Memory Development with Heteroskedastic Bayesian Last Layer Probabilistic Deep Neural Networks
George Velentzas, Costas S Tzafestas, Mehdi Khamassi

To cite this version:

HAL Id: hal-04249890
https://hal.sorbonne-universite.fr/hal-04249890
Submitted on 24 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Extended Abstract: Memory Development with Heteroskedastic Bayesian Last Layer Probabilistic Deep Neural Networks

Georgios Velentzas\(^1\), Costas Tzafestas\(^2\), Mehdi Khamassi\(^1\)

Abstract— Learning world models in model-based Reinforcement Learning (MBRL) enables sample-efficient learning, while avoiding model-bias via uncertainty estimates of the transition dynamics. Moreover, it enables using Model Predictive Control (MPC) or Recurrent Neural Networks (RNN) for efficient action planning. However, current uncertainty estimates of the prediction (transition) step are not provided in a closed-form that could potentially be used for analytically estimating changes in the dynamics function. Here, we propose a hybrid method to capture both the aleatoric (data) uncertainty via Probabilistic Deep Neural Networks (PrDNN), and the epistemic (model) uncertainty through a Bayesian Last Layer.

I. INTRODUCTION AND RELATED WORK

Learning world models of the environment dynamics enables Model Predictive Control (MPC), a process through which an agent can anticipate environment states resulting from specific actions, allowing to plan future action sequences. This has been widely used in MBRL and applied to real-world robotic tasks (e.g., target reaching with a robotic arm, locomotion control of a half-cheetah, etc.) [1]. However, when uncertainty of transitions is overlooked, model-bias [2] and sample-inefficiency are induced, as described in [3].

Probabilistic MPC methods have been proposed [4], while methods for estimating the uncertainty of predictions in deep learning have been thoroughly studied [5], [6], [7], [3], [8], [9]. The validity of some approaches has been criticized [10], nevertheless, the enriched usability of such estimates is unquestionable; for example, adding an epistemic component [10], [9]. The validity of some approaches has been questioned, thereby demonstrating that current uncertainty estimates of the predictive distributions limits the tools needed for real-time capture of changes in the dynamics function.

Here, we derive an analytic way of estimating the predictive (transition) distribution of the dynamics function by incorporating multivariate Bayesian regression at the last layer of a PrDNN [3], [12]. Including heteroskedastic noise, we utilize both types of uncertainty [13], aleatoric (data) and epistemic (model), to recognize changes in the dynamics that require to learn and memorize different world models. We introduce a memory schema, and we show numerical simulations of a 3-dof robotic arm in a target-reaching task.

\(^{1}\)Sorbonne University, CNRS, Institute of Intelligent Systems and Robotics, Paris, France. georgiev@isir.upmc.fr, mehdi.khamassi@sorbonne-universite.fr

\(^{2}\)Division of Signals, Control and Robotics, School of Electrical and Computer Engineering, NTUA, Greece. ktzaf@cs.ntua.gr

II. METHODS

We assume a world of infinite environments, where each environment with index \(e = 1, 2, \ldots \) is characterised by its own dynamics. If \(s \in \mathbb{R}^p\) denotes the state representation vector, \(u \in \mathbb{R}^u\) denotes the action vector, \(x = [s^T, u^T]^T\), where \(x \in \mathbb{R}^s\), denotes the concatenation of those and \(y = s' - s\) denotes the state displacement vector after a control \(u\) is applied for a time period of \(dt\), we make the assumption that the transition dynamics of the environment with index \(e\) is described by \(y = f_e(x) + n_e(x)\), where \(f_e : \mathbb{R}^e \mapsto \mathbb{R}^p\) is the dynamics function, and \(n_e(x) \sim \mathcal{N}_p(0, \Sigma^e(x))\) is a heteroskedastic transition noise which encapsulates aleatoric uncertainty. With \(\mathcal{P}^e\) denoting the state displacement transition distribution, the above implies that \(\mathcal{P}^e(y|x) = \mathcal{N}(f_e(x), \Sigma^e(x))\). The agent’s objective is to be able to perform a task in each different environment, without having any prior knowledge of the environment’s index, and quickly adapt after each transition. In this work, we assume that each interaction with the environment is taking place in episodes, where each episode’s time horizon is unknown.

A. Modeling the dynamics function for each environment

If \(X\) and \(Y\) denote the matrices of \(n\) transition observations from a single environment, where the \(i\)-th rows of \(X\) and \(Y\) are \(x_i^T\) and \(y_i^T\) respectively, we model the problem as multivariate Bayesian linear regression with learned non-linear features and heteroskedasticity, encapsulating both the epistemic and aleatoric uncertainty, such as

\[
Y = \Phi_\omega(X)B + E_{\omega,\theta}(X) \tag{1}
\]

where \(\Phi_\omega(X)\) is a \(n \times d\) matrix with its \(i\)-th row being \(\phi_\omega(x_i)^T\), and \(\phi_\omega : \mathbb{R}^s \mapsto \mathbb{R}^d\) denoting a parameterized (by weights \(\omega\) non-linear feature mapping function, \(B\) is a \(d \times p\) matrix of Bayesian weights. If \(\Sigma_{\omega,\theta} : \mathbb{R}^s \mapsto \mathcal{A}\) denotes a parameterized function (by weights \(\omega, \theta\)), with \(\mathcal{A}\) being the set of positive definite matrices, \(E_{\omega,\theta}(X)\) is a \(n \times p\) random matrix with independent rows, where the \(i\)-th row is \(e_{\omega,\theta}(x_i)^T\), with \(e_{\omega,\theta}(x_i) \sim \mathcal{N}_p(0, \Sigma_{\omega,\theta}(x_i))\). For simplicity, we omit the parameter weights notation and take the matrix vectorization of both sides of equation Eq.1, thus,

\[
\text{vec}(Y^T) = [\Phi(X) \otimes I_p] \text{vec}(B^T) + \text{vec}(E(X)^T) \tag{2}
\]

We denote \(y = \text{vec}(Y^T)\), \(\beta = \text{vec}(B^T)\), \(\varepsilon = \text{vec}(E(X)^T)\), \(\Phi_I = \Phi(X) \otimes I_p\), therefore Eq. 2 can be simply written as

\(y = \Phi_I\beta + \varepsilon\), which implies that \(y|\beta \sim \mathcal{N}_{n \times p}(\Phi_I\beta, S_{\text{NET}})\), with \(S_{\text{NET}} = \text{diag}(\Sigma(x_1), \ldots, \Sigma(x_n))\). This notation is convenient, as \(y\) denotes the concatenation of state displacement
computing the likelihood \( P \) of recent transition observations. In the control phase, a sliding window of length \( W \) is used for extracting efficient Cross Entropy Method \([14]\). At each time step, \( P \) is computed with the \( \text{model} \). During the first episode, the transition observations \( x \) is trained. The experiment is performed in episodes, where the \( \text{episode} \) denotes the environment's index. The endpoint’s color denotes the dynamic \( \text{model} \). In conclusion, our method for deriving predictive distribution of \( \text{model} \) is already sufficient for MPC to drive the arm to the target. In Episode 7 the environment is reversed to \( e = 1 \). In the first two episodes \( 0.02 \) sec \( P_2 \) is initially employed, but the change in the dynamics is detected and \( P_1 \) is re-engaged from timestep 3 onward, with \( \text{MPC} \) being able to drive the arm to the target. In our experiments, the agent was able to capture all changes in 4 different environments by demonstrating fast adaptation and employing the appropriate model, without populating the memory with redundant memory blocks. Due to space limitations we only show a single example.

In conclusion, our method for deriving predictive distributions with full uncertainty estimates, by disentangling epistemic uncertainty from aleatoric uncertainty, enabled fast recognition and adaptation to changes in the dynamics of the environment. This paves the way for new applications of world models in model-based deep reinforcement learning.
REFERENCES


