Causal Discovery and Reinforcement Learning: A Synergistic Integration

Arquímides Méndez-Molina arquimides.mendez@gmail.com

Eduardo F. Morales emorales@inaoep.mx

L. Enrique Sucar esucar@inaoep.mx

Instituto Nacional de Astrofísica, 'Optica y Electrónica Luis Enrique Erro # 1 Tonantzintla Puebla 72840 México

Abstract

Both Reinforcement Learning (RL) and Causal Modeling (CM) are indispensable parts in the road for general artificial intelligence, however, they are usually treated separately, despite the fact that both areas can effectively complement each other in problem solving. On one hand, the interventional nature of the data generating process in RL favors the discovery of the underlying causal structure. On the other hand, if an agent knows the possible consequences of its actions, given by causal models, it can make better selections of them, reducing exploration and, therefore, accelerating the learning process. Also, ensuring that such an agent maintains a causal model for the world it operates in, improves interpretability and transfer learning, among other benefits. In this article, we propose a combination strategy to provide an intelligent agent with the ability to simultaneously learn and use causal models in the context of reinforcement learning. The proposed method learns a Causal Dynamic Bayesian Network for each of the agent actions and uses those models to improve the action selection process. To test our algorithm, experiments were performed on a simple synthetic scenario called the "coffee-task". Our method achieves better results in policy learning than a traditional model-free algorithm (Q-Learning), and it also learns the underlying causal models. We believe that the results obtained reveal several interesting and challenging directions for future work.

Keywords: Reinforcement Learning; Causal Discovery; Causal Dynamic Bayesian Networks.

1. Introduction

One of the goals of Artificial Intelligence is to create autonomous agents that learn through interaction with their environment. One framework that emerges from this purpose is Reinforcement Learning (RL) Sutton and Barto (2018), in which an agent explores the environment to learn a task through rewards associated with each of the actions taken on each situation (state) along the way. Determining the best action to take on each state is known as an optimal policy. RL algorithms have shown to be effective in discovering optimal policies in various domains such as video games Vinyals et al. (2019), robotics Akkaya et al. (2019), and medical care Gottesman et al. (2018). However, current reinforcement learning systems do not take advantage of high-level processes such as Causal Models Pearl (2009) to exploit patterns beyond the associative ones.

Causal Discovery (CD) aims to uncover the cause-effect relationships between different variables Pearl (2009). Learning causal relations in the real world is a challenging task for

which many algorithms have been proposed. One of the main limitations of causal discovery is the need to make interventions on the model variables to guarantee the correct model. In several domains, such as medicine, it is often very difficult, expensive or unethical to make such interventions. However, once the causal model is known, it can provide the information needed for an intelligent system to predict the effect of actions/interventions in a system, improving planning and making counterfactual predictions.

Both Reinforcement Learning (RL) and Causal Modeling (CM) play an essential role in artificial intelligence; however, they are usually treated separately, despite the fact that both areas can effectively complement each other. Although, there seems to be a natural connection between these fields, the different research communities are still separated. The combination of RL and CM promises several advantages. On the one hand, the interventional nature of the data that can be obtained by the agent and the temporal order of these observations obtained from RL favor the discovery of the causal structure, although with some limitations since only the effects of the agent's actions can be discovered. On the other hand, if an agent knows the possible consequences of its actions, it can make a better selection of them. This is particularly relevant in RL, because the knowledge given by a causal model can significantly reduce the exploration process and therefore accelerate learning. In addition, the learned CM can be transferred to other tasks and used for more interpretable models.

At present, the first works focusing on the relationship between RL and CM are beginning to emerge and they can be divided into three main groups:

1. Use Causal Models as side information to improve Reinforcement Learning algorithms ($CM \rightarrow RL$): These works assume a known causal model relating the state, action and reward variables so the RL agent can make a better action selection while learning the policy for the given task, in this way the agent can learn faster. Most of these works are limited to Multi Armed Bandit (MAB) settings Lee and Bareinboim (2019, 2018); Lattimore et al. (2016) and also a completely defined Causal Model is assumed, which is hard to obtain in the real world.

2. Use RL data to learn causal relationships of the environment ($RL \rightarrow CM$): The main limitation of this reduced group of works is that the structure is given, so only the parameters are learned Madumal et al. (2019) or the agent, instead of learning a policy for the task, uses reinforcement learning as a search strategy to find the graph that achieves the best reward Zhu and Chen (2019a).

3. Simultaneously do both tasks ($RL \leftrightarrow CM$): Finally, there are some more recent works Nair et al. (2019); Kansky et al. (2017) that simultaneously do both tasks: learn causal effects from an agent communicating with the environment, and then optimize its policy based on the learned causal relations ($RL \leftrightarrow CM$). Among their limitations we can mention the use of only observational data and the structural assumptions used according to the specific problem.

In this article we focus on the last point ($RL \leftrightarrow CM$): How can we provide an intelligent agent with the ability to simultaneously learn and use causal models in the context of reinforcement learning? To this end, we learn one causal model from each of the agent's

actions using the interventional nature of the data collected by the agent while leaning the task. These models are represented as causal Dynamic Bayesian Networks (DBN's) relating states variables at time (i) with state and reward variables at time $(i+1)$, and are discovered using a score-based greedy algorithm. On the other hand, an algorithm is proposed to use the (partially) learned causal models to guide the agent to take actions that lead to positive reward states and thus accelerate the learning process. Both approaches are combined in an algorithm called *Causal-Q-Learning*.

We tested Causal-Q-Learning on the "Coffee-Task" Boutilier et al. (1995) and show that it learns the task in fewer episodes than a traditional reinforcement algorithm, and obtains a causal model that can be directly transferred to similar tasks.

2. Related Work

Combing RL and CM has been proposed on psychology works like Zhu and Chen (2019b). The author contrasts a model-free system that learns to repeat actions that lead to reward with a model-based system that learns a probabilistic causal model of the environment which it then uses to plan action sequences. Evidence from neuropsychology suggests that these two systems coexist in the brain, both competing and cooperating with each other Dolan and Dayan (2013).

Model-based reinforcement learning, e.g., Sutton (1991); Deisenroth and Rasmussen (2011); Silver et al. (2017), has focused on learning the transition $(s'|s,a)$ and reward functions $(r|s,a)$, while a causal model explicitly models the nature of the relationships between a set of state variables.

Some works have focused on how to combine RL and CM to improve transfer between similar tasks. In Nair et al. (2019) is presented a method for causal induction using visual observations for goal directed tasks. During each training episode, the agent samples each training environment and uses an interaction policy π_I to probe the environment and collect a trajectory of visual observations. Then, using supervised learning, they train a causal induction model F, which takes as input the trajectory of observational data and constructs C, which captures the underlying causal structure. The predicted structure C is given as input to the policy π_G conditioned on goal g, which learns to use the causal model to efficiently complete a specified goal in a given training environment. At test time, F and π_G are fixed and the agent is evaluated on new environments with unseen causal structures. Their main limitation is that the causal relations are just direct binary relations.

Schema networks Kansky et al. (2017) are an example of how learning causal relationships and using them to plan can result in better transfer than model-free policies. In this work, the authors introduce an object-oriented generative physics simulator capable of disentangling multiple causes of events and reasoning backwards through causes to achieve goals. Schema Networks can learn the dynamics of an environment directly from data. Compared with methods like Asynchronous Advantage Actor-Critic and Progressive Networks on a suite of breakout game variations; Schema Networks report better results on training efficiency and zero-shot generalization, demonstrating faster, more robust learning and better transfer. Strong assumptions about the structure of the causal models are made in this work.

Another example can be seen in Gonzalez-Soto et al. (2018). The authors propose a decision-making procedure in which an agent holds beliefs about its environment, which are

used to make a choice and then are updated using the observed outcome. The agent, using its current beliefs, generates a local causal model and chooses an action from it as if that model was the true one. Then, after it observes the consequences of its actions, its beliefs are updated according to the observed information to make a better choice the next time. The agent, in addition to learning a policy to choose actions, will also learn a causal model from the environment, since the causal model it forms will approximate the true model. In the experiments however, only the case where (i) the causal model is completely known and (ii) only the structure is known, are analyzed. The problem of discovering the variables itself and the connections between them is left as future work.

3. Proposed Method

We consider goal-conditioned Markov decision processes, which have an underlying causal structure describing the behavior of the environment Nair et al. (2019). A goal-conditioned MDP is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{C}, \mathcal{P}, \mathcal{G}, r, \gamma, \phi)$, where $\mathcal{S}$ denotes the state space; $\mathcal{A}$ is the set of possible actions; $\mathcal{X}$ is the set of causal macro-variables[1] describing the state of the environment at a high abstraction level (see Chalupka et al. (2015)); $\mathcal{C}$ is the set of causal graphs relating variables on $\mathcal{X}$ at two consecutive time steps, one graph for each action $a \in A$; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ defines the probability transition function between states given an action; $\mathcal{G}$ is the goal space where its elements are vectors of variables on $\mathcal{X}$; $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathrm{R}$ is the reward function which yields the immediate reward conditioned on the goal $g \in \mathcal{G}$; γ is the discount factor; and $\phi : \mathcal{S} \to \mathcal{X}$ is a function which associates the state space to the macro-variables space. The goal is to learn an optimal policy $\pi^*_g : \mathcal{S} \times \mathcal{G} \to \mathcal{A}$ which maximizes the total expected return $R = \sum_k^\infty \gamma^k r(s_k, a_k, g)$.

In the considered scenarios, the agent has no information of the models (transition, reward or causal relations) and we want to integrate causal discovery during reinforcement learning so that the system can learn faster π than a traditional RL algorithm, and also discover causal models. In this section, we first present the task, then describe Causal-Q-Learning, and then explain each of its main steps.

3.1 Coffee task description

We will illustrate our method on the coffee task introduced in Boutilier et al. (1995). An office robot is ask to go to a coffee shop, buy coffee, and return to deliver it to a user in her office. On the way it could be raining, so the robot will get wet unless it has taken an umbrella (available at the office) before leaving. The robot must learn a sequence of optimal actions that allow it to solve the task with the highest possible reward. We assume the problem can be modeled as a completely observable MDP. A state S is described by six binary state variables: SL, the location of the robot (office or coffee shop); SU, whether the robot has an umbrella; SR, whether it is raining; SW, whether the robot is wet; SC, whether the robot has coffee; and SH, whether the user has coffee.

The robot has four actions: GO, changes its location and the robot can get wet if it rains and it does not have an umbrella; BC (buy coffee) causes it to hold coffee if it is in

1. A high-level or macro variable is a function defined from other variables Chalupka et al. (2015), which summarizes information about some aspect of the data structure.

the coffee shop; *GU* (get umbrella) causes it to hold an umbrella if it is in the office; and *DC* (deliver coffee) causes the user to hold coffee if the robot has coffee and is in the office. The action *GO* is simplified; by executing the *GO* action the robot successfully navigates from one place to another. However, we assume a stochastic environment. The robot gets a reward of 0.9 whenever the user has coffee plus a reward of 0.1 whenever it is dry, in addition, it receives a positive reward of 0.05 on each sub-goal (being in the coffee shop, buying the coffee and returning to the office). A penalty of -1.0 is given if the robot does not take the umbrella and it rains, and -0.1 is given in all other cases.

3.2 Simultaneous RL + CD

The combination strategy between RL and CD is illustrated in Algorithm 1. First, the agent performs (T) reinforcement learning (RL) episodes, in which it starts to learn a policy to solve the task, and also collects interventional data for each of its actions. Specifically, the values of the state variables prior to the action (s_i) and the values of the state variables after the action (s_j) plus the value of the reward variable at time (j) are stored for each action. At the end of the (T) episodes, partial causal models (CD) are learned using the datasets of each action. Note that every time the agent performs causal discovery, the corresponding models are updated according to the new collected data. In the following (T) episodes, RL is performed but now using the partial causal models ($RL + CD$). This cycle of RL, CD, RL+CD is repeated until the agent has learned the optimal policy and the causal models can not be improved.

Algorithm 1: Simultaneos RL + CD

```
   input :
   output: A value function Q, a set of causal models G
 1 while True do
 2 |   for i ← 0 to T_steps do
 3 |   |   Q ← reinforcement_learning()
 4 |   end
 5 |   foreach a ∈ A do
 6 |   |   G [a] ← causal_discovery(a)
 7 |   end
 8 |   for i ← 0 to T_steps do
 9 |   |   Q ← rl_using_causal_model(G)
10 |   end
11 |   if ¬(policy_improvement ∨ model_improvement) then
12 |   |   break;
13 |   end
14 end
15 return Q, G
```

3.3 Causal Discovery using RL data

The coffee task can be solved using a model-free reinforcement learning algorithm, however, there are a number of causal relationships of the type $(S, A \rightarrow S')$ and $(S, A \rightarrow R)$ that can be discovered and used to learn the optimal policy more efficiently. One way to represent such relationships is through a Causal DBN for each action. In this scenario the Markov assumption is fulfilled, therefore, the effects of a given action in the next state and the reward

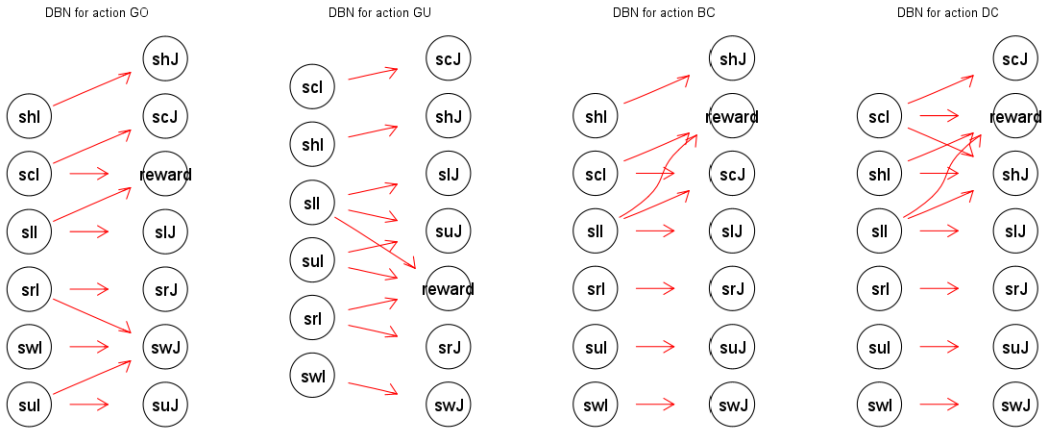


Figure 1: Ground truth causal DBNs relating state variables and reward at consecutive time steps (I) and (J) for each action in the coffee task.

are completely determined by the current state of the world, so we can use a "two-slice" causal DBN. We have one set of nodes representing the state of the world prior to the action, S_i, another set representing the state of the world after the action, S_j, and directed arcs representing causal relations between them.

Figure 1 depicts the ground truth causal DBNs for each action in the coffee task. The agent has no prior knowledge of any of these models. However, we will use these models as a guide in order to measure the quality of their discovery with the data collected by the agent. The value of a state variable S_j as a result of executing an action depends on the values of state variables S_i that have edges to S_j in the DBN. For example, after executing the action GO the robot will be wet (swJ) at time j if it is raining (srI) and has no umbrella (suI) or if it is already wet (swI) at time i. In the set of state variables at time j we include R that represents the immediate reward obtained after executing the specified action. For simplicity, we have also discretized this variable to two possible values: $(+)$ indicating a positive reward and $(-)$ indicating a negative reward.

While the agent is learning the task we save, on each episode, all the values of the state variables at time (i), the action, the values of the state variables at time (j) and the reward, and then we group the data by the corresponding action. When we reach the number of episodes (T), we stop to collect data and the learning process, regardless of whether or not the agent has learned the optimal policy. At this point, for the coffee task, we have 4 data sets (one for each action) with 13 variables (6 for state at time i, 6 for state at time j and one for the reward at time j), and a different number of observations depending on how often the agent has executed the corresponding action during the RL process. To discover the causal models, we can use any of the existing causal discovery algorithms in the literature which can be supplemented with additional information. In our experiments we use the Hill Climbing (HC) algorithm from the bnlearn package Scutari (2010). HC is a score-based structure learning algorithm. We supplement the algorithm with a set of constraints as additional information (for example, no variable of time (j) can cause a variable at time (i) and variables at the same time cannot cause each other). We compare the discovered graph

Algorithm 2: Action selection based on interventional queries.

Input : A state s sense by the agent, a set of actions A, a set of causal DBN models C, one for each action.
Output: An action a.

```
1  probs ← {}
2  foreach a ∈ A do
3  |  CM ← C[a]
4  |  p ← P(reward = (+)|s, CM)
5  |  probs ← probs + p
6  end
7  max_value = max(probs)
8  max_index = index(probs, max_value)
9  if max_value > threshold then
10 |  return a[max_index]
11 end
12 return None
```

against the ground truth using the structural Hamming distance (minimum number of edge insertions, deletions, and changes needed to transform a model into another) as accuracy measure (lower is better). Those models may not be perfect at the beginning, but become better with more data.

3.4 RL using Causal Models

It has been shown Molina et al. (2020); Feliciano-Avelino et al. (2021) that a causal model relating state, action and reward can be used to accelerate a reinforcement learning process by guiding the action selection process. In practice, it is very unlikely that the agent will have such a model beforehand. In fact, our main goal is for the agent to be able to learn/use such a model while learning the task. The models used in these works are not DBNs; however, their action selection algorithm can be easily adapted to work with several DBNs (one per action), see Algorithm 2. On line 4 we calculate the probability that the corresponding action, given the value of the state variables at a given time, will lead to an immediate positive reward. As future work, we could think of a sequence of actions instead of a single action leading to a future positive reward. We calculate that value for each action, and then we just select the one with higher probability. If that probability is higher than a certain threshold, then we select the corresponding action. So the agent will execute that action no matter if the current exploration strategy suggests another action.

4. Experiments and results

The experiments performed aim to test our hypothesis that by combining task learning with causal discovery in a single algorithm can reduce the learning time and obtain the corresponding causal models. We test our method in the stochastic version of the coffee task described above, and compare it against Q-Learning Watkins and Dayan (1992). The optimal policy to solve such a task is to take the umbrella (in case it rains on the way), go to the coffee shop, buy the coffee, return to the house and deliver the coffee to the user. Our hypothesis here is that our agent (that learns and uses causal models) can learn the optimal policy faster (in terms of episodes) when compared to the model-free agent.

In RL it is well known that a fundamental problem is determining the exploration-exploitation strategy. ϵ indicates how often the agent must explore (take a random action in a given state), instead of exploit (take the best action in the given state according to the value function). The lower the value, the less the exploration. This exploration-exploitation dilemma has an impact on our proposed method because more exploration favors the causal discovery process, as we will see in the results, but eventually we need to explore less to achieve convergence on learning the optimal policy. To compare the algorithms, we performed experiments using two different exploration strategies. The first uses a fixed exploration in all episodes at different levels ($\epsilon \in \{0.9, 0.5, 0.1\}$). The second uses a decayed exploration at different levels ($\lambda \in \{fast = 0.01, normal == 0.005, slow == 0.001\}$). In this strategy, both agents start at ($\epsilon_0 = 0.9$) and then exponentially decrease the value of the epsilon on each episode by (λ) using the following formula: $\epsilon(t) = \epsilon_0 e^{-\lambda t}$

In all the experiments, we use a learning rate of $\alpha = 0.1$ and a discount factor of $\gamma = 0.8$. We run 10 trials of each experiment for each algorithm and report the average reward among the episodes. In each episode, both agents start at a random state (excluding the terminal state) and stop when they reach the goal state (the user has the coffee) or when a maximum number of steps (99) is executed. First, we run the Q-Learning algorithm to determine the total number of episodes for our experiments. We see that 600 episodes is enough to achieve convergence (actually the Q-Learning agent seems to learn the optimal policy at around episode 250). Our agent uses $T = 30$ and $th = 0.7$. T indicates the number of episodes in which it alternates between performing traditional RL and RL using the model. That is, during the first 30 episodes, it performs RL, learns the causal models with the collected data, and in the next 30 episodes it performs RL using the learned causal models and repeats the cycle. th is the causal threshold used by the action selection algorithm (see Algorithm 2).

In Figure 2 we can see an analysis of how the task learning behaves for our algorithm (orange line) versus the baseline (blue line) using the fixed strategy at different levels of exploration. The y-axis indicates the average reward, and the x-axis the episode number. In subplots from (d) to (f), we can see how the model discovery (just for our agent) behaves for each of the actions. On the y-axis we observe the structural hamming distance (shd) which measures the similarity between the learned model and the ground truth and on the x-axis we have the number of episodes. In scenario (a-d) we can see that none of the agents can learn the optimal policy, because both are taking random actions 90% of the time. However, it is clear that our agent achieves high rewards each time it uses the causal models (e.g. from episodes 210 to 240). Which is explained on Figure 2(d) were we can see how all causal models are correctly discovered by episode 200, so every time that our agent uses the causal model for action selection it takes the action that gives a positive immediate reward. The opposite happens in scenario (c-f). Due to the lack of exploration ($\epsilon = 0.1$), the learned causal models are incorrect. So every time our agent uses the causal models, it takes incorrect actions that lead to the accumulation of negative reward. A middle ground between these scenarios can be found in scenario (b-e).

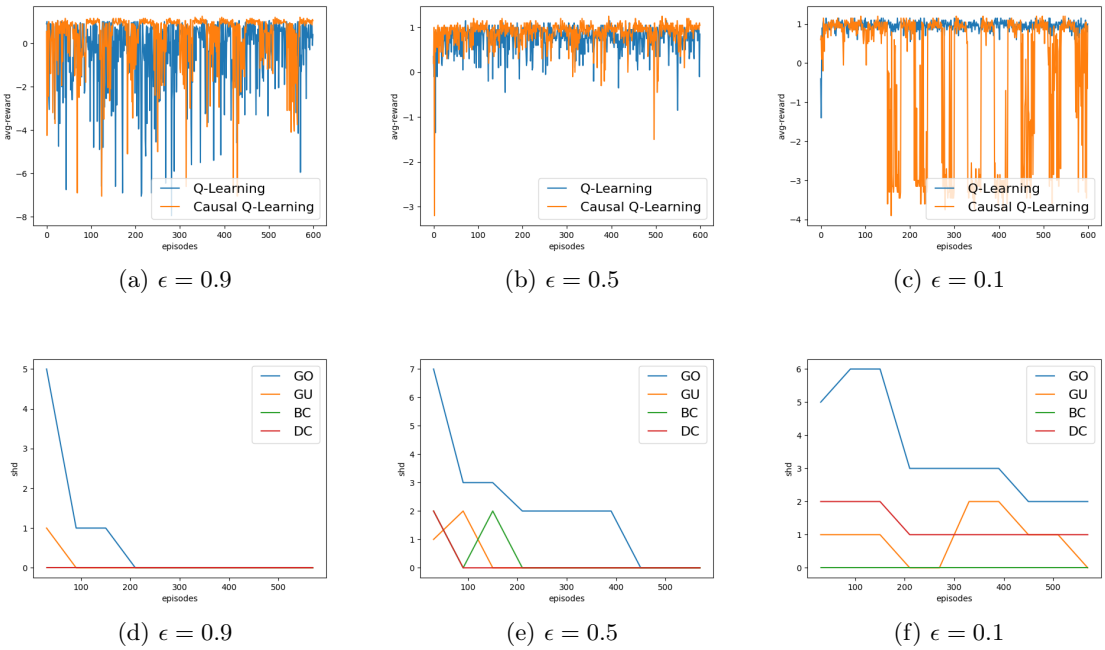
The results of these first experiments show us the need to find a balance between exploration and exploitation for the correct functioning of our proposed method. For this reason, we then performed experiments where the exploration decayed exponentially as the episodes progressed. In Figure 3, we can see the results using the decayed strategy at different levels of decay ($\lambda \in \{fast = 0.01, normal = 0.005, slow = 0.001\}$). The first thing we can high-

light is how, unlike the previously used fixed strategy, in this strategy both agents manage to learn the optimal policy in fewer episodes. In scenario (a) and (b) we can observe how our agent learns in fewer episodes while managing to fully discover the causal models for all actions (see subplots (d) and (e)). In scenario (c) where exploration decreases more slowly, our agent manages to learn the models faster, however, after 600 episodes we still cannot see convergence in the rewards.

4.1 Discussion

As we can see from the results, our method performs better using a decreasing exploration strategy. In this scenario it manages to learn the optimal policy for the task in fewer episodes than a traditional model-free agent and also learns correctly the causal models for each of the actions. The learned models could be transferred to other tasks that share state variables and rewards.

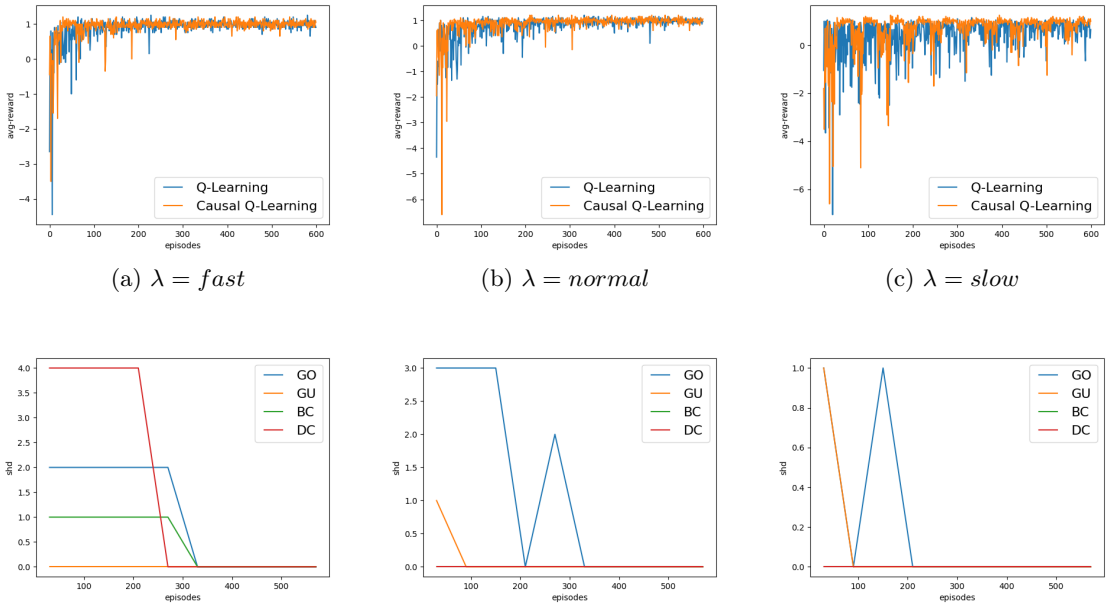
It is important to mention that in these experiments we used predefined values of (T) (interval to learn the causal model) and (th) (causal threshold) based on the initial estimate of the total number of episodes. For a more realistic application, these parameters should be estimated at run time, since the agent cannot know *a priori* how many episodes it will take to learn the task, nor what are the correct causal models it has to discover. With respect to the T parameter, we can use the cardinality of the variables of the corresponding causal model and the amount of observations, so that when the first discovery is made, there is enough data available to infer a (partial) model. With respect to the threshold t, we can



(a) $\epsilon = 0.9$ (b) $\epsilon = 0.5$ (c) $\epsilon = 0.1$

(d) $\epsilon = 0.9$ (e) $\epsilon = 0.5$ (f) $\epsilon = 0.1$

Figure 2: Q-Learning vs Causal Q-Learning using a fixed exploration strategy at different levels.



(a) $\lambda = fast$ (b) $\lambda = normal$ (c) $\lambda = slow$

(d) $\lambda = fast$ (e) $\lambda = normal$ (f) $\lambda = slow$

Figure 3: Q-Learning vs Causal Q-Learning using a decayed exploration strategy at different levels.

perform simulations with the learned model and observe the results to know if both the transition and the reward are correct, and based on that determine if it is convenient to be guided by the model for the selection of actions.

Also critical is the size of the agent's action space, since the number of causal models to be discovered is equal to the total number of actions. To mitigate this limitation, a relational representation of the actions could be used, which could drastically reduce the number of actions. We are aware that experiments in more complex environments are required, with sparse rewards, more actions, and more state variables, in order to make our proposal scalable. Nevertheless, we consider that the results obtained constitute a good starting point for future experiments.

5. Conclusions

The combination of Causal Discovery and Reinforcement learning is an exciting emergent field. A method for simultaneously learning a policy with reinforcement learning and learning and using causal models to accelerate the learning process was presented, in which the agent alternately explores the environment to learn a policy and uses the observations to discover the underlying causal models to select actions. The method was tested in different experimental settings. Based on the results we can conclude that: (i) There is a trade-off between optimal policy learning and causal discovery. High exploration favors causal discovery but hampers reinforcement learning. (ii) The underlying causal models can be

discovered using a Causal Dynamic Bayesian Network for each agent action and a score-based causal discovery algorithm with good results. (iii) Learning (partially) correct causal models can be used to improve reinforcement learning. (iv) Using a decreased exploration strategy our method was able to converge to an optimal police in less episodes than a traditional model free algorithm, while correctly discovering the causal models.

The results indicate that the presented methodology is a suitable alternative for solving tasks where the environment is governed by a causal model. Using those models for action selection can reduce the learning time with respect to a trial-and-error interaction with the environment. The proposed approach was implemented over a commonly used model free RL algorithm but it can be easily transfer to other algorithms. It is left for future work to determine the T and th parameters at run-time, and also to perform tests on more complex scenarios.

Acknowledgments

This work was partially supported by CONACYT, Project A1-S-43346 and scholarship 754972 (first author).

References

I. Akkaya, Andrychowicz, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

C. Boutilier, R. Dearden, M. Goldszmidt, et al. Exploiting structure in policy construction. In *IJCAI*, volume 14, pages 1104–1113, 1995.

K. Chalupka, P. Perona, and F. Eberhardt. Visual causal feature learning. In *Proceedings of the 31th Conference on Uncertainty in Artificial Intelligence*, page 181–190. AUAI Press, 2015. ISBN 9780996643108.

M. Deisenroth and C. E. Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472. Citeseer, 2011.

R. J. Dolan and P. Dayan. Goals and habits in the brain. *Neuron*, 80(2):312–325, 2013.

I. Feliciano-Avelino, A. Méndez-Molina, E. F. Morales, and L. E. Sucar. Causal based action selection policy for reinforcement learning. In *Mexican International Conference on Artificial Intelligence*, pages 213–227. Springer, 2021.

M. Gonzalez-Soto, L. E. Sucar, and H. J. Escalante. Playing against nature: causal discovery for decision making under uncertainty. *arXiv:1807.01268*, 2018.

O. Gottesman, Johansson, et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018.

K. Kansky, T. Silver, and et al. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In *Proc of the 34th Int Conf on ML*, pages 1809–1818, 2017.

F. Lattimore, T. Lattimore, and M. D. Reid. Causal bandits: Learning good interventions via causal inference. In *Advances in NIPS*, pages 1181–1189, 2016.

S. Lee and E. Bareinboim. Structural causal bandits: where to intervene? In *Advances in Neural Information Processing Systems*, pages 2568–2578, 2018.

S. Lee and E. Bareinboim. Structural causal bandits with non-manipulable variables. In *Proceedings of the AAAI Conference*, volume 33, pages 4164–4172, 2019.

P. Madumal, T. Miller, L. Sonenberg, and F. Vetere. Explainable reinforcement learning through a causal lens. *arXiv preprint arXiv:1905.10958*, 2019.

A. M. Molina, I. F. Avelino, E. F. Morales, and L. E. Sucar. Causal based q-learning. *Research in Computing Science*, 149:95–104, 2020.

S. Nair, Y. Zhu, S. Savarese, and L. Fei-Fei. Causal induction from visual observations for goal directed tasks. *arXiv preprint arXiv:1910.01751*, 2019.

J. Pearl. *Causality*. Cambridge university press, 2009.

M. Scutari. Learning bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03.

D. Silver, Schrittwieser, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

O. Vinyals, Babuschkin, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019.

C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

S. Zhu and Z. Chen. Causal discovery with reinforcement learning. *CoRR*, abs/1906.04477, 2019a. URL `http://arxiv.org/abs/1906.04477`.

S. Zhu and Z. Chen. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019b.