Context-Adaptive Reinforcement Learning using Unsupervised Learning of Context Variables

Hamid Eghbal-zadeh[1,2, *] **Florian Henkel**[2,*] **Gerhard Widmer**[1,2]

[1] LIT Artificial Intelligence Lab, Johannes Kepler University, Linz, Austria
[2] Institute of Computational Perception, Johannes Kepler University, Linz, Austria
hamid.eghbal-zadeh@jku.at

Abstract

In Reinforcement Learning (RL), changes in the context often cause a distributional change in the observations of the environment, requiring the agent to adapt to this change. For example, when a new user interacts with a system, the system has to adapt to the needs of the user, which might differ based on the user's characteristics that are often not observable. In this Contextual Reinforcement Learning (CRL) setting, the agent has to not only recognise and adapt to a context, but also remember previous ones. However, often in CRL the context is unknown, hence a supervised approach for learning to predict the context is not feasible. In this paper, we introduce **C**ontext-**A**daptive **R**einforcement **L**earning **A**gent (CARLA), that is capable of learning context variables in an unsupervised manner, and can adapt the policy to the current context. We provide a hypothesis based on the generative process that explains how the context variable relates to the states and observations of an environment. Further, we propose an experimental protocol to test and validate our hypothesis; and compare the performance of the proposed approach with other methods in a CRL environment.

1 Introduction

In Reinforcement Learning, an agent interacts with an environment through receiving observations, executing actions, and receiving rewards. The goal of the agent is to maximise the cumulative reward that is defined based on the task at hand. In some scenarios however, the behaviour of the environment as well as the distribution of the observations can change over time. Under certain conditions, the change in the observation distribution is caused by some variability that changes the *context* of the environment. Therefore, a change in context affects the distribution of the environment's observations. As such changes may occur numerous times, not only does the agent have to adapt to the new contexts, but it also has to *remember* the previous ones. This problem is known as Contextual Reinforcement Learning (CRL).

As an example, consider a setting where users interact with a website, and the goal of the website is to adapt to the user's needs, which might change depending on the current user. However, the behaviour of the user – the environment – is actually affected by some *unobserved* parameters such as age and gender. If the goal of the agent – the website – is to adapt to the needs of the user, it is often helpful to be able to infer the user's characteristics and adapt to them. Another example is a robot that sees the world through a camera, where the time of the day (day/night) or the surrounding location can affect how the robot perceives its environment. Hence, it is crucial that an agent can detect a context, and be able to adapt to it.

[*] Equal contribution.

Pre-registration workshop NeurIPS (2020), Vancouver, Canada.

Several approaches have been proposed to address this problem. For example, models that can adapt to the changes of the environment by having better exploration strategies [6, 16], have been used to tackle environments with changing dynamics. As the context changes, the exploitation of the current policy is no longer as effective, and the agent's previous policy will no longer be suitable to tackle the changes in the environment. Hence, the agent needs to explore new observations, in order to accumulate more reward. Another approach to adapt to new contexts, is to use options in a hierarchical reinforcement setting, where a meta-policy switches between a set of available policies [1, 5]. In [7], Hallak et al. define a Contextual Markov Decision Process (CMDP), as a constrained Partially Observable Markov Decision Process (POMDP), where each context is parameterised as an MDP. In this setting, they propose a solution to tackle CRL assuming a fixed observation space over different contexts, and the agent picking a suitable policy, given the available context. In contrast to the Contextual MDPs as a special case of POMDPs, Jiang et al. [10] propose a generalisation of MDPs and POMDPs known as Contextual Decision Processes (CDPs), where there is a general context space that the observations are drawn from. Although this formulation is quite general, this work focuses on problems with low Bellman ranks, which corresponds to MDPs with low-rank transition matrix, or small observation space.

In this paper, we provide a definition for Contextual Reinforcement Learning that assumes changing the context, affects the distribution of the states of the environment, resulting in a change in the distribution of the observations. Our definition is motivated by the generative process in a contextual world, where the context variables affect the states of the generative model of the world. Given this definition, we provide a solution using unsupervised learning of the context variable that allows for a better adaptation of the policy based on the context. More generally, in this work we are trying to answer the following questions:

1. Does knowing the context variable help the policy to better adapt to different contexts?
2. What characteristics does a predictive model need to predict context from observations?
3. Can our learnt context variable help the policy to better adapt to different contexts?

In order to answer these questions, we conduct a set of experiments to test the performance of an agent with and without knowing the context variable. Additionally, we conduct experiments to investigate whether disentanglement is actually helpful for estimating the context. Further, using our proposed approach, we estimate the context variable in an unsupervised manner, and compare the performance of agents with and without this estimated variable.

2 Related work

Contextual RL: Contextual settings have been mainly explored in Multi-armed bandits [13]. Hallak et al. [7] propose contextual MDPs (CMDPs), extending the standard MDP formulation with multiple contexts that change the underlying dynamics. They introduce an algorithm that is able to detect different contexts and optimize the CMDP. However, their work is focused on low-dimensional observation-spaces and, only a small number of fixed contexts is considered. In contrast, our work is proposed for high-dimensional observation-space such as images, and can deal with a variable number of contexts as it incorporates a continuous multivariate context variable. Another work formulates contextual decision processes as a generalization of MDPs and POMDPs [10], where the observations themselves or their history, respectively, form the context. Our approach differs from this formulation by explicitly distinguishing between context and observations, and having a generative view on the observations based on states that depend on a context.

Eysenbach et al. [5] propose to use mutual information between the context and the observations as a learning signal, and the entropy of the policy over different contexts as a regularisation term to improve exploration, and better adapt to the change of context. This approach assumes the context variable is known to the policy. Achiam et al. [1] propose VALOR and use a variational auto-encoder (VAE) that first encodes context to trajectory via policy, and subsequently decodes the trajectory back to the initial context using a probabilistic recurrent decoder that assigns high probabilities to trajectories that are unique to a context. Their approach also assumes that the context variable is known, and is used as a supervised signal to train the decoder. A different model-based approach is explained in [16]. An ensemble of dynamic models is trained to predict next observation given the current observation and action. The variance over the output of this ensemble is used as intrinsic reward to train the policy. This approach improves the exploration, which is helpful in contextual settings as the agent better adapts to new contexts.

Representation Learning for RL: Recently, several approaches to learn better representations for RL have been proposed. Higgins et al. [9] propose DARLA following a two stage learning approach. First, an agent learns disentangled state representations using β-VAEs [8] from a high dimensional observation space. Second, based on the previously learned disentangled state representation, the agent has to learn a policy to solve a given task. In contrast to our work, the learned disentangled state representations are not explicitly used to infer different contexts, and the policy is directly learned using the disentangled features, while as we will explain, in our work the disentanglement is only used for learning the context variable, and the agent can learn an unconstrained representation from the observations, in addition to the context variable. Similarly, Stooke et al. [18] propose to decouple representation learning from the RL task. By applying image augmentation [12] and a contrastive loss for learning state representations from raw pixel-observations, they are able to outperform end-to-end trained RL agents on various environments. Such a contrastive learning approach was also applied in [17].

3 Problem definition

In this section, we provide a generative view to Contextual Reinforcement Learning (CRL), and detail the relation between context variables and the states of the environment. Based on this view, we provide a solution for CRL that can automatically recognise the change in the states of the environment, accordingly predict the new context, and adapt the policy to the new context.

3.1 Contextual Reinforcement Learning

A Partially Observable Markov Decision Process (POMDP) is defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O})$, with $\mathcal{S}$ being the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the transition probabilities, and $\mathcal{R}$ the reward function. In this setting, the agent does not directly observe the true states of the environment, but receives observation $o \in \Omega$. This observation is generated from the underlying system state s and the received action a, according to the probability distribution $o \sim \mathcal{O}(o \mid s, a)$.

In this work we consider finite-horizon episodic Contextual POMDPs (CPOMDPs). At the beginning of each episode an agent will encounter a specific POMDP depending on a randomly sampled context $c \in \mathcal{C}$, which we assume to not change over time within an episode. While for regular POMDPs, the goal of an RL agent is to learn a policy $\pi(a \mid o)$ that maximizes the expected cumulative reward, in CPOMDPs the agent has to learn a policy $\pi(a \mid o, c)$ that further depends on a context c.

3.2 A Generative View on Contextual Reinforcement Learning

Generative Process: We assume a generative process is in place such that everything within the environment is happening in a two-step generative process. First, a multivariate latent random variable z is sampled from a distribution $P(z)$, where z corresponds to semantically meaningful factors of variation of the observations (e.g, shape, colour of the objects; density of objects). Second, the observation x is sampled from a conditional distribution $P(x \mid z)$. We assume that the observation space has higher dimensionality than the semantic space, hence, the data space can be explained with substantially lower dimensional and semantically meaningful latent variable z, and is mapped to the high dimensional observation space x.

Generative Process in Contextual Reinforcement Learning: In Contextual Reinforcement Learning, we assume that the environment $\mathrm{E}_z(o_t, a_t)$ generates the next observation o_{t+1}, given the current observation o_t and action a_t, i.e., $o_{t+1} = \mathrm{E}_z(o_t, a_t)$, with z being a variable controlling its statics (e.g, shape or size of objects). In our generative view, the observations of an episode are generated from a generative model $\mathrm{E}_z(o_t, a_t)$ in 3 steps as follows. In the first step, a multivariate latent random variable $c \in \mathcal{C}$ is sampled from a distribution $P(c)$, where c corresponds to a context. In the second step, a multivariate latent random variable z is sampled from a conditional distribution $P(z \mid c)$, where z corresponds to the state of the environment that controls the statics, defining how the environment generates the next observation, given he current observation and action during an episode. In the third step, the next observation o_{t+1} is generated from the environment's generative model $\mathrm{E}_z(o_t, a_t)$.

4 Proposed Approach

In this section, we propose **C**ontext-**A**daptive **R**einforcement **L**earning **A**gent (CARLA), which is capable of adapting to new contexts in an environment, without any supervision or knowledge about the available contexts.

CARLA consists of two parallel networks: a *context network*, and a *representation network*. The context network aims at learning the context variable, while the representation network is aiming at learning a suitable representation from the environment. The output of these two networks are then further feed into the policy network, where an adaptive policy is formed given the environment variables and the context variable. The policy network then adapts the current policy, based on the context variable. A block-diagram of CARLA is provided in Figure 1 (left).

As detailed in Section 3.2, our assumption in the generative process is that the context variables define the statics of the environment, which in turn defines the distribution of the observations within an episode. The aim of the context network is to reverse this process and estimate the context vector given the observations. As shown in Figure 1 (right), it contains two main modules: a feature disentanglement module and a context learning module. The context network first estimates the the environment's statics, and further uses it to learn the context variable. This context factor is then feed to the policy network, in order to adapt the policy to the current context.

The feature disentanglement module is an encoder part of a Variational Autoencoder (VAE) [11], which is trained with annealing the Kullback Leibler (KL) Divergence term of the Evidence Lower Bound (ELBO). The VAE is trained using random samples drawn from an experience replay buffer [14]. The context learning module is trained online given the observations received in each episode, along with the representation network and the policy network by optimizing the RL objective. This module learns upon the disentangled states of the environment, extracted using the feature disentanglement module explained above.

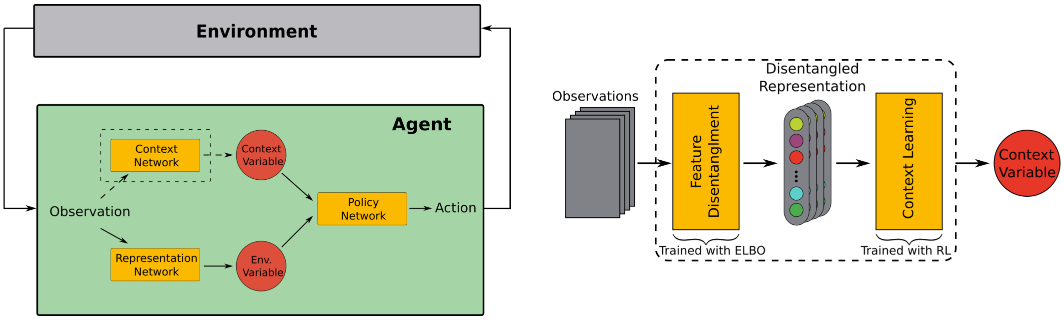

Figure 1: left) Block diagram of CARLA. right) Block diagram of the Context Network. Hidden variables are shown with circles, while processing units are shown with squares.

The graphical models for various approaches in CRL is provided in Figure 2. As can be seen, our model has a different graphical model than DARLA [9] and VALOR [1]. The main idea in CARLA, is to learn disentangled factors from the environment using pre-selected training data, in a similar manner to DARLA. In contrast, CARLA uses a recurrent context network that can build the sequential relationship for the disentangled factors, to predict the context variable, which might be useful in a partially-observable setting to infer the dynamics. Further, CARLA allows the agent to learn an unconstrained representation from the environment during training the agent via interacting with the environment. Although VALOR uses a sequential decoder, it differs from CARLA in various ways. For example VALOR assumes an observed context, in contrast to CARLA which estimates the context variable using a sequence of disentangled factors.

5 Experimental Protocol

In this section, we detail our experimental setup and evaluation strategy, in order to demonstrate the effectiveness of the proposed approach in tackling CRL. In our evaluation, we are testing several hypotheses to answer the following questions:

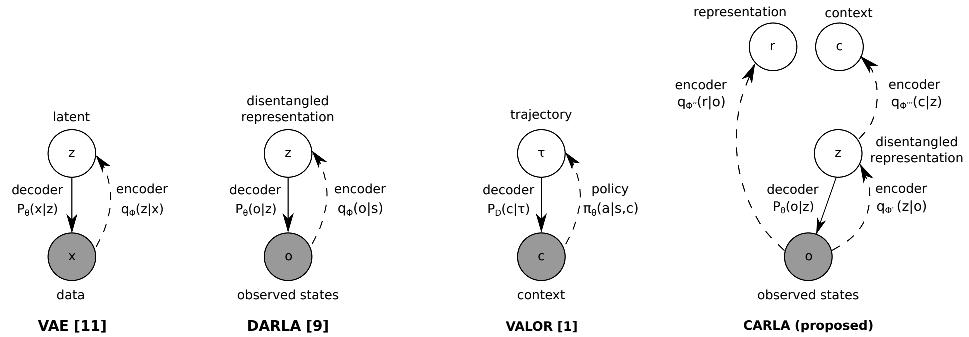

Figure 2: Comparison of the graphical models in different approaches. The solid lines represent generation, and the dashed line represent inference. Gray circles represent observed variables, while white circles represent hidden variables.

Does knowing the context help the performance of the agent?
To investigate this, we conduct an experiment testing whether adding a context variable as an additional input to the policy network helps the policy to better adapt to context changes. For evaluation, we will compare the performance in terms of the cumulative reward of a baseline agent that does not use the context information, to an agent that has access to this additional information.

What characteristics does a predictive model need to *predict* context from observations?
In this experiment, we evaluate how well a context can be learned from the observations. Our goal is to determine on the one hand which representations capture the most information about the context in an unsupervised manner and on the other hand which modelling technique (feed-forward vs. sequential) is more suitable to learn the context variable. To this end, we investigate whether feature disentanglement helps in learning the context, by training unsupervised VAEs. We compare a vanilla VAE [11] that does not perform feature disentanglement, to an annealed VAE [2] that incorporates it. Subsequently, given the features extracted by each of the VAEs, we compare a feed-forward classifier to a recurrent one, for learning the context from a single, or a sequence of representations, extracted by the VAEs from observations, respectively. A train-test split on the observations and their respective context label is used to evaluate the generalisation of the context classifiers.

Can our learnt context variable help the policy to better adapt to different contexts?
Finally, to test our full setup, we compare CARLA with the context variable being jointly trained using the RL objective, against two baseline agents in terms of the accumulated reward. The first agent does not have the additional context information, which basically is CARLA without the context network. For the second agent, we remove the representation network and the context learning from CARLA leaving only the feature disentanglement, which is thus similar to DARLA.

For all our experiments, we use a modified dynamic obstacle gridworld environment [3, 4] as follows. The task of an agent will be to reach a goal position, while collecting and avoiding certain objects. The agent receives a reward of +1 and -1 for *good* and *bad* objects, respectively. Whether an object is good or bad will be based on a certain context, e.g., a specific configuration of differently colored and shaped objects that allow for a clear distinction between contexts. We consider a fully and a partially observable variant of this environment to properly compare feed-forward and recurrent context learning, by either showing the whole grid or a subset. For the VAEs, we use the architecture from [9]. However, as shown in [2], annealing the KL term provides a better disentanglement than the β-VAE, which was used in [9]. Hence, we use the annealing technique proposed in [2] for training the VAE. Similar to [9], we use an experience replay buffer to draw i.i.d. samples for optimizing the VAE objective. For collecting the observations we follow two different strategies. First, we will store observations that are received by the agent during training in an online fashion. Second, as this might cause the VAE to overfit to its recent experience and not generalize across all possible observations, we will use a different agent to collect the observations that simply avoids all objects and moves around in the world, similar to what is proposed in [9]. To train the RL agents, we use vanilla policy gradient as well as the hyperparameters reported in [1]. For the context network, we compare two architectures: a 1-layer LSTM (64 neurons), and a 1-layer MLP (64 neurons), and the policy network is always a 2-layer MLP (64 neurons). For all feed-forward hidden layers with non-linearities, we apply ReLU activation [15].

Acknowledgments

Work supported by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (grant agreement number 670035).

References

[1] Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

[2] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[3] Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[4] Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. `https://github.com/maximecb/gym-minigrid`, 2018.

[5] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[6] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[7] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

[8] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[9] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[10] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[11] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

[12] Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. *arXiv preprint arXiv:2004.13649*, 2020.

[13] John Langford and Tong Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, 2007.

[14] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3-4):293–321, 1992.

[15] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

[16] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. *arXiv preprint arXiv:1906.04161*, 2019.

[17] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[18] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.