Nested Reinforcement Learning for Leukemia

Mr. J. Joseph Ignatious [1], Ms. V.Lawanya [2], Ms. N.Kirubashankari [3], Ms. G.Saranya [4]

[1] *(ECE Department, V.R.S College of Engineering and Technology, India)*
[2] *(ECE Department, V.R.S College of Engineering and Technology, India)*
[3] *(ECE Department, V.R.S College of Engineering and Technology, India)*
[4] *(ECE Department, V.R.S College of Engineering and Technology, India)*

ABSTRACT : Reinforcement Learning with Particle Filter (RLPF) is simulation-based techniques useful in solving Markov decision processes if their transition probabilities are not easily accessible if it's have a very large number of states. The main idea of RLPF is to use particle filtering as a method for choosing the sampling points, for calculating a parameter vector for each trial, the impact of step sizes when function approximation is combined with RLPF. This method used to detect leukemia from the number of White Blood Cell (WBC) in microscopic images to help clinician for diagnosis of leukemia and blood related disease. The RLPF method have implemented by using image enhancement and segmentation method. The analysis such as grayscale conversion, image sharpening, contrast adjustment, and morphological operation, proved that leukemia can be detected by fine tune of parameter with accuracy. Therefore, image enhancement and segmentation technique to help clinician in diagnosis of leukemia and other blood related diseases.

KEYWORDS - Particle filter, Q-Learning, Reinforcement Learning (RL), Semi-Markov Decision Process (SMDP), Short Stochastic Path (SSP).

I. INTRODUCTION

Reinforcement Learning (RL) is propagation construct methodology that is significant in light of far reaching scale and complex Markov Decision Process (MDP) [2]. In this paper, they addressed the some portion of step sizes (learning rules) in diminished prize issues and that of the building up segment of the most Short Stochastic Path (SSP) in ordinary prize issues and the thought of showing survival probability (downside peril) inside RL[1]. They looked at the impact of these components on the characteristics of rehashes and investigate by the measure of characteristics can meander from the qualities gained from component programming. In the setting of step sizes, they performed a study using some standard fundamentals to choose how they perform. For typical prize issues, the SSP-building up framework grants them to differentiate the estimations of the rehashes with those gotten from an identical worth accentuation count; here, on the other hand, diverse step-sizes are required and an exploratory concentrate necessities to consider that. In like manner make and test a NRL figuring that models the survival-probability of a system. Typically, the survival probability is portrayed with gratefulness to a known target wage. The survival probability of a system is the probability that the wage in unit the reality of the situation will become obvious eventually the target. It is direct related to the downside risk in operations research and the exceedance probability in the assurance business [3]. Our proposed one is Reinforcement Learning with Particle Filter (NRL) for survival probabilities, and after that numerically exhibit that the rehashes in the related NRL computation converge to a perfect game plan. It is our conviction that our results will be valuable to an utilizing so as to practice agent motivated NRL. The straggling left over of this article is sorted out as takes after. Section 1 demonstrates a discussion of our examinations with set apart down prize, and it gives our results the SSP-building up framework all things considered prize issues, and Section 2 looks at our computation with the survival probability thoughts. Region 3 shuts this paper.

1. Discounted Reward

The effect of the rate of union of direct and polynomial step sizes on the qualities to which Q-values in NRL merge has been concentrated on in [4]. They have built up hypothetically that direct standards (e.,g,1/k, where k indicates the emphasis number) can take an exponential time to join while polynomial guidelines (e.g.,1/k where is some whole number) can unite in polynomial time. In this paper, our objective is less eager and wishes to direct explores different avenues regarding some basic step sizes to test how they perform experimentally and how far they stray from the ideal qualities in an observational setting. It is surely understood that, by and by, one needs to calibrate the execution of a NRL calculation through trials of various step sizes, and our trustiness is that it will be advantageous to a pragmatic client of these calculations to know how a portion of the understood principles perform under known conditions. We chose accompanying principles: 1/k, a/(b+k) (note that 1/k is an exceptional instance of this), and log(k)/k. An impediment of a/(b+k) is that one needs to lead various trials to decide suitable estimations of an and b, where as alternate standards don't have such parameters. We will confine our keenness with respect to the strange Q Learning figuring [6] for which union has been developed under odd conditions in different works (see e.g., [6]). Overall the written work, the step sizes are required to satisfy some fundamental conditions, for instance, and where shows the step size in the accentuation. For some distinctive less comprehended conditions, see [7]; all the three principles satisfy these conditions. Our tests will investigate the execution of a Q-Learning figuring with that of value cycle [8] which of preparing the quality limit registers Q-values. Let r (i, a, j) show the prize earned in going from state i to state j under movement a. Let p (i, a, j) demonstrate the probability related with the same move. Use μ to mean a technique for which μ(i) will connote the (deterministic) action to be picked in state i; e.g., (2,1). We will imply a procedure with movement 2 in state 1 and action 1 in state 2. Let imply the discount part. Similarly, Pμ and Rμ will mean the move probability furthermore, move prize cross sections, independently, associated with course of action μ. Finally, Q (i,a) will mean the Q-regard for state i and action a.

1.1 Parameters for mdp1

The first test instance, ie) mdp1, is a 2-state MDP with the following parameters: λ = 0.8, and

$$P(1,1) = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}; \; P(2,2) = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix};$$

$$R(1,1) = \begin{bmatrix} 6.0 & -5 \\ 7.0 & 12 \end{bmatrix}; \; R(2,2) = \begin{bmatrix} 10.0 & 17 \\ -14 & 13 \end{bmatrix}.$$

1.2 Parameters for other test events

The other test events, which are described as takes after. Each one of the parameters for the remaining test events are vague to those of mdp1 with the going with uncommon cases: mdp2 — r(1,1,2)=5 and r(2,2,1)=14; mdp3—r(1,2,1)=12; mdp4 — r(1,1,1)=16.

1.3 Numerical results

Now the numerical results got in five settings:

Q-Learning with the three unmistakable step-size norms, RLPF that uses the log-standard for the neuron's learning administer, and regard accentuation performed with Q-values; see Table 1. The estimation of a = 150 and b = 300 in our tests. Also, ε = 0.01 in the value accentuation computation [7]; the essential change in that count is: For all (i,a) do until ε - union: $Q(i,a) \leftarrow \sum_j p(i,a,j)\,[r(i,a,j) + \lambda \max_b Q(j,b)]$. The RLPF computations were continuing running for 10,000 accentuations, with an examination probability set at 0.5 all through. The PC activities were made in MATLAB.

This table investigates the Q-values procured by method for Q Learning (Q-L) under the diverse step-size rules, by method for RLPF, and by method for worth cycle using Q-values (Q-VI). Q-L-stomach muscles will mean Q-Learning with rule a/(b+k), Q-L-k will mean Q-Learning with standard 1/k besides, Q-Learning with the log rule will be implied by Q-L-log.

Table 1

	Method	**Q(1,1)**	**Q(1,2)**	**Q(2,1)**	**Q(2,2)**
mdp1	Q-VI	43.74	52.02	49.87	47.28
mdp1	Q-L-ab	43.40	51.97	50.84	44.63
mdp1	Q-L-k	10.46	17.74	18.62	15.52
mdp1	Q-L-log	38.24	46.79	43.26	41.24
mdp1	RLPF	42.90	50.90	49.54	48.26
mdp2	Q-VI	50.67	54.76	56.34	60.45
mdp2	Q-L-ab	49.55	53.53	55.11	59.94
mdp2	Q-L-k	16.12	19.38	20.08	22.53
mdp2	Q-L-log	44.61	49.16	49.07	52.78
mdp2	RLPF	49.99	54.70	58.27	63.01
mdp3	Q-VI	51.36	59.83	55.66	52.59
mdp3	Q-L-ab	48.26	59.82	55.66	50.18
mdp3	Q-L-k	11.47	20.72	19.17	17.89
mdp3	Q-L-log	42.36	53.67	48.09	44.60
mdp3	RLPF	47.20	58.43	55.19	52.38
mdp4	Q-VI	47.97	39.91	48.36	46.02
mdp4	Q-L-ab	46.52	39.29	47.93	42.53
mdp4	Q-L-k	15.10	8.16	17.96	14.64
mdp4	Q-L-log	41.73	33.57	41.38	38.52
mdp4	RLPF	47.34	39.42	48.71	48.46

The results show that while all the NRL counts meet to the perfect approach, the 1/k-rule produces values that avoid the perfect Q-values made by the value accentuation estimation. Possibly this behavior can be improved by diminishing examination, yet that will exhibit additional parameters for tuning. What is charming is that speculatively every one of the gauges is guaranteed to take us to perfect Q-values. The best execution was made by the a/(b+k) standard; it must be noted, then again, that the log-rule which does not have any tuning parameter performs much better than the 1/k-rule similarly as approximating the quality limit. The poor execution of 1/k can be elucidated by the way that it spoils quickly. Also, encouraging is the execution of RLPF count that uses a log-standard for the neuron's internal learning and an a/(b+k)- standard for the estimation. The results demonstrate that 1/k (used as a piece of [9] is possibly not an impeccable choice for most cases the log guideline has every one of the reserves of being promising, and there is a need to find parameter-less principles (which don't have parameters), for instance, a) and b) that can be used without mind boggling experimentation. It ought to be raised that in unlimited scale issues, one does not have the benefit of acknowledging what the perfect quality limit is and it is greatly fundamental that one has a stage size choose that takes one close optimality. In immense scale issues, it is exceptionally possible that the standard which causes paramount deviation from the perfect regard work truly drives one to a risky methodology.

2. Average Reward

Now turn our respect for typical prize MDPs. The computational studies with a Q-Learning count that uses two time scales for updating and from this time forward requirements two differing step sizes in the meantime. Different estimations with showed joining properties fuse a variation of Q-Learning in light of relative worth cycle (see e.g., [6]). Here, our wish is to inspect the impact of the stochastic briefest route overall compensate issues in NRL. The perfect quality limit using a regard accentuation computation is enlisted for typical prize. Let ρ^{μ} mean the typical prize of the system μ and ρ^{*} mean the perfect ordinary prize. By then if ρ^{*} is known, one can add to a quality cycle count for ordinary prize issues. It must be seen that such a quality cycle count is being focused on here only for the sole motivation behind testing how far the Q-Learning figuring strays from the thought values (doubtlessly, by ρ^{*} is dark, and one must use distinctive estimations; see e.g.,[8]).

The quality accentuation computation will have the going with major change: For all (i,a) do until ε - meeting:

$\mathrm{Q(i,a)} \leftarrow \sum_{j}$ p(i,a, j) [r(i,a, j) $-\rho^{*}$+max$_{b}$ Q(j,b)]. The Q-learning computation with its SSP-setting up framework is portrayed in the Appendix. It has two stage sizes: α (k) for the Q-regard and β (k) for the estimation of ρ, where $\lim_{k\to\infty} \beta(k)/\alpha(k) = 0$. The experiments used as a part of the last territory with the understanding that there is as of now no discount segment. The results are masterminded in Table 2. The Q-learning estimation ran for 10,000 cycles and used $\varepsilon = 0.01$; in like manner mdp1 — ρ^{*} = 10.56, mpd2 — ρ^{*}

= 11.53, mdp3 — ρ^* = 12.00 and mdp4 — ρ^* = 9.83. These qualities for ρ^* were controlled by a careful evaluation of the typical prize of each deterministic course of action. The examination probability was settled at 0.5 for both exercises. The results show that the quality limit, which is described as

v(i) = max $_a$ Q(i,a), is sensibly approximated by the Q-Learning count, though some Q-qualities are not by any stretch of the imagination all around.

3. Is Bellman Optimality Worth Fulfilling?

The numerical results of this range and the past section raise a basic issue. Is Bellman optimality, which infers achieving the quality limit that would happen due to comprehension the Bellman scientific proclamation, really worth fulfilling, or would it be okay for an estimation to create the perfect game plan? Note that in Section 2.3, the 1/k-rule and the log standard produce perfect methodologies, notwithstanding the way that the quality limit they make strays amazingly from that created by element programming (Bellman numerical proclamation). The same is substantial for the results for ordinary prize. This is an issue that requires further examination. A key question that should be had a tendency to is: the measure of deviation in the quality limit can be persevered? In that capacity, by what sum can the value limit go off to some faraway place without achieving a blemished methodology? The reaction to this request might make prepared to comprehending the MDP without strict adherence to Bellman measures. It has exhibited that that for any given state, if the incomparable estimation of the screw up in the quality limit is not as much as half of the aggregate estimation of the refinement between the Q-estimation of the perfect limit and the Q-estimation of the dangerous movement by then that bumble can be persevered. Nevertheless, a through and through examination of this issue may wind up being of noteworthiness later on — especially in the association of limit speculation, where shown clear deviation from Bellman optimality.

This table takes a gander at the Q-values got past Q-Learning (Q-L) for ordinary prize (see Appendix) and through quality cycle using Q-values (Q-VI). For mdp2

α (k) = 500/ (1000+k) and β (k) =150/ (300+k), while for the remaining events α (k) = 150/ (300+k) and β (k) = 50/ (49+k) are used.

Table 2

	Method	Q(1,1)	Q(1,2)	Q(2,1)	Q(2,2)
mdp1 mdp1	Q-L RLPF	-4.46 -8.99	0.18610 1.2789	-0.76 -2.47	-4.12 -4.80
mdp2 mdp2	Q-L RLPF	-3.85 -2.85	1.517 0.47	5.48 8.32	7.10 8.31
mdp3 mdp3	Q-L RLPF	-4.99 -10.80	0.1961 0.49	-5.81 -4.446	-6.19 -8.39
mdp4 mdp4	Q-L RLPF	-2.14 -1.1904	-7.29 -9.24	-0.58 0.125	-3.45 -2.858

II. SURVIVAL PROBABILITY

The thought of peril has been focused on in the setting of NRL through utility limits [10], distinction disciplines [10] and probability of entering denied states [12]. See [13] for an earlier work. Contrast disciplines in the setting of MDPs were inspected in [14]. In this paper, consider the disciplines associated with downside risk which is portrayed concerning a goal. Given a center for the one-stage reward, and describe the downside peril (DR) to be the probability of the prize falling underneath the emphasis on; this threat should be minimized. In this way 1−DR show the probability of survival, this is opened up. If one considers costs instead of prizes, the probability of surpassing the target will be the related downside danger; this is in like manner called the exceedance probability in disaster showing [3]. Next, NRL figuring is presented by us.

1. Reinforcement Learning

Reinforcement learning (RL) is a machine learning approach, in which the goal is to find a policy π that maximizes the expected future return, calculated based on a scalar reward function R(.) € R. The argument of R(.) can be defined in different ways, e.g. it could be a state s, or a state transition, or a state-action pair, or a whole trial as in the case of episodic RL, etc. The policy π determines what actions will be performed by the RL agent, and is usually state dependent.

Originally, the RL problem was formulated in terms of a Markov Decision Process (MDP) or Partially Observable MDP (POMDP). In this formulation, the policy π is viewed as a direct mapping function (π : s→a) from state s € S to action a € A.

Alternatively, instead of trying to learn the explicit mapping from states to actions. In this case, the policy π is considered to depend on some parameters $\theta \xi R^N$, and is written as a parameterized function $\pi(\theta)$. The episodic reward function becomes $R(\tau(\pi(\theta)))$, where τ is a trial performed by following the policy. The reward can be abbreviated as $R(\tau(\theta))$ or even as R(θ), which reacts the idea that the behavior of the RL agent can be incensed by only changing the values of the policy parameters θ. Therefore, the outcome of the behavior, which is represented by the reward R(θ), can be optimized by only optimizing the values θ. This way, the RL problem is transformed into a black-box optimization problem with cost function R(θ),

However, it is infeasible to use conventional numeric optimization methods to maximize R(θ) if we want to apply RL to real-world problems, because the cost function is usually expensive to evaluate. For example, each cost function evaluation requires conducting at least one trial which could involve costly real-world experiments or computationally expensive simulations. Therefore, alternative optimization methods are desired, which are tuned to the specific need of RL to reduce as much as possible the number of reward function evaluations (trials).

This paper aims to bring new ideas into the domain of RL, borrowed from statistics. They proposed a novel direct policy search RL algorithm which provides an alternative solution to the RL problem, and new possibilities for RL algorithms in general. The aim is to provide an intuitive presentation of the ideas rather than concentrate on the deeper mathematics underlying the topic. RL is generally used to solve the so-called Markov decision problem (MDP). In other words, the problem that you are attempting to solve with RL should be an MDP or its variant. The theory of RL relies on dynamic programming (DP) and artificial intelligence (AI). they will begin with a quick description of MDPs. We will discuss what we mean by “complex” and “large-scale” MDPs. Then explain why RL is needed to solve complex and large-scale MDPs. The semi-Markov decision problem (SMDP) will also be covered. The tutorial is meant to serve as an *introduction* to these topics and is based mostly on the book: “Simulation-based optimization: Parametric Optimization techniques and reinforcement learning”. The book discusses this topic in greater detail in the context of simulators. There are at least two other textbooks namely:

(i) Neuro-dynamic programming (lots of details on convergence analysis) and (ii) Reinforcement Learning: An Introduction (lots of details on underlying AI concepts). They describe a basic RL algorithm that can be used for average reward SMDPs. Note that if *t*(*i, a, j*) = 1 for all values of *i, j*, and *a*, we have an MDP. Hence our presentation will be for an SMDP, but it can easily be translated into that of an MDP by setting *t*(*i, a, j*) = 1 in the steps.

It is also important to understand that the transition probabilities and rewards of the system are not needed if any one of the following is true:

1. We can play around in the real world system choosing actions and observing the rewards
2. if we have a simulator of the system.

The simulator of the system can usually be written on the basis of the knowledge of some other easily accessible parameters. For example, the queue can be simulated with the knowledge of the distribution functions of the inter-arrival time and the service time. Thus the transition probabilities of the system are usually **not** required for writing the simulation program. Also, it is important to know that the RL algorithm that we will describe below requires the updating of certain quantities (called *Q*-factors) in its database whenever the system visits a new state.

When the simulator is written in C or in any special package such as ARENA, it is possible to update certain quantities that the algorithm needs whenever a new state is visited.

Usually, the updating that we will need has to be performed immediately after a new state is visited. In the simulator, or in real time, it IS possible to keep track of the state of the system so that when it changes, one can update the relevant quantities. The key idea in RL is store a so-called *Q*-factor for each state-action pair in the system. Thus, *Q*(*i, a*) will denote the *Q*-factor for state *i* and action *a*. The values of these *Q*-factors are initialized to suitable numbers in the beginning (e.g., zero or some small number to all the Q-factors). Then the system is simulated (or controlled in real time) using the algorithm. In each state visited, some action is selected and the system is allowed to transition to the next state. The immediate reward (and the transition time) that is generated in the transition is recorded as the feedback. The feedback is used to update the *Q*-factor for the action selected in the previous state. Roughly speaking if the feedback is good, the *Q*-factor of that particular action and the state in which the action was selected is increased rewarded using the Relaxed-SMART algorithm. If the feedback is poor, the *Q*-factor is punished by reducing its value.

Then the same reward-punishment policy is carried out in the next state. This is done for a large number of transitions. At the end of this phase, also called the learning phase, the action whose *Q*-factor has the highest value is declared to be the optimal action for that state. Thus the optimal policy is determined. Note that this strategy does not require the transition probabilities.

In this section we present a simple example as 'proof of concept'. More extensive examples are being planned and will be reported elsewhere.

Consider a queuing system,

$$Q_{n+1} = (Q_n - D_n I\{X_n > 0\} + \xi_{n+1}) \wedge 100.$$

Here, given 0 < b << a < 1,

- D_n (the departure process) is i.i.d., D_n = 1 with probability μ_n € {a,b} ,n>0 otherwise,
- €$_n$(the arrival process) is i.i.d., €$_n$ = 1 with probability $\lambda, b < \lambda < a$ and 0 otherwise,
- $\{\mu_n\}$ (The service rate) is a fa; bg-valued Markov chain with two states: 'working' when μ_n = a and 'faulty' where μ_n = b.

Denote by p $_2$ (.|.) the transition probabilities of { μ_n }. We have limited the maximum queue size to 100, i.e., assumed a finite buffer of size 100. In case of buffer overflow, the extra packets are assumed lost.

In this example the observed process is the queue length { Q_n } and the state process is { μ_n, Q_n }. Comparing with our earlier notation, the correspondence is:

$$X_n \leftrightarrow \{\mu_n, Q_n\}, Y_n \leftrightarrow Q_n$$

Thus Xn € {1…………,100} *{a, b} := S and Yn € {1; ::::; 100g}:= 0. The transition probability function is: p(u',I'),i'(i,u)) = P(X$_{i+1}$=(I',u'),Y$_{n+1}$=I'|X$_n$ =(u,i)), where for 0<i<100

The nonlinear filter then turns out to be

$$v_{n+1}(j,q) = \sum_{i=a,b} v_n(i,Q_n) p(j,Q_{n+1}) | i, Q_n)$$

Where q= Q_{n+1} ,otherwise $v_{n+1}(j,q)$ =0.

The flow chart of Reinforcement Learning is shown below.

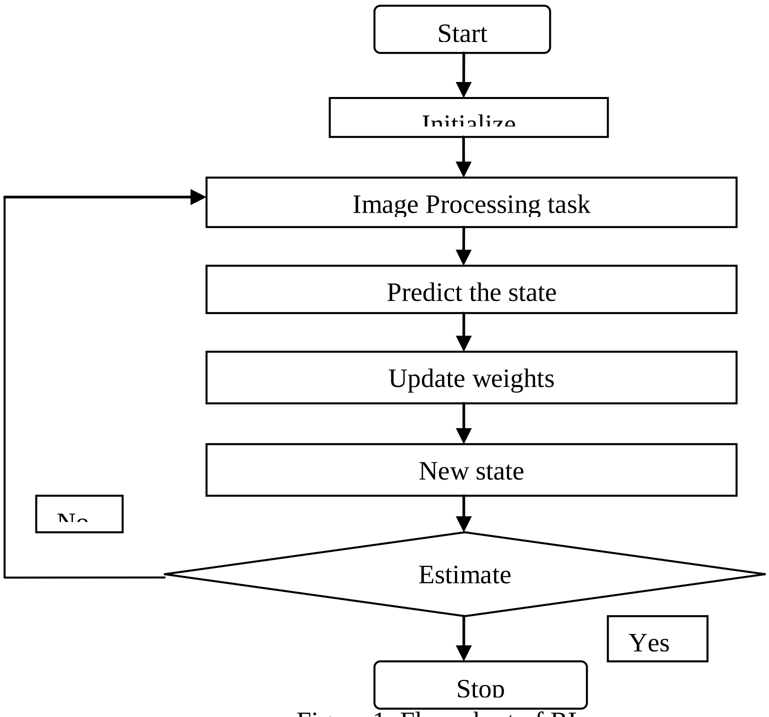

Figure 1: Flow chart of RL

2. Partical filter

Particle filters, also known as Sequential Monte Carlo methods[1], originally come from statistics and are similar to importance sampling methods. Particle filters are able to approximate any probability density function, and can be viewed as a `sequential analogue' of Markov chain Monte Carlo (MCMC) batch methods.

Although particle filters are mainly used in statistics, there are a few other research areas in which particle filters have found application. For example, in the domain of probabilistic robotics [2] , particle filters are extensively and successfully used, e.g. for performing Monte Carlo localization of mobile robots with respect to a global map of the terrain[3], and also for Simultaneous Localization and Mapping (SLAM) task[4]The potential of applying particle filters in the RL domain appears to be largely unexploredso far.

To the best of our knowledge, there are only two partial attempts to apply particle filters in RL in the existing published work, done by Notsu et al and Samejimaet al, respectively, as follow they studied traditional RL with discrete state and action spaces. They used the Actor-Critic method for performing value iteration, with state value updates using the TD-error. In this conventional framework, they proposed a particle filter for segmentation of the action space. Their goal was to do what they called `domain reduction', or trying to minimize the number of discrete actions available at each state, by dividing the action space in segments., they extended the same idea to traditional continuous RL and used Q-learning with function approximation.

They used a similar particle filter approach for segmenting the continuous state and action spaces into discrete sets by particles, and applied it to inverted pendulum tasks., they studied neurophysiology and created a reinforcement learning model of an animal behaviour. Their goal was to predict the behaviour of a monkey during an experimental task. They used traditional Q-learning, and built a Bayesian network representation of the Q-learning agent. In this framework, particle filtering was used to estimate action probability in order to predict the animal behaviour.

In both of these existing approaches, particle filters were used in a limited way, as a technique to solve some partial problem within a traditional RL framework.

The propose a rather different approach. First, we propose a completely new view of the link between particle filters and RL. Then, we propose an entirely novel RL algorithm for direct global policy search, based on particle filters as the core for the RL algorithm itself. In our framework, the search is performed in the policy space defined by the selected policy parameterization, and the process is viewed as a black-box optimization.

The particle filter itself is the core of the proposed RL algorithm, and is responsible for guiding the exploration and exploitation, by creating particles, each of which represents a whole policy.

The flow chart of Particle filter is shown below.

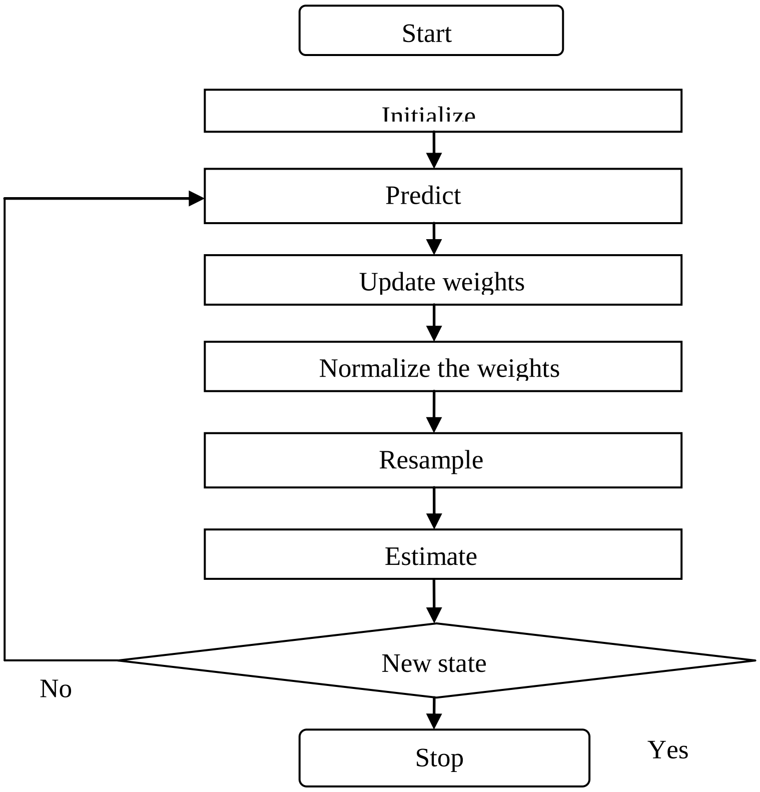

Figure 2: Flow chart of PF

Details of the proposed novel view and algorithm follow.

3. RLPF for Survival

A Q-regard version of the Bellman scientific explanation can be made from Equation (2) above. From that, it is not hard to decide a RLPF count. Subsequent to the Bellman correlation models the perfect estimation of the objective limit ϕ^*, (this is undifferentiated from ρ^* in the peril impartial Bellman numerical articulation for ordinary prize), and need to use a figuring that uses relative qualities. Our estimation's central components are as takes after. In the introductory step, pick some state-action pair to be a perceived state-action pair; call it (i^*, a^*). It can be shown that with likelihood1, $\lim_{k\to\infty} Q^k(i^*, a^*) = \rho^*$ [5]; intuition recommends that with probability 1:

$$\lim_{k\to\infty} Q^k(i^*, a^*) = \phi^*$$

Clearly, as communicated former, a theoretical affirmation is a subject of future work. Now lead diversion tests to choose how the computation.

3.1 Parameters for Test Events

Use four test events named mdp5 through mdp8. For all the test events, τ = 8 and θ = 2. Depict them next.

mdp5:
Indistinguishable to mdp1 except for: r(1,1,1) = 3; r(1,1,2) = 11; r(1,2,1) = 6;
r(2,2,2) = 7.
mdp6:
Indistinguishable to mdp1 except for: r(1,1,1) = 3; r(1,1,2) = 11; r(2,1,2) = 9;
r(1,2,1) = 6; r(2,2,2) = 7.
mdp7:
Indistinguishable to mdp1 to the extent the move probabilities, nevertheless, with the going with prize structures:

$$R_{(1,1)} = \begin{bmatrix} 9.0 & -1 \\ 12.0 & 8 \end{bmatrix}; \quad R_{(2,2)} = \begin{bmatrix} 6.0 & 20 \\ -14 & 7 \end{bmatrix}$$

mdp8:
Indistinguishable to mdp1 to the extent the move probabilities, nevertheless, with the going with prize structures:

$$R_{(1,1)} = \begin{bmatrix} 3.0 & 7 \\ 9.0 & 1 \end{bmatrix}; \; R_{(2,2)} = \begin{bmatrix} 6.0 & 9 \\ 14 & 7 \end{bmatrix}$$

4. Simulation Tests

At first analyzed through thorough appraisal the ordinary prize and the downside peril for each procedure in the 4 test samples. The ordinary prize is $\rho^\mu = \sum_{i\in\zeta} \pi(i) \sum_{j\in\zeta} p(i, \mu(i), j) r(i, \mu(i), j).$The disadvantage peril is described in Equation (1). The obliging probabilities of each state can be managed by unwinding the built up invariant numerical articulations: $\sum_{j\in\zeta} \pi^\mu(j) p(j, \mu(i), i) = \pi^\mu(i)$ for all $i \in \zeta$ and $\sum_{i\in\zeta} \pi^\mu(i) = 1.$The results are displayed in Table 3. On each one of the representations, the figuring centered to perfect courses of action in 10,000 accentuations. Modify the examination probability at an estimation of 0.5 for each movement. In the table the value to which $Q(i^*, a^*)$ unites are appeared. Here, don't present all the Q values in light of the way that have not stood out them from qualities from component programming, and consequently the qualities free from any other individual go on nothing. What is extra captivating is the quality to which blends.

As is typical from Equation (3), it centers to a value near ϕ^* . In like manner, observe that for mdp5, mdp6, and mdp7, the threat impartial perfect course of action (that supports ρ) does not agree with the peril tricky perfect approach (that increases our risk punished score, ϕ).

5. Semi-Markov Control

A trademark and indispensable increase of MDP theory is to Semi-Markov Decision Process (SMDPs) [8], where the time spent in each move is shown as a sporadic variable. Let t(i,a,j)) imply the time spent in going from i to j under movement a. At first require the meanings of the risk measures considered already. Downside risk will be portrayed as:

$$DR^{\mu} = \sum_{i\in\zeta} \pi^{\mu}(i)\, p(i,\mu(i),j) I\left(\frac{r(i,\mu(i),j)}{t(i,\mu(i),j)} < \tau \right).$$

The relating Bellman numerical explanation would be:

$$\max_{a\in A(i)} \left[\sum_{j\in\zeta} p(i,a,j) \left\{ r(i,a,j) - \theta I(r(i,a,j) < \tau t(i,a,j)) - \phi^{*} t(i,a,j) + J(j) \right\} \right]$$

Semi-change in the SMDP can be portrayed as:

$$\sum_{i\in\zeta} \pi^{\mu}(i) \sum_{j\in\zeta} p(i,\mu(i),j)(\tau t(i,\mu(i),j) - r(i,\mu(i),j))^{2}{}_{+}.$$

The SMDP Bellman numerical explanation for semi-contrast can be procured from that of downside threat through substitution of the pointer limit by A RLPF count can in like manner be resolved for semi-change. For change, the need to portray a few sums first. Use the energizing prize speculation (RRT) since fundamental the SMDP, one has a reclamation process. Consider an including process , and let imply the time between the event and the event in the technique; In the occasion that connotes a game plan of non-negative i.i.d subjective variables, then is an energizing procedure. Let mean the prize assembled in the nth reviving in the reclamation process basic the SMDP. Also, let and The ordinary prize for the SMDP can be shown through the RRT to be : where (the movement an in each state i is described by the methodology under thought)

$$E[R] = \sum_{i\in\zeta} \pi(i) \sum_{j\in\zeta} p(i,a,j) r(i,a,j) \text{ and}$$

$$E[T] = \sum_{i\in\zeta} \pi(i) \sum_{j\in\zeta} p(i,a,j) t(i,a,j).$$

The characteristic definition for the asymptotic difference is characterized in (4) beneath. From the RRT, realized that 1 (w.p.1), $\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{E[T]}$ using which can work out the following:

$$\sigma^{2} \equiv \lim_{t\to\infty} \frac{\sum_{n=1}^{N(t)} \left[R_n - \rho T_n \right]^{2}}{t}$$

$$= \lim_{t\to\infty} \sum_{n=1}^{N(t)} \left[\frac{R_n^2 - 2\rho T_n R_n + \rho^2 T_n^2}{N(t)} \right] \frac{N(t)}{t}$$

$$= \frac{E\left[R^2\right]}{E\left[T\right]} - 2\rho \frac{E\left[T.R\right]}{E\left[T\right]} + \rho^2 \frac{E\left[T^2\right]}{E\left[T\right]} \quad \text{(W.P.1)}$$

$$= \frac{E\left[R^2\right]}{E\left[T\right]} - 2\rho \frac{E\left[T\right]E\left[R\right]}{E\left[T\right]} + \rho^2 \frac{E\left[T^2\right]}{E\left[T\right]}$$

(since T and R independent)

$$= \frac{E\left[R^2\right]}{E\left[T\right]} - 2\rho^2 E\left[T\right] + \rho^2 \frac{E\left[T^2\right]}{E\left[T\right]},$$

where

$$E[R^2] = \sum_{i\in\zeta} \pi(i) \sum_{j\in\zeta} p(i,a,j) r^2(i,a,j) \text{ and}$$

$$E[T^2] = \sum_{i \in \zeta} \pi(i) \sum_{j \in \zeta} p(i,a,j) t^2(i,a,j).$$

Utilizing $E[R], E[R^2], E[T]$ and $E[T^2]$ one can characterize the change of the SMDP. At that point if ρ^* means the normal prize of the strategy that upgrades a change punished SMDP, then the Bellman mathematical statement for the fluctuation punished SMDP can be acquired by supplanting the pointer capacity in the relating mathematical statement for drawback hazard by $(r(i,a,j) - \rho^* t(i,a,j))^2$.

III. CONCLUSION

This paper exhibited an exact investigation of (i) the utilization of distinctive step-sizes in marked down RLPF, (ii) the utilization of most brief stochastic ways in normal prize RLPF, and (iii) the idea of survival likelihood or drawback hazard in RLPF. The experimental study with the stride size (Section 1) demonstrates that the 1/k-principle does not give off an impression of being a dependable or powerful decision even on exceptionally little issues, and that the $(a/b+k)$- standard performs extremely well on little issues, however the estimations of a and b need to be resolved. The log-guideline performs sensibly well, and its leeway is that it doesn't have any tuning parameters. The observational study with the stochastic ways shows that utilizing SSP establishing. one acquires sensible approximations of the genuine quality capacity. Our observational results do point to the requirement for Concentrate the amount of deviation can be endured from Bellman optimality. At last, introduce another RLPF calculation that permits the improvement of a survival-likelihood punished target capacity. Numerical results on little test issues demonstrate that the algorithm performs well. A theoretical study of this algorithm is a topic for future research.

This table lists the ρ, DR and ϕ values of all the policies along with the value of $Q^{\infty}(i^*, a^*)$; use i^* = 1and a^* = 1. The values in bold are those for the optimal policy.

TABLE III

	μ	ρ^μ	DR^μ	Φ^μ	$Q(i^*, a^*)$
mdp5	(1,1)	7.35	0.53	6.20	6.201
	(1,2)	3.785	1.00	2.1800	-
	(2,1)	7.80	0.7	7.08	-
	(2,2)	5.7	0.963	3.8	-
mdp6	(1,1)	7.9143	0.4	7.143	6.505
	(1,2)	3.34	0.98	2.08	-
	(2,1)	7.84	0.92	7.34	-
	(2,2)	5.167	0.833	3.5	-
mdp7	(1,1)	7.3429	0.3714	7.0	7.380
	(1,2)	4.01	0.572	2.24	-
	(2,1)	7.14	0.62	6.01	-
	(2,2)	6.066	0.833	5.000	-
mdp8	(1,1)	3.2	1.0286	2.0429	-
	(1,2)	5.72	1.098	4.96	-
	(2,1)	6.88	0.54	3.2	-
	(2,2)	8	0.7967	4.266	5.614

References

Journal Papers:

[1]. Grossi, P., and H. Kunreuther. " Catastrophe modeling: A new approach to managing risk". Springer. 2005.

[2]. Even-Dar, E., and Y. Mansour. *"*Learning rates for Q learning, *Journal of Machine Learning Research" 5:1–25*. 2003.

[3]. Borkar, V. S., and S. Meyn. " The ODE method for convergence of stochastic approximation and reinforcement learning". *SIAM Journal of Control and Optimization"38(2):447–469*.2000.

[4]. Borkar, " V. S. Asynchronous stochastic approximation, *SIAM Journal of Control and Optimization" 36No3:840–851*. 1998.

[5]. Puterman, M. L. Markov "Decision processes". New York: Wiley Interscience. 1994.

[6]. Gosavi, A. "A reinforcement learning algorithm based on policy iteration for average reward: Empirical results with yield management and convergence analysis. Machine Learning"55(1):5–29. 2004a.

[7]. Borkar, V. "Q-learning for risk-sensitive control. Mathematics of Operations Research" 27(2):294–311. 2002.

[8]. Geibel, P., and F. Wysotzki. "Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research" 24:81–108*. 2005.

[9]. Filar, J.,L.Kallenberg, and H.Lee. "Variance-Penalised Markov decision Processes.Mathematics of operations Research" 14(1):147-161.1989.

Proceedings Papers:

[10]. Abhijit Gosavi "On Step Sizes, Stochastic Shortest Paths, and Survival Probabilities in Reinforcement Learning" *Proceedings of the Winter Simulation Conference, USA,* 2008.

[11]. Sato, M., and S. Kobayashi. "Average-reward reinforcement learning for variance-penalized Markov decision problems, In *Proceedings of the 18th International Conference on Machine Learning",* 473–480. 2001.

[12]. Heger, M. " Consideration of risk in reinforcement learning". *Proceedings of the 11th International Conference on Machine Learning":* 105–111. 1994.

Books:

[13]. Morgan Kauffman. Sutton. R., and A. G. Bartow. " *Reinforcement learning: An introduction. Cambridge*" MA, USA: The MIT Press. 1998.

Theses:

[14]. Watkins, C., May. " *Learning from delayed rewards*". Ph. D. thesis, Kings College, Cambridge, England. 1989.