2022-04

Information Theoretic Error Bounds on NISQ Learning Systems

B.Tech. Project

Sankalp Gambhir

Advisor **Prof. Sai Vinjanampathy**

Email sgambhir@iitb.ac.in

Abstract In this report, we review the classical optimization problem and attempts to solve it using quantum computers of currently achievable scales, called noisy intermediate-scale quantum (NISQ) computers. We present a mathematical review of the structure of involved spaces, and discuss the constraints on computational precision imposed by the architecture. Finally, we discuss entropic bounds from contexts in optimal control to generalize them to variational quantum algorithms (VQAs), identifying a computational bottle neck and establishing an information-theoretic uncertainty bound on the expressiveness achievable with practically implementable ansatzes for VQAs.

Contents

1 Introduction

There has been long-standing interest in constructing systems capable of learning from experience since even before computers in their modern form have existed. In the last few decades, with computing power skyrocketing exponentially coupled with leaping advances in theory of learning systems and statistical inference, these problems became tractable and eventually came into use ubiquitously. With applications ranging from image recognition systems for surveillance to identifying cosmic objects for astrophysical applications, they have found widespread adoption in industry and academia. These systems excel in problems where producing a precise mathematical model for the problem at hand is intractable, exploiting general techniques to instead infer a model from available data. With their advent, however, has come an ever rising need for computing power to facilitate their operation. This has found data centers of unprecedented scales consuming enormous amounts of power to provide the instant predictions we've come to rely on.

With snowballing energy and space requirements of classical computers in the form of GPU clusters and Application Specific Integrated-Circuits (ASICs), there has been a spark of interest in offloading this computation onto quantum computers, which, till recently, have largely remained a rare species spotted only in labs surrounded by helium-cooled superconductors and white-coated predators. Current scales of available quantum computers (100 physical qubits), however, still lack the power required to fully tackle these challenges while maintaining reliable error-levels or adding their own error checking and correction. This has motivated using quantum computers to run bottle necked computational subroutines with classical control systems. As such, these systems generally lack error correction, and thus earn themselves the title of 'noisy'. These form the basis of computation considered in this thesis, Noisy Intermediate-Scale Quantum (NISQ) computers.

However, making these computers fault-tolerant in practice has been a tall order, and seems to be at least a few years away. The strategy today is thus to explore how we can use NISQ systems to achieve a quantum advantage. However, working with NISQ, one must account for the limited number and connectivity of qubits, as well as the incoherence issues that limit circuit depth [1]. To tackle these issues and approach a quantum advantage, variational quantum algorithms (VQAs) have emerged as the leading candidate. As an analogue to classical machine learning techniques, they leverage the toolbox of optimization techniques, outsourcing the memory requirement and parameter control to a classical counterpart.

VQAs are characterized by parametrized quantum circuits, wherein the parameters are controlled by a classical computer running an optimization routine, updating them based on the measurement outcomes of the circuit. The technique has shown great promise and bypasses several of the issues arising from the lack of capability for error checking and correction, in turn arising from the small number of available qubits, in quantum computers expected to be manufactured within the next few years. The bulk of the computation requiring memory, such as parameter updation and gradient computation for optimization is generally performed on the classical controller. Effectively, VQAs trade the coherence required for independent quantum computation against access to classical memory.

Recently, these computers have been developed and used for experiments on supervised learning tasks, either by performing the learning task on a quantum computer, or by using on to speed up subtasks such as kernel-estimation [2, 3]. Recently, there has been significant progress towards solving currently looming problems such as barren plateus [4], and these models have also been show to be reasonably robust and error-tolerant in simulations [5].

Despite the promising advances in VQAs driven by access to a classical puppeteer, the benefits cannot be taken for granted. Communication with a classical system comes at its own cost, one paid in information entropy. These costs have been long studied in purely classical information channels, bounding the 'amount of data' that can be transferred over a channel between two parts of a system communicating with each other. This cost fundamentally limits the size of problems that can be computed by the system. Communicating over a classical channel, there is no basis to expect VQAs to be free from these chains either.

1.1 Outline of New Results

In this thesis, we study the architecture of a VQA, the processes, and mathematical structures involved in its functioning. Finally, we establish an information theoretic uncertainty theorem that bounds the expressiveness of a VQA ansatz choice in practice, limiting the problems computable within VQAs, by establishing a tradeoff with its trainability.

1.2 Structure

In section 2, definitions and relevant results in classical computing, physics, and quantum information are presented. section 3 reviews Variational Quantum Algorithms and their architecture, while section 4 discusses bounds on their expressiveness imposed by the architecture.

2 Preliminaries

2.1 Classical Computing

2.1.1 Optimisation Techniques

The discussion of optimization techniques in classical computing is a long and arduous one. We refer the reader to a common text on the matter for a detailed discussion [6, 7], while reviewing the general ideas briefly here.

The goal of an optimization problem, given (generally) a function $\mathcal{L} : X \to \mathbb{R}$ is to find the minimum value in its range, and sometimes an inverse in the domain, i.e., output a point $x_* \in X$, such that

$$x_* = \arg\min \mathcal{L}(x) \; ,$$

or the argument to $\mathcal{L}$ which minimizes it. In some problems, a local minima may suffice, and in others, a global minimum may be the requirement. Depending on the nature of the domain X, different techniques may be employed to find x_*. These include gradient descent and its reductions (finite element methods, etc.), Hessian-aided descent, stochastic gradient descent, among others.

The computation of the loss function can be computationally taxing with increasing number of parameters, making the optimization impractical to perform on conventional computing devices. With the advent of quantum computers, while independent quantum computation may seem out of reach, there have been attempts at using them as slot-in replacements for speeding up subtasks such as loss-function computation. We focus on variational quantum algorithms, where the classical computer offloads part of the task to a quantum module in this manner. The rest of the discussion is agnostic of the optimization routine, beyond, of course, the existence of one.

In a NISQ system, there is generally little to no attempt at error correction, and the general goal is to capitalize on what is possible with the short available coherence times, without devoting a majority of the system's resources to error checking and correction. As such, these systems are unable to support high-depth circuits with computationally involved analytical gradient based approaches. An effective optimizer in control of such a temporally-bound circuit should try to utilize techniques minimizing the number of measurements or function evaluations, as the relevant modules generally form the bottleneck of the computation [see 8, chapter II.D].

2.2 Quantum Regime

2.2.1 Hilbert Space

Definition 1. *A Hilbert space is a vector space $\mathcal{H}$ equipped with an inner product $\langle f, g\rangle \; \forall f, g \in \mathcal{H}$ such that the norm defined by*

$$\|f\| = \sqrt{\langle f, f\rangle}$$

turns $\mathcal{H}$ into a complete metric space [9].

Physical quantities — such as energy, momentum, and position — are represented as operators over a Hilbert space $\mathcal{H}$ to which the wavefunctions belong [10]. For our purposes, we will assume $\mathcal{H}$ to always have as its base field the field of complex numbers, $\mathbb{C}$.

Herein, the complex inner product is assumed to be linear in the second factor, i.e.,

$$\langle f, \lambda g\rangle = \lambda\langle f, g\rangle; \quad \langle \lambda f, g\rangle = \bar{\lambda}\langle f, g\rangle$$

$\forall f, g \in \mathcal{H}$ and $\lambda \in \mathbb{C}$.

For every bounded operator A acting on a Hilbert Space, there is a unique bounded operator A^* called its *adjoint* such that

$$\langle f, Ag \rangle = \langle A^* f, g \rangle \, .$$

We will assume relevant quantities to be linear operators $\cdot : \mathcal{H} \to \mathcal{H}$ with adjoints where necessary, [see 10, Appendix A] for details.

2.2.2 Quantum Computation

In a quantum system, 'computation' in its essence is jugglery of probability amplitudes of states using unitary actions. After action of unitaries as necessary, the coefficients are estimated using a measurement schema and post-processed to recover the computational result. See [11] for a detailed study.

2.2.3 State Construction and Embedding

To perform computation in the quantum regime, data first needs to be converted to a format in which it can be acted upon by quantum operators. This is again an embedding function, specifically from the input domain to the Hilbert space of quantum states for the system ansatz. See [12] for details.

2.2.4 Information Matrices and Distance Measures

Consider a parametrized distance function on quantum states

$$d(\boldsymbol{\theta}, \boldsymbol{\theta}') = d(\rho(\boldsymbol{\theta}), \rho(\boldsymbol{\theta}')) \, . \tag{1}$$

For close enough $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ we can write a Taylor expansion of the distance function. Clearly, the zeroth and first terms vanish, as the distance of a state from itself is zero, and it forms a minimum for the distance function. We have then the second order term

$$d(\rho(\boldsymbol{\theta}), \rho(\boldsymbol{\theta} + \boldsymbol{\delta})) = \frac{1}{2} \boldsymbol{\delta}^\top F(\boldsymbol{\theta}) \boldsymbol{\delta} \, , \tag{2}$$

with F representing the metric tensor

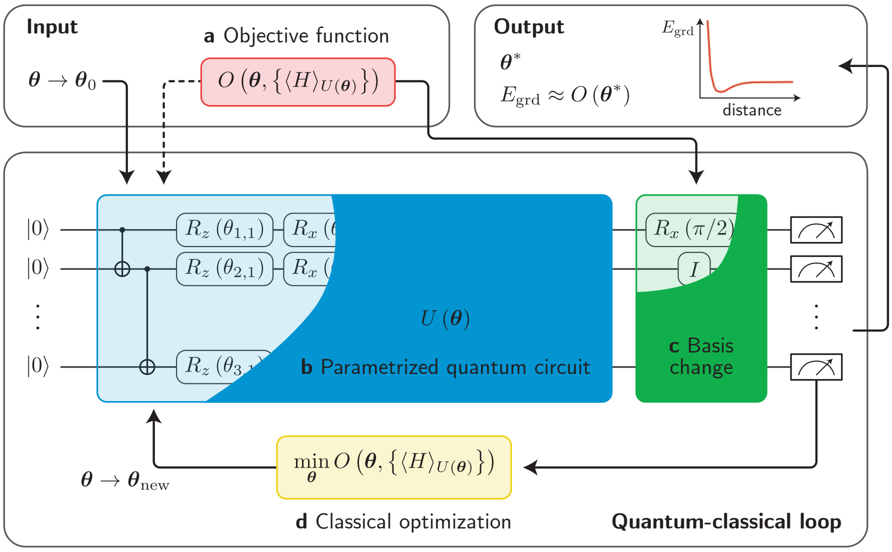

Input
$\boldsymbol{\theta} \to \boldsymbol{\theta}_0$
a Objective function
$O\left(\boldsymbol{\theta}, \{\langle H\rangle_{U(\boldsymbol{\theta})}\}\right)$
Output
$\boldsymbol{\theta}^*$
$E_{\mathrm{grd}} \approx O\left(\boldsymbol{\theta}^*\right)$
E_{grd}
distance
$|0\rangle$
$|0\rangle$
$|0\rangle$
$R_z\left(\theta_{1,1}\right)$
R_x
$R_z\left(\theta_{2,1}\right)$
R_x
$R_z\left(\theta_{3,1}\right)$
$U\left(\boldsymbol{\theta}\right)$
b Parametrized quantum circuit
$R_x\left(\pi/2\right)$
I
c Basis change
$\min_{\boldsymbol{\theta}} O\left(\boldsymbol{\theta}, \{\langle H\rangle_{U(\boldsymbol{\theta})}\}\right)$
$\boldsymbol{\theta} \to \boldsymbol{\theta}_{\mathrm{new}}$
d Classical optimization
Quantum-classical loop

Figure 1: *Diagrammatic representation of a Variational Quantum Algorithm (VQA) [taken from 8, Figure 2].*

$$F(\boldsymbol{\theta}) = \frac{\partial^2}{\partial\delta_i\partial\delta_j} d(\boldsymbol{\theta},\boldsymbol{\theta}+\delta)\Big|_{\delta=0} . \tag{3}$$

Therefore, F captures all the information necessary to define a distance metric. F is called the Quantum Fisher Information Matrix [13], and it is defined as

$$\left[F_{ij}\right] = 4\mathrm{Re}\left[\langle\partial_i\psi(\boldsymbol{\theta})|\partial_j\psi(\boldsymbol{\theta})\rangle - \langle\partial_i\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})\rangle\,\langle\psi(\boldsymbol{\theta})|\partial_j\psi(\boldsymbol{\theta})\rangle\right] . \tag{4}$$

3 Variational Quantum Algorithms

With the goal of optimizing learning computations using quantum computers in mind, we need an abstract idea of how to implement this connection. A *Variational Quantum Algorithm* (VQA) is any such system based on a proposed architecture for a classically controlled quantum computer [8]. Figure 1 presents the proposed architecture. The following subsection presents an expanded view of the computation.

3.1 Building Blocks

A VQA computation has 4 major components, as shown in Figure 1:

- objective function — the encoding of the problem at hand as an optimization,
- parametrized quantum circuit (PQC) — circuit encoding a unitary operator parametrized by classically controlled parameters $\boldsymbol{\theta}$,
- measurement scheme — the system performing basis changes and transferring outputs to the control system, and
- classical optimizer — a classical objective minimizer which controls the PQC.

These components form a modular computation model where each of the components can be swapped and improved individually to relieve bottle necks and adapt to the problem at hand, to control the expressiveness of the system or avoid treacherous optimization landscapes [14].

3.1.1 Objective Function

The *objective* or *loss function* [14] forms the target of the optimization problem at hand. This can be any function that can be encoded in an operational form, i.e., written as or decomposed into quantum operators. In most cases, this can be expected to be something akin to the Hamiltonian of a system [8], thus making the minimal, ground state energy, the optimization target. This may also be called the *parametrized cost* of the computation. Subject to the optimization constraints, the target of the system is then to find the optimal parameter input

$$\boldsymbol{\theta}_* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, p_0(\boldsymbol{\theta})) \, ,$$

where $p_0(\boldsymbol{\theta})$ represents the parametrized probability to measure the output in the state $|0\rangle$.

3.1.2 Parametrized Quantum Circuits (PQCs)

The module central to the design of a VQA is the parametrized quantum circuit, denoted by $U(\boldsymbol{\theta})$. It is the component of the circuit which performs the actual ‘computation’ and outputs the state that best meets the objective. It does so by acting on the input state a series of unitary transformations parametrized by controllable inputs. We assume the circuit to have an L-layered structure as

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\boldsymbol{\theta}_l), \quad U_l(\boldsymbol{\theta}) = \prod_{k=1}^{K} e^{-i\theta_{lk}H_k} \ , \tag{5}$$

where the index l indicates the layer, and the index k spans the traceless Hermitian operators $\{H_k\}$ that generate the space of unitaries for the chosen ansatz. Here, $\boldsymbol{\theta}$ decomposes as a set of vectors of parameters $\boldsymbol{\theta}_l$ for each of the indexed layers, which in turn map to individual parametrized unitary actions indexed by k. Finally, $M = K \cdot L$ gives the number of trainable parameters of the system [see 14, section II.A]. The experimental design to tune these parameters depends heavily on the hardware design chosen for the PQC, and may be mechanical, electronic, or optoelectronic in practice [15, 16, 17, 18, 19, 20, 21].

This general description of a PQC subsumes most ansatzes studied in literature [22]. These include the hardware-efficient ansatz [23], quantum alternating operator ansatz (QAOA) [24], Hamiltonian variational ansatz [25], quantum optimal control ansatz [26], among others [27, 28, 29]. These correspond to specific configurations of layer sizes and choices of the generators. This generic hardware structure allows us to use the well-formed foundations of landscape theory (subsection 3.2) to discuss advantages and limitations of VQAs independent of the specific problem being tackled with them.

The choice of generators is intimately tied to the reachable states of the system, and the landscape needed to be traversed to get there. This determines many things about the VQA, including, but not limited to, the problems solvable within the framework, and the time and hardware constraints required to train the system. For further detail, see discussion in subsection 3.2.

Assuming for now that the space spanned by the generators contains our target unitary, we proceed with the discussion of the computation. After the application of the PQC, the initial state $|\Psi_0\rangle$ is transformed as

$$|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\Psi_0\rangle \ . \tag{6}$$

Typically, the input state is chosen to be a zero-valued product state in the computational basis representation, i.e., $|\Psi_0\rangle = |00\ldots00\rangle = |0\rangle^{\otimes n}$. Other choices of the initial state may be made based on the problem requirements, possibly even to depend on some variational parameters itself as $|\Psi_0\rangle = P(\phi)|0\rangle^{\otimes n}$, with $P(\phi)$ a parametrized unitary, and ϕ the set of variational parameters. We discuss these as subjects of study in the future in section 5.

3.1.3 Measurement Scheme

to actually extract the information contained in the quantum state and compute the relevant objective function, one needs to compute its expectation value. This is where the quantum computer is needed to compute the expectation values by performing certain operations. For example, the Hadamard test is one way of obtaining expectation values. Extracting information from a quantum state means gaining information about the amplitudes of whichever basis states that we are interested in, but this amplitude is a measure of the probability of that state occurring. As such, it is not straight forward and we need to perform certain manipulations (by applying relevant quantum gates) so as to “read-out” the solution state of interest.

In practice, optimization techniques which require fewer measurements or function evaluations may be preferred, as a high sampling rate causes the measurement process to become a bottleneck for the system. Further, the optimization should be reasonably resilient to noise and the limited precision of the experimental equipment.

3.1.4 Parameter Optimization and Classical Control

After obtaining the loss function from the measurement and post-processing, a classical control system may treat it as output from a black box, at which point the optimizer may be oblivious of the fact the computation is sourced from a quantum computer, and apply any optimization technique of choice. Based on the chosen scheme, the controller readjusts the parameters.

The optimization technique may be a classical one — gradient based, Hessian based, etc — or quantum-aware, taking advantage of the hardware structure, or to combat specific issues such as quantum noise [30, 31, 32, 33, 34].

3.2 Quantum Landscape Theory

To discuss the limits of an architecture, and the issues associated with it, it is most critical to pursue a study of the mathematical structures involved in bringing about its functioning. In particular, the mathematical spaces one moves through in the process of optimizing using a VQA, and the morphisms that fly one over and through these landscapes take center-stage in the discussion at the boundary where the architecture breaks down.

In subsubsection 3.1.2, we suggested the problem of the target unitary not existing in the space reachable in our circuit configuration. In this section, we elaborate on this issue, and discuss the related problems of studying the loss landscape, how it emerges, and how it affects the optimization process. We begin with a review of Quantum Landscape Theory [see 14, chapter II.B].

To study the landscapes, one must first be aware of the spaces each of the objects relevant to the computation belong to. This is diagrammatically illustrated in Figure 2. First, the input parameter set $\boldsymbol{\theta}$ is seen as a vector in $\mathbb{R}^M$. The PQC then represents an embedding of $\mathbb{R}^M$ into the unitary space of appropriate size $d \in \mathbb{N}$, $\mathcal{U}(d)$. Its action on the input state is the map $U(\boldsymbol{\theta}) : \mathcal{H} \to \mathcal{H}$. Finally, the measurement scheme maps the output state to a real-valued loss, which is used by the optimizer to recompute the parameters. Succinctly, the action of the model arises from the following transformations

$$\mathbb{R}^M \to \mathcal{U}(d) \to \mathcal{H} \to \mathbb{R}\,. \tag{7}$$

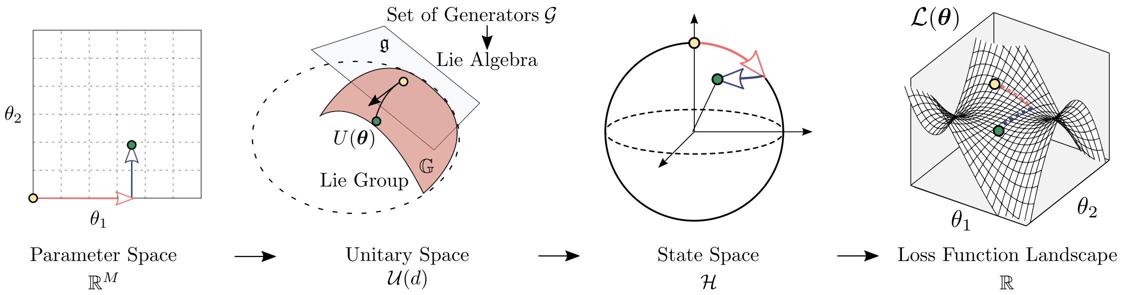


Figure 2: *Relevant mathematical spaces for VQA [taken from 14, Figure 2].*

3.2.1 Parameter Space to Unitary Group ($\mathbb{R}^M \to \mathcal{U}(d)$)

The first map, i.e., the map from the space of parameters to the unitary group, will be the focal point of the rest of this section. The unitaries generated by this map, and thus the chosen ansatz, are characterized by an object called the Dynamical Lie Algebra (DLA) of the system [see 35, chapter 3]. This represents the space formed by (Lie) closure of the individual operators in the architecture, under repeated application and commutation. These operators form the *generators* of the DLA.

Definition 2 (Set of Generators)**.** *Consider a PQC of the form Equation 5. The set of generators $\mathcal{G} = \{H_k\}_{k=0}^{K}$ is defined as the set (of size K) of the Hermitian operators that generate the unitaries in a single layer of $U(\boldsymbol{\theta})$.*

Definition 3 (Dynamical Lie Algebra (DLA))**.** *Given a set of generators $\mathcal{G}$, its DLA $\mathfrak{g}$ is defined as the span of its Lie closure, or the space generated by $\mathcal{G}$ after closure with repeated nested commutation. Mathematically,*

$$\mathfrak{g} = \mathrm{span}\,\langle iH_1, iH_2, \ldots, iH_k \rangle_{Lie}\,,$$

where $\langle S \rangle_{Lie}$ denotes the Lie or the nested-commutator closure of S.

The set of reachable unitaries is then a subset of the Lie group $\mathbb{G}$ generated by $\mathfrak{g}$,

$$\{U(\boldsymbol{\theta})\}_{\boldsymbol{\theta}} \subseteq \mathbb{G} \subseteq \mathcal{SU}(d)\,. \tag{8}$$

$\mathbb{G}$ can also be generated completely from the underlying Lie algebra as $e^{\mathfrak{g}}$.

It would seem at first glance that a configuration of generators should be chosen to be as expressive as possible, which is to have $\mathbb{G}$ be as close to $\mathcal{SU}(d)$ as possible, however, this often leads to trainability issues such as barren plateaus due to randomly chosen initial parameters [22, 36, 37]. As such, the ansatz is generally either chosen to make the problem convenient, i.e. problem-inspired ansatz [26], or to make the implementation convenient, i.e. hardware-efficient ansatz [38].

3.2.2 Unitary Group to State Space ($\mathcal{U}(d) \to \mathcal{H}$)

Recall that for the map from the unitary group to the state space, the unitary output from the first map acts on states in the input set. Specifically, choosing the input set to be a training set $\mathcal{S} = |\psi_\mu\rangle$. Then, the second map (now parametrized by μ) is defined as

$$U(\boldsymbol{\theta}) \mapsto U(\boldsymbol{\theta})|\psi_\mu\rangle \ . \tag{9}$$

The reachable set from each starting state is called its *orbit*. In many cases, when the states in $\mathcal{S}$ have certain symmetries, the DLA in turn decomposes as the direct sum of the subspaces invariant under the symmetries

$$\mathfrak{g} = \bigoplus_\nu \mathfrak{g}_\nu \ . \tag{10}$$

There is no restriction on whether the states in the training set share or respect any symmetries of the PQC itself. In this way, the DLA serves as a focal point to determine the expressiveness in terms of unitaries as well as the set of reachable states in the Hilbert space.

Next, we wish to see how the output state changes with varying parameters $\boldsymbol{\theta}$. So, consider an infinitesimal perturbation to the parameters $\boldsymbol{\delta} \in \mathbb{R}^M$, and we can then quantify the distance between the initial and perturbed state. Define $|\psi_\mu(\boldsymbol{t})\rangle = U(\boldsymbol{t})|\psi_\mu\rangle \ \forall t \in \mathbb{R}^m$. Writing the distance function (second order) as discussed in subsubsection 2.2.4

$$d(|\psi_\mu(\boldsymbol{\theta})\rangle, |\psi_\mu(\boldsymbol{\theta}+\boldsymbol{\delta})\rangle) = \frac{1}{2}\boldsymbol{\delta}^\top F_\mu(\boldsymbol{\theta})\boldsymbol{\delta} \ , \tag{11}$$

where $F_\mu(\boldsymbol{\theta})$ is the Quantum Fisher Information Matrix (QFIM) for $|\psi_\mu(\boldsymbol{\theta})\rangle$. The QFIM plays a crucial role in quantum-aware optimizers such as the quantum natural gradient descent [31, 32, 33, 34]. Further, the rank of the QFIM quantifies the number of independent parameters in the state space that changing the parameters can allow us to explore. The rank of the QFIM, observed over the whole parameter space, thus gives us a lower bound on the number of parameters that must be communicated to the PQC to explore the entire unitary space available to it.

3.2.3 State Space to Loss Landscape ($\mathcal{H} \to \mathbb{R}$)

Finally, the loss landscape generated by the composition map is characterized by the classically computed landscape of the real-valued loss function, i.e., using the $M \times M$ Hessian matrix with indexed entries

$$\left[\nabla^2 \mathcal{L}(\boldsymbol{\theta})\right]_{ij} = \partial_i \partial_j \mathcal{L}(\boldsymbol{\theta}) \,. \tag{12}$$

Computing the gradient and Hessian matrix allows us to form a quadratic model of the loss function, with the Hessian's eigenvectors characterizing curvature at each point. The rank of the Hessian once again gives us the number of independent directions explorable by change in parameters, emphasizing similarly how the QFIM functions as a measure of curvature in the state space.

4 Information Theoretic Limits

Having proposed a structure to solve learning problems with a NISQ system, we are ill-fated to then deal with the myriad of issues that arise with daring to use it. Barring issues specific to the quantum states, low coherence times, inaccurate measurements, et cetera, our structure itself imposes some bottlenecks on what we can achieve with it. In particular, the act of converting and transferring data between the representations used by the quantum and classical halves induces the issue of information transfer and limits upon it.

The idea of information theoretic bounds goes back to the father of information theory, Claude Shannon himself [39]. Work from Nyquist, Hartley, and Shannon [40] built up the structure of information theory to quantify the maximum 'amount of data' that could be transferred through a noisy communications channel and the changes to said effective 'amount' on the implementation of error correcting codes over the channel.

The analysis for the hybrid computing case has partially been performed in a general setting [41]. The authors suggest in the paper an extension to construction of unitaries, but do not explore it further. To the best of our knowledge, this has not been discussed in other literature since. Here, we continue the discussion for the specific case of VQAs with the goal to parametrize the discussion with circuit characteristics (depth, width, DLA) and discuss the bounds on VQA computation from the optimal control theorems presented in the paper.

4.1 Summary of Bounds on Quantum Optimal Control

A quantum system (target of control) can be presented as a dynamical equation

$$\dot{\rho} = \mathcal{L}(\rho, \gamma(t)) \, , \tag{13}$$

where ρ is the density matrix representing the current state of the system, $\dot{\rho}$ its time evolution, γ the externally applied control pulse, and $\mathcal{L}$, here, the resulting Liouvillian superoperator [see 42, section IV]. The same notation is used for the loss function earlier, and is kept here only to be consistent with the source. The two will not be used together in this thesis.

The dynamics are subject to the boundary condition $\rho(t = 0) = \rho_0$, and the unitary part of $\mathcal{L}$ must be generated by a Hamiltonian

$$\hat{H} = \hat{H}_D + \gamma(t)\hat{H}_C \, , \tag{14}$$

where $\hat{H}_D$ and $\hat{H}_C$ are the drift and control Hamiltonians respectively. The dynamics can be generalized to have several control Hamiltonians and corresponding pulses, but the extension is straightforward and skipped here for simplicity.

Now, for choices of the control pulse γ, define the set of reachable states as the set $\mathcal{W}$, a manifold with dimension $\mathcal{D}_{\mathcal{W}}$, which is a subset of the space of density matrices of dimension $\mathcal{D}_\rho$, with of course $\mathcal{D}_{\mathcal{W}} \leq \mathcal{D}_\rho$. Thus, given a goal state $\bar{\rho}$, and an initial state ρ_0 the problem is to find a (not necessarily unique) optimal control pulse $\bar{\gamma}$ such that it drives the initial state to a final state within an ϵ-ball around the goal state. This can be written as a functional minimization

$$\bar{\rho}(t) = \arg\,\min_{\gamma(t)} \mathcal{F}(\rho_0, \bar{\rho}, \gamma(t), [\lambda_i]) \, , \tag{15}$$

where the functional $\mathcal{F}$ quantifying the distance between states may also include constraints introduced via the Lagrangian multipliers $\{\lambda_i\}$.

To the end of this optimization, suppose one adjusts the control pulse with a classical channel. In the ideal noiseless case, Hartley's Law bounds the information transfer as

$$b_\gamma = T \Delta\Omega \kappa_s \, , \tag{16}$$

where T is the pulse duration, $\Delta\Omega$ the bandwidth, and κ_s is the bit depth of the pulse. With the control pulse γ's extreme levels as γ_{max} and γ_{min}, define $\Delta\gamma = \gamma_{max} - \gamma_{min}$. Finally, set the minimum variation to be $\delta\gamma$. Then,

$$\kappa_s = \log(1 + \frac{\Delta\gamma}{\delta\gamma})\,. \tag{17}$$

These parameters can be used to give an error bound on the achievable state $\left\|\rho - \bar{\rho}\right\| > \epsilon$, with

$$\epsilon \geq 2^{-\frac{T\Delta\Omega\kappa_s}{\mathcal{D}_{\mathcal{W}^+}}}\,. \tag{18}$$

For a detailed build up to the results, see [41].

4.2 Bounds on PQC Optimization

Similar to the Optimal Control scenario, the VQA architecture also presents the same infrastructural bounds. It stands to reason that a similar result should extend to generation of unitaries using a PQC. This is suggested in [41, Supplemental Material], but not explored further. We seek to establish lower bounds on the information theoretic error on learning the unitaries in terms of the circuit parameters — circuit width, depth, and choice of generators (ansatz).

Following the discussion in subsection 4.1, subsection 3.2, and [14] about QFIMs and entropy for a given number of parameters, we attempt to reconcile these two theories.

In general, $\mathcal{D}_{\mathcal{W}^+} \sim$ maximum achievable rank of QFIM, since both represent the over dimensionality of the reachable solution space. While [14, 36] suggest the existence of a 2-design may allow for this maximum achievable rank to scale linearly or polynomially with the number of qubits n, problems currently solvable within the 2-design framework remain to be found. For most ansatzes used in practice, to attain a usable level of expressiveness, the dimension of the problem is exponential in the number of qubits accessible. This is well within expectations due to the exponential scaling of the size of a Hilbert space in general.

$$T\Delta\Omega\kappa_s \geq \mathcal{D}_{\mathcal{W}^+} \cdot \log_2(\frac{1}{\epsilon}) \sim 2^n \log_2(\frac{1}{\epsilon}) \tag{19}$$

This, in turn, following Equation 18, establishes a bound on the minimum precision attainable with a given bandwidth, or conversely the bandwidth and/or time required to attain a given level of precision. It is clear to see this required bandwidth scales with $\mathcal{D}_{\mathcal{W}^+}$, and as such, scales, in general, exponentially with the number of qubits utilized.

The established bandwidth constraints have several consequences on the design of practical NISQ systems implementing VQAs, constraining the minimum size and capability of hardware required to solve certain problems. It affects most the general purpose hardware-efficient ansatzes which have been used due to their ability to solve a large set of problems, hence being expressive, whilst being easier to implement in practice. This is akin to using a universal gate to implement circuits of choice. However, with these results in mind, it may be more tractable to build problem-specific ansatzes, which may require hardware reconfiguration to solve different problems, but may be the only path forward to combat the exponential scaling of the Hilbert space and thus the bandwidth requirements.

Extending this result to usable bounds for the outputs requires further study of the maps discussed in subsection 3.2. A preliminary result over the state space is discussed in Appendix B.

5 Conclusion and Future Work

In this report we reviewed Variational Quantum Algorithms and optimization theory, finally establishing bounds on the expressiveness achievable by choice of ansatz in a VQA. These bounds fundamentally limit the set of problems finding whose solutions within the VQA framework may be tractable in practice. We are continuing work on studying specific ansatzes and the limits imposed on them to develop a precise idea of the intractable problems. The effects of this bound on the possibility of achieving a quantum advantage on NISQ systems remains unclear, and is a subject for further study. A more mathematically precise discussion of the landscapes generated within the framework, and a study of their transformations is expected to allow a more through understanding of their limits at each stage.

Acknowledgements

I am very grateful to Prof. Sai Vinjanampathy for his unrelenting support in matters academic and otherwise. This project would not have been possible without his guidance. I thank Karthik Dasigi and Drishti Baruah for their constant involvement in discussions regarding the project and report.

References

[1] M. Cerezo et al. 'Variational quantum algorithms'. In: *Nature Reviews Physics* 3.9 (Sept. 2021), pp. 625–644. ISSN: 2522-5820. DOI: 10.1038/s42254-021-00348-9. URL: https://doi.org/10.1038/s42254-021-00348-9.

[2] Vojtěch Havlíček et al. 'Supervised learning with quantum-enhanced feature spaces'. In: *Nature* 567.7747 (Mar. 2019), pp. 209–212. ISSN: 1476-4687. DOI: 10.1038/s41586-019-0980-2. URL: https://doi.org/10.1038/s41586-019-0980-2.

[3] Edward Farhi and Hartmut Neven. *Classification with Quantum Neural Networks on Near Term Processors*. 2018. DOI: 10.48550/ARXIV.1802.06002. URL: https://arxiv.org/abs/1802.06002.

[4] Kaining Zhang et al. *Toward Trainability of Quantum Neural Networks*. 2020. DOI: 10.48550/ARXIV.2011.06258. URL: https://arxiv.org/abs/2011.06258.

[5] Maria Schuld et al. 'Circuit-centric quantum classifiers'. In: *Phys. Rev. A* 101 (3 Mar. 2020), p. 032308. DOI: 10.1103/PhysRevA.101.032308. URL: https://link.aps.org/doi/10.1103/PhysRevA.101.032308.

[6] Stephen Boyd, Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[8] Kishor Bharti et al. 'Noisy intermediate-scale quantum (NISQ) algorithms'. In: *arXiv preprint arXiv:2101.08448* (2021).

[9] Giovanni Sansone. *Orthogonal functions*. Vol. 9. Courier Corporation, 1959.

[10] Brian C Hall. *Quantum theory for mathematicians*. Vol. 267. Springer, 2013.

[11] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[12] Seth Lloyd et al. *Quantum embeddings for machine learning*. 2020. arXiv: 2001.03622 [quant-ph].

[13] Johannes Jakob Meyer. 'Fisher Information in Noisy Intermediate-Scale Quantum Applications'. In: *Quantum* 5 (Sept. 2021), p. 539. ISSN: 2521-327X. DOI: 10.22331/q-2021-09-09-539. URL: http://dx.doi.org/10.22331/q-2021-09-09-539.

[14] Martin Larocca et al. 'Theory of overparametrization in quantum neural networks'. In: *arXiv preprint arXiv:2109.11676* (2021).

[15] JM Taylor et al. 'Fault-tolerant architecture for quantum computation using electrically controlled semiconductor spins'. In: *Nature Physics* 1.3 (2005), pp. 177–183.

[16] Zhirui Gong et al. 'Magnetoelectric effects and valley-controlled spin quantum gates in transition metal dichalcogenide bilayers'. In: *Nature communications* 4.1 (2013), pp. 1–6.

[17] AM Stoneham, AJ Fisher and PT Greenland. 'Optically driven silicon-based quantum gates with potential for high-temperature operation'. In: *Journal of Physics: Condensed Matter* 15.27 (2003), p. L447.

[18] Richard W Wagner et al. 'Molecular optoelectronic gates'. In: *Journal of the American Chemical Society* 118.16 (1996), pp. 3996–3997.

[19] Shawn Sederberg et al. 'Vectorized optoelectronic control and metrology in a semiconductor'. In: *Nature Photonics* 14.11 (2020), pp. 680–685.

[20] Britton WH Baugher et al. 'Optoelectronic devices based on electrically tunable p–n diodes in a monolayer dichalcogenide'. In: *Nature nanotechnology* 9.4 (2014), pp. 262–267.

[21] Dimitris G Angelakis et al. 'A proposal for the implementation of quantum gates with photonic-crystal waveguides'. In: *Physics Letters A* 362.5-6 (2007), pp. 377–380.

[22] Martin Larocca et al. *Diagnosing barren plateaus with tools from quantum optimal control*. 2021. arXiv: 2105.14377 [quant-ph].

[23] Abhinav Kandala et al. 'Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets'. In: *Nature* 549.7671 (2017), pp. 242–246.

[24] Edward Farhi, Jeffrey Goldstone and Sam Gutmann. 'A quantum approximate optimization algorithm'. In: *arXiv preprint arXiv:1411.4028* (2014).

[25] Dave Wecker, Matthew B Hastings and Matthias Troyer. 'Progress towards practical quantum variational algorithms'. In: *Physical Review A* 92.4 (2015), p. 042303.

[26] Alexandre Choquette et al. 'Quantum-optimal-control-inspired ansatz for variational quantum algorithms'. In: *Physical Review Research* 3.2 (2021), p. 023092.

[27] Stuart Hadfield et al. 'From the quantum approximate optimization algorithm to a quantum alternating operator ansatz'. In: *Algorithms* 12.2 (2019), p. 34.

[28] Linghua Zhu et al. 'An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer'. In: *arXiv preprint arXiv:2005.10258* (2020).

[29] Juneseo Lee et al. 'Progress toward favorable landscapes in quantum combinatorial optimization'. In: *Physical Review A* 104.3 (2021), p. 032401.

[30] Wim Lavrijsen et al. 'Classical Optimizers for Noisy Intermediate-Scale Quantum Devices'. In: (Apr. 2020). URL: https://www.osti.gov/biblio/1615327.

[31] James Stokes et al. 'Quantum natural gradient'. In: *Quantum* 4 (2020), p. 269.

[32] Bálint Koczor and Simon C Benjamin. 'Quantum natural gradient generalised to non-unitary circuits'. In: *arXiv preprint arXiv:1912.08660* (2019).

[33] Julien Gacon et al. 'Simultaneous perturbation stochastic approximation of the quantum fisher information'. In: *arXiv preprint arXiv:2103.09232* (2021).

[34] Tobias Haug and MS Kim. 'Natural parameterized quantum circuit'. In: *arXiv preprint arXiv:2107.14063* (2021).

[35] Domenico d'Alessandro. *Introduction to quantum control and dynamics*. Chapman and Hall/CRC, 2021.

[36] Zoë Holmes et al. 'Connecting ansatz expressibility to gradient magnitudes and barren plateaus'. In: *arXiv preprint arXiv:2101.02138* (2021).

[37] Jarrod R McClean et al. 'Barren plateaus in quantum neural network training landscapes'. In: *Nature communications* 9.1 (2018), pp. 1–6.

[38] Marcello Benedetti, Mattia Fiorentini and Michael Lubasch. 'Hardware-efficient variational quantum algorithms for time evolution'. In: *Physical Review Research* 3.3 (July 2021). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.3.033083. URL: http://dx.doi.org/10.1103/PhysRevResearch.3.033083.

[39] Claude Elwood Shannon. 'A mathematical theory of communication'. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[40] Ralph VL Hartley. 'Transmission of information 1'. In: *Bell System technical journal* 7.3 (1928), pp. 535–563.

[41] S Lloyd and Simone Montangero. 'Information theoretical analysis of quantum optimal control'. In: *Physical review letters* 113.1 (2014), p. 010502.

[42] Daniel Manzano. 'A short introduction to the Lindblad master equation'. In: *AIP Advances* 10.2 (Feb. 2020), p. 025106. ISSN: 2158-3226. DOI: 10.1063/1.5115323. URL: http://dx.doi.org/10.1063/1.5115323.

[43] Nello Cristianini, John Shawe-Taylor et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[44] Hanamantagouda P Sankappanavar and Stanley Burris. 'A course in universal algebra'. In: *Graduate Texts Math* 78 (1981).

[45] Corinna Cortes and Vladimir Vapnik. 'Support-vector networks'. In: *Machine learning* 20.3 (1995), pp. 273–297.

[46] Shengqiao Li. ‘Concise formulas for the area and volume of a hyperspherical cap’. In: *Asian Journal of Mathematics and Statistics* 4.1 (2011), pp. 66–70.

A Classification Problems

A.1 Learning Problem

Learning [43] can be broadly defined as attempting to learn the input-output pattern given sample data. For this thesis, we consider three major categories of learning problems:

- Binary Classification — input points in a chosen domain, and a binary output label for each point.
- Multi-Label Classification — input points in a chosen domain, and one of n labels as output for each point.
- Regression — input points in a chosen domain with real-valued output.

A.2 Classification Problem

We take as input elements $\{x_i\}$, generally called *feature vectors*, in a chosen domain X called the *feature space* and output an element from a finite set $L = l_i$ of labels.

The problem is called binary classification if $|L| = 2$.

Formally, we attempt to learn a function $f : X \to L$ given a set of inputs in the domain, and possibly paired output labels.

The problem proceeds in two manners given the form of inputs: if provided input-output pairs, the problem is called a *supervised learning problem*, while attempting to learn a set of labels given just (clustered) inputs is called *unsupervised* learning. We focus on supervised classification here.

The set of input-output pairs provided is called the *training data*.

Given the difficulty of working with discretized domains, the input domain is generally converted to be a subset of a Euclidean space, using a suitable *embedding function*.

A.3 Embedding

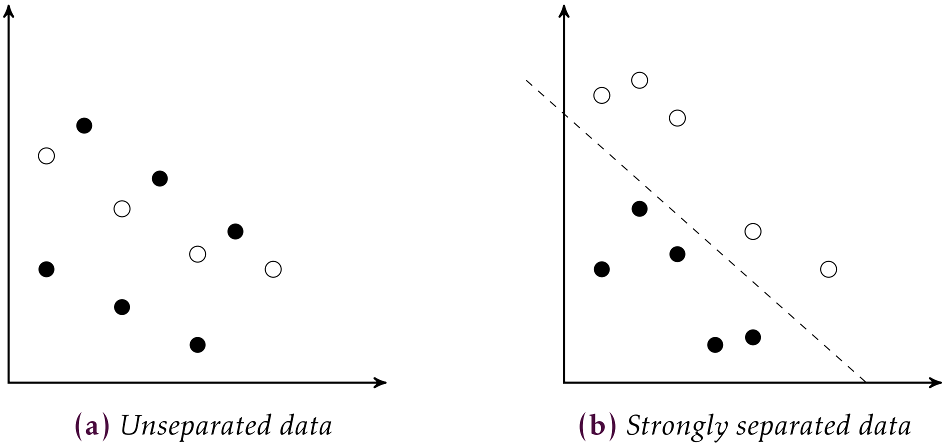
An embedding of X in Y is a function $f : X \to Y$ that is injective and such that image$(X) \subseteq Y$ has the same structure as X. The exact restrictions on the map to be structure-preserving depend on the structures of the domain and the co-domain [44]. It is denoted here as $f : X \hookrightarrow Y$.

For example, a topological embedding, i.e., the embedding of a topological space, will be restricted to preserve its associated structure of open sets. A field embedding, similarly, will be restricted to preserve the field operations + and ×.

For a given arbitrary feature space X, it is generally embedded into $\mathbb{R}^n$ for some n.

A.4 Linear Classification

Classification generally proceeds by producing linear functions as candidate labelling functions (supplemented with a discretization function) and fitting them to the training data. For simplicity, we first restrict the discussion to binary classifiers.



(a) *Unseparated data* **(b)** *Strongly separated data*

Figure 3: *The magic of the (strong) separation axiom.*

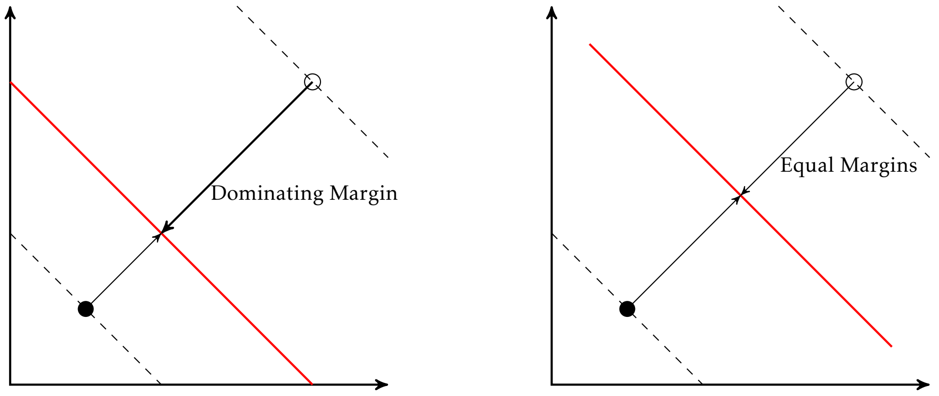
Given a set of points which, due to an embedding, may be assumed to be in $X \subseteq \mathbb{R}^n$, attempting to classify them may still be an arduous task if the spatial regions corresponding to the labels are intertwined. Thus, to make the problem tractable, we restrict the data to be *strongly* separated, i.e., assume that there always exists a set of hyperplanes (of size $|L|-1$) that isolates the points with each label within the domains created by intersection. In the binary case, this is just one label on either side of the hyperplane.

A.5 Support Vector Machine

A support vector machine is a classifier model which constructs a hyperplane or a set of hyperplanes in the feature space optimizing classifier separation depending on the objective [45].

We will synonymously use the terms 'Support Vector Machine' and that of its common model 'Maximal Margin Classifier', which is more appropriately what we use here.

As the name suggests, a maximal margin classifier SVM tries not only to construct a set of hyperplanes, but to find the set such that their margin from the data is maximized. This builds upon the intuitive idea of a good separator being further away from the given data points. See Figure 4.



(a) *Unoptimized margins* **(b)** *Maximal margin classifier*

Figure 4: *Illustration of different margins for hyperplanes.*

Formally, we characterize a hyperplane in $\mathbb{R}^n$ as a pair $(\boldsymbol{w}, b)$, with $\boldsymbol{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$, such that for all points $\boldsymbol{x}$ on the hyperplane

$$\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0 \,.$$

Geometrically, $\boldsymbol{w}$ is the vector normal to the hyperplane, and b is the bias or offset from origin.

Note that by moving to $\mathbb{R}^{n+1}$, we can convert the hyperplane to one without bias (passing through the origin)

$$\begin{gathered}\langle \boldsymbol{w}, \boldsymbol{x} \rangle + b = 0 \;, \\ \langle (\boldsymbol{w} \oplus [b]), (\boldsymbol{x} \oplus [1]) \rangle + 0 = 0 \;.\end{gathered}$$

which is the hyperplane $(\boldsymbol{w} \oplus [b], 0)$ in $\mathbb{R}^{n+1}$, and $[b]$ is the one element vector containing b. So, without loss of generality, we work with hyperplanes without bias.

Now, given the training dataset $(\boldsymbol{x_i}, y_i)$, with $y_i = \pm 1$, we can write constraints on $\boldsymbol{w}$ as

$$\forall i \; y_i \cdot \langle \boldsymbol{w}, \boldsymbol{x_i} \rangle > 0 \;, \tag{20}$$

that is, x_i is on the same side of the hyperplane as indicated by y_i as the sign of the inner product corresponds to the same.

By scaling $\boldsymbol{w}$ (without changing the hyperplane), we can construct the constraint system

$$\forall i \; y_i \cdot \langle \boldsymbol{w}, \boldsymbol{x_i} \rangle \geq 1 \;. \tag{21}$$

Since we are scaling $\boldsymbol{w}$, we choose an appropriate optimization target, its norm.

Since this is a constrained optimization, we write its Lagrangian

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha}) = \frac{1}{2} \langle \boldsymbol{w}, \boldsymbol{w} \rangle + \sum_i \alpha_i \left[y_i \cdot \langle \boldsymbol{w}, \boldsymbol{x_i} \rangle \right] \;, \tag{22}$$

where $\{\alpha_i\}$ are the Lagrangian multipliers. For the optimal solution, the Lagrangian is stationary, i.e.,

$$\frac{\partial \mathcal{L}(\boldsymbol{w}, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = 0 \, ,$$
$$\boldsymbol{w} + \sum_i \alpha_i y_i \boldsymbol{x_i} = 0 \, . \tag{23}$$

Substituting this expression for $\boldsymbol{w}$ in the Lagrangian itself, we get a Lagrangian dependent solely on the parameters $\{\alpha_i\}$ which we can optimize over. By minimizing this Lagrangian, we obtain an optimal $\boldsymbol{w}$ which is the maximum margin classifier.

With $\boldsymbol{w}$ fixed at its optimal value, we get a simple computational method to classify all new incoming points $\boldsymbol{x} \in \mathbb{R}^n$, given by

$$\operatorname{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle) \tag{24}$$

returning a label ±1 (or anomalously zero, if you happen to pick a point on the hyperplane, which can be remedied by making one side's boundary soft, by changing $>$ to $\geq$).

As the major 'quantum' modification, we will discuss how the constrained linear algebra computation is offloaded to a quantum circuit in section 3.

B Extension to Quantum Support Vector Machines

Given training data embedded as n-qubit quantum states $\{|x_i\rangle\}$ with corresponding labels $y_i = \pm 1$, a QSVM implemented as a VQA attempts to learn a unitary $U(\boldsymbol{\theta})$ such that

$$\operatorname{sgn} \, \langle 0|^{\otimes n} \, U(\boldsymbol{\theta})^* \, |x_i\rangle = y_i \forall i \, . \tag{25}$$

Setting $|w\rangle = U(\boldsymbol{\theta}) \, |0\rangle^{\otimes n}$ recovers the familiar classical SVM from Equation 24.

An ϵ error in the unitary correspondingly generates an error in $|w\rangle$ and generates a conical section (more generally, a hypersector of an n-hypersphere) as seen in Figure 5.

Picking a point randomly in the space outside the training data and attempting to classify it, we find an error probability proportional to the volume of the

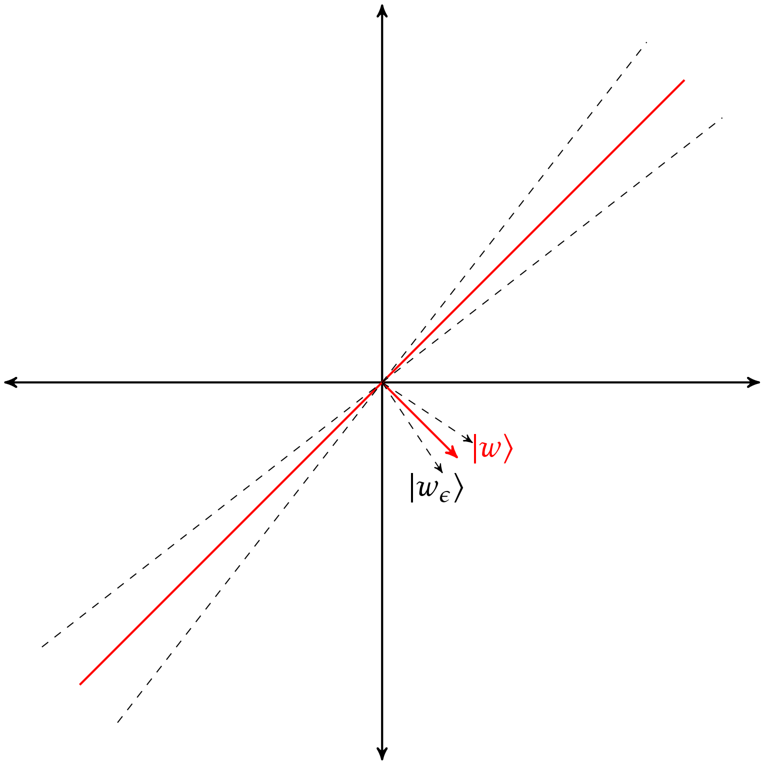


Figure 5: *Intuitive representation of error in the hyperplane normal vector.*

hypersector generated by the error in $|w\rangle$. We have using volume formulae from [46]

$$\begin{aligned} p_{\text{error}} &= \lim_{r\to\infty} \frac{2 \cdot V_{\text{sector}}(r)}{V_{\text{sphere}}(r)} \\ &= \lim_{r\to\infty} \frac{2 \cdot V_{\text{sphere}}(r) \cdot 0.5 \cdot I_{\sin^2\phi}(\frac{n-1}{2}, \frac{1}{2})}{V_{\text{sphere}}(r)} \\ &= I_{\sin^2\phi}(\frac{n-1}{2}, \frac{1}{2})\,, \end{aligned} \tag{26}$$

where I is the incomplete Beta function,

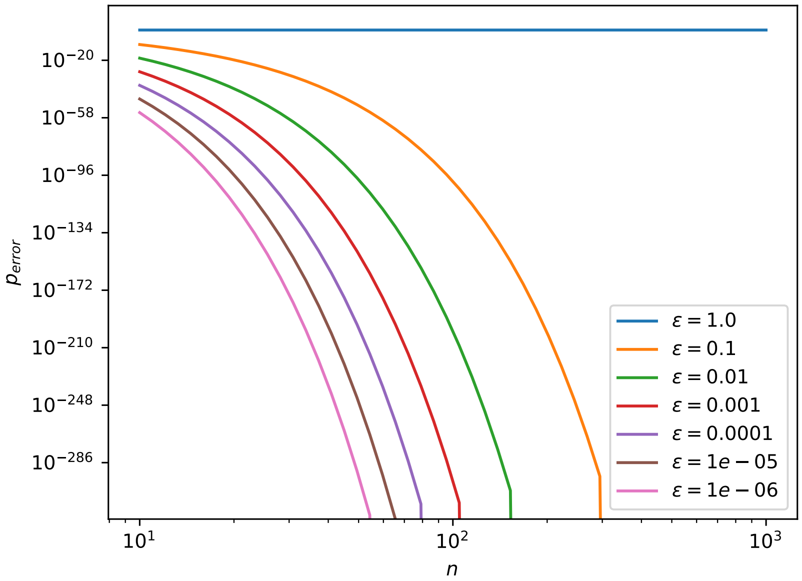


Figure 6: p_{error} *with change in* n *at several values of* ϵ.

$$I_x(a,b) = \frac{B(x;a,b)}{B(a,b)} = \frac{\int\limits_0^x t^{a-1}(1-t)^{b-1}\mathrm{d}t}{\int\limits_0^1 t^{a-1}(1-t)^{b-1}\mathrm{d}t}\,, \tag{27}$$

and ϕ is the angular distortion, and it is seen from $|w\rangle = U(\boldsymbol{\theta})|0\rangle$ that $\sin\phi \sim \epsilon$. Finally, we have,

$$p_{\text{error}} \sim I_{\epsilon^2}(\frac{n-1}{2}, \frac{1}{2})\,. \tag{28}$$

For a fixed ϵ, this error probability falls off quite quickly with n. See Figure 6 for plots of the probability with varying n at different values of ϵ. The form of the function suggests that while there is a fundamental limit to learning the unitaries, it may not always be a hindrance to be wary of, provided the system is of sufficiently high dimension.

However, it is to be noted that this analysis uses a fixed value of the unitary distortion, ϵ. In a real system, this is certainly not expected to be independent of the circuit width (number of qubits, dimension of state space). Preliminary calculations as described in section 4 suggest that ϵ is a function of the circuit area, i.e., the circuit width and depth product. This is related to the quantity M described in subsubsection 3.1.2. So, an increase in width correspondingly leads to a *decrease* in depth of the circuit to maintain the same error bounds for the unitary output.

The depth of the circuit has been heavily correlated to the barren plateau problem [14]. Increasing the depth of the circuit has been identified to remove spurious minimas from the problem and overparametrization beyond a certain depth (characterized by the QFIM) leads to deepening of present minimas, with a tradeoff towards barren plateaus [22]. This new result presents a second tradeoff. If we keep the error constant, and thus by the hypothesis, the area, the reduction in error by increase in width (upto a certain level, see Figure 6) would require a decrease in depth, leading to spurious minimas. This tradeoff is a subject of our current study.

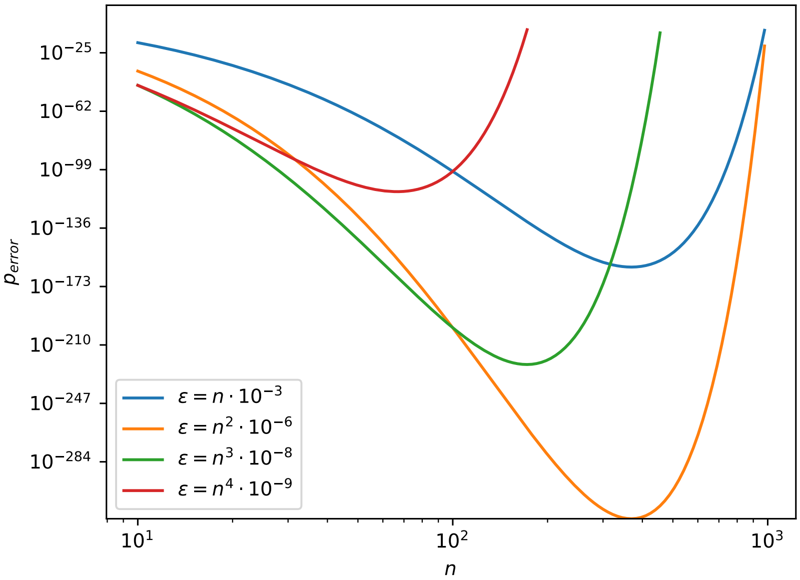


Figure 7: *p_{error} scaling assuming ϵ changes with n.*

For illustration, values of the error probability are plotted in Figure 7 with different scaling functions chosen for ϵ dependent on n, assuming a fixed circuit depth. Note that this calculation suggests the existence of a width (number of qubits) for a given depth and circuit configuration where generealisation error is minimized.

As we discussed in section 4, it is within reason to expect this scaling to be atleast polynomial, and for several choices of ansatzes, exponential as well. Existence of a 2-design has been suggested as leading to the possibility of linear scaling ansatzes. However, any interesting problems solvable within this framework remain to be found as of right now [14]. See [36] for a detailed discussion of t-designs and their consequences.

We note from the figure that the after attaining the minima, the error seems to explode combinatorially (due to the Beta function). However, this is in part expected to be a result of the small-angle approximation breaking down. We expect a more complicated relation to exist between sinϕ and ϵ, which were derived to be approximately equal for small values of ϕ and by extension of ϵ.