Solving Stochastic Point Location Problem in a Dynamic Environment with Weak Estimation

Asieh Abolpour Mofrad
University College of Oslo and Akershus
Oslo, Norway
asieh.abolpour-mofrad@hioa.no

Anis Yazidi
University College of Oslo and Akershus
Oslo, Norway
anis.yazidi@hioa.no

Hugo Lewi Hammer
University College of Oslo and Akershus
Oslo, Norway
hugo.hammer@hioa.no

ABSTRACT

The Stochastic Point Location (SPL) problem introduced by Oommen [7] can be summarized as searching for an unknown point in the interval under a possibly faulty feedback. The search is performed via a Learning Mechanism (LM) (algorithm) that interacts with a stochastic environment which in turn informs it about the direction of the search. Since the environment is stochastic, the guidance for directions could be faulty. The first solution to the SPL problem which was pioneered by Oommen [7] two decades ago relies on discretizing the search interval and performing a controlled random walk on it. The state of the random walk at each step is considered to be the estimation of the point location. The convergence of the latter simplistic estimation strategy is proved for an infinite resolution. However, the latter strategy yields rather poor accuracy for low resolutions. In this paper, we present sophisticated tracking methods that outperform Oommen strategy [7]. Our methods revolve around tracking some key statistical properties of the underlying random walk using the family of weak estimators. Furthermore, we address the settings where the point location is non-stationary, i.e. LM is searching with uncertainty for a (possibly moving) point in an interval. In such settings, asymptotic results are no longer applicable. Simulation results show that the proposed methods outperform Oommen method for estimating point location by reducing the estimated error up to 75%.

CCS CONCEPTS

• Mathematics of computing → Markov processes; **Optimization with randomized search heuristics**; **• Theory of computation →** *Streaming models*; **• Computing methodologies →** *Reinforcement learning*; **• Applied computing →** Psychology;

KEYWORDS

Stochastic point location (SPL); Stochastic learning weak estimator (SLWE); Random walk

1 INTRODUCTION

Stochastic Point Location (SPL) is a fundamental optimization problem that was pioneered by Oommen [7] and ever since it has received increasing research interest [4, 11]. A Learning Mechanism (LM) attempts to locate a unique point λ^* in an interval whilst the only assistance are the information from a random environment (E) which informs it, possibly erroneously, whether the location is to the left or to the right of the point. A large class of optimization problems could be also modeled as the instantiation of the SPL problem [11].

Interestingly, the SPL problem deals with dynamic environments since the assumption that the parameter or point location in SPL does not change over time is not valid in many real-life settings such as web-based applications [3]. Sliding window [5] is a traditional strategy for estimation in non-stationary environments. However, choosing the appropriate width of the window is crucial; too small width results in a poor estimate, whereas if one increases the width, the stale values prior to the change might degrade the whole estimation process. The Stochastic Learning Weak Estimator (SLWE) [10] figures among the most prominent estimators for non-stationary distributions. In this article, we propose to integrate the SLWE as the inherent part of a more sophisticated and accurate solution for the SPL. The recursive update form of the SLWE makes it a viable strategy in our problem, since the tracked distribution in question is updated at each time step. Therefore, our strategy for estimation of point location revolves around tracking the distribution at each time step and estimating the point based upon it.

It is worth mentioning that Continuous Point Location with Adaptive Tertiary Search (CPL-ATS) strategy [8] is another method of solving SPL which systematically and recursively searches for sub-intervals that λ^* is guaranteed to locate in, with an arbitrarily high probability. A series of guessing which starts with the mid-point of the given interval, estimates the point location and repeats until the requested resolution is achieved. The given interval is partitioned into three sub-intervals where three LA work in parallel in each sub-interval and at least one of them will be eliminated from further search. So it is crucial in CPL-ATS to construct the partition and elimination process. This method is further developed into the CPL with Adaptive d-ary Search (CPL-AdS) Strategy [9] where the current interval is partitioned into d sub-intervals instead of three. The larger d results in faster convergence but the decision table of elimination process gets more complicated. An extension of the CPL-AdS scheme which could also operate in non-stationary environments is presented in [4]. The decision formula is proposed to modify the decision table in [9] to resolve certain issues of original CPL-AdS scheme.

There is a wide range of scientific and real-life problems that can be modeled as the instances of SPL problem, such as adaptive data encoding, web-based applications, etc. [3]. SPL can be also used to find the appropriate dose in clinical practices and experiments [6].

A possible interesting application which we focus on in our ongoing research is to determine the difficulty level of a cognitive training method by SPL. One of the key challenges faced by many learning methods is to find the level of the participant in order to provide suitable level of training. To the best of our knowledge, in most legacy methods, alternating between different training levels and scenarios is simply done by increasing the difficulty if the task is managed (once or in a fraction of repeated times) or by decreasing (or fixing) the difficulty level if it is not managed. We believe that this problem could be modeled by SPL with certain conditions; such as non-stationary point location (since the manageable difficulty level will change as time advances for trained participant).

Spaced retrieval training (SRT) [2] for instance, is a method of learning and retaining target information by recalling that information over increasingly longer intervals which is especially used for people with dementia [1]. For progressive diseases like dementia, it is so important to estimate the ability level (point location in SPL) as soon as possible, as the ability usually rapidly changes during time (affected by training, disease and patient's condition).

The rest of paper is organized as follows. In Section 2, the SPL problem is defined formally. Section 3 is devoted to SLWE method and the way we use it to estimate the point location. Sections 3.1 and 3.2 focus in more details on the design and update of the two probability vectors and the three estimations based upon them which are compared in Section 3.3. Simulation results and a comparison with Oommen method are presented and discussed in Section 4. Finally, we conclude in Section 5.

2 STOCHASTIC POINT LOCATION PROBLEM IN A DYNAMIC SETTING

This problem considers a learning mechanism (LM) that is moving within $[0, 1]$ interval and which attempts to locate a point ($0 \leq \lambda^*(n) \leq 1$) that may change over time n. The environment E is considered informative whenever it informs the LM correctly about the right direction of the unknown point with probability $p > 0.5$. This probability of receiving a correct response, which reflects the "effectiveness" of the environment is known by LM and assumed to be constant.

As aforementioned, we would like to estimate $\lambda^*(n)$ as soon as possible. We follow the model presented on [7] and discretize the interval and perform a controlled random walk on it, characterized by $\lambda(n)$ which is a quantity related to the current position of the random walker. More precisely, we subdivide the unit interval into $N + 1$ discrete points

$$\{0, 1/N, 2/N, \cdots, (N-1)/N, 1\},$$

where N is called the resolution of the learning scheme. Let $\lambda(n)$ be the current location at time step n:

- If the environment E suggests increasing $\lambda(n)$ and $0 \leq \lambda(n) < 1$ then
  $\lambda(n+1) = \lambda(n) + 1/N$
- If E suggests decreasing $\lambda(n)$ and $0 < \lambda(n) \leq 1$ then
  $\lambda(n+1) = \lambda(n) - 1/N$
- If ($\lambda(n) = 1$ and E suggests increasing $\lambda(n)$) OR ($\lambda(n) = 0$ and E suggests decreasing $\lambda(n)$) then:
  $\lambda(n+1) = \lambda(n)$

Hereafter the binary function $E(n)$ stands for the environment answer at step n where $E(n) = 1$ refers to the environment suggestion to increase $\lambda(n)$ and $E(n) = 0$ refers to the environment suggestion to decrease $\lambda(n)$.

Based on results presented in [7], in the stationary case in which $\lambda^*(n) = \lambda^*$, this random walk will converge into a value arbitrarily close to λ^* when $N \to \infty$ & $n \to \infty$. However the above asymptotic results are not valid for the non-stationary SPL. Practically we might experience some constraints both on time $n \leq T$ and on the resolution $N \leq R$. Throughout the rest of this paper we pursue better estimates for $\lambda^*(n)$ than $\lambda(n)$.

3 ESTIMATION STRATEGIES

In this section for finding an estimation of $\lambda^*(n)$ based on the random walk, two multinomially distributed random variables are considered. We track their probability distribution with SLWE method [10] and estimate the $\lambda^*(n)$ from the estimated distributions.

3.1 Probability vector for states

Let $X(n)$ be a multinomially distributed random variable which takes its values from set $\{x_i | x_i = i/N \text{ for } i = 0, 1, \ldots, N\}$ and suppose $x_z \leq \lambda^* < x_{z+1}$ for a specific z. Further let $P(X(n) = x_i) = s_i(n)$, $i = 0, 1, \ldots, N$. Further let $x(n)$ be the concrete realization of $X(n)$ at time step n. From the definitions of Section 2, we have $x(n) = \lambda(n)$

The SLWE method estimates the probabilities $S(n) = [s_0(n), s_1(n), \ldots, s_N(n)]^T$ by maintaining a running estimate $P(n) = [p_0(n), p_1(n), \cdots, p_N(n)]^T$ of $S(n)$ where $p_i(n)$ is the estimate of $s_i(n)$ at time n. The updating rule is (the rules for other values of $p_j(n), j \neq i$, are similar):

$$\begin{aligned} p_i(n+1) &\leftarrow \alpha p_i(n) + (1-\alpha) \text{ when } x(n) = x_i \\ &\leftarrow \alpha p_i(n) \text{ when } x(n) \neq x_i \end{aligned} \tag{1}$$

α is a user-defined parameter, $0 < \alpha < 1$ for updating the probability distribution. The intuition behind the updating rule is that if $x(n) \neq x_i$ we should decrease our estimate p_i which is given by the second part of the updating rule. Similarly, if $x(n) = x_i$ we should increase our estimate which is given by the first part of the updating rule.

It is worth mentioning that in [10], $X(n) = X$, i.e. it is not modeled as a function of time and as a result $S(n) = [s_0, s_1, \cdots, s_N]^T$ is time-invariant. The theorems and results also proved in the asymptotic case when $n \to \infty$ which is in contradiction with the non-stationary assumption for environment. It is discussed that in practice the convergence takes place after a relatively small value of n. For instance if the environment switches its multinomial probability vector after 50 steps, the SLWE could track this change. However, we prefer to use the notation in a way that the point location and thereafter the multinomially probability vector clearly shown to be non-stationary. SLWE converges weakly, independent of α value, however the rate of convergence is a function of α.

Now that we obtained the estimate of probability vector at each time step, $P(n)$, the next step will be to consider three statistical parameters of this distribution to estimate $\lambda^*(n)$.

(1) The first estimation methodology is to choose the state with maximum probability as the estimate:

$$\begin{aligned} z &= \arg\max_i (p_i(n)) \\ \hat{\lambda}_{\max}(n) &= x_z \end{aligned} \tag{2}$$

where $\hat{\lambda}_{\max}(n)$ is the estimate and for non-unique z, the last visited state with the max probability value is chosen.

(2) The expected value of the $P(n)$ at step n is another possible estimation method:

$$\hat{\lambda}_{\exp}(n) = \sum_{i=0}^{N} x_i p_i(n) \tag{3}$$

(3) The median is the third estimation method which is given by:

$$\begin{aligned} &\hat{\lambda}_{\text{med}}(n) = x_z \text{ where } z \text{ is the index satisfying} \\ &\sum_{i=0}^{z} p_i(n) \geq 0.5 \text{ and } \sum_{i=z}^{N} p_i(n) \geq 0.5 \end{aligned} \tag{4}$$

Intuitively, it makes sense to estimate $\lambda^*(n)$ by the most visited state as given by (2). However, if the system varies rapidly, the estimate $P(n)$ will be quite poor and in such case taking the expectation, as given by (3), may be a more robust alternative. Finally, the median in (8) might be also a more robust alternative than the max in (2).

In Section 3.2 we suggest to use the probabilities for different state transitions instead of the probabilities given in Section 3.1. We argue that this might perform better since the estimation Markov chain will have many transitions around the true and unknown $\lambda^*(n)$.

3.2 Probability vector for transitions

Let x_{i+} denote the fact that the Markov chain makes a transition from x_i to x_{i+1} or from x_{i+1} to x_i. Further let $X_+(n)$ denote a multinomially distributed variable over the possible state transition $x_{i+}, i = 0, 1, \ldots, N-1$.

Further we define the portion of state transitions that go from x_i to x_{i+1} or from x_{i+1} to x_i, given by $P(X_+(n) = x_{i+}) = s_{i+}(n), i = 0, 1, \ldots, N-1$.

Let $x_+(n)$ be the concrete realization of $X_+(n)$ at time step n. From the definitions in Section 2 we have $x(n) = [\lambda(n), \lambda(n) + 1/N)$ or $[\lambda(n) - 1/N, \lambda(n))$.

The SLWE method estimates $S_+(n) = [s_{0+}(n), s_{1+}(n), \ldots, s_{(N-1)+}(n)]^T$ by maintaining a running estimate of $P_+(n) = [p_{0+}(n), p_{1+}(n), \cdots, p_{(N-1)+}(n)]^T$ of $S_+(n)$ where $p_{i+}(n)$ is the estimate of $s_{i+}(n)$ at time n. The updating rule is (the rules for other values of $p_{j+}(n), j \neq i$, are similar):

$$\begin{aligned} p_{i+}(n+1) \quad &\leftarrow \alpha p_{i+}(n) + (1-\alpha) \\ &\text{ if a transition } x_i \rightarrow x_{i+1} \text{ or } x_{i+1} \rightarrow x_i \\ &\leftarrow \alpha p_{i+}(n) \text{ otherwise} \end{aligned} \tag{5}$$

Again, α is a user-defined parameter, $0 < \alpha < 1$ for updating the probability distribution.

Similarly, based on $P_+(n)$ we propose three statistical parameters to estimate $\lambda^*(n)$ as follows:

(1) The first estimate is to choose the state with maximum probability as the estimate:

$$\begin{aligned} z &= \arg\max_i (p_{i+}(n)) \\ \hat{\lambda}^+_{\max}(n) &= \frac{x_z + x_{z+1}}{2} \end{aligned} \tag{6}$$

since the maximum value refers to a pair that LM transits the most, we take the middle point of pair as $\hat{\lambda}^+_{\max}(n)$. For non-unique z, the last visited pair with the max probability value is chosen.

(2) The expected value of the $X_+(n)$ at step n is used in the next estimation method:

$$\hat{\lambda}^+_{\exp}(n) = \sum_{i=0}^{N} \bar{x}_{i+} p_i(n) \tag{7}$$

where $\bar{x}_{i+} = \dfrac{x_i + x_{i+1}}{2}$.

(3) The median is used in the third estimation which is described as follows:

$$\begin{aligned} &\hat{\lambda}^+_{\text{med}}(n) = \bar{x}_{z+} \text{ where } z \text{ is the index satisfying} \\ &\sum_{i=0}^{z} p_{i+}(n) \geq 0.5 \text{ and } \sum_{i=z}^{N} p_{i+}(n) \geq 0.5 \end{aligned} \tag{8}$$

and where $\bar{x}_{z+} = \dfrac{x_z + x_{z+1}}{2}$

3.3 A theoretical comparison of the two estimation strategies

The two multinomially distributed random variables $X(n)$ and $X_+(n)$ and their probability vector $S(n)$ and $S_+(n)$ are clearly related. In this part we will compare them and show why the second strategy is more efficient.

Assume that $x_z < \lambda^*(n) < x_{z+1}$. For an index i below z we have a transition to the right with probability p and a transition to the left with probability $1-p$ and opposite for $i > z$. For $z = i$, we will have a probability p in both directions since the transition will result in crossing the value $\lambda^*(n)$. Consequently the probabilities are related as follows

- for $i < z$:
  $s_{i+}(n) = ps_i(n) + (1-p)s_{i+1}(n)$
- for $i = z$:
  $s_{i+}(n) = ps_z(n) + ps_{z+1}(n)$
- for $i > z$:
  $s_{i+}(n) = (1-p)s_i(n) + ps_{i+1}(n)$

We expect that $[s_0(n), \cdots, s_z(n)]^T$ is increasing and $[s_{z+1}(n), \cdots, s_N(n)]^T$ is decreasing. As a result $s_z(n)$ and $s_{z+1}(n)$ take the maximum value. We observe that the distribution of transitions yields higher probabilities around the true z ($s_{z+}(n)$) since $p > 0.5$. More specifically, for the true z we multiply with $p > 0.5$ both terms ($ps_z(n) + ps_{z+1}(n)$), and not for any other value of z. This gives a sharper peak of the distribution $S_+(n)$ around z, compared to the distribution $S(n)$. Therefore we expect that the strategies in Section 3.2 will perform better than the strategies in Section 3.1.

4 EXPERIMENTAL RESULTS

In this paper we assume that $\lambda^*(n)$ varies over time n. In such a setting, it is hard (or impossible) to deduce asymptotic proofs. Therefore we resort to simulation experiments to evaluate the performance of the estimators suggested in this paper.

Two types of comparisons are presented to show the advantages of the new strategies. The first comparison type, depicted in Fig. 1

and Fig. 2, is to illustrate that our estimations are smoother (in a single run) than random walk itself which Oommen considered as the estimate of $\lambda^*(n)$. To do so, we first define $\lambda^*(n)$ to be constant for a period of time and then switches to another random number for the same period of time and so on. In Fig. 1 the number of steps is set to $T = 10000$ and $\lambda^*(n)$ changes each 2000 steps. The probability of receiving correct guidance from environment is 0.7 while $\alpha = 0.97$. For the sake of brevity, we just show the second strategy for estimations, i.e. the one with transitions since the other one yields similar results.[1] As shown in the figure, $\lambda(n)$ exhibits a zigzag behavior and the rest of the curve displays more stable behaviour. Next, we define $\lambda^*(n)$ to be a sine function that changes

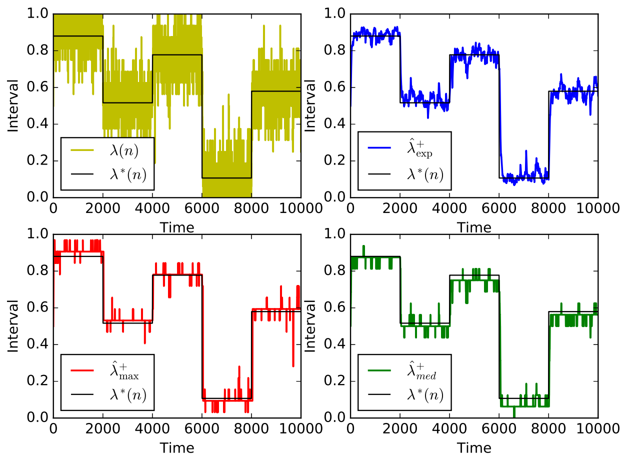

Figure 1: This figures illustrates the case where the resolution $N = 16$, environment effectiveness $p = 0.7$ and $\lambda^*(n)$ gets a random value in the $[0, 1]$ which changed after each 2000 steps. Time is set to $T = 10000$ and $\alpha = 0.97$. Each sub-figure shows how the estimate tracks $\lambda^*(n)$.

continuously between 0.7 and 0.9. In Fig. 2 the number of steps is set to $T = 10000$ and $\lambda^*(n) = 0.8 + 0.1 \sin((n/10)^\circ)$ where the sine argument is in degree. $\alpha = 0.97$ and the probability of receiving correct direction from environment is 0.7. Again $\lambda(n)$ moves in a zigzag manner and the rest of the curve is smoother.

The second comparison is based on the measured error when resolution and environment effectiveness are fixed to $N = 16$ and $p = 0.7$ respectively. To measure the estimation error we use the mean absolute error (MAE) which simply is defined by:

$$\text{MAE} = \frac{1}{T} \sum_{n=1}^{T} \mid \hat{\lambda}(n) - \lambda^*(n) \mid$$

where T is the total time steps and $\hat{\lambda}(n)$ is the estimated value at time step n.

Again we consider two different scenarios for the evolution of $\lambda^*(n)$. The first one is represented in Fig. 6 and illustrates the case of discrete changes, i.e. after each 2000 time steps the point location is

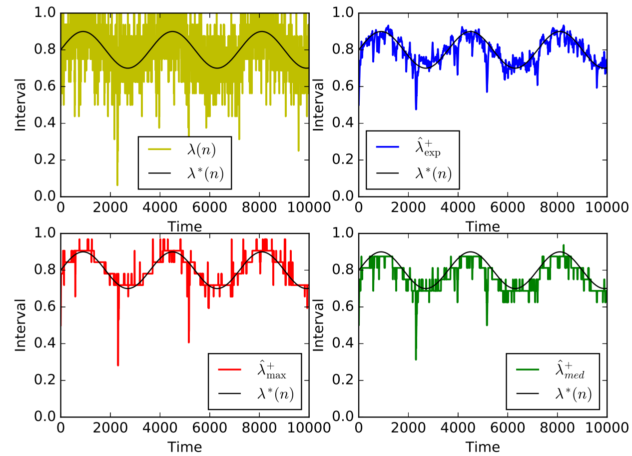

Figure 2: This figures illustrates the case where the resolution resolution $N = 16$, environment effectiveness $p = 0.7$ and $\lambda^*(n) = 0.8 + 0.1 \sin(n^\circ)$. Time is set to $T = 400$ and $\alpha = 0.97$. Each sub-figure shows how the estimate tracks $\lambda^*(n)$.

changed randomly. The comparison for errors is depicted in Fig. 5 where an average over 1000 runs is taken for MAE errors. The best estimate is achieved for α near 0.97 and the second strategy, as we expected, performs slightly more efficiently. However, interestingly, all the 6 different estimates for $0.7 \leq \alpha \leq 0.99$ perform better than $\lambda(n)$. The reduced percent error[2] between Oommen and our strategies in this simulation is up to $\frac{0.073-0.027}{0.073} \times 100 = 63\%$.

The second result which is represented in Fig. 6 concerns the case of continuous changes where $\lambda^*(n) = 0.8 + 0.1 \sin((n/10)^\circ)$. The comparison for errors is depicted in Fig. 5 where an average of 1000 runs is taken for MAE errors. The best estimate is achieved at α near 0.98. Similar to the Fig. 3, all the 6 different estimates for $0.7 \leq \alpha \leq 0.99$ perform better than $\lambda(n)$ and the estimates for the second strategy performs slightly better. The reduced percent error is equal to $\frac{0.0175-0.071}{0.0175} \times 100 = 75\%$.

5 CONCLUSION

A wide range of real-life problems can be modeled as an instantiation of the SPL problem, particularly when the environment is considered to be non-stationary. The random walk based solution due to Oommen [7] converges theoretically to a value arbitrarily close to the point location whenever both the resolution N, and the time n go to infinity. However, an arbitrarily large resolution is not realistic in real life scenarios. To get more reliable results under limited resolution we follow the same principles as Oommen method- which discritizes the interval and performs a controlled random walk- and rather take a step forward by presenting alternative estimation methods. Our schemes rely on defining two multinomially distributed random variables, then tracking their

[1] The second strategy slightly outperforms the first strategy and both of them outperform the Oommen strategy.

[2] The calculation formula for this relative error is $\%\text{Error} = \frac{\text{Actual change}}{x_{\text{reference}}} = \frac{\text{Error}_{\lambda(n)} - \text{Error}_{\hat{\lambda}(n)}}{\text{Error}_{\lambda(n)}} \times 100$.

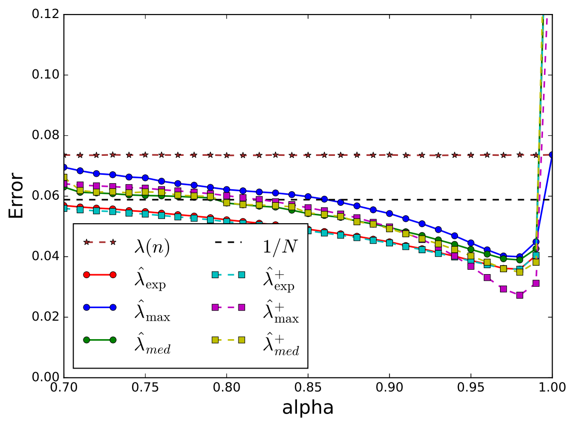

Figure 3: This figures illustrates the case where the resolution $N = 16$, environment effectiveness $p = 0.7$ and $\lambda^*(n)$ that gets a random value in the $[0, 1]$ which changed after each 2000 steps. Time is set to $T = 10000$, $0.7 \leq \alpha \leq 1$. Average is taken over 1000 runs and error measure is the MAE.

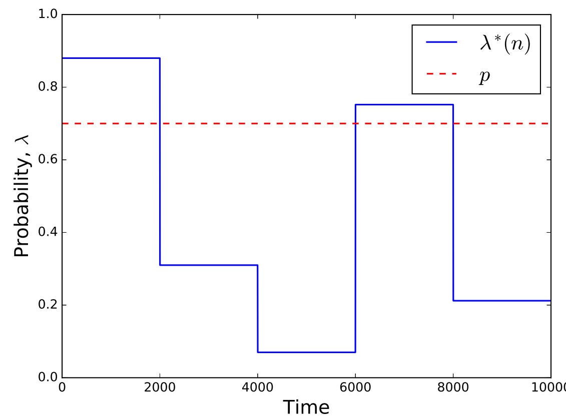

Figure 4: This figure shows the evolution of $\lambda^*(n)$ and, environment effectiveness $p = 0.7$ corresponding to Fig. 3

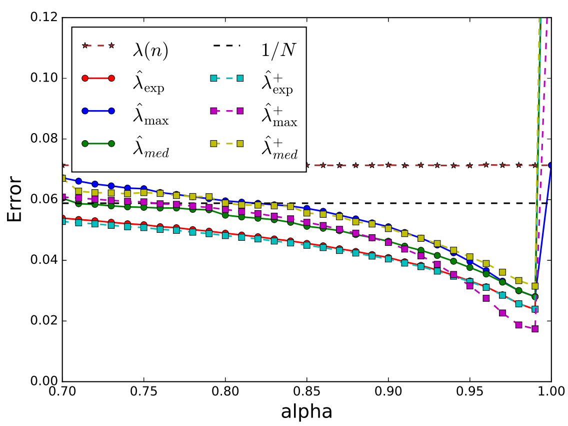

Figure 5: This figures illustrates the case where the resolution $N = 16$, environment effectiveness $p = 0.7$ and $\lambda^*(n) = 0.8 + 0.1\sin((n/10)^\circ)$. Time is set to $T = 10000$, $0.7 \leq \alpha \leq 1$. Average is taken over 1000 runs and error measure is the MAE.

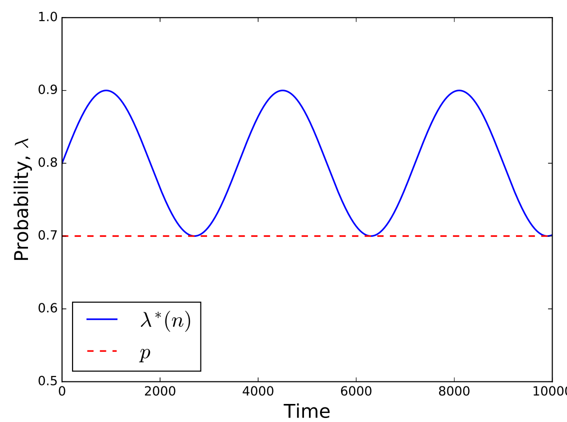

Figure 6: This figure shows the evolution of $\lambda^*(n)$ and, environment effectiveness $p = 0.7$ corresponding to Fig. 3

probability vectors using the SLWE method, and finally devising differed estimation methods involving the concepts of maximum, expectation, and median. The results indicate that these methods are smoother than the random walk itself and can track the changes more efficiently.

The second strategy which uses pairs as the events of multinomially distributed random variable is slightly better than the first one. In the simulation part, we considered environment effectiveness fixed $p = 0.7$ and the resolution $N = 16$. Interestingly, the estimated error was reduced up to 75%.

REFERENCES

[1] Cameron J Camp, Jean W Foss, Ann M O'Hanlon, and Alan B Stevens. 1996. Memory interventions for persons with dementia. *Applied Cognitive Psychology* 10, 3 (1996), 193–210.

[2] Cameron J Camp, G Gilmore, and P Whitehouse. 1989. Facilitation of new learning in Alzheimer's disease. *Memory, aging, and dementia: Theory, assessment, and treatment* (1989), 212–225.

[3] M Anwar Hossain, Jorge Parra, Pradeep K Atrey, and Abdulmotaleb El Saddik. 2009. A framework for human-centered provisioning of ambient media services. *Multimedia Tools and Applications* 44, 3 (2009), 407–431.

[4] De-Shuang Huang and Wen Jiang. 2012. A general CPL-AdS methodology for fixing dynamic parameters in dual environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 42, 5 (2012), 1489–1500.

- [5] Yeong Min Jang. 2000. Estimation and prediction-based connection admission control in broadband satellite systems. *ETRI journal* 22, 4 (2000), 40–50.
- [6] EE Kpamegan and N Flournoy. 2008. Up-and-down designs for selecting the dose with maximum success probability. *Sequential Analysis* 27, 1 (2008), 78–96.
- [7] B John Oommen. 1997. Stochastic searching on the line and its applications to parameter learning in nonlinear optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 27, 4 (1997), 733–739.
- [8] B John Oommen and Govindachari Raghunath. 1998. Automata learning and intelligent tertiary searching for stochastic point location. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 28, 6 (1998), 947–954.
- [9] B John Oommen, Govindachari Raghunath, and Benjamin Kuipers. 2006. Parameter learning from stochastic teachers and stochastic compulsive liars. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 36, 4 (2006), 820–834.
- [10] B John Oommen and Luis Rueda. 2006. Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments. *Pattern Recognition* 39, 3 (2006), 328–341.
- [11] Anis Yazidi, Ole-Christoffer Granmo, B John Oommen, and Morten Goodwin. 2014. A novel strategy for solving the stochastic point location problem using a hierarchical searching scheme. *IEEE transactions on cybernetics* 44, 11 (2014), 2202–2220.