Counterfactual Reasoning with Dynamic Switching Models for HIV Therapy Selection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Model-based approaches to disease progression are desirable because they potentially allow one to reason about the future effects of a series of treatment choices easily. However, the heterogeneity of treatment choices and response in HIV has made it challenging for model-based methods alone to succeed. We present a kernelised version of a model-based RL approach which allows us to accurately forward-simulate counterfactuals — how well might an alternative treatment have worked — to achieve state-of-the-art treatment recommendations.

1 Introduction

Model-based reinforcement learning techniques are appealing because they allow us to reason about possible future outcomes and events, and use that information to act appropriately in the present: a person that knows their future risk of stroke may choose to change their current diet and lifestyle to reduce that risk; a person who knows which HIV treatments will lead to future drug resistance may choose a different set of therapies. Model-based approaches assess these risks by first building a state-space model that captures the underlying process: we may posit that the patient's underlying physiological state s evolves based on actions a, and emits observations o based on some distributions $p(o|s, a)$. In contrast, kernel-based approaches assess these risks by finding patients with similar histories h; if two histories h and h' are similar, then perhaps the corresponding patients will experience the similar outcomes if they try the same action. However, they tend to fail if an agent finds itself in completely new territory as the dynamics of its not-so-near nearest neighbours may give a poor indication of what might happen next. Such situations are common when modelling disease progression, where there is often a long tail of distinct cases.

Kernel methods typically perform better in many applications (e.g. [1]) because modelling complex dynamical systems such as disease progression is difficult. To retain the benefits of having a model in which one can perform true planning and counterfactual reasoning, [8, 4] and [2] present methods for incorporating kernels directly into models such as Partially Observable Markov Decision Processes (POMDPs). These methods build dynamical systems by predicting next states on the basis of the next states of the agent's current nearest neighbours. However, they tend to fail if an agent finds itself in completely new territory — a common situation when modelling disease progression, where long tails of distinct cases may exist.

Recently, [9] used a Mixture-of-Experts (MoE) which switched between policies from a simple kernel regression (not a kernelised dynamical model like those above) and policies derived from a traditional state-space model learnt on the same data. Applying this model to produce HIV treatment recommendations, they found that for outlier patients, decisions based on a simplified model were better than incorrectly presuming treatment response would be similar to dissimilar patients. In this paper, we build on this idea by introducing the notion of *kernelised dynamical switching (KDS)*.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

Contributions The work of [9] mixes kernel and model-based approaches on a *policy* level, wherein the mixture-of-experts chooses between therapy policies for a patient. We instead propose an approach for combining kernel and model-based approaches on a *model* level; that is, we switch between kernels and model to predict next states for a patient at each particular time point. Thus we have a fully model-based system in which we can plan. By *smoothly* mixing between these predictions at each time step, it enables finer-grained modelling of a patient's possible treatment responses, and as a result, leads to more interpretable decisions. On a real cohort of HIV patients, we demonstrate that dynamically switching between model and kernel-based predictions significantly outperforms previous methods and produces superior treatment policies.

Related Work While there is little work on directly incorporating kernel-based predictions into model-based planning, there are some related threads. The first is *combining knowledge from different sources*. In this vein, Alonso et al. [7] trade off knowledge from both simulations and physical experiments by explicitly representing different sources of information and their associated costs using an entropy-based search. A related approach in [3] incorporates information from simulations as a prior in experiments. Similar efforts based on transfer learning have been proposed, for instance [14]. More closely related, are attempts are based on *regularising model-based predictions* using sample rollouts [13] or using kernel Bayes' rule [12]. However, leveraging kernel predictions and model-based learning specifically for simulating counterfactuals in planning is, to our knowledge, novel.

2 Kernelised Dynamic Switching Models

We introduce the notion of kernelised dynamic switching to leverage predictions of a POMDP model and kernel function for simulating counterfactuals in a model-based setting. Given a patient with a history h_t, we would like to choose the parameters of a POMDP model M and kernel function $k(\cdot,\cdot)$ that allow us to optimise over the predictions of future observations $p(o_{t+1}|h_t)$ (and hence rewards). We can formulate this in terms of the following loss function:

$$\mathcal{L} = \sum_s \sum_t^T \left(p_s(o_{t+1}|h_t) - \left(\phi_{t+1}\hat{p}_{Ms}(o_{t+1}|h_t) + (1-\phi_{t+1}) \sum_{h'_t} \alpha k(h_t, h'_t)\hat{p}_k(o_{t+1}|h'_t) \right) \right)^2 . \tag{1}$$

Here, $p_s(o_{t+1}|h_t)$ denotes the true probability of a future observation for a particular patient sequence s and history h_t. $\hat{p}_{Ms}(o_{t+1}|h_t)$ and $\hat{p}_k(o_{t+1}|h'_t)$ denote the estimates of this probability under both the POMDP model M and through a kernel-based regression respectively. α is a normalising constant. The ϕ parameters trade off the model-based and kernel-based predictions at each forward time step for each patient sequence, in order to minimise the loss.

Optimising the loss function via a multilayer perceptron The loss function in Equation 1 *cannot* be optimised directly since it requires knowledge of the true future observation probability at test time — something which we cannot observe. We introduce a *surrogate network* function $\hat{\phi} : \theta \to \mathbb{R}$ to approximate ϕ. Here, θ denotes the collection of POMDP and kernel parameters, as well as the quantile distances between patients in the data set. Our approximate function $\hat{\phi}$ is implemented as a multilayer perceptron network and is differentiable. During training time, this allows us to compute

$$\min_\theta \sum_{t'} (\phi_{t'}(\theta) - \hat{\phi}_{t'}(\theta))^2 + \lambda||\Psi(\theta)||, \tag{2}$$

where the true ϕ parameters are given by the softmax transformation of POMDP observation probabilities for each patient at each time step, and $\Psi(\theta)$ is a regularisation term with strength $\lambda > 0$. During forward simulation at test time, this is used to *predict* a suitable ϕ value for each forward time step t'. The future rewards may be computed analogously. In doing so, we can approximately optimise the loss function in Equation 1, and trade off the kernel and model-based predictions as necessary. The kernelised dynamic switching procedure is shown here as Algorithm 1.

Require:
$\hat{\phi}(\cdot, \theta)$: MLP prediction function, with parameters θ
$D = \{b_t\}_{n=1}^{N}$: belief states for each patient at time t
$H = \{h_t\}_{n=1}^{N}$: histories of each patient at time t
$k(\cdot, \cdot)$: kernel parameters
Ω, T, R: POMDP parameters

function KDS($\hat{\phi}$)
  while search depth has not been reached **do**
    Branch on an action a_t
    Predict $\hat{\phi}$ based on Ω, T, R and $k(\cdot, \cdot)$ and h_t, b_t
    if $\hat{\phi}$ exceeds randomly drawn ϵ **then**
      Sample o_t from POMDP with b_t
    else Sample o_t from nearest neighbours with $k(\cdot, \cdot)$
    Use the same $\hat{\phi}$ to weight rewards R
    Update belief b_t according to o_t and a_t
    Add o_t, a_t and r_t to existing history h_t
  Backpropagate values up through the search tree to get a_t^*
  return Updated b_t and optimal action a_t^*

3 Experiments

We demonstrate the performance of KDS with a toy example and an application to the HIV therapy selection task. In both cases, we compare the performance of KDS against using a kernelised planner alone, a POMDP planner, the mixture-of-experts approach described in [9], and a random switching policy[1]. We evaluate our results using two off-policy evaluation methods, namely weighted importance sampling (WIS) and doubly-robust off-policy evaluation (DR) [5].

Toy Example Consider a system that evolves deterministically through 4 states: S_1, S_2 or S_3, and finally absorbs in S_4. Each agent has a variant that belongs to one of two types: A and B. Agents with variants of type A deterministically go through state S_2, and agents with variants of type B deterministically go through S_3. At each stage, there are three actions available: 0, 1 or 2. At each time step, the agent observes its variant (which is of one of the two types), as well as its reward. By construction, a four-state POMDP cannot learn the optimal policy for this model since the dynamics depend on the hidden type of the agent's variant[2]. We compare the performance of KDS against the aforementioned baselines. Our surrogate network consists of 15 input units and a hidden layer of 25 units. For the kernelised planning approach, we use a kernel that matches based on the length of the agent's history, action choices, and an observation dependent on the type of variant. We use a forward search depth of 4 across all baselines.

HIV Therapy Selection We make use of a subset of the EuResist database [15] consisting of HIV genotype and treatment response data for 32 960 patients, together with their corresponding CD4$^+$ and viral load measurements, gender, age, risk group, and the past treatments recorded. The database has previously been used to build models such as the therapy alignment model, to predict the outcome of a particular therapy [16, 10]. The rewards are specified as in [9][3]. We compare KDS to the following baselines on a hold-out set of 3 000 patients: (i) the long-term alignment kernel based on [1], where the policy chooses a therapy for a patient based on the nearest neighbours with the highest long-term reward; (ii) a 20-state POMDP, where the observation space consists of (a) binning the values of the viral load using a log scale, (b) 70 mutations that may occur as a result of therapy together with a patient's CD4$^+$ count, gender, risk group. Here, we model time in discrete increments of 6 months; (iii) A mixture-of-experts approach which combines POMDP and kernel policies from (i) and (ii) using a neural network architecture with a gating layer as in [9]. We perform a forward search for therapy choices that optimise outcomes over a 3 year horizon (5 - 6 forward

[1]randomly switching between using the POMDP for action selection or using the kernel.
[2]Further details concerning experimental setup are in the supplement
[3]See supplement for setup details

	DR	WIS
Random	–5.84 ± 2.61	–7.79 ± 3.71
Kernel	4.39 ± 1.74	4.86 ± 2.85
POMDP	3.09 ± 1.16	3.84 ± 2.42
MoE	5.62 ± 1.02	5.81 ± 2.37
KDS	**6.08 ± 1.14**	**6.19 ± 1.03**

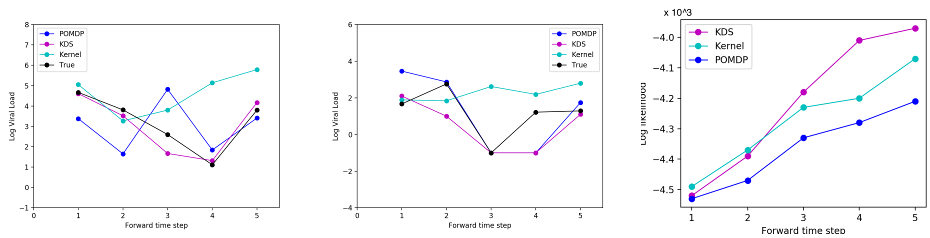
Table 1: Comparison of performances of KDS vs. baselines for the toy example.

	DR	WIS
Random	–7.31 ± 3.72	–11.48 ± 4.36
Kernel	9.35 ± 2.61	6.42 ± 3.93
POMDP	3.37 ± 2.15	3.86 ± 2.38
MoE	11.52 ± 1.31	12.25 ± 2.01
KDS	**12.47 ± 1.38**	**14.25 ± 1.27**

Table 2: Comparison of performances of KDS vs. baselines for HIV therapy selection.

steps). Our surrogate network here consists of 100 input units and 2 hidden layers of 50 units each. Table 2 compares the performance of KDS against the baselines. A higher value indicates a better performing treatment policy over the long-term future.

4 Discussion



(a) (b) (c)

Figure 1: (a) - (b) Forward simulation of viral load in two sample patients across baselines; (c) Comparison of log-likelihood across baselines

Combining POMDPs and kernels on a *model* level produces different policies to mixing on a *policy* level. The results from Tables 1 and 2 show that KDS and MoE produce different policies respectively. In both cases, KDS outperforms the MoE approach.

Dynamically switching between the kernel and POMDP produces the best policy. The results from Tables 1 and 2 show that KDS outperforms its competitors and produces policies with higher accumulated rewards. Our post-hoc analysis suggests that the kernel based approach tends to be used early for predicting outcomes, while the POMDP is used later. One possible explanation for this would be that over time, a patient's treatment history gradually diverges from its nearest neighbours; as a result, there may be fewer patients that share similar characteristics and hence fewer action choices available from the data itself to consider when planning. This is the point beyond which only the POMDP is used for decision-making.

KDS enables us to forward simulate in a *fully model-based* setting whilst combining kernel-based knowledge, thus leading to policies that can be easily interpreted. In simulating counterfactuals we can inspect the results not only in terms of future actions or treatment recommendations, but also holistically for the kinds of observations, or mutations and biomarker values we can expect. A particular example of this is shown in Figure 1(a)[4]. In this particular case, we observe that forward simulation via KDS enables us to simulate counterfactuals that are closer to the ground truth in comparison to the other baselines. A counterexample of this is provided in Figure 1(b), where simulation using the KDS policy would produce similar outcomes to the ground truth early on, but different outcomes later on. In this instance, the KDS policy is potentially better than the ground truth

[4]Additional comparisons with other baselines are provided in the supplement

policy since it is able sustain a suppressed viral load for longer[5]. Note that forward simulation of observations such as the viral load, cannot be achieved using a MoE approach. Importantly, because we can trace through the forward predictions which drive the policies learned, we can assess the feasibility of future treatment options more effectively.

References

[1] Jasmina Bogojeska, Daniel Stöckel, Maurizio Zazzi, Rolf Kaiser, Francesca Incardona, Michal Rosen-Zvi, and Thomas Lengauer. History-alignment models for bias-aware prediction of virological response to hiv combination therapy. In *AISTATS*, pages 118–126, 2012.

[2] Byron Boots, Geoffrey Gordon, and Arthur Gretton. Hilbert space embeddings of predictive state representations. *arXiv preprint arXiv:1309.6819*, 2013.

[3] Mark Cutler and Jonathan P How. Efficient reinforcement learning for robots using informative simulated priors. 2015.

[4] Kenji Fukumizu, Le Song, and Arthur Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14(1):3753–3783, 2013.

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

[6] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[7] Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1557–1563. IEEE, 2017.

[8] Yu Nishiyama, Abdeslam Boularias, Arthur Gretton, and Kenji Fukumizu. Hilbert space embeddings of pomdps. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 644–653. AUAI Press, 2012.

[9] Sonali Parbhoo, Jasmina Bogojeska, Maurizio Zazzi, Volker Roth, and Finale Doshi-Velez. Combining kernel and model-based learning for HIV therapy selection. *In Proceedings of the AMIA Summit on Clinical Research Informatics (CRI)*, 2017.

[10] Mattia CF Prosperi, Andre Altmann, Michal Rosen-Zvi, Ehud Aharoni, Gabor Borgulya, Fulop Bazso, Anders Sönnerborg, Eugen Schülter, Daniel Struck, Giovanni Ulivi, et al. Investigation of expert rule bases, logistic regression, and non-linear machine learning techniques for predicting response to antiretroviral treatment. *Antivir Ther*, 14(3):433–42, 2009.

[11] Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *Journal of Machine Learning Research*, 12(May):1729–1770, 2011.

[12] Yang Song, Jun Zhu, and Yong Ren. Kernel bayesian inference with posterior regularization. In *Advances in Neural Information Processing Systems*, pages 4763–4771, 2016.

[13] Erik Talvitie. Model regularization for stable sample rollouts. In *UAI*, pages 780–789, 2014.

[14] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

[15] Maurizio Zazzi et al. Predicting response to antiretroviral treatment by machine learning: The euresist project. *Intervirology*, 55:123–127, 1 2012.

[16] Maurizio Zazzi, R Kaiser, A Sönnerborg, Daniel Struck, Andre Altmann, Mattia Prosperi, M Rosen-Zvi, A Petroczi, Y Peres, E Schülter, et al. Prediction of response to antiretroviral therapy by human experts and by the euresist data-driven expert system (the eve study). *HIV medicine*, 12(4):211–218, 2011.

[5] A viral load of -1 indicates a viral load below detection limits

Supplementary Material

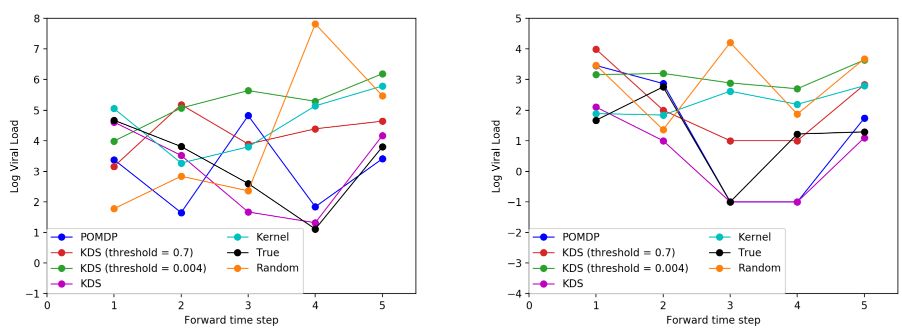
Toy Example

Consider a system that evolves deterministically through 4 states: S_1, S_2 or S_3, and finally absorbs in S_4. Each agent has a variant that belongs to one of two types: A and B. Agents with variants of type A deterministically go through state S_2, and agents with variants of type B deterministically go through S_3. At each stage, there are three actions available: 0, 1 or 2. At each time step, the agent observes its variant (which is of one of the two types), as well as its reward, which is given by:

$$S_1 \begin{cases} r(a_0) = -10 \\ r(a_1) = 5 \\ r(a_2) = 5 \end{cases} \qquad S_2 \begin{cases} r(a_0) = 0 \\ r(a_1) = 5 \\ r(a_2) = -10 \end{cases} \qquad S_3 \begin{cases} r(a_0) = 0 \\ r(a_1) = -10 \\ r(a_2) = 5 \end{cases} \qquad S_4 \left\{ r = 0 \right.$$

Thus, the optimal policy for all agents is to initially take either action 1 or 2. Agents with variants of type A subsequently transition to S_2 where the optimal action is action 1, while agents with variants of type B transition to S_3 where the optimal action is action 2. Action 0 is safe in states S_2 or S_3. By construction, a four-state POMDP cannot learn the optimal policy for this model because the dynamics depend on an additional hidden variable, the type of the agent's variant. Without the variant information, from the POMDP's perspective, it is equally likely to transition from S_1 or S_2 starting from S_0; not knowing where it will end up, it will initially suggest the safe policy of selection action 0 at the second time-step. For the kernelised planning approach, we use a kernel that matches based on the length of the agent's history, action choices, and an observation dependent on the type of variant. Such a choice will lead to optimal policies for agents with common variants. However, agents with rare variants will match to some arbitrary other agent, and we can expect the performance of the kernelised planner for those agents to be poor. Here, falling back on the POMDP will produce the optimal policy. We use a forward search depth of 4 across all baselines.

HIV Therapy Selection



(a) (b)

Figure 2: (a) - (b) Forward simulation of viral load in two sample patients across all baselines

We are interested in optimising the therapy choice for a particular patient based on long-term outcomes. The rewards in this case are specified by:

$$r_t = \begin{cases} -0.7 \log V_t + 0.6 \log T_t - 0.2|M_t|, & \text{if } V_t \text{ is above detection limits} \\ 5 + 0.6 \log T_t - 0.2|M_t|, & \text{if } V_t \text{ is below detection limits,} \end{cases}$$

where V_t is the viral load (in copies/mL), T_t is the CD4$^+$ count (in cells/mL), and $|M_t|$ is the number of mutations at time t respectively. This function is identical to the reward function presented in [9]. It penalises instances where a patient's viral load increases and rewards instances where a patient's CD4$^+$ count increases. It also penalises on the basis of the number of mutations a patient has at a particular time, as these may ultimately contribute to resistance and therapy failure. There is a bonus for if the viral load is below detectable limits to sustain this over time. The action space in this setting consists of the 312 frequently occurring drug combinations in the cohort. Here, we model time in discrete increments of 6 months. We compared the performance of KDS to the baselines mentioned in the paper as well as two additional baselines where we set ϵ from Algorithm 1 to 0.004 and 0.7 respectively. Figures (2a) and (2b) illustrate forward simulation of the viral loads across all the baselines in the two patients described in the paper.

Reinforcement Learning Background

Many problems, including therapy selection, involve making a sequence of decisions with long-term consequences. The reinforcement learning (RL) framework formalises the sequential decision-making process for HIV therapy selection as a series of exchanges between an agent and its environment. At each time step, the agent selects an action a and the environment returns observations o as well as an immediate reward r. Given a history of length t, $h_t = \{a_1, o_1, r_1 \ldots, a_t, o_t, r_t\}$, the agent's goal is to choose the subsequent action to maximise the discounted sum of its expected rewards, $\mathbb{E}[\sum_t \gamma_t r_t]$, where $\gamma \in [0, 1)$ trades off between current and future rewards. The decision-making task may be formulated as a POMDP [6]. A POMDP m is defined by a finite set of hidden states $\mathcal{S}$ (e.g. a patient's true physiological state), actions $\mathcal{A}$ and observations $\mathcal{O}$. A transition function $T(s'|s, a)$ specifies the probability of transitioning from state s to s' when taking an action a. Similarly, an observation function $\Omega(o|s, a)$ specifies the probability of observing o from state s when taking action a. The reward function $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ specifies the immediate reward that an agent receives upon performing an action from a particular state.

Model-based RL methods learn models of the domain by approximating the dynamics $T(s'|s, a)$ and $\Omega(o|s, a)$ for each s and a. The model is subsequently used to compute an optimal policy $\pi^*(s, a)$ via planning [11], which may produce further samples from which the model can be further refined. The two phases of model learning and planning are typically interleaved repeatedly.