A Generalisation of Classically Intractable Learning Problems that are Solvable with Quantum Machine Learning

Saleh Naghdi

A thesis in fulfilment of the requirements for the degree of

Bachelor of Science (Honours)

School of Physics

Faculty of Science

The University of New South Wales

2022

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: **Naghdi**

First name: **MohammadSaleh** Other name/s: **Saleh**

Abbreviation for degree as given in the University calendar: **BSc (Honours)**

School: **School of Physics** Faculty: **Faculty of Science**

Title: A Generalisation of Classically Intractable Learning Problems that are Solvable with Quantum Machine Learning

Abstract

In response to the many computational bottlenecks faced by classical machine learning, the field of quantum machine learning has arisen over the past decade to perform certain tasks comparatively faster. By leveraging quantum phenomena such as entanglement and superposition, the field of quantum machine learning has recently shown promise in claiming quantum speedup, that is, performing some tasks exponentially faster than their classical counterparts. However, to date, most of these algorithms rely on quantum RAM — a speculative data loading model whose experimental realisation has proved to be very difficult due to hardware constraints. It was not until 2021 when Liu *et al.* demonstrated the first quantum machine learning model without quantum RAM, called the `QKE-SVM`, that obtains provable quantum speedup in performing a specific $+1/-1$ binary classification task. In particular, they showed that only the `QKE-SVM` can embed the data points provided in their specific task into a *quantum feature space* wherein the $+1$ and -1 labels become largely separated and distinguishable. In this we deconstruct the classification task by Liu *et al.* to see if there are other examples that admit the same exponential quantum speedup. We derive three mathematical properties of Liu *et al.'s* binary classification task that enabled it to be solved exponentially faster by the `QKE-SVM`. We then devise two new binary classification problems, OF_P^1, and QR_P, with the former being based on the encryption protocol of order finding, and the latter that of quadratic residuosity. We show that QR_P and OF_P^2 ultimately fail to admit quantum speedup with a `QKE-SVM` beacause they satisfy some but not all of the three mathematical properties exhibited by Liu *et al.'s* task. By combining the best aspects of these two problems, we design a third and final classification task OF_P^1 based also on order finding which we prove does in fact admit quantum speedup for satisfying all three properties. Together with the Liu *et al.'s* result, the OF_P^2 serves as the only other instance of a classification task that a quantum machine learning model can complete exponentially faster than all classical counterparts. Our framework represents a new tool by which classically intractable classification tasks can be discovered that are efficiently solvable using a quantum machine learning model, which is the primary goal of quantum machine learning.

Declaration relating to disposition of project thesis/dissertation

I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

Signature **MohammadSaleh Naghdi** Witness **Casey Myers** Date **11 November, 2022**

FOR OFFICE USE ONLY Date of completion of requirements for Award

Originality Statement

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Saleh Naghdi
17 November, 2020

Copyright Statement

I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

Saleh Naghdi
17 November, 2020

Authenticity Statement

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

Saleh Naghdi
17 November, 2020

Abstract

In response to the many computational bottlenecks faced by classical machine learning, the field of quantum machine learning has arisen over the past decade to perform certain tasks comparatively faster. By leveraging quantum phenomena such as entanglement and superposition, the field of quantum machine learning has recently shown promise in claiming quantum speedup, that is, performing some tasks exponentially faster than their classical counterparts. However, to date, most of these algorithms rely on quantum RAM — a speculative data loading model whose experimental realisation has proved to be very difficult due to hardware constraints. It was not until 2021 when Liu *et al.* demonstrated the first quantum machine learning model without quantum RAM, called the `QKE-SVM`, that obtains provable quantum speedup in performing a specific $+1/-1$ binary classification task. In particular, they showed that only the `QKE-SVM` can embed the data points provided in their specific task into a *quantum feature space* wherein the $+1$ and -1 labels become largely separated and distinguishable. In this thesis we deconstruct the classification task by Liu *et al.* to see if there are other examples that admit the same exponential quantum speedup. We derive three mathematical properties of Liu *et al.'s* binary classification task that enabled it to be solved exponentially faster by the `QKE-SVM`. We then devise two new binary classification problems, OF_P^1, and QR_P, with the former being based on the encryption protocol of order finding, and the latter that of quadratic residuosity. We show that QR_P and OF_P^1 ultimately fail to admit quantum speedup with a `QKE-SVM` because they satisfy some but not all of the three mathematical properties exhibited by Liu *et al.'s* task. By combining the best aspects of these two problems, we design a third and final classification task OF_P^2 based also on order finding, which we prove does in fact admit quantum speedup as it satisfies all three properties. Together with Liu *et al.'s* result, the OF_P^2 serves as the only other instance of a classification task that a quantum machine learning model can complete exponentially faster than all classical counterparts. Our framework represents a new tool by which classically intractable classification tasks can be discovered that are efficiently solvable using a quantum machine learning model, which is the primary goal of quantum machine learning.

Acknowledgement

I extend my warmest gratitude to my supervisors, Prof. Michelle Simmons and Dr. Casey Myers for their invaluable advice, continuous support, and patience during my honours. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

Contents

Abstract **iii**

Acknowledgement **iv**

Contents **v**

List of Figures **vii**

1 Introduction **1**

1.1 Quantum Machine Learning without Quantum RAM 1

2 The Support Vector Machine **3**

2.1 High-level motivation 3

2.2 Mathematical formulation 4

3 The Quantum Kernel Estimation Support Vector Machine **8**

3.1 Quantum Kernel Estimation 8

3.2 A quantum feature map achieving quantum advantage 10

4 Determining properties of classically intractable promise problems admitting quantum speedup **14**

4.1 Statistical Zero-Knowledge: a class of problems admitting a representation in quantum feature space 15

4.2 Initial state preparation and Index erasure for efficient preparation of quantum feature maps 17

4.3 Two new classically intractable learning problems which are distinct from the DLP 20

4.4 Summary and reflections 26

5 A new classically intractable promise problem based on order finding 28

5.1 Preliminaries 28

5.2 The OF_P^2 promise problem 31

5.3 Summary 36

6 Conclusion and Outlook 37

References 38

List of Figures

2.1 **Steps taken in the Support Vector Machine for identifying a wide-margin decision boundary that separates a linearly separable dataset.** First we require a linearly separable dataset as in 2.1a to ensure that there exists a line that can separate the dataset to begin with. Some lines are better than others, and in 2.1b the optimal line is the one with the widest margin as this separates the two labels maximally. To obtain this optimal line, we first parameterise it with a normal vector $\vec{w}$ satisfing 2.2. We insist that the data points $\vec{x}_{\pm}$ lying on the 'gutters' are those that precisely yield a ± 1 to the line equation 2.1c. Using this fact, we can derive an expression for the margin of the decision boundary Δ as the projection of $\vec{x_+} - \vec{x_-}$ along $\hat{w}$. It is this Δ that we need to maximise to obtain the optimal boundary. 5

2.2 **The SVM can still obtain a decision boundary for an originally linearly inseparable dataset if there exists some *feature space* equipped with a map φ in which the dataset becomes linearly separable once again.** . . 6

3.1 **General design of a quantum circuit serving as the kernel function of the map $\mathbf{x} \mapsto |\mathbf{x}\rangle_{\mathbf{U}}$ as defined in 3.4**. It is sufficient to measure the probability of measuring $|0\rangle$ to obtain the inner product between pairs of data points in the feature Hilbert space. 9

3.2 **Use of the quantum feature map 4.1 to separate green/red (+1/-1) instances from each other under the labeling rule of 3.5.** The dots on the circles represent the inputs to the problem's corresponding labelling rule which outputs a label corresponding to red/green (or $+1/-1$). For example, in 3.2a, the data point $x = 28$ is green because $log_{11}(28) = 14 \in [1, m-1] = [s, s + (p-1)/2 - 1] = [11, 24] \implies f_{s=11}(x) = +1$. 3.2b represents where the data points are mapped to under the action of $x \mapsto \log_{11}(x)$. For example, since $x = 29 \mapsto log_{11}(28) = 14$, $x = 29$ is mapped to the position of 14. The concentric arcs denote which points are swept in the superpositions formed from the quantum feature state corresponding to a data point 4.1. For example, the superposition obtained from $|6\rangle \mapsto |C_{6,k}\rangle$ (shown as the red arc) sweeps the first $2^k - 1$ elements whose discrete logarithm is greater than $log_{11}(6) = 26$. The blue, green and red colours of the arcs indicate that the corresponding superposition state represents a hyperplane $|\phi_1 1\rangle$, the feature state of some $+1$ instance $|C_{6,k}\rangle$, and the feature state of some -1 instance $|C_{24,k}\rangle$ respectively. Note that the the green arc ($+1$ feature state) overlaps much more with the hyperplane than does the red arc, indicating the hyperplane has a wide-margin. 11

4.1 **The relationship of DLP_P with respect to several different classes of problems allows it to satisfy criteria C.1—C.3 required for admitting quantum speedup.** DLP_P is a member of SZK (orange dotted circle) and formulated as an SZK instance (orange wedge), it is *equivalent* to the (conjectured) classically intractable discrete logarithm problem DLP, which belongs to $AHSP$ (**C.1**). As a member of SZK, DLP_P can be further reduced to a (potentially harder) *QSampling* problem that defines the quantum feature space (**C.2**). The *QSampling* version of the problem (yellow wedge) can be efficiently solved using the $DLP \in AHSP$ for index erasure, together with a known initial state preparation routine ISP (**C.3**). 15

4.2 **Three SZK promise problems which include the original DLP_P, and two new ones, OF_P^1 and QR_P, whose red/green (or $-1/+1$) instances are shown to be randomly distribution in their original space, but well separated in a corresponding quantum feature space.** The dots on the circles represent the inputs to the problem's corresponding labelling rule which outputs a label corresponding to red/green (or $+1/-1$). For example, in 4.2a, the data point $x = 1$ is green because $ord_{61}(1) = 1 \in [1, m-1] = [1, 19] \implies h_{OF^1}(x) = +1$. The bottom row displays where the data points are mapped to under the action of the classically intractable map $x \mapsto CI(x)$. So, in the case of 4.2a, $x = 1 \mapsto ord_{61}(1) = 1$ and $x = 1$ is mapped to the position of 1. The concentric arcs denote which points are swept in the superposition of the quantum feature state corresponding to a data point. For example, in dataset 4.2f, the superposition (shown in blue) obtained from $|1\rangle \mapsto |C_1\rangle$ sweeps all elements which are quadratic residues (*c.f.* Eq (4.26)), which corresponds to all the green dots. The blue, green and red colours of the arcs indicate that the superposition state represents a hyperplane, the feature state of some $+1$ instead, and the feature state of some -1 respectively. Note that in all cases, the green arcs (feature states) overlap much more with the hyperplane than do the red arcs, indicating the hyperplane has a wide-margin. 21

4.3 **The OF_P^1 is missing an efficient routine for initial state preparation (ISP) and thus C.3, which would have otherwise allowed it to satisfy criteria C.1—C.3 required for quantum speedup.** The class relations of the OF_P^1 are entirely analogous to that of the DLP_P outlined in Figure 4.1. It is unknown whether the OF_P^1, as an instance of SZK, is *equivalent* to the classically intractable task of order finding $OF \in AHSP$ — only that it is *as hard* as it. . . 22

4.4 **The QSampling instance (yellow wedge) of the QR_P has no known way to be solved using a problem in $AHSP$, and thus fails to satisfy C.3**. The index erasure problem required to build the quantum feature map in (4.26) does make use of an efficiently implementable initial state preparation scheme (ISP), but is itself not efficient. Instead of an exponential speedup, we can still obtain a quadratic improvement by solving index erasure using the algorithm provided in Algorithm 2. Note that a guide to reading the other elements of this diagram has been outlined in Figure 4.1. 24

5.1 **The relationship of OF_P^2 with respect to several different classes of problems indicating that it is a classically intractable promise problem admitting an efficiently implementable quantum feature map in which it is linearly separable, *i.e.* it satisfies criteria C.1—C.3.** OF_P^2 is is a member of SZK (orange wedge) and is *at most* as difficult as the classically intractable problem of order finding OF, which belongs to $AHSP$. We conjecture that the OF_P^2 *is* classically intractable just as Liu *et al.* do for the DLP_P (**C.1**). As a member of SZK, OF_P^2 can be further reduced to a potentially harder $QSampling$ problem (yellow wedge) that defines a quantum feature space (**C.2**). The $QSampling$ version of the problem can be efficiently solved using $OF \in AHSP$ together with a known initial state preparation routine ISP (**C.3**). 29

5.2 **Smoothed histograms displaying several classical machine learning models being drastically outperformed by the `QKE-SVM` at the task of OF_P^2 (5.2a). 5.2b and 5.2c respectively show kernel matrices corresponding to the classical and quantum feature maps that were used, revealing that the latter clearly separates instances based on their binary label unlike the former.** The bottom panel of 5.2a summarises the average accuracy of each model flanked by a σ error bar. The dataset on which the models were trained had an even distribution of $+1$ and -1 instances, thus the classical models on average performed no better than random guessing ($\mu_c = 49\%$) suggesting the classical intractability of the OF_P^2. 35

Chapter 1

Introduction

1.1 Quantum Machine Learning without Quantum RAM

In 2019, Google claimed it has accomplished quantum supremacy whereby there quantum computer Sycamore performed in 200 seconds a calculation that would otherwise have taken a classical supercomputer 10,000 years [1]. In our increasingly computer-automated world, the demand for algorithms that not only solve problems but that do so quickly has never been higher. This search for faster algorithms drives much of the impetus behind the nascent field of quantum computing which focuses on designing quantum algorithms that afford a *quantum speed-up* — that is, be efficiently able to solve certain problems which would otherwise take classical algorithms much longer to solve.

Due to its wide applicability in all domains of technology, from predicting the shape of proteins to stock market analytics, the field of machine learning remains one particular branch of algorithmics for which quantum speed-ups are highly sought after [2]. Machine learning generally refers to the task of building parameterised models that use data to fine-tune or 'train' their parameters as befits the requirements of a certain task or problem. The endeavour to construct quantum machine learning models which boast speedups in the task of training or testing thus captures the primary goal of the burgeoning field of *quantum machine learning* (QML) [3].

Quantum machine learning emerged in 2009 following the seminal work of Lloyd *et al.* in which they formulated a quantum machine learning model that claims to perform the important task of principal component analysis exponentially faster than the best known classical models [4]. Their breakthrough algorithm imminently led to a proliferation of QML models promising quantum speedup, many of which adopted similar techniques employed in the original paper [5–8].

Despite their promises for quantum speedup, there remains a large caveat to all of these QML models. In order to be successfully implemented, most QML models share a dependency on an architecture known as quantum RAM, which, analogous to classical RAM, enables classical data to be efficiently loaded into and retrieved from a quantum computer [9]. However, the issue with relying on quantum RAM is that it is yet to be experimentally realised as it is difficult to store quantum states long times without them eventually decohering. This relegates the

promised quantum speedups behind the breakthrough QML findings to mere conjectures [10]. Additionally, the recent discovery of quantum-inspired classical algorithms has revealed that it is possible to classically gain as much of a computational speedup as what many QML models that are based on quantum RAM achieve by assuming a classical input model that is equivalent to quantum RAM [11]. This raises the suspicion that the source of these speedups lies not in the design of the quantum algorithm, but rather its dependence on an input model that is yet to be implemented.

In light of the impractical nature of quantum RAM, heuristic QML models began to be developed in 2018 which could be implemented on the noisy intermediate-scale quantum (NISQ) computers of today [12]. Though it was not until 2021 when Liu *et al.* showcased the first instance of such a QML model that did not rely on quantum RAM but managed to admit a provable quantum speedup over all classical counterparts on a specific classification task [13]. The authors contrived a specific classification task together with a corresponding dataset in order to show that their model, called a Quantum Kernel Estimation Support Vector Machine (`QKE-SVM`), enjoys this quantum speed-up because it is able to identify patterns in the dataset which no classical model can without using exponential resources. While this finding served as a proof of concept for the potential of QML to produce quantum speed-ups, the contrived nature of the classification task on which the model was performed limits its usage [**?**, 14].

As the only rigorous demonstration in the literature of a QML model exhibiting provable quantum speedup, the `QKE-SVM` and the classification task designed by Liu *et al.* deserve to be deconstructed should this quantum speedup be extended to a broader range of tasks. The purpose of this honours is to rigorise mathematical properties of the model and learning problem used by Liu in order to generalise the findings to new classification tasks that enjoy a quantum speedup. To this end, we hope to derive a framework which serves as a diagnostic tool for whether a learning problem will admit a quantum speedup via the `QKE-SVM` or not.

Chapter 2

The Support Vector Machine

Liu *et al.'s* quantum machine learning model which exhibits quantum speedup without the aid of quantum RAM is actually based on a famous classical model in machine learning known as the Support Vector Machine. To appreciate the quantum speedup afforded by Liu *et al.*, in this section we lay the foundations for the Support Vector Machine [15].

In the following, we first provide a high-level motivation for the emergence of the Support Vector Machine as a machine learning model intended for classification tasks. Then, we establish with mathematical formalism of the SVM which eventually leads us to a technique in machine learning known as the *kernel trick*. It is this *kernel trick* that Liu *et al.* capitalise on to achieve quantum speedup in their quantum algorithm [16].

2.1 High-level motivation

Consider we are given a distribution of data points that are already labelled with one of two classes: blue or red. As new data points arrive, what is the best way to classify them based on the binary-labelled distribution we are given? A technique would be to draw a curve that partitions the space of data points into two subsets, one corresponding to each class. Now suppose we constrain the partitioning curve so that it can only be a straight line. The Support Vector Machine is able to identify the one straight line that *optimally* partitions the dataset with respect to class labels (figure 2.1a)

A straight line is considered optimal if it partitions the dataset while remaining as far away from all the data points as possible; figure 2.1c displays how the optimal line is flanked by the widest margin. The intuition behind defining the optimal partition to have the widest margin is that any new data point that lies within a margin's length of the training samples with the same label is correctly classified.

What if the distribution of data points is not able to be partitioned by a straight line? In that case, we can map the data points to a (potentially much larger) space where a linear partition *does* exist, and feed these mapped data points to the support vector machine. However, mapping

vectors to larger spaces is often costly and grows in the dimension of the feature space. We have reached the limitations of our high-level overview of the SVM, as plunging into the mathematics reveals that you do not even need an explicit map to enjoy the benefits of a non-linear partition. In fact, all you need is a measure of similarity between vectors in that space, which is much more efficient to compute than an explicit map.

2.2 Mathematical formulation

Consider an input space $X \subset \mathbb{R}^N$ and a sample $S \subset X$ of m labelled training instances, $S = \{(x_i, y_i),\ i \in [m]\}$. We work with a binary labelled dataset where the labels take only one of two values $y_i \in \{+1, -1\}$.

For now, we will assume there exists a line or hyperplane that can separate the dataset S such that all the $+1$ data points lie on one side of the hyperplane, and -1 on the other. This means that the dataset is linearly separable. Later we will relax this condition to consider linearly inseparable datasets.

Consider the figure in 2.1b. If we take $\vec{w}$ to be the normal to the partitioning hyperplane, then we can determine whether a data point labelled with $+1\,(-1)$ $\vec{x}_i$ lies above (below) the hyperplane by the projection of $\vec{x}_i$ along $\vec{w}$, if it is greater (lower) than some threshold b:

$$\begin{cases} \vec{w} \cdot \vec{x}_i \ > b & y_i = +1 \\ \vec{w} \cdot \vec{x}_i \ < b & y_i = -1 \end{cases} \implies \begin{cases} \vec{w} \cdot \vec{x}_i + b \ > 0 & y_i = +1 \\ \vec{w} \cdot \vec{x}_i + b \ < 0 & y_i = -1 \end{cases} \quad \text{where } b \leftarrow -b$$
$$\implies y_i(\vec{w} \cdot \vec{x}_i + b) > 0 \tag{2.1}$$

The inequality 2.1 represents the condition of linear separability. The decision boundary is used to classify a new data point $\vec{u}$ by seeing on which side of the hyperplane it lies:

$$y_u := \operatorname{sgn}(\vec{w} \cdot \vec{u} + b) \tag{2.2}$$

The goal of the SVM is to find the optimal line or hyperplane that best partitions the dataset. Recall that the optimal decision boundary is one that best separates the dataset and has the widest margin. Let us find a quantifiable measure for this so-called margin. We insist that $(\vec{w}, b)$ take values such that the closest data point $\vec{x}_{+1}$ above ($\vec{x}_{-1}$ below) the hyperplane yields exactly the value $+1$ (-1) when substituted in for 2.1. Figure 2.1b illustrates we can obtain the thickness of the margin Δ:

$$\Delta = (\vec{x}_{+1} - \vec{x}_{-1}) \cdot \frac{\vec{w}}{\|\vec{w}\|} \tag{2.3}$$
$$= (\vec{x}_{+1} \cdot \vec{w} - \vec{x}_{-1} \cdot \vec{w}) \frac{1}{\|\vec{w}\|} \tag{2.4}$$
$$= \big((1/y_{+1} - b) - (1/y_{-1} - b)\big) \frac{1}{\|\vec{w}\|} \quad \text{from 2.1} \tag{2.5}$$
$$= \frac{2}{\|\vec{w}\|} \tag{2.6}$$

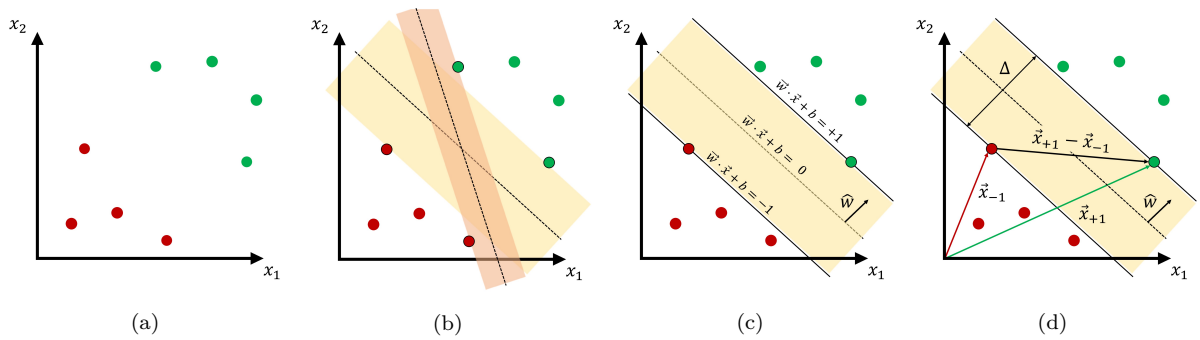


Figure 2.1: **Steps taken in the Support Vector Machine for identifying a wide-margin decision boundary that separates a linearly separable dataset.** First we require a linearly separable dataset as in 2.1a to ensure that there exists a line that can separate the dataset to begin with. Some lines are better than others, and in 2.1b the optimal line is the one with the widest margin as this separates the two labels maximally. To obtain this optimal line, we first parameterise it with a normal vector $\vec{w}$ satisfing 2.2. We insist that the data points $\vec{x}_\pm$ lying on the 'gutters' are those that precisely yield a ± 1 to the line equation 2.1c. Using this fact, we can derive an expression for the margin of the decision boundary Δ as the projection of $\vec{x_+} - \vec{x_-}$ along $\hat{w}$. It is this Δ that we need to maximise to obtain the optimal boundary.

By the *wide-margin* criterion, the SVM is to maximise $\frac{2}{\|\vec{w}\|}$ or equivalently minimise $\frac{1}{2}\|\vec{w}\|^2$ under the constraint that the decision boundary separate the dataset with a margin:

$$(\vec{w}, b)_{optimal} = \min_{\vec{w},\, b} \frac{1}{2}\|\vec{w}\|^2 \quad \text{ensure}: y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1\,. \tag{2.7}$$

2.2.1 The dual form of the objective function and classification rule

This constrained optimisation problem 2.8 can be reformulated and solved using the well-known method of Lagrange multipliers, which gives the following objective function to minimise:

$$\mathcal{L} = \frac{1}{2}\|\vec{w}\|^2 - \sum_i \alpha_i \Big(y_i(\vec{w}\cdot\vec{x}_i + b) - 1 \Big) \tag{2.8}$$

where we have introduced α_i as the Lagrange multipliers. Differentiating $\mathcal{L}$ with respect to the parameters $(\vec{w}, b)$ and setting them to zero yields:

$$\left.\begin{aligned} \frac{\partial \mathcal{L}}{\partial \vec{w}} &= \vec{w} - \textstyle\sum_i \alpha_i y_i \vec{x}_i \\ \frac{\partial \mathcal{L}}{\partial b} &= \textstyle\sum_i \alpha_i y_i \end{aligned}\right\} \stackrel{!}{=} 0 \implies \left\{\begin{aligned} \vec{w} &= \textstyle\sum_i \alpha_i y_i \vec{x}_i \\ \textstyle\sum_i \alpha_i y_i &= 0 \end{aligned}\right. \tag{2.9}$$

Substituting the constraints in 2.9 back into the objective function 2.8 and the classification rule 2.2, respectively, gives

$$\mathcal{L} = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \tag{2.10}$$

and

$$y_u := \operatorname{sgn}\Big(\sum_i \alpha_i y_i \vec{x}_i \cdot \vec{u} + b\Big) \tag{2.11}$$

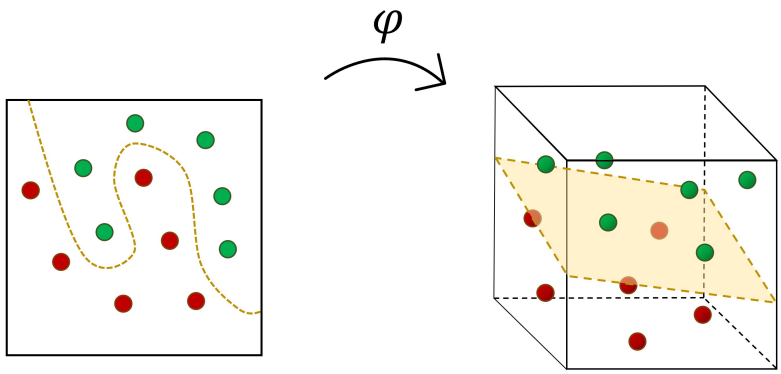


Figure 2.2: **The SVM can still obtain a decision boundary for an originally linearly inseparable dataset if there exists some *feature space* equipped with a map φ in which the dataset becomes linearly separable once again.**

where $b = 1 - \sum_i \alpha_i y_i \vec{x}_i \cdot \vec{x}_{+1}$ and x_{+1} is a data point that lies on the margin.

The transformation from 2.2 to 2.11 and 2.8 to 2.10 is known as the *dual form*. Note that in the dual form, the data points x_i only occur in inner products with other data points and never in isolation. This structure lends to an incredibly useful technique known as the kernel trick which enables binary classification even on datasets that are not linearly separable as we will see in the following section.

2.2.2 The kernel trick

In figure 2.1d we see a dataset that is linearly inseparable, so that minimising the objective function 2.10 for this dataset will not produce an optimal decision boundary. While the dataset remains linearly inseparable in its original space, figure 2.2 illustrates that mapping the data point to some larger space $\vec{x} \mapsto \varphi(\vec{x})$ makes the dataset once again linearly separable. We call this space the *feature space*, because it reveals the hidden features of the data points. It suffices to replace $\vec{x}$ with $\varphi(\vec{x})$ in order for the hyperplane in the feature space to be obtained using the same techniques as above.

However, we are confronted with a problem: mapping data points into spaces with larger (and at times, infinite) dimensions is very costly. Fortunately, the dual form of the optimisation problem means that we can evade explicitly mapping data points if we can efficiently compute the inner products of the data points in feature space:

$$K(\vec{x}_i, \vec{x}_j) = \langle \varphi(\vec{x}_i), \varphi(\vec{x}_j) \rangle. \tag{2.12}$$

We can avoid the map $\vec{x} \mapsto \varphi(\vec{x})$ provided we have access to a function $K(\vec{x}_i, \vec{x}_j)$ that computes the inner product above, which is the case for many useful feature maps φ that are otherwise too costly to compute explicitly. K is referred to as the *kernel function*. Replacing every inner product in the linear dual form 2.10 with the kernel function is referred to as the *kernel trick*:

$$\mathcal{L} = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j) \tag{2.13}$$

and

$$y_u := \operatorname{sgn}(\sum_i \alpha_i y_i K(\vec{x}_i, \vec{u}) + b) \tag{2.14}$$

Coincidentally, quantum computers are very good at computing costly inner products in large, exponentially sized Hilbert spaces. We will see in the next section that this means quantum computers make for very good kernel functions [12].

Chapter 3

The Quantum Kernel Estimation Support Vector Machine

In the previous section, we established that the support vector machine is able to identify the widest-margin hyperplane that best partitions a linearly separable dataset. The SVM can also partition linearly *inseparable* datasets as long as the data points are mapped to a user-specified space in which the dataset *is* once again linearly separable. In fact, the SVM only requires a function that computes *inner products* between pairs of data points within the user-specified feature space to be able to identify a nonlinear decision boundary. We call such a function the kernel function K which efficiently calculates the inner product of data points in feature space when only provided the data points in their original space:

$$K(\vec{x}_i, \vec{x}_j) = \langle \varphi(\vec{x}_i), \varphi(\vec{x}_j) \rangle \,. \tag{3.1}$$

By only implicitly operating in the feature space, the kernel function circumvents the cost of first explicitly mapping x to the feature space in order to compute inner products. The use of the kernel function in this way is referred to as the *kernel trick*.

So where does quantum computing enter the picture? Remarkably, quantum computers are able to map data points to feature spaces that no classical computer can and better yet, efficiently calculate inner products within them. This means that quantum circuits can naturally serve as kernel functions for feature spaces that are inaccessible to classical computers. In this section we will see how quantum computing can be used in the kernel trick to help identify nonlinear decision boundaries which no classical computers can identify.

3.1 Quantum Kernel Estimation

Support you are given a d dimensional data point $\vec{x} \in Rd$. Now consider an m-qubit unitary gate $U(\vec{x})$ that is parameterised by the entries of the data point $\vec{x}$. If we define $|U(\vec{x})\rangle := U(\vec{x}) \, |0\rangle^{\otimes m}$, the map

$$\varphi : Rd \to \mathcal{H} \cong C2^m, \, \vec{x} \mapsto |U(\vec{x})\rangle \tag{3.2}$$

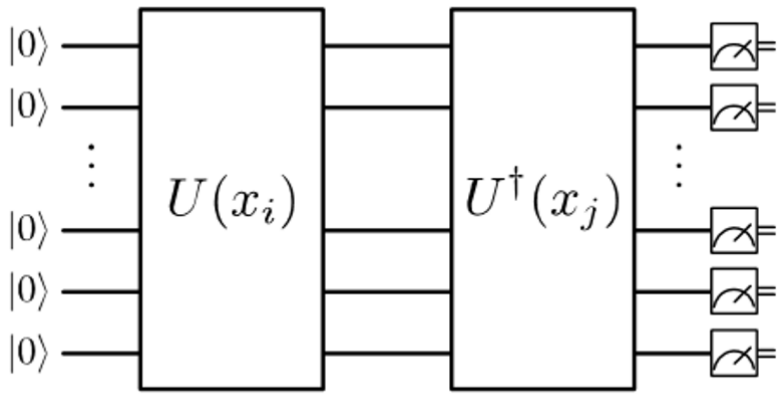


Figure 3.1: **General design of a quantum circuit serving as the kernel function of the map $\mathbf{x} \mapsto |\mathbf{x}\rangle_{\mathbf{U}}$ as defined in 3.4**. It is sufficient to measure the probability of measuring $|0\rangle$ to obtain the inner product between pairs of data points in the feature Hilbert space.

can be interpreted as a feature map. Calculating $|U(\vec{x})\rangle$ involves applying the unitary $U(\vec{x})$ to a register $|0\rangle^{\otimes m}$. This is a very natural operation for quantum computers, the feature map φ can be readily computed using a quantum circuit. Moreover, it is often the case that simulating $|U(\vec{x})\rangle$ with a classical computer is exponentially difficult for larger unitaries. As a result, not only is φ efficient to calculate with a quantum computer, but is intractable on a classical computer. In this sense, the quantum feature map φ would be rendered *inaccessible* to a classical computer.

Recall from the kernel trick that we do not even require the explicit coordinates of $\varphi(\vec{x})$ — only its inner products with other data points. The Hilbert space $\mathcal{H}$ is equipped with the canonical Hilbert-Schmidt inner product:

$$\langle |\psi\rangle, |\phi\rangle\rangle_{\mathcal{H}} := |\langle\psi|\phi\rangle|^2 . \tag{3.3}$$

Using this definition, we can clarify the kernel function K that is able to efficiently compute inner products between feature states under φ:

$$\begin{aligned} K(\vec{x}_i, \vec{x}_j) &= \langle \varphi(\vec{x}_i), \varphi(\vec{x}_j)\rangle \\ &= \langle |U(\vec{x}_i)\rangle, |U(\vec{x}_j)\rangle\rangle_{\mathcal{H}} \\ &= |\langle U(\vec{x}_i)|U(\vec{x}_j)\rangle|^2 . \\ &= |\langle 0|U(\vec{x}_i)^\dagger U(\vec{x}_j)|0\rangle|^2 . \end{aligned} \tag{3.4}$$

3.4 is yet another very natural expression in quantum computing and it simply corresponds to evaluating the probability of measuring $|0\rangle$ on the state $U(\vec{x}_i)^\dagger U(\vec{x}_j)|0\rangle$ whose circuit representation is shown in Figure 3.1 [12]. The technique of using the quantum circuit 3.1 to evaluate the kernel function for subspace spanned by $\{|U(\vec{x})\rangle \forall \vec{x}\} \subset \mathcal{H}$ is called *quantum kernel estimation*.

We recall that the original motive behind employing the kernel trick was to efficiently identify non-linear decision boundaries for linearly inseparable datasets that are linearly separable in some feature space. With quantum kernel estimation, we now have the option to use the kernel trick for *quantum* feature maps φ. However, it must be known a priori that mapping the

originally linearly inseparable dataset to the feature space that the kernel K implicitly operates on makes the dataset linearly separable. So, the question remains whether there are any linearly inseparable datasets that become linearly separable *only* when mapped to a quantum feature space. This would imply that the SVM could only solve this dataset using quantum kernel estimation, as no classical feature space exists in which the dataset is linearly separable. Indeed, Liu *et al.* answer this in the affirmative. In the following section we see that by defining a very specific dataset equivalent to a famous classically intractable problem, Liu *et al.* are able to show that only with quantum kernel estimation is an SVM able to separate the dataset.

3.2 A quantum feature map achieving quantum advantage

In the spirit of achieving quantum advantage, what is needed is a quantum feature map that *provably* outperforms all classical learners on some dataset. This is precisely what was demonstrated by Liu *et al* in Ref. [13] for a different adhoc dataset based on the Discrete Logarithm Problem (DLP) called the DLP_P which is widely conjectured to be classically intractable. The authors prove that if there were a classical classifier able to learn their constructed dataset, or more technically, *promise problem*, then that classifier could also solve the DLP, thus contradicting its conjectured intractability. By producing a quantum feature map which they prove has a good max-margin hyperplane, they can consequently claim that their quantum classifier outperforms all classical learners. A high-level summary of this finding is shown in Figure 3.2b.

As the only current instance of end-to-end quantum machine learning achieving quantum advantage in the literature, the work of Liu *et al.* deserves a closer exploration which we will now cover in the following section.

3.2.1 A DLP-based dataset

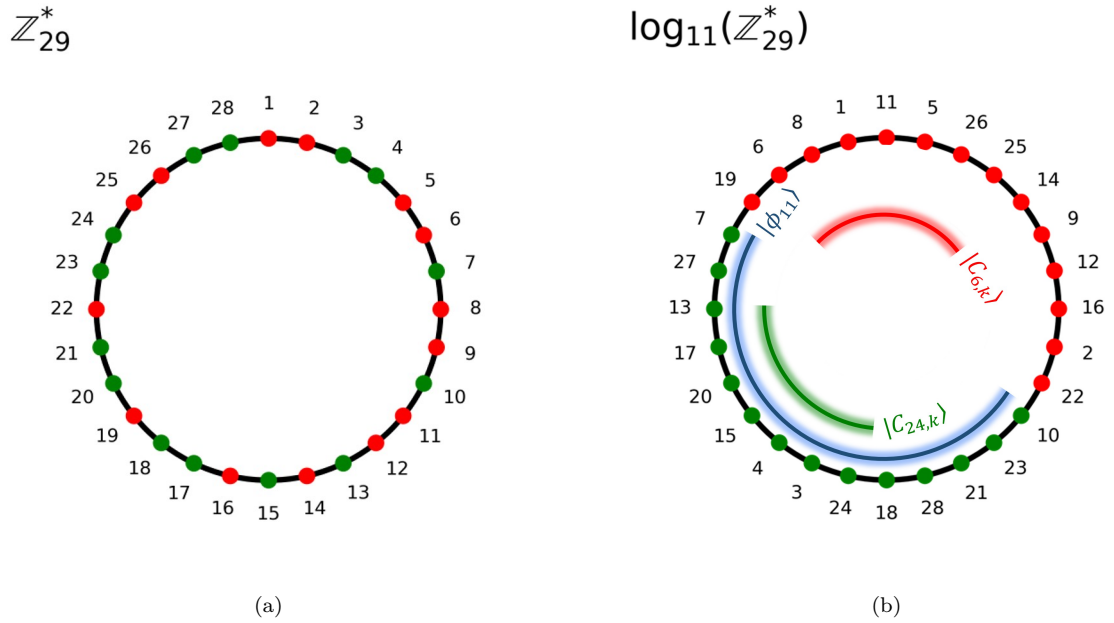
The Discrete Log Problem (DLP) is formulated as follows [17]. Let p be a prime number and $Z*_p = \{1, ..., p-1\}$ the integers under multiplication modulo p. If g is a generator of $Z*_p$, then given a $y \in Z*_p$, find $x \in Z*_p$ such that $y = g^x$ or equivalently, evaluate the discrete logarithm $\log_g(y) = x$. In summary,

$DLP(p, g)$ A prime number p, a generator g of the set $Z*_p = \{1, 2, \cdots, p-1\}$, and any element $y \in Z*_p$ What is $\log_g(y)$?

It is widely conjectured that there is no classical algorithm capable of efficiently evaluating $\log_g(y)$, whereas a quantum DLP solver does exist that is based on Shor's algorithm. This places the DLP within the Bounded-error Quantum Polynomial time (BQP) complexity class [18].

Given this definition of the DLP, Liu *et al.* define a binary classification problem also known as a *promise problem*. The promise problem is called the DLP_P and it takes place over the elements of $Z*_p$ which are labelled according to the following hidden binary labelling rule:

$$f_s(x) = \begin{cases} +1, & \text{if } \log_g(x) \in [s, s + \frac{p-3}{2}] \\ -1, & \text{else.} \end{cases} \tag{3.5}$$



(a) (b)

Figure 3.2: **Use of the quantum feature map 4.1 to separate green/red (+1/-1) instances from each other under the labeling rule of 3.5.** The dots on the circles represent the inputs to the problem's corresponding labelling rule which outputs a label corresponding to red/green (or $+1/-1$). For example, in 3.2a, the data point $x = 28$ is green because $log_{11}(28) = 14 \in [1, m-1] = [s, s+(p-1)/2-1] = [11, 24] \implies f_{s=11}(x) = +1$. 3.2b represents where the data points are mapped to under the action of $x \mapsto \log_{11}(x)$. For example, since $x = 29 \mapsto log_{11}(28) = 14$, $x = 29$ is mapped to the position of 14. The concentric arcs denote which points are swept in the superpositions formed from the quantum feature state corresponding to a data point 4.1. For example, the superposition obtained from $|6\rangle \mapsto |C_{6,k}\rangle$ (shown as the red arc) sweeps the first $2^k - 1$ elements whose discrete logarithm is greater than $log_{11}(6) = 26$. The blue, green and red colours of the arcs indicate that the corresponding superposition state represents a hyperplane $|\phi_1 1\rangle$, the feature state of some $+1$ instance $|C_{6,k}\rangle$, and the feature state of some -1 instance $|C_{24,k}\rangle$ respectively. Note that the the green arc ($+1$ feature state) overlaps much more with the hyperplane than does the red arc, indicating the hyperplane has a wide-margin.

for a choice of $s \in Z*_p$. Note that the interval in the definition sweeps half of the elements.[1]

To illustrate what this labelling rule corresponds to, it is instructive to first imagine the $p-1$ many elements of $Z*_p$ as being distributed evenly along a circle. While in this representation the class labels seem to be distributed randomly (Figure 3.2a) – and indeed, they are to a classical classifier – mapping each point to its discrete logarithm $x \mapsto \log_g(x)$ reveals the underlying classification rule, whereby the two labels are separated by a diameter line intercepting the point s, with $+1$ (-1) data points lying clockwise (anticlockwise) from s (Figure 3.2b).

The existence of this diameter partition already hints at the possibility of there being a hyperplane in some feature space that differentiates between the two class labels, and indeed, the next section shows that there is one such feature map. Importantly, however, it is that the feature map must necessarily be quantum which distinguishes this case from the previous.

[1]The addition in the interval is a slight abuse of notation as addition is not a native operation of $Z*_p$, so we have (p-1) + 1 = 1

3.2.1.1 An expressive quantum feature map

Let $n = \log_2(p)$ be the minimum number of bits required to represent prime number p and define (y, g, s) as before. The quantum feature map is defined as

$$|y\rangle \mapsto \sum_{i=0}^{2^k-1} |y \cdot g^i \mod p\rangle := |C_{y,k}\rangle \,. \tag{3.6}$$

for a choice of a constant $k \leq n$. The circuit for this feature map will make use of the DLP-solver which was shown in the previous section to organise the dataset into well-separated class clusters.

As a first step towards building the desired quantum feature map, consider the following function:

$$\begin{aligned} &C_{y,k} : \{0,1\}^k \to \{0,1\}^n \\ &C_{y,k}(i) = y \cdot g^i \mod p \,. \end{aligned}$$

By using elementary multiplication modulo p – the circuit for which is well understood and efficient[2]– it is easy to implement the reversible operator

$$\begin{aligned} &\hat{C}_{y,k} : \mathcal{H}^{\otimes 2n} \to \mathcal{H}^{\otimes 2n} \\ &\hat{C}_{y,k} |i\rangle |0\rangle^{\otimes n} = |i\rangle |C_{y,k}(i)\rangle \end{aligned}$$

where $|i\rangle$ uses the k least significant bits from the first register. Now using the DLP circuit from Ref. [19] the following operator can be reproduced:

$$U_y |C_{y,k}(i)\rangle |0\rangle^{\otimes k} = |i\rangle |C_{y,k}(i)\rangle \tag{3.7}$$

Finally, this enables the following superposition state to be obtained up to the addition and removal of auxiliary qubits:

$$|0\rangle \xrightarrow{H^{\otimes k}} \sum_{i=0}^{2^k-1} |i\rangle \xrightarrow{\hat{C}_{y,k}} \sum_{i=0}^{2^k-1} |i\rangle |C_{y,k}(i)\rangle \xrightarrow{U_y^\dagger} \sum_{i=0}^{2^k-1} |C_{y,k}(i)\rangle := |C_{y,k}\rangle \tag{3.8}$$

To see why this feature map is useful for learning the dataset shown in Figure 3.2a, we begin by showing there exists a hyperplane in the respective feature space capable of separating the classes with a large margin.

For a promise problem f_s, the superposition state,

$$|\phi_s\rangle = \sum_{i=0}^{(p-1)/2-1} |g^s \cdot g^i \mod p\rangle \,, \tag{3.9}$$

[2]Using the multiplication operator $U_{k,p} |l\rangle = |k \cdot l \mod p\rangle$ and exponentiation operator $V_{k,p} |l\rangle = |k^l \mod p\rangle$, one can at once see that $|C_{y,k}(i)\rangle = U_{y,p}^i V_{g,p} |i\rangle$ [19]

consists of basis states that sweep the data points lying along one semicircle, as shown for the green data points in Figure 3.2b. This state represents a discriminative half plane since taking its inner product with a feature vector 4.1 gives

$$\begin{aligned}
|\langle\phi_s|C_{y,k}\rangle|^2 &= \sum_{i=0}^{2^k-1}\sum_{j=0}^{(p-1)/2-1} \langle y\cdot g^i|g^s\cdot g^j\rangle \\
&= \sum_{i=0}^{2^k-1}\sum_{j=0}^{(p-1)/2-1} \delta_{g^{i+\log_g(y)},g^{j+s}} = \sum_{i=0}^{2^k-1}\sum_{j=0}^{(p-1)/2-1} \delta_{i+\log_g(y),j+s} \\
&= |[\log_g(y), \log_g(y)+2^k-1]\cap[s, s+(p-1)/2-1]| \\
&= |\max(\log_g(y), s),\ \min(\log_g(y)+2^k-1, s+(p-1)/2-1])|
\end{aligned}$$

which as illustrated in figure **??** produces a different value depending on the class label of y

$$= \begin{cases} \Delta & \text{for } (1-\Delta)\% \text{ of } \{y|f_s(y)=+1\} \\ 0 & \text{for } (1-\Delta)\% \text{ of } \{y|f_s(y)=-1\} \end{cases}$$

where $\Delta = \frac{2^k}{(\frac{(p-1)}{2})} = \frac{2^{k+1}}{p-1}$ is the ratio of the arc length spanned by the feature vector to that of the feature halfspace.

To clarify the sense in which this $|\phi_s\rangle\langle\phi_s|$ is a hyperplane, we can alternatively represent both it and additionally any $|C_{y,k}\rangle\langle C_{y,k}|$ with respect to the 4^n-dimensional Pauli basis $\{P_\alpha\}$, identifying them with their Fourier coefficients, $\vec{w}_s = (w_{s,\alpha})$ and $\vec{y} = (y_\alpha)$, respectively, as defined using the trace inner product from **??**. By the exact same procedure as in **??**, the Hilbert-Schmidt inner product of the space $C2^n2^n$ is thus equivalent to the Euclidean space $R4^n$:

$$\langle|\phi_s\rangle\langle\phi_s|, |C_{y,k}\rangle\langle C_{y,k}|\rangle_{HS} = |\langle\phi_s|C_{y,k}\rangle|^2 = \sum_{\alpha=0}^{4^n-1} w_{s,\alpha}y_\alpha = \langle\vec{w}_s, \vec{y}\rangle_{\mathbb{R}}\,.$$

Taking $b = \frac{\Delta}{2}$, the inner products can be recast as Euclidean vectors:

- $\vec{w}_s\cdot\vec{y} - b = \frac{\Delta}{2}$ for $(1-\Delta)\%$ of $\{y|f_s(y)=+1\}$
- $\vec{w}_s\cdot\vec{y} - b = -\frac{\Delta}{2}$ for $(1-\Delta)\%$ of $\{y|f_s(y)=-1\}$.

where $\vec{w}_s$ and b can naturally be interpreted as the normal vector to a hyperplane and offset term respectively; at least $(1-\Delta)\%$ of the data points labelled +1 (-1) lie above (below) the hyperplane. Thus, this hyperplane separates the classes with a soft margin of $\Delta/\|w_s\|$.

Classical hardness of learning DLP_P Not only is the promise problem described above solvable for a quantum classifier, but it is provably unsolvable on a classical computer. This was shown in Ref. [13] in a proof by contradiction, that if a classical algorithm achieves at least $1/2+O(n^{-t})$ test accuracy on the promise problem DLP_P, then that same algorithm could solve the $DLP_{\frac{1}{2}}$ for more than $1/2+O(n^{-t})$ of the inputs and thus the DLP. The contradiction is that the DLP is conjectured to be classically difficult. Thus, implying that there does not exist a classical algorithm capable of attaining a test accuracy better than $1/2+O(n^{-t}) < 0.99$ and thus learning the promise problem.

Chapter 4

Determining properties of classically intractable promise problems admitting quantum speedup

Liu *et al.* were able to demonstrate a quantum speedup for their DLP-based promise problem DLP_P due to three characteristics that it exhibits:

C.1 The DLP_P is classically intractable.

C.2 There exists a quantum feature space $\mathcal{V}$ in which the DLP_P becomes linearly separable. We say the DLP_P *admits* a linear representation in quantum feature space.

C.3 The quantum feature map $\varphi_{\mathcal{V}}$ and by extension the kernel function $K_{\mathcal{V}}$ associated with the quantum feature space $\mathcal{V}$ can be computed efficiently.

C.2 and **C.3** together imply that the DLP_P can be efficiently solved using a Quantum Kernel Estimation Support Vector Machine (QKE-SVM). Subsequently, **C.1** suggests that no analogous classical model exists that could solve the DLP_P better than just randomly guessing, therefore completing the claim for quantum speedup.

As summarised in figure 4.1, we will see that each of these three characteristics originates from some sufficient corresponding mathematical property of the DLP_P:

O.1 The DLP_P is equivalent to the DLP which is conjectured to be classically difficult.

O.2 The DLP_P belongs to a class of problems that have Statistical Zero Knowledge (SZK) proofs [20].

O.3 The $DLP_P \equiv DLP$ belongs to a class of problems known as the Abelian Hidden Subgroup Problems (AHSP) which are efficiently solvable using a quantum computer [19].

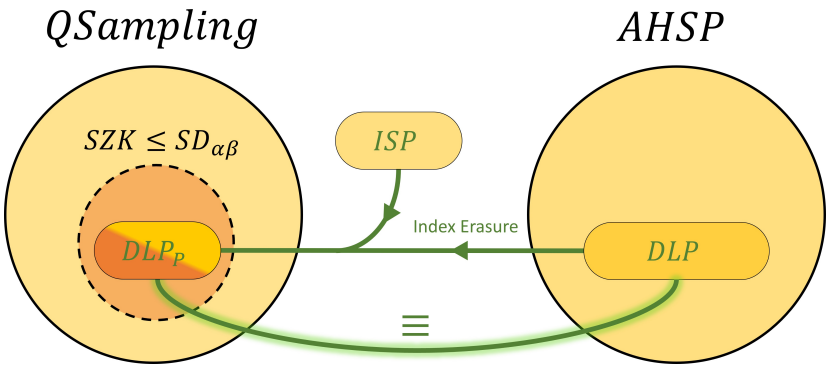


Figure 4.1: **The relationship of DLP_P with respect to several different classes of problems allows it to satisfy criteria C.1—C.3 required for admitting quantum speedup.** DLP_P is a member of SZK (orange dotted circle) and formulated as an SZK instance (orange wedge), it is *equivalent* to the (conjectured) classically intractable discrete logarithm problem DLP, which belongs to $AHSP$ (**C.1**). As a member of SZK, DLP_P can be further reduced to a (potentially harder) *QSampling* problem that defines the quantum feature space (**C.2**). The *QSampling* version of the problem (yellow wedge) can be efficiently solved using the $DLP \in AHSP$ for index erasure, together with a known initial state preparation routine ISP (**C.3**).

Investigating these properties would help to guide the search for other promise problems that could enjoy a quantum speedup in much the same way as the DLP_P. In what follows, we will derive the properties **O.1**-**O.3** from the special case of the DLP_P and then discuss them more generally. Then, we construct two promise problems which we show satisfy some but not all of these properties to see how that affects their suitability for obtaining a quantum speedup. Investigating the two promise problems will set the stage for constructing a third and final promise problem different to the DLP_P that does in fact demonstrate quantum speedup.

4.1 Statistical Zero-Knowledge: a class of problems admitting a representation in quantum feature space

What mathematical property of the DLP_P allows it to admit a linearly separable representation in quantum feature space? To answer this question, let us take a closer look at the quantum feature map used by Liu *et al.*:

$$|y\rangle \mapsto \sum_{i=0}^{2^k-1} |y \cdot g^i \mod p\rangle := |C_{y,k}\rangle \,. \tag{4.1}$$

We recall that the state $|\phi_s\rangle = |C_{g^s,\lfloor \log_2(p)\rfloor -1}\rangle$ represents a hyperplane which separates the data points with a large soft margin Δ:

$$|\langle \phi_s | C_{y,k}\rangle|^2 - \Delta/2 \begin{cases} \geq \Delta/2 & \text{for most } y \text{ s.t. } f(y) = +1 \\ \leq -\Delta/2 & \text{for most } y \text{ s.t. } f(y) = -1. \end{cases} \tag{4.2}$$

It is worth highlighting the fact that $|\langle \phi_s | C_{y,k}\rangle|$ represents the *overlap* between the basis states of $|C_{y,k}\rangle$ and $|\phi_s\rangle$.[1] Differential overlap between the $+1$ and -1 instances in (4.2) is what indicates the large margin feature of the hyperplane.

[1]For a concrete example, observe that $|\psi\rangle = |1\rangle + |3\rangle + |5\rangle$ and $|\phi\rangle = |3\rangle + |4\rangle + |5\rangle$ have an overlap $\langle \psi | \phi\rangle = 2$.

To generalise this notion of an *overlap* further, let us return to the quantum mechanical interpretation of quantum states as probability (amplitude) distributions:

$$|\psi\rangle = \sum_i \lambda_i |i\rangle \implies \text{Probability of measuring } |i\rangle \text{ is } |\lambda_i|^2 \tag{4.3}$$

In this perspective, $|C_{y,k}\rangle$ represents a uniform distribution over the values $\{y \cdot g^i \mid \forall i \in [0, 2^k - 1]\}$. We can reinterpret the *overlap* between $|C_{y,k}\rangle$ and $|\phi_s\rangle$ as a measure of similarity between the uniform distributions represented by each of $|C_{y,k}\rangle$ and $|\phi_s\rangle$.

To summarise, what we have just done is recast these feature states as probability distributions, and their inner products with respect to a hyperplane to mean the *distance* between the probability distributions. The reason we have done this will become clear in a moment.

We may define this probability distribution independently from the quantum state. Indeed, the image of the function

$$C_{y,k} : \{0,1\}^k \to \{0,1\}^n, \quad i \mapsto y \cdot g^i \mod p \tag{4.4}$$

has the associate probability distribution

$$D_{C_{y,k}} : \{0,1\}^n \to \{0,1\}, \quad x \mapsto \frac{|C_{y,k}^{-1}(\{x\})|}{2^k} \tag{4.5}$$

In words, $D_{C_{y,k}}(x)$ refers to the proportion of inputs i in the domain whose image is $C_{y,k}(i) = x$. Since $C_{y,k}$ is a bijection, there is a unique input corresponding to each output and thus $D_{C_{y,k}}(x) = \frac{1}{2^k} \, \forall \, i \in \{0,1\}^k$. This is precisely the uniform distribution that is represented by $|C_{y,k}\rangle$, or more explicitly:

$$|C_{y,k}\rangle = \sum_{x \in \{0,1\}^n} \sqrt{D_{C_{y,k}}(x)} \, |x\rangle \tag{4.6}$$

If instead of quantum states $|C_{y,k}\rangle$ and $|\phi_s\rangle$ we worked with their probability distribution representations, $D_{C_{y,k}}$ and $D_{C_{g^s, \lfloor \log_2(p) \rfloor - 1}}$, then the analogue of (4.2) will be the overlap between the two probability distributions:

$$|D_{C_{y,k}} - D_{C_{g^s, \lfloor \log_2(p) \rfloor - 1}}| \begin{cases} \geq \alpha & \text{for most } y \text{ s.t. } f(y) = +1 \\ \leq \beta & \text{for most } y \text{ s.t. } f(y) = -1. \end{cases} \tag{4.7}$$

for some $0 < \beta < \alpha < 1$. This problem is an instance of the more general problem known as *Statistical Difference* which is formulated as the following promise problem:

- Input: two classical circuits C_0 and C_1 mapping bit strings $\{0,1\}^n \mapsto \{0,1\}^m$ with distributions D_{C_i} obtained via sampling uniformly from the inputs $z \in \{0,1\}^n$. Two real numbers $0 < \beta < \alpha < 1$ satisfying $\alpha^2 > \beta$.
- Promise: if the distance between the probability distributions $|D_{C_0} - D_{C_1}| > \alpha$, accept. Otherwise if $|D_{C_0} - D_{C_1}| < \beta$, reject.

The upshot of this formulation is that there is an entire class of problems that reduces to this very problem. This is the class of problems admitting a *statistical zero-knowledge (SZK) proof*. Informally, a SZK problem consists of an interactive proof system whereby a *verifier* tries to verify that an element x satisfies some condition P via a two-way exchange of messages between itself and an all-knowing *prover*. For a problem to be categorised as SZK, however, the verifier must not learn anything during its exchange with the prover other than the fact that x satisfies P (or does not) [21]. Besides the DLP, the SZK contains many other problems including Quadratic Residuoisity (QR) and the Closest Vector Problem (CVP).

Of course, the connection of Statistical Difference to quantum algorithms requires that given circuits C_0 and C_1, the state (4.6) be efficiently prepared. This is not trivial, and it refers to the general task of Quantum Sampling or *QSampling* often encountered in quantum state generation. In QSampling, a classical boolean circuit C is given and we are asked to produce a corresponding quantum circuit which encodes the distribution D_C as the following superposition of states:

$$|C\rangle = \sum_{z\in\{0,1\}^m} \sqrt{D_C(z)}\,|z\rangle \tag{4.8}$$

We have thus established that $SZK \leq SD_{\alpha\beta} \leq QS$, and that solving a SZK problem is equivalent to solving the corresponding QSampling problem to which it can be reduced.

We saw that the reduction of SZK to QSampling in fact *induces* a quantum feature map represented by $|C\rangle$. As a result, any problem in SZK admits a representation in some quantum feature space and specifically the one that is induced by its reduction to QSampling. To see this more concretely, consider a learning problem based on the SZK. If the classical circuit C in Eq. 4.8 is parameterised by instances x of the dataset, then the procedure $x \mapsto |C_x\rangle$ can be interpreted as a quantum feature map.

Indeed, the quantum feature map constructed in Ref. [13] to solve the DLP-based dataset is directly based on the mapping induced by the QSampling reduction of the DLP_P. Ultimately, any problem in SZK admits a quantum feature map representation by virtue of its reduction to a QSampling problem.

4.2 Initial state preparation and Index erasure for efficient preparation of quantum feature maps

In the previous section we established that for every promise problem belonging to the class of Statistical Zero Knowledge problems, there exists a quantum feature space in which the promise problem becomes linearly separable. However, it is a different question entirely to ask whether there exists an *efficient* mapping into that induced quantum feature space. To investigate this, let us once again begin with Liu *et al.*'s special case: what mathematical property of the DLP_P allows the quantum feature *map* in Eq. (4.1) to be efficiently evaluated using a quantum computer?

The series of operations from Eq. (3.8) used to obtain the feature state $|C_{y,k}\rangle$ can be divided

into two main components:

$$\underbrace{|0\rangle \xrightarrow{H^{\otimes k}} \sum_{i=0}^{2^k-1} |i\rangle \xrightarrow{U_{\text{mod}}} \sum_{i=0}^{2^k-1} |i\rangle\, |C_{y,k}(i)\rangle}_{\text{Initial state preparation}} \overbrace{\xrightarrow{U_y^\dagger} \sum_{i=0}^{2^k-1} |C_{y,k}(i)\rangle}^{\text{Index erasure}} := |C_{y,k}\rangle \ .$$

To elucidate what the steps of initial state preparation and index erasure each entails, we will consider 3.8 in a slightly more general form. Let us replace the function $C_{y,k} : \{0,1\}^k \to \{0,1\}^n$ as described in (4.5) with an arbitrary function $f : X \to Y$. In this case, (3.8) is generalised to

$$\underbrace{|0\rangle \xrightarrow{H^{\otimes k}} \sum_{i=0}^{2^k-1} |i\rangle \xrightarrow{U_{\text{sift}}} \sum_{i\in X} |i\rangle \xrightarrow{U_f} \sum_{i\in X} |i\rangle\, |f(i)\rangle}_{\text{Initial state preparation}} \overbrace{\xrightarrow{U_{\text{uncomp}}} \sum_{i\in X} |f(i)\rangle}^{\text{Index erasure}} := |f\rangle \ . \tag{4.9}$$

We will now define these two steps with reference to the arbitrary function f, referring back to the special case of the DLP_P where $f = C_{y,k}$ only where it consolidates understanding.

Initial state preparation. As the name suggests, this step concerns producing the initial quantum state to which the algorithm in question will be applied. The desired form of the initial state is completely informed by the nature of the subsequent operations that will be done to it. In our case, the subsequent operation will be index erasure (see below), and for that, we require an initial superposition state consisting of all inputs entangled with their respective image under f:

$$\sum_{i\in X} |i\rangle\, |f(i)\rangle \ . \tag{4.10}$$

Eq. (4.9) indicates that generating the initial state above requires three operators:

1. $H^{\otimes k}$: Hadamard gates for inducing a uniform superposition over all basis states counting up to $2^k - 1$. This operation is efficient and ubiquitous in all of quantum computing.

2. U_{sift}: This operator pares down the superposition to include only basis states that are in the domain $X = \text{dom}(f)$. This is a non-trivial task and depends on the domain in question. For the DLP_P, we have that $X = \text{dom}(C_{y,k}) = \{0,1\}^k$ which is precisely the uniform superposition obtained from the previous step. In this case there is no need for paring down and the operator is simply the identity $U_{\text{sift}} = I$. Note that this will not always be the case (see section **??** for such a counterexample).

3. U_f: This operator applies the function f linearly to a superposition of input basis states. It is a well-known fact that an unitary oracle can be constructed efficiently for a function if it is efficiently computable on classical computers, which is almost always the case. In the case of the DLP_P, $f = C_{y,k} : i \mapsto y \cdot g^i \mod p$, which merely uses modular multiplication and exponentiation, both of which are efficient elementary operations on classical computers. Thus, an efficient oracle for such a function exists, $U_f = U_{\text{mod}}$ cf. 3.8.

Thus, in the task of initial state preparation the only operator whose efficient implementation depends non-trivially on the function f is U_{sift}. Assuming that we have successfully obtained (4.9), we are ready for the subsequent task of index erasure.

Algorithm 1 Index Erasure

Input The initial state $\sum_{i\in X} |i\rangle f(i)$. An oracle U_f with action $|i\rangle|0\rangle \xrightarrow{U_f} |i\rangle|f(i)\rangle$ implementing a one-way function $f : X \to Y$ that is M-to-one. An oracle $|x\rangle|y\rangle|q\rangle \xrightarrow{U_g} |x\rangle|y\rangle|q \oplus g_f(x,y)\rangle$ implementing the verifier $g_f(x,y) = [y = f(x)]$. A unitary operator U_X producing a uniform superposition over the input states.

Output The state $\sum_{i\in X} |f(i)\rangle$.

Note For clarity, the algorithm will begin with the output first from which the initial state will be reversibly reconstructed.

1:
$$\sum_{i\in X} |f(i)\rangle|0\rangle \xrightarrow{U_X} \sum_{i\in X}\sum_{j\in X} |f(i)\rangle|j\rangle$$

2: Up to the addition and removal of a third register set to $|+\rangle$, obtain:
$$\xrightarrow{U_g} \sum_{i\in X}\sum_{j\in X} (-1)^{g_f(f(i),j)} |f(i)\rangle|j\rangle$$

Note that out of the $|X|^2$ states, $|X|/M$ will be 'marked' with a negative phase, so the success probability of their measurement is $|X|/M$.

3: Perform amplitude amplification on the previous state:
$$\xrightarrow{U_{AA}} \sum_{i\in X} |f(i)\rangle|i\rangle$$

4: Note that following the steps (1)-(3) but in reverse yields the desired operation $U_{UNCOMP} := (U_X U_g U_{AA})^\dagger$.

Index erasure. This step refers to the task of uncomputing ('erasing') the input ('index') of f from a superposition of paired input-output states *i.e.* (4.10). Whether there exists an efficient unitary U_{uncomp} capable of erasing the index of the state (4.10) is in general nontrivial and completely dependent on the function f. Constructing U_{uncomp} typically involves finding its adjoint

$$|f(i)\rangle \xrightarrow{U^\dagger_{\text{uncomp}}} \sum_{j\in f^{-1}(\{f(i)\})} |j\rangle|f(i)\rangle \tag{4.11}$$

which in the special case of the DLP_P corresponds to

$$|C_{y,k}(i)\rangle \xrightarrow{U_y} \sum_{j\in C^{-1}_{y,k}(\{C_{y,k}(i)\})} |j\rangle|C_{y,k}(i)\rangle = |i\rangle|C_{y,k}(i)\rangle\ , \tag{4.12}$$

so that $U^\dagger_{\text{uncomp}} = U_y$. It is well-known that U_y is efficiently implementable because it solves the DLP problem which is a member of the Abelian hidden subgroup problems (AHSP), all of which are efficiently solvable using a quantum computer [19]. Abelian hidden subgroup

problems usually involve some *one-way* function f that is easy to compute on every input but whose preimage is computationally hard to calculate. In the case of the DLP_P, observe that $C_{y,k}$ is efficient to compute on inputs i, but calculating the preimage $C_{y,k}^{-1}(\{x\}) = C_{y,k}^{-1}(x) = \log_g(x)$ is conjectured to be classically difficult. For the general case of any arbitrary one-way function f that does not originate from the AHSP, the most naive approach involves explicitly computing $f(i) \forall i \in X$ and building a lookup table, which would take $\mathcal{O}(|X|)$. Instead, in Algorithm 1, we provide an algorithm for index erasure that takes $\mathcal{O}(\sqrt{|X|})$ number of steps, affording a quadratic speedup from the naive approach. Of course, the ideal case is if f belongs to the AHSP where index erasure can be computed exponentially faster, $\mathcal{O}(\log(|X|))$.

In summary, the quantum feature map for a SZK promise problem can be efficiently implemented if it admits efficient initial state preparation and index erasure as in the DLP_P. The SZK promise problem will have efficient initial state preparation if there exists an operator U_{sift} that efficiently generates elements of the domain in superposition. If the SZK promise problem also reduces to an Abelian Hidden Subgroup Problem, we also gain efficient index erasure.

In the next section, we will identify and investigate two new SZK promise problems. The first promise problem belongs to the AHSP but has no efficient initial state preparation, while the second promise problem has an efficient initial state preparation, but does not belong to the AHSP.

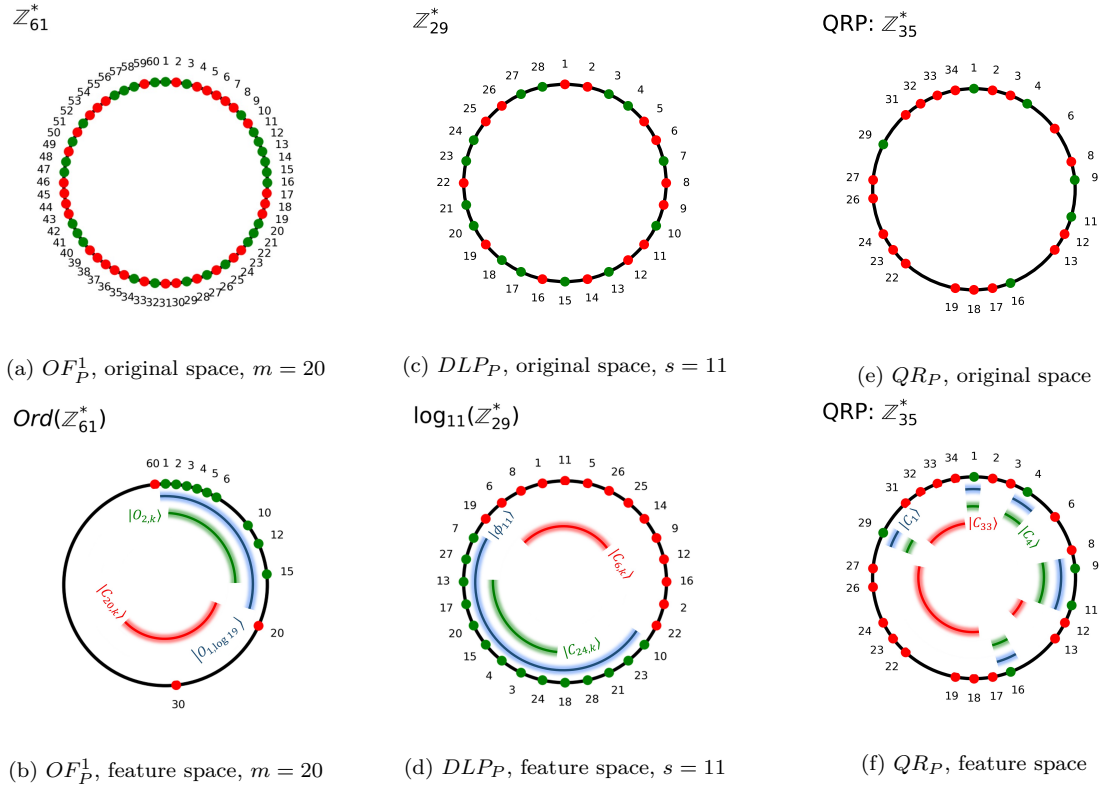
4.3 Two new classically intractable learning problems which are distinct from the DLP

In order to construct a binary labelled dataset based on QSampling that is *classically intractable* (CI) we need to identify some feature which is difficult for classical computers to compute. For a dataset with elements from a group, G, and a hidden set I, the schematic form of such a dataset should have the following labeling rule:

$$h(x) = \begin{cases} +1, & CI(x) \in I \\ -1, & CI(x) \in G - I\,. \end{cases} \tag{4.13}$$

In the case of ref [13], the dataset consists of elements from $\mathbb{Z}_p^*$, the multiplicative group of integers modulo p , whose elements x have a classically intractable property being that of their discrete logarithm element $CI(x) = \log_g(x)$.

We now provide two datasets based on classically intractable problems on which a quantum kernel estimator (QKE) performs polynomially better than all search-based classical learners [12]. We adopt the technique outlined in algorithm 1 to produce a quantum feature map that is able to correctly learn the dataset with quadratic speedup.



(a) OF_P^1, original space, $m = 20$

(c) DLP_P, original space, $s = 11$

(e) QR_P, original space

(b) OF_P^1, feature space, $m = 20$

(d) DLP_P, feature space, $s = 11$

(f) QR_P, feature space

Figure 4.2: **Three SZK promise problems which include the original DLP_P, and two new ones, OF_P^1 and QR_P, whose red/green (or $-1/+1$) instances are shown to be randomly distribution in their original space, but well separated in a corresponding quantum feature space.** The dots on the circles represent the inputs to the problem's corresponding labelling rule which outputs a label corresponding to red/green (or $+1/-1$). For example, in 4.2a, the data point $x = 1$ is green because $ord_{61}(1) = 1 \in [1, m-1] = [1, 19] \implies h_{OF^1}(x) = +1$. The bottom row displays where the data points are mapped to under the action of the classically intractable map $x \mapsto CI(x)$. So, in the case of 4.2a, $x = 1 \mapsto ord_{61}(1) = 1$ and $x = 1$ is mapped to the position of 1. The concentric arcs denote which points are swept in the superposition of the quantum feature state corresponding to a data point. For example, in dataset 4.2f, the superposition (shown in blue) obtained from $|1\rangle \mapsto |C_1\rangle$ sweeps all elements which are quadratic residues (*c.f.* Eq (4.26)), which corresponds to all the green dots. The blue, green and red colours of the arcs indicate that the superposition state represents a hyperplane, the feature state of some $+1$ instead, and the feature state of some -1 respectively. Note that in all cases, the green arcs (feature states) overlap much more with the hyperplane than do the red arcs, indicating the hyperplane has a wide-margin.

4.3.1 Order finding

In elementary number theory, the *order* of an integer x with respect to a prime number p is the smallest integer $r \in \mathbb{Z}_p^*$ satisfying:

$$a^r = 1 \mod p \tag{4.14}$$

Finding the order of an arbitrary integer x provided a large p is considered to be classically intractable and in fact equivalent to the difficult task of factorising integers which most of RSA encryption relies on. It is clear that the classically intractable feature $CI(x)$ of an integer x in this case refers to the order $CI(x) = ord_p(x)$.

The first promise problem that we will construct is in perfect analogy to the DLP_P but with

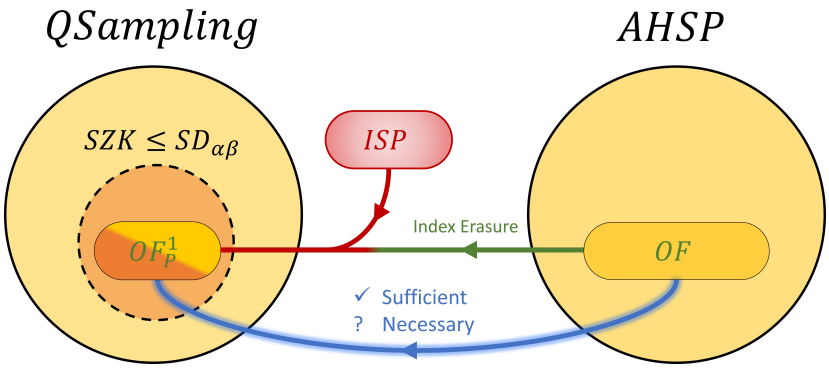


Figure 4.3: **The OF_P^1 is missing an efficient routine for initial state preparation (ISP) and thus C.3, which would have otherwise allowed it to satisfy criteria C.1—C.3 required for quantum speedup.** The class relations of the OF_P^1 are entirely analogous to that of the DLP_P outlined in Figure 4.1. It is unknown whether the OF_P^1, as an instance of SZK, is *equivalent* to the classically intractable task of order finding $OF \in AHSP$ — only that it is *as hard* as it.

two changes. In place of the discrete logarithm $\log_g(x)$, the labeling rule is determined by another classically intractable property known as the multiplicative order $ord_p(x)$. Additionally, the user-specified partition s (*c.f.* (3.5)) is replaced by m which is the median order of the multiplicative group $\mathbb{Z}_p^*$.[2]

$$h_{OF^1}(x) = \begin{cases} +1, & ord_p(x) \in [1, m-1] \\ -1, & ord_p(x) \in [m, p-1]\,. \end{cases} \tag{4.15}$$

The use of the median order m means that about half of the elements have $ord_p(x) \leq m\ (> m)$ and are labelled $+1$ (-1).

A quantum feature space wherein OF_P^1 is linearly separable. In analogy to the DLP_P, this promise problem also belongs to the SZK. By reducing it to a Statistical Difference instance, and furthermore, a problem in QSampling, we induce the follow quantum feature map:

$$|x\rangle \mapsto \sum_{\substack{i=0 \\ i+order_p(x)|p-1}}^{2^k-1} \sum_a |a \ : \ ord_p(a) = i + ord_p(x) \mod p\rangle := |O_{x,k}\rangle \tag{4.16}$$

for a choice of k. To intuit this feature map, recall that for the DLP_P, the feature map (4.1) sweeps an arc over all elements whose discrete logarithm lie in an interval. Similarly, the feature map 4.16 sweeps an arc over elements with orders in the interval $[ord_p(x),\ ord_p(x) + 2^k - 1]$. It follows that the hyperplane that partitions the dataset is the one that sweeps all the elements with orders $\leq m$, *i.e.* $|O_{1,\log(m-1)}\rangle$.

Efficient implementation of the quantum feature map We have shown that $OF1_P$ is linearly separable under the map 4.16 — but is this map efficiently implementable on a quantum

[2] m is given by the median of the set $ord_p(\mathbb{Z}_p^*)$. For example, for $p = 179$, 88 elements have order 89, 88 elements have order 178, 1 element has order 1, and 1 element has order 2. Thus, the median m order is 89.

computer? To answer this, we must ask whether the steps of initial state preparation and index erasure are both efficient. Observe that 4.16 may be obtained by applying index erasure on the state

$$\sum_{\substack{i=0 \\ i+order_p(x)|p-1}}^{2^k-1} \sum_a |ord_p(a)\rangle\, |a \ : \ ord_p(a) = i + ord_p(x)\rangle \xrightarrow{U^\dagger_{\text{ord}}} |O_{x,k}\rangle \tag{4.17}$$

The index erasure is carried out by $U^\dagger_{\text{ord}}$ which is defined as

$$|0\rangle\, |a\rangle \xrightarrow{U_{\text{ord}}} |ord_p(a)\rangle\, |a\rangle \tag{4.18}$$

in terms of its adjoint. U_{ord} solves the order-finding problem OF which, like the DLP, belongs to the Abelian Hidden Subgroup Problems. Thus, U_{ord} is efficiently implementable using a quantum computer. This is shown in the right-hand side of figure 4.3.

It remains to show that the initial state (4.17) can be efficiently computed. It can be shown that that the initial state can be produced efficiently if a uniform superposition of all possible orders in the interval $[ord_p(x),\, ord_p(x) + 2^k - 1]$ is given:

$$\sum_{\substack{i=0 \\ i+ord_p(x)|p-1}}^{2^k-1} |i + ord_p(x)\rangle \tag{4.19}$$

This is where OF^1_P encounters a problem: there is no known way to produce this initial state. From this we can infer that even if a promise problem has an efficient index erasure routine, this does not necessarily imply that the initial state required for index erasure can be prepared efficiently.

The OF^1_P represents an instance where a quantum feature space exists in which it is linearly separable, but no efficient quantum feature map into said space is known. For the OF^1_P, this is because no efficient algorithm is known for preparing the initial state 4.19. In the next section, we construct another promise problem with the same properties, except that it has an efficient initial state preparation routine, but none for index erasure.

4.3.2 Quadratic residues

In number theory, an element $x \in \mathbb{Z}^*_N$ is called a *quadratic residue* or perfect square if there exists a $y \in \mathbb{Z}^*_N$ such that $x = y^2 \mod N$. We say that xRN if x is a quadratic residue. The task of identifying whether a given x is a quadratic residue is classically difficult and to date no quantum operation exists that can solve it exponentially faster [22]. Due to the Chinese remainder theorem, it is known a priori that of the $\phi(N)$ elements of $\mathbb{Z}^*_N$ there is a total of $\phi(N)/4$ quadratic residues so that the proportion of quadratic residues is always a quarter. As a result, a benchmark ZeroR classifier can at best achieve a 75% accuracy on this dataset.

Our dataset will be sampled from $\mathbb{Z}^*_N$ which is the multiplicative group of integers modulo $N = p \times q$, where p and q are large primes. The number of elements in $\mathbb{Z}^*_N$ is given by Euler's

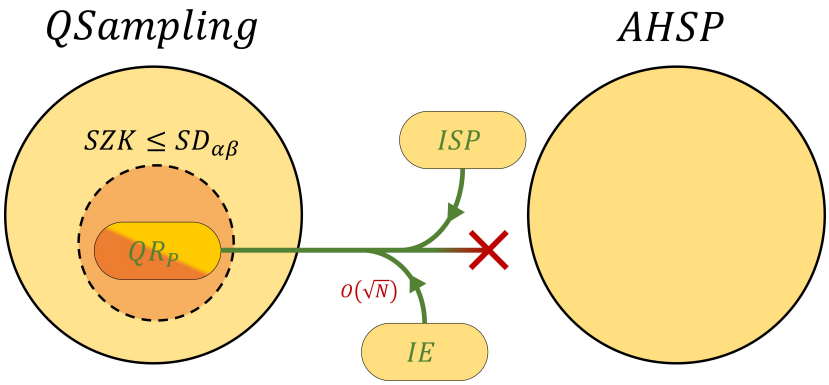


Figure 4.4: **The QSampling instance (yellow wedge) of the QR_P has no known way to be solved using a problem in $AHSP$, and thus fails to satisfy C.3**. The index erasure problem required to build the quantum feature map in (4.26) does make use of an efficiently implementable initial state preparation scheme (ISP), but is itself not efficient. Instead of an exponential speedup, we can still obtain a quadratic improvement by solving index erasure using the algorithm provided in Algorithm 2. Note that a guide to reading the other elements of this diagram has been outlined in Figure 4.1.

totient function $\varphi(N) = (p-1)(q-1)$ which counts the number of integers below N that are relatively prime to N[3].

With the definition above, let us define the following classically intractable dataset:

$$h_{QR}(x) = \begin{cases} +1, & CI_R(x) \text{ is true} \\ -1, & CI_R(x) \text{ is false}\,. \end{cases} \tag{4.20}$$

The requirement for a classically intractable property of an element is now fulfilled by the criterion of quadratic residuosity, that is:

$$CI_{QR}(x) := \text{Is it true that } x \text{ is a quadratic residue?} \tag{4.21}$$

Figure 4.2e displays a fully labelled set of $\mathbb{Z}_N^*$ for $N = 35 = 5 \cdot 7$ according to the rule above. Note that the occasional gap indicates an integer that is not coprime with N and is thus not a member of the set. Also notice that exactly 25% of the data points are labelled as quadratic residues, reflecting the proportion or quadratic residues in $\mathbb{Z}_N^*$ derived earlier.

A quantum feature space where in the QR_P is linearly separable. The QR_P is a well-known Statistical Zero Knowledge problem. When reduced from a SZK to a QSampling problem, Aharonov was able to show that QR_P is linearly separable in the following quantum feature space [20]:

$$x \mapsto \sum_{r \in \mathbb{Z}_N^*} |r^2 x \mod N\rangle := |C_x\rangle\,. \tag{4.26}$$

The quantum feature map (4.26) seems less familiar than those encountered in the DLP_P and OF_P^1. As always, we will show that $|C_1\rangle$ is a hyperplane that separates the $+1$ and -1 instances from each other with a large margin. Note that $|C_1\rangle = \sum_{r \in \mathbb{Z}_N^*} |r^2\rangle$ is by definition the superposition of all quadratic residues in $\mathbb{Z}_N^*$.

[3]for example, $N = 15$ is coprime with $1, 2, 4, 7, 8, 11, 13, 14$, so $\varphi(15) = 8$

Algorithm 2 Quadratic Residuosity state generation

Input Working register initalised to $|0\rangle$, an element $x \in \mathbb{Z}_N^*$, and the quantum oracle U_f implementing $f(x) = x^2 \mod N$.

Output The state $|C_x\rangle$.

1: Generate a uniform superposition over the elements of $\mathbb{Z}_N^*$ using the method of Ref. [23] for generating a prime state:

$$|0\rangle \xrightarrow{H^{\otimes n}} \sum_{i=0}^{N-1} |i\rangle \xrightarrow{U_{\text{gcd}}} \sum_{i=0}^{N-1} |i\rangle\, |gcd(i,N)\rangle \xrightarrow{U_{\text{AA}}} \sum_{i\in\mathbb{Z}_N^*} |i\rangle\, |1\rangle \equiv \sum_{i\in\mathbb{Z}_N^*} |i\rangle \tag{4.22}$$

Note that amplitude amplification is possible because the success probability of i and N being coprime is known a priori to be $\phi(N)/N = (p-1)(q-1)/pq \approx 1$ for large p, q [24].

2:

$$\xrightarrow{U_f} \sum_{i\in\mathbb{Z}_N^*} |i\rangle\, |i^2 \mod N\rangle \tag{4.23}$$

3: Using the technique of Algorithm 1

$$\xrightarrow{U_{\text{UNCOMP}}} \sum_{i\in X} |i^2 \mod N\rangle \tag{4.24}$$

4: Modular multiplication given by

$$\xrightarrow{U_\times} \sum_{i\in X} |i^2 \cdot x \mod N\rangle = |C_x\rangle \tag{4.25}$$

We now make use of two theorems which are proved in [20]:

Theorem 1. *$\forall r \in \mathbb{Z}_N^*$, $C_x(r) = r^2 x$ is a quadratic residue $\iff$ x is a quadratic residue.*

Theorem 2. *$x \in \mathbb{Z}_N^*$ is quadratic residues $\implies \forall z \in \mathbb{Z}_N^*\, \exists r \in \mathbb{Z}_N^*$ such that $z = r^2 x \mod p$*

From Theorem 1, we see that if x is a quadratic residue, then so are all the basis states of $|C_x\rangle$. Additionally, from Theorem 1 we know that for every quadratic residue z, there will be a corresponding basis state $|C_x(r)\rangle = |z\rangle$. Thus, $|C_x\rangle$ is a uniform superposition of all quadratic residues. But this is exactly the definition of $|C_1\rangle$, so $\langle C_1|C_x\rangle = 1$. Alternatively, if x is a quadratic nonresidue, then by Theorem 1, neither is $r^2 x\, \forall r$, so that $|C_1\rangle$ and $|C_x\rangle$ contain no overlap and $\langle C_1|C_x\rangle = 0$. To summarise:

$$\left|\langle C_1|C_x\rangle^2\right| = \begin{cases} = 1 & h_{QR}(x) = +1 \\ = 0 & h_{QR}(x) = -1. \end{cases} \tag{4.27}$$

Preparing the quantum feature map for the QR_P. Once again, we want to efficiently prepare (4.26), which in turn requires us to find efficient initial state preparation followed by

efficient index erasure. By inspection, we may obtain $|C_x\rangle$ from index erasure in the following way:

$$\sum_{i\in\mathbb{Z}_N^*} |i\rangle\, |i^2 \mod N\rangle \xrightarrow{U_{\text{uncomp}}} \sum_{i\in\mathbb{Z}_N^*} |i^2 \mod N\rangle \xrightarrow{U_\times} \sum_{i\in X} |i^2 \cdot x \mod N\rangle = |C_x\rangle \tag{4.28}$$

While the modular multiplications step, U_x, has a well-known efficient implementation, it is not clear how one would obtain U_{uncomp}. If we had the modular root operator,

$$|0\rangle\, |i^2 \mod N\rangle \xrightarrow{U_\surd} (|i\rangle + |N-i\rangle)\, |i^2 \mod N\rangle\,, \tag{4.29}$$

then we would easily have $U_{uncomp} = U_\surd^\dagger$. Indeed, an efficient operator $U_\surd$ would be able to solve the quadratic residue problem, the conjectured difficulty of which has served it well as a robust encryption protocol. Indeed, the QRP does not belong to the AHSP either, so finding an efficient quantum feature map for the state above on that basis is difficult. If we are willing to compromise on exponential speedup, we may still obtain a quadratic improvement by using the general-purpose index erasure algorithm in algorithm 1. In this way, we can compute U_{uncomp} with quadratic speedup $O(\sqrt{|\mathbb{Z}_n^*|})$ [22, 25].

Even though no algorithm for index erasure is known, we provide an efficient method to prepare the initial state $\sum_{i\in\mathbb{Z}_N^*} |i\rangle\, |i^2 \mod N\rangle$ as illustrated in 4.22. Overall, algorithm 2 presents a method for generating the quantum feature map for QR_P with quadratic speedup than what would be expected in a brute force calculation.

Much like the OF_P^1, the QR_P admits a quantum feature space wherein it become linearly separable, but has no known efficient quantum feature map into it. The QR_P has no efficient algorithm for index erasure, which highlights the importance of the index erasure instance belonging to $AHSP$ for it to be efficiently implementable using a quantum computer.

4.4 Summary and reflections

By closely examining the DLP_P, we have identified that promise problems belonging to a class known as SZK admit quantum feature spaces in which they become linearly separable. Then we saw that efficiently implementing a map into these spaces requires efficient routines for initial state preparation and index erasure. If the index erasure operator solves an Abelian Hidden Subgroup Problem (AHSP), then the problem is rendered classically intractable all while quantumly implementable. Otherwise, we derived Algorithm 2 which provides an implementation of general, non-AHSP operators for index erasure with *quadratic* speedup.

We then examined two new promise problems, QR_P and OF_P^1, which belong to SZK and thus have a quantum feature space in which they become linearly separable. However, the quantum feature maps for both of these promise problems ultimately prove difficult to be efficiently implemented.

In terms of index erasure, the QR_P requires a modular square root operator $U_\surd^\dagger$ which has no analogue in AHSP. So, the QR_P feature map remains intractable for classical *and* quantum

computers. In contrast, the index erasure step for the OF_P^1 corresponds to the task of order finding $U_{\text{ord}}^\dagger$ which is famously in AHSP and thus classically intractable but quantumly tractable.

As for initial state preparation, in the QR_P, the initial state is a superposition of elements in a multiplicative ring, for which we have provided an efficient algorithm 4.22. On the other hand, the OF_P^1 requires a superposition of multiplicative orders, which has no known implementation. It should be emphasised that the index erasure step is what determines what initial state must be prepared. Ultimately, it is the difficulty of index erasure that nullifies the QR_P and OF_P^1 as potential candidates.

We remind the reader that the ultimate goal of this thesis is to construct a classically intractable promise problem that is linearly separable in some efficiently implementable quantum feature space. To this end, let us characterise what our promise problem should exhibit:

1. Our candidate should belong to SZK so that such a quantum feature space exists.
2. From our two earlier case studies, it seems that index erasure complicates the implementation of a quantum feature map, as either the operation itself or the initial state it requires may be difficult to implement. Therefore, we want the implementation of the quantum feature map to not require index erasure. Instead, we want a quantum feature space where inputs and outputs can remain entangled.
3. We should be able to confidently conjecture that the problem is classically intractable.

It might seem too optimistic to claim we can evade index erasure, but a key distinction is that our promise problem need not be restricted to the form of 4.13. In the next section, we will see that by relaxing the definition of our labeling rule, we can discover a promise problem that admits all the characteristics above. Indeed, this promise problem will be classically intractable but solvable by a quantum machine learning model!

Chapter 5

A new classically intractable promise problem based on order finding

We now present one of the main results of this thesis. Guided by the framework we designed in the previous section based on Liu *et al.'s* work, we provide a new classically intractable promise problem that can be efficiently learned by a quantum classifier.

Our promise problem is called the OF_P^2 as, similar to its predecessor OF_P^1, it is based on order finding. In what follows, we will introduce some preliminary concepts required to understand the promise problem. We show that the OF_P^2 is at most as difficult as order finding, and argue as to why we may conjecture that it is classical intractable[1]. Then, we prove that the OF_P^2 belongs to SZK and as a consequence, admits a quantum feature space in which it becomes linearly separable. Unlike the promise problems OF_P^1 and QR_P from the previous section, we will discover that the quantum feature map for OF_P^2 will be efficient to implement, chiefly because it circumvents index erasure. Thus, learning OF_P^2 remains classically intractable all while admitting an efficiently implementable quantum feature map that can solve it. By way of simulations, we demonstrate that the quantum support vector machine (*c.f.* QKE-SVM) outperforms all classical models in learning this promise problem by accessing the quantum feature space.

5.1 Preliminaries

- A semiprime is an integer that factorises into exactly two odd primes $n = pq$. n is called a *large* semiprime if $n > 10^{10}$.
- $\mathbb{Z}_n^*$ is the multiplicative group of integers modulo n whose order is given by Euler's totient function $\varphi(n)$. For a semiprime n, $\varphi(n) = (p-1)(q-1)$.

[1] Liu *et al.* make a similar conjecture regarding the DLP_P. All references to classical intractability are in fact fundamentally conjectured as proving so requires solving the infamous $P = NP$ problem.

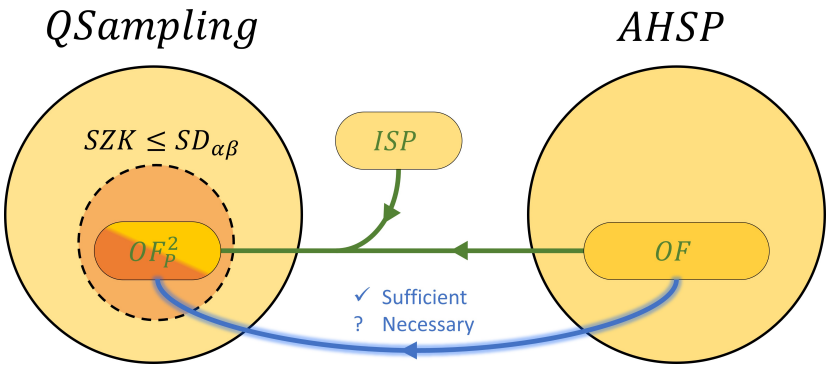


Figure 5.1: **The relationship of OF_P^2 with respect to several different classes of problems indicating that it is a classically intractable promise problem admitting an efficiently implementable quantum feature map in which it is linearly separable, *i.e.* it satisfies criteria C.1—C.3.** OF_P^2 is is a member of SZK (orange wedge) and is *at most* as difficult as the classically intractable problem of order finding OF, which belongs to $AHSP$. We conjecture that the OF_P^2 *is* classically intractable just as Liu *et al.* do for the DLP_P (**C.1**). As a member of SZK, OF_P^2 can be further reduced to a potentially harder *QSampling* problem (yellow wedge) that defines a quantum feature space (**C.2**). The *QSampling* version of the problem can be efficiently solved using $OF \in AHSP$ together with a known initial state preparation routine ISP (**C.3**).

- The multiplicative order of an element $a \in \mathbb{Z}_n^*$ is defined as the smallest integer r satisfying $a^r = 1 \mod n$, where we denote $r = ord_n(a)$
- The order $ord_n(a)$ for any $a \in \mathbb{Z}_n^*$ divides $\varphi(n)$.
- If an integer k has the prime factorisation $k = 2^a m$ where m is odd, then denote by $\mathcal{E}$ the function that counts the power of the factor of 2 occurring in the factorisation of k, $\mathcal{E}(k) = a$. We will refer to $\mathcal{E}$ as the *evenness* function.

Much of the results in this section will rely on the following theorem which we prove in this thesis for the first time:

Theorem 3. *If n is a semiprime integer with $\varphi(n) = 2^a m$ where m is an odd integer and $a \geq 2$, then the proportion of elements in $\mathbb{Z}_n^*$ with odd orders excluding the trivial element 1 is $\frac{1}{2^a}(1 - \frac{1}{m}) \to \frac{1}{2^a}$ as $m \to \infty$.*

Proof. We begin by first highlighting that $\mathbb{Z}_n^*$ is

- Finite as $|\mathbb{Z}_n^*| = \varphi(n)$ which is well-defined; and
- Abelian, as modular multiplication is commutative: $a \times b \mod n = b \times a \mod n$.

Since $\mathbb{Z}_n^*$ is a finite abelian group, by the Converse Lagrange Theorem (CLT), for all $k \,|\, \varphi(n)$, there exists at least one subgroup $H \leq \mathbb{Z}_n^*$ whose order is exactly k. In particular, since $m|\varphi(n) = 2^a m$, then there exists a subgroup H of order m. Now suffice it to prove that

$$x \in \mathbb{Z}_n^* \text{ has an odd order } \iff x \in H \leq \mathbb{Z}_n^* \text{ where } |H| = m, \text{ which exists by the CLT} \tag{5.1}$$

$\Leftarrow$ If $x \in H$, then the subgroup $\langle x \rangle$ generated by x must also be contained in H:

$$x \in H \implies \langle x \rangle \leq H \tag{5.2}$$

The number of elements in the $\langle x \rangle$ is precisely the order of x, $ord_n(x)$. By Lagrange's theorem, this generated subgroup must divide the group H

$$|\langle x \rangle| = ord_n(x) \mid |H| = m \tag{5.3}$$

But since m is odd, so must $ord_n(x)$ [21].

$\Rightarrow$ From the Chinese Remainder Theorem,

$$\mathbb{Z}_n^* \cong \mathbb{Z}_p^* \times \mathbb{Z}_q^*$$

Since p and q are prime numbers,

$$\cong C_{\varphi(p)} \times C_{\varphi(q)} \cong C_{p-1} \times C_{q-1}$$

By the fundamental theorem of arithmetic, rewrite $p-1$ and $q-1$ in terms of their prime factorisations $p-1 = 2^{a_1} p_1^{k_1} \dots p_i^{k_i}$, $q-1 = 2^{a_2} q_1^{l_1} \dots q_j^{l_j}$ so that

$$\varphi(n) = (p-1)(q-1) = (2^{a_1} p_1^{k_1} \dots p_i^{k_i})(2^{a_2} q_1^{l_1} \dots q_j^{l_j}) \tag{5.4}$$

$$= 2^{a_1} 2^{a_2} p_1^{k_1} \dots p_i^{k_i} q_1^{l_1} \dots q_j^{l_j} \overset{!}{=} 2^a m \tag{5.5}$$

It is a well-known fact that $C_{rs} \cong C_r \times C_s$ for coprime r and s. Using this result along with the factorisation in 5.5, we can further simplify 5.4 as

$$\begin{aligned} C_{p-1} \times C_{q-1} &\cong C_{2^{a_1} p_1^{k_1} \dots p_i^{k_i}} \times C_{2^{a_2} q_1^{l_1} \dots q_j^{l_j}} \\ &\cong C_{2^{a_1}} \times C_{p_1^{k_1}} \dots \times \dots C_{p_i^{k_i}} \times C_{2^{a_2}} \times C_{q_1^{l_1}} \dots \times \dots C_{q_j^{l_j}} \end{aligned}$$

which up to reordering is

$$\cong C_{2^{a_1}} \times C_{2^{a_2}} \times C_{p_1^{k_1}} \dots \times \dots C_{p_i^{k_i}} \times C_{q_1^{l_1}} \dots \times \dots C_{q_j^{l_j}} .$$

Note that a cyclic group C_{2^k} only has elements whose orders divide 2^k and thus are even, except for the trivial element $1 \in C_1 \leq C_{2^k}$:

$$x \in C_{2^k} \text{ has odd order} \iff x = 1 \in C_1 \leq C_{2^k} \tag{5.6}$$

So if $x \in \mathbb{Z}_n^*$ has odd order, then

$$x \overset{!}{\in} C_1 \times C_1 \times C_{p_1^{k_1}} \dots \times \dots C_{p_i^{k_i}} \times C_{q_1^{l_1}} \dots \times \dots C_{q_j^{l_j}} \leq \mathbb{Z}_n^* . \tag{5.7}$$

But observe that the second term is a subgroup of order $1 \cdot 1 \cdot p_1^{k_1} \dots p_i^{k_i} q_1^{l_1} \dots q_j^{l_j} = m$, which is precisely the subgroup H

From the above proof, it follows that of the $\varphi(n)$ elements in $\mathbb{Z}_n^*$ there are $m-1$ elements with odd orders, excluding the trivial element 1 which has order 1. Thus, the proportion of such elements is

$$\frac{m-1}{\varphi(n)} = \frac{m-1}{2^a m} = \frac{1}{2^a}(1-\frac{1}{m})\,. \tag{5.8}$$

□

Corollary 3.1. *Fermat semiprimes which are defined as $s = p'q'$ where $p' = 2^{2^{k_1}}+1$ and $q's = 2^{2^{k_2}}+1$ are Fermat primes have no elements with odd orders except the trivial element* 1.

Proof. For a fermat prime

$$\varphi(s) = (2^{2^{k_1}})(2^{2^{k_2}}) = 2^{2^{k_1}+2^{k_2}} = 2^{a'} \tag{5.9}$$

Thus, by Theorem 4, the proportion of elements with odd orders is $\frac{1}{2^a}(1-1/1) = 0$. □

5.2 The OF_P^2 promise problem

Our instances are large semiprmes $n = pq \in \mathbb{Z}$ and are labelled according to the following rule:

$$h(n) = \begin{cases} +1, & \mathcal{E}(\phi(n)) = 2 \\ -1, & \mathcal{E}(\phi(n)) \geq 4 \end{cases} \tag{5.10}$$

In words, h labels all semiprimes n whose corresponding $\varphi(n)$ is promised to attain an evenness of exactly 2, in which case it is labelled $+1$, or an evenness of ≥ 4, in which case it is labelled -1. Note that 2 is the minimum evenness that $\varphi(n)$ can attain for a semiprime n since

$$n = pq \implies \varphi(n) = (p-1)(q-1)$$

p, q are primes and odd by definition so $p-1$, $q-1$ are even

$$\begin{aligned} &\implies \exists k_1, k_2 \in \mathbb{Z}\ \varphi(n) = (2k_1)(2k_2) = 2^2(k_1k_2) = 2^{2+a}m,\ a \in \mathbb{Z}_{\geq 0} \text{ and } m \text{ odd} \\ &\implies \mathcal{E}(\varphi(n)) = 2 + a \geq 2 \end{aligned}$$

So, the $+1$-labelled semiprimes are those whose $\varphi(n)$ has a minimum evenness (of 2).

Conjectured classical intractability The task of computing the evenness of $\varphi(n)$ has the following reduction:

$$\begin{aligned} \text{Compute } \mathcal{E}(\varphi(n)) &\xrightarrow{\text{Reduces to}} \text{Compute } \varphi(n) = (p-1)(q-1) \\ &\xrightarrow{\text{Equivalent to}} \text{Factorise } n = pq. \end{aligned}$$

From this we can conclude that OF_P^2 is *at most* as hard as factorising integers, which is classically difficult. If it is in fact *as hard*, then OF_P^2 is classically intractable. However, it is much likelier that is *easier*, since it is very unlikely that computing $\varphi(n)$ reduces to calculating $\mathcal{E}(\varphi(n))$.

Nevertheless, it is not known whether the evenness of $\varphi(n)$ can be calculated through any other means that do not explicitly require $\varphi(n)$ itself. In light of this, we conjecture as to the classically intractability of the OF_P^2. Note that Liu *et al.* similarly make a conjecture regarding the classical intractability of DLP_P. As a result, the OF_P^2 is considered classically intractable to the same extent that the DLP_P is. In this way, our claim to the OF_P^2's classical intractability is no less rigorous than that of Liu *et al.'s* DLP_P.

5.2.1 OF_P^2 is an instance of SZK

The promise problem over prime numbers is actually a specific instance of `Statistical Difference` which you will recall belongs to SZK. We will provide some set-up before the proof. Fix a semiprime integer n satisfying $\phi(n) = 2^a m$ where m is an odd integer and $a \geq 2$. We will show that instances n satisfying $h(n) = +1$ known as *yes* instances will have a bounded statistical difference from a certain uniform distribution. Consider the classical circuit

$$C_x : [1, k] \to [1, k] \times [0, 1], \quad i \mapsto (i, P(ord_x(i))) \tag{5.11}$$

Notice that the image of this circuit varies depending on the choice of parameter x:

$$C_n(i) = \begin{cases} (i, \quad 0) & \text{over } (1 - \frac{\phi(n)}{n})k \xrightarrow[n\to\infty]{} 0 & \text{elements} \\ (i, +1) & \text{over } (1 - \frac{1}{2^a})(\frac{\phi(n)}{n})k \xrightarrow[n\to\infty]{} (1 - \frac{1}{2^a})k & \text{elements} \\ (i, -1) & \text{over } (\frac{\phi(n)}{n})(\frac{1}{2^a})k \xrightarrow[n\to\infty]{} (\frac{1}{2^a})k & \text{elements} \end{cases} \tag{5.12}$$

We will henceforth work in the limit of large n for simplicity. Even though the distribution of the images ultimately remains probabilistic, we know that for large k, the probability that any one integer is not coprime with n is $\approx \varphi(n)/n$ and similarly, that an element has an odd order $\approx 1/2^a$. From the distribution of the images in 5.12, we can deduce the probability of occurrence of each:

$$\begin{aligned} D_{C_n}((i, 0)) &= 0 \\ D_{C_n}((i, +1)) &= (1/k)(1 - \frac{1}{2^a}) \\ D_{C_n}((i, -1)) &= (1/k)\frac{1}{2^a} \end{aligned}$$

Most notably, the image of a circuit corresponding to a Fermat semiprime $s = p'q'$ is (*c.f.* Corollary 3.1)

$$C_s(i) = (i, +1) \quad \forall i \quad \implies D_{C_n}((i, +1)) = \frac{1}{k} \, \forall i \, . \tag{5.13}$$

Having laid out the image of the classical circuit $C_n(i)$, we will prove that the OF promise problem lies in the statistical difference dataset.

Theorem 4. *The order-finding promise problem OF_P^2 outlined in 5.10 belongs to the class of Statistical Zero Knowledge problems.*

Proof. Our general approach will consist of proving that the OF promise problem reduces to an instance of the Statistical Difference problem, which is a SZK-complete problem and thus

belongs to SZK itself. Consider two circuits $C_n(i)$ and $C_s(i)$ as defined in 5.12. The difference in the probability distributions generated by both these circuits varies drastically based on the type of instance that n belongs to:

$$
\begin{aligned}
|D_{C_n} - D_{C_s}| &= \frac{1}{2}\sum_{i=1}^{k}\sum_{j\in\{-1,0,+1\}} |D_{C_n}((i,j)) - D_{C_s}((i,j))| \\
&= \frac{1}{2}(\sum_{i=1}^{k}|D_{C_n}((i,+1)) - D_{C_s}((i,+1))| + \sum_{i=1}^{k}|D_{C_n}((i,-1)) - D_{C_s}((i,-1))|) \\
&= \frac{1}{2}(\sum_{i=1}^{k}|\frac{1}{k}(1-\frac{1}{2^a}) - \frac{1}{k}| + \sum_{i=1}^{k}|\frac{1}{k}(\frac{1}{2^a}) - 0|) \\
&= \frac{1}{2}(\sum_{i=1}^{k}\frac{1}{2^a k} + \sum_{i=1}^{k}\frac{1}{2^a k}) \\
&= \frac{1}{2^a} \\
&\begin{cases} = \frac{1}{4} & h(n) = +1 \\ \leq \frac{1}{16} & h(n) = -1 \end{cases}
\end{aligned}
$$

To summarise, taking $\alpha = 1/4$ and $\beta = 1/16$, the statistical difference of -1 and $+1$ instances with respect to C_s is

$$
\begin{aligned}
h(n) = +1 &\implies |D_{C_n} - D_{C_s}| \geq \alpha \\
h(n) = -1 &\implies |D_{C_n} - D_{C_s}| \leq \beta\,.
\end{aligned}
$$

Finally, observing that $0 \leq \beta \leq \alpha^2 \leq 1$, the promise problem defined by h is by definition an instance of Statistical Difference $SD_{\alpha,\beta}$ and consequently in SZK. □

5.2.2 An efficiently implementable quantum feature map solving the OF_P^2

Since the OF_P^2 belongs to SZK it naturally inherits a quantum feature space in which it become linearly separable. This feature state is given by the `QSampling` instance of 5.11:

$$
|n\rangle \mapsto \sum_{i\in\mathbb{Z}^*_{n,\,\leq k}} |i\rangle\,|P(ord_n(i))\rangle\ . \tag{5.14}
$$

where $\mathcal{P}(x) = x \mod 2$ returns the parity of integer x. We will see later that we can make this quantum feature space even more linearly separative of $+1$ and -1 instances if we encode the parity as a phase instead of a basis state:

$$
|n\rangle \mapsto \sum_{i\in\mathbb{Z}^*_{n,\,\leq k}} (-1)^{\mathcal{P}(ord_n(i))}\,|i\rangle = |\phi_n\rangle \tag{5.15}
$$

As suggested by the proof of Theorem 4, this feature map is able to linearly separate OF_P^2 as $|\phi_s\rangle$ represents a hyperplane when s is a Fermat semiprime, which we remind the reader only has elements with even orders (*c.f.* corollary 3.1):

$$
|\phi_s\rangle = \sum_{i\in\mathbb{Z}^*_{s,\,\leq k}} (-1)^{\cancelto{0}{\mathcal{P}(ord_n(i))}}|i\rangle := \sum_{i\in\mathbb{Z}^*_{s,\,\leq k}} |i\rangle \tag{5.21}
$$

Algorithm 3 Quantum feature map for order-finding

Input Working register initalised to $|0\rangle$, an instance of the dataset, n, which is a large semiprime. Also provided as input is the oracle $U_{\mathcal{P}}$ for checking parity

Output The state $|\phi_n\rangle$ from 5.15.

1: Generate a uniform superposition over the interval $[1, k]$:

$$|0\rangle \xrightarrow{H^{\otimes m}} \sum_{i=0}^{m-1} |i\rangle \xrightarrow{U_{AA}} \sum_{i=1}^{k} |i\rangle \tag{5.16}$$

Here $m = \lceil \log_2 k \rceil$.

2: Retain only the states corresponding to the elements of $\mathbb{Z}^*_{n,\leq k}$ using a similar method to Ref. [23] for generating a prime state:

$$\sum_{i=1}^{k} |i\rangle \xrightarrow{U_{\text{gcd}}} \sum_{i=1}^{k} |i\rangle |gcd(i,n)\rangle \xrightarrow{U_{\text{AA}}} \sum_{i\in\mathbb{Z}^*_{n,\leq k}} |i\rangle |1\rangle \equiv \sum_{i\in\mathbb{Z}^*_{n,\leq k}} |i\rangle \tag{5.17}$$

Note that amplitude amplification is possible because the success probability of i and n being coprime is known a priori to be $\phi(N)/N = (p-1)(q-1)/pq \approx 1$ for large p, q [24].

3: Apply the derandomised algorithm for order-finding:

$$\xrightarrow{U_{ord}} \sum_{i\in\mathbb{Z}^*_{n,\leq k}} |i\rangle |ord_n(i)\rangle \tag{5.18}$$

4: Apply $U_{\mathcal{P}}$ to mark states of odd order with a negative phase:

$$\xrightarrow{U_f} \sum_{i\in\mathbb{Z}^*_{n,\leq k}} (-1)^{\mathcal{P}(ord_n(i))} |i\rangle |ord_n(i)\rangle \tag{5.19}$$

5: Uncompute the second register:

$$\xrightarrow{U^\dagger_{ord}} \sum_{i\in\mathbb{Z}^*_{n,\leq k}} (-1)^{\mathcal{P}(ord_n(i))} |i\rangle = |\phi_n\rangle \tag{5.20}$$

The state $|\phi_s\rangle$ partitions semiprimes based on their labels:

$$|\langle\phi_s|\phi_n\rangle| = \begin{cases} \approx \frac{3}{4} - \frac{1}{4} = \frac{1}{2} & h(n) = +1 \\ \gtrapprox \frac{15}{16} - \frac{1}{16} = \frac{7}{8} & h(n) = -1 \end{cases} \tag{5.22}$$

This large-margin criterion, means that $+1$ and -1 instances can be confidently differentiated with a Qunatum Kernel Estimation Support Vector Machine (QKE-SVM) provided the quantum feature map is efficiently implementable. Indeed, we present algorithm 3 which efficiently prepares the state in 5.15.

5.2.3 Results and simulations

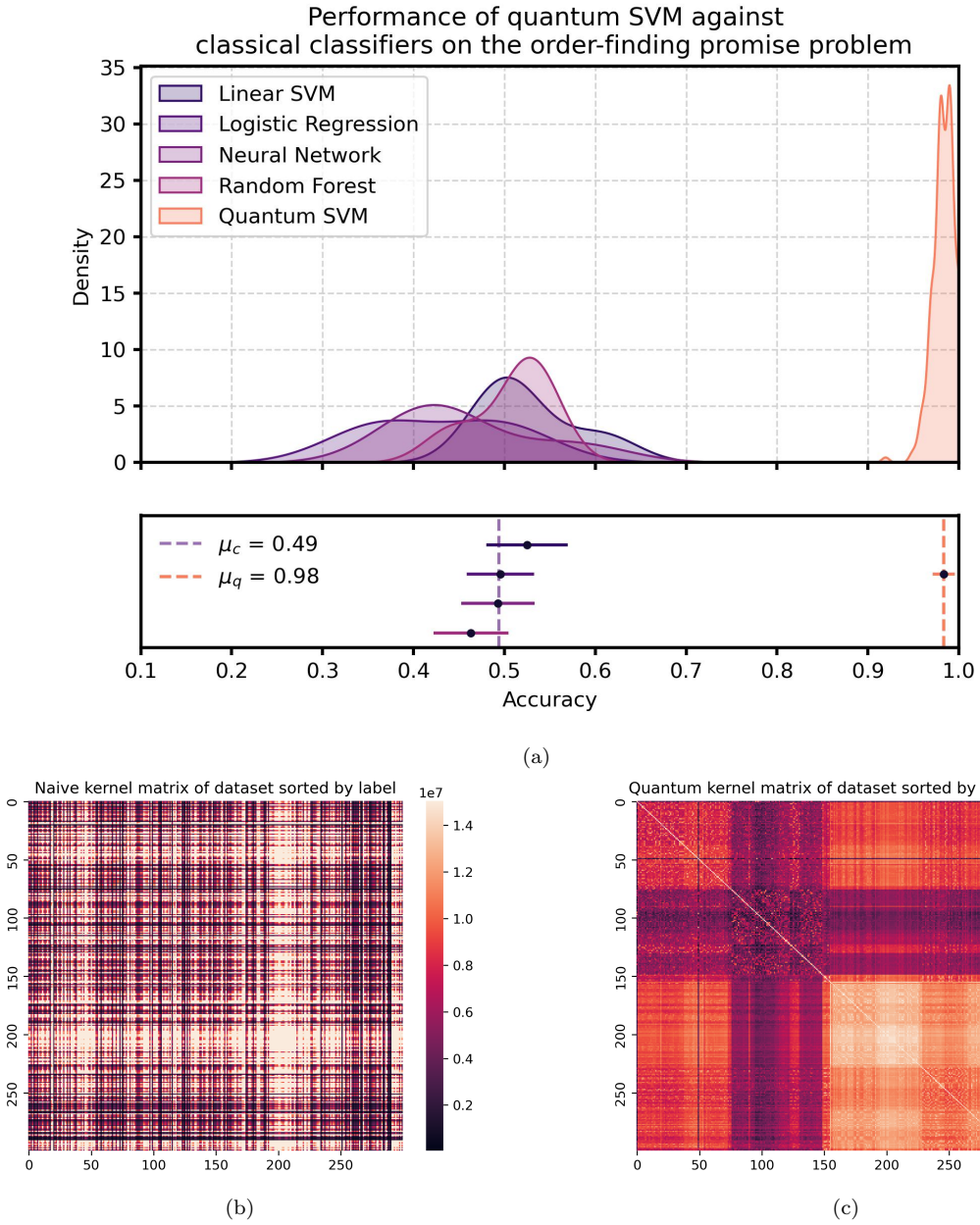


Figure 5.2: **Smoothed histograms displaying several classical machine learning models being drastically outperformed by the `QKE-SVM` at the task of OF_P^2 (5.2a). 5.2b and 5.2c respectively show kernel matrices corresponding to the classical and quantum feature maps that were used, revealing that the latter clearly separates instances based on their binary label unlike the former.** The bottom panel of 5.2a summarises the average accuracy of each model flanked by a σ error bar. The dataset on which the models were trained had an even distribution of $+1$ and -1 instances, thus the classical models on average performed no better than random guessing ($\mu_c = 49\%$) suggesting the classical intractability of the OF_P^2.

5.2.4 Discussion

In Figure 5.2a, we see the distribution of accuracies obtained by classical machine learning models compared to that of the `QKE-SVM` in classifying data points from the order-finding promise problem OF_P^2. Observe that the classical models collectively achieve an average accuracy of $\mu_c = 49\%$. Since the dataset used for training had equal proportions of $+1$ and -1 instances, this indicates that the classical models performed no better than mere random guessing, suggesting the classical intractability of the OF_P^2. In stark contrast to this, the `QKE-SVM` attains a near perfect average accuracy of $\mu_c = 98\%$ and with much lower variance than the classical models. The fact that the `QKE-SVM` has been able to efficiently solve the OF_P^2 to such high accuracy demonstrates quantum speedup as one could imagine the classical models requiring exponentially more training instances to be able to perform comparably well.

To see why there is such a wide gap in performance, it is instructive to compare the `QKE-SVM` with its classical counterpart, the `Linear-SVM`. Figures 5.2b and 5.2c are the kernel matrices used by the `Linear-SVM` and `QKE-SVM` respectively. The instances are sorted with respect to labels, so that the first 150 instances are labelled -1, and the second half are labelled $+1$. Recall that each entry corresponds to an inner product $K_{\mathcal{V}}(n_1, n_2)$ between two semprimes n_1 and n_2 as represented in some feature space $\mathcal{V}$. For the linear SVM, this feature space is merely the original space, and $K_{\text{classical}}(n_1, n_2) = n_1 n_2$. The corresponding kernel matrix 5.2b reveals no structure that is correlated with the class labels as the cross-hatching patterns appear randomly distributed. From this we can infer that this simple feature map used by the linear SVM cannot linearly separate between the $+1$ and -1 instances. Indeed, it is a feature of the OF_P^2 that no feature map efficiently implementable by a classical computer can linearly separate the dataset (*c.f.* the condition of classical intractability **O.1**). On the other hand, the quantum kernel matrix 5.2c used by the `QKE-SVM` reveals entry values that are clearly correlated with their class labels. The bright square occupying the entire fourth quadrant in contrast to the relatively darker patch of the first and third quadrants illustrates that in the quantum feature space (5.15), $+1$ instances are more similar than to each other than they are to -1 instances. Thus, there exists a wide-margin hyperplane (*c.f.* Equation (5.22)) that linearly separates the dataset. The difference between these two kernel matrices explains why the `QKE-SVM` is able to drastically outperform `Linear-SVM` and more generally, all classical models.

5.3 Summary

We have designed a new classically intractable promise problem based on order finding called OF_P^2 which admits quantum speedup. We argued that the OF_P^2 is classically intractable as it is unlikely to be solvable without first performing the classically intractable task of integer factorisation which is equivalent to order finding. We then showed that the OF_P^2 belongs to SZK and thereby admits a linearly separable representation in some quantum feature space. Then, we implemented the quantum feature map into this space, which was made efficient due to its not requiring index erasure. Through simulations, we evaluated the performance of classical models on the OF_P^2 against that of `QKE-SVM` and saw that while classical models can perform no better than random guessing, the `QKE-SVM` achieves a near perfect accuracy.

Chapter 6

Conclusion and Outlook

In this thesis, we have provided the only other instance of a classically intractable promise problem, the OF_P^2, which can be efficiently solved by a quantum machine learning model. More broadly, we have provided a general framework by which new classically intractable promise problems admitting quantum speedup can be constructed.

We obtained this framework by deconstructing Liu *et al.'s* DLP_P and identifying three sufficient properties that render a promise problem classically intractable all while solvable by a quantum machine learning model:

1. The problem belongs to the class of Statistical Zero Knowledge (SZK) problems. This induces a quantum feature space in which the promise problem becomes linearly separable.
2. Generating the quantum feature space is only as difficult as an Abelian Hidden Subgroup Problem (AHSP). This ensures the quantum feature space can be implemented efficiently using a quantum computer.
3. The problem itself is conjectured to be as difficult as an Abelian Hidden Subgroup Problem (AHSP). This confers the promise problem (conjectured) classical intractability.

We first investigated two promise problems, the OF_P^1 and QR_P, both of which failed to admit a quantum speedup as they satisfied only some, but not all of the above criteria. Along the way, we also provided a quadratic quantum speedup for the QR_P by designing an algorithm for the task of index erasure. We then combined the best parts of both OF_P^1 and QR_P to devise the OF_P^2, which does admit quantum speedup.

Future research can be based on constructing more novel promise problems using the framework above. The task of 'engineering' promise problems is admittedly highly contrived, especially when the field of machine learning concerns practical promise problems obtained from the real world, such as tumour classification or image recognition. Therefore, future work should be dedicated to using the framework above as a diagnostic tool for identifying *real world* — and not merely mathematical — datasets that can uniquely be solved by quantum computers.

References

[1] F. Arute, K. Arya, R. Babbush *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, p. 505, 2019.

[2] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[3] N. Liu and P. Rebentrost, "Quantum machine learning for quantum anomaly detection," *Physical Review A*, vol. 97, no. 4, p. 042315, Apr. 2018, arXiv:1710.07405 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1710.07405

[4] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nature Physics*, vol. 10, no. 9, pp. 631–633, Sep. 2014, arXiv:1307.0401 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1307.0401

[5] A. Sornsaeng, N. Dangniam, P. Palittapongarnpim, and T. Chotibut, "Quantum diffusion map for nonlinear dimensionality reduction," *Physical Review A*, vol. 104, no. 5, p. 052410, Nov. 2021, arXiv:2106.07302 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2106.07302

[6] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," Feb. 2020, arXiv:2001.03622 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2001.03622

[7] J. Jäger and R. V. Krems, "Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines," Jul. 2022, arXiv:2207.05865 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2207.05865

[8] I. Cong and L. Duan, "Quantum Discriminant Analysis for Dimensionality Reduction and Classification," *New Journal of Physics*, vol. 18, no. 7, p. 073011, Jul. 2016, arXiv:1510.00113 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1510.00113

[9] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Physical Review Letters*, vol. 100, no. 16, p. 160501, Apr. 2008, arXiv:0708.1879 [quant-ph]. [Online]. Available: http://arxiv.org/abs/0708.1879

[10] S. Arunachalam, V. Gheorghiu, T. Jochym-O'Connor, M. Mosca, and P. V. Srinivasan, "On the robustness of bucket brigade quantum ram," *New Journal of Physics*, vol. 17, no. 12, p. 123010, 2015.

[11] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019, pp. 217–228.

[12] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019. [Online]. Available: http://www.nature.com/articles/s41586-019-0980-2

[13] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Physics*, vol. 17, no. 9, pp. 1013–1017, Sep. 2021. [Online]. Available: https://www.nature.com/articles/s41567-021-01287-z

[14] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, "Covariant quantum kernels for data with group structure," Mar. 2022, number: arXiv:2105.03406 arXiv:2105.03406 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2105.03406

[15] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, Jun. 2008, publisher: Institute of Mathematical Statistics. [Online]. Available: https://projecteuclid.org/journals/annals-of-statistics/volume-36/issue-3/Kernel-methods-in-machine-learning/10.1214/009053607000000677.full

[16] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, "The complexity of quantum support vector machines," Feb. 2022, arXiv:2203.00031 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2203.00031

[17] R. Jozsa, "Quantum factoring, discrete logarithms, and the hidden subgroup problem," *Computing in science & engineering*, vol. 3, no. 2, pp. 34–43, 2001.

[18] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, "The complexity of quantum support vector machines," Feb. 2022, arXiv:2203.00031 [quant-ph]. [Online]. Available: http://arxiv.org/abs/2203.00031

[19] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition.* Cambridge University Press, 2010.

[20] D. Aharonov and A. Ta-Shma, "Adiabatic Quantum State Generation and Statistical Zero Knowledge," Jan. 2003, arXiv:quant-ph/0301023. [Online]. Available: http://arxiv.org/abs/quant-ph/0301023

[21] M. Tompa and H. Woll, "Random self-reducibility and zero knowledge interactive proofs of possession of information," in *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, Oct. 1987, pp. 472–482, iSSN: 0272-5428.

[22] H.-F. Wang, S. Zhang, Y.-F. Zhao, and K.-H. Yeon, "Quantum Mechanical Algorithm for Solving Quadratic Residue Equation," *International Journal of Theoretical Physics*, vol. 48, no. 11, pp. 3262–3267, Nov. 2009. [Online]. Available: http://link.springer.com/10.1007/s10773-009-0115-x

[23] J. I. Latorre and G. Sierra, "Quantum Computation of Prime Number Functions," Sep. 2013, arXiv:1302.6245 [quant-ph]. [Online]. Available: http://arxiv.org/abs/1302.6245

[24] M. Mosca and C. Zalka, "Exact quantum Fourier transforms and discrete logarithm algorithms," Jan. 2003, arXiv:quant-ph/0301093. [Online]. Available: http://arxiv.org/abs/quant-ph/0301093

[25] Y. Shi, "Quantum lower bounds for the collision and the element distinctness problems," in *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*, 2002, pp. 513–519, arXiv:quant-ph/0112086. [Online]. Available: http://arxiv.org/abs/quant-ph/0112086