Novelty Detection and Adaptation:
A Domain Agnostic Approach

Marina Haliem[a], Vaneet Aggarwal[a] and Bharat Bhargava[a]

[a] *Purdue University, West Lafayette, IN, USA*

Abstract
Novelties are surprises that a system encounters. System must learn about the characteristics, detect, understand, and adapt to novelty in not only the environment but in agents that interact with it. The context, timing, duration, extent, duration of novelty must be considered in agent's adaption and accommodation. This research contributes towards building AI/ML system that can adapt to fluid novelties in open world. Many real-world problems are stochastic and encounter sudden novelties, which results in highly dynamic environments. Therefore, a robust framework is needed to identify various novelties that can occur, recognize the changes in the underlying environment, and adapt policies to maximize the long-term cumulative rewards. To achieve this, we propose ideas to adopt a change point detection algorithm to detect the changes in the distribution of experiences, and to develop an agent that is capable of recognizing novelties and making informed decisions according to the changes in the underlying environment. These ideas can be adapted in various domains by tuning the agent's objective function, where it will still capture the changes in the corresponding underlying environment. This research contributes to the SAIL-ON effort [Ted Senator 2019].

Keywords 1
Novelties, Novelty Generation, Decision-Making, Change Point Detection, Dirichlet Processes

1. Introduction

Novelties occur in many systems and environments and agents must learn about them and accommodate them. We list a few examples to provide understanding and a basis for research ideas.

- A car going on a steep hill in dark, rain. The car is not on main road. Main road on flat terrain, good weather will be normal. Steep hill, darkness, rain/snow and road with weak soil, vegetation will be novelties.
- A person from USA driving in India. Many novelties occur: no stop and yield signs, left hand drive, mix of traffic vehicles (bicycle, rickshaw, horse/bullock/oxen driven carts, scooters, three-wheeler, along with trucks, buses) and narrow single lane roads and unpaved roads. How can the drivers (or driver) visiting from USA train, learn, and adjust to drive safely in India.
- A cheating or sudden change in the rules of playing games such as chess, basketball, monopoly while game is being played. In addition, a novelty may be that the objective of a player changes: instead of winning or losing, objective becomes a tie decision
- An attack, malicious activities and threats cyber or otherwise. How can a child or older person deal with novelties of pickpockets, scoundrels, thieves, purse snatchers, etc. The objective may become a survival of a person. How can a system continue to

operate in unknown adverse conditions and situations such as collaborative attacks in cyberspace?

- A man walking a cat or rhinoceros or hippopotamus (walking a dog, elephant or horse is not a novelty)

We consider the scenario of dynamic environments, where a novelty occurs that alters the dynamics of a system and its model and transforms itself to incorporate the novelty. The environment changes between the pre-novelty model and the post-novelty model dynamically as shown in Figure 1 below.
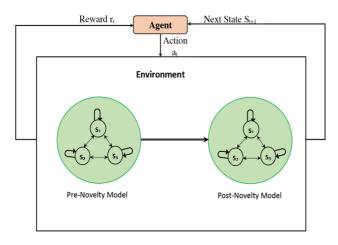

Figure 1: Dynamic Environment

The implication of the non-stationary environment *[Kaplanis et al., 2019]* is as follows. When the agent exercises a control at at time t, the next state S_{t+1} as well as the reward r_t are functions of the active environment model dynamics. We assume the knowledge that the environment switches from a pre-novelty model to a post-novelty model due to an unexpected change in the world state. However, neither the context information of each model nor the change points when the change occurs, are known to the agent. In the Open World, environments are characterized by their high dynamicity where novelties can occur and alter the representation of the world ω, and thus the state space S and the action space A. We assume the environment is partially observable by our agent, so the agent has the knowledge of the state representation $S_t \in \omega$, that is part of the surrounding world ω. Different types of

novelties impose different levels of difficulty when it comes to the ability to detect and adapt to these changes as well as the time consumed until detection and adaptation. We investigate the different types of novelties, and discuss approaches that allow the agent to detect and adapt to these changes.

2. Decision-making with replay memory

In decision-making, the task is for the agent, at each time step t, to select an action $a_t \in A(S_t)$ based on the current state of the environment $S_t \in \omega$ where S_t is the state representation of the environment that is observable by our agent, and $A(S_t)$ is the finite set of possible actions in state S_t. The agent selects the action that maximizes its objective function at each time step. After an action is executed, the agent receives a reward r_t, and state of the environment is updated to S_{t+1}. The transitions of the form (S_t, a_t, r_t, S_{t+1}) are stored in a cyclic buffer, known as the *"replay buffer" [Lin, 1992]*. This buffer enables the agent to randomly sample from and train on prior observations.

3. Novelty Types

The main components that affect the decision-making process of an agent are the State Space, Action Space and the transition probabilities that it learns in order to reach the optimal policy. Novelties can be categorized to deal with ethe following:

1. **State space changes:** novelties that alter the environment representation will require the state space to be expanded so it can accommodate these changes. For example, a new state in the environment that is different from every state in the agent's experience memory.

2. **Action space changes:** novelties in the dynamic interactions or context may lead to a different action space, that will be modified and fed back to the agent.

3. **No state/action set change, transition probability changes:** novelties that

change the set of rules that govern the environment dynamics, for instance: getting a 6 on dice is giving additional turn rather than stopping there, or getting a 1 on the dice moves 3 steps rather than 1.

4. **No state/action set change, Reward function changes:** Goal-related novelties will require a re-design of the reward function to reflect the new objectives of the system. For example, forcing a draw in a game is equivalent to a win.

4. Model-Free Detection and Adaptation

Some novel events that occur in the environment may lead to combinations of the types of novelties mentioned above, for example: in a game of Monopoly - adding credit score to player profiles, impacting ability to set up real estate on properties or adding a risk attribute to hotels and properties affecting rent collecting potential. To handle such dynamicity, a model-free approach is proposed where the agent learns the dynamics of the environment through real-time interactions with the environment rather than having a rigid pre-defined model fed to the agent. According to the consequences of the occurrence of a novelty, the agent might take longer/shorter time to detect the change and adjust to it which result in higher/lower difficulty level. To achieve this goal, the agent collects experience tuples while simultaneously following a model-free learning algorithm to learn an approximately optimal policy. Instead of assuming any specific structure, the model-free approach allows the agent to dynamically learn the change. Two of the approaches that the agent can follow are:

- **Environment Observation:** For novelties that lead to 1 and 2 types as discussed earlier, a modification is needed in the state and action spaces. This can be done online as the agent detects the new states/actions; but might cause a delay in adaption until

the agent learns the new space. The change can be only from the agent's perspective, not a universal change in the world state ω, where at time step t, the agent gets a representation of the environment that is identified as to be unknown. This can be caused by a new state in the environment that is different from every state in the agent's experience memory, or a change in the set of rules that affect the set of allowable actions/moves that the agent can take.

- **Change Point Detection:** For novelties that lead to types 3 and 4, the proposed method works in tandem with a change point detection algorithm, to get information about the changes in the environment *[Haliem et al., 2020]*. The learning begins by obtaining experience tuples E_t according to the dynamics and reward function of current active model. The state and reward obtained are stored as experience tuples, since model information is not known. The samples can be analysed for context changes in batch mode or online mode. We adopt the online parametric Dirichlet changepoint (ODCP) detection algorithm proposed in *[Singh et al., 2019]* to examine the data consisting of experience tuples. This algorithm transforms any discrete or continuous data into compositional data and utilizes Dirichlet parameter likelihood testing to detect change points. Although ODCP requires the multivariate data to be i.i.d samples from a distribution. The justification in *[Padakandla et al., 2019]* explains the utilization of ODCP in the Markovian setting, where the data obtained does not consist of independent samples. The full algorithm for the Dirichlet change point detection algorithm is shown in Algorithm 1 below, where the input is the data consisting of experience tuples E_t that are stored by our agent. In this algorithm, the maximum likelihood estimation of Dirichlet distribution parameters is

calculated for the cumulative data stored through experience tuples using equation 1 below:

$$\alpha_i^* = argmax_\alpha \log \Gamma \left(\sum_l \alpha_l \right)$$

$$- \sum_l \log \Gamma (\alpha_l)$$

$$+ \sum_l ((\alpha_l - 1)(\log \hat{x}_l)),$$

$$where \; \hat{x}_l$$

$$= \frac{1}{T} \sum_i \log(x_{i_l}) \qquad [1]$$

Then, the log likelihood given distribution Q_0 is calculated using equation 2 below:

$$LL(x_l, ..., x_T, Q) = \left(\sum_{i=1}^{T} \log(Q(x_i)) \right)$$

$$Q(x_i) = \frac{1}{B(\alpha)} \prod_{l=1}^{d} x_{i_l}^{\alpha_l - 1},$$

$$B(\alpha) = \frac{\prod_{l=1}^{d} \Gamma(\alpha_l)}{\Gamma(\sum_{l=1}^{d} \alpha_l)},$$

$$where \; d = |x_i| \;\; Dimensioality \; of \; x_i, x_l$$

$$\geq 0, and \; \sum_{l=1}^{d} = 1 \qquad [2]$$

Then, at each time step t, that is seen as a potential change point, we split the data into two parts (prior and after this time step t), and we estimate the maximum likelihood as well as the sum of log likelihood for both partitions using the equations above. Finally, the algorithm returns the point in time T^* associated with the maximum log likelihood to be the potential change point. If the difference between this value and the log likelihood of our unsplit original data turns out to be greater than our threshold, then we declare that a change has been detected at time T^*.

After the agent detects that a change has occurred, it restarts the decision-making process accounting for that change. At every time step t, it obtains a representation for the environment, S_t and calculates a reward r_t associated with each possible action in the action space according to the dynamics and reward function of current active model (whether it is the pre/post novelty model).

Based on this information, the agent takes an action where the expected discounted future reward is maximized. One approach for dealing with novelty can be seen in *[Haliem et al., 2020 - 2]* where it is applied to a multi-agent ride-sharing system.

Algorithm 1 Dirichlet Change Point Detection Algorithm

1: **Input** Time Window $[1 \ldots T]$, Data $[x_1 \ldots x_T]$.
2: **Output** T^*: Change Point (if there is a change).
3: **procedure** DCP($[x_1 \ldots x_T]$)
4:     $Q_0 \leftarrow$ **Estimate** Drichlet Parameters for $[x_1 \ldots x_T]$ using Eq. 1
5:     $LL_0 \leftarrow$ **Estimate** Log-Likelihood for $[x_1...x_T]$ under Q_0 - Eq. 2
6:     $(T^*, LL^*) \leftarrow$ ESTIMATE_2WINDOW($[x_1 \ldots x_T]$)
7:     $Z^* \leftarrow LL^* - LL_0$
8:     **if** $Z^* >$ threshold **then**
9:         **Return** Change point at T^*.
10:     **else**
11:         No change, **Return**
12: **procedure** ESTIMATE_2WINDOW($[x_1 \ldots x_T]$)
13:     **for** $t \in 1 \ldots T-1$ **do**
14:         $Q_1 \leftarrow$ **Estimate** Drichlet Parameters for $[x_1 \ldots x_t]$ - Eq. 1
15:         $Q_2 \leftarrow$ **Estimate** Drichlet Parameters for $[x_{t+1}...x_T]$ - Eq. 1
16:         $LL_t \leftarrow$ Log-Likelihood for $[x_1 \ldots x_t]$ under Q_1 using Eq. 2 + Log-Likelihood for $[x_{t+1} \ldots x_T]$ under Q_2 using Eq. 2
17:     $LL^* \leftarrow max_{(t \in 1...T-1)} LL(t)$
18:     $T^* \leftarrow argmax_{(t \in 1...T-1)} LL(t)$
19:     **Return** (T^*, LL^*)

---

5. Conclusion

In this paper, we established a theory of novelties occurring in an Open World setting. In our method, we propose utilizing a Change Point Detection algorithm in addition to environment observation to allow the agent to detect the change that occurred in the underlying environment and recognizes what types of novelties took place. This is a crucial step that will then allow the agent to adjust accordingly and start learning and performing well in this modified environment. Our approach can be tailored to fit various domains by designing the objective function of the agent to reflect the specific goals of this domain. For example, such setting could be utilized in a Ride-Sharing system as in *[Haliem et al., 2020 - 2]* by utilizing the agent's reward function proposed by the authors. In [Boult et al., 2020], authors are developing a unifying framework for formal theories of novelty and more information about the thought process to understand novelties is available in their paper

in AAAI-2021. Terry is a speaker in the workshop on Novelties in Open World during the ISIC conference organized by Prof. Bharat Bhargava on Feb 25, 2021.

There are many universities and organizations working on the characterization of novelties and languages to express them and novelties hierarchy and evaluation in the SAIL-ON effort. Some of these ideas will be discussed in the workshop.

6. Acknowledgements

7. References

[Ted Senator, 2019] https://www.darpa.mil/program/science-of-artificial-intelligence-and-learning-for-open-world-novelty

[Kaplanis et al., 2019] Christos Kaplanis, Murray Shanahan, and Claudia C. 2019. *Policy consolidation for continual reinforcement learning.* arXiv preprint arXiv:1902.00255 (2019).

[Mnih et al., 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller. *Human-level control through deep reinforcement learning.* nature, 518(7540):529–533.

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

[Haliem et al., 2020 - 1] Marina Haliem, Ganapathy Mani, Vaneet Aggarwal, and Bharat Bhargava. 2020. *A Distributed Model-Free Ride-Sharing Algorithm with Pricing using Deep Reinforcement Learning.* Computer Science in Cars Symposium. Association for Computing Machinery, New York, NY, USA, Article 5, 1–10. DOI:https://doi.org/10.1145/3385958.3430484

[Haliem et al., 2020 - 2] Marina Haliem, Vaneet Aggarwal, and Bharat Bhargava. 2020. *AdaPool: An Adaptive Model-Free Ride-Sharing Approach for Dispatching using Deep Reinforcement Learning.* Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys 2020). Association for Computing Machinery (ACM), 304–305.

[Padakandla et al., 2019] Sindhu Padakandla, Shalabh Bhatnagar, et al. 2019. *Reinforcement learning in non-stationary environments.* arXiv preprint arXiv:1905.03970 (2019).

[Singh et al., 2019] Nitin Singh, Pankaj Dayama, Vinayaka Pandit, et al. 2019. *Change Point Detection for Compositional Multivariate Data.* arXiv preprint arXiv:1901.04935 (2019).

[Boult et al., 2020] T. E. Boult, P. A. Grabowicz, D. S. Prijatelj, R. Stern, L. Holder, J. Alspector, M. Jafarzadeh, T. Ahmad, A. R. Dhamija, C. Li, S. Cruz, A. Shrivastava, C. Vondrick, W. J. Scheirer , Towards a Unifying Framework for Formal Theories of Novelty, in Proceedings of The Thirty-Fifth *AAAI* Conference on Artificial Intelligence (*AAAI*-21) February 2-9, *2021*.