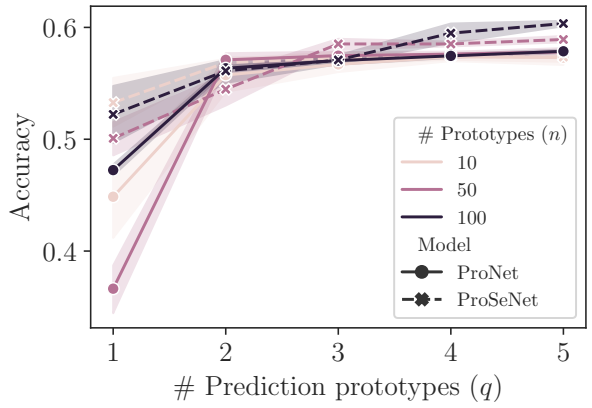

Figure 7: A comparison between ProNet (FNN encoder) and ProSeNet (RNN encoder) using a varying number of prototypes n and prediction prototypes q.

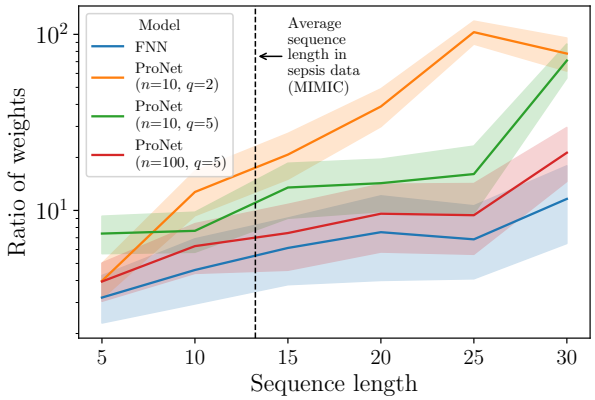

Figure 8: The ratio of the importance weights under $\hat{\mu}$ and μ for increased sequence lengths. We compute $\hat{\mu}$ using both a feedforward neural network and ProNet models with different prototype settings.The bias of using prototypes increases with the sequence length, but for a larger number of training and prediction prototypes, ProNet performs similarly to FNN.

H). Here, we report accuracy, the area under the ROC curve (AUC) and the static calibration error (SCE) (Nixon et al. 2019), a multiclass extension of the expected calibration error. The prototype models are superior to the (regularized) logistic regression (LR) model but they perform slightly worse than black-box models, RF, FNN and RNN. However, as we saw in Figure 7, with increased numbers of prototypes, ProSeNet has the capacity to approach the performance of these models, at least in terms of accuracy.

Bias Due to Increased Sequence Length

According to the results presented above, the use of prototypes introduces a small bias in the estimated propensity. It is natural to ask what this means for the sequential setting, where multiple propensities are multiplied together to form the importance weights. To quantity this effect, we consider the synthetic environment of sepsis management provided by Oberst and Sontag (2019). We use the notebook `learn_mdp_parameters.ipynb` from their GitHub repository to estimate the true parameters of the MPD.[4] We then learn an optimal behavior policy using policy iteration. We refer to the code for further details about the environment.

We collect trajectories of the behavior policy of various lengths, from 5 to 30 time steps, and for each trajectory length, we estimate the behavior policy from data using an FNN as well as ProNet models with different settings for the number of prototypes. Given an arbitrary target policy π, we can estimate $V(\pi)$ using both the true behavior policy and its estimators. Any difference in the value estimates stems from the difference in the importance weights. In Figure 8, we plot the ratio of the weights under $\hat{\mu}$ and μ against the trajectory lengths for different estimators of μ. As the horizon increases, we see that the average weights ratio of the simplest prototype model with $n = 10$ and $q = 2$ greatly separates from that of the plain FNN. However, with additional (prediction) prototypes, the effect is smaller.

To estimate the effect on the value estimate, we learn a target policy π from trajectories of length 15 (close to the average sequence length in the data from the MIMIC-III database) and compute differences between the true value of π and several WIS estimates using the estimators in Figure 8. Note that we can compute the true value of π by simply running it in the simulator. We observe final rewards $r^i = \pm 1$ based on survival of the simulated patients. On average, the estimated value has an absolute difference from the true value that amounts to 0.40 for the FNN (standard deviation 0.24), 0.46 (0.26) for ProSeNet with $n = 10$ and $q = 2$, 0.52 (0.31) for ProSeNet with $n = 10$ and $q = 5$ and 0.44 (0.28) for ProSeNet with $n = 100$ and $q = 5$.

E Related Work

Issues with importance sampling methods for OPPE are well known. Several works aim at describing issues related to high variance (Gottesman et al. 2019), or mitigating them using methodological advances (Precup 2000; Thomas and Brunskill 2016; Jiang and Li 2016; Schneeweiss et al. 2009; Swaminathan and Joachims 2015). Others move the goalposts, using the weights to identify a new study population for which the policy's value can be efficiently estimated (Li, Morgan, and Zaslavsky 2018; Fogarty et al. 2016). Oberst et al. (2020) emphasize the value of interpretability in this endeavour to communicate the generalizability of the estimate. Our method is compatible with each approach, allowing for transparent descriptions of variance issues, identifying new study populations and for use as plug-in estimates.

Interpretability is a an important component of learning systems deployed in increasingly critical functions (Rudin 2019; Lipton 2018). Rule-based estimators, such as rule list (Wang and Rudin 2015) and decision trees, are often fa-

[4]The code is available at https://github.com/clinicalml/gumbel-max-scm/tree/sim-v2.

Table 1: A summary of test-set performance of different estimators of the behavior policy $p_\mu(A \mid H_t)$. For ProNet and ProSeNet, $n = 10$ and $q = 2$. The 95 percent confidence intervals are calculated from bootstraps.

Model	Accuracy (↑)	SCE (↓)	AUC (↑)
LR	0.38 (0.38, 0.39)	0.0112 (0.0110, 0.0115)	0.88 (0.88, 0.88)
RF	0.62 (0.61, 0.62)	0.0037 (0.0034, 0.0039)	0.93 (0.93, 0.93)
FNN	0.61 (0.61, 0.61)	0.0041 (0.0039, 0.0044)	0.93 (0.92, 0.93)
ProNet ($n = 10$, $q = 2$)	0.56 (0.55, 0.56)	0.0069 (0.0067, 0.0072)	0.90 (0.90, 0.90)
RNN	0.62 (0.62, 0.63)	0.0056 (0.0053, 0.0058)	0.94 (0.94, 0.94)
ProSeNet ($n = 10$, $q = 2$)	0.57 (0.57, 0.58)	0.0057 (0.0054, 0.0059)	0.91 (0.91, 0.91)

vored for their short descriptions but generalize poorly to sequential inputs which are the focus of this work. Gottesman et al. (2020) proposed an approach for interpretable OPPE which highlights transitions in data whose removal would have a large impact on the estimate. This approach is related to ours but answers a different set of questions.

Evaluating policies using direct sample-to-sample comparison has a long tradition in policy evaluation through the use of matching estimators of causal effects, see e.g., (Rosenbaum and Rubin 1983; Rubin 2006; Kallus 2020). While favored for its transparency, this approach is typically only used to compare two deterministic policies such as treat-all or treat-none. Matching often relies either on specifying a similarity function in advance or on an estimate of the behavior policy. In high-dimensional settings, this often leads to bias or lost interpretability. Our approach aims to combine the transparency of matching estimators with the flexibility of representation learning methods.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations in this work were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at Chalmers Centre for Computational Science and Engineering (C3SE) partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

References

Crump, R. K.; Hotz, V. J.; Imbens, G. W.; and Mitnik, O. A. 2009. Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1): 187–199.

Fogarty, C. B.; Mikkelsen, M. E.; Gaieski, D. F.; and Small, D. S. 2016. Discrete optimization for interpretable study populations and randomization inference in an observational study of severe sepsis mortality. *Journal of the American Statistical Association*, 111(514): 447–458.

Gottesman, O.; Futoma, J.; Liu, Y.; Parbhoo, S.; Celi, L.; Brunskill, E.; and Doshi-Velez, F. 2020. Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions. In *International Conference on Machine Learning*, 3658–3667. PMLR.

Gottesman, O.; Johansson, F.; Komorowski, M.; Faisal, A.; Sontag, D.; Doshi-Velez, F.; and Celi, L. A. 2019. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1): 16–18.

Hanna, J.; Niekum, S.; and Stone, P. 2019. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, 2605–2613. PMLR.

Jiang, N.; and Li, L. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, 652–661. PMLR.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3: 160035.

Kallus, N. 2020. Generalized optimal matching methods for causal inference. *Journal of Machine Learning Research*, 21(62): 1–54.

Komorowski, M.; Celi, L. A.; Badawi, O.; Gordon, A. C.; and Faisal, A. A. 2018. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature medicine*, 24(11): 1716–1720.

Li, F.; Morgan, K. L.; and Zaslavsky, A. M. 2018. Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521): 390–400.

Li, F.; Thomas, L. E.; and Li, F. 2019. Addressing extreme propensity scores via the overlap weights. *American journal of epidemiology*, 188(1): 250–257.

Li, O.; Liu, H.; Chen, C.; and Rudin, C. 2018. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Lipton, Z. C. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57.

Ming, Y.; Xu, P.; Qu, H.; and Ren, L. 2019. Interpretable and steerable sequence learning via prototypes. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 903–913.

Namkoong, H.; Keramati, R.; Yadlowsky, S.; and Brunskill, E. 2020. Off-policy Policy Evaluation For Sequential Decisions Under Unobserved Confounding. *arXiv preprint arXiv:2003.05623*.

Nixon, J.; Dusenberry, M. W.; Zhang, L.; Jerfel, G.; and Tran, D. 2019. Measuring Calibration in Deep Learning. In *CVPR Workshops*, volume 2–7.

Oberst, M.; Johansson, F.; Wei, D.; Gao, T.; Brat, G.; Sontag, D.; and Varshney, K. 2020. Characterization of overlap in observational studies. In *International Conference on Artificial Intelligence and Statistics*, 788–798. PMLR.

Oberst, M.; and Sontag, D. 2019. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, 4881–4890. PMLR.

Peng, X.; Ding, Y.; Wihl, D.; Gottesman, O.; Komorowski, M.; Lehman, L.-w. H.; Ross, A.; Faisal, A.; and Doshi-Velez, F. 2018. Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning. In *AMIA Annual Symposium Proceedings*, volume 2018, 887. American Medical Informatics Association.

Precup, D. 2000. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, 80.

Robins, J. 1986. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12): 1393–1512.

Rosenbaum, P. R.; and Rubin, D. B. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1): 41–55.

Rosenbaum, P. R.; et al. 2010. *Design of observational studies*, volume 10. Springer.

Rubin, D. B. 2006. *Matched sampling for causal effects*. Cambridge University Press.

Rubinstein, R. Y.; and Kroese, D. P. 2016. *Simulation and the Monte Carlo method*, volume 10. John Wiley & Sons.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215.

Schneeweiss, S.; Rassen, J. A.; Glynn, R. J.; Avorn, J.; Mogun, H.; and Brookhart, M. A. 2009. High-dimensional propensity score adjustment in studies of treatment effects using health care claims data. *Epidemiology (Cambridge, Mass.)*, 20(4): 512.

Swaminathan, A.; and Joachims, T. 2015. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, 814–823.

Thomas, P.; and Brunskill, E. 2016. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, 2139–2148. PMLR.

Thomas, P. S. 2015. *Safe reinforcement learning*. Ph.D. thesis, University of Massachusetts Libraries.

Wang, F.; and Rudin, C. 2015. Falling rule lists. In *Artificial Intelligence and Statistics*, 1013–1022. PMLR.