ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Controlling Adaptive Quantum-Phase Estimation with Scalable Reinforcement Learning

Pantita Palittapongarnpim[1], Peter Wittek[2,3] and Barry C. Sanders[1,4] *

1- Institute for Quantum Science and Technology, University of Calgary
Calgary, Alberta T2N 1N4 Canada

2- ICFO-The Institute of Photonic Sciences
Castelldefels (Barcelona), 08860 Spain

3- University of Borås
Borås, 501 90 Sweden

4- Program in Quantum Information Science
Canadian Institute for Advanced Research
Toronto, Ontario M5G 1Z8 Canada

Abstract. We develop a reinforcement-learning algorithm to construct a feedback policy that delivers quantum-enhanced interferometric-phase estimation up to 100 photons in a noisy environment. We ensure scalability of the calculations by distributing the workload in a cluster and by vectorizing time-critical operations. We also improve running time by introducing accept-reject criteria to terminate calculation when a successful result is reached. Furthermore, we make the learning algorithm robust to noise by fine-tuning how the objective function is evaluated. The results show the importance and relevance of well-designed classical machine learning algorithms in quantum physics problems.

1 Introduction

Quantum-enhanced metrology aims to estimate an unknown parameter ϕ with the goal of overcoming the standard quantum limit (SQL), which is $\Delta\phi \propto N^{-0.5}$ [1] where N is the number of particles used in the measurement. Quantum states can be used to achieve power-law scaling better than -0.5. In particular, we consider adaptive quantum-enhanced metrology, which performs sequential measurements on a sequence of N single-particle pulses. After each measurement, an automatic system adjusts a controllable parameter Φ, affecting the outcome of the subsequent measurements. The use of multi-particle entanglement allows the measurement to overcome the independent and identical distribution condition and consequently the SQL. The advantage of using this approach is that the technology for single particle detection is readily available and so make the approach feasible to implement. The challenge of adaptive quantum-enhanced metrology is in generating a feedback policy that can achieve the enhanced precision. To this end, we develop a learning algorithm that can be applied to problems that include noise present in physical systems.

*This work is financially supported by NSERC and AITF.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

Reinforcement learning is gaining attention in quantum physics problems. Examples include an agent-based model in measurement-based quantum computation [2], mapping quantum gates on a spin system [3], optimization in ultracold-atom experiments [4], and earlier work on the adaptive quantum phase estimation problem using heuristic optimization [5, 6]. These previous works showed promising results, especially when relying on differential evolution (DE), but DE is unable to deliver a successful policy when noise is included. Another algorithm using particle swarm optimization (PSO) succeeds under noisy conditions but breaks down in precision scaling after 45 particles even in the absence of noise.

The DE-based algorithm also exhibits this breakdown but at over 90 particles, which means that DE-based algorithm outperforms PSO by a factor of two with respect to the number of particles. Building on the DE-based reinforcement-learning algorithm, we advance the state of the art in two ways: (1) We develop a scheme that can operate when practical imperfections such as noise and loss are included. The primary means to this is the way in which the objective function is evaluated. (2) We improve scalability to a higher number of particles. This is achieved by accept-reject criteria that allow an early or a late termination of calculations. Furthermore, we vectorize the time-critical operations to efficiently use the parallel resources available in contemporary CPUs and GPUs.

2 Generating feedback policy as a learning problem

We consider the problem of optical interferometric-phase estimation, which is well-studied due to its connection to many problems in physics, such as the gravitational wave detection [7, 8]. The interferometer has two input and two output modes. The N-photon entangled state is injected into the interferometer one photon at a time. Neglecting loss, the photon comes out from either of the output modes with a probability that depends on $\phi - \Phi$. Our interferometer model allows Gaussian noise on the phase shift. We label the outcome by $u \in \{0, 1\}$, where 0 refers to the photon exiting the first port and 1 to the photon exiting the second port. The sequence of outcomes from the first to the m^{th} photon is given by $\boldsymbol{u}_m$. The path from which a photon exits provides information to determine the adjustment of Φ for the next round of measurement. Once all photons are expended, the estimate of ϕ is inferred from Φ.

We now explain how to construct the feedback policy, which is a set of rules that determines how Φ is adjusted. In the m^{th} round of measurement, the policy is a function of the sequence of previous outcomes $\boldsymbol{u}_{m-1} \in \{0, 1\}^{m-1}$. This is better understood by representing a policy as a binary decision tree. An advantage of using this representation is that the size of a policy is readily determined and its size scales as $2^N - 1$. To make searching for the policy tractable, we impose a rule that $\Phi_m = \Phi_{m-1} + (-1)^{u_m} \Delta_m$, thereby reducing the policy from a full decision tree to a vector $\varrho = (\Delta_1, \ldots, \Delta_N)$. The search space becomes an N-dimensional, real, continuous space bounded in each dimension by $[0, 2\pi)$.

As the measurement outcomes $\boldsymbol{u}_m$ is a string of discrete random variables,

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

the estimate of phase from this scheme is also discrete. The increase in N allows us to obtain increasingly precise estimate. We determine the imprecision of the measurement using the Holevo variance $V_{\mathrm{H}}(\theta) = S(\theta)^{-2} - 1$, where the distribution sharpness function $S(\theta) = \left| \sum_{k=1}^{K} \frac{\mathrm{e}^{\mathrm{i}\theta_k}}{K} \right|$ and $\theta = \phi - \Phi$. The number $K = 10N^2$ indicates the size of the sample set, a set which is also used as training data for the learning algorithm. We uniformly sample the value of ϕ from $[0, 2\pi)$ in the training stage to generate both ϱ and its corresponding imprecision. The sharpness function S serves as the objective function.

The photons are modelled to have a small probability η of being lost before entering the interferometer, and the automatic system is instructed to do nothing in the absence of a detection signal. We assume small loss in which case we optimize without loss but test the performance accounting for loss. If the test of the policy under lossy conditions fails, we return to the optimization algorithm to come up with a better policy and test this policy under lossy condition. If the test succeeds, we adopt the policy; if the test fail, we repeat.

3 Noise-Resistant Global Optimization Heuristics

Due to DE's ability to find successful policies in a high-dimensional search space [9], we employ DE for the problem of noisy phase estimation for N up to 100 and observe that DE does not perform as well as PSO. In fact, the algorithm fails to deliver better than SQL scaling altogether. To devise a noise-resistant global optimization algorithm for our scheme, we use the mean value $\bar{S}$ instead of S to determine the performance of a policy. This strategy is one of the many strategies proposed in the literature to create noise-resistant DE [10, 11] and is found to work best for our problem.

The principle behind the use of mean objective value is as follows: if noise is added to the fitness function, the process of averaging recovers the true objective value. The optimization using this value is therefore a close approximation to the noiseless optimization. The major drawback of this approach is that computing the objective function multiple times makes the procedure computationally expensive. Therefore, determining the smallest sample size of $\{S\}$ necessary to recover S is crucial. To this end, we employ the heuristic applied to PSO in the previous work [5]. The method updates $\bar{S}$ by computing one new sample of S every iteration until a better offspring is generated. As the candidates approaches the optimal value, the probability of generating an offspring that is better than the parent decreases, and as a result the sample size for evaluating $\bar{S}$ increases. Therefore, this method leads to a dynamic selection of sample size. The computational resources is allocated towards candidates that are close to optimal in order to obtain accurate estimations of their fitness values.

Although noise added to the phase shift is not additive in S due to the exponential dependence, exceeding SQL is possible because computing $\bar{S}$ from G samples is equal to computing a single S using sample size GK, thereby providing a better estimate of imprecision than for the sample size of K.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

4 Improving Scalability

In order for the optimization algorithm to find a solution in a search space that scales up to 100 dimensions, we implement a set of heuristics and criteria to ensure that only successful policies are accepted. Previously the algorithm accepted a policy after a fixed number of iterations regardless of whether the population converge. However, as the dimension of the search space increases, so does the time for the population to converge. Eventually, the algorithm fails to deliver a policy that passes the test.

The issue of increasing convergence time with the increase in search-space dimension manifested as the stagnation in imprecision for $N > 90$. We change the criterion for accepting a policy from a fixed number of iterations to only if V_{H} is within a distance corresponding to a confidence interval of 0.98 from the inverse power-law line. Thus, we guarantee that the policy from our algorithm always delivers a power-law scaling better than SQL.

In order to calculate the acceptable error, we collect the V_{H} values from $N = \{4, 5, dots, 93\}$ to determine the linear equation that describe the relationship between $\log V_{\mathrm{H}}(N)$ and $\log N$, which should be linear given a power-law between V_H and N. Using this linear relationship, we can predict the next data point and calculate the acceptable error using a formula from statistics, namely

$$\delta_y = t^*_{n'-2} \sqrt{\frac{\sum_{i=1}^{n'} (y'_i - y_i)^2}{n' - 2} \left(\frac{1}{n'} + \frac{(x' - \bar{x})^2}{\sum_{i=1}^{n'} (x_i - \bar{x})^2} \right)}, \tag{1}$$

where n' is the number of data points, $i = \{1, 2, \ldots, n'\}$, and y'_i is the value of $\log V_{\mathrm{H}}$ as predicted by the linear equation. Here $y_i = \log V_{\mathrm{H}}(N_i)$ and $x_i = \log N_i$ whereas x' corresponds to the value of N for which the error is calculated. The value $t^*_{n'-2}$ is the quantile on the Student's t distribution for $n' - 2$ data points, which we approximate using a normal distribution. The difference between $\log V_{\mathrm{H}}(N)$ from the simulation and the predicted data point is computed. If the difference is smaller than δ_y, we accept the policy. Otherwise, the optimization continues.

The computational complexity of the algorithm is polynomial, but it has a high degree, and therefore it is important to identify the performance critical parts of the implementation. Profiling the code, we identify that over 90% of the execution time is spent on generating random numbers, which are primarily used in calculating the Holevo variance. Generating random numbers as they are needed, one by one, is not efficient on contemporary hardware, as the operations can be vectorized to use the single-instruction multiple-data architectures of the central and the graphics processing units. Abstracting the random number generation routines and introducing a buffer, we are able to vectorize the respective operations. We study two approaches: one relies on the CPU, using the Intel Vector Statistical Library (VSL), the other on graphics processing units. Eventually the VSL-based vectorized solution prove to be more scalable.

The noise-resistant DE variant works as follows.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

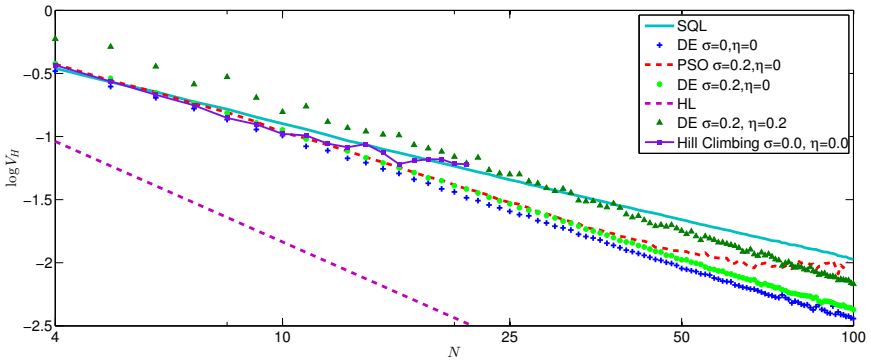

Fig. 1: Log of Holevo variance from adaptive interferometric-phase estimation.

Step 1 Initialize the population of size N_P randomly.
Step 2 Evaluate the objective function for each candidate *twice*, and store the mean objective value and the sample size.
Step 3 Generate a donor for each of candidate $V_i(t)$, where t is the iterative time step, from three other candidates $\{V_{i,1}(t), V_{i,2}(t), V_{i,3}(t)\}$ chosen randomly using the rule

$$D_i(t) = \begin{cases} V_{i,1}(t) + F(V_{i,2}(t) - V_{i,2}(t)) & \text{if rand} \leq C_r, \\ V_i(t) & \text{else}, \end{cases} \tag{2}$$

where $C_r \in [0, 1]$ is the crossover rate, and rand $\in [0, 1]$ is a random number.
Step 4 Evaluate the mean objective value for each of the new candidates from two samples.
Step 5 Compare new and the old candidate using the mean objective value

$$V_i(t+1) = \begin{cases} D_i(t) & \text{if } \bar{f}(D_i(t)) > \bar{f}(V_i(t)), \\ V_i(t) & \text{else}, \end{cases} \tag{3}$$

where $\bar{f}()$ reads the mean objective value in the memory.
Step 6 Evaluate the objecting function once, and update the mean value and the sample size.
Step 7 Repeat step 3 to 6 until the criterion to terminate the algorithm is met.
Step 8 Compute the objective value of the entire population for 10 more times before selecting the candidate with the highest mean objective value as the solution.

5 Results

The scaling of the Holevo variance is $V_H \propto N^{-1.421}$ when width of the Gaussian distribution is 0.2 rad and the probability of losing a photon is 0.2. This exceeds

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

the scaling of N^{-1} expected from SQL. Policies that are found using stochastic hill climbing breakdown at 20 photons, and noise-resistant PSO shows the breakdown at 45 photons. Using noise-resistant DE with accept-reject criteria, the scaling continues up to 100 photons (Figure 1).

The limitation at 100 photons is due to the computational time and the rounding error in the generation of large multi-particle entangled state. The time required to find a policy under accept-reject criteria from 94 to 100 photons are between 1.5 to 3 hours per data point.

6 Conclusion

We have devised a reinforcement-learning algorithm that generates feedback policies for adaptive phase estimation including noise and loss. We are able to achieve enhanced precision better than SQL up to 100 photons using noise-resistant variant of DE and accept-reject criteria. This work can be used as the basis to develop learning algorithms for solving more complex estimation problems, such as estimating more than one unknown parameters, which has an application in the characterization of quantum information processing devices.

References

[1] V. Giovannetti, S. Lloyd, and L. Maccone. Advances in quantum metrology. *Nat. Photon.*, 5(4):222–229, Mar 2011.

[2] M. Tiersch, E. J. Ganahl, and H. J. Briegel. Adaptive quantum computation in changing environments using projective simulation. *Sci. Rep.*, 5:12874, Aug 2015.

[3] L. Banchi, N. Pancotti, and S. Bose. Quantum gate learning in engineered qubit networks: Toffoli gate with always-on interactions. *arXiv:1509.04298*, Sep 2015.

[4] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. N. Kuhn, I. R. Petersen, A. Luiten, J. J. Hope, N. P. Robins, and M. R. Hush. Fast machine-learning online optimization of ultra-cold-atom experiments. *arXiv:1507.04964*, Jul 2015.

[5] A. Hentschel and B. C. Sanders. Machine learning for precise quantum measurement. *Phys. Rev. Lett.*, 104(6):063603, Feb 2010.

[6] N. B. Lovett, C. Crosnier, M. Perarnau-Llobet, and B. C. Sanders. Differential evolution for many-particle adaptive quantum metrology. *Phys. Rev. Lett.*, 110(22):220501, May 2013.

[7] C. M. Caves. Quantum-mechanical noise in an interferometer. *Phys. Rev. D*, 23(8):1693–1708, Apr 1981.

[8] B. P. Abbott et al. Observation of gravitational waves from a binary black hole merger. *Phys. Rev. Lett.*, 116(6):061102, Feb 2016.

[9] J. Vesterstrom and R. Thomsen. A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *IEEE C. Evol. Computat.*, volume 2, pages 1980 – 1987, Jun 2004.

[10] S. Das, A. Konar, and U. K. Chakraborty. Improved differential evolution algorithms for handling noisy optimization problems. In *IEEE C. Evol. Computat.*, volume 2, pages 1691–1698, Sep 2005.

[11] L. Barone, A. Di Pietro, and L. While. Applying evolutionary algorithms to problems with noisy, time-consuming fitness functions. In G. Greenwood, editor, *IEEE C. Evol. Computat.*, volume 2, pages 1254–1267, Jun 2004.