Towards An End-To-End Approach For Quantum Principal Component Analysis

(Article begins on next page)

Towards An End-To-End Approach For Quantum Principal Component Analysis

Emanuele Dri*, Antonello Aita†, Tommaso Fioravanti†, Giulia Franco†, Edoardo Giusto*, Giacomo Ranieri‡, Davide Corbelletto‡, and Bartolomeo Montrucchio*

*DAUIN, Politecnico di Torino, Torino, Italy
emanuele.dri@polito.it, edoardo.giusto@polito.it, bartolomeo.montrucchio@polito.it
†IBM Italia, IBM, Milano, Italy
antonello.aita@ibm.com, tommaso.fioravanti@ibm.com, giulia.franco@ibm.com
‡ Intesa Sanpaolo, Torino, Italy
giacomo.ranieri@intesasanpaolo.com, davide.corbelletto@intesasanpaolo.com

***Abstract*—Quantum Machine Learning has gained significant attention in recent years as a way to leverage the relationship between quantum information and machine learning. Principal Component Analysis (PCA) is a fundamental technique in machine learning, and the potential for its quantum acceleration has been extensively studied. However, an algorithmic end-to-end implementation remains challenging. This paper covers quantum PCA implementation up to extracting the principal components. We extend existing processes for quantum state tomography to extract the eigenvectors from the output state, addressing the challenges of dealing with complex amplitudes in the case of non-integer eigenvalues. Finally, we apply our implementation to a practical quantum finance use case related to interest rate risk, and present the results of our experiments.**

***Index Terms*—quantum computing, quantum machine learning, principal component analysis, quantum tomography, interest rate risk, quantum finance**

I. Introduction

Quantum Machine Learning (QML) is a broad definition encompassing many techniques that have garnered significant attention within the scientific community in the last decade [1]–[5]. Two distinct outlooks emerge from the literature: near-term QML, which replaces a machine learning model with a generic quantum computation, and a fault-tolerant machine perspective that evaluates algorithms based on asymptotic computational complexity. The latter focuses on improving existing methods (rather than developing new ones) by exploring how a quantum computer could perform the same computation asymptotically faster than classical machine learning techniques [6].

A popular strategy in this context applies quantum routines for solving linear algebraic problems. Many algorithms that involve inverting matrices or finding eigenvalues and eigenvectors can be reduced to these problems. The underlying principle of the linear algebra approach in QML is to encode the matrix into a quantum state and apply various methods to process it efficiently [6].

Reference [7] was the first to propose a quantum algorithm for Principal Component Analysis (PCA) by combining some of the most promising techniques in this field. In particular, by employing density matrix exponentiation and quantum matrix inversion [8], the authors derive a quantum PCA approach that, applied to an unknown low-rank density matrix, reveals the matrix's eigenvectors and eigenvalues in quantum form, providing an exponential advantage. A variation of this algorithm was later employed in [9] to propose the application of quantum PCA in pricing financial derivatives. Moreover, proposals based on the Quantum Singular Value Thresholding (QSVT) algorithm [10] have emerged in recent years [11], [12].

Despite these progresses, the quantum PCA algorithm still misses an end-to-end implementation. One of the challenges resides in building a generalized algorithmic approach for the quantum state tomography process used to extract multiple eigenvectors, especially in the presence of complex amplitude states. Additionally, some of the most efficient implementations of quantum PCA cannot deal with matrices with non-integer eigenvalues [12]. To advance the field, a coherent and general code implementation that deals with these shortcomings, is needed.

In this paper, we briefly introduce the main concepts regarding Quantum PCA in Section II, then in Section III we present a detailed implementation based on the open-source Qiskit SDK [13] that can handle matrices with arbitrary eigenvalues. Our implementation includes the necessary state tomography process for extracting multiple eigenvectors from the output state, extending and adapting the approach presented in [14]. We provide a detailed description of this process, addressing the challenges of dealing with complex amplitude states. Section IV contains the complexity analysis for the resulting methodology. Finally, in Section V we apply our implementation to a practical and relevant quantum finance use case [15] and summarize our conclusions in Section VI.

II. Quantum Principal Component Analysis

PCA is a multivariate statistical technique for data reduction and pattern recognition in complex datasets. The method is

979-8-3503-4323-6/23/$31.oo ©2023 IEEE
DOI 10.1109/QCE57702.2023.10175

based on a linear transformation of the original data into a new coordinate system, where the variables are uncorrelated and sorted in order of importance, allowing for dimensionality reduction while preserving most of the variability of the original data [16]. The transformed variables called *principal components* are calculated as linear combinations of the original ones. PCA is widely used in various fields, such as image processing, finance, and biology. In order to perform PCA, it is necessary to compute the eigenvalues and eigenvectors of the input matrix containing the original dataset. The eigenvectors represent the directions of maximum variance in the data, while the eigenvalues represent the variance each eigenvector explains. This process can be computationally intensive, particularly for large datasets.

In this regard, a quantum algorithm with exponential advantage for PCA of non-sparse density matrices was proposed [7] and later extended to non-Hermitian and non-square matrices via embedding matrices [17]. We now proceed to illustrate its main steps.

A. Data Loading

First, it is necessary to prepare the initial quantum state through amplitude encoding of the input matrix $A \in \mathbb{R}^{N\times N}$ with $\mathrm{tr}\,[A] = 1$.

$$|\psi_A\rangle = \sum_{i=1}^{N}\sum_{j=1}^{N} A_{ij}|i\rangle|j\rangle = \sum_{k=1}^{r} \sigma_k\,|u_k\rangle\,|u_k\rangle \tag{1}$$

where r is the rank of the matrix, σ_k are the singular values and u_k are the singular vectors of A. The second equality is true due to the Gram-Schmidt decomposition. Note that if $\mathrm{tr}\,[A] \neq 1$, it is sufficient to add a classical preprocessing step and normalize the matrix with respect to its trace and then, postprocess the output multiplying the obtained eigenvalues by the trace itself.

B. Unitary Transformation

A core operation is the efficient construction of the unitary transformation e^{-iAt}, fundamental for phase estimation. In this regard, there are proposals for an efficient preparation that works with low-rank matrices [7], [17]. It assumes the ability to access multiple copies of A, which is true in our case. Under this assumption, the technique allows to efficiently generate e^{-iAt} with accuracy ϵ in $\mathcal{O}\left(t^2\epsilon^{-1}\right)$ steps.

C. Phase Estimation

At this stage, it is possible to apply the quantum phase estimation algorithm [18] to get the eigenvalues and, subsequently, the eigenvectors of the input matrix. To determine the eigenvalues with an n-bit precision, we will need an n-qubit counting register E.

$$|0\rangle^E\,|\psi_A\rangle^M \xrightarrow{U_{\mathrm{PE}}} \sum_{k=1}^{r} \sigma_k\,|\lambda_k\rangle^E\,|u_k\rangle\,|u_k\rangle \tag{2}$$

Figure 1 illustrates the corresponding quantum circuit and the steps of the phase estimation algorithm [19].

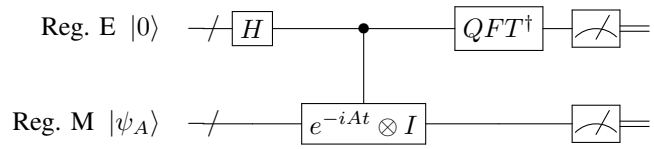

Fig. 1. The circuit that performs phase estimation [19] to produce the state $\sum_{k=1}^{r}\sigma_k\,|\lambda_k\rangle^E\,|u_k\rangle\,|u_k\rangle$.

At this point, what we obtain in the output state is a superposition of the eigenvalues and eigenvectors in the E and M registers, respectively. State tomography is then required to extract the principal components.

D. Quantum State Tomography

Quantum state tomography is a technique used to determine the quantum state of a system [19].

There is a lack of clear and general approaches in the literature for quantum PCA that address the issue of principal component reconstruction. To fill this gap, we start from the tomography proposed in [14] that illustrates a working methodology for the reconstruction of pure states with real amplitudes when we have an efficient unitary to prepare the states. However, if the eigenvalues of the matrix are not integers, an exact representation is impossible. Thus, the tomography must be extended to fit the new constraints.

III. Methodology

In the following Sections, we will illustrate the main contributions of our work.

A. Extended Quantum Tomography Approach

Here, we describe our general version of the algorithm proposed in [14] that can handle states with complex amplitudes.

The extended algorithm includes two main steps: *Probability Estimation* (1) and *Sign Estimation* (2). Step (1) provides an estimate of the probability p_i of each basis state composing the output quantum state. Then, step (2) provides the sign for each amplitude of probability value regardless of the nature of the statevector (real or complex). This approximation works fine because, even in the case of complex amplitudes (due to an approximate representation of non-integer eigenvalues), $\sqrt{p_i}$ is the modulus of the amplitude ($|z| = \sqrt{Re(z)^2 + Im(z)^2}$). In this case, we have the contribution of the imaginary part that could, in principle, affect the reconstruction of the principal components. However, if the resolution, intended as the size of the output registers E (r_{PE} qubits), is sufficiently high, the imaginary component will be sufficiently small (with respect to the real one) to be safely approximated to zero.

Probability Estimation: This step is essentially similar to that described in the original implementation [14]: for a unit vector $x \in \mathbb{R}^d$, N_c copies of the state $|x\rangle$ that we want to reconstruct are needed (with $N_c = \frac{36d\ln d}{\delta^2}$ dependent on the length of the statevector and on δ, where $\|\widetilde{x} - x\|_2 < \sqrt{7}\delta$ with high probability [14]).

In our case, we use the circuit resulting from the encoding part of the input matrix and the PE operator and measure its

registers (whose state will represent $|x\rangle$) N_c times. In this way, we obtain the estimates of the absolute value of the probability amplitudes using a frequentist approach, namely as $\sqrt{p_i} = \sqrt{count_i/N_c}$, where $count_i$ corresponds to the number of times we observe the ith outcome,

Sign Estimation: To estimate the signs, it is necessary to use a control qubit to create N_c copies of the state

$$\frac{1}{\sqrt{2}}|0\rangle \sum_{i\in[d]} x_i|i\rangle + \frac{1}{\sqrt{2}}|1\rangle \sum_{i\in[d]} \sqrt{p_i}|i\rangle \tag{3}$$

To obtain this state, we need two unitary operators, U_x and U_p, that, when applied to a target qubit $|0\rangle$, allow us to obtain $U_x|0\rangle = |x\rangle$ and $U_p|0\rangle = |p\rangle$ (with $|p\rangle = \sum_{i\in[d]} \sqrt{p_i}\,|i\rangle$). Starting with a circuit with t target qubits, we add a control qubit and apply an H gate to put it in superposition. We now have the state $|+\rangle\,|0\rangle_t$. Then, conditioned on the control qubit, we apply operator U_p or U_x to the target registers $|0\rangle_t$.

Then, we apply an additional H gate to the control qubit to obtain the following state:

$$\frac{1}{2}\sum_{i\in[d]} \left[(x_i + \sqrt{p_i})\,|0,i\rangle + (x_i - \sqrt{p_i})\,|1,i\rangle\right] \tag{4}$$

with the sums and differences on the target qubits (sums if the control qubit assumes value 0, differences if it assumes value 1). This construction recalls the well-known concept of the *Hadamard test* [6].

Finally, we measure all qubits, including the control one. we only consider counts relative to the control qubit being 0 and check for each ith state whether the number of counts is greater than $0.4 \cdot p_i \cdot N_c$. Based on the result, we assign a positive (if greater) or negative sign to the corresponding value of $\sqrt{p_i}$.

B. Principal Components Reconstruction

The next step is to focus on the qubits related to the eigenvalues. Thus, we take the last r_{PE} qubits and aggregate the various p_i for each eigenvalue. Then, we analyze the resulting aggregated values to find the k biggest local maxima (where k is the number of eigenvalues of the matrix). Those are the estimated eigenvalues.

Moreover, we obtained the input circuit's reconstructed statevector. From register M, we can now get the eigenvectors (factoring out the opportune coefficients) since they are encoded in the register's output distribution, as reported in Equation 2.

C. Qiskit Implementation

All the code related to the implementation of the proposed methodologies and their respective experiments is available in a dedicated repository [20].

1) Initialization: The ideal approach would involve a qRAM that loads the input matrix, as suggested in [11], [12], [17]. Despite continuous progress [21]–[23], this protocol still needs a more straightforward and efficient implementation [24], out of the scope of this work. Our initialization protocol is instead based on the method illustrated in [25], which is the basis of the *StatePreparation* class in Qiskit [26]. Therefore, the second approach serves as a placeholder for experiments on a reduced scale. Currently, the scientific community is actively working on alternatives with potentially non-exponential scaling for data loading [27], [28] and on techniques for significant reduction of circuit depth [29], which are showing progress in overcoming the limitations related to efficient scaling. In future works, the authors intend to select the most promising emerging techniques and test them on the current implementation.

2) Probability estimation: We implement the phase estimation using the dedicated *PhaseEstimation* class in Qiskit [30]. We provide as parameters the number of qubits we will use to encode the eigenvalues (which will determine the resolution of the estimated values) and the unitary transformation e^{-iAt}. Regarding the latter, the authors emphasize that they are unaware of any implementation in Qiskit (or other SDKs) of the method proposed in [7] and [17]. Therefore, in terms of code, the operator is precomputed, similarly to what happens for existing code implementations of analogous algorithms [31].

3) Sign estimation: In regard to sign estimation, we leverage the properties of the *QuantumCircuit* class [32] to generate a controlled operator with the circuit used for absolute value estimation, which corresponds to U_x. Conversely, we reuse the *StatePreparation* class to load the $\sqrt{p_i}$ and then create the controlled operator U_p. The final step is to add a control qubit and apply these two operators to create the circuit shown in Figure 2.

IV. Complexity Analysis

Here, we will provide details on both space and time complexity for each step of the Quantum PCA algorithm.

A. Space Complexity

1) Initialization: To encode the elements of an $N \times N$ matrix, we will need N^2 configurations. Therefore, the initialization step will require a register M with $O(\log(N^2))$ qubits.

2) Phase Estimation: As mentioned in Section II-C, using a register E with n qubits we can determine the eigenvalues with a resolution of $\frac{1}{2^n}$. Again, when performing phase estimation on non-integer quantities, choosing an appropriate resolution for the number of qubits is crucial. If the resolution is not high enough, the imaginary part of the amplitudes may not be negligible, leading to incorrect estimations. Therefore, the appropriate choice of resolution is critical for the success of phase estimation.

3) Sign Tomography: As can be evinced from Figure 2, this step will require a register with a number of qubits equal to the sum of those needed for the previous two steps ($O(\log(N^2) + n)$) plus an additional control qubit.

B. Time Complexity

Here, the focus is on the number of operations required to execute the quantum PCA algorithm.

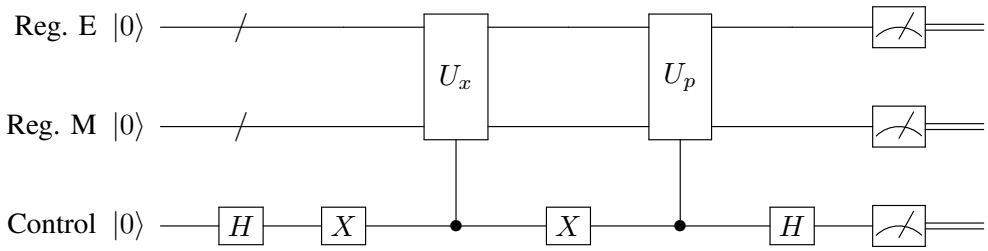

Fig. 2. The circuit that performs sign estimation using the controlled operators U_x and U_p plus a control qubit adequately prepared.

1) Initialization: Previous works assumed that matrix elements are stored in M via qRAM, which presents a time to run the circuit proportional to $O(\log(N^2))$ where $A \in \mathbb{R}^{N\times N}$. Our Qiskit implementation makes use of the StatePreparation class, whose peculiarities we discussed in Section III-C.

2) Unitary Transformation: Using the methodology in [7], [17], if A is low-rank, e^{-iAt} can be simulated using $O(t^2\epsilon^{-1})$ copies of A, where ϵ is the error of the Hamiltonian simulation. Thus, we need $T = O(t^2\epsilon^{-1})$ copies of A to implement e^{-iAt} to accuracy ϵ in time $O(t^2\epsilon^{-1}\log(N^2))$.

3) Phase Estimation: The complexity of the Quantum Fourier Transform (QFT) needed for phase estimation is $O(n^2)$ or $O(n\log(n))$, which is significantly lower in comparison to the Hamiltonian simulation. Therefore, the time complexity for obtaining U_{PE} is dominated by the contribution described above: $O(t^2\epsilon^{-1}\log(N^2))$ an not by QFT.

4) Tomography: U_x will share the same complexity as the base quantum PCA algorithm, again dominated by the Hamiltonian simulation. For U_p, we can use the same approach we used in *B.1* to encode the estimated amplitudes on a total of $\log(N^2) + n$ qubits. As mentioned in Section III-A, we need $N_c = \frac{36d\ln d}{\delta^2}$ copies of the sign estimation circuits. Thus, the overall complexity for this step will be $O(\frac{36d\ln d}{\delta^2}t^2\epsilon^{-1}\log(N^2))$.

V. Experiments

The authors deemed it valuable to propose a practical use case for the experiments conducted. For this reason, we use quantum PCA for a well-known process in the financial domain: generating the evolution of interest rate curves.

TABLE I
Means and Standard Deviations for Asset Returns

Asset	Mean	Standard Deviation
Asset A	0.05	0.10
Asset B	0.03	0.08

A. Interest Rate Risk

Quantum finance is an emerging field that explores the intersection between quantum computing and financial applications, intending to develop more efficient and accurate financial models and algorithms [33]. Recently, there has been an increasing interest in applying quantum algorithms to fields of finance such as portfolio optimization [34], [35], option pricing [36], [37], and risk management [38], [39].

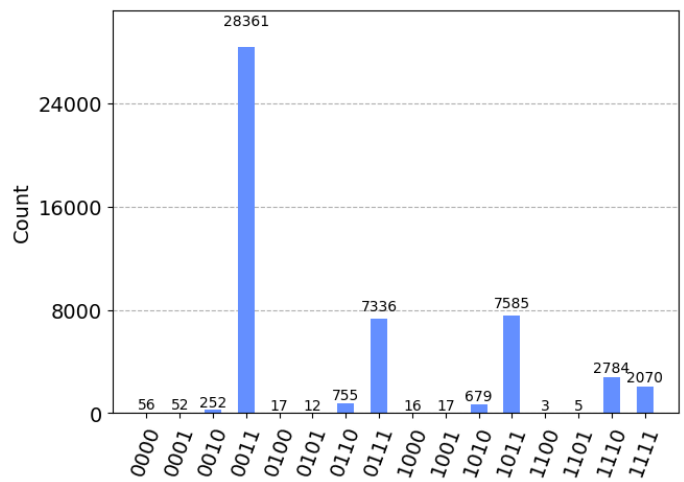

Fig. 3. Counts for the phase estimation circuit executed using the IBM ideal simulator.

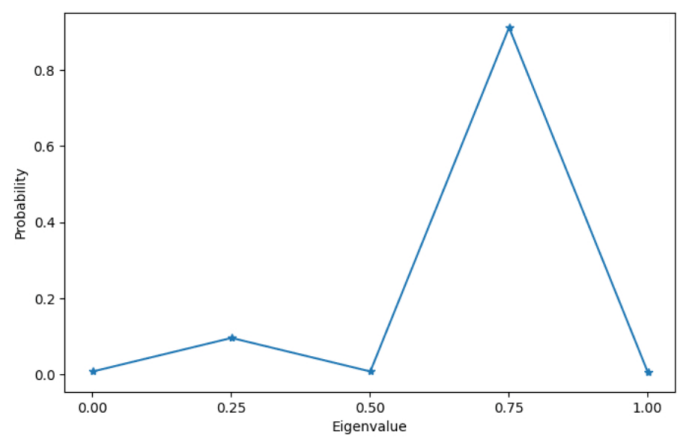

Fig. 4. Aggregated probability (resulting from the counts in Figure 3) of observing each possible eigenvalue.

Another critical concern for financial institutions is the management of interest rate risk. In this context, a typical use case consists of taking the daily computed differences among tenors[1] used to describe interest rate curves and use PCA to reduce the variables and reproduce the same features of the initial dataset. Essentially, an Autoregressive Moving

[1]Length of time until a financial instrument, such as a bond or a loan, reaches maturity or is due to be repaid.

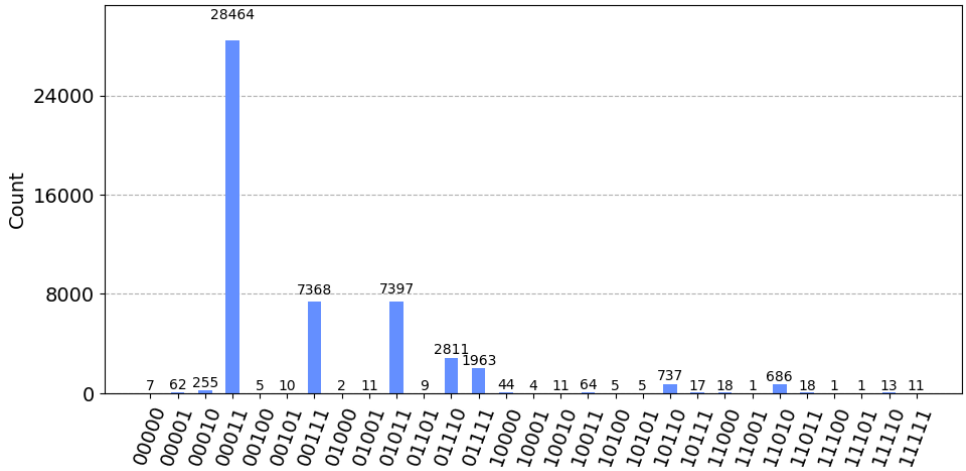

Fig. 5. Counts for the sign estimation circuit executed using the IBM ideal simulator.

Average (ARMA) model [40] is built for each interest rate tenor, cleaned from outliers, and the principal component analysis is applied to residuals to focus on modeling volatility movements.

For our purpose, we generated a matrix of appropriate dimensions (2×2) with synthetic data validated by domain experts who confirmed their plausibility.

$$A = \begin{bmatrix} 0.6507 & 0.2122 \\ 0.2122 & 0.3493 \end{bmatrix}$$

$$\lambda_1 = 0.760 \quad u_1 = \begin{bmatrix} 0.889 & 0.459 \end{bmatrix}$$

$$\lambda_2 = 0.240 \quad u_2 = \begin{bmatrix} -0.459 & 0.889 \end{bmatrix}$$

In this example, the diagonal entries represent the variances of each asset's returns, while the off-diagonal entries represent the covariance between hypothetical asset A's and asset B's returns. To generate this synthetic data, we simulated 1000 scenarios of returns for each asset over a one-year period and then calculated the covariance matrix based on these simulated returns.

B. Results

Ideal simulations were performed using the *ibmq qasm* simulator [41]. The parameters used were $r_{PE} = 2$ (so the eigenvalues will be estimated with a resolution of $\frac{1}{2^2}$) and $N_c = 50000$ shots. The experiment required simulating 4 qubits (2 qubits for register M and 2 for register E) for the circuit related to amplitude estimation and 5 for that related to sign estimation, which involves adding a control qubit.

The results of the amplitude estimation are shown in Figure 3. As can be seen, most of the executions produced the expected output states (those related to the eigenvalues 01 and 11, i.e., 1 and 3 in binary, so $\frac{1}{2^2} = 0.25$ and $\frac{3}{2^2} = 0.75$ as demonstrated by Figure 4). This is consistent with the claims in Section III-B Regarding the estimation of the signs of the approximated amplitudes, Figure 5 shows the output of the corresponding circuit. We emphasize that here, we are interested in states where the first qubit from the bottom (the control qubit) takes on a value of zero.

The authors deemed it essential to conduct experiments also on actual QPUs. In particular, the tests were carried out using the IBM Perth quantum machine, which presents 7 qubits and a quantum volume of 32.

The outcomes of these experiments are available in the above-mentioned repository [20]. While the results are reasonably close to those of the ideal simulation in the case of amplitude estimation , the same cannot be said for the sign circuit. Here, the increase in depth entails a higher runtime which brings decoherence phenomena.

VI. Conclusions

This paper proposes a detailed quantum PCA approach for principal component extraction from covariance matrices. We extend existing tomography methods to handle cases with complex amplitudes, assuming that the imaginary part remains small, which is a reasonable assumption if we use an adequate resolution. We also present a Qiskit implementation of the proposed algorithm and apply it to a relevant use case in the financial domain. The results of our experiments demonstrate how our method effectively provides the expected outcomes for small-scale settings. Scaling the algorithm will require effective error mitigation and correction protocols, as well as a subsequent increase in quantum resources.

In future work, the authors aim to tackle the considerations regarding an efficient data loading which remain highly relevant, with recent publications showing promising results [28].

Acknowledgment

The authors would like to thank Giacomo Nannicini and Anupam Prakash for the constructive discussions on quantum state tomography, as well as the whole Eagle team for making this work possible. Concerning Intesa Sanpaolo, the authors would also like to thank the Enterprise Risk Analytics Office for providing the real business scenario of potential application for Quantum PCA and the Quantum Competence Center for helping the team in realizing the relative algorithm.

REFERENCES

[1] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "q-means: A quantum algorithm for unsupervised machine learning," in Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/16026d60ff9b54410b3435b403afd226-Paper.pdf

[2] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," Nature Communications, vol. 12, no. 1, May 2021. [Online]. Available: https://doi.org/10.1038/s41467-021-22539-9

[3] M. Schuld and N. Killoran, "Is quantum advantage the right goal for quantum machine learning?" PRX Quantum, vol. 3, p. 030101, Jul 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.3.030101

[4] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, "Generalization in quantum machine learning from few training data," Nature Communications, vol. 13, no. 1, Aug. 2022. [Online]. Available: https://doi.org/10.1038/s41467-022-32550-3

[5] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," Phys. Rev. A, vol. 103, p. 032430, Mar 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.103.032430

[6] M. Schuld and F. Petruccione, Machine Learning with Quantum Computers. Springer International Publishing, 2021. [Online]. Available: https://doi.org/10.1007/978-3-030-83098-4

[7] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," Nature Physics, vol. 10, no. 9, pp. 631–633, Jul. 2014. [Online]. Available: https://doi.org/10.1038/nphys3029

[8] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett., vol. 103, p. 150502, Oct 2009. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.103.150502

[9] A. Martin, B. Candelas, A. Rodríguez-Rozas, J. D. Martín-Guerrero, X. Chen, L. Lamata, R. Orús, E. Solano, and M. Sanz, "Toward pricing financial derivatives with an ibm quantum computer," Phys. Rev. Res., vol. 3, p. 013167, Feb 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.3.013167

[10] B. Duan, J. Yuan, Y. Liu, and D. Li, "Efficient quantum circuit for singular-value thresholding," Physical Review A, vol. 98, no. 1, Jul. 2018. [Online]. Available: https://doi.org/10.1103/physreva.98.012308

[11] J. Lin, W.-S. Bao, S. Zhang, T. Li, and X. Wang, "An improved quantum principal component analysis algorithm based on the quantum singular threshold method," Physics Letters A, vol. 383, no. 24, pp. 2862–2868, Aug. 2019. [Online]. Available: https://doi.org/10.1016/j.physleta.2019.06.026

[12] C. He, J. Li, W. Liu, J. Peng, and Z. J. Wang, "A low-complexity quantum principal component analysis algorithm," IEEE Transactions on Quantum Engineering, vol. 3, pp. 1–13, 2022. [Online]. Available: https://doi.org/10.1109/tqe.2021.3140152

[13] M. Treinish, J. Gambetta, T. Soolu, P. Nation, P. Kassebaum, D. M. Rodríguez, S. De La Puente González, J. Lishman, H. Shaohan, L. Bello, J. Garrison, K. Krsulich, H. Junye, J. Yu, M. Marques, J. Gacon, D. McKay, E. Arellano, J. Gomez, L. Capelluto, T. Scholten, A. Mitchell, A. Panigrahi, g. Leron, R. Rafey Iqbal, S. Wood, I. Toshinari, A. Pozas-Kerstjens, and C. J. Wood, "Qiskit," 2023. [Online]. Available: https://doi.org/10.5281/ZENODO.2573505

[14] I. Kerenidis and A. Prakash, "A quantum interior point method for lps and sdps," ACM Transactions on Quantum Computing, vol. 1, no. 1, oct 2020. [Online]. Available: https://doi.org/10.1145/3406306

[15] J. Hull, Risk management and financial institutions, 6th ed., ser. Wiley finance. Hoboken, New Jersey: Wiley, 2023.

[16] I. T. Jolliffe, Principal component analysis, 2nd ed., ser. Springer series in statistics. New York: Springer, 2002.

[17] P. Rebentrost, A. Steffens, I. Marvian, and S. Lloyd, "Quantum singular-value decomposition of nonsparse low-rank matrices," Phys. Rev. A, vol. 97, p. 012327, Jan 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.97.012327

[18] A. Y. Kitaev, "Quantum measurements and the abelian stabilizer problem," 1995.

[19] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th ed. USA: Cambridge University Press, 2011.

[20] "code repository," https://github.com/Eagle-quantum/QuPCA, 2023.

[21] C. T. Hann, C.-L. Zou, Y. Zhang, Y. Chu, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, "Hardware-efficient quantum random access memory with hybrid quantum acoustic systems," Phys. Rev. Lett., vol. 123, p. 250501, Dec 2019. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.123.250501

[22] D. K. Park, F. Petruccione, and J.-K. K. Rhee, "Circuit-based quantum random access memory for classical data," Scientific Reports, vol. 9, no. 1, Mar. 2019. [Online]. Available: https://doi.org/10.1038/s41598-019-40439-3

[23] T. M. L. de Veras, I. C. S. de Araujo, D. K. Park, and A. J. da Silva, "Circuit-based quantum random access memory for classical data with continuous amplitudes," IEEE Transactions on Computers, vol. 70, no. 12, pp. 2125–2135, 2021.

[24] O. D. Matteo, V. Gheorghiu, and M. Mosca, "Fault-tolerant resource estimation of quantum random-access memories," IEEE Transactions on Quantum Engineering, vol. 1, pp. 1–13, 2020.

[25] V. Shende, S. Bullock, and I. Markov, "Synthesis of quantum-logic circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 6, pp. 1000–1010, 2006.

[26] [Online]. Available: https://qiskit.org/documentation/stubs/qiskit.circuit.library.StatePreparation.html

[27] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," npj Quantum Information, vol. 5, no. 1, Nov. 2019. [Online]. Available: https://doi.org/10.1038/s41534-019-0223-2

[28] M. H. Gordon, M. Cerezo, L. Cincio, and P. J. Coles, "Covariance matrix preparation for quantum principal component analysis," PRX Quantum, vol. 3, p. 030334, Sep 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PRXQuantum.3.030334

[29] L. Madden and A. Simonetto, "Best approximate quantum compiling problems," ACM Transactions on Quantum Computing, vol. 3, no. 2, mar 2022. [Online]. Available: https://doi.org/10.1145/3505181

[30] "PhaseEstimation." [Online]. Available: https://qiskit.org/documentation/stubs/qiskit.circuit.library.PhaseEstimation.html

[31] "HHL." [Online]. Available: https://qiskit.org/documentation/stubs/qiskit.algorithms.linear_solvers.HHL.html

[32] "QuantumCircuit Class." [Online]. Available: https://qiskit.org/documentation/stubs/qiskit.circuit.QuantumCircuit.html

[33] R. Orús, S. Mugel, and E. Lizaso, "Quantum computing for finance: Overview and prospects," Reviews in Physics, vol. 4, p. 100028, Nov. 2019. [Online]. Available: https://doi.org/10.1016/j.revip.2019.100028

[34] D. Venturelli and A. Kondratyev, "Reverse quantum annealing approach to portfolio optimization problems," Quantum Machine Intelligence, vol. 1, no. 1-2, pp. 17–30, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s42484-019-00001-w

[35] I. Kerenidis, A. Prakash, and D. Szilágyi, "Quantum algorithms for portfolio optimization," in Proceedings of the 1st ACM Conference on Advances in Financial Technologies. ACM, Oct. 2019. [Online]. Available: https://doi.org/10.1145/3318041.3355465

[36] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, "Option pricing using quantum computers," Quantum, vol. 4, p. 291, Jul. 2020. [Online]. Available: https://doi.org/10.22331/q-2020-07-06-291

[37] S. Ramos-Calderer, A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, J. Cortada, J. Planagumà, and J. I. Latorre, "Quantum unary approach to option pricing," Phys. Rev. A, vol. 103, p. 032414, Mar 2021. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.103.032414

[38] S. Woerner and D. J. Egger, "Quantum risk analysis," npj Quantum Information, vol. 5, no. 1, Feb. 2019. [Online]. Available: https://doi.org/10.1038/s41534-019-0130-6

[39] E. Dri, E. Giusto, A. Aita, and B. Montrucchio, "Towards practical quantum credit risk analysis," in Journal of Physics: Conference Series, vol. 2416. IOP Publishing, Dec. 2022. [Online]. Available: https://doi.org/10.1088/1742-6596/2416/1/012002

[40] G. C. George E. P. Box, Gwilym M. Jenkins and G. M. Ljung, "Time series analysis: Forecasting and control," Journal of Time Series Analysis, vol. 37, no. 5, pp. 709–711, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jtsa.12194

[41] "Simulators overview." [Online]. Available: https://quantum-computing.ibm.com/lab/docs/iql/manage/simulator#qasm