Interpretable Reinforcement Learning for Healthcare with Decision Sets

Dylan Randle
dylanrandle@g.harvard.edu

Nam Luu Nhat
namluunhat@g.harvard.edu

Nicholas Stern
nicholas_stern@g.harvard.edu

Abstract

Modern reinforcement learning agents are often trained with black-box models which makes it difficult or impossible to interpret the policies they learn. In high-stakes environments this can present a significant impediment to real-world deployment. This paper proposes decision sets to mimic black-box agents and provide interpretability. Our results demonstrate that decision sets can be a viable alternative to black-box models if we are willing to sacrifice some performance for significantly improved interpretability.

1 Introduction

In reinforcement learning (RL), agents are trained to maximize rewards obtained by taking actions at various states in an environment. For complex tasks, this process can be a “black box," in that it is unclear how the agent makes these decisions. For high-stakes environments such as healthcare, the lack of interpretability of models can be a significant barrier to their implementation. People are generally less likely to trust what they cannot understand. This paper aims to address the lack of transparency by leveraging decision sets to illuminate which aspects of the state space led the agent to choose an action. Specifically, our goal is to approximate the policy of a black-box model with a decision set and thereby explain each action the agent takes based on a rule associated with the current state of the environment.

We chose decision sets because, unlike most other methods, they do not rely on a hierarchical structure (i.e. like decision trees or decision lists) to make predictions. Hierarchical structures are harder to interpret because, in order to simulate a prediction, all conditions at higher levels in the hierarchy need to be remembered. This is a significant barrier to interpretability as humans are less capable of remembering a larger number of items at once. Decision sets, on the other hand, learn a set of independent rules which do not rely on each other, which increases their interpretability. For the purposes of comparison, we test other interpretable models (e.g. decision trees, generalized additive models) to showcase options a practitioner may have and highlight the trade-off we observe between performance and interpretability.

2 Related Work

The authors of [13] develop “Neurally Directed Program Search” to find programmatic policies and evaluate them on the TORCS car-racing environment. However, we do not find this method to yield highly interpretable policies as the output of their model is a complex program specified in a domain-specific language that was designed for the task. To a layperson, it appears inscrutable.

Preprint. Under review.

Recent work in [2] has furthered development of imitation-based learning approaches and used decision trees to approximate neural network policies to verify them (e.g. for safety). Their approach extends DAGGER [11] to include the original Q-values from the black-box model to improve learning performance of the imitation policy.

To the best of our knowledge, we are the first to apply decision sets to the problem of policy interpretation for RL.

3 Method

Formally, we consider a Markov Decision Process (MDP) defined by a set of states S, actions A, and rewards $R : S \times A \rightarrow \mathbb{R}$. Our agent seeks to maximize a discounted sum of rewards at each time step:

$$G = \sum_{t=0}^{\infty} \gamma^t r_t \tag{1}$$

where $\gamma \in [0, 1)$ and r_t is the reward received at time t.

The scope of our project encompasses multiple strategies to approximate an already learned policy with an interpretable, rule-based method. In this paper, we present results from: (1) supervised learning on data generated from an already learned policy, (2) a principled imitation learning algorithm called DAGGER [11]. We make use of an environment that simulates HIV drug treatment schedules [5], and highlight the algorithmic and computational challenges encountered along the way.

3.1 Decision Sets

Decision sets [7] are concise, non-overlapping if-then rules that operate on a group of observed attributes or features, x, and match them with a class label, c. An example decision set is shown in Figure 1.

If Respiratory-Illness=Yes **and** Smoker=Yes **and** Age$\geq$ 50 **then** Lung Cancer

If Risk-LungCancer=Yes **and** Blood-Pressure$\geq$ 0.3 **then** Lung Cancer

If Risk-Depression=Yes **and** Past-Depression=Yes **then** Depression

Figure 1: Example of a decision set with attributes outlined in blue, and labels in red.

The combination of attributes and operators within each statement is called a predicate, p, while a predicate and its corresponding class label together are called a rule, r. The conjunction of one or more predicates is called an itemset, s, and the conjunction of one ore more rules is called a decision set, $\mathcal{R}$.

Decision sets are fit to data by first generating candidate rules through a frequent itemset mining algorithm (e.g. Apriori [1]). These candidates are then either included or excluded according to the objective function, shown in Eq. 2, which is a linear combination of seven terms that emphasize the following properties:

$$F(\mathcal{R}) = \sum_{i=1}^{7} \lambda_i f_i(\mathcal{R}) \tag{2}$$

- Prefer decision sets with fewer rules and fewer predicates within each rule.
- Discourage rules from overlapping in feature space.
- Encourage at least one rule for each class.
- Favor high precision and recall.

The algorithm to select candidates that maximize the objective function is the Smooth Local Search (SLS) algorithm, and capitalizes on the fact that the objective function is non-negative, non-normal, non-monotone, and submodular. This yields a $\frac{2}{5}f^*$ (expected) lower bound on the solution quality [6], where f^* is the optimal objective value.

In words, the algorithm is composed of three iterative steps: (1) evaluate how the addition of each rule impacts the objective function, using a decision set sampled with bias from a set A of previously-added high-value rules. (2) Add candidates to A if they significantly improve the objective, and re-evaluate all the candidates. (3) Remove candidates from A if they significantly worsen the objective, and re-evaluate all the candidates. Once no candidates are added or removed, the process terminates and returns a decision set, again sampled with bias from A (but including, as always, the entire domain of rules). The algorithm is stochastic in nature, and in general the objective value lower bound is achieved in expectation.

3.2 Training an Interpretable Policy

Algorithm 1: Supervised Learning

```
Train expert policy: $\pi^*$
Generate dataset, $\mathcal{D}$, comprised of $\epsilon$-greedy $(s, a)$ tuples from $\pi^*$
Fit interpretable classifier on $\mathcal{D}$
Extract greedy policy $\hat{\pi}$ such that $\hat{\pi} : s \to \hat{a}$.
```

Our initial approach, outlined in Algorithm 1, was to first learn an “expert" policy using any preferred black-box method, then generate trajectories from the learned policy following an ϵ-greedy action selection strategy[1]. With this dataset of (state, action) tuples, we then framed the problem as a supervised learning task of predicting actions from states.

The states were one-hot encoded so that we could apply the Apriori algorithm to mine candidate itemsets. We then ran SLS [6] to optimize the objective. Note, we sorted the resulting decision set by the rules' accuracy on the training data; in the case of tie-breaking (e.g. when a data point satisfies multiple rules), the rule with higher accuracy was chosen. For predictions that did not satisfy any rules, the majority class was predicted.[2] We tuned the hyperparameters of the decision set using a random search.

Algorithm 2: DAGGER

```
Initialize $\mathcal{D} \leftarrow \emptyset$
Initialize $\hat{\pi}$ to any policy in $\Pi$
for $i = 1$ to $N$ do
    Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$
    Sample $T$-step trajectories using $\pi_i$
    Get dataset $\mathcal{D}_i = (s, \pi^*(s))$ of visited states by $\pi_i$ and actions given by expert
    Aggregate datasets $\mathcal{D} = \mathcal{D} \bigcup \mathcal{D}_i$
    Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$
end
return best $\hat{\pi}_i$ on validation.
```

One limitation of Algorithm 1 is that the typical assumption that the data are independent and identically distributed (i.i.d.) is violated, since the data collected by the agent in the future are dependent on the agent's current policy. This problem is well-studied in the literature of imitation learning, where methods have been developed to more closely mimic an expert policy. A natural starting place for imitation learning is DAGGER, shown as Algorithm 2 [11]. DAGGER gradually augments the training dataset with examples from the mimic policy itself. Instead of evaluating the

[1] We learn via the expert because decision sets are non-differentiable classification models and thus not easily amenable to direct methods like Q-learning or policy gradient

[2] This choice of tie-breaking and default class labels is motivated by [7]

mimic policy through accuracy on a test set, the mimic policy is evaluated in a simulated environment, and has been shown capable of achieving similar performance to the expert policy.

We proceeded to implement these two algorithms in a simple Grid World, as a proof of concept, as well as in an environment that simulates HIV treatments and has a richer, continuous state space and bridges us over to the medical domain.

4 Experimental Results

4.1 Grid World

As an initial proof of concept, we evaluated the supervised approach on a small Grid World problem (shown in Figure 2). We found the optimal policy for this environment using the Value Iteration algorithm [12]. From this optimal policy, we collected 500 state-action pairs sampled with a small amount of randomness $\epsilon = 0.1$ to allow us to collect a variety of transitions.

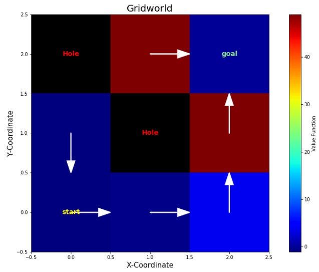
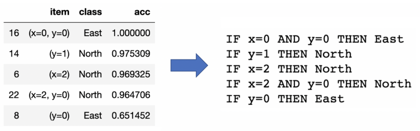


Figure 2: Example of our Grid World environment.

We trained the decision set model to predict actions given the state and achieved high accuracy on a held-out test set. Figure 3 shows the learned policy. We have overlaid the policy in Figure 2.



	item	class	acc
16	(x=0, y=0)	East	1.000000
14	(y=1)	North	0.975309
6	(x=2)	North	0.969325
22	(x=2, y=0)	North	0.964706
8	(y=0)	East	0.651452

```
IF x=0 AND y=0 THEN East
IF y=1 THEN North
IF x=2 THEN North
IF x=2 AND y=0 THEN North
IF y=0 THEN East
```

Figure 3: The decision set is able to recover the optimal policy for Grid World.

We found that the decision set model is able to capture the exact optimal policy and therefore validate our assumption that it is possible to recover an optimal policy from the collected transitions of an expert.

4.2 HIV Simulator

The HIV simulator environment explores the optimal drug schedule to administer to HIV patients. The state space is 6-dimensional (continuous), and represents various volume densities of infected and healthy blood cells. There are 4 unique actions, which represent the possible combinations of activating (or not activating) Reverse Transcriptase Inhibitor (RTI) and Protease Inhibitor (PI). The reward is a linear function that penalizes treatment (due to the side effects), and encourages higher populations of cytotoxic T-lymphocytes, cancer killing cells.

4.2.1 Supervised Approach

We first implemented Algorithm 1 with an FQI-trained ExtraTreesRegressor[3] as the expert policy [4]. Evaluating the expert agent by simulating additional trajectories resulted in cumulative reward of $\sim 2.8 \times 10^{10}$. We trained a decision set on a dataset, $\mathcal{D}$, of 30,000 state-action pairs sourced from the expert agent traversing 150 trajectories in the environment, each with 200 steps. As the states

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html

were continuous, we binned them in a histogram fashion (i.e. equally spaced bins on the range of the data) and determined via experiment that 30 bins per state was optimal. Interestingly, we saw that a more accurate classifier did not necessarily imply higher rewards. The two former points are highlighted in Figure 7 in the Appendix.

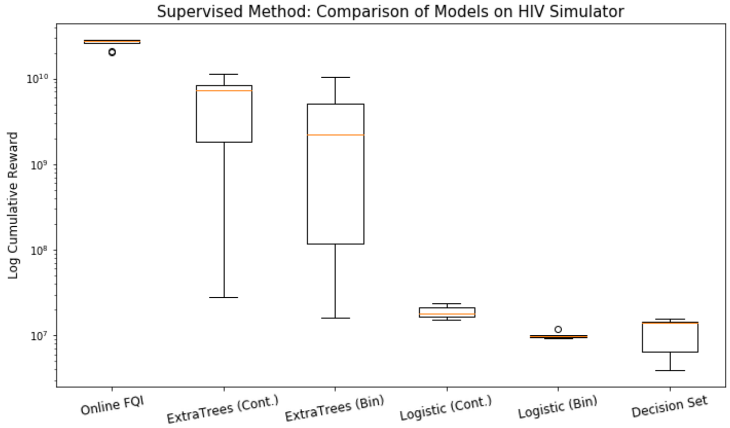


Figure 4: Log cumulative rewards calculated over 10 replications with the HIV simulator. We see that none of the models are able to capture the same reward as the original, online FQI policy. Furthermore we see that binning states (required by decision set models) appears to have a slight negative effect on performance as measured by mean and standard deviation of cumulative rewards.

As we can see in Figure 4, none of the approximated policies are able to recover the optimal policy. Even an ExtraTrees model, which is the exact same model used for the original optimal agent, is unable to recover it and suffers from high variance. However, this is not entirely surprising as the typical assumption in supervised learning that the data are independent and identically distributed is violated in this scenario. This is because the data collected at future times depends on the value of the current policy. Both a linear principle component analysis (PCA) and a non-linear t-Distributed Stochastic Neighbor Embedding (TSNE) suggest that it is difficult to separate the actions in state space directly from the collected dataset (see Figure 8 and 9 in the Appendix). The two former points indicate that approximating a policy directly from supervised data may be very challenging in general, echoing previous work in imitation learning and specifically the findings of [2].

4.2.2 Imitation Approach

Motivated by these results, we implemented Algorithm 2 using the same expert policy as above. To validate efficacy of this algorithm, we first trained an ExtraTreesClassifier as the mimic policy and obtained significantly better results than with the supervised approach. In Figure 5, we see that the ExtraTrees policy almost exactly captures the optimal agent with much lower variance than in the result from Algorithm 1. We were also able to train a single decision tree of depth 10 and roughly recover the optimal agent. However, we were unable to reach the same performance with the decision set. It suffers from high variance, but is about an order of magnitude better than logistic regression on average, and two orders of magnitude better than a random policy.

We also trained another newly popular model in the interpretable machine learning space, GA2M [3]. Recently open-sourced by Microsoft [3] and termed the Explainable Boosting Machine (EBM), it is a form of generalized additive model that fits decision trees to each feature independently but enables interpretability by its additive nature. EBM performed very well, recovering the optimal policy, while being modestly interpretable (see Figure 14 in the Appendix).

In Figure 6 we include the learned decision set. We can see that while it is not entirely interpretable on its own (30 rules), each rule is highly interpretable as they include a maximum of 2 predicates. Even though it does not perform as well as the decision tree or EBM models, we believe that it still may be worth the loss in performance due to the large gain in interpretability. In the Appendix, we plot the decision tree (Figure 13) and EBM (Figure 14) models for comparison of interpretability. We also plot the decision tree and decision set performance against a notion of

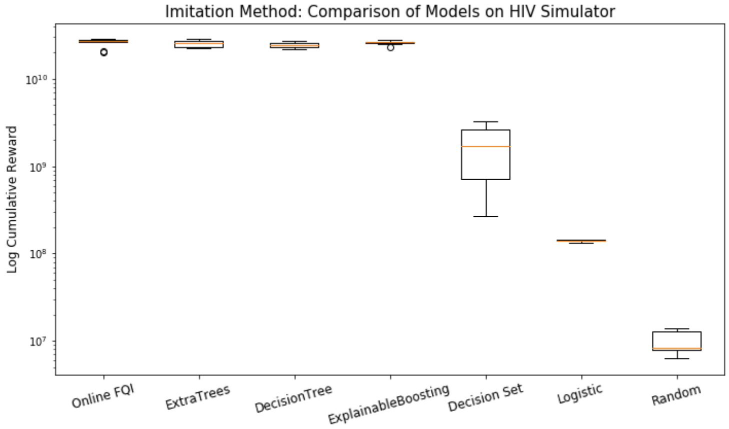


Figure 5: Log cumulative rewards of models trained with DAGGER. Error bars are calculated using 10 replications of the randomized HIV simulator. With DAGGER, we see that multiple models can recover the optimal policy.

interpretability (using the proxies of depth and number of rules, respectively) in Figures 12 and 11 in the Appendix.

```
IF 2.4 <= non-infected macrophages <= 2.6 THEN RTI
IF 2.2 <= non-infected macrophages <= 2.4 THEN RTI
IF 5.6 <= non-infected CD4+ T-lymphocytes <= 5.7 THEN RTI
IF 3.6 <= non-infected macrophages <= 3.8 THEN No Treatment
IF 3.4 <= number of free HI viruses <= 3.6 AND 1.6 <= infected macrophages <= 1.8 THEN RTI and PI
IF 1.3 <= infected macrophages <= 1.4 THEN RTI
IF 2.9 <= non-infected macrophages <= 3.1 THEN RTI and PI
IF 3.4 <= number of free HI viruses <= 3.6 AND 2.6 <= non-infected macrophages <= 2.9 THEN RTI and PI
IF 2.3 <= infected CD4+ T-lymphocytes <= 2.5 THEN RTI
IF 2.7 <= number of cytotoxic T-lymphocytes <= 2.8 THEN No Treatment
IF 2.7 <= infected CD4+ T-lymphocytes <= 2.8 THEN PI
IF 5.6 <= non-infected CD4+ T-lymphocytes <= 5.6 THEN RTI
IF 1.6 <= infected macrophages <= 1.8 THEN RTI and PI
IF 5.7 <= non-infected CD4+ T-lymphocytes <= 5.8 THEN RTI
IF 5.5 <= non-infected CD4+ T-lymphocytes <= 5.5 THEN RTI and PI
IF 2.1 <= number of cytotoxic T-lymphocytes <= 2.3 THEN RTI
IF 2.6 <= infected CD4+ T-lymphocytes <= 2.7 THEN RTI
IF 5.5 <= non-infected CD4+ T-lymphocytes <= 5.6 THEN PI
IF 1.7 <= number of cytotoxic T-lymphocytes <= 1.9 THEN RTI
IF 2.6 <= non-infected macrophages <= 2.9 AND 2.2 <= infected CD4+ T-lymphocytes <= 2.3 THEN RTI
IF 5.4 <= non-infected CD4+ T-lymphocytes <= 5.5 THEN No Treatment
IF 3.8 <= number of free HI viruses <= 4.1 THEN No Treatment
IF 5.8 <= non-infected CD4+ T-lymphocytes <= 5.8 THEN RTI
IF 3.6 <= number of free HI viruses <= 3.8 AND 2.9 <= non-infected macrophages <= 3.1 THEN RTI
IF 2.3 <= number of cytotoxic T-lymphocytes <= 2.4 THEN RTI
IF 3.4 <= number of free HI viruses <= 3.6 AND 2.9 <= non-infected macrophages <= 3.1 THEN RTI
IF 1.9 <= infected CD4+ T-lymphocytes <= 2.1 THEN RTI
IF 2.5 <= infected CD4+ T-lymphocytes <= 2.6 THEN No Treatment
IF 3.1 <= non-infected macrophages <= 3.4 THEN No Treatment
IF 1.6 <= number of cytotoxic T-lymphocytes <= 1.7 THEN RTI
```

Figure 6: Learned interpretable decision set on the HIV simulator with DAGGER.

5 Conclusion

We have shown that it is possible to learn interpretable policies by approximating those learned by black-box models. We believe that this work is a step in the direction of greater interpretability for RL and that it could be of significant use in deploying real-world RL-based agents that require human interpretability and user trust. We have presented various methods for achieving interpretability that come at different costs of performance (decision trees, EBMs, decision sets), and with different notions of interpretability. We hope that further research in the areas of rule-based and interpretable models will provide fruitful advancements in our ability to distill complex policies into trustable and interpretable ones.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[2] O. Bastani, Y. Pu, and A. Solar-Lezama. Verifiable reinforcement learning via policy extraction. In *Advances in Neural Information Processing Systems*, pages 2494–2504, 2018.

[3] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

[4] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6(Apr):503–556, 2005.

[5] D. Ernst, G.-B. Stan, J. Goncalves, and L. Wehenkel. Clinical data based optimal sti strategies for hiv: A reinforcement learning approach. pages 667 – 672, 01 2007.

[6] U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, July 2011.

[7] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1675–1684, New York, NY, USA, 2016. ACM.

[8] H. Nori, S. Jenkins, P. Koch, and R. Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

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

[10] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Vaughan, and H. M. Wallach. Manipulating and measuring model interpretability. *arXiv*, abs/1802.07810, 2018.

[11] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[12] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[13] A. Verma, V. Murali, R. Singh, P. Kohli, and S. Chaudhuri. Programmatically interpretable reinforcement learning. *CoRR*, abs/1804.02477, 2018.

[14] H. Yang, C. Rudin, and M. Seltzer. Scalable bayesian rule lists. *CoRR*, abs/1602.08610, 2016.

A Appendix

Please find additional figures referenced in the main text provided below.

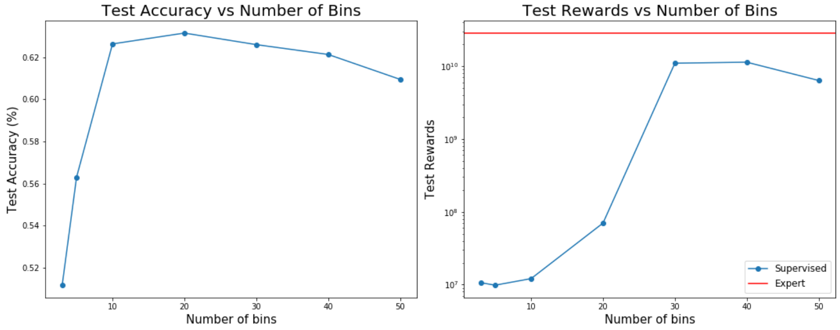


Figure 7: The relationship of the number of bins to the accuracy of an ExtraTreesClassifier, and its performance in the environment. Interestingly, high accuracy on the test set does not necessarily imply high performance in the environment. We chose 30 bins since this achieves high reward but at lower computational complexity and with increased interpretability compared to 40 or 50.

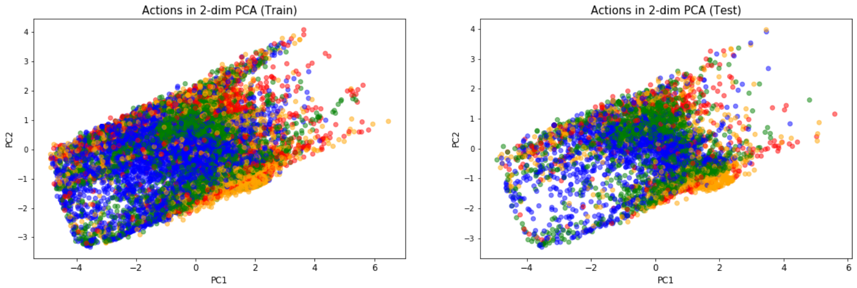


Figure 8: A principle components analysis (PCA) over the states from the sampled trajectories in the HIV simulator environment. The actions are color coded and the points reflect the top two principle components of the state matrix. We see that, at least in the linear projection of PCA, the actions are not easily separable.

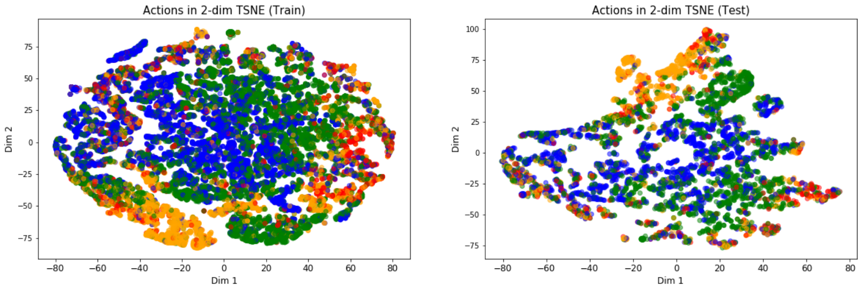


Figure 9: A t-Distributed Stochastic Neighbor Embedding (TSNE) over the states from the sampled trajectories in the HIV simulator environment. The actions are color coded and the points reflect the first and second components. We see that even in the non-linear projection of TSNE the actions are not easily separable.

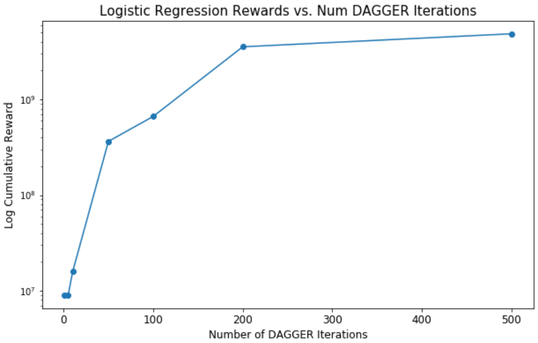


Figure 10: Analysis of the effect of number of iterations (i.e. episodes) of DAGGER on cumulative rewards of Logistic Regression on the HIV simulator environment. We see that 500 iterations performs the best, but used $N = 100$ because of the very long time taken to train decision sets even for this value (~ 15 hours).

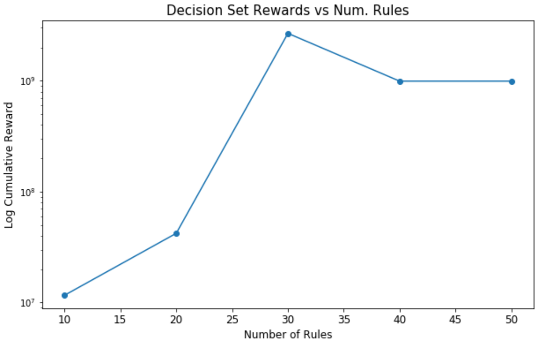


Figure 11: Analysis of effect of number of rules in a decision set on cumulative rewards on the HIV simulator environment. We changed the number of rules by keeping only the most accurate ones as measured by accuracy on the collected DAGGER dataset (i.e. dropping rules at the bottom of the sorted decision set, which is motivated by the tie-breaking method used in [7]). We see a decrease in performance beyond 30 rules because the rules in the decision set are randomly sampled (according to the SLS algorithm), and the ones we are adding back are the worst-performing in terms of accuracy and are likely to be sub-optimal.

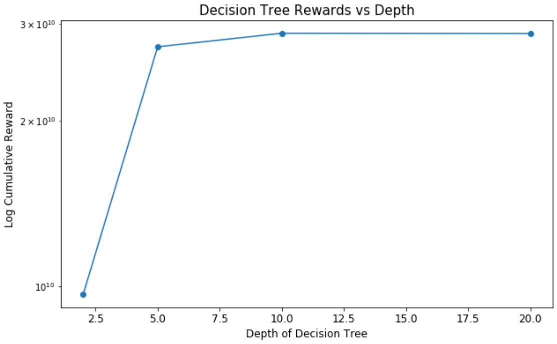


Figure 12: Analysis of the effect of decision tree depth on cumulative rewards on the HIV simulator environment. The results were obtained using $N = 100$ iterations of the DAGGER algorithm. We see that by $\sim$ depth $= 10$ the agent reaches the optimal reward.

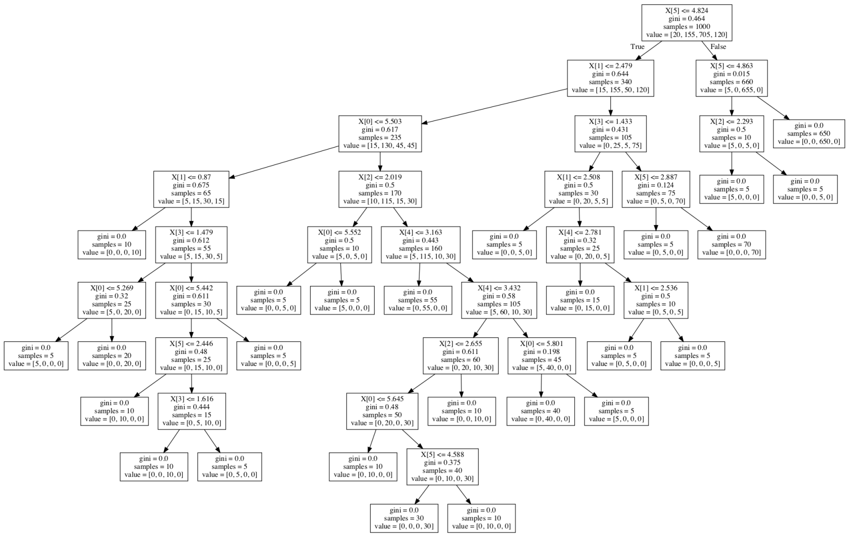


Figure 13: Interpretation of the depth 10 decision tree. While it performs very well, it is difficult to interpret (especially simulate) as the hierarchy requires that each previous split be kept in mind when understanding a prediction.

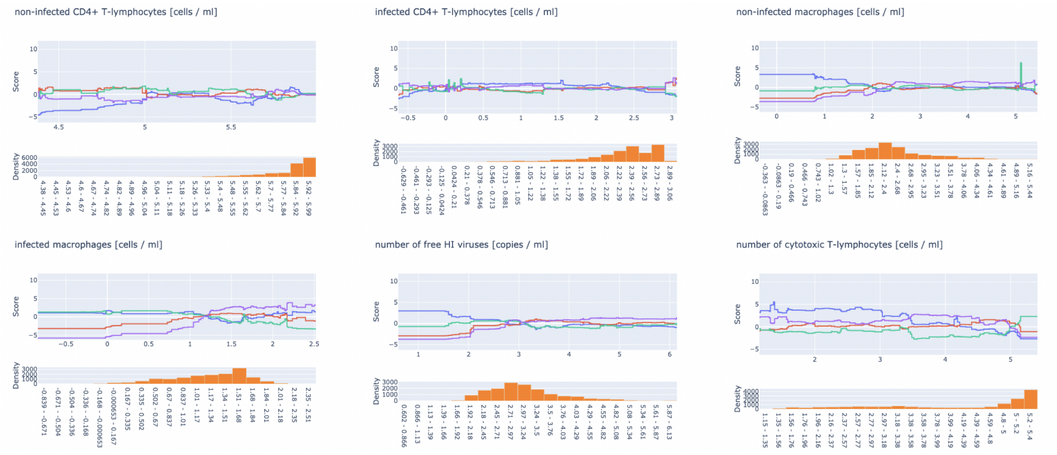


Figure 14: Interpretation for Explainable Boosting Machine model (i.e. GA2M/EBM). For each input X_i, we can plot the effect that its value has on the final model output due to the fact that the model is additive. Here we see the logit score of each class (treatment) for each of the 6 state measurements. Treatment 0 (no drugs) is blue, treament 1 (RTI) is red, treatment 2 (PI) is green and treatment 3 (RTI + PI) is purple.