Adaptive Reinforcement Learning Method for Sequential Decision Task: A Review

Pramod Patil[1], Ankur Verma[2]

[1]Professor, Department of Computer Engineering DYPIET Pimpri, SavitriBai Phule Pune University, India

[2]Department of Computer Engineering DYPIET Pimpri, SavitriBai Phule Pune University, India

Abstract: ***There are many dynamic situations in which sequential actions come with circumstances favorable. These consequences of actions can include at a multitude of times after the action is taken, and it shall be concern with the strategies for specify action on the basis of both their short term and long term consequences. A proposed model based approach which requires constructing the model of state transaction and payoff probabilities. Task of such kind can be termed as a dynamical system whose behavior changes over time under the impact of a decision maker's action. This modeling of the behavior of the system is greatly simplified by the concept of state. Decision policy associates on action with each system states. There is a great practical importance of adaptive method, if this adaptive method can make improvement in decision policy sufficiently rapidly may be less. It proposes methods for estimating optimal policy in the absence of a complete model of the decision tasks which are known as adaptive or decision model.***

Keywords: *Reinforcement Learning, Decision policy, state-action function, Q-Learning, Temporal Difference Learning.*

1. Introduction

Computer scientists are interested in developing devices and programs based on the life science by studding its engineering. These studies are based on "synthetic learning" which has produced various methods and mathematical theories for pattern classification, prediction and adaptive control of dynamic systems. The problem still arises in synthetic learning is the nature of correspondence lies between the behavior of a system in classical conditioning experiment and mathematical theories and the computational procedures. Justification to these problems observed that the classical conditioning experiments are far from computationally trivial. These computational methods are also useful for adaptive prediction by making use of "Temporal Difference model of conditioning".

Earlier dynamic programming was used in behavioral ecology for calculating information pertinent to decision making at each stage on the basis of information previously calculated from that stage to the task's end but that worked backward. Than the TD procedure is overcoming the dynamic programming by accomplishing the same results by repeated forward passes through a decision tasks instead of back-to-front computation used by dynamic programming.

This TD model is defined under the reinforcement learning which defines that for achieving the goal the interaction with the problem of learning is framed to be straight forward. The "learner" and "decision-maker" are called the agent. The environment is the thing through which agent interacts from the outside. Through these interactions the agent continuously selects actions and environment replying to those actions and providing new behaviors providing the agent. The rewards garneted by the environment also tried to maximize over time by the agent using a special signal. A task is defined by a full specification of an environment, with single instance of the reinforcement learning problem.

The tasks in which the consequences of an action can emerge at a multitude of times after the action is taken are only considered and such strategies for selecting an action on the basis of both there short-term and long-term consequences are interested. Some of the situations in which system has to make sequence of actions to bring about circumstances favorable for their survival can be formulated in terms of the dynamical system. The behavior of such systems unfolds over time under the influence of decision makers actions.

Modeling the behavior of such system is greatly simplified by the concept of static. The state of a system at a particular time is a description of the condition of the system at that time that it is sufficient to determine all aspects of the future behavior of the system when combined with knowledge of the systems future input. Describing a decision task in terms of system states permits one to make a relatively simple statement of how action and state sequence determines the total amount of pay off an agent receives.

2. Literature Survey

Esther Levin [1] et al. proposed a quantitative model for learning the dialog strategy. They optimized the problem of dialog design by an objective function reflecting over different dialog dimensions required for a given application. They also shown that by state space, action set and strategy a dialog system can described as a sequential decision process. Additionally they described that a dialog system can relate with stochastic model known as Markov decision process (MDP). They also produced a combination of supervised and reinforcement learning for effective use of training data available. Which they tested for learning in an air travel information system (ATIS). The experimental results they presented in there paper show a state space representation, a simple criterion, and a simulated user parameterization in order to learn automatically a complex dialog behavior.

Larry D. Pyeatt [2] et al. presented an approach based on decision tree for the approximation of a function in the reinforcement learning. That is useful in scaling reinforcement learning problems with large number of states and actions. Also they compared the decision tree providing better learning performance over neural network function approximation and solving large infeasible problems using table lookup. These comparisons were held on the mountain car, pole balance problems and a simulated automobile race car.

Mircea Preda [3] et al. proposed a new method constructing decision trees of using reinforcement learning. This method is much efficient that it creates more and more decision trees because it learn this from the training set which constraint is to be tested first in order to classify a subset of examples. Through this the new method is suitable for solving problems where training set changes frequently and classification rules also changes over time. This method is also helpful where various constraints have various testing costs. Performance results and the summary of the features of the implemented algorithm are also concluded.

Lucian Busoniu [4] et al. proposed an approximate, model based Q-iteration algorithm relying on a fuzzy partition state space and discretization of action space. Using their assumptions on continuity of the dynamics and reward function, they shown an consistent algorithm, i.e., that as the approximation accuracy increases the optimal solution is obtained asymptotically. There experimental study indicated that “a continuous reward function is also important” for a predictable improvement for performance when there is increase in approximation accuracy.

Ioannis Partalas [5] et al. studied the pruning problem of an ensemble classifier using reinforcement learning. This contributed a new approach of pruning which takes the use of Q-learning algorithm for approximating an optimal policy for including and excluding all classifiers from ensemble. Comparisons made between the approaches of state-of-the-art pruning and the combination methods shows good results. Also an extension providing improvement over time is presented for certain performance critical-domains.

Mill´an-Giraldo [6] et al. made decisions regarding processing of incomplete and unlabeled incoming objects and also guessing the missing attributes value. These decisions are solved by including them in the training set and by asking regarding the class label at the relevant time. This was not possible in earlier for the complete set of attributes due to the dependency of the attributes. This made possible by the empirical results produced by the better framework of isolatedly. This framework uses the most of human effort and computer effort together by involving them in a behavioral manner.

Prof. Pramod Patil [8] et al. they implement the reinforcement learning algorithm on data streams of diabetes. There algorithm works the data streams and differentiate the values for blood glucose level and for insulin dose and takes the decision for next insulin dose. Depending on taken state and action the payoff is assign to the decision. That helped in classifying the data for doses of diabetes and also helped in making decisions at a particular time for giving specific quantity of dose. In comparison with other methods their proposed algorithm was faster. They also tested the proposed methodology on diabetes data set.

Stefanos Doltsinis Pyeatt [9] et al. through this paper approached ramp-up as a sequential adjustment and tuning process to a desirable performance manufacturing system in the fastest possible time. They focused on developing a Markov decision process (MDP) model for ramp-up of production stations and enabling its analysis. Their aim is also to capture the cause and effect of the station’s response to improve the effectiveness of the process and operator adaptation or adjustment of a station. They investigated the Q-batch learning algorithm’s application combined with an MDP model. There approach was applied to a station of highly automated production where sundry ramp-up processes are carried out. They also learned policy that are applied and compared against previous ramp-up cases.

Jihye Bae [10] et al. introduced a temporal difference algorithm for estimating a value function in reinforcement learning. Correntropy is a robust cost function used by a kernel adaptive system. To find a proper policy they integrated the algorithm with Q-learning. They tested the proposed method with synthetic problems and quantified its robustness. They also applied the same algorithm over a monkey’s neural states in reinforcement learning brain machine interface. The results they observed were potentially beneficial.

3. Reinforcement Learning

Reinforcement learning (RL) has become most active area of research now days. The objective of reinforcement learning is to getting maximizing rewards by mapping the state with the actions. In RL an autonomous agent follows a trial and error process in order to reach its goal by learning optimal action to perform in each state. The flow diagram of the RL is shown.

Methodology:
Step 1. The agent chooses an action on each state.
Step 2. This action may take the agent in new state.
Step 3. In new state the agent will get reward.
Step 4. Repeat the steps (1, 2, 3).

Agent eventually learns the action to obtain maximum reward.

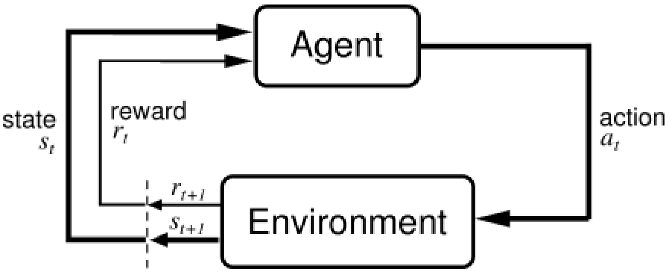


Figure 1: Reinforcement Learning Process

Nomenclature Section

No.	*Symbol*	*Meaning*
1	St	State of agent at time t.
2	a_t	Action taken by agent at time t.
3	π	Policy taken by agent
4	Q	Action value function
5	ε	Exploration chance
6	α	Learning Rate
7	γ	Discount factor
8	τ	User supplied threshold

Elements of Reinforcement learning:

A. *Policy:* A policy is the core of reinforcement agent in the way that it is alone to determine the behavior of the agent. The policy only decides the action which is to be taken by the agent by the previous states.

B. *Reward* Function: Reward is defining the goal to the agent. Reward maps the action of previous state with a single number. The main goal of the agent to maximize the total reward over long run. The reward only decides that weather the policy is good or bad in order to maximize reward.

C. *Value Function:* Value function describes which policy is good for long run. Value of a state defines the amount of total reward an agent can look for to assemble over the future, initial from the state.

D. *Model of the environment:* This defines the behavior of the environment. That is for a given state the agent may predict the next state and the reward for the next state.

4. Temporal Difference Learning

TD learning is the combination of Monte Carlo and dynamic programming. TD methods are capable of learning directly from the raw experience neither requiring the environment dynamics model. TD estimation algorithm used in reinforcement learning is capable of predicting a measure of total amount of expected reward as well as the other quantities over the future.

Simple every - visit Monte Carlo method:

$$V(s_t) \leftarrow V(s_t) + \alpha\,[\underset{\downarrow}{R_t} - V(s_t)]$$

Target: the actual return after time *t*

The simplest TD method, TD(0) :

$$V(s_t) \leftarrow V(s_t) + \alpha\,[\underbrace{r_{t+1} + \gamma V(s_{t+1})}_{\downarrow} - V(s_t)]$$

Target: an estimate of the return

ON and OFF policy of Temporal Difference:

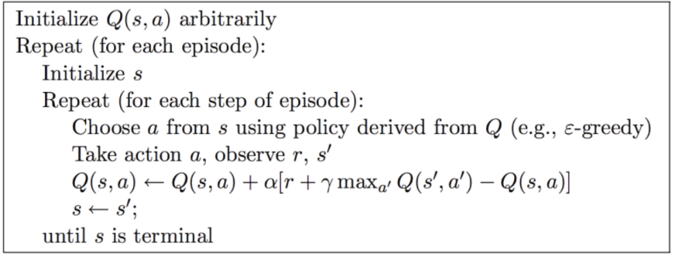
A. Sarsa: On-Policy TD Control*:*

Sarsa concentrates in the limiting to the greedy policy. Also concentrates on the probability of an optimal policy and action-value function until all state-action pairs are traversed infinitely as shown in algorithm.

B. Q-Learning: Off-Policy TD Control:

Q-learning is a model which is free from reinforcement learning technique. It uses an action-value function for learning that ultimately gives the look for an optimal policy for actions followed by a given state-action and also that given action.

Q-Learning: Off-Policy TD Control algorithm



```
Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
   Initialize $s$
   Repeat (for each step of episode):
      Choose $a$ from $s$ using policy derived from $Q$ (e.g., $\varepsilon$-greedy)
      Take action $a$, observe $r$, $s'$
      $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
      $s \leftarrow s'$;
   until $s$ is terminal
```

5. Dynamic Programming

“Dynamic Programming" (DP) refers to a grouping of algorithms that can be used to compute optimal policies given for a perfect model of the environment in a Markov decision process (MDP). The key idea of DP is to use the value functions for organizing and structuring the search for good policies. Classical DP algorithms are having limited utility in the reinforcement learning because of their assumption of a perfect model and their great computational expense, but they still are very important in the perspective of theory. DP makes available an essential foundation for understanding the methods. Although, all of these methods can be viewed as attempts to achieve much the same effect as DP, only with the less computation and also without assuming a perfect model of the environment.

6. Conclusion

This paper presents a survey on the system that will keep track of unexpected power scenarios using an agent and actions taken regarding it. And according to the outcome the actions will be rewarded or will receive penalty according to the action. The system will further be helpful for taking independent decisions for existing scenarios; although the system also intimates at the end with optimal action for the scenario. It will be the decision of the user to implement provided action, because there may always be exceptions or priorities for the user that may be affected due to the provided action.

7. Acknowledgement

I would like to take this opportunity to acknowledge the contribution of certain people without which it would not have been possible to complete this paper work. I would like to express my special appreciation and thanks to my advisor Professor Pramod Patil, you have been a overwhelming mentor for me.

References

[1] Esther Levin, Roberto Pieraccini and Wieland Eckert “A stochastic model of human-machine interaction for learning dialog strategies” *IEEE TRANSACTIONS ON*

SPEECH AND AUDIO PROCESSING, VOL. 8, NO. 1, JANUARY 2000

[2] Larry D. Pyeatt and Adele E. Howe "Decision tree function approximation in reinforcement learning" *CiteSeer* , 07/2001

[3] Mircea Preda "Adaptive building of decision trees by reinforcement learning" *Stevens Point, Wisconsin, USA* ©2007 , ISBN: 978-960-6766-01-5

[4] Lucian Busoniu, Damien Ernst, Bart De Schutter, and Robert Babuska "Consistency of fuzzy model-based reinforcement learning" 2008 *IEEE International Conference on Fuzzy Systems.*

[5] Ioannis Partalas, Grigorios Tsoumakas and Ioannis Vlahavas "Pruning an ensemble of classifiers via reinforcement learning" *Neurocomputing - IJON* , vol. 72, no. 7-9, pp. 1900-1909, 2009

[6] M. Mill´an-Giraldo, V. Javier Traver, and J. Salvador S´anchez "On-line classification of data streams with missing values based on reinforcement learning" *Springer-Verlag Berlin, Heidelberg* ©2011 , ISBN: 978-3-642-21256-7

[7] Sander Adam, Lucian Busoniu, and Robert Babuska "Experience replay for real-time reinforcement learning control" *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS — PART C: APPLICATIONS AND REVIEWS*, VOL. 42, NO. 2, MARCH 2012

[8] Prof. Pramod Patil, Dr. Parag Kulkarni, and Ms. Raczhana Shirsath "Learning and sequential decision making for medical data streams using rl algorithm" *International Journal of Research in Computer and Communication Technology*, Vol. 2, Issue 7, July-2013

[9] Stefanos Doltsinis, Pedro Ferreira, and Niels Lohse "An MDP model-based reinforcement learning approach for production station ramp-up optimization: q-learning analysis" *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS*, VOL. 44, NO. 9, SEPTEMBER 2014

[10] Jihye Bae, Luis G. Sanchez Giraldo, Jose C. Principe, Joseph T. Francis "Correntropy kernel temporal differences for reinforcement learning brain machine interfaces" 2014 *International Joint Conference on Neural Networks (IJCNN)* July 6-11, 2014, Beijing, China

Author Profile

Pramod D.Patil obtained his Bachelor's degree in Computer Science and Engineering from Swami Ramanand Tirth Marathwada University, India. Then he obtained his Master's degree in Computer Engineering and pursuing PhD in Computer Engineering majoring in Mining Data Streams both from Pune University, INDIA. Currently, he is a Research Scholar in Department of Computer Engineering at COEP, Pune University, INDIA. His specializations include Database Management System, Data Mining, and Web Mining. His current research interests are Mining Data Streams.

Ankur Verma obtained his Bachelor's degree in Computer Science from Rajasthan Technical University, India. Now pursuing Master's degree in Computer Engineering Pune University, India. His dissertations work on Data Stream Mining. [2]Ankur Verma, PG student, Computer department, DYPIET, Pune