Visualizing High-Dimensional MDPs with Model-Free Monte Carlo

Sean McGregor
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97331
RLDM@seanbmcgregor.com

Rachel Houtman
College of Forestry
Oregon State University
Corvallis, OR 97331
rachel.Houtman@oregonstate.edu

Claire Montgomery
College of Forestry
Oregon State University
Corvallis, OR 97331
claire.montgomery@oregonstate.edu

Ronald Metoyer
College of Engineering
University of Notre Dame
Notre Dame, IN 46556
rmetoyer@nd.edu

Thomas G. Dietterich
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97331
tgd@oregonstate.edu

Abstract

Policy analysts wish to visualize a range of policies for large simulator-defined Markov Decision Processes (MDPs). One visualization approach is to invoke the simulator to generate on-policy trajectories and then visualize those trajectories. When the simulator is expensive, this is not practical, and some method is required for generating trajectories for new policies without invoking the simulator. The method of Model-Free Monte Carlo (MFMC) can do this by stitching together state transitions for a new policy based on previously-sampled trajectories from other policies. This "off-policy Monte Carlo simulation" method works well when the state space has low dimension but fails as the dimension grows. This paper describes a method for factoring out some of the state and action variables so that MFMC can work in high-dimensional MDPs. The new method, MFMCi, is evaluated on a very challenging wildfire management MDP whose state space varies over more than 13 million state variables. The dimensionality of forestry domains makes MFMC unrealistic, but factorization reduces the stitching operation to 8 state features. The compact representation allows for high-fidelity visualization of policies.

Keywords: Visualization, Markov Decision Processes, Model-Free Monte Carlo, Surrogate Modeling, State Space Factorization

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1331932.

1 Introduction

As reinforcement learning systems are increasingly deployed in the real-world, methods for justifying their ecological validity becomes increasingly important. For example, consider the problem of wildfire management in which land managers must decide when and where to fight fires on public lands. Our goal is to create an interactive visualization environment in which policy analysts can define various fire management polices and evaluate them through comparative visualizations. The transition dynamics of our fire management MDP are defined by a simulator that takes as input a detailed map of the landscape, an ignition location, a stream of weather conditions, and a fire fighting decision (i.e., suppress the fire vs. allow it to burn), and produces as output the resulting landscape map and associated variables (fire duration, area burned, timber value lost, fire fighting cost, etc.). The simulator also models the year-to-year growth of the trees and accumulation of fuels. Unfortunately, this simulator is extremely expensive. It can take up to 7 hours to simulate a single 100-year trajectory of fire ignitions and resulting landscapes. How can we support interactive policy analysis when the simulator is so expensive?

Our approach is to develop a surrogate model that can substitute for the simulator. We start by designing a small set of "seed policies" and invoking the slow simulator to generate several 100-year trajectories for each policy. This gives us a database of state transitions of the form (s_t, a_t, r_t, s_{t+1}), where s_t is the state at time t, a_t is the selected action, r_t is the resulting reward, and s_{t+1} is the resulting state. Given a new policy π to visualize, we apply the method of Model-Free Monte Carlo (MFMC) developed by Fonteneau et al. to simulate trajectories for π by stitching together state transitions according to a given distance metric Δ. Given a current state s and desired action $a = \pi(s)$, MFMC searches the database to find a tuple $(\tilde{s}, \tilde{a}, r, s')$ that minimizes the distance $\Delta((s, a), (\tilde{s}, \tilde{a}))$. It then uses s' as the resulting state and r as the corresponding one-step reward. We call this operation "stitching" (s, a) to $(\tilde{s}, \tilde{a})$. MFMC is guaranteed to give reasonable simulated trajectories under assumptions about the smoothness of the transition dynamics and reward function and provided that each matched tuple is removed from the database when it is used.

Fonteneau et al. (2010) apply MFMC to estimate the expected cumulative return of a new policy π by calling MFMC n times and computing the average cumulative reward of the resulting trajectories. We will refer to this as the MFMC estimate $V^{\pi}_{MFMC}(s_0)$ of $V^{\pi}(s_0)$.

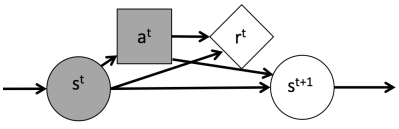
In high-dimensional spaces (i.e., where the states and actions are described by many features), MFMC breaks because of two related problems. First, distances become less informative in high-dimensional spaces. Second, the required number of seed-policy trajectories grows exponentially in the dimensionality of the space. The main technical contribution of this paper is to introduce a modified algorithm, MFMC with independencies (MFMCi), that reduces the dimensionality of the distance matching process by factoring out certain exogenous state variables and removing the features describing the action. In many applications, this can very substantially reduce the dimensionality of the matching process to the point that MFMC is again practical.

This paper is organized as follows. First, we introduce our method for factoring out exogenous variables. The method requires a modification to the way that trajectories are generated from the seed policies. Second, we conduct an experimental evaluation of MFMCi on our fire management problem. We show that MFMCi gives good performance for three different classes of policies and that for a fixed database size, it gives much more accurate visualizations.

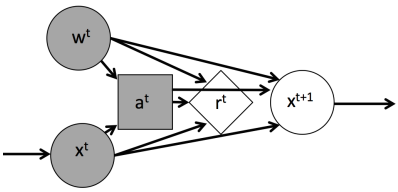
2 Factoring State to Improve MFMC

We work with the standard finite horizon undiscounted MDP (Bellman 1957; Puterman 1994), denoted by the tuple $\mathcal{M} = \langle S, A, P, R, s_0, h \rangle$. S is a finite set of states of the world; A is a finite set of possible actions that can be taken in each state; $P : S \times A \times S \mapsto [0, 1]$ is the conditional probability of entering state s' when action a is executed in state s; $R(s, a)$ is the finite reward received after performing action a in state s; s_0 is the starting state; and $\pi : S \mapsto A$ is the policy function that selects which action $a \in A$ to execute in state $s \in S$. We additionally define D as the state transition database.

In this paper, we focus on two queries about a given MDP. First, given a policy π, we wish to estimate the expected cumulative reward of executing that policy starting in state s_0: $V^{\pi}(s_0) = \mathbb{E}[\sum_{t=0}^{h} R(s_t, \pi(s_t))|s_0, \pi]$. Second, we are interested in visualizing the distribution of the states visited at time t: $P(s_t|s_0, \pi)$. In particular, let $v^1, \ldots, v^m$ be functions that compute interesting properties of a state. For example, in our fire domain, $v^1(s)$ might compute the total area of old growth Douglas fir and $v^2(s)$ might compute the total volume of harvestable wood. Visualizing the distribution of these properties over time gives policy makers insight into how the system will evolve when it is controlled by policy π.



(a) The standard MDP transition.



(b) MDP transition with *exogenous* (w) and *Markovian* variables (x).

Figure 1: MDP probabilistic graphical models.

We now describe how we can factor the state variables of an MDP in order to reduce the dimensionality of the MFMC stitching computation. State variables can be divided into Markovian and Time-Independent random variables. A time-independent

random variable x_t is exchangeable over time t and does not depend on any other random variable (including its own previous values). A (first-order) Markovian random variable x_{t+1} depends on its value x_t at the previous time step. In particular, the state variable s_{t+1} depends on s_t and the chosen action a_t. Variables can also be classified as endogenous and exogenous. The variable x_t is exogenous if its distribution is independent of $a_{t'}$ and $s_{t'} \setminus \{x'_t\}$ for all $t' \leq t$. Non-exogenous variables are endogenous. The key insight of this paper is that if a variable is time-independent and exogenous, then it can be removed from the MFMC stitching calculation as follows.

Let us factor the MDP state s into two vectors of random variables: w, which contains the time-independent, exogenous state variables and x, which contains all of the other state variables (see Figure 1). In our wildfire suppression domain, the state of the trees from one time step to another is Markovian, but our policy decisions also depend on exogenous weather events such as rain, wind, and lightning.

We can formalize this factorization as follows.

Definition 2.1. A Factored Exogenous MDP is an MDP such that the state (x, w) and next state (x', w') are related according to

$$Pr(x', w'|x, w, a) = Pr(w')Pr(x'|x, w, a). \tag{1}$$

This factorization allows us to avoid computing similarity in the complete state space s. Instead we only need to compute the similarity of the Markov state x. Without the factorization, MFMC stitches (s, a) to the $(\tilde{s}, \tilde{a})$ in the database D that minimizes a distance metric Δ, where Δ has the form $\Delta((s, a), (\tilde{s}, \tilde{a})) \mapsto \mathbb{R}^+$. Our new algorithm, MFMCi, makes its stitching decisions using only the Markov state. It stitches the current state x by finding the tuple $(\tilde{x}, \tilde{w}, a, r, x')$ that minimizes the lower-dimensional distance metric $\Delta_i(x, \tilde{x})$. MFMCi then adopts $(\tilde{x}, \tilde{w})$ as the current state, computes the policy action $\tilde{a} = \pi(\tilde{x}, \tilde{w})$, and then makes a transition to x' with reward r. The rationale for replacing x by $\tilde{x}$ is the same as in MFMC, namely that it is the nearest state from the database D. The rationale for replacing w by $\tilde{w}$ is that both w and $\tilde{w}$ are exchangeable draws from the exogenous time-independent distribution $P(w)$, so they can be swapped without changing the distribution of simulated paths.

There is one subtlety that can introduce bias into the simulated trajectories. What happens when the action $\tilde{a} = \pi(\tilde{x}, \tilde{w})$ is not equal to the action a in the database tuple $(\tilde{x}, \tilde{w}, a, r, x', w')$? One approach would be to require that $a = \tilde{a}$ and keep rejecting candidate tuples until we find one that satisfies this constraint. We call this method, "Biased MFMCi", because doing this introduces a bias. Consider again the graphical model in Figure 1. When we use the value of a to decide whether to accept $\tilde{w}$, this couples $\tilde{w}$ and $\tilde{x}$ so that they are no longer independent.

An alternative to Biased MFMCi is to change how we generate the database D to ensure that for every state $(\tilde{x}, \tilde{w})$, there is always a tuple $(\tilde{x}, \tilde{w}, a, r, x', w')$ for every possible action a. To do this, as we execute a trajectory following policy π, we simulate the result state (x', w') and reward r for each possible action a and not just the action $a = \pi(x, w)$ dictated by the policy. We call this method "Debiased MFMCi". This requires drawing more samples during database construction, but it restores the independence of $\tilde{w}$ from $\tilde{x}$.

Fonteneau et al. (2013; 2014; 2010) derived bias and variance bounds on the MFMC value estimate $V^{\pi}_{MFMC}(s_0)$. We prove improved bounds for debiased MFMCi in a paper now under review.

3 Experimental Evaluation

In our experiments we test whether we can generate accurate trajectory visualizations for a wildfire, timber, vegetation, and weather simulator of Houtman et al. The aim of the wildfire management simulator is to help US Forest Service land managers decide whether suppress a wildfire on National Forest lands. Each 100-year trajectory takes up to 7 hours to simulate.

The landscape is comprised of approximately one million pixels, each with 13 state variables. When a fire is ignited by lightning, the policy must choose between two actions: *Suppress* (fight the fire) and *Let Burn* (do nothing). Hence, $|A| = 2$.

The simulator spreads wildfires with the FARSITE fire model (Finney 1998) according to the surrounding pixel variables (X) and the hourly weather. MFMCi can treat the weather variables and the ignition location

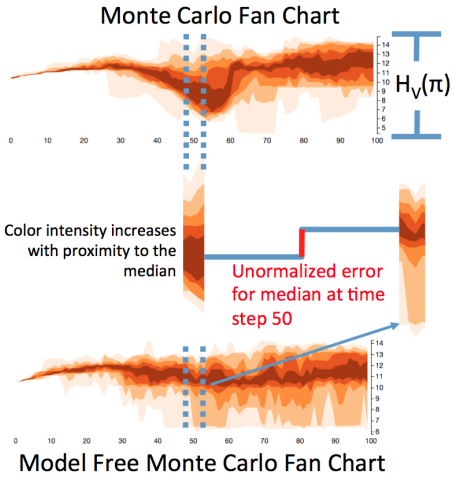


Figure 2: Top: A fan chart generated by Monte Carlo simulations from the expensive simulator. Bottom: A fan chart generated from the MFMC surrogate model. x axis is the time step and y axis is the value of the state variable at each time step. Each change in color shows a quantile boundary for a set of trajectories generated under policy π. Middle: Error measure is the distance between the median of the Monte Carlo simulations (left) and the median of the MFMC/MFMCi surrogate simulations (right). The error is normalized across fan charts according to $H_v(\pi)$, which is the Monte Carlo fan chart height for policy π and variable v.

as exogenous variables because the decision to fight (or not fight) a fire has no influence on weather or ignition locations. Further, changes in the Markov state do not influence the weather or the spatial probability of lightening strikes.

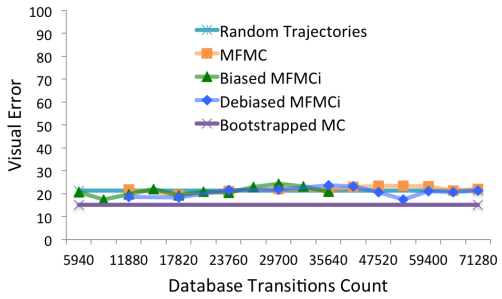
After computing the extent of the wildfire on the landscape, the simulator applies a cached version of the Forest Vegetation Simulator (Dixon 2002) to update the vegetation of the individual pixels. Finally, a harvest scheduler selects pixels to harvest for timber value.

We constructed three policy classes that map fires to fire suppression decisions. We label these policies INTENSITY, FUEL, and LOCATION. The INTENSITY policy suppresses fires based on the weather conditions at the time of the ignition and the number of days remaining in the fire season. The FUEL policy suppresses fires when the landscape accumulates sufficient high-fuel pixels. The LOCATION policy suppresses fires starting on the top half of the landscape, and allows fires on the bottom half of the landscape to burn (which mimics the situation that arises when houses and other buildings occupy part of the landscape). We selected these policy classes because they are functions of different components of the Markov and exogenous state.

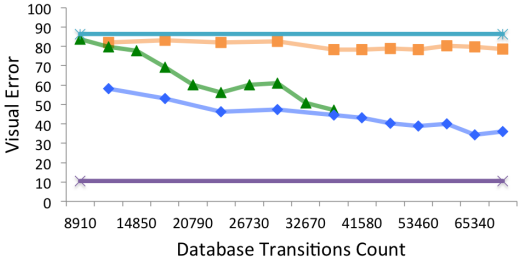
We seed the database with one trajectory for each of 360 policies whose parameters are sampled according to a grid over the INTENSITY policy space. We evaluate MFMCi by generating 30 trajectories for each policy from the ground truth simulator.

For our distance metric Δ_i, we use a weighted Euclidean distance computed over the mean/variance standardized values of 7 landscape features. An additional feature, the time step (*Year*), is added to the distance metric with a very large weight to ensure that MFMCi will only stitch from one state to another if the time steps match.

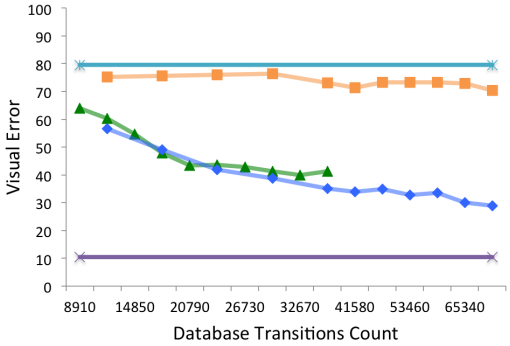
To visualize the trajectories, we employ the visualization tool MDPVIS (McGregor et al. 2015). The key visualization in MDPVIS is the fan chart, which depicts various quantiles of the set of trajectories as a function of time (see Figure 2). To evaluate the quality of the fan charts generated using surrogate trajectories, we define visual fidelity error in terms of the difference in vertical position between the true median as sampled with Monte Carlo trajectories and its position under the surrogate. Specifically, we define error(v, t) as the offset between the Monte Carlo location of the median and its MFMCi-modeled location for state variable v in time step t. We normalize the error by the height of the fan chart for the rendered policy ($H_v(\pi)$). The weighted error is thus $\sum_{v \in S} \sum_{t=0}^{h} \frac{\text{error}(v,t)}{H_v(\pi)}$. This error is measured for 20 variables related to the counts of burned pixels, fire suppression expenses, timber loss, timber harvest, and landscape ecology.



(a) Visual fidelity errors for a weather *intensity* policy class. Fires are suppressed based on a combination of the weather and how much time is left in the fire season.



(b) Visual fidelity errors for a ignition *location* policy class. Fires are always suppressed if they start on the top half of the landscape, otherwise they are always allowed to burn.



(c) Visual fidelity errors for a *fuel* accumulation policy class. Fires are always suppressed if the landscape is at least 30 percent in high fuels, otherwise the fire is allowed to burn.

Figure 3: Policy classes for the wildfire domain under a variety of distance metrics and sampling procedures.

3.1 Experimental Results

We evaluated the visual fidelity under three settings: (a) debiased MFMCi (exogenous variables excluded from the distance metric Δ_i; debiasing tuples included in the database D), (b) MFMC (exogenous variables included in Δ), and (c) biased MFMCi (exogenous variables excluded from Δ_i and the extra debiasing tuples removed from D). We also compare against two baselines that explore the upper and lower bounds of the visual error. First, we show that the lower bound on visual error is not zero. Although each policy has true quantile values at every time step, estimating these quantiles with 30 trajectories is inherently noisy. We estimate the achievable visual fidelity by bootstrap resampling the 30 ground truth trajectories and report the average visual fidelity error. Second, we check whether the error introduced by stitching is worse than visualizing a set of random database trajectories. Thus the bootstrap resample forms a lower bound on the error, and comparison to the random trajectories detects stitching failure. Figure 3 plots "learning curves" that plot the visualization error as a function of the size of the database D. The ideal learning curve should show a rapid decrease in visual fidelity error as $|D|$ grows.

4 Discussion

For each policy class, we chose one target policy from that class and measured how well the behavior of that policy could be simulated by our MFMC variants. Recall that the database of transitions was generated using a range of INTENSITY policies. When we apply the MFMC variants to generate trajectories for an INTENSITY policy, all methods (including random trajectory sampling) produce an accurate representation of the median for MDPVIS. When the database trajectories do not match the target policy, MFMCi outperforms MFMC. For some policies, the debiased database outperforms the biased databases, but the difference decreases with additional database samples. Next we explore these findings in more depth.

INTENSITY **Policy.** Figure 3a shows the results of simulating an INTENSITY policy that suppresses all moderate intensity fires that start late in the fire season. This policy suppresses approximately 60 percent of fires. There are many trajectories in the database that agree with the target policy on the majority of fires. Thus, to simulate the target policy it is sufficient to find a policy with a high level of agreement and then sample the entire trajectory. This is exactly what MFMC, MFMCi, and Biased MFMCi do. All of them stitch to a good matching trajectory and then follow it, so they all give accurate visualizations as indicated by the low error rate in Figure 3a. Unsurprisingly, we can approximate INTENSITY policies from a very small database D built from other INTENSITY policies.

LOCATION **Policy.** Figure 3b plots the visual fidelity error when simulating a LOCATION policy from the database of INTENSITY policy trajectories. When D is small, the error is very high. MFMC is unable to reduce this error as D grows, because its distance metric does not find matching fire conditions for similar landscapes. In contrast, because the MFMCi methods are matching on the smaller Markov state variables, they are able to find good matching trajectories. The debiased version of MFMCi outperforms the biased version for the smaller database sizes. In the biased version the matching operation repeatedly stitches over long distances to find a database trajectory with a matching action. Debiased MFMCi avoids this mistake. This explains why debiased MFMCi rapidly decreases the error while biased MFMCi takes a bit longer but then catches up at roughly $|D|$ =40,000.

FUEL **Policy.** The FUEL policy shows a best case scenario for the biased database. Within 7 time steps, fuel accumulation causes the policy action to switch from let-burn-all to suppress-all. Since all of the trajectories in the database have a consistent probability of suppressing fires throughout all 100 years, the ideal algorithm will select a trajectory that allows all wildfires to burn for 7 years (to reduce fuel accumulation), then stitch to the most similar trajectory in year 8 that will suppress all future fires. The biased database will perform this "policy switching" by jumping between trajectories to find one that always performs an action consistent with the current policy.

In summary, our experiments show that MFMCi is able to generalize across policy classes and that it requires only a small number of database trajectories to accurately reproduce the median of each state variable at each future time step. In general, it appears to be better to create a debiased database than a biased database having the same number of tuples.

Our stakeholders in forestry plan to apply the MFMCi surrogate to their task of policy analysis.

References

[Bellman1957] Bellman, R. 1957. *Dynamic Programming*. New Jersey: Princeton University Press.

[Dixon2002] Dixon, G. 2002. *Essential FVS : A User's Guide to the Forest Vegetation Simulator*. Number November 2015. Fort Collins, CO: USDA Forest Service.

[Finney1998] Finney, M. A. 1998. *FARSITE: fire area simulator model development and evaluation*. Missoula, MT: USDA Forest Service, Rocky Mountain Research Station.

[Fonteneau et al.2010] Fonteneau, R.; Murphy, S. A.; Wehenkel, L.; and Ernst, D. 2010. Model-Free Monte Carlo-like Policy Evaluation. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)* 217–224.

[Fonteneau et al.2013] Fonteneau, R.; Murphy, S. a.; Wehenkel, L.; and Ernst, D. 2013. Batch Mode Reinforcement Learning based on the Synthesis of Artificial Trajectories. *Annals of Operations Research* 208(1):383–416.

[Houtman et al.2013] Houtman, R. M.; Montgomery, C. A.; Gagnon, A. R.; Calkin, D. E.; Dietterich, T. G.; McGregor, S.; and Crowley, M. 2013. Allowing a Wildfire to Burn: Estimating the Effect on Future Fire Suppression Costs. *International Journal of Wildland Fire* 22(7):871–882.

[McGregor et al.2015] McGregor, S.; Buckingham, H.; Dietterich, T. G.; Houtman, R.; Montgomery, C.; and Metoyer, R. 2015. Facilitating Testing and Debugging of Markov Decision Processes with Interactive Visualization. In *IEEE Symposium on Visual Languages and Human-Centric Computing*.

[Puterman1994] Puterman, M. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, 1st edition.