Towards High Confidence Off-Policy Reinforcement Learning for Clinical Applications

Abhyuday Jagannatha [1] **Philip Thomas** [1] **Hong Yu** [1 2]

Abstract

We study the properties of off-policy reinforcement learning algorithms when applied to a real world clinical scenario. Towards this end, we evaluate standard off-policy training methods on ventilation and sedation control using the MIMIC-III dataset. We further evaluate off-policy policy evaluation (OPE) methods in the context of this problem. We analyze the limitations exhibited by these methods and propose possible solutions.

1. Introduction

Off-policy reinforcement learning refers to the task of learning a policy while using a possibly different behavior policy to interact with the environment. This class of Reinforcement Learning (RL) algorithms can be trained using retrospective data, and therefore is of particular interest in the context of clinical applications. For example, the training data could be collected through behavior policies employed by clinicians in a hospital. This property allows off-policy RL methods to build a model of the action policy without being actively interacting with a patient. However, methods in this class come with their own set of challenges. The two main challenges are how to efficiently learn the action policy from a limited dataset (which is quite common in clinical settings) and how to evaluate the learned policy without any real-world usage.

Several methods have been proposed for off-policy learning such as Q-learning (Sutton & Barto, 1998), off policy actor-critic (Degris et al., 2012) and their variants (Gu et al., 2016; Munos et al., 2016). The main methods for off-policy policy evaluation (OPE) are based on either model estimation (Hallak et al., 2015) or importance sampling (Precup et al., 2000; Thomas et al., 2015; Jiang & Li, 2015). The model estimation methods work by forming an estimate of the underlying Markov decision process (MDP). They are more sample efficient (Hester et al., 2010; Hallak et al., 2015), however it may be difficult to quantify the *approximation error* or the *model bias* of these methods. It is often even more difficult to choose an appropriate function class in the clinical domain due to its complexity. The second evaluation method uses importance sampling in order to build an estimate of the expected return (Precup et al., 2000; 2001). These importance sampling estimates are unbiased, but they can have very high variance leading to unstable evaluations.

It is also imperative that OPE methods provide performance guarantees before being deployed in the real world. This is particularly important when using a high variance estimator like importance sampling. Such guarantees can be provided by calculating confidence bounds on the OPE return. Statistical tools like concentration inequalities (Thomas et al., 2015) or bootstrap confidence bounds (Hanna et al., 2017) can be used to calculate lower bounds on off policy return.

Reinforcement learning methods have been used to learn policies for several clinical applications. Ernst et al. (2006) use fitted Q-iteration (FQI) (Ernst et al., 2005) to obtain treatment policies for simulated HIV data. Prasad et al. (2017) use FQI with Gaussian process based preprocessing on ICU data for ventilation and sedation control. Nemati et al. (2016) use Q-learning with a partially observable MDP formulation to learn a policy for heparin dosage. Raghu et al. (2017) use Q-learning to optimize policy for sepsis treatment.

In this work we examine the limitations of standard off-policy RL and OPE methods for clinical applications. Towards this end, we apply off-policy RL methods in a clinical scenario. We enumerate a set of properties that are necessary for a clinical RL algorithm. We design our experiment with these guidelines using standard, widely used RL and OPE methods. We analyze the properties and limitations of these methods. Lastly, we discuss possible extensions to them in order to improve their utility in the clinical domain.

[1]College of Information and Computer Sciences, University of Massachusetts, Amherst [2]Department of Computer Science, University of Massachusetts, Lowell. Correspondence to: Abhyuday Jagannatha <abhyuday@cs.umass.edu>.

CausalML workshop, ICML 2018.

2. Nomenclature

We adapt the nomenclature used by Thomas et al. (2015). We use $\mathcal{S}$ and $\mathcal{A}$ to denote state and action spaces. $\pi(a|s;\theta)$ denotes a policy or a probability density of taking action a, conditioned on the state s and parameterized by θ. The shorthands π_b and π_e denote the behavior and evaluation policy respectively. In a clinical application we are provided the data set D, that consists of K trajectories (patient histories). Each trajectory τ_i is composed of sequence of state, action and reward.

Rewards for a trajectory τ_i at time-step t are denoted by $r_{t,i}$. The total reward throughout the trajectory τ_i is denoted by r_i.

3. Properties of a Clinical RL Algorithm

We enumerate a set of properties that are required in an off-policy RL algorithm for a clinical application. This is not an exhaustive list, since we focus on properties relevant to our current model. For example, properties like interpretability of results are outside the scope of this work.

Off policy: As discussed in the introduction section, Off-policy is a *de facto* model in the clinical domain due to its reliance on retrospective data.

Confidence bounds: To deploy an off-policy method in the real world, it needs to provide "safety" guarantees. This is especially true for applications in clinical settings, where a sub-optimal action can have extremely negative consequences. These guarantees can be provided in terms of lower bounds on evaluation metrics using methods like concentration inequalities or bootstrap.

Estimation of underlying models: Both class of methods used for off-policy evaluation depend on estimation of underlying model dynamics. In the case of model based evaluation, it is the estimate of the underlying MDP. Whereas, in the case of importance sampling methods it is the estimate of the behavior policy. Care should be taken to reduce approximation and estimation errors in both cases.

For importance sampling methods, a few additional conditions must be met. The behavior policy should be *soft* (i.e., it should have a non zero probability $\pi_b(a|s;\theta)$ for any action $a \in \mathcal{A}$ and state $s \in \mathcal{S}$). It may also be possible that the data is obtained through multiple behavior policies, instead of a single one.

Action Space: Clinical therapies involve decisions in both discrete and continuous action spaces. Decisions like intubation or dosage of oral medications can be modeled through discrete action spaces. But actions like dosage rates of IV drugs entail continuous action spaces. Additionally, clinical applications may have very large action spaces, due to a large number of drug and procedure choices. A clinical RL method should therefore be able to deal with continuous or large action space.

Long horizon: Hospital stays for critical patients in ICU wards or even out-patient therapy typically spans long time intervals. For example, ICU stays in the MIMIC-III dataset (Johnson et al., 2016) have a median length of 2.1 days while a median hospital stay is 6.9 days. In an ICU, where several decisions are taken within an hour, this could translate to a very long horizon. The variance of importance sampling estimators may grow exponentially with the length N of the time horizon (Glynn, 1994; Guo et al., 2017) which increases the sample complexity of these methods exponentially. Therefore an off-policy algorithm and its evaluation method should be designed to deal with a long horizon.

4. Dataset Description

For our analysis we use a simplified version of the clinical application used by Prasad et al. (2017). They look at the problem of ventilation management and sedation regulation in ICU patients gathered from MIMIC-III (Johnson et al., 2016) data. Mechanical ventilators are medical devices which help critical care patients breath by assisting in ventilation of air to and from the patient's lungs. The process of mechanical ventilation is initiated and finished by intubation and extubation procedures respectively.

Weaning a patient off the ventilator is a challenging task. Extubating the patient before their respiratory system is strong enough to function on their own, leads to repeated re-intubations which increase the chance of complications. On the other hand, ventilating the patient longer than necessary increases discomfort and elongates ICU stay. As noted by Prasad et al. (2017) a successful extubation event in retrospective data only provides an upper bound on the time that the patient was ready for extubation. However, repeated intubation and extubation indicate a sub-optimal control policy. Therefore, to simplify this analysis, we compose our reward function based only on the number of failed extubation procedures. A reward is provided only at the end of each trajectory, and no reward is given for the intermediate states. So if a trajectory only has one extubation procedure, we give it a final reward of 0. For a trajectory with k failed extubations ($k+1$ total extubations) we give final reward of $-k$. Each end reward was offset by $+0.8$ in order to approximately center the reward returns.

While on mechanical ventilation, these patients also require sedatives to reduce discomfort and improve patient safety. However, providing patients with the correct sedative and dosage is also a non-trivial task. Both over-sedation and under-sedation are detrimental to the overall health of the

patient.

We limit our action space to only sedatives and intubation/extubation choices. We use six commonly used sedatives in the ICU data. To further simplify our scenario, we do not consider the dosages for each sedative. This reduces our action space to only seven binary decisions. Six action variables controls whether that drug will be administered. The seventh binary variable, denotes whether the patient is on mechanical ventilation. Overall this leads to 128 unique action combinations.

We use eighteen input features which are relevant to our problem. The features are Arterial pH, Inspired O2 Fraction, O2 Flow (lpm), PEEP Set, Plateau Pressure, Mean Airway Pressure, Arterial CO2 Pressure, Systolic Blood Pressure, Diastolic Pressure, Peak Inspiratory Pressure, Mean Blood Pressure, Spontaneous Respiratory Rate, Respiratory Rate, Tidal Volume, SpO2, weight, age and gender. These features are sampled at different frequencies. To deal with irregularly sampled time series data, we use a simple interpolation scheme that uses the most recent value for each feature. The input features are normalized by scaling $[\mu - 3\sigma, \mu + 3\sigma]$ to $[-1, 1]$.

In order to reduce the number of time steps in the trajectory, we split the time-line based on when a change in action space variables was recorded in the data. This changes our problem formulation by only requiring a decision when one was made in the true dataset. However, in the context of our analysis of off-policy evaluation, this simplification should not introduce any bias. Additionally, this simplification does not completely eliminate the long horizon problem. Our processed dataset still has a significant number of trajectories that are longer than 100 time steps. In total we extracted 4436 patient trajectories with an average trajectory length of 101.2.

5. Methods

We used two off-policy learning methods, namely FQI and direct policy search to understand the challenges in using RL algorithms for clinical applications. The policies generated through these methods were evaluated using weighted importance sampling (Precup et al., 2000). We use bootstrap sampling to provide the percentile bootstrap confidence lower bounds. The methodology and the relevant challenges are described in the following subsections.

5.1. Fitted Q Learning

We use Fitted Q Iteration (Ernst et al., 2005) to estimate the action-value function for our problem. Similar to Q-Learning, FQI can be used in an off-policy setting. It uses tuples of (s_t, a_t, r_t, s_{t+1}) with the aim of estimating q^*,the optimal action value function, irrespective of the behavior policy π_b used to create the dataset. For further details about FQI refer to Ernst et al. (2005).

We use a simple one-layer neural network with approximate RBF kernel feature maps (Rahimi & Recht, 2008) to estimate the Q function. To simplify inference, the output dimensionality of the neural network is equal to number of possible action combinations. In our problem formulation we have seven binary actions as described in Section 4 and therefore we have 128 possible unique action combinations. Only 75 of these action combinations are used in our dataset, so our Q network has a 75 dimensional output.

FQI uses a regression algorithm to iteratively update the Q estimate towards the Q value calculated from the next step. This broadly translates to running the following update for each (s_t, a_t, r_t, s_{t+1}) tuple in the dataset.

$$Q^{i+1}(s_t, a_t) \overset{FQI}{\longleftarrow} r_t + \gamma \max_{a'} Q^i(s_{t+1}, a') \tag{1}$$

Here, Q^i is the approximation obtained by the regression algorithm for the i^{th} iteration of FQI and γ is the discount parameter. In the $(i+1)^{th}$ FQI iteration, the regression algorithm uses the right hand side of (1) as the target.

It has been well documented (Glorot & Bengio, 2010) that initialization of the neural network has a significant effect on its performance. While single layer linear networks are not affected by saturation regions, due to the rarity of certain actions combinations in the clinical domain it is still important to carefully initialize the weights and biases of the Q network. For example, in our problem formulation, the reward can be between $[-9.2, 0.8]$ as described in Section 4. Consider the case where the initial biases of our Q network are kept at a positive value, while the weights are scaled using Xavier initialization (Glorot & Bengio, 2010). Notice that the right hand side of (1) has a max over all actions, but since it is the target of the FQI update, the max action itself is never updated.

This phenomenon can cause issues with FQI training if there are certain action combinations (say a_{max}) that are rarely present in the dataset. Since the network bias corresponding to a_{max} is updated very slowly, it remains at the relatively high initialization value. Therefore, for most states the max operation on the RHS of (1) will return $Q^i(s_t, a_{max})$. This results in very slow or stalled training if a_{max} is extremely rare. It is possible to encounter such rare action combinations, especially in the large action space of a clinical scenario. Therefore, it is important to offset the biases so that the initial output of Q-network underestimates the actual Q value. We accomplish this by initializing all biases to large negative values and initializing the weights using variance scaling.

We train the Q network till convergence using mini-batch stochastic gradient descent. The convergence is estimated

by calculating the mean squared TD error on a validation set. The exact procedure for validation and convergence is described in Section 6.

5.2. Direct Policy Search

The second method uses a gradient free optimization method to search for a policy that maximizes the return over a training dataset D_t.

$$\theta' \in \underset{\theta}{\text{argmax}} \quad J(\pi_\theta | D_t) \tag{2}$$

Here, π_θ is a policy function parametrized by θ. The function $J(\pi|D_t)$ evaluates the policy π over the training dataset and outputs a return. It is formally defined in Section 5.4. We can use black box optimization algorithms like Evolution Strategy optimizers, to search for the policy which can maximize the return $J(\pi|D_t)$.

In this work, we use CMA-ES (Hansen et al., 2003) to find suitable parameters for the policy function. The general design of the policy function is the same as that used in Section 5.1. We use approximate RBF kernel maps to generate the input representation which is then fed into a single layer neural network. However, the output size of this network is only seven instead of 2^7. Since this method uses the policy estimate directly, it does not require a max over all action combinations for inference. This significantly reduces its complexity during inference as compared to action-value methods like Q learning. Another advantage of this method is that it can model problems with both continuous or categorical action spaces. We use an identical discrete action space in order to compare this with FQI.

To normalize all seven outputs as individual action probabilities, we use a sigmoid function as the output activation layer. The probability of best action is obtained by taking a product over all seven individual action probabilities.

5.3. Estimation of Behavior Policy

As mentioned in Section 3, estimation of the behavior policy is an important step in IS based OPE methods (Precup et al., 2000). The aforementioned WIS and IS estimators use the behavior policy probabilities $\pi_b(a_t|s_t)$ to calculate the weights. We use Kernel density estimation to estimate the conditional $\pi_b(a_t|s_t)$. Kernel density estimation (Aitchison & Aitken, 1976; Bowman, 1980) is a non-parametric estimation method which is biased but consistent. We use a KDE formulation from Li & Racine (2003) for categorical action variables and continuous state variables.

The histogram of behavior policy estimated from training data and evaluated on a held out test set is shown in Figure 1. Since, we have 75 valid action combinations, a uniform random probability over all actions would amount to a probability estimate of 0.0133. The figure shows that KDE estimates are generally much higher than 0.0133 for (state, action) pairs in the test dataset. This suggests that it is able to successfully fit the behavior policy.

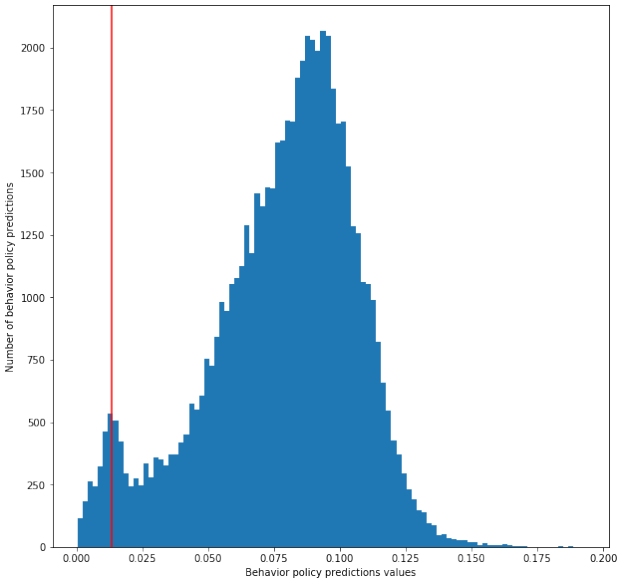

Figure 1. Histogram of behavior policy predictions $\pi_b(a_t|s_t)$ for all state, action pairs in the held-out test set. The red vertical line is at $x = 0.0133$ which is the value of a uniform random distribution over actions.

5.4. Importance Sampling

Importance sampling can be used to produce an unbiased estimate of the expected return of the evaluation policy π_e, using trajectories that were constructed using a different policy π_b. While importance sampling is unbiased if π_b is known, it is shown to have very high variance. Weighted Importance Sampling (WIS) estimator drastically reduces this variance, but is a biased estimate. Nevertheless, the biased WIS estimator is more useful in practice due to its limited variance.

In our case since we do not explicitly know π_b and our KDE estimate is biased, the IS estimator is also biased. Therefore to reduce the variance we use WIS to estimate the returns of our evaluation policies. Additionally, since we only have a reward at the end of a trajectory, per-decision WIS (Precup et al., 2000) and WIS estimators are the same. However, in cases where the reward is distributed throughout the trajectory, per-decision WIS estimators should be preferred. The IS weight of a trajectory τ_i in the dataset D using evaluation policy π_e and behavior policy π_b is

$$w(\tau_i, \pi_e, \pi_b) = \prod_{t=1}^{T_i} \frac{\pi_e(a_t|s_t;\theta)}{\pi_b(a_t^i|s_t^i)}. \tag{3}$$

The WIS return $J(D, \pi_e, \pi_b)$ calculated on a dataset D is

$$J(D, \pi_e, \pi_b) = \frac{\sum_{i=1}^{K} r_i w(\tau_i, \pi_e, \pi_b)}{\sum_{i=1}^{K} w(\tau_i, \pi_e, \pi_b)}. \quad (4)$$

Mixed Policy Evaluation

The deterministic policy provided by Q-learning or FQI is incompatible with off-policy policy evaluation (OPE) because it will reduce $w(\tau_i, \pi_e, \pi_b)$ to zero, if even one of its predicted actions deviates from the behavior policy. The policy obtained from CMA-ES is not deterministic, but can still produce small enough values that cause numerical instability while calculating $J(D, \pi_e, \pi_b)$. To mitigate these problems, we use a mixed policy (Kakade et al., 2003) in order to estimate the WIS return. The mixed policy is calculated as

$$\pi_m = \alpha\pi_e + (1 - \alpha)\pi_b. \quad (5)$$

We use $\alpha = 0.5$ in our case. Using a mixed policy also ensures that the evaluation policy is closer to the behavior policy. This reduces the variance of the importance sampling estimator. The WIS returns discussed in later sections denote the return obtained through the mixed policy $J(D, \pi_m, \pi_b)$.

5.5. Confidence Intervals

We use bootstrap samples to calculate the confidence intervals. Bootstrapping does not have any assumptions on the distribution of the samples, and is suitable for our task. Bootstrap methods have been used in reinforcement learning (Hanna et al., 2017) as well as biomedical applications (Forman et al., 2004; Njeh et al., 2000; Rochon et al., 2008). We adapt the bootstrap method used by Hanna et al. (2017) to calculate the 95% confidence lower bound for our off-policy methods.

We modify the procedure as shown in Algorithm 1. Both FQI and CMA-ES are used to produce the evaluation policies.

Algorithm 1: Confidence Intervals for evaluation policy π_e

input: Evaluation policy π_e,Test Dataset D_T with N trajectories, a behavior policy π_b, a confidence value $\delta \in [0, 1]$, number of bootstrap samples M, and policy mixing parameter α

output: $(1 - \delta)$ confidence lower bound for WIS return on D_T

for $i = 1$ **to** M **do**
  $\tilde{D_T^i} \longleftarrow \{\tau_1^i, ..., \tau_N^i\}$ where $\tau_j^i \sim \mathcal{U}(D_T)$ // $\mathcal{U}$ is the uniform distribution
  $\pi_m \longleftarrow \alpha * \pi_e + (1 - \alpha) * \pi_b$
  for $j = 1$ **to** N **do**
    $w_j^i \longleftarrow$ **IS-weight**(τ_j^i, π_m, π_b)
    $W_j^i \longleftarrow \text{Clip}(w_j^i, 10^{-20}, 10^3)$
  end for
  $\hat{V}_i \longleftarrow$ **WIS-estimate**$(W^i, \tilde{D_T^i})$ // Weighted importance sampling return
end for
sort$(\{\hat{V}_i | i \in [1, M]\})$ // Sort ascending
$l \longleftarrow \lfloor \delta M \rfloor$
Return $\hat{V}_l$

6. Experiments and Results

We randomly shuffle the dataset and select 600 trajectories as the held out test set. The remaining trajectories form the training set for the algorithms. Both FQI and CMA-ES are trained for 100 epochs or till convergence criteria are met. We treat a small split from the training data as the validation set to tune the hyper-parameters and to calculate convergence. In the case of FQI, the learning-rate, the learning rate scheduler and the l2 weight penalty are the tunable hyper-parameters. The γ parameter for FQI in (1) is fixed at 0.9. In the case of CMA-ES, only the σ is tuned. In both CMA-ES and FQI, we run multiple training session for each hyper-parameter combination and select the best policy based on the validation set.

The policy obtained from these training methods is mixed with the KDE estimated behavior policy to obtain π_m from (5). The mixed policy π_m is then used to estimate the importance sampling weights for the trajectories in the test dataset. Figure 2 plots the histogram of log importance sampling weights. The most important observation in this figure is the distribution tail on the right side. It denotes a few values that are between 20-50. This implies that importance sampling weights for certain trajectories approach extremely large values. In both IS and WIS estimators, such a large IS weight will dominate all other values. In order to avoid this, we clip all importance weights at 1000. This reduces the effect of the large outliers.

Using Algorithm 1 we can now obtain the bootstrap distributions of WIS returns for both FQI and CMA-ES on our test split. The mean and 95% confidence bounds for these are shown in the Table 1. A histogram of the returns from 1000 bootstrap samples are also provided in Figure 3. The mean reward under the behavior policy is -0.0216. Based on these results, the bootstrap bounds for WIS returns are well above the mean reward under behavior policy.

It is important to note that the mean importance sampling return from the bootstraps is very low for CMA-ES. This can be explained by the fact that we use WIS return as the optimization objective. So it only searches for a parameter combination which maximizes the WIS return, irrespective

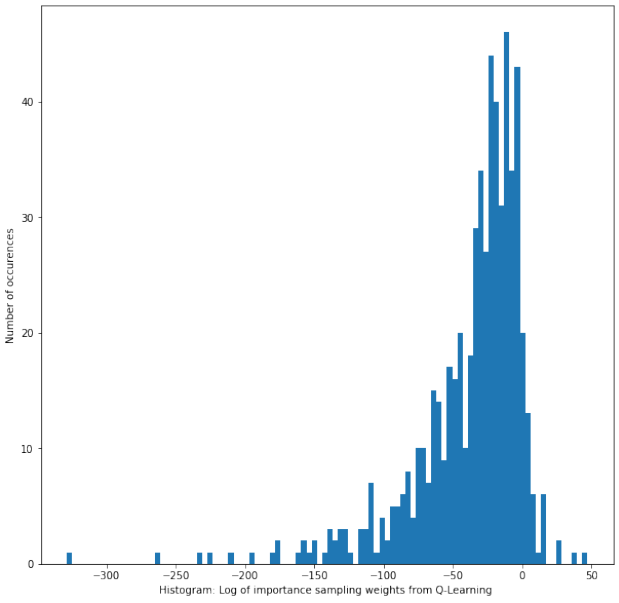

Figure 2. Histogram of log importance sampling ratios. The evaluation policy used is a mixed policy π_m where the evaluation policy is trained using FQI. α is 0.5

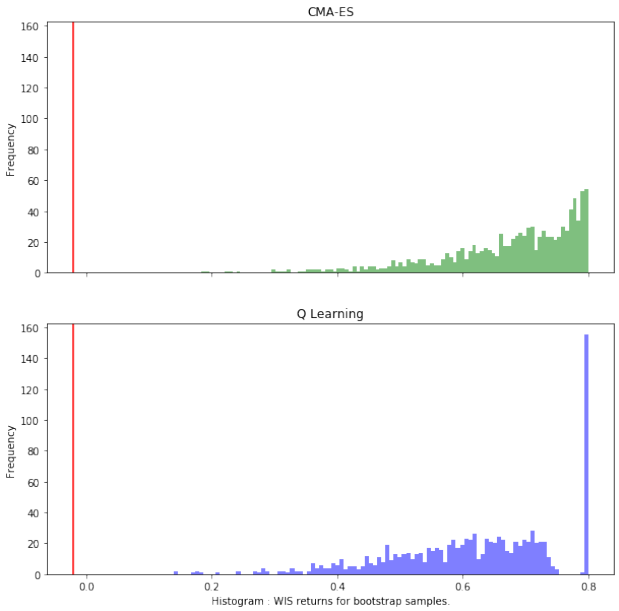

Figure 3. Histogram of WIS returns for bootstrap samples of the test dataset. The policy evaluated is a mixed policy π_m. Here the evaluation policy is trained using Q-Learning(FQI) and CMA-ES. α is 0.5. The red line indicates the average reward return under the behavior policy.

METHOD	WIS CB	WIS MEAN	IS MEAN
BEHAVIOR POLICY	—	—	-0.0216
CMA-ES	0.4504	0.6744	0.0045
FQI	0.3766	0.6269	13.0724

Table 1. This table shows 95% confidence lower bound and mean of the WIS return for both algorithm. The median and confidence lower bounds are calculated from 1000 bootstrap samples. The mean of IS returns for these samples are also provided. The mean reward under the behavior policy is -0.0216.

of whether the result is actually a better policy. In order to obtain meaningful policies with CMA-ES or similar policy search methods a more specialized objective function needs to be used. FQI or Q Learning in this regard seems to behave better than CMA-ES. It leads to improved WIS and IS returns even though it does not directly optimize for them. The histogram for Q Learning in Figure 3 is bi-model with a very sharp peak near the max-reward. One interpretation of this maybe that several examples in our test set produce very high IS weights. Even though these weights are clipped at 1000, they still dominate the WIS return in their bootstrap samples and push the result near to its maximum possible value of 0.8.

7. Discussion

In the previous sections we have used off-policy model-free training methods to propose policies for our clinical scenario. These policies have been tested by obtaining bootstrap based lower bounds for WIS returns. However, there are certain other conditions that still need to be satisfied in order to ensure that we cover all the aspects required in a good RL algorithm for clinical applications.

One important requirement is to ensure our evaluations are stable even when the horizon is long. In our experiments we have clamped the values of WIS to reduce the variance of IS weights. However, a few IS weights clamped at 1000 can still dominate the WIS return and overshadow the contribution of other examples. This can be seen in the Figure 2 where a very small portion of the IS weights attain values near a thousand or higher. A large majority of weights are closer to one or below it, but they contribute much less to the WIS return than small minority of high IS weights. This is also evidenced in the bimodal histogram for Q-learning in Figure 3. In real life clinical scenarios, the horizon may be much longer than our mean horizon length of 101.

A possible solution to this would be to clip the IS weights to even smaller values, but that would result in severely limiting the return of our evaluation policy in all cases irrespective of their contexts. Another possible solution might be to ensure that the learned policy only deviates from the behavior policy for a small number of states in the trajectory. Our predictions currently deviate from the behavior

policy almost continuously during the episode length. This is evident from the Figure 4 where the cumulative product of likelihood ratios is continuously changing. The second trajectory in Figure 4 is an extreme example of this. Here the cumulative product first reaches very low values and then shifts upwards to reach a very high IS weight at the end of the trajectory. The behavior of CMA-ES policy is also similar to that shown in Figure 4.

If we bias our prediction model to avoid frequent deviations, it may naturally reduce the variance of the IS estimator. Additionally, if only a few π_m values differ from π_b in a very long trajectory, most of the likelihood ratios will be very close to one. This reduces the long horizon problem in IS estimators. It is also easier for end users to interpret the differences in behavior and evaluation policy if they diverge only for a few states in the trajectory. One caveat of this assumption is it may lead to bad solutions if the behavior policy is very far from optimal. However in clinical scenarios, we can safely assume that the policies used by trained clinicians is most likely close to optimal.

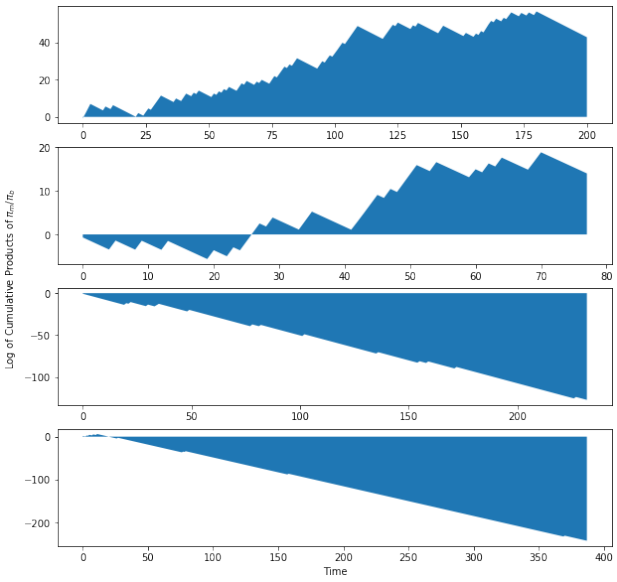

Figure 4. This graph shows the progression of unclipped per-decision IS weights (cumulative product of π_m/π_b ratios) over four different trajectories in our test dataset. The evaluation policy used is π_m, the mixed policy obtained with FQI training. The four plots shown here are examples of trajectories with extreme IS weight predictions. The top two episodes a have very high IS weight and the bottom two episodes have very low IS weights.

The design of the training algorithm also needs careful consideration. FQI, which has been very effective in our off-policy experiments, and similar action value methods do not handle continuous action spaces well. On the other hand, direct policy search can work well with continuous action spaces but without a well designed objective customized to the problem, it may not produce a meaningful policy. Another class of methods based on Actor Critic models (Degris et al., 2012) can also be considered.

Other possible improvements include behavior policy estimation. In this work we assume all decisions are taken through a common behavior policy. Depending on the dataset, this assumption can be relaxed to estimate several physician specific behavior policies. This may help account for physician specific trends. Lastly, clinical applications can also be treated as a partially observable MDP instead of the MDP formulation we use. This may be helpful because it is often not possible to account for all variables in a clinical or biomedical application.

8. Conclusion

We have used a simplified clinical scenario to analyze the performance of FQI and CMA-ES in the context of clinical applications. We show that these off policy algorithms need to be further improved, in order to be effective enough for real world deployment. We also show that variance reduction measures such as clamped WIS estimator can also give unreliable results. Further steps need to be taken, both in the design of policy estimators and IS return estimators in order to generate a high confidence policy for clinical use. We have outlined the specific areas of improvement. Implementing these recommendations remains as our future work.

Acknowledgements

Research reported in this publication was in part supported by National Heart, Lung, and Blood Institute (NHLBI) of the National Institutes of Health under award number R01HL125089.

References

Aitchison, John and Aitken, Colin GG. Multivariate binary discrimination by the kernel method. *Biometrika*, 63(3): 413–420, 1976.

Bowman, Adrian W. A note on consistency of the kernel method for the analysis of categorical data. *Biometrika*, 67(3):682–684, 1980.

Degris, Thomas, White, Martha, and Sutton, Richard S. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

Ernst, Damien, Geurts, Pierre, and Wehenkel, Louis. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

Ernst, Damien, Stan, Guy-Bart, Goncalves, Jorge, and Wehenkel, Louis. Clinical data based optimal sti strategies for hiv: a reinforcement learning approach. In *Decision and Control, 2006 45th IEEE Conference on*, pp. 667–672. IEEE, 2006.

Forman, Daniel E, Butler, Javed, Wang, Yongfei, Abraham, William T, O'Connor, Christopher M, Gottlieb, Stephen S, Loh, Evan, Massie, Barry M, Rich, Michael W, Stevenson, Lynne Warner, et al. Incidence, predictors at admission, and impact of worsening renal function among patients hospitalized with heart failure. *Journal of the American College of Cardiology*, 43(1):61–67, 2004.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Glynn, Peter W. Importance sampling for markov chains: Asymptotics for the variance. *Stochastic Models*, 10(4): 701–717, 1994.

Gu, Shixiang, Lillicrap, Timothy, Ghahramani, Zoubin, Turner, Richard E, and Levine, Sergey. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

Guo, Zhaohan Daniel, Thomas, Philip S, and Brunskill, Emma. Using options for long-horizon off-policy evaluation. *arXiv preprint arXiv:1703.03453*, 2017.

Hallak, Assaf, Schnitzler, Francois, Mann, Timothy, and Mannor, Shie. Off-policy model-based learning under unknown factored dynamics. In *International Conference on Machine Learning*, pp. 711–719, 2015.

Hanna, Josiah P, Stone, Peter, and Niekum, Scott. Bootstrapping with models: Confidence intervals for off-policy evaluation. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 538–546. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

Hansen, Nikolaus, Müller, Sibylle D, and Koumoutsakos, Petros. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

Hester, Todd, Quinlan, Michael, and Stone, Peter. Generalized model learning for reinforcement learning on a humanoid robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 2369–2374. IEEE, 2010.

Jiang, Nan and Li, Lihong. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Johnson, Alistair EW, Pollard, Tom J, Shen, Lu, Li-wei, H Lehman, Feng, Mengling, Ghassemi, Mohammad, Moody, Benjamin, Szolovits, Peter, Celi, Leo Anthony, and Mark, Roger G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Kakade, Sham Machandranath et al. *On the sample complexity of reinforcement learning*. PhD thesis, 2003.

Li, Qi and Racine, Jeff. Nonparametric estimation of distributions with categorical and continuous data. *journal of multivariate analysis*, 86(2):266–292, 2003.

Munos, Rémi, Stepleton, Tom, Harutyunyan, Anna, and Bellemare, Marc. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1054–1062, 2016.

Nemati, Shamim, Ghassemi, Mohammad M, and Clifford, Gari D. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 2978–2981. IEEE, 2016.

Njeh, CF, Hans, D, Li, J, Fan, B, Fuerst, T, He, YQ, Tsuda-Futami, E, Lu, Y, Wu, CY, and Genant, HK. Comparison of six calcaneal quantitative ultrasound devices: precision and hip fracture discrimination. *Osteoporosis International*, 11(12):1051–1062, 2000.

Prasad, Niranjani, Cheng, Li-Fang, Chivers, Corey, Draugelis, Michael, and Engelhardt, Barbara E. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*, 2017.

Precup, Doina, Sutton, Richard S, and Singh, Satinder P. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 759–766. Morgan Kaufmann Publishers Inc., 2000.

Precup, Doina, Sutton, Richard S, and Dasgupta, Sanjoy. Off-policy temporal-difference learning with function approximation. 2001.

Raghu, Aniruddh, Komorowski, Matthieu, Celi, Leo Anthony, Szolovits, Peter, and Ghassemi, Marzyeh. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*, 2017.

Rahimi, Ali and Recht, Benjamin. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rochon, James, Protiva, Petr, Seeff, Leonard B, Fontana, Robert J, Liangpunsakul, Suthat, Watkins, Paul B, Davern, Timothy, and McHutchison, John G. Reliability of the roussel uclaf causality assessment method for assessing causality in drug-induced liver injury. *Hepatology*, 48(4):1175–1183, 2008.

Sutton, Richard S and Barto, Andrew G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Thomas, Philip S, Theocharous, Georgios, and Ghavamzadeh, Mohammad. High-confidence off-policy evaluation. 2015.