Cost Of Interpretability-Regionwise Constant Maps

Deepak Badarinath
deepak.badarinath@stats.ox.ac.uk
Department of Statistics
University of Oxford

Agni Orfanoudaki
Agni.Orfanoudaki@sbs.ox.ac.uk
Saïd Business School
University Of Oxford

Soroush Saghafian
soroush_saghafian@hks.harvard.edu
Harvard Kennedy School
Harvard University

Abstract

For machine learning algorithms to be applicable in human-centric fields such as healthcare, law-making, or industrial design, it is essential to develop interpretable techniques. It is of utmost importance that the physician/government lawmaker/factory worker understand why an algorithm gave the answer that it did. In this paper, we define interpretable policies as being regionwise constant maps. Previous work has computed the optimal policy which takes actions on a partitioned state space. Our work is the first that aims to compute the regions as well as derive the optimal action to take on the partition. We compute upper bounds on the cost of interpretability as the error in summarizing the final policy by a region-wise constant map. We see that this is given by the function summarization error for the final policy. We run experiments to check how our approach performs with the dimension and size of the state space. We compute the optimal interpretable policy for different final policies.

1 Introduction

It is no surprise that automation of decisions using artificial intelligence has led to better outcomes for humanity. To allow for the creation of decision-making agents in critical applications such as healthcare, law-making, or industrial manufacturing the output of an algorithm must be interpretable. Towards this goal, dynamic decision-making algorithms whose output is interpretable are the need of the hour.

For humans to understand the output of a dynamic decision-making algorithm, the output of the policy learning method must be *interpretable*. Traditional ways of defining interpretability involve tree-based final policies Zhu et al. (2015), or policies based on lists Zhang et al. (2018). An information-theoretic way to think of interpretability is to have the final policy be a region-wise constant map. Here the number of regions present in the policy represents the information stored in the policy. The accuracy of the final policy increases as the number of regions in the final policy increases.

Computing piecewise constant approximations of functions has been studied in the signal processing literature and algorithms exist to find the piecewise constant approximator for a given function in $1D$ space such as Bergerhoff et al. (2019), Dar & Bruckstein (2019). In our paper, we show that the cost of interpretability (the difference in the values of the optimal policy and the optimal interpretable policy) is bounded by a region-wise constant approximation to the optimal final policy. Further based on the geometry of the splitting procedure, we can derive the approximation to the optimal interpretable policy.

Petrik & Luss (2016) compute the optimal interpretable policy for a partitioned state space. They achieve this by formulating the problem of calculating the optimal value function of a policy as a Mixed Integer Linear Program for which we get an exact solution when the policies are restricted to take values on the quotient of the state space. In our excerpt, we show that we can upper bound the cost of interpretability using regionwise constant maps and we get that this is given by approximating the final policy using such a map. We now compute the optimal partition using a tree-based splitting mechanism where we decide the splits based on which split minimizes the total distance from the median. We can think of it as an L^1 version of the CART algorithm from Steinberg (2009).

Our contributions can be summarized as follows:

- We define a notion of interpretability that is given by regionwise constant maps. Further, we define the cost of interpretability problem as a function of the depth of the optimal interpretable policy.
- We show that the cost of interpretability can be bounded by the regionwise constant approximation to the final obtained policy.
- We run experiments to compute the cost of interpretability using a tree-based splitting mechanism. We check to see how our method performs with noise, dimension, and size.
- We define the cost of interpretability as the difference obtained between the optimal value function and the optimal interpretable policy. We hope to do future work on this front.

Our paper is organised in the following manner. In Sec. 2 we discuss notions of interpretability that already occurs in the literature. In Sec. 3 we introduce the setting for the problem and come up with the formalism for the blocked value iteration method. In Sec. 4 we simplify the derived upper bounds on the cost of interpretability and show that it is given as a regionwise constant approximation of the final policy. In Sec. 5 we experimentally look at the behaviour of our method in custom environments, and see the dependence with dimension, size, and other environmental parameters. In Sec. 6 we discuss the pros and cons of the method and mention future work to extend this line of research. In Sec. 7, we conclude the paper and discuss the main insights that can be drawn.

2 Related Work

In this section, we introduce different notions of interpretability for Markov Decision Processes and the reinforcement learning problem.

In Grand-Clément (2021) value iteration using backward induction is performed. In this paper, they compute the optimal tree policy by maximizing the total reward received at every timestep for the reinforcement learning problem in a greedy fashion. Our approach works by performing value iteration on the partitioned state space. We compute the cost of interpretability by noticing the difference in the value functions of each element in the partition with the leader.

In Petrik & Luss (2016) they assume that a partition of the state space is given through an interpretability map from states to a new space which they define to be observations. They obtain policies on the quotient of the state space by defining interpretable policies to be those deterministic policies that take the same value on the pre-image of the interpretability map. One can hope to define any partition of the state space by using such a map, and then use their method to compute the exact optimal policy. The problem we wish to solve involves computation of the interpretability map. We can extend the Mixed Integer Linear Program obtained in this to the case where we actually have to compute the partition of the $1 - D$ space.

The paper Givan et al. (2003) gives another way to obtain this partitioning of a Markov Decision Process. In this paper, they define a factored MDP- a Markov Decision Process obtained over a smaller state space after combining equivalent states. For MDPs with a massive state space,

this would result in exponentially fewer states and we would be able to use algorithms running in polynomial time over the state space on this reduced Markov Decision Process. They group together states to form quotient MDPs. They also come up with approaches to perform model minimization for MDPs.

In Hein et al. (2018) we perform reinforcement learning on existing trajectory data to obtain policies which are represented by basic algebraic equations that are restricted to an adequate complexity. Genetic Programming for Reinforcement Learning approaches work as follows; we find the optimal policy amongst the set of all possible equations which can be built upto a certain complexity. This can be thought of as a version of solving the cost of interpretability problem from the introduction since we are choosing those policies to solve the problem that are below a certain degree of complexity. Here policies that are represented as simple algebraic equations are preferred.

In Petrik & Luss (2016) they assume a partition of the space is known through an interpretability map. This maps states to a new space which they define to be the observation space. They obtain policies on the quotient of the state space by defining interpretable policies to be deterministic maps that take the same value on the pre-image of the interpretability map. One can hope to define any partition of the state space by using such a map, and then use their method to compute the exact optimal policy. The problem we wish to solve involves the computation of the interpretability map.

The article Glanois et al. (2024) is a survey on interpretable reinforcement learning that speaks about interpretability in different contexts. They start by introducing the notion of interpretable inputs, where the agent performs symbol extraction to convert a high-dimensional input from the world into a lower-dimensional perception model. Next, they speak about interpretable transition models where the probability transitions are made to be interpretable by learning decision trees or other graphical models to represent the probability transitions. Another area where interpretability can be introduced is when learning the reward transitions. The ability to understand how the reward dynamics work in a given environment proves to be useful in learning which actions an agent would take and at what stage. The last area they speak about, which is related to our line of work is interpretability in the policies. In this section, they list literature where the agent learns tree-based interpretable policies either directly while solving for the optimal value function or as a post-doc method after obtaining an approximation of the optimal interpretable policy. In our paper, we showed that a way to upper bound the cost of interpretability is by performing an interpretable approximation of the obtained final policy. We provide theoretical justification to the claim that a region-wise constant approximation of the policy would help us give upper bounds on the cost of interpretability.

3 Background

We are given a Markov Decision Process (S, A, P, r) with a finite state space S, action space (can be finite/infinite) A with a metric $d(a, a')$ which measures distances between actions, rewards $r(s, a)$, and uniform transition kernels $P(s'|s, a) \forall s, s' \in S, a \in A$. We also assume that the time horizon $T = \infty$, and since r and P are independent of time, we aim to compute policies that are uniform in time. We assume we can do K clusters of the state space, and we wish to find a partition of the state space $S = \sqcup_{i=1}^{K} S_i$ and a policy π, such that $\pi|_{S_i} = a_i \in A$.

An 'interpretable' policy would be parametrized as $[(S_i, a_i)|i \in [K]]$. We can also impose restrictions on how we partition the space, for example in $\mathbb{R}^d$, we might have each S_i be a box. The following method gives a bound on the cost of interpretability. We do this by performing the blocked value iteration, where we assume a partition of the state space with a chosen leader set. We measure the discrepancy in the value function of the leader set when compared to the other elements in the partition and perform the value iteration procedure accordingly on the leaders, whilst copying the actions to the members of the partition.

4 Method

In this setting, we want to define an interpretable policy as a piece-wise constant policy that takes values on a partition of the state space S with the number of clusters k. For this, we need to find a partition of the state space $S = \sqcup_{k=1}^{K} S_i$ and a policy π, such that $\pi|_{S_i} = a_i \in A$ is an action. An interpretable policy would be parameterized as $[(S_i, a_i)|i \in [K]]$ and $\sqcup_{i=1}^{K} S_i = S$ and $a_i \in A$. We can also impose restrictions on how we do this partitioning, for example in $\mathbb{R}^d$, we might have that each S_i be a box. The following method gives a bound on the cost of interpretability.

4.1 Lead-and-value-iterate

For every S_i, we wish to pick a leader s_i such that the final policy π obeys whatever the optimal policy does on s_i^*, i.e $\pi(s) = \pi^*(s_i) \forall s \in S_i$. If we can compute the optimal policy, then we can obtain π by

$$\pi(s) := \pi^*(s_i) \forall s \in S_i \tag{1}$$

After doing this, we can obtain the cost of interpretability for a given partition $\mathcal{P}$ and a set of leader sets S^* in the partition. To recall notation, let $\mathcal{P} := \sqcup_{k=1}^{K} S_k$, and $S^* := \{s_i^*\}_{i=1}^{K}$. Assume we have a measure ds on our space, and we measure distances using this (we can now get a general version of an optimization problem for both the multidimensional and single-dimensional cases) We then define the cost of interpretability for the partition and the leader set as

$$C(\mathcal{P}, S^*) := \frac{1}{|S|} \int_S |V_{\pi^*}(s) - V_\pi(s)| ds \tag{2}$$

This is equal to

$$= \frac{1}{|S|} \sum_{i=1}^{K} \int_{S_k} (V_{\pi^*}(s) - V_\pi(s)) ds$$

4.2 Finding the cost of interpretability

Here we expand the expression to find a bound on the cost of interpretability, along the way we see that we need assumptions on γ, Lipschitz continuity of P, and r in the action space to get the bound. We then develop discrete algorithms that lessen the value of this bound for different cases of S and metrics d, namely S is unstructured, a subset of the real line, a subset of n dimensional space, and d is L^1/L^2.

By the Bellman equation on π^* and π, we know that

$$V_{\pi^*}(s) = r(s, \pi^*(s)) + \gamma P_{\pi^*(s)} V_{\pi^*}(s) \tag{3}$$

When $s \in S_k$, we have that

$$V_\pi(s) = r(s, \pi^*(s_k)) + \gamma P_{\pi^*(s_k)} V_\pi(s) \tag{4}$$

Subtracting 3 from 4, we get

$$(r(s, \pi^*(s)) - r(s, \pi^*(s_k))) + \gamma (P_{\pi^*(s)} V_{\pi^*}(s) - P_{\pi^*(s_k)} V_\pi(s)) \tag{5}$$

To simplify we make assumptions on the rewards and transitions of the model. Assume that there exists a metric d, and constant $C_r > 0$, such that for all actions $a, b \in A$

$$|r(s, a) - r(s, b)| \leq C_r d(a, b) \tag{6}$$

The first part in [5] then simplifies as

$$|r(s, \pi^*(s)) - r(s, \pi^*(s_k))| \leq C_r d(\pi^*(s), \pi^*(s_k)) \tag{7}$$

The second part is then given by $\gamma|P_{\pi^*(s)}V(s) - P_{\pi^*(s_k)}V(s)|$. We expand this out by

$$\gamma|\int_S P(s'|s,\pi^*(s))V_{\pi^*}(s')ds' - \int_S P(s'|s,\pi^*(s_k))V_{\pi^*}(s')ds' + \tag{8}$$

$$\int_S P(s'|s,\pi^*(s_k))V_{\pi^*}(s')ds' - \int_S P(s'|s,\pi^*(s_k))V_{\pi}(s')ds'| \tag{9}$$

The final two terms of 8 can be brought together to yield

$$\int_S P(s'|s,\pi^*(s)) - P(s'|s,\pi^*(s_k))V_{\pi^*}(s') \tag{10}$$

Here we now bring in smoothness assumptions on the transitions,

$$|P(s'|s,a) - P(s'|s,b)| \leq C_p d(\pi^*(s),\pi^*(s_k)) \tag{11}$$

And we assume that value functions are uniformly bounded, which does happen when rewards are bounded and $\gamma < 1$

$$V_\pi(s) \leq V_b \tag{12}$$

So the inequality in 10 is now

$$\gamma\int_S P(s'|s,\pi^*(s)) - P(s'|s,\pi^*(s_k))V_{\pi^*}(s') \leq \gamma V_b C_p d(\pi^*(s),\pi^*(s_k)) \tag{13}$$

Bringing together 7 and 13 and taking the summation over the partitions and the integral inside, we get

$$(C_r + \gamma V_b C_p)\frac{\sum_{k=1}^K \int_{S_k} d(\pi^*(s),\pi^*(s_k))ds}{|S|} \tag{14}$$

The first two terms of 8 is given by

$$\int_{S_k}\frac{\gamma}{|S|}\sum_{k=1}^K\int_S P(s'|s,\pi^*(s_k))(V_{\pi^*}(s') - V_\pi(s')ds') \tag{15}$$

We can now define an averaged probability density by bringing the summation inside, this yields the following

$$\bar{P}_{\mathcal{P},S^*}(s') := \frac{1}{|S|}\sum_{k=1}^K\int_{S_k} P(s'|s,\pi^*(s_k))ds \tag{16}$$

We can see that $\int_S \bar{P}_{\mathcal{P},S^*}(s')ds' = 1$ and hence it is a probability measure on the state space S. Coming back to 16, we see

$$\int_{S_k}\frac{\gamma}{|S|}\sum_{k=1}^K\int_S P(s'|s,\pi^*(s_k))(V_{\pi^*}(s') - V_\pi(s')ds') \leq \gamma\int_S \bar{P}_{\mathcal{P},S^*}(s')(V_{\pi^*}(s') - V_\pi(s'))ds' \tag{17}$$

Then we have

$$\gamma\int_S \bar{P}_{\mathcal{P},S^*}(s')(V_{\pi^*}(s') - V_\pi(s'))ds' \leq \gamma|S|C(\mathcal{P},S^*) \tag{18}$$

This gives us another assumption, namely

$$\gamma < \frac{1}{|S|} \tag{19}$$

After moving [19] to the left, we then get

$$C(\mathcal{P}, S^*) \leq \frac{1}{1-\gamma|S|}(C_r + \gamma V_b C_p)\frac{\sum_{k=1}^{K}\int_{S_k} d(\pi^*(s), \pi^*(s_k))ds}{|S|} \tag{20}$$

We only have to bound $\frac{\sum_{k=1}^{K}\int_{S_k} d(\pi^*(s), \pi^*(s_k^*))ds}{|S|}$, and we need to do the following optimization

$$argmin_{\mathcal{P}, S^*}\frac{\sum_{k=1}^{K}\int_{S_k} d(\pi^*(s), \pi^*(s_k))ds}{|S|} \tag{21}$$

Theorem 1 *We can obtain an upper bound on the cost of interpretability $C(\mathcal{P}, S^*)$ for regionwise constant policies by solving the partition problem given by:*

$$C(\mathcal{P}, S^*) \leq \frac{1}{1-\gamma|S|}(C_r + \gamma V_b C_p)\frac{\sum_{k=1}^{K}\int_{S_k} d(\pi^*(s), \pi^*(s_k)ds}{|S|} \tag{22}$$

For doing so, we look at particular cases of d and different connectivity structures in how we can group the S_is. In the first case, assume $d(a, b) = ||a - b||_2$, and assume a free unstructured finite state space, i.e we can group any set of elements. Then the discrete optimization problem returns

$$\sum_{k=1}^{K}\sum_{s\in S_k} ||\pi^*(s) - \pi^*(s_k^*)|| \tag{23}$$

And we need to find $S = \sqcup_{k=1}^{K} S_k$ and $S^* \subset S$ to minimize [23].

Finding the result to the bound [23] can be thought of as computing the regionwise constant approximation to the final policy. The goal here would be to perform the partitioning so that we can compute the regions S_i and the leader elements s_i^*. Future work can be done on how to bound 22 where we simplify the bounds obtained for different state spaces and conditions for performing the partitioning.

5 Computational Experiments

The goal of our computational experiments is to visually examine the optimal interpretable policy obtained and analyze it's behaviour with different environmental constraints such as the dimension and the state of the space. In this analysis we wish to demonstrate the performance of blocked value iteration with the dimension and size of the state space.

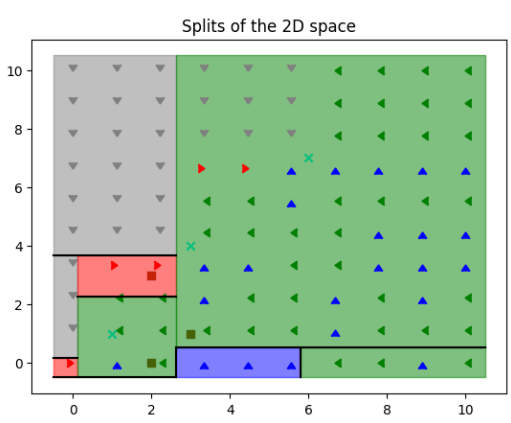
Our experimental setups work as follows: we define environments or use existing reinforcement learning environments for which we have access to a (noisy) version of the final policy π^*. To the final optimization problem obtained in Equation. 23, we perform tree-based splits to obtain a regionwise constant map. We look at the performance of these methods in three different environments.

5.1 Experimental Setup

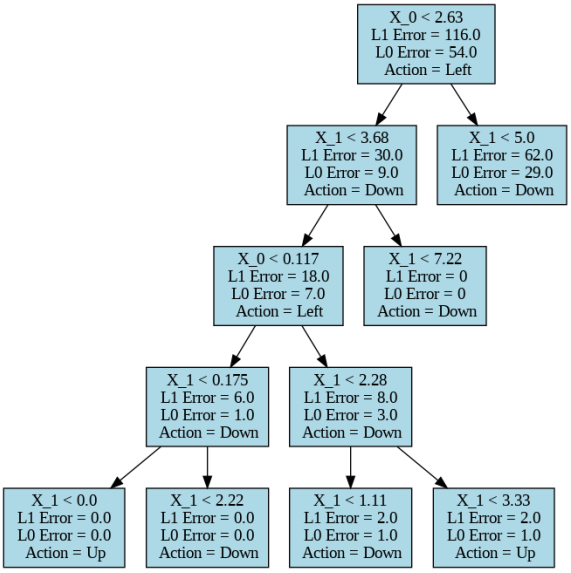
In this subsection we list the different environmental setups in which we wish to deploy our algorithms. Initially we start with the two dimensional grid environment for which we assume that to know the true oracle policy which is affected with noise. Here we compute the optimal splits and the actions to take by using a grid-based partitioning.

5.1.1 Two Grid Environment

In the two dimensional grid setup (Figure 1) we have a state space S to be given by a 10×10-$2D$ grid. The state space also has obstacles ($\mathcal{O}$) and goals ($\mathcal{G}$) that are denoted by a square and



(a) Splits obtained for the 2 dimensional space; arrows represent the actions chosen by the oracle, the colored regions represent the region-wise constant approximations. The goals 'x' are placed at $(1,1),(3,4)$, and $(6,7)$. The obstacles denoted by a square are placed at $(2,0),(3,1)$, and $(2,3)$



(b) Plot of the decision tree

Figure 1: Decision tree and it's graphical illustration for the two dimensional grid environment

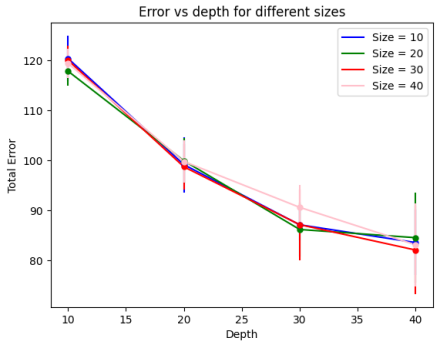
an x respectively. The actions are given by up, down, left, and right. The transition matrix is deterministic and we perform transitions as determined by the actions. A transition cannot occur if an obstacle is on the path towards making the transition. The objective of the agent is to reach one of the multiple goals placed at the different locations. When we reach the optimal goal we obtain a reward $r(s,a)=1$ if $s \in \mathcal{G}$. As in the setup, we assume an infinite horizon Markov Decision Process. To check the robustness of our method, in the final oracle policy, we also add random noise of 0.2. The agent decides to move with the prescribed action with a probability of 0.8, else the agent decides to take a random action with a probability of 0.2. For different values of k, we wish to find the optimal splits of the space $S = \sqcup_{i=1}^{K} S_i$ and the optimal actions to take $a_i \in \mathcal{A}$ for $i \in [K]$ such that we obtain the maximal reward.

5.2 Results

In Figure 1 we are given a grid of values with values ranging on the x-axis and y-axis from 0 to 10. We plot the individual actions taken by the oracle at all the states in the box from 0 to 10 in the x and y axes. We can attribute the presence of noisy actions in the plot to the case when we perform an action that is not the optimal one, which happens 20% of the time. We place 3 goals at points $[(1,1),(3,4),(6,7)]$ and we place 3 obstacles are placed at $[(2,0),(3,1),(2,3)]$. We can see in Figure 1 that the splits are regionwise constant approximations to the final policy and this is captured even in the presence of noise in the sampling of the final action.

For different lengths over the state space two dimensional grid state space we compute the total error vs depth in Figure 2a. We can see how the errors drop with increase in depth and decrease in size. In this plot and Figure 2 to counter for the randomness within the experimental setup, we sample the same point in the graph 10 times, and we plot the mean of the sampled points. The variance is plotted as an error bar where we look at the distance of the sampled points from the mean. We observe the large variance of the observed plot with size, which implies that with the different sizes, the error graphs do not decrease or increase.

In the next set of experiments, we assume a functional form for the final policy with a varying dimension. The goal of this technique is to measure how the methods perform with environmental parameters such as the dimension and size of the plots. To this function, we perform the tree based



(a) Error vs depth for different lengths of the 2D grid state space

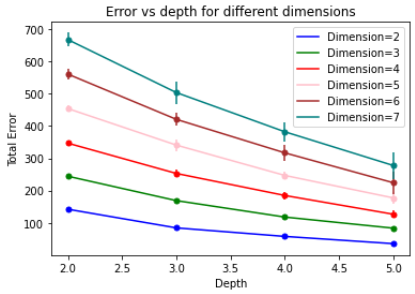


Figure 2: Error versus depth for different dimensions and $L1$ norm.

splitting to divide the regions of the plot into the regions where we follow different actions obtained in the plot. To account for randomness within the experimental setup, we sample the graph a total of 10 times and plot the mean and error bars as the variance for the sampling. The functional form we assume for the policy is given by:

$$f(x, \lambda) = \frac{\exp(\lambda \sum_{i=1}^{d} x_i)}{1 + \exp(\lambda \sum_{i=1}^{d} x_i)} \tag{24}$$

In Figure 2 we look at the error obtained in the tree based splitting procedure for different norms and depths and look at it's dependence on dimension. We also look at the behaviour of the splitting based mechanism of the final policy. We observe that the error decreases with an increase in dimension which aligns with our intuition of what should occur.

6 Discussion

Our work demonstrates that when we define interpretability to be given by region-wise constant maps, we can derive bounds on the cost of interpretability. The main insight of the method is that we can bound the cost of interpretability as a region-wise constant approximation on the final policy. Experimentally we also observe how it grows with the size and dimension of the state space. Possible limitations of this technique include the solving of the optimization problem 23. We solve this problem using a tree-based splitting mechanism which was informed according to intuition on how splitting worked for the CART algorithm Steinberg (2009). Further, we can improve the theoretical bounds by designing different techniques to optimize for 23. Also, we have not addressed how we would solve 23 on different state spaces and connectivity assumptions. For example, one could assume our state space is a space in $\mathbb{R}^d$ and we can split the space into polygonal regions instead of splits. Another limitation of the proposed technique is also that interpretability is obtained after obtaining the final policy, this could potentially lead to two sources of error in the accuracy estimation problem which are an approximation of the final policy and a piece-wise constant approximation of it. We must work on this further to get interpretability and accuracy in one shot.

In this paper, we worked with a specific definition of interpretability - region-wise constant maps. In future work we wish to generalize this notion to different definitions of what it means to be interpretable. We define the **cost of interpretability** problem as one in which we compute the optimal policy over a policy subspace and we come up with algorithms to compute upper bounds on the cost of interpretability. The ideas we have in mind are list based policies, trees, vector-field policies where the state space is a manifold, and Gaussian vector fields where our notion of interpretability is the given by the variance which is the sum of the Eigenvalues of the covariance matrix. This work would build upon similar work by Bertsimas et al. (2019) where interpretability

is defined for paths and models that occur during the course of development of a machine learning algorithm. Borrowing ideas from Saghafian (2023), we also wish to extend this line of work in the presence of model ambiguity i.e when we have a cloud of models and use the MEU-α metric to evaluate our approach.

7 Conclusion

In this paper, we came up with a notion of interpretability for a Markov Decision Process. We assumed a general definition of region-wise constant maps which could potentially give us flexibility as to how we choose the regions or do the partitioning of the space. We then use a blocked value iteration approach in the presence of a leader set to derive bounds on the cost of interpretability. We do this by performing value iteration on the leaders and under the assumption that every element of the subset follows the action proposed by the leaders. This gives us region-wise constant maps where we obtain policies that take the same action on every element of the partition. Further, we show that we can bound the cost of interpretability as the region-wise constant approximation on the final policy in 23. This leads us to design tree-based splits to perform the optimization which ensures we get a tree-based policy. By assuming we know noisy versions of the final policy for custom environments we show how the error in optimization scales with dimension and size of the problem.

References

Leif Bergerhoff, Joachim Weickert, and Yehuda Dar. Algorithms for piecewise constant signal approximations. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5, 2019. doi: 10.23919/EUSIPCO.2019.8902559.

Dimitris Bertsimas, Arthur Delarue, Patrick Jaillet, and Sebastien Martin. The price of interpretability. *arXiv preprint arXiv:1907.03419*, 2019.

Yehuda Dar and Alfred M Bruckstein. On high-resolution adaptive sampling of deterministic signals. *Journal of Mathematical Imaging and Vision*, 61:944–966, 2019.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1):163–223, 2003. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(02)00376-4. URL https://www.sciencedirect.com/science/article/pii/S0004370202003764. Planning with Uncertainty and Incomplete Information.

Claire Glanois, Paul Weng, Matthieu Zimmer, Dong Li, Tianpei Yang, Jianye Hao, and Wulong Liu. A survey on interpretable reinforcement learning. *Machine Learning*, pp. 1–44, 2024.

Julien Grand-Clément. *Robust and Interpretable Sequential Decision-Making for Healthcare*. Columbia University, 2021.

Daniel Hein, Steffen Udluft, and Thomas A Runkler. Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence*, 76:158–169, 2018.

Marek Petrik and Ronny Luss. Interpretable policies for dynamic product recommendations. In *UAI*, 2016.

Soroush Saghafian. Ambiguous dynamic treatment regimes: A reinforcement learning approach. *Management Science*, 10 2023. doi: 10.1287/mnsc.2022.00883.

Dan Steinberg. Cart: Classification and regression trees. 2009. URL https://api.semanticscholar.org/CorpusID:116184048.

Yichi Zhang, Eric B Laber, Marie Davidian, and Anastasios A Tsiatis. Interpretable dynamic treatment regimes. *Journal of the American Statistical Association*, 113(524):1541–1549, 2018.

Ruoqing Zhu, Donglin Zeng, and Michael R Kosorok. Reinforcement learning trees. *Journal of the American Statistical Association*, 110(512):1770–1784, 2015.