Quantum Machine Learning for Recognizing Gaussian Discord

by

Luyao Wang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering (Quantum Information)

Waterloo, Ontario, Canada, 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The quantum correlations measured by quantum discord have been paying significant attention since they are an essential resource for quantum information. This thesis focuses on the quantum discord in continuous-variable (CV) quantum computing systems, especially the quantum discord of the bipartite Gaussian states. We recognize the Gaussian quantum discord using quantum machine learning.

We review in detail the CV quantum computing systems that implement hybrid quantum-classical machine learning algorithms. Both Gaussian and non-Gaussian transformations are necessary to construct universal quantum computing systems. The structure of quantum discord can be studied using Gaussian states. The analytical solutions of Gaussian quantum discord are the labels of Gaussian states data set used for training and evaluating machine learning models. We presented the classical machine learning optimization algorithm back-propagating (BP) of the neural network to realize quantum Gaussian discord.

We proposed the supervised hybrid quantum-classical optimization performed on the variational quantum circuits for the Gaussian discord classification tasks. Moreover, we implemented a hybrid quantum-classical machine learning algorithm: Quantum Kitchen Sinks (QKS) for noisy intermediate-scale quantum (NISQ) devices. QKS uses the parametric variational quantum circuits to achieve non-linearly transformation from classical inputs to higher-dimensional feature vectors. Simulating the QKS on classical computers with the help of PennyLane, we demonstrated that the variational quantum circuits provide more excellent performance than the classical linear classification algorithm, successfully improving the classification accuracy from 70.12% up to 98.64%.

Acknowledgements

I would like to thank all the people who helped me complete my master studies.

First, I would like to thank my supervisor, Christopher Wilson, for giving me the chance to explore the quantum world. Chris's extensive knowledge helped me clarify my research direction, and he was very supportive when I encountered problems. Chris' patience and guidance are keys to supporting me through my studies.

Next, I would like to thank my group members at EQSL. Cindy is a great friend, and we often discuss a variety of topics, the progress of our work, the hardships we face or just the minutiae of our lives. I am also grateful to Jimmy, Sandbo, Jamal, and many others, whose exploration and hard work in quantum physics are the benchmarks I can learn.

I would also like to thank my friends, Katherine and Preston, for accompanying me through every tough time.

Lastly, I am thankful to my parents for supporting me to study abroad. They give me confidence whenever I am questioning myself. No matter what decision I make, they respect my decision and encourage me to be persistent. My family is my strongest support.

Table of Contents

List of Figures

Chapter 1

Introduction

With the rapid development of quantum information theory, the study of quantum correlations is becoming more critical because correlations are a fundamental resource in information processing. For a long time, the understanding of quantum correlations was limited to quantum entanglement, an important feature distinguishing between classical mechanics and quantum mechanics and a non-classical resource for quantum information processing. However, several recent studies have found that quantum entanglement does not characterize all quantum correlations in quantum systems, and separable (not entangled) states, conventionally corresponding to classical correlation, can also be used for quantum information processing with advantages beyond classical information processing [8, 46]. It has been proposed to classify the correlations into classical and quantum correlations and to use quantum correlations captured by quantum discord as resources for quantum information processing. Quantum discord is defined by the difference between two classically equivalent expressions for mutual information when the involved systems are quantum [43]. Researchers may use quantum discord to recognize the quantum nature of correlations and use the Gaussian states to explore the quantum discord inside the continuous variables quantum systems. The analytical solutions of quantum discord for Gaussian bipartite states are derived by overcoming the optimization challenge included in the quantum discord definition. We aim to uncover a general approach for estimating the quantum discord that may provide a good prediction.

In 1981, Richard Feynman proposed that quantum computers could simulate the quantum mechanical physical world [17]. In 1985, Feynman further proposed using quantum algorithms to solve practical but complicated problems [16]. In recent years, quantum computers have demonstrated their potential to push the computation boundaries in quantum chemistry, quantum optimization and machine learning.

Machine learning is one of the most intelligent branches in the field of artificial intelligence. In the 1980s, symbolic learning was the mainstream of machine learning; since the 1990s, the mainstream has evolved into statistical machine learning. In that time, machine learning has developed from purely theoretical and modelling research to applied research to solve real-life practical problems. Representative techniques for statistical learning are Support Vector Machine (SVM) and, more generally, kernel methods. After the general acceptance of SVM, kernel methods have gradually become one of the fundamental elements of machine learning.

R.S.Machalski et al. [39] proposed learning from samples, also known as generalized inductive learning, which covers supervised learning such as classification problems, unsupervised learning such as clustering problems, and semi-supervised learning. In recent years, deep learning has become a new trend in machine learning. The revolution in hardware technology and the improvement in computer speed have made it possible to employ algorithms with high complexity. The main application areas of deep learning are neural networks; in a narrow sense, deep learning is a neural network with many layers. Deep learning involves high complexity models and therefore requires well-tuned parameters to obtain high performance.

Quantum Machine Learning (QML) combines quantum computing and machine learning to implement QML algorithms on quantum computers and deal with quantum data. Machine learning is used to create models for complicated processes and discover the features of quantum systems; meanwhile, quantum computing has the potential to significantly speed up tasks.

The noisy intermediate-scale quantum (NISQ) devices contain 50-100 noisy qubits but give imperfect control over those qubits [47]. The implemented quantum circuits size limits the computational power of NISQ devices. We want to explore the maximum computational power of current NISQ devices since the NISQ devices are considered a milestone toward developing more powerful quantum devices and systems. The hybrid quantum-classical algorithms may lead the NISQ computers to approach their actual benefits since such quantum computers alone cannot yet demonstrate their computational speedup capabilities. Furthermore, the hybrid quantum-classical algorithms create a way to achieve quantum advantage on NISQ devices. With the rapid development of NISQ technology, many hybrid quantum-classical algorithms are driven and proposed, such as variational quantum eigensolver [45], quantum approximate optimization algorithm [13], quantum-assisted Helmholtz machines [5] and the variational classifier [14, 53].

1.1 Overview

The purpose of my research detailed in this thesis is to investigate and recognize the quantum discord of quantum Gaussian states, which are continuous-variable states, by quantum machine learning algorithms.

Chapter 2 introduces the basic building blocks of using continuous-variable (CV) quantum computing systems to encode quantum information. We then present Gaussian states, which play an important role in CV systems. We also cover various CV gates: Gaussian and non-Gaussian gates to achieve the universality of CV quantum systems and common CV measurements to measure the output of CV systems.

Chapter 3 reviews the derivation process of quantum discord, which measures the quantum correlation of CV quantum systems. We then show two approaches to obtaining the analytical solutions of quantum discord for Gaussian states. Those quantum Gaussian discord solutions will be the labels for the Gaussian state data set.

Chapter 4 presents the fundamental concepts of machine learning. We also introduce neural networks, which are the core of deep learning and the back-propagation (BP) algorithms used to optimize the parameters in neural networks. We also discuss the performance of BP in the Gaussian discord classification task.

In Chapter 5, we start by introducing the quantum machine learning concept and a machine learning library PennyLane used for implementing hybrid quantum-classical computations. We then show how to optimize a variational quantum circuit by training samples labelled by quantum Gaussian discord to classify new Gaussian state samples. Furthermore, we then describe a hybrid quantum-classical algorithm of QML called Quantum Kitchen Sinks (QKS) and discuss its classical components and quantum node structure and the performance for recognizing Gaussian quantum discord.

Chapter 2

Theoretical background

Continuous-variable (CV) quantum computing is an alternative paradigm for quantum computing, which contrasts with the discrete nature of qubit-based systems. In this section, we review the related content of CV systems and the standard Gaussian states. Meanwhile, both the Gaussian transformation and non-Gaussian transformation are important resources to construct a universal CV system. We would further utilize the CV quantum computation platform to implement a quantum machine learning algorithm.

2.1 Continuous-variable Quantum computation

Discrete-variable (DV) and continuous-variable (CV) systems are two different approaches to quantum information processing. DV systems, with basic element of the qubit, have finite and individually addressable states, analogous to classical computers. A CV system, whose elementary units are qumodes, has an infinite-dimensional Hilbert space with continuous operators, including momentum and position operators. The most fundamental CV system is the quantum harmonic oscillator, such as a quantized mode of electromagnetic radiation. A harmonic oscillator can be associated with a pair of bosonic mode operators $(\hat{a}, \hat{a}^\dagger)$ called annihilation and creation operators, as well as the quadrature field operators $(\hat{x}, \hat{p})$ representing the position and momentum canonical observables. The above operators of CV systems are similar to the Pauli operators $(\hat{\sigma_x}, \hat{\sigma_y}, \hat{\sigma_z})$ in the qubit model. The mode operators and quadrature operators are related by

$$\begin{aligned} \hat{x} &= \sqrt{\frac{\hbar}{2}}(\hat{a} + \hat{a}^\dagger) \\ \hat{p} &= -i\sqrt{\frac{\hbar}{2}}(\hat{a} - \hat{a}^\dagger). \end{aligned} \tag{2.1}$$

A N-qumode CV system corresponding to N quantum harmonic oscillators is associated with the Hilbert space $\mathcal{H}^{\otimes N} = \otimes_{i=1}^{N}\mathcal{H}_i$ and N pairs of mode operators $\{\hat{a}_i, \hat{a}_i^\dagger\}_{i=1}^N$ and quadrature field operators $\{\hat{x}_i, \hat{p}_i\}_{i=1}^N$. In particular, those operators should obey the commutation relations:

$$\begin{aligned} [\hat{a}_i, \hat{a}_i^\dagger] &= \delta_{ij} \\ [\hat{x}_i, \hat{p}_j] &= i\hbar\delta_{ij}. \end{aligned} \tag{2.2}$$

The expression of a quantum state in a DV system could be a linear superposition of two orthogonal unit basis states $|0\rangle$ and $|1\rangle$ giving: $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ where $\alpha, \beta \in \mathbb{C}$ and satisfying $|\alpha|^2 + |\beta|^2 = 1$. In addition, a pure qubit state can be visualized as living on the so-called Bloch sphere if we write $|\varphi\rangle = \cos(\theta/2)|0\rangle + \sin(\theta/2)e^{i\phi}|1\rangle$ where θ and ϕ to determine a point on the surface of Bloch sphere. If a point is on the Bloch sphere's surface, it corresponds to a pure state; otherwise, it's a mixed state if the point is inside the sphere.

The states in CV systems can express in both Fock space and phase space due to the wave-particle duality. The Fock space has countable and discrete basis states $\{|n\rangle\}_{n=0}^{\infty}$ which are referred to commonly as photon number basis and are the eigenstates of photon number operator $\hat{n} = \hat{a}^\dagger\hat{a}$ with $\hat{n}|n\rangle = n|n\rangle$. Though discrete, the Fock basis is infinite dimensional. It is therefore often convenient to express CV states in the phase space of $\hat{x}$ and $\hat{p}$. In place of coordinates represented by conjugate variables in the classical phase space, the coordinates of phase space in quantum mechanics are the position operator $\hat{x}$ and momentum operator $\hat{p}$ in a Hilbert space. Both have continuous eigenspectra, giving

$$\hat{x}|x\rangle = x|x\rangle, \hat{p}|p\rangle = p|p\rangle \tag{2.3}$$

with continuous eigenvalues $x \in \mathbb{R}, p \in \mathbb{R}$ and continuous, infinite basis with eigenstates $\{|x\rangle\}_{x\in\mathbb{R}}, \{|p\rangle\}_{p\in\mathbb{R}}$. Those two bases can be transformed by Fourier transformation. Unlike the classical case, one quantum state cannot be expressed as a point in the phase space due to the uncertainty principle. In general, the quantum state can be uniquely determined by its density matrix or a so-called quasiprobability function over the phase space. The Wigner function is perhaps the most well-known of the family of possible quasiprobability functions, which a density matrix can transform with all information inside the quantum state.

2.2 Gaussian states

Assume there is a quantum system with all information encoded in N qumodes. We can arrange a column vector of N pairs of quadrature field operators

$$\hat{\mathbf{r}} = (\hat{x}_1, \hat{p}_1, \hat{x}_2, \hat{p}_2, ..., \hat{x}_N, \hat{p}_N)^T. \tag{2.4}$$

The $2N$ position and momentum operators contained in $\hat{r}$ satisfy the commutation relations

$$[\hat{r}_i, \hat{r}_j] = i\hbar\Omega_{ij}, (i, j = 1, 2, ..., 2N) \tag{2.5}$$

where Ω_{ij} is an element of the $2N \times 2N$ symplectic form

$$\Omega = \bigoplus_{k=1}^{N} \omega, \quad \omega = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}. \tag{2.6}$$

Based on this N-qumode symplectic form, we can define the Weyl operator which is a multi-mode displacement operator

$$D(\xi) = \exp\bigl(i\hat{\mathbf{r}}^T\Omega\xi\bigr) \tag{2.7}$$

where $\xi \in \mathbb{R}^{2N}$. Moreover, the characteristic function of a N-qumodes state with density operator $\hat{\rho}$ has form [4]

$$\chi(\xi) = Tr[\hat{\rho}D(\xi)]. \tag{2.8}$$

We can now obtain the Wigner function used to describe $\hat{\rho}$ by Fourier transform acted on the characteristic function

$$W(\mathbf{r}) = \int_{\mathbb{R}^{2N}} \frac{d^{2N}\xi}{(2\pi)^{2N}} \exp\bigl(-i\mathbf{r}^T\Omega\xi\bigr)\chi(\xi) \tag{2.9}$$

where $\mathbf{r}$ are $2N$-dimensional real continuous variables used for the total system and are eigenvalues of quadratures operators in $\hat{\mathbf{r}}$. The above Wigner function is a quasi-probability distribution with properties that can be normalized to one as normal probability distribution but may take negative values. The Wigner function $W(\mathbf{r})$ transforms the N-qumode quantum state $\hat{\rho}$ from its Hilbert space $\mathcal{H}^{\otimes N}$ to a $2N$ real symplectic space $\kappa = (\mathbb{R}^{2N}, \Omega)$ named as phase space. A Gaussian state is defined as a state whose Wigner function and characteristic function are Gaussian functions in the phase space κ.

It is sufficient for Gaussian states to be completely characterized by their first and second moments of quadrature operators, which are mean vector $\bar{\mathbf{r}}$ and covariance matrix $\mathbf{V}$, respectively [22]

$$\begin{aligned} \bar{\mathbf{r}} &= \langle \hat{\mathbf{r}} \rangle = Tr(\hat{\mathbf{r}}\hat{\rho}) \\ \mathbf{V}_{ij} &= \frac{1}{2} \langle \Delta\hat{r}_i \Delta\hat{r}_j + \Delta\hat{r}_j \Delta\hat{r}_i \rangle \end{aligned} \tag{2.10}$$

with $\Delta\hat{r}_i = \hat{r}_i - \langle \hat{r}_i \rangle$ which refers to either position or momentum quadrature operator's variance of mode i. The diagonal elements of covariance matrix could be expressed as

$$\mathbf{V}_{ii} = \begin{cases} \left\langle (\Delta\hat{x}_{(i+1)/2})^2 \right\rangle & i = 1, 3, ..., 2N-1 \\ \left\langle (\Delta\hat{p}_{i/2})^2 \right\rangle & i = 2, 4, ..., 2N \end{cases} . \tag{2.11}$$

In addition, the Wigner function of any Gaussian states can be expressed by $\mathbf{r}$ as

$$W(\mathbf{r}) = \frac{\exp\left(-\frac{1}{2}(\mathbf{r} - \bar{\mathbf{r}})\mathbf{V}^{-1}(\mathbf{r} - \bar{\mathbf{r}})\right)}{(2\pi)^N \sqrt{\det \mathbf{V}}}. \tag{2.12}$$

It has been shown that the covariance matrix and mean vector of Gaussian states contain a complete description of the states. It is therefore useful to study their properties, as they are a much more compact representation of the state than a Wigner function or density matrix. The valid and physical covariance matrix $\mathbf{V}$ describing Gaussian state $\hat{\rho}$ should be a $2N \times 2N$ real symmetric and positive semi-definitive matrix and follow the uncertainty principle [57]

$$\mathbf{V} + i\Omega \geq 0 \tag{2.13}$$

which implies that the eigenvalues of Hermitian matrix $\mathbf{V} + i\Omega$ should be non-negative [2, 9].

A real symplectic matrix $\mathbf{S}$ should satisfy the condition

$$\mathbf{S}\Omega\mathbf{S}^T = \Omega \tag{2.14}$$

where Ω has been defined in 2.6. The symplectic matrix for a N-mode system is a invertible, $2N \times 2N$ matrix with $\det(\mathbf{S}) = 1$. Based on the Williamson decomposition, for the covariance matrix $\mathbf{V}$ of a N-mode Gaussian state, there exists a symplectic matrix $\mathbf{S}$ which can diagonalize $\mathbf{V}$ such that

$$\mathbf{V} = \mathbf{S} \bigoplus_{k=1}^{N} \begin{pmatrix} v_k & 0 \\ 0 & v_k \end{pmatrix} \mathbf{S}^T \tag{2.15}$$

where v_k are the symplectic eigenvalues of $\mathbf{V}$ [64]. Furthermore, it is more convenient to calculate the symplectic eigenvalues v_k of $\mathbf{V}$ by taking the absolute value of the standard eigenvalues of $i\Omega\mathbf{V}$ [56]. The uncertainty principle can also be expressed as $v_k \geq 1$ for all k.

The covariance matrix of the two-mode Gaussian states has a standard form with diagonalizable cross-correlation sub-blocks [10, 58]

$$\mathbf{V} = \begin{pmatrix} a & 0 & c & 0 \\ 0 & a & 0 & d \\ c & 0 & b & 0 \\ 0 & d & 0 & b \end{pmatrix} \tag{2.16}$$

where a and b respectively represent the local correlation of two modes; as well as, c and d represent the inter-mode correlations between first mode and second mode shown as

$$\begin{aligned} a &= (\left\langle (\Delta\hat{x}_1)^2 \right\rangle + \left\langle (\Delta\hat{p}_1)^2 \right\rangle)/2 \\ b &= (\left\langle (\Delta\hat{x}_2)^2 \right\rangle + \left\langle (\Delta\hat{p}_2)^2 \right\rangle)/2 \\ c &= |d| = (\langle \Delta\hat{x}_1\Delta\hat{x}_2 \rangle - \langle \Delta\hat{p}_1\Delta\hat{p}_2 \rangle)/2. \end{aligned} \tag{2.17}$$

If $c = d = 0$, the corresponding covariance matrix would be a product state.

We know that a quantum state can also be expressed as an evolution which starts from a vacuum state $|0\rangle$ and evolves according to a Hamiltonian, H, for an evolution time t as

$$|\Psi(t)\rangle = \exp(-itH)\,|0\rangle\,. \tag{2.18}$$

For a bosonic system, H can be expressed as a polynomial function of quadrature operators so that $H = H(\hat{x}, \hat{p})$. Gaussian states are states generated by Hamiltonians that are at most quadratic in the operators $\hat{x}, \hat{p}$. These Hamiltonians correspond to linear systems, in that they have linear equations of motion.

2.2.1 Pure Gaussian states

A single qumode pure Gaussian state could be completely characterized by two complex continuous variables: displacement parameter α and squeezing parameter z [36]. In fact, Gaussian state is named because it fits the Gaussian distribution that the displacement parameter and squeezing parameter are corresponding to the mean and variance of Gaussian distribution, respectively. We would review several common single-mode pure Gaussian states and summarize their displacement and squeezing parameters, moreover, expressions in Gaussian formulation.

Vacuum state

The most elementary Gaussian state is the vacuum state, and the Wigner function of the vacuum state is shown in Fig.2.1a,

$$|0\rangle = \frac{1}{\sqrt[4]{\pi\hbar}} \int dx e^{-x^2/(2\hbar)} |x\rangle \tag{2.19}$$

which represents not only the eigenstate of annihilation operator with eigenvalue equal to zero, giving $\hat{a}|0\rangle = 0$ but also a Fock state with $n = 0$, as well as the mean vector $\bar{\mathbf{r}} = (0,0)$ and covariance matrix $\mathbf{V} = \frac{\hbar}{2}\mathbf{I}$. The vacuum state is often used as the initial state for computation.

Coherent state

A coherent state is the vacuum state displaced by a displacement operator $D(\alpha)$ to a point α in phase space [62]

$$|\alpha\rangle = D(\alpha)|0\rangle \tag{2.20}$$

with $D(\alpha) = \exp\big(\alpha\hat{a}^\dagger - \alpha^*\hat{a}\big)$, $\alpha \in \mathbb{C}$. Since $\hat{a}|\alpha\rangle = \alpha|\alpha\rangle$, the coherent state is eigenstate of annihilation operator $\hat{a}$ with eigenvalue α. The displacement operator $D(\alpha)$ is a unitary operator with $D(\alpha)D^\dagger(\alpha) = D^\dagger(\alpha)D(\alpha) = \mathbb{I}$ due to $D(-\alpha) = D^\dagger(\alpha) = D^{-1}(\alpha)$. We can rewrite $D(\alpha) = \exp\Big(-\frac{|\alpha|^2}{2}\Big)\exp\big(\alpha\hat{a}^\dagger\big)\exp(-\alpha^*\hat{a})$ so that coherent state can be expressed in Fock basis as

$$|\alpha\rangle = \exp\left(-\frac{|\alpha|^2}{2}\right)\sum_{n=0}^{\infty}\frac{\alpha^n}{\sqrt{n!}}|n\rangle . \tag{2.21}$$

Meanwhile, the corresponding mean vector is $\bar{\mathbf{r}} = 2\sqrt{\frac{\hbar}{2}}(Re(\alpha), Im(\alpha))$ and covariance matrix $\mathbf{V} = \frac{\hbar}{2}\mathbf{I}$. In the propagating laser fields, the optical coherent states play an important role since an ideal laser beam's quantum state consists of N copies of identical pure coherent states mixed together [60].

Squeezed state

The squeezed state is a squeezed vacuum state without displacement

$$|z\rangle = S(z)|0\rangle \tag{2.22}$$

with $S(z) = \exp\big(\frac{1}{2}(z^*\hat{a}^2 - z\hat{a}^{\dagger 2})\big)$, $z = r\exp(i\phi)$ and $r \geq 0$. Similarly, the squeezed operator is also unitary as displacement operator. Because any Gaussian states can be expressed by Fock states, squeezed states have even number Fock states expansion, giving

$$|z\rangle = \frac{1}{\sqrt{\cosh r}} \sum_{n=0}^{\infty} \frac{\sqrt{(2n)!}}{2^n n!} (-\exp(i\phi)\tanh r)^n \, |2n\rangle \, . \tag{2.23}$$

In Gaussian formulations, the mean vector and covariance matrix of a squeezed state are $\bar{\mathbf{r}} = (0,0)$ and $\mathbf{V} = \frac{\hbar}{2} R(\phi/2) \begin{bmatrix} e^{-2r} & 0 \\ 0 & e^{2r} \end{bmatrix} R(\phi/2)^T$, respectively. The $R(\phi)$ inside covariance matrix is a rotation operator with the unitary form $R(\phi) = \exp(i\phi\hat{n})$ where $\hat{n} = \hat{a}^\dagger \hat{a}$ is the photon number operator.

Displaced squeezed state

Combining the displacement operator and squeezed operator together on a vacuum state, we get the definition of displaced squeezed state (also called squeezed coherent state)

$$|\alpha, z\rangle = D(\alpha) S(z) \, |0\rangle \, . \tag{2.24}$$

It is important to note that those two operators do not commute so that $D(\alpha)S(z) \neq S(z)D(\alpha)$. Furthermore, $\bar{\mathbf{r}} = 2\sqrt{\frac{\hbar}{2}}(Re(\alpha), Im(\alpha))$ and $\mathbf{V} = \frac{\hbar}{2} R(\phi/2) \begin{bmatrix} e^{-2r} & 0 \\ 0 & e^{2r} \end{bmatrix} R(\phi/2)^T$ for the displaced squeezed state. The displaced squeezed state is the most general single-qumode pure Gaussian state.

2.2.2 Mixed Gaussian state

Thermal radiation is electromagnetic radiation emitted by any matter at finite temperatures and is an inevitable noise resource. The thermal states of matter causing thermal radiation describe the states of a system in thermal equilibrium at non-zero temperatures T.

The thermal state is an important example of mixed Gaussian state, with density matrix

$$\rho_{th}(\bar{n}) = \sum_{n=0}^{\infty} \frac{\bar{n}^n}{(1+\bar{n})^{n+1}} \, |n\rangle \langle n| \tag{2.25}$$

with the mean photon number $\bar{n} = Tr(\rho(\bar{n})\hat{n})$. In addition, the mean vector is $\bar{\mathbf{r}} = (0,0)$, and covariance matrix is $\mathbf{V} = (2\bar{n}+1)\frac{\hbar}{2}\mathbf{I}$.

2.3 Universality and CV gates

The quantum computer over DV is universal if it can implement any required unitary transformation over finite qubits using quantum logic gates to any degree of accuracy. For a CV quantum computer, the definition of universality is the ability to approach any unitary transformation, which is related to Hamiltonian is a polynomial function of mode operators, to arbitrary precision in finite steps [32]. However, Ref. [32] showed that universality requires non-Gaussian resources, either gates or resource states.

Gaussian gates transform Gaussian states to Gaussian states; such a process is a linear transformation. Several Gaussian gates are related to the Gaussian states from the last section. The common Gaussian gates act on one-mode or two-mode, and are at most quadratic in mode operators such as displacement $D(\alpha)$, rotation $R(\phi)$, squeezing $S(z)$ and beamsplitter $BS(\theta, \phi)$ gates. The basic non-Gaussian gates are single-mode gates which are degree 3 or higher in the mode or quadrature operators such as cubic phase gate $V(\gamma)$ and Kerr gate $K(\kappa)$. Additionally, any CV gate for the multi-mode system can decompose single-mode gates and two-mode gates due to universality.

Similar to the unitary transformation U operating on $\hat{\rho}$ in the Hilbert space $\mathcal{H}$, the symplectic transformation $\mathbf{S}$ acts on the mean vector and covariance matrix in the phase space κ [11]

$$\hat{\rho}^{'} = U\hat{\rho}U^{\dagger} \quad \Leftrightarrow \quad \begin{aligned} \bar{\mathbf{r}}^{'} &= \mathbf{S}\bar{\mathbf{r}} \\ \mathbf{V}^{'} &= \mathbf{S}\mathbf{V}\mathbf{S}^{T} \end{aligned} \tag{2.26}$$

where $\mathbf{S}$ is a symplectic matrix defined in 2.14. Indeed, each unitary operator on Gaussian states can be linked with an associated symplectic matrix.

Displacement gate

The displacement gate has the form

$$D(\alpha) = e^{\alpha\hat{a}^{\dagger} - \alpha^{*}\hat{a}} = e^{r(e^{i\phi}\hat{a}^{\dagger} - e^{-i\phi}\hat{a})} \tag{2.27}$$

with $\alpha = re^{i\phi}$ where $r \geq 0$ and $\phi \in [0, 2\pi)$. It could convert $\hat{a}$ and $\hat{a}^{\dagger}$ by Campbell-Baker Hasdorff formula

$$\begin{aligned} D^{\dagger}(\alpha)\hat{a}D(\alpha) &= \hat{a} + \alpha I, \\ D^{\dagger}(\alpha)\hat{a}^{\dagger}D(\alpha) &= \hat{a}^{\dagger} + \alpha^{*}I. \end{aligned} \tag{2.28}$$

For Heisenberg picture, the displacement gate affects the position and momentum operators,

$$\begin{aligned} D^{\dagger}(\alpha)\hat{x}D(\alpha) &= \hat{x} + \sqrt{2\hbar}Re(\alpha), \\ D^{\dagger}(\alpha)\hat{p}D(\alpha) &= \hat{p} + \sqrt{2\hbar}Im(\alpha) \end{aligned} \tag{2.29}$$

which explains that the $D(\alpha)$ make displacement of state with α in phase space, see Fig.2.1b.

Squeezing gate

Squeezing gate for single-mode is defined as

$$S(z) = e^{\frac{1}{2}(z^*\hat{a}^2 - z\hat{a}^{\dagger 2})} \tag{2.30}$$

where $z = re^{i\phi}$ with $r \geq 0$ and $\phi \in [0, 2\pi)$. Similarly, it could transform $\hat{a}$ and $\hat{a}^\dagger$ as

$$\begin{aligned} S^\dagger(z)\hat{a}S(z) &= \cosh r\hat{a} - e^{i\phi}\sinh r\hat{a}^\dagger, \\ S^\dagger(z)\hat{a}^\dagger S(z) &= \cosh r\hat{a}^\dagger - e^{-i\phi}\sinh r\hat{a}. \end{aligned} \tag{2.31}$$

The squeezing gate acts on quadrature operators,

$$\begin{aligned} S^\dagger(z)\hat{x}S(z) &= e^{-r}\hat{x}, \\ S^\dagger(z)\hat{p}S(z) &= e^{r}\hat{p}. \end{aligned} \tag{2.32}$$

Additionally, the associated symplectic matrix of squeezing is

$$S_{squeezing}(r, \phi) = \begin{pmatrix} \cosh r - \cos\phi\sinh r & -\sin\phi\sinh r \\ -\sin\phi\sinh r & \cosh r + \cos\phi\sinh r \end{pmatrix}. \tag{2.33}$$

In phase space, the squeezing operator acts on a vacuum state (see Fig.2.1c) so that one quadrature squeezed by a factor e^{-r} has a lower uncertainty; as a consequence, the orthogonal quadrature with a higher uncertainty would be stretched by a factor e^r to obey the uncertainty principle. In the limitation that the squeezing parameter is sufficiently large with $z \approx r$, we can approach the zero momentum eigenstates $|0\rangle_p$ and zero position eigenstates $|0\rangle_x$ via squeezing on vacuum state

$$\begin{aligned} |0\rangle_p &\approx S(-r)\,|0\rangle\,, \\ |0\rangle_x &\approx S(r)\,|0\rangle\,. \end{aligned} \tag{2.34}$$

Phase rotation gate

The phase rotation operation is defined as

$$R(\phi) = e^{i\phi\hat{n}} \tag{2.35}$$

where $\hat{n} = \hat{a}^\dagger\hat{a}$ is photon number operator. We could obtain transformation of mode operators according to

$$\begin{aligned} R^\dagger(\phi)\hat{a}R(\phi) &= e^{i\phi}\hat{a}, \\ R^\dagger(\phi)\hat{a}^\dagger R(\phi) &= e^{-i\phi}\hat{a}^\dagger. \end{aligned} \tag{2.36}$$

The phase rotation gate rotates a state represented by position and momentum quadratures in phase space by an angle of ϕ, see Fig.2.1d,

$$\begin{aligned} R^\dagger(\phi)\hat{x}R(\phi) &= \cos\phi\hat{x} - \sin\phi\hat{p}, \\ R^\dagger(\phi)\hat{p}R(\phi) &= \sin\phi\hat{x} + \cos\phi\hat{p}. \end{aligned} \tag{2.37}$$

The symplectic transformation corresponding to a unitary phase rotation is

$$S_R(\phi) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix}. \tag{2.38}$$

Beamsplitter Gate

The beamsplitter gate is a two-mode gate with form

$$BS(\theta,\phi) = e^{\theta(e^{i\phi}\hat{a}_i\hat{a}_j^\dagger - e^{-i\phi}\hat{a}_i^\dagger\hat{a}_j)}. \tag{2.39}$$

We could mix two modes $\hat{a}_i$ and $\hat{a}_j$ by applying beamsplitter operator on mode operators

$$\begin{aligned} BS^\dagger(\theta,\phi)\hat{a}_i BS(\theta,\phi) &= \cos\theta\hat{a}_i - e^{-i\phi}\sin\theta\hat{a}_j = T\hat{a}_i - R^*\hat{a}_j, \\ BS^\dagger(\theta,\phi)\hat{a}_j BS(\theta,\phi) &= \cos\theta\hat{a}_j + e^{i\phi}\sin\theta\hat{a}_i = T\hat{a}_j + R\hat{a}_i \end{aligned} \tag{2.40}$$

where $T = \cos\theta$ is transmittivity amplitude, and $R = e^{i\phi}\sin\theta$ is reflectivity amplitude of the beamsplitter with $|T|^2 + |R|^2 = 1$. The special case of beamsplitter gate is 50-50 beamsplitter which has $\theta = \pi/4$ such that $|T|^2 = |R|^2 = \frac{1}{2}$, see Fig.2.1e.

If we apply a beamsplitter operation on random two single-mode Gaussian states whose covariance matrices are $A = a\mathbf{I}$ and $B = b\mathbf{I}$, the covariance matrix of output state is obtained by $\mathbf{V}_{BS} = S_{BS}(A \oplus B)S_{BS}^\dagger$ [48] as

$$\mathbf{V}_{BS} = \begin{pmatrix} (a-b)\cos^2(\theta) + b & 0 & (a-b)\sin\theta\cos\theta e^{i\phi} & 0 \\ 0 & (a-b)\cos^2(\theta) + b & 0 & (a-b)\sin\theta\cos\theta e^{-i\phi} \\ (a-b)\sin\theta\cos\theta e^{-i\phi} & 0 & (b-a)\cos^2(\theta) + a & 0 \\ 0 & (a-b)\sin\theta\cos\theta e^{i\phi} & 0 & (b-a)\cos^2(\theta) + a \end{pmatrix} \tag{2.41}$$

where S_{BS} is the symplectic transformation matrix of beamsplitter operator. If the initial two states are both thermal states with thermal photon number N_1, N_2 for each state, their covariance matrices can be replaced by $A = (N_1 + \frac{1}{2})\mathbf{I}$ and $B = (N_2 + \frac{1}{2})\mathbf{I}$, respectively.

Two-mode squeezing gate

The two-mode squeezing operator has definition as

$$S_2(z) = e^{(z\hat{a}_i^\dagger \hat{a}_j^\dagger - z^* \hat{a}_i \hat{a}_j)} \tag{2.42}$$

where $z = re^{i\phi}$ with squeezing parameter $r \geq 0$ and squeezing phase angle $\phi \in [0, 2\pi)$. Two-mode squeezing operator could transform the mode operators as

$$\begin{aligned} S_2(z)^\dagger \hat{a}_i S_2(z) &= \hat{a}_i \cosh(r) - \hat{a}_j^\dagger e^{i\phi} \sinh(r), \\ S_2(z)^\dagger \hat{a}_j S_2(z) &= \hat{a}_j \cosh(r) - \hat{a}_i^\dagger e^{i\phi} \sinh(r). \end{aligned} \tag{2.43}$$

In addition, the two-mode squeezing gate could be decomposed into two single-mode squeezing gate with opposite squeezing parameters between two 50-50 beamsplitters

$$S_2(z) = BS^\dagger(\pi/4, 0)[S(z) \otimes S(-z)]BS(\pi/4, 0). \tag{2.44}$$

The two-mode squeezed (TMS) vacuum state (see Fig.2.1f) can be obtained by a two-mode squeezing operator acting on the vacuum state, a representative state for a two-mode continuous-variable entangled state. The TMS vacuum state has zero mean and covariance matrix

$$\mathbf{V}_{TMSvacuum} = \begin{pmatrix} \cosh 2r & 0 & \sinh 2r & 0 \\ 0 & \cosh 2r & 0 & -\sinh 2r \\ \sinh 2r & 0 & \cosh 2r & 0 \\ 0 & -\sinh 2r & 0 & \cosh 2r \end{pmatrix}. \tag{2.45}$$

More generally, the correlations a, b, c, d in the covariance matrix of TMS thermal state would be calculated by [15]

$$\begin{aligned} a &= \cosh 2r + 2N_1 \cosh^2 r + 2N_2 \sinh^2 r \\ b &= \cosh 2r + 2N_2 \cosh^2 r + 2N_1 \sinh^2 r \\ c &= -d = (1 + N_1 + N_2) \sinh 2r \end{aligned} \tag{2.46}$$

where N_1, N_2 are the mean thermal photon number of each mode.

Cubic phase gate

The cubic phase gate is a single mode gate which is a non-Gaussian resource for CV computation with definition

$$V(\gamma) = e^{i\frac{\gamma}{3\hbar}\hat{x}^3}. \tag{2.47}$$

The cubic phase state $|\gamma\rangle$ is generated by applying the cubic phase gate on the zero momentum eigenstate $|0\rangle_p$ as $|\gamma\rangle = V(\gamma)\,|0\rangle_p$.

Kerr gate and Cross-Kerr gate

The Kerr gate which acts on a single mode is given by

$$K(\kappa) = e^{i\kappa\hat{n}^2} \tag{2.48}$$

since the relative Hamiltonian of Kerr interaction is $H = (\hat{a}^\dagger\hat{a})^2 = \hat{n}^2$.

The cross-Kerr gate also provides non-Gaussian transformation as Kerr gate, but it implements on two-mode as

$$CK(\kappa) = e^{i\kappa\hat{n}_1\hat{n}_2} \tag{2.49}$$

where the Hamiltonian of cross-Kerr interaction is $H = \hat{n}_1\hat{n}_2$.

2.4 CV measurements

In place of Pauli-basis measurements $|0/1\rangle\langle 0/1|, |\pm\rangle\langle\pm|, |\pm i\rangle\langle\pm i|$ on qubits, we consider the Gaussian measurements of homodyne and heterodyne detection, as well as non-Gaussian measurement of photon counting.

Homodyne measurement

Homodyne detection is a measurement that is projected onto the eigenstates of a single field quadrature operator $\hat{x}$ or $\hat{p}$. More generally, homodyne measurement could be expanded to projection onto eigenstates of the Hermitian operator $\hat{x}_\phi$ with operator $|x_\phi\rangle\langle x_\phi|$.

$$\hat{x}_\phi = \cos\phi\hat{x} + \sin\phi\hat{p} \tag{2.50}$$

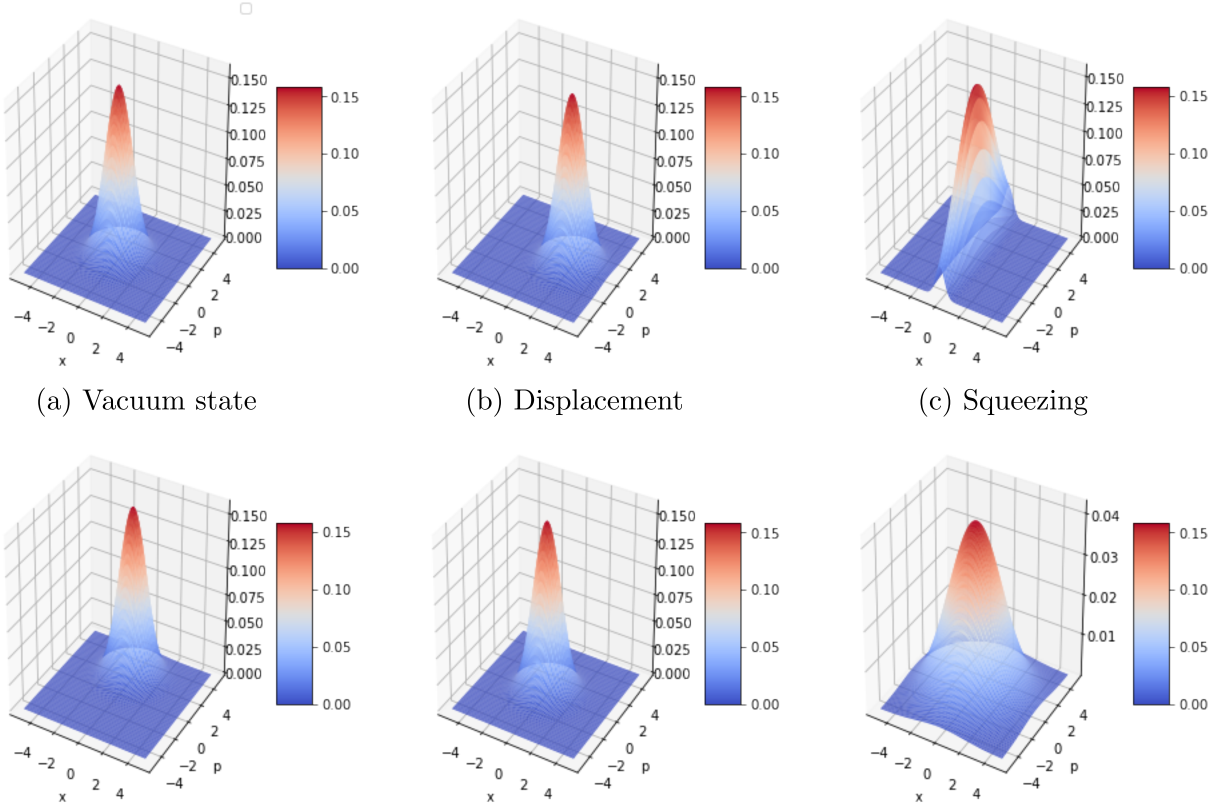
This is the same as rotating the state by $-\phi$ and taking a homodyne measurement on $\hat{x}$, and would return a real measurement value. It is important to note that the conditional state of the remaining modes preserves its Gaussian Character if we take a homodyne measurement on one of the modes in a multi-mode Gaussian state.

Heterodyne measurement

Unlike homodyne measurements which only measure a single quadrature operator, heterodyne detection simultaneously measures both quadratures $\hat{x}$ and $\hat{p}$. Because these two conjugate operators do not commute, Heisenberg's uncertainty principle limits the simultaneous measurement's precision. That is, heterodyne detection must add noise to the measurement. A heterodyne measurement projects onto the basis of coherent states with operator $\frac{1}{\sqrt{\pi}}|\alpha\rangle\langle\alpha|$. The output value is complex and can be expressed by the Husimi Q function defined $Q(\alpha) = \frac{1}{\pi} \langle\alpha| \rho |\alpha\rangle$. Again, the conditional state of the remaining modes is still a Gaussian state after heterodyne measurement.

Photon counting

Photon counting measurement projects the states onto the Fock basis $|n\rangle$ with corresponding measurement operator $|n\rangle \langle n|$ so that the measurement output values are non-negative integer. Unlike homodyne and heterodyne measurements, a photon counting measurement projects the system into a non-Gaussian basis, except if the the output is $n = 0$. (The vacuum state is unique in being both a Gaussian state and a Fock state.)



(a) Vacuum state

(b) Displacement

(c) Squeezing

(d) Phase rotation

(e) Beamsplitter

(f) Two-mode squeezing

Figure 2.1: Operation of common Gaussian gates. Wigner functions of (a) vacuum state, (b) coherent state obtained by acting displacement gate on vacuum state, (c) squeezed vacuum state, (d) phase rotation of coherent state, (e) beamsplitter of two-mode vacuum state, and (f) TMS vacuum state, which are drawn using Strawberry Fields. [26]

Chapter 3

Gaussian Quantum Discord

Quantum entanglement was the first kind of quantum correlation studied. It has a wide range of applications in quantum information processing with various research. However, people have realized there are other types of quantum correlation, with quantum discord being one broad category. Quantum discord measures the quantum correlation of bipartite quantum systems and is an essential resource for quantum information processing, especially in mixed-state quantum systems. Researchers have extended quantum discord to the CV systems and analytically evaluated the quantum discord for two-mode Gaussian states, which is thus named Gaussian quantum discord. This section introduces how to measure quantum discord. It starts with two equivalent expressions of mutual information in the classical information theory and then expands to quantum analogues of mutual information. Furthermore, we demonstrate two approaches to obtaining analytical solutions for different kinds of Gaussian states.

3.1 Quantum discord definition

For bipartite quantum states, there exist classical correlation and quantum correlation [41]. The difference between classical and quantum correlation originates from that the measurement will disturb the quantum systems but won't disturb the classical systems, and the disturbance can be used to calculate the quantum correlation [34]. Furthermore, the classical correlation can be qualified as the information of one subsystem gained by measurements on the other subsystem. For pure quantum states, the quantum correlation accords with the quantum entanglement. Quantum discord can be used to measure the quantum correlation of mixed states, which is defined by the difference between two classically equivalent expressions for mutual information, which need not be the same for quantum systems [21].

From classical information theory, the Shannon entropy $H(A)$ describes the average uncertainty in a discrete random variable A with outcomes $a_1...a_n$ that occur with probability $P(a_1)...P(a_n)$. The entropy of A is defined as

$$H(A) = -\sum_{i=1}^{n} P(a_i) \log P(a_i). \tag{3.1}$$

The mutual information measures the total correlation between two random variables A and B, which reduces the uncertainty of A due to knowledge of B. Due to Bayes' and its property, the two classical expressions of mutual information I and J are equal

$$\begin{aligned} I(A:B) &= H(A) + H(B) - H(A,B) \\ J(A:B) &= H(A) - H(A|B) \end{aligned} \tag{3.2}$$

where $H(A|B)$ is conditional entropy and $H(A,B)$ is joint entropy.

Now the above concepts in classical systems are extended to quantum systems. To obtain the quantum mutual information of a bipartite quantum state, we replace corresponding classical probability distributions by density matrices $\rho_A, \rho_B, \rho_{AB}$ as well as generalizing Shannon entropy to $S(\rho) = -Tr(\rho \log \rho)$ von Neumann entropy

$$I(\rho_{AB}) = S(\rho_A) + S(\rho_B) - S(\rho_{AB}). \tag{3.3}$$

The quantum mutual information contains the total correlations of the system, i.e., the quantum mutual information is the sum of the classical correlation and the quantum correlation. However, the quantum analog $J(\rho_{AB})$ couldn't be as easily substituted as for $I(\rho_{AB})$ since it involves a generalization of the conditional entropy $H(A|B)$. The $J(\rho_{AB})$ can be obtained with the help of a set of measurements $\{B_i\}$,$\sum_i B_i = \mathbf{I}$ made on B where i corresponds to the different measurement results. After the measurement, the state of A when result B_i is detected becomes

$$\rho_{A|B_i} = \frac{\mathrm{Tr}_B(\rho_{AB} B_i)}{\mathrm{Tr}_{AB}(\rho_{AB} B_i)}. \tag{3.4}$$

The probability of this result is $p_i = \mathrm{Tr}_{AB}(\rho_{AB} B_i)$ so that the generalization of the conditional entropy of state A can be defined as

$$S(\rho_A|\{B_i\}) = \sum_i p_i S(\rho_{A|B_i}). \tag{3.5}$$

Accordingly, the quantum generalization $J(\rho_{AB})$ is expressed as

$$J(\rho_{AB})_{\{B_i\}} = S(\rho_A) - S(\rho_A|\{B_i\}). \tag{3.6}$$

The magnitude of $J(\rho_{AB})$ is related to the choice of the measurement base, and after maximizing overall measurement results, which means find the infimum value of $S(\rho_A|\{B_i\})$ term, we obtain the classical correlation as

$$J(\rho_{AB})_{\{B_i\}} = S(\rho_A) - inf_{\{B_i\}}S(\rho_A|\{B_i\}). \tag{3.7}$$

The quantum discord, which measures the quantum correlation, is equal to the difference between the total correlation and classical correlation [21]

$$D(\rho_{AB})_{\{B_i\}} = I(\rho_{AB}) - J(\rho_{AB})_{\{B_i\}} = S(\rho_B) - S(\rho_{AB}) + inf_{\{B_i\}}S(\rho_A|\{B_i\}). \tag{3.8}$$

Since we only consider quantum discord with measurement made on B, we can abbreviate it as

$$D(\rho_{AB}) \equiv D(\rho_{AB})_{\{B_i\}}. \tag{3.9}$$

Quantum discord has the following properties:

(1) Generally, quantum discord is asymmetric since the conditional entropy is asymmetric such that $D(\rho_{AB})_{\{A_i\}} \neq D(\rho_{AB})_{\{B_i\}}$.

(2) For pure states, quantum discord is equal to quantum entanglement; they are generally not equal for mixed states.

(3) Quantum discord is non-negative $D(\rho_{AB}) \geq 0$ since the concavity of conditional entropy. The quantum discord is equal to zero if and only if for product state with $\rho_{AB} = \sum_i B_i \rho_{AB} B_i$.

(4) $D(\rho_{AB})_{\{B_i\}} \leq S(\rho_B)$, however $D(\rho_{AB})_{\{B_i\}} \leq S(\rho_A)$ does not necessarily hold.

(5) Quantum discord is invariant under local unitary transformation such as $D(\rho_{AB}) = D((U^A \otimes U^B)\rho_{AB}(U^A \otimes U^B)^\dagger)$.

(6) For separable states except for product states, the quantum discord must be smaller than 1. On the contrary, if the quantum discord is smaller than 1, the corresponding Gaussian state has the probability of being an entangled state.

3.2 Analytical solution of Gaussian quantum discord

The analytical solution of quantum discord is complicated to obtain since the final part in the definition of quantum discord need to be optimized. In [30, 33], the analytical solution of quantum discord for a kind of two-qubit states was demonstrated. In [3, 35], a more general analytical solution for quantum discord of X-states that only diagonal and anti-diagonal elements in the covariance matrices are nonzero was further obtained. We would focus on a family of CV states , i.e., the Gaussian state, as an example, to introduce how to get Gaussian quantum discord for bipartite Gaussian states. Recall 2.16 the diagonal sub-blocks expression of covariance matrix for any two-mode Gaussian state, and re-write it as

$$\sigma_{AB} = \begin{pmatrix} \alpha & \gamma \\ \gamma^T & \beta \end{pmatrix} \tag{3.10}$$

where $\alpha = a\mathbf{I}$, $\beta = b\mathbf{I}$, $\gamma = diag\{c, d\}$. The correlations a, b, c and d are determined by the four symplectic invariants $A = det\alpha = a^2$, $B = det\beta = b^2$, $C = det\gamma = cd$, $D = det\sigma_{AB} = (ab - c^2)(ab - d^2)$. Additionally, the symplectic eigenvalues $v_\pm$ of a covariance matrix can be expressed as

$$2v_\pm^2 = (A + B + 2C) \pm \sqrt{(A + B + 2C)^2 - 4D} \tag{3.11}$$

where $v_- \geq 1$ to ensure the corresponding Gaussian state are meaningful due to the uncertainty principle.

In 2010, Adesso derived the closed solution of quantum discord for general two-mode Gaussian states after measurements made on mode B by projecting on pure Gaussian states [1]. The measurement state B is given by

$$\sigma_0 = \begin{pmatrix} \lambda & 0 \\ 0 & 1/\lambda \end{pmatrix} \tag{3.12}$$

where $\lambda = 1$ or $\lambda \to 0$ implies that the executed measurement is heterodyne or homodyne, respectively. The reduced state A whose covariance matrix is independent of the measurement result is described by [12, 18, 20]

$$\varepsilon = \alpha - \gamma(\beta + \sigma_0)^{-1}\gamma^T \tag{3.13}$$

Now the Gaussian quantum discord with covariance matrix σ_{AB} is expressed as

$$D(\sigma_{AB}) = f(\sqrt{B}) - f(v_-) - f(v_+) + inf_{\sigma_0} f(\sqrt{det\varepsilon}) \tag{3.14}$$

with $f(x) = (\frac{x+1}{2}) \ln\big(\frac{x+1}{2}\big) - (\frac{x-1}{2}) \ln\big(\frac{x-1}{2}\big)$.

We can let $inf_{\sigma_0} det\varepsilon = E^{min}$. If $(D - AB)^2 \leq (1 + B)C^2(A + D)$, the conditional entropy is minimized by heterodyne measurements which project on the coherent state basis so that

$$E^{min} = \frac{2C^2 + (-1 + B)(-A + D) + 2|C|\sqrt{C^2 + (-1 + B)(-A + D)}}{(-1 + B)^2}. \tag{3.15}$$

Otherwise, the homodyne measurements which project on the squeezed state basis minimize the conditional entropy such that

$$E^{min} = \frac{AB - C^2 + D - \sqrt{C^4 + (-AB + D)^2 - 2C^2(AB + D)}}{2B}. \tag{3.16}$$

We additionally verify a set of formulas for Gaussian quantum discord, which are easier to derive and return the same result as Adesso obtained. Firstly, we should find the covariance matrix ε of reduced state A and the corresponding symplectic eigenvalues for three different states. Then, we substitute the above results into Gaussian quantum discord expression to get some simple analytical results.

Recall the expression of ε and substitute the diagonal elements of γ with $d = \pm c$ so that

$$\varepsilon = \begin{pmatrix} a - \frac{c^2}{b+\lambda} & 0 \\ 0 & a - \frac{c^2}{b+1/\lambda} \end{pmatrix}. \tag{3.17}$$

For heterodyne measurements which $\lambda = 1$, the $\varepsilon = p_\varepsilon \mathbf{I}$ with $p_\varepsilon = a - \frac{c^2}{b+1}$. The covariance matrix elements of a two-mode squeezed (TMS) vacuum state can be defined by $a = b = 2N_S + 1, c = \sqrt{a^2 - 1} = \sqrt{4N_S(N_S + 1)}$ where N_S is the squeezing number of photons so that $p_\varepsilon = 1$ and $\varepsilon = \mathbf{I}$ which means the state A reduced to vacuum state. For symmetric Beamsplitter (BS) state, the covariance matrix has elements $a = b = 2N_S + 1, c = \sqrt{4N_S^2} = 2N_S$ and now $p_\varepsilon = \frac{3N_S+1}{N_S+1} \approx 1 + 2N_S$ for $N_S \ll 1$. Furthermore, the covariance matrix of a asymmetric BS state takes the form $a = 2N_S + 1, b = 2N_I + 1, c = \sqrt{4N_S N_I}$; therefore, $p_\varepsilon = 2N_S + 1 - \frac{4N_S N_I}{2(N_I+1)} \approx 1$ for $N_I \to \infty$.

For homodyne measurements where $\lambda \to 0$, the ε takes the form

$$\varepsilon = \begin{pmatrix} a - \frac{c^2}{b} & 0 \\ 0 & a \end{pmatrix}. \tag{3.18}$$

We can re-define $p_\varepsilon = a - \frac{c^2}{b}$, and the TMS state has $p_\varepsilon = \frac{1}{2N_S+1} \approx 1 - 2N_S$ if $2N_S \ll 1$. For the symmetric BS state, $p_\varepsilon = 2 - \frac{1}{2N_S+1} \approx 1 + 2N_S$ if $2N_S \ll 1$. In addition, the asymmetric BS state has $p_\varepsilon = 2N_S + 1 - \frac{4N_SN_I}{2N_I+1} \approx 1$ for $N_I \to \infty$.

To find the symplectic eigenvalues $v_\pm$, we start with symmetric states such as TMS state and symmetric BS state. Recall expression of $v_\pm$ in which $C = \pm c^2$ and $D = (a^2 - c^2)^2$ now, we can define $\triangle = A + B + 2C = 2a^2 + 2C$ and $\beta^2 = \triangle^2 - 4D = 8a^2(c^2 + C)$ so that

$$\begin{aligned} v_\pm &= \frac{1}{2}(\triangle \pm \beta) \\ &= a^2 + C \pm \frac{a}{2}\sqrt{8(C + c^2)}. \end{aligned} \tag{3.19}$$

For the general TMS vacuum states, we can substitute $C = -c^2$ into above expression so that $v_\pm = \sqrt{a^2 - c^2}$, furthermore $v_\pm = 1$ since $a = 2N_S + 1, c = \sqrt{4N_S(N_S + 1)}$. The BS states have $C = c^2$ such that $v_\pm = a \pm c$, and $v_+ = 4N_S + 1, v_- = 1$ for a vacuum symmetric BS state.

Now, let's consider the Gaussian quantum discord of TMS vacuum states if heterodyne measurements made on mode B, thus

$$\begin{aligned} D(\sigma_{AB}) &= f(2N_S + 1) \\ &= (N_S + 1)ln(N_S + 1) - N_S ln N_S \end{aligned} \tag{3.20}$$

since $\sqrt{B} = 2N_S + 1, v_\pm = 1, \sqrt{det\varepsilon} = 1$. Furthermore, $D(\sigma_{AB}) \approx -N_S ln N_S$ for $N_S \ll 1$. Similarly, if homodyne measurements made on mode B, discord has same result due to $\sqrt{det\varepsilon} = \sqrt{a^2 - c^2} = 1$.

The Gaussian quantum discord of symmetric BS states in which $\sqrt{det\varepsilon} = \frac{3N_S+1}{N_S+1}$ based on heterodyne measurements has

$$D(\sigma_{AB}) = f(2N_S + 1) - f(4N_S + 1) + f(\frac{3N_S + 1}{N_S + 1}) \tag{3.21}$$

moreover $p_\varepsilon \approx 2N_S + 1$ for $N_S \ll 1$ so that $D(\sigma_{AB}) \approx 2f(2N_S + 1) - f(4N_S + 1) \approx 2N_S ln2$.

Indeed, it's more complicated for the asymmetric BS states to find Gaussian quantum discord. To find the symplectic eigenvalues $v_\pm$, we can let

$$\begin{aligned} \overline{p} &= \frac{a + b}{2} \\ , \delta p &= \frac{a - b}{2}, \\ \chi &= \sqrt{\delta p^2 + c^2} \end{aligned} \tag{3.22}$$

and use the results obtained in symmetric BS case so that $\triangle = 2(\overline{p}^2 + \chi^2)$, $\beta = 4\overline{p}\chi$ and $v_\pm = \frac{1}{2}(\triangle \pm \beta) = \overline{p} \pm \sqrt{\delta p^2 + c^2}$. For a vacuum case, the symplectic eigenvalues are calculated by $v_\pm = (N_S + N_I + 1) \pm \sqrt{((N_S - N_I)^2 + 4N_S N_I} = 2\overline{N} + 1 \pm 2\overline{N}$ where $\overline{N} = \frac{N_S+N_I}{2}$, hence $v_+ = 4\overline{N} + 1, v_- = 1$. Rewrite $p_\varepsilon = 2N_S + 1 - \frac{4N_S N_I}{2N_I+1} = 1 + \frac{2N_S}{N_I+1}$ and substitute it into discord expression, the Gaussian quantum discord of asymmetric BS states is

$$D(\sigma_{AB}) = f(2N_I + 1) - f(4\overline{N} + 1) + f(1 + \frac{2N_S}{N_I + 1}). \tag{3.23}$$

Particularly, there is infinite idler with weak signal measured that $N_I \to \infty, N_S \ll 1$, as a result $D_{IS} \approx f(2N_S + 1) - f(2N_I + 1) + f(2N_I + 1) \approx f(2N_S + 1)$ which looks closed to TMS state discord. In comparison, if measurements made on idler, $D_{SI} \approx f(2N_I + 1) - f(2N_I + 1) + f(1 + \frac{2N_S}{N_I+1}) \approx f(1 + \frac{2N_S}{N_I+1})$.

The values of quantum Gaussian discord for TMS vacuum states, symmetric BS vacuum states and asymmetric BS vacuum states are shown in Fig.3.1.

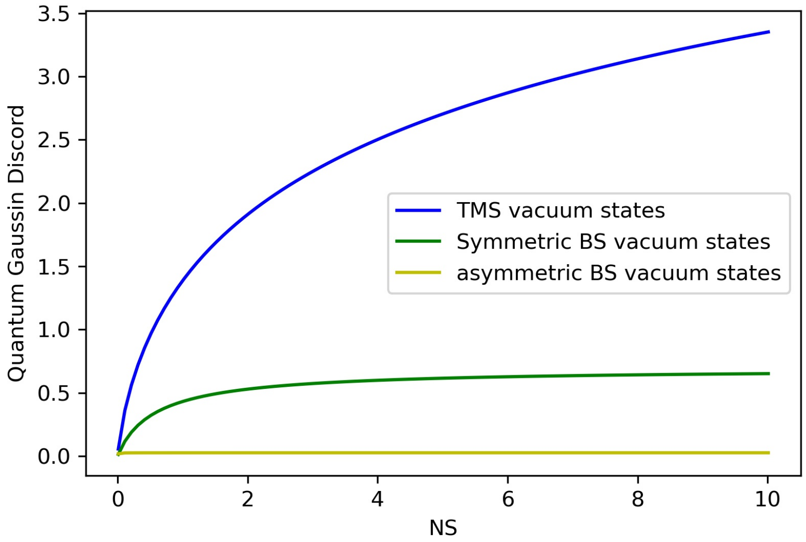


Figure 3.1: Analytical solutions of quantum Gaussian discord for TMS vacuum states, symmetric BS vacuum states and asymmetric BS vacuum states. Note that the values of quantum Gaussian discord for the asymmetric BS vacuum states are small but not zero.

Chapter 4

Classical neural network with error Back-Propagating

This chapter presents the first approach for recognizing Gaussian discord by performing a back-propagating (BP) algorithm on a double hidden-layer neural network. Machine learning is such a process that how a machine approximately builds a parametric model via learning from known data set to predict the output of unknown data. The BP algorithm is gradient descent and back-forward updates neural network parameters from the output layer to the input layer.

4.1 Machine Learning

Machine learning (ML) is the necessary product of developing artificial intelligence research to a certain stage. A data set is a collection of relevant records, and each record is correspondent to an object defined as a sample in ML. Matters that reflect the performance of nature in a certain aspect are called features, and the space into which the feature is tangled becomes the feature space. Generally, assuming that $\mathcal{D} = \{x_1, x_2, ..., x_m\}$ is a data set containing m samples, and each sample is described by d features so that $x_i = (x_{i1}, x_{i2}, ..., x_{id})$ is a feature vector of the d-dimensional feature space $\mathcal{X}$.

Learning a model from data is called training, and this process is accomplished by executing some learning algorithm. The data used in the training process is called the training data set, and each sample in it is called a training sample. The learned model responds to some underlying pattern of the data in question and is also called a hypothesis.

To build the relevant predictive model, we need to obtain the results about the training samples named label. For a given data set $\mathcal{D} = \{(x_i, y_i)|x_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}$ $(i = 1, 2, ..., m)$, (x_i, y_i) represents i^{th} samples where y_i is the label of x_i belonging to label space $\mathcal{Y}$.

ML requires the construction of a map from the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$ as $f(x) : \mathcal{X} \rightarrow \mathcal{Y}$, where the unknown function $f(x)$ is target function. Since the target function is unknown, we could only find the closest one to the target function. The approximate target function is a parameterized function $f(x; \theta)$ which could be the sum of some nonlinear functions with optimal performance on training data set, and the parameters θ could be learned by the optimization process. The process of finding optimal parameters to maximize model performance is called deep learning in the neural network. After learning the model, using the model to make predictions is called testing, and the sample being tested is called the test sample. In this paper, we focus on a binary classification task whose predicted values are discrete.

4.2 Neural Networks

The most fundamental element in Neural Network (NN) is neuron (also named as perceptron). The neuron receives input values x_i which multiplying by connection weight w_i and then comparing with threshold b, so the output of neuron y is obtained by passing this sum to an activation function $f(x)$, giving as

$$y = f(\sum w_i x_i - b). \tag{4.1}$$

The above mentioned situation can be abstracted into a simple model, which is the McCulloch–Pitts neuron model, see Fig.4.1 that has been used until now [38].

The activation function should be a non-linear function for which can introduce nonlinear properties. For example, the step function is the most straightforward activation function

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0. \end{cases} \tag{4.2}$$

Another standard activation function is the sigmoid function which output is bounded in the range $(0, 1)$.

$$f(x) = \frac{1}{1 + \exp\{-x\}} \tag{4.3}$$

A neural network is formed by interconnecting a set of neurons in a specific structure [27]. The part of a neural network that receives input data is called the input layer, and

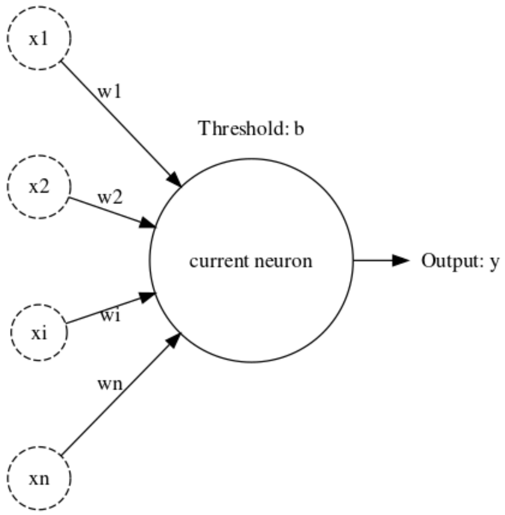


Figure 4.1: McCulloch–Pitts neuron model

the part that is used to output the result is the output layer. The layer of neurons between the input layer and output layer is called the hidden layer. Several famous neural networks exist, such as multi-layer feedforward neural network with derivative convolution-neural network and recurrent neural network with long short-time memory.

4.3 Back-Propagating algorithm

A well-known algorithm in the neural network is Back-Propagating (BP) which is one of the most widely used machine learning algorithms for many practical problems [52]. Here we show a double-layer feedforward neural network example, see Fig.4.2, for explaining the error Back-Propagating (BP) algorithm, which is used for updating weights and thresholds of a given neural network. The BP algorithm is the application and extension of gradient descent in neural network models [63, 52]. For a data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, $x_i \in \mathbb{R}^d, y_i \in \mathbb{R}^l$, each input sample is described by d features and output is l-dimensional real value vector; meanwhile, the first hidden layer has q neurons and the second hidden layer has p neurons.

Assuming the threshold of j^{th} neuron in output layer is represented by θ_j, and thresholds for h^{th} neuron in first hidden layer and k^{th} neuron in second hidden layer are γ_h and μ_k,

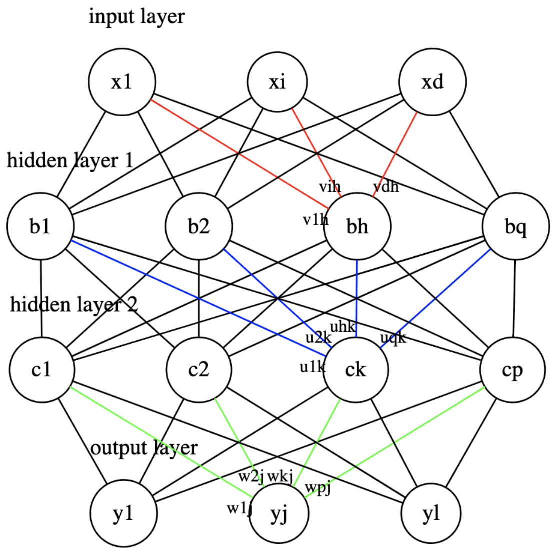


Figure 4.2: Neural network with two hidden layers

respectively. The connection weight between i^{th} neuron in the input layer and h^{th} neuron in the first hidden layer is v_{ih} so that the input and output of h^{th} neuron in the first hidden layer are $\alpha_h = \sum_{i=1}^{d} v_{ih}x_i$ and $b_h = f_1(\alpha_h - \gamma_h)$. Similarly, assuming u_{hk} is the connection weight between h^{th} neuron in first hidden layer and k^{th} neuron in second hidden layer, the input and output for k_{th} neuron are $\lambda_k = \sum_{h=1}^{q} u_{hk}b_h$ and $c_k = f_2(\lambda_k - \mu_k)$. For j^{th} neuron in output layer with connected weight w_{kj}, the input and output are $\beta_j = \sum_{k=1}^{p} w_{kj}c_k$ and $y_j = f_3(\beta_j - \theta_j)$. The activation functions f_1, f_2, f_3 inside above expressions might be different.

Matrices could express those. For the feature space of data set $\mathcal{D}$, assuming the threshold as a dummy node with input -1 gives

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} & -1 \\ x_{21} & x_{22} & \dots & x_{2d} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{md} & -1 \end{pmatrix}. \tag{4.4}$$

The transposed weight matrix for input layer to hidden layer 1 has the form

$$V^T = \begin{pmatrix} v_{11} & v_{12} & \dots & v_{1(q-1)} & v_{1q} \\ v_{21} & v_{22} & \dots & v_{2(q-1)} & v_{2q} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ v_{d1} & v_{d2} & \dots & v_{d(q-1)} & v_{dq} \\ \gamma_1 & \gamma_2 & \dots & \gamma_{(q-1)} & \gamma_q \end{pmatrix}. \tag{4.5}$$

Similarly, if the left two transposed weight matrices are U^T and W^T, the final output could be

$$\hat{y} = f_3(f_2(f_1(XV^T)U^T)W^T). \tag{4.6}$$

For a training sample $(x_e, y_e) \in \mathcal{D}$, and the output of neural network is $\hat{y}_e = (\hat{y}_1^e, \hat{y}_2^e, ..., \hat{y}_l^e)$ with $\hat{y}_j^e = f(\beta_j - \theta_j), j = 1, 2, ..., l$. Meanwhile, the mean-square error on (x_e, y_e) is

$$E_e = E(y_e, \hat{y}_e) = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^e - y_j^e)^2. \tag{4.7}$$

The goal of BP algorithm is to minimize the accumulated mean-square error E on the data set $\mathcal{D}$ where

$$E = \frac{1}{m}\sum_{e=1}^{m} E_e. \tag{4.8}$$

There are $(d + p + 1)q + (l + 1)p + l$ parameters need to be confirmed: $d \times q$ weights from the input layer to the first hidden layer, $q \times p$ weights from the first hidden layer to the second hidden layer, $p \times l$ weights from the secon hidden layer to the output layer; and $q + p + l$ thresholds for each layer except input layer. BP is an iterative learning algorithm that uses a generalized perceptron learning rule to update the parameter estimates in each round of the iteration, such as for a random parameter v

$$v \leftarrow v + \triangle v. \tag{4.9}$$

We use connected weight w_{kj} from the second hidden layer to the output layer for derivation. The BP algorithm is gradient descent which modifies parameters in the negative step direction of the target. For a given studying rate $\eta \in (0, 1)$,

$$\triangle\, w_{kj} = -\eta \frac{\partial E_e}{\partial w_{kj}} \tag{4.10}$$

the weights w_{kj} are first effect on the input value β_j of j^{th} neuron in output layer, then on this neuron's output value $\hat{y}_j^e$, and finally to the E_e so that

$$\frac{\partial E_e}{\partial w_{kj}} = \frac{\partial E_e}{\partial \hat{y}_j^e}\frac{\partial \hat{y}_j^e}{\partial \beta_j}\frac{\partial \beta_j}{\partial w_{kj}}. \tag{4.11}$$

We can define the gradient term in the output layer as

$$\begin{aligned} g_j &= -\frac{\partial E_e}{\partial \hat{y}_j^e}\frac{\partial \hat{y}_j^e}{\partial \beta_j} \\ &= -(\hat{y}_j^e - y_j^e)f^{'}(\beta_j - \theta_j) \\ &= \hat{y}_j^e(1-\hat{y}_j^e)(y_j^e - \hat{y}_j^e) \end{aligned} \tag{4.12}$$

if the activation function is sigmoid function with property $f^{'}(x) = f(x)(1-f(x))$. According to the definition of β_j, it is obvious that

$$\frac{\partial \beta_j}{\partial w_{kj}} = c_k. \tag{4.13}$$

Now, the updated formulas for the weights from second hidden layer to output layer and thresholds of output layer are obtained as

$$\begin{aligned} \triangle w_{kj} &= \eta g_j c_k, \\ \triangle \theta_j &= -\eta g_j. \end{aligned} \tag{4.14}$$

Similarly, for second hidden layer to the first hidden layer, there are

$$\begin{aligned} \triangle u_{hk} &= \eta r_k b_h, \\ \triangle \mu_k &= -\eta r_k \end{aligned} \tag{4.15}$$

where $r_k = \sum_{j=1}^{l} w_{kj} g_j f^{'}(\lambda_k - \mu_k)$. For the first hidden layer to input layer, we have

$$\begin{aligned} \triangle v_{ih} &= \eta e_h x_i, \\ \triangle \gamma_h &= -\eta e_h \end{aligned} \tag{4.16}$$

where $e_h = \sum_{k=1}^{p} u_{hk} r_k f^{'}(\alpha_h - \gamma_h)$.

Obviously, to calculate the gradient terms r_k of the second hidden layer, the gradient terms g_j at the output layer must be known, i.e., the gradient terms are derived from the last layer to the first layer, which is why the algorithm is called the BP algorithm. When the number of layers in the neural network is enormous, the gradient terms are almost zero after passing backwards several layers. This problem is called gradient vanishing, with one possible solution being to learn multi-Level Hierarchies [28].

The learning rate η controls the update step in each algorithm iteration; if it is too large, it tends to oscillate, and if it is too small, it converges too slowly. For each layer, the learning rate η could be different for more precise regulation.

4.4 Recognizing the Gaussian quantum discord by BP algorithm with neural network

For recognizing the Gaussian discord by the BP algorithms, the first step is to construct a generalized Gaussian state data set. The analytical solutions of quantum discord for different kinds of Gaussian states could be seen as labels. The input point of each Gaussian state sample is the covariance matrix.

There are two general types of parameters in machine learning: those that need to be learned and estimated from data, called model parameters, and tuning parameters in machine learning algorithms, which need to be set artificially, called hyper-parameters. The parameters in the neural networks usually are weights and thresholds. For the neural network using BP algorithm, the choice of hyper-parameters: learning rate η, the number of hidden layers and the number of neurons in each hidden layer determine the model's performance. The setting of these hyper-parameters needs to base on experience and many attempts.

We set the number of hidden layers as two, the first hidden layer has 256 neurons and the second hidden layer has 128 neurons. Since each covariance matrix of bipartite Gaussian state is described by 16 elements, the number of neurons of the input layer is 16. The expected output of the neural network is one real value that represents the predicted Gaussian quantum discord, thus the output layer only consists of one neuron.

We construct a Gaussian data set to evaluate the performance of the double hidden layers neural network. The number of samples of the Gaussian data set is 800. The Gaussian data set is constructed by randomly choosing correlations a, b, c, d in the covariance matrix standard form shown in 2.16; then we calculate the Gaussian quantum discord using its analytical solutions 3.15 or 3.16 as labels. The result of inputting the Gaussian data set into the neural network is shown in Fig.4.3.

The result performs poorly for various reasons, such as the hyper-parameters and activation functions are not appropriate, and we deal with the Gaussian states as the classical information. To overcome the limitations of recognizing Gaussian states using classical machine learning, we will introduce quantum machine learning in Chapter 5, which demonstrates the ability to handle Gaussian quantum discord classification task.

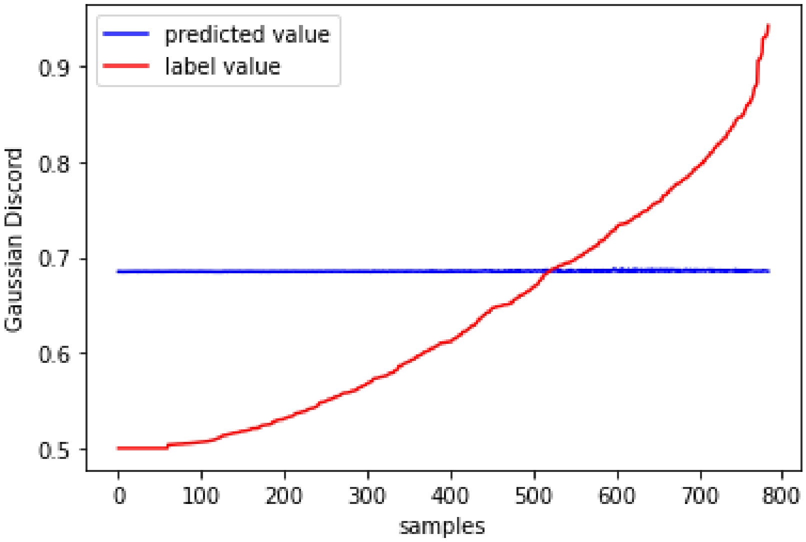


Figure 4.3: Recognizing Gaussian quantum discord using double hidden layers neural network with BP algorithms. The predicted results marked as blue versus the label values marked as red are obtained based on the Gaussian data sets that the input Gaussian data set is constructed by randomly choosing a, b, c, d

Chapter 5

QKS Algorithm for Recognizing Gaussian Discord

Classical machine learning seems ineffective in recognizing Gaussian states since it may not appropriately acquire and deal with the complete quantum information inside quantum states. The states would need to be measured to produce classical information to feed into the algorithm. In comparison, quantum machine learning could directly take quantum states as input. In this chapter, we will simulate a QML algorithm acting directly on Gaussian quantum states using the PennyLane package. PennyLane is an open-source framework in Python 3, could help us implement quantum subroutines of a given quantum-classical hybrid algorithm on classical simulators or quantum hardware. The problem of classifying Gaussian quantum discord into two groups could be approached by a supervised classifiier using hybrid optimization performed on a variational quantum circuit. In particular, we have applied the Quantum Kitchen sinks (QKS) [65] algorithm to recognize Gaussian quantum discord, which has a classical counterpart known as random kitchen sinks (RKS) [51, 49, 50].

5.1 Quantum Machine learning

Quantum machine learning expands machine learning and is expected to solve both classical and quantum machine learning problems on quantum devices. Quantum operations and quantum systems are used in quantum machine learning to demonstrate the quantum advantages in solving classical problems; moreover, quantum machine learning combines quantum computing and classical machine learning algorithms. The computer hardware used to run classical algorithms has always determined the limitations of what computers can learn, such as the parallel GPU clusters that enable the implementation of deep learning using neural networks [55]. We could wish to see if quantum computers can reduce their training time. Recently, as the area of quantum machine learning has progressed, several proposals for quantum machine learning algorithms that can be implemented on CV systems have been developed, which could help recognize some Gaussian states properties. Instead of ideal and universal quantum computers, we prefer developing quantum machine learning algorithms on the near-term quantum computers.

With the advent of the noisy intermediate-scale quantum (NISQ) device, the potential of conducting practical quantum information processing computations has emerged. While NISQ technologies do not have fault tolerance or a very large number of qubits, both of which are needed for code-breaking applications, they do open up new possibilities for accessing quantum information. The power of near-term quantum computers may be researched using NISQ devices which need completely different approaches than error-corrected quantum computers to perform computations. Hybrid algorithms, which combine classical and quantum processing, have been proposed as a way to achieve quantum enhancement by utilizing the available NISQ computers.

In quantum machine learning, the quantum circuits are trainable and differentiable. That is the generalization of deep learning in machine learning with the application in quantum optimization. There are two approaches to train quantum circuits: classical simulators and quantum hardware. The first approach is more suitable for the small quantum circuit, which simulates on classical software with the help of optimization and machine learning tools. On the other hand, the hardware-based method cannot extract quantum information and only could return measurement values.

5.2 PennyLane

PennyLane is an open-source software framework built on Python 3 to explore quantum machine learning on large-scale quantum computing [6]. PennyLane is developed by a Canadian quantum technology company called Xanadu. The main proposal of PennyLane is the optimization and automatic differentiation of quantum-classical hybrid computations due to their capacity to calculate the gradients of a random variational quantum circuit. The automatic differentiation provides an algorithm of calculating $\nabla f(x, y)$ for some function $f(x, y)$ which may be a mathematical function or arbitrary numerical code.

Pennylane supports both DV (qubit-model) and CV (continuous-variable) quantum systems and constructs generalized architecture for near-term quantum computers. Furthermore, the quantum computer could be regarded as a classical neural network and trained on PennyLane. The framework is compatible with any gate-based quantum simulator or hardware via the diverse plugin systems. As a result, the limitations of what computers can learn were pushing since the relevant optimization and running process could execute quantum hardware like QPU instead of GPU. For another example, it is accessible to Strawberry Fields [26] via the PennyLane-SF plugin to simulate photonic quantum circuits. Both Strawberry Fields and PennyLane are developed by Xanadu which is a quantum technology company in Canada.

5.3 Training variational quantum circuit

Variational quantum circuits are the core components of quantum-classical hybrid algorithms for NISQ devices. They are parameterized quantum circuits and can be optimized by classical optimization algorithms [37, 40]. A standard variational quantum circuit has three components: initial state, a quantum circuit parameterized by variational parameters, and measurement of output. Those circuits can be decomposed into Gaussian and non-Gaussian gates to ensure the universality and nonlinear character of CV models. To entirely operate a variational circuit, firstly, we prepare an initial state, often the vacuum state $|0\rangle$. The next step is applying a unitary gate sequence whose parameters are a function of some inputs x and free variable θ. The outputs of a variational circuit are the measurements of some fixed observables in the final state. The parameters of unitary gates are classical quantities.

5.3.1 Optimization of variational quantum circuits based on Gradient descent

The goal of optimization of variational quantum circuits is similar to classical optimization, both to estimate some unknown functions that can optimize the model's performance. The parameters of approximate target function are usually optimized based on some machine learning algorithms such as the classical BP algorithm, which minimizes mean square error and then updates gradient terms. The ability to compute gradients of variational quantum circuits is analogous to the BP algorithm, which two are both iterative methods for updating parameters [40]. Gradient terms can be obtained via automatic differentiation on PennyLane. Therefore, this could be used to compute the cost function's gradient terms with respect to all variables by applying gradient descent algorithms to minimize the cost function. As a result, the variational quantum circuits would exhibit optimal performance with optimal parameters.

The training process is a hybrid scheme that integrates a classical loop to update the parameters and quantum processing to extract the gradients. Assume the target function is $f(x;\Theta)$, the cost function can be defined as

$$C(\Theta) = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i;\Theta))^2 \tag{5.1}$$

where m is the total number of samples in the data set, and x_i is the i^{th} sample with label y_i. Each step t is a gradient descent process, and each variable $\theta_i \in \Theta$ is updated as

$$\theta_i^{t+1} = \theta_i^t - \eta^t \partial_{\theta_i} C(\Theta) \tag{5.2}$$

where η^t are the learning rate that may depend on gradient terms, and the gradient term is partial derivative with respect to individual variable θ_i.

In place of the standard BP algorithm in the classical machine learning, since it cannot handle quantum information inside the quantum node, computing the gradient of quantum nodes with respect to inputs or all variables is necessary by using a quantum device that not only executes the quantum node but also computes its gradient. There are two approaches for calculating derivatives of a quantum node: analytical and numerical derivatives supported by all classical simulators and quantum hardware. The analytical derivative for the target function $f(x;\theta) = f(\mu)$ with respect to μ (μ can be the input x or the variable θ) of a quantum node is

$$\partial_\mu f(\mu) = c(f(\mu + s) - f(\mu - s)) \tag{5.3}$$

where c, s are fixed and finite parameters. The analytical derivative would return the exact derivatives, compared with the numerical derivatives which approximately calculate derivatives, giving

$$\partial_\mu f(\mu) \approx \frac{f(\mu + \triangle\mu) - f(\mu - \triangle\mu)}{2\triangle\mu} \tag{5.4}$$

where $\triangle\mu$ is a small change.

The parameters of the variational quantum circuit would be optimized by a given optimizer such as momentum optimizer and Nesterov momentum optimizer, both of which are based on gradient descent [59, 42]. The momentum optimizer accumulates a velocity vector along the direction of ongoing decrease cost as

$$\begin{aligned} \theta^{(t+1)} &= \theta^{(t)} + v^{(t+1)} \\ v^{(t+1)} &= mv^{(t)} - \eta^t \nabla f(\theta^{(t)}) \end{aligned} \tag{5.5}$$

where m is the momentum coefficient. The Nesterov momentum optimizer is an extension to the momentum optimizer with the present position shifted by the momentum when calculating the gradient of the target function

$$v^{(t+1)} = mv^{(t)} - \eta^t \nabla f(\theta^{(t)} + mv^{(t)}). \tag{5.6}$$

5.3.2 Pre-processing

Fig.5.1 shows the scheme of pre-processing the Gaussian states data set before directly feed inputs into models.

Indeed, the task that classifies two classes of quantum Gaussian discord is a supervised hybrid optimization task executed on a variational quantum classifier. Before directly feeding the Gaussian data set into optimizer, it is essential to separate the Gaussian states into two classes: if the corresponding Gaussian discord is bigger than one so that label with 1 indicated that those Gaussian states are entangled; otherwise, the Gaussian discord is less than one so that the label would be -1 shown that such Gaussian states are either separable or entangled.

The classification problem shares the basic assumption that the number of training samples for different categories is equal so that make sure we are not over-fitting the sample data set. If the number of training samples differs slightly from category to category, it usually has little effect. However, if the difference is significant, it can cause disturbance to the optimization process. In the initial Gaussian data set, the number of -1 classes is much larger than the number of 1 classes; then the optimizer will only return a learner that will always predict new samples as negative classes; however, such a learning result is worthless.

This situation in which the number of training samples varies greatly across categories in a classification problem is called a class imbalance. There are two main methods to deal with class imbalance: over-sampling and under-sampling. For the initial construed Gaussian data set, over-sampling the positive class samples in the data set, i.e., adding some positive cases to close the number of positive and negative samples. Otherwise,

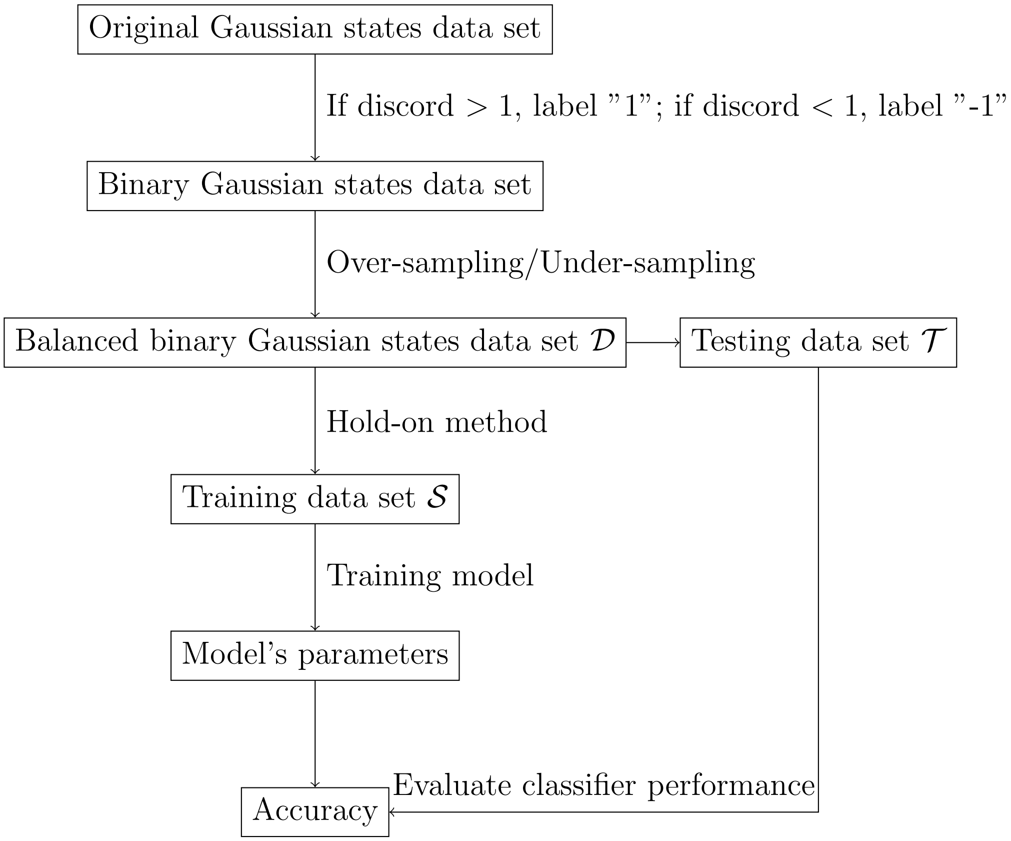

Original Gaussian states data set
If discord > 1, label "1"; if discord < 1, label "-1"
Binary Gaussian states data set
Over-sampling/Under-sampling
Balanced binary Gaussian states data set $\mathcal{D}$
Testing data set $\mathcal{T}$
Hold-on method
Training data set $\mathcal{S}$
Training model
Model's parameters
Evaluate classifier performance
Accuracy

Figure 5.1: Pre-processing scheme

under-sampling is to remove some negative class samples before feeding them into the optimizer. The time overhead of the under-sampling method is much smaller than that of the over-sampling method because the former discards many negative samples, making the training set smaller than the initial training set.

It is important to note that the over-sampling method cannot simply repeat the initial positive samples; otherwise, it will lead to severe over-fitting. The representative algorithm SMOTE adds additional positive samples by interpolating the positive samples in the training set [7]. Meanwhile, to avoid information loss due to under-sampling randomly abandoning negative samples, we can apply EasyEnsemble, which uses the integrated learning mechanism [31]. The negative samples are divided into several sets for different optimizers. There is no important information that seems to be lost globally.

To evaluate the present classification algorithm's performance, it is essential to divide the original data set $\mathcal{D}$ into two sub-datasets: training data set $\mathcal{S}$ and testing data set $\mathcal{T}$. The hold-out method is the simplest method for splitting data set into two mutually exclusive sets, as $\mathcal{D} = \mathcal{S} \cup \mathcal{T}, \mathcal{S} \cap \mathcal{T} = \emptyset$. After training the model on $\mathcal{S}$ and then obtaining optimal model parameters, the performance of the model is evaluated by $\mathcal{T}$, in other words, the unseen data. It is commonly choosing 65% to 80% of $\mathcal{D}$ as training data set $\mathcal{S}$, while the remaining samples would be used for testing.

5.3.3 Optimization of variational quantum circuit for recognizing Gaussian discord

The original Gaussian data set for optimizing the variational quantum circuit consists of vacuum and thermal TMS states, symmetric BS states and asymmetric BS states. For the vacuum case, each class of states are built by choosing 100 evenly spaced N_S over the interval $[0.1, 10.01]$ and $N_I = 1000 \times N_S$; for the thermal case, we additional set $N_{th} = 0.1$. The number of samples of the Gaussian data set is 800. The corresponding Gaussian quantum discord can be calculated by 3.20, 3.21 and 3.23. As the first step shown in Fig.5.1 that separates the 800 Gaussian states of the original Gaussian data set into two classes, we obtain a binary Gaussian states data set that 188 samples with label 1 and 612 samples with label -1. Based on the hold-out method, we chose 75% of the binary Gaussian states data set as the training data set, which contains a total of 600 samples with 140 Gaussian states in class 1 and 460 Gaussian states in class -1.

The Fig.5.2 shows the structure of a variational quantum circuit used for optimization, inspired by [25]. Every input data, which is the covariance matrix of a two-mode Gaussian state $|\psi\rangle$, is prepared as the initial state. The Gaussian gates, including beamsplitter and displacement gates, perform the linear transformation, and cross-Kerr and cubic phase rotation gates provide nonlinear transformation of data and thus construct a nonlinear quantum model. For the binary classification task, we choose that this variational quantum circuit would return the expectation value $\left\langle \hat{x}_{\frac{\pi}{4}} \right\rangle$ of the first qumode as the output value, which is obtained by the homodyne measurement on the generalized quadrature $\hat{x}_\phi$ with $\phi = \frac{\pi}{4}$. If the output value is positive, we set it as class 1; otherwise, the output value is negative so that we set it as class -1.

The Fig.5.3 shows the optimization result and the cost result using Nesterov momentum optimizer after pre-processing the data set. The cost is decreased as expected, but the classification accuracy trends stable around 76.67%. Note that the proportion of class -1 in the Gaussian states training data set is $460/600 \approx 0.7667$ that coincides with the above accuracy. In case this result is just a coincidence, different variational quantum circuit

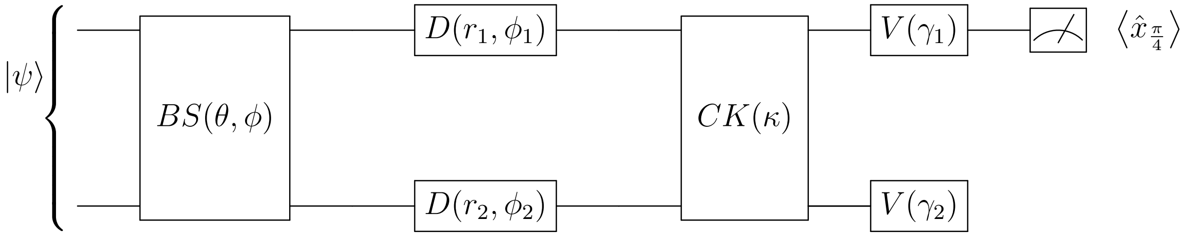


Figure 5.2: Variational quantum circuit for optimization contains beamsplitter and displacement gates to introduce linear transformation, and uses cross-Kerr and cubic phase gates to perform nonlinear transformation.

structures or the same circuit with different outputs were tried but still returned similar results. Those results indicate a failing training example for binary classification since the model could only present negative output to only recognize one class in the data set.

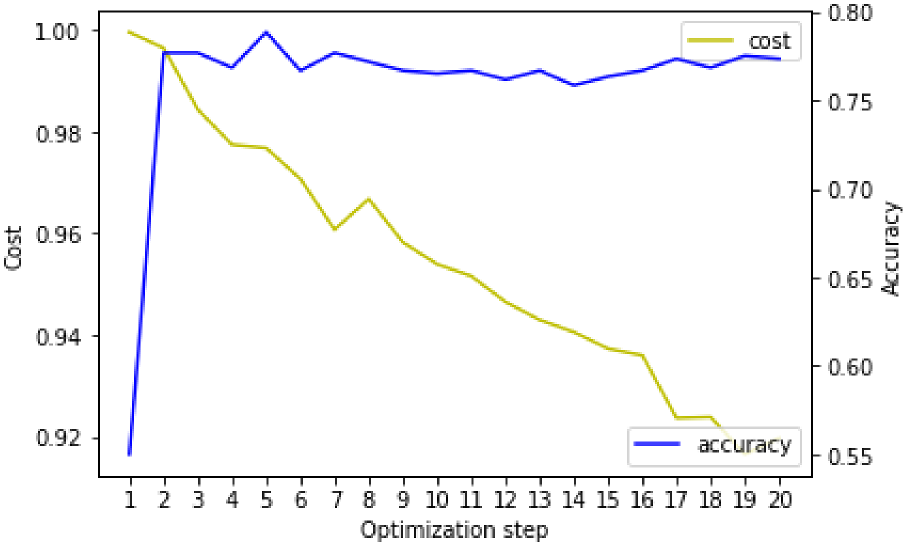


Figure 5.3: Cost and accuracy of optimization variational circuit

5.4 QKS

The QKS algorithm is inspired by a classical machine learning algorithm called Random Kitchen Sinks (RKS), which replaces the optimization process over the kernel by linearly summing many randomized nonlinear functions to estimate the kernel [51, 49, 50]. The QKS is a hybrid algorithm designed for NISQ devices and combines the classical machine learning approaches with quantum nodes used to achieve nonlinear transformation from inputs to higher-dimensional feature vectors [65].

5.4.1 Kernel methods and QKS

For a given data set $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}, y_i \in \{-1, 1\}$, the most basic idea of classification problem in machine learning is to find a divisible decision boundary in the original input space to separate different classes of samples. However, in some practical and complex classification tasks, the original sample space may not find a decision boundary that correctly separates the two samples classes. For such a problem, the samples could be mapped from the input space to a higher dimensional feature space such that the samples are linearly separable within this feature space. Kernel methods, including many classical machine learning algorithms used for pattern recognition tasks, including classification tasks, would perform computations over a kernel function defined in the original input space instead of workin in the transformed feature space. Let the $\phi(x)$ denote the feature vector after mapping input data point x into feature space, the kernel function between samples x_i and x_j is defined as

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle. \tag{5.7}$$

The inner product of two data points in the feature space is equal to the result of their computation in the original input space using the kernel function $k(x_i, x_j)$. Using the kernel function enables working in the higher-dimensional feature space without explicitly transforming all of the data. More generally, the kernel matrix $\mathbf{K}$ for a data set $\mathcal{D} = \{x_1, x_2, , ..., x_m\}$ belong to input space $\mathcal{X}$ is semi-positive defined

$$\mathbf{K} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_m) \\ k(x_2, x_1) & k(x_2, x_2) & \dots & k(x_2, x_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_m, x_1) & k(x_m, x_2) & \dots & k(x_m, x_m) \end{bmatrix}. \tag{5.8}$$

For each kernel matrix, there exists a corresponding feature mapping ϕ.

The performance of the feature space determines the accuracy of the kernel methods in classification tasks. Indeed, we do not know what kernel function is appropriate without knowing the feature mapping, and the kernel function only implicitly defines this feature space. The choice of the kernel function is significant and decides whether the input data can be mapped to a suitable feature space to demonstrate excellent performance. One of the popular classical kernel methods is the support vector machine (SVM), which executes a kernel with intensive computations and effective training.

In contrast to kernel methods, the RKS algorithm works directly in a transformed feature space, but where some of the parameters of the transformation are random. RKS inspires the QKS algorithm, with the transformation now being generated by a random quantum circuit. The implementations of the QKS do not need to calculate the kernel.

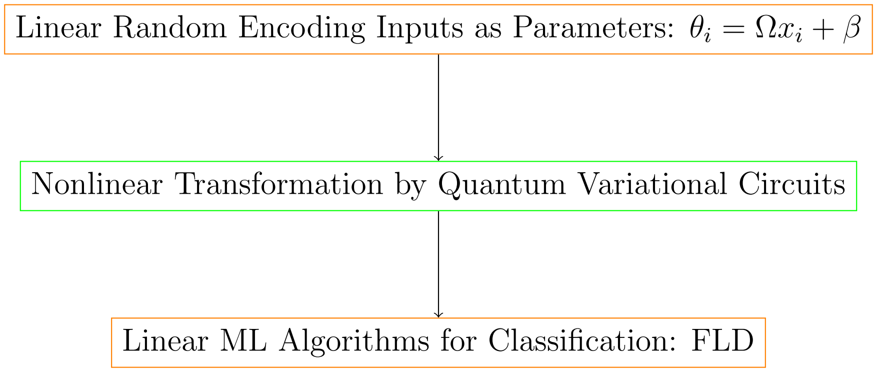

Linear Random Encoding Inputs as Parameters: $\theta_i = \Omega x_i + \beta$
Nonlinear Transformation by Quantum Variational Circuits
Linear ML Algorithms for Classification: FLD

Figure 5.4: Quantum Kitchen Sinks algorithm. The classical nodes are marked with orange, and the quantum node is marked with green.

The QKS is shown in the Fig.5.4 where the input data is classical linear randomly encoded into the quantum variational circuit gate parameters. The quantum variational circuit performs nonlinear transformations of data. We expect to obtain a higher-dimensional feature vector by running the fixed quantum circuit many episodes and stacking those classical outputs of the variational circuit whose parameters are randomly encoded with the same input data. This process matches the fundamental idea of kernel methods, which map the input data into a large Hilbert space of quantum system to perform practical computations on such quantum feature space. Those feature vectors are post-processing by some linear classical machine learning algorithms used for classification. For the quantum Gaussian discord classification task, the bipartite Gaussian states in the data set are described by the covariance matrix. For each Gaussian state, we randomly encode the correlations into circuit parameters and set them as the initial state of the quantum variational circuit.

5.4.2 Classical processes in QKS

To avoid the classical parts in the hybrid algorithm being too powerful, we could require that the classical components of QKS are linear. As a result, we can assume that the power of the algorithm is only added by the nonlinear transformation of the quantum node. The first linear process encodes each input data into the desired number of parameters in quantum variational circuits and repeats for finite episodes. Then, the Fisher linear discriminate (FLD) would use the output of measurements in quantum variational circuits to classify two classes of quantum Gaussian discord.

Classical linear encoding

The first step in QKS is to randomly encode input data as the gate parameters of the quantum variational circuit. Here, we deal with the four entries a, b, c, d of the covariance matrix in standard form in each Gaussian state as classical input data. In other words, each input data is described by a vector in a four dimensional space. To explain the linear encoding process, we could assume that a data set $\mathcal{D} = \{x_1, x_2, ..., x_m\}$ contains m samples and each sample has form $x_i = (a_i, b_i, c_i, d_i)$. Then, a $(p \times 4)$-dimensional weight matrix Ω and a p-dimensional bias vector β could be used to encode input data into desired p gate parameters, giving as

$$\theta_i = \Omega x_i + \beta$$

$$\Rightarrow \begin{pmatrix} \theta_{i1} \\ \theta_{i2} \\ \vdots \\ \theta_{ip} \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} \\ w_{21} & w_{22} & w_{23} & w_{24} \\ \vdots & \vdots & \vdots & \vdots \\ w_{p1} & w_{p2} & w_{p3} & w_{p4} \end{pmatrix} \begin{pmatrix} a_i \\ b_i \\ c_i \\ d_i \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}. \tag{5.9}$$

To ensure all information of the input data encoding into gate parameters, all elements in the weight matrix are designed to be non-zero. The weight matrix Ω is randomly generated following the normal distribution with zero mean and standard deviation σ, and the randomly generated bias vector β has uniform distribution

$$\begin{aligned} \Omega &\sim \mathcal{N}(0, \sigma^2) \\ \beta &\sim \mathcal{U}(0, 2\pi). \end{aligned} \tag{5.10}$$

FLD

Fisher linear discriminant (FLD), also named Linear discriminant analysis (LDA), is a classic linear supervised machine learning algorithm for binary classification, which learns and classifies data using the measurements of quantum variational circuits [29]. Strictly speaking, LDA and FLA are slightly different in that the former assumes the same full-rank covariance matrix for both classes. The primary goal of FLD is to identify the best projection line for separating data from two classes. In the other word, the work is projecting the samples X onto a line with undermined direction v in order to produce a scalar y

$$y = v^T X. \tag{5.11}$$

To best separate two classes, the optimized projection line determined by direction vector v could be obtained by making the scatter in the same class as smaller as possible and making the projected means of different classes as far away from each other as possible, see Fig.5.5. A more mathematical expression maximizes the ratio of the measure of separation between classes to total within-class variance.

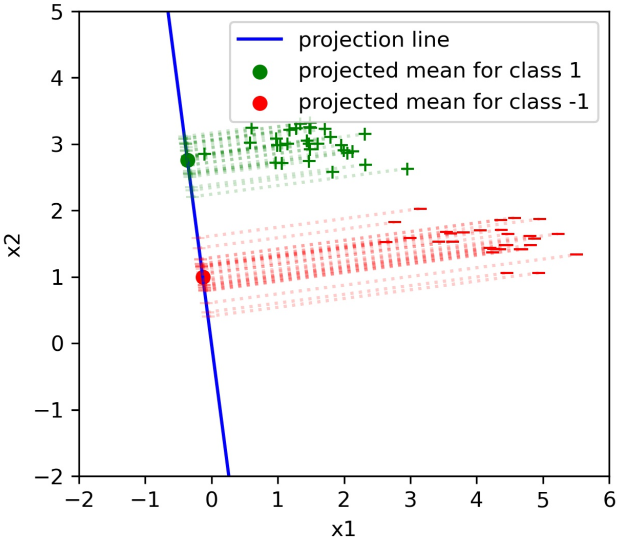


Figure 5.5: 2D scheme of FLD applied on binary classes data. The green "+" and red "-" respectively represent +1 class and -1 class. The blue solid line is the target projection line. The green solid circle and red solid circle are the projected means for +1 class and -1 class, respectively.

Assume a m-sample data set $\mathcal{X} = \{(x_i)|x_i \in \mathbb{R}^d\}$ $(i = 1, 2, ..., m)$, each sample is a d-dimensional vector, and n_1 samples come from -1 class and n_2 samples belong to $+1$ class. The projection of x_i onto a projection line with direction v is given by $y_i = v^T x_i$.

The means of two classes in the original feature space are

$$\begin{aligned} \mu_1 &= \frac{1}{n_1} \sum_{i \in class-1} x_i, \\ \mu_2 &= \frac{1}{n_2} \sum_{i \in class+1} x_i. \end{aligned} \tag{5.12}$$

The means of two classes after projection are

$$\begin{aligned} \tilde{\mu_1} &= \frac{1}{n_1} \sum_{i \in class-1} y_i &= \frac{1}{n_1} \sum_{i \in class-1} v^T x_i = v^T \mu_1, \\ \tilde{\mu_2} &= \frac{1}{n_2} \sum_{i \in class+1} y_i &= \frac{1}{n_2} \sum_{i \in class+1} v^T x_i = v^T \mu_2. \end{aligned} \tag{5.13}$$

The $|\tilde{\mu_1} - \tilde{\mu_2}|$ is the distance between the projected means as well as a measure of separation between two classes. The scatters for projected samples of two classes have expressions

$$\begin{aligned} \tilde{s}_1^2 &= \sum_{i \in class-1} (y_i - \tilde{\mu_1})^2, \\ \tilde{s}_2^2 &= \sum_{i \in class+1} (y_i - \tilde{\mu_2})^2. \end{aligned} \tag{5.14}$$

Obviously, the scatter is similar as variance with additional multiplied by total number of samples.

For well separating two classes, the FLD is to find the optimized direction v of projection line, which can maximize an objective function

$$J = \frac{|\tilde{\mu_1} - \tilde{\mu_2}|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}. \tag{5.15}$$

Firstly, we need to transform the expression of objective function as a function of v so that $J = J(v)$ with the help of scatter matrix. The ”within-class” scatter matrix is:

$$\begin{aligned} S_W &= S_1 + S_2 \\ &= \sum_{i \in class-1} (x_i - \mu_1)(x_i - \mu_1)^T + \sum_{i \in class+1} (x_i - \mu_2)(x_i - \mu_2)^T \end{aligned} \tag{5.16}$$

where S_1, S_2 are the scatter matrices for two classes before projection. Meanwhile, we can re-write the scatter of -1 class $\tilde{s}_1^2$ by substituting $y_i = v^T x_i$ and $\tilde{\mu_1} = v^T \mu_1$, giving

$$\begin{aligned} \tilde{s}_1^2 &= \sum_{i \in class-1} (v^T x_i - v^T \mu_1)^2 \\ &= \sum_{i \in class-1} v^T (x_i - \mu_1)(x_i - \mu_1)^T v \\ &= v^T S_1 v. \end{aligned} \tag{5.17}$$

Similarly, we can obtain $\tilde{s}_2^2 = v^T S_2 v$ so that the denominator of objective function, which is the "within-class" scatter matrix after projection, could re-write as

$$\begin{aligned} \tilde{s}_1^2 + \tilde{s}_2^2 &= v^T S_1 v + v^T S_2 v \\ &= v^T S_W v. \end{aligned} \tag{5.18}$$

The original "between-class" scatter matrix is defined as

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T. \tag{5.19}$$

Furthermore, the separation of two projected means would be expressed as

$$\begin{aligned} (\tilde{\mu_1} - \tilde{\mu_2})^2 &= (v^T \mu_1 - v^T \mu_2)^2 \\ &= v^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T v \\ &= v^T S_B v. \end{aligned} \tag{5.20}$$

As a result, the objective function is expressed in terms of v, S_B and S_W as

$$J(v) = \frac{v^T S_B v}{v^T S_W v}. \tag{5.21}$$

To maximize the objective function $J(v)$, the straight-forward approach is to derive it with respect to v, and find which v making derivation equals to zero.

$$\frac{\partial J(v)}{\partial v} = \frac{2 S_B v (v^T S_W v) - 2 S_W v (v^T S_B v)}{(v^T S_W v)^2} = 0 \tag{5.22}$$

Now, we could solve $S_B v (v^T S_W v) - S_W v (v^T S_B v) = 0$ by firstly dividing by $v^T S_W v$ and then the equation rises to an eigenvector problem as

$$S_W^{-1} S_B v = \lambda v \tag{5.23}$$

if S_W is not singular matrix so that S_W^{-1} exists. Finally, the optimized direction v of the projection line is found in a closed formula

$$v \propto S_W^{-1}(\mu_1 - \mu_2). \tag{5.24}$$

The FLD would project the anonymous data to the optimal projection line determined by training, and the projected data would return into one dimension. More generally, FLD projects samples into a lower-dimensional space, usually smaller than the original dimension of samples' feature space. The FLD could be referred to as a classical and supervised dimension reduction algorithm.

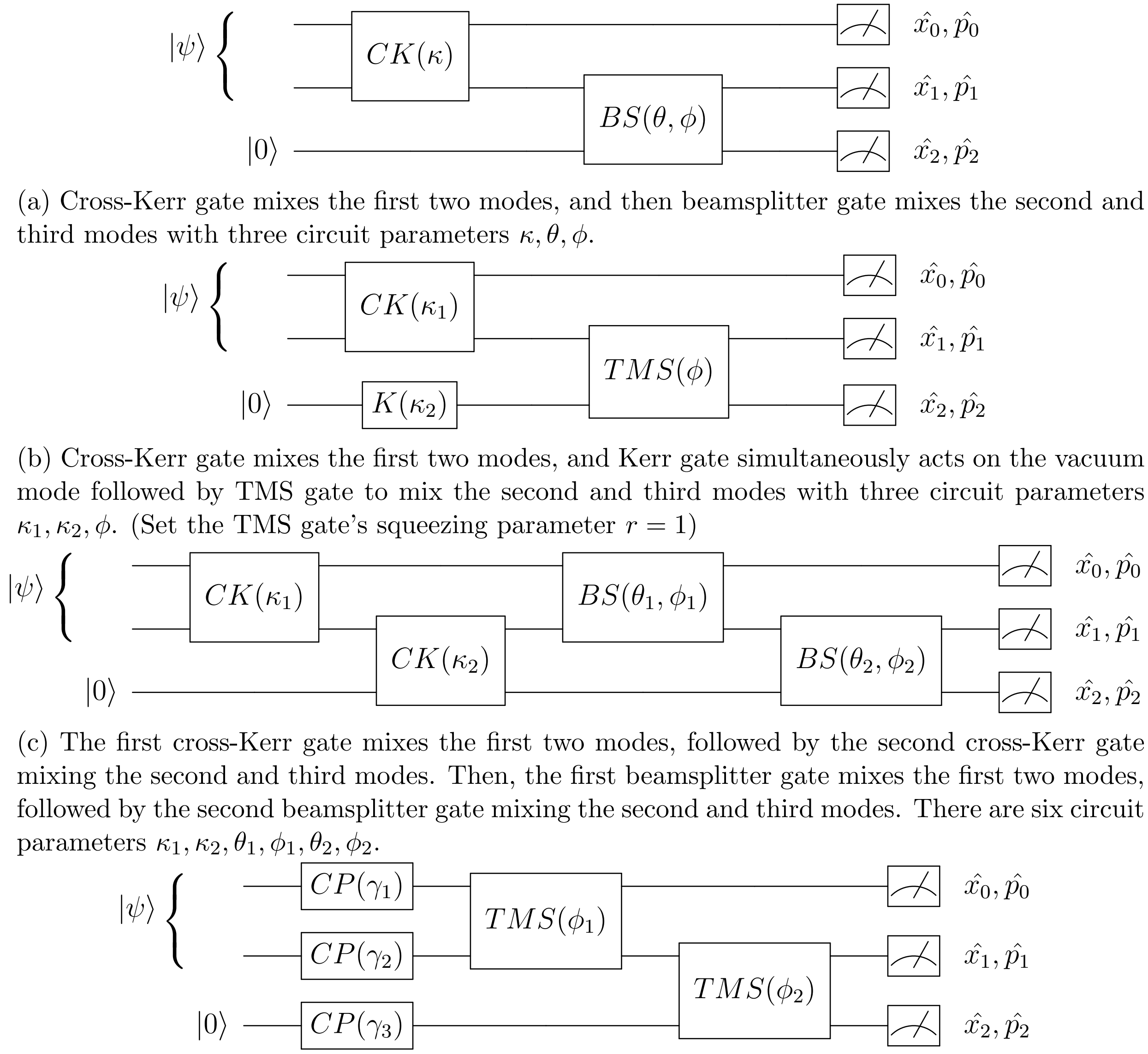
5.4.3 Variational quantum circuit in QKS

The quantum variational circuit, which performs various nonlinear data transformations employing a quantum system, is a key element of the QKS. We implemented different quantum variational circuits to find out which one performs better, see Fig.5.6. Each input data as bipartite Gaussian state $|\psi\rangle$ is prepared as the initial state of the first two modes, and the third mode is in a vacuum state $|0\rangle$. Those quantum variational circuits should contain non-Gaussian gates to ensure nonlinear transformations. Each output of this quantum variational circuit stacks the expectation value $\langle\hat{x}\rangle$ of three modes to get a 3-dimensional feature vector, which is obtained by homodyne measurement on position quadrature observable $\hat{x}$.

5.4.4 QKS Simulation

The parameters of the variational quantum circuit, which is also the model parameters, are obtained by randomly encoding each input data. The number of parameters generated by the encoding should be the same as the number of parameters needed in the circuit.

We evaluate the QKS algorithm's performance based on the system's two key hyperparameters: standard deviation σ and the number of episodes E. The standard deviation σ is used to generate random weight matrices following normal distribution in the classical linear encoding process. We repeat E times for each data point that encodes input data into circuit parameters and measures the quantum variational circuit's classical outputs; thus, we obtain a $(3 \times E)$-dimensional feature vector. The number of episodes E would determine the feature space dimension after feeding data into quantum nodes. The encoding parameter $\{\Omega_e, \beta_e\}$ of each episode should be constant for the training and testing process.



(a) Cross-Kerr gate mixes the first two modes, and then beamsplitter gate mixes the second and third modes with three circuit parameters κ, θ, ϕ.

(b) Cross-Kerr gate mixes the first two modes, and Kerr gate simultaneously acts on the vacuum mode followed by TMS gate to mix the second and third modes with three circuit parameters κ_1, κ_2, ϕ. (Set the TMS gate's squeezing parameter $r = 1$)

(c) The first cross-Kerr gate mixes the first two modes, followed by the second cross-Kerr gate mixing the second and third modes. Then, the first beamsplitter gate mixes the first two modes, followed by the second beamsplitter gate mixing the second and third modes. There are six circuit parameters $\kappa_1, \kappa_2, \theta_1, \phi_1, \theta_2, \phi_2$.

(d) Three cubic phase gates act on three modes. Then, the first TMS gate mixes the first two modes, followed by the second TMS gate mixing the second and third modes with five gate parameters $\gamma_1, \gamma_2, \gamma_3$ and ϕ_1, ϕ_2. (Set the both two TMS gates' squeezing parameters $r = 1$)

Figure 5.6: Various variational quantum circuits as the kitchen sinks. The initial state of the first two modes is prepared as bipartite Gaussian state $|\psi\rangle$, and the third mode is in a vacuum state $|0\rangle$

The initial Gaussian states data set for recognizing Gaussian discord was similar to the data set used to optimize variational quantum circuit, which evenly chose N_S, but with additional over-sampling step to get a balanced binary Gaussian states data set. After applying SMOTE algorithm, the initial Gaussian states data set contains a total of 1224 Gaussian states in which there is an equal number of samples in class -1 and class 1. 70% of the initial Gaussian states data set is holding as the training data set, which consists of 420 Gaussian states in class -1 and 436 Gaussian states in class 1. The size of test data set is 368 with 192 Gaussian states in class -1 and 176 Gaussian states in class 1.

Applying the FLD to classify the initial Gaussian states dataset, the classification accuracy was approximately 98%. However, when $E = 5$ and choosing 10 evenly spaced σ over the interval $[0.1, 10.0]$, the best classification accuracy obtained by implementing first three variational quantum circuits shown in Fig.5.6a, 5.6b and 5.6c were all below 80%, but the fourth circuit (see Fig.5.6d) returned the best accuracy around 95%. Those results indicate two facts: this present Gaussian state data set is too simple to demonstrate QKS's capability in the classification task, and the fourth variational quantum circuit is a strong candidate to implement QKS for recognizing Gaussian discord.

To show the power of QKS in the Gaussian discord classification tasks, we re-constructed the Gaussian states data set, which contains both vacuum and thermal TMS, symmetric BS and asymmetric BS states and "scramble" those states' one mode by an angle ϕ using phase rotation transformation

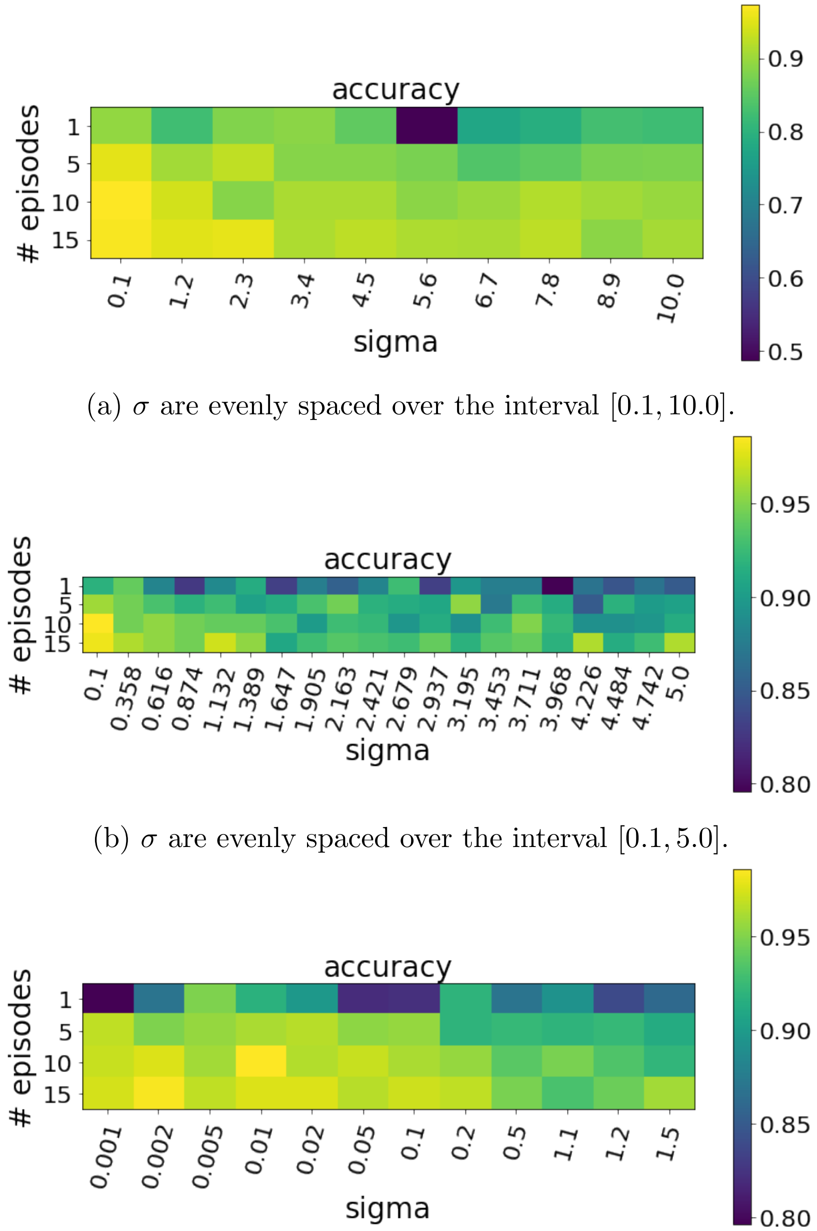
$$\mathbf{V}^{'} = (S_R(\phi) \oplus \mathbf{I})\mathbf{V}(S_R(\phi) \oplus \mathbf{I})^{\dagger} \tag{5.25}$$

where $\mathbf{V}$ is the standard form of covariance matrix shown in 2.16, and $S_R(\phi)$ is the symplectic transformation of phase rotation, see 2.38. We set that the rotational angle is $\phi = \frac{\pi}{4}$, and randomly choose N_S, N_I, N_{th} from the uniform distribution. Based on the pre-processing scheme Fig.5.1, we implement the SMOTE algorithm and hold-out method to get a balanced Gaussian state data set, and 70% of the balanced data set are used as training data set. The size of the training data set is 989 with 493 Gaussian states in class 1 and 496 Gaussian states in class -1; the test data set contains 425 Gaussian states with 214 Gaussian states in class 1 and 211 Gaussian states in class -1.

We chose the variational quantum circuit shown in Fig.5.6d as the kitchen sinks. The three cubic phase gates with circuit parameters $\gamma_1, \gamma_2, \gamma_3$ introduce nonlinear transformation for each mode. Two TMS gates mix the three modes which we set the squeezing parameters r for both equal to one and the squeezing phase angles $\phi_1, \phi_2 \in [0, 2\pi)$ as circuit parameters. As shown in 5.9, we randomly initialized the weight matrix following the normal distribution and the bias vector following the uniform distribution to encode the input data points into those five circuit parameters.

The classification accuracy as the function of standard deviation σ and the number of episodes E are shown in Fig.5.7. We narrow the range of σ to determine the best-performing interval. When the σ were evenly spaced over the interval $[0.1, 10.0]$, the best classification was 97.33% when $\sigma = 0.1$ and $E = 10$, see Fig.5.7a, which is a significant improvement over the baseline 70.12% when only use FLD. For a narrower interval $[0.1, 5.0]$, the best classification achieved was 98.64% when $\sigma = 0.1$ and $E = 10$, see Fig.5.7b. When the σ were not evenly spaced over the interval $[0.001, 1.5]$, the QKS algorithm can achieve the best accuracy up to 98.59% when $\sigma = 0.01$ and $E = 10$, see Fig.5.7c. We can observe that for the Gaussian discord classification task using QKS, the optimal σ is small somewhere around the interval $[0.01, 0.1]$ and the optimal $E = 10$. The resolution of the resulting figure is low since the simulation process is constrained by classical computers, so the simulation process for different hyper-parameters usually takes a long time.

Based on the simulation results, we proved the QKS algorithm's power by successfully solving the Gaussian discord binary classification task. Even though we implemented the variational quantum circuit with only 3-qumode, we showed that the classification accuracy is significantly improved compared with the case that only uses the classical linear classification algorithm.



(a) σ are evenly spaced over the interval $[0.1, 10.0]$.

(b) σ are evenly spaced over the interval $[0.1, 5.0]$.

(c) σ are not evenly spaced over the interval $[0.001, 1.5]$.

Figure 5.7: QKS simulation results: the classification accuracy as the function of standard deviation σ and the number of episodes E. (a) The σ are evenly spaced over the interval $[0.1, 10.0]$. (b) The σ are evenly spaced over the interval $[0.1, 5.0]$. (c) The σ are not evenly spaced over the interval $[0.001, 1.5]$.

Chapter 6

Conclusion

In chapter 2, we reviewed relevant concepts of the CV quantum computing systems, including the standard Gaussian states, a family of universal CV gates and different measurement types for CV systems.

Chapter 3 focused on the derivation process of quantum discord, which is the discrepancy between two quantum analogues of classical equivalent expressions of mutual information. We then explored the quantum discord in the CV systems and presented two approaches to obtain the analytical solutions of Gaussian quantum discord.

In chapter 4, we overviewed classical machine learning, the neural network in deep learning and how to update parameters of neural networks by back-propagating algorithm. We presented the results that recognized the Gaussian quantum discord by back-propagating with the double hidden layers neural network.

Chapter 5 began with expanding classical machine learning to quantum machine learning. PennyLane, a machine learning library on Python, is used to simulate variational quantum circuits. We then presented how to implement the supervised hybrid optimization of a variational quantum circuit for recognizing Gaussian discord. We also introduced the typical pre-processing that balances the original data set and then separates the data set into training and testing data set before directly feeding inputs in the models. We showed the capability of a hybrid quantum-classical machine learning algorithm called QKS, which is belonged to quantum kernel methods. The variational quantum circuits, the core of QKS, significantly improve the classification accuracy over the classical linear classifier so that work out the Gaussian discord classification task.

6.1 Future Directions

We just have contributed the theoretical work to simulate the QKS algorithm in classical computers for recognizing the Gaussian discord. To evaluate the QKS performance on the Gaussian discord classification task, the Gaussian state data set, containing the covariance matrices of a family of Gaussian states as inputs and corresponding Gaussian discord as labels, are artificially set based on experience and calculations. However, such Gaussian states data set does not provide a comprehensive picture of the quantum nature of the Gaussian states.

We can further test the capability of QKS for realizing Gaussian discord by experiment. The superconducting parametric cavity has showed its power to develop CV quantum computing by experimentally implementing the QKS algorithms to classify a synthetic data set [24]. One direction is to experimentally generate the Gaussian state to obtain a more practical Gaussian state data set, and also implement the variational quantum circuits following the experiment setup in [24].

References

[1] Gerardo Adesso and Animesh Datta. Quantum versus classical correlations in gaussian states. *Physical review letters*, 105(3):030501, 2010.

[2] Gerardo Adesso and Fabrizio Illuminati. Entanglement in continuous-variable systems: recent advances and current perspectives. *Journal of Physics A: Mathematical and Theoretical*, 40(28):7821, 2007.

[3] Mazhar Ali, ARP Rau, and Gernot Alber. Quantum discord for two-qubit x states. *Physical Review A*, 81(4):042105, 2010.

[4] SM Barnett and PM Radmore. Methods in theoretical quantum optics, clarendon, 1997.

[5] Marcello Benedetti, John Realpe-Gómez, and Alejandro Perdomo-Ortiz. Quantum-assisted helmholtz machines: A quantum–classical deep learning framework for industrial datasets in near-term devices. *Quantum Science and Technology*, 3(3):034007, 2018.

[6] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[8] Animesh Datta, Anil Shaji, and Carlton M Caves. Quantum discord and the power of one qubit. *Physical review letters*, 100(5):050502, 2008.

[9] Maurice A De Gosson. *Symplectic geometry and quantum mechanics*, volume 166. Springer Science & Business Media, 2006.

[10] Lu-Ming Duan, Géza Giedke, Juan Ignacio Cirac, and Peter Zoller. Inseparability criterion for continuous variable systems. *Physical Review Letters*, 84(12):2722, 2000.

[11] Biswadeb Dutta, Narasimhaiengar Mukunda, Rajiah Simon, et al. The real symplectic groups in quantum mechanics and optics. *Pramana*, 45(6):471–497, 1995.

[12] Jens Eisert, Stefan Scheel, and Martin B Plenio. Distilling gaussian states with gaussian operations is impossible. *Physical review letters*, 89(13):137903, 2002.

[13] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[14] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[15] Alessandro Ferraro, Stefano Olivares, and Matteo GA Paris. Gaussian states in continuous variable quantum information. *arXiv preprint quant-ph/0503237*, 2005.

[16] Richard P Feynman. Quantum mechanical computers. *Optics news*, 11(2):11–20, 1985.

[17] Richard P Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press, 2018.

[18] Jaromír Fiurášek. Gaussian transformations and distillation of entangled gaussian states. *Physical review letters*, 89(13):137904, 2002.

[19] Michael P Frank, Liviu Oniciuc, Uwe H Meyer-Baese, and Irinel Chiorescu. A space-efficient quantum computer simulator suitable for high-speed fpga implementation. In *Quantum Information and Computation VII*, volume 7342, page 734203. International Society for Optics and Photonics, 2009.

[20] Géza Giedke and J Ignacio Cirac. Characterization of gaussian operations and distillation of gaussian states. *Physical Review A*, 66(3):032316, 2002.

[21] Leah Henderson and Vlatko Vedral. Classical, quantum and total correlations. *Journal of physics A: mathematical and general*, 34(35):6899, 2001.

[22] A Holevo. Some statistical problems for quantum gaussian states. *IEEE Transactions on Information Theory*, 21(5):533–543, 1975.

[23] Robin L Hudson. When is the wigner quasi-probability density non-negative? *Reports on Mathematical Physics*, 6(2):249–252, 1974.

[24] Shih-Chun Hung. Quantum computation with superconducting parametric cavity. Master's thesis, University of Waterloo, 2020.

[25] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019.

[26] Nathan Killoran, Josh Izaac, Nicol'as Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. Strawberry Fields: A software platform for photonic quantum computing. *Quantum*, 3:129, 2019.

[27] Teuvo Kohonen. The'neural'phonetic typewriter. *computer*, 21(3):11–22, 1988.

[28] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[29] Cheng Li and Bingyu Wang. Fisher linear discriminant analysis. *CCIS Northeastern University*, 2014.

[30] Nan Li and Shunlong Luo. Total versus quantum correlations in quantum states. *Physical Review A*, 76(3):032327, 2007.

[31] Tian-Yu Liu. Easyensemble and feature selection for imbalance data sets. In *2009 international joint conference on bioinformatics, systems biology and intelligent computing*, pages 517–520. IEEE, 2009.

[32] Seth Lloyd and Samuel L Braunstein. Quantum computation over continuous variables. In *Quantum information with continuous variables*, pages 9–17. Springer, 1999.

[33] Shunlong Luo. Quantum discord for two-qubit systems. *Physical Review A*, 77(4):042303, 2008.

[34] Shunlong Luo. Using measurement-induced disturbance to characterize correlations as classical or quantum. *Physical Review A*, 77(2):022301, 2008.

[35] A Maldonado-Trapp, Anzi Hu, and Luis Roa. Analytical solutions and criteria for the quantum discord of two-qubit x-states. *Quantum Information Processing*, 14(6):1947–1958, 2015.

[36] Leonard Mandel and Emil Wolf. *Optical coherence and quantum optics*. Cambridge university press, 1995.

[37] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[38] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[39] Ryszard S Michalski. A theory and methodology of inductive learning. In *Machine learning*, pages 83–134. Elsevier, 1983.

[40] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[41] Kavan Modi, Tomasz Paterek, Wonmin Son, Vlatko Vedral, and Mark Williamson. Unified view of quantum and classical correlations. *Physical review letters*, 104(8):080501, 2010.

[42] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[43] Harold Ollivier and Wojciech H Zurek. Quantum discord: a measure of the quantumness of correlations. *Physical review letters*, 88(1):017901, 2001.

[44] Harold Ollivier and Wojciech H. Zurek. Quantum discord: A measure of the quantumness of correlations. *Phys. Rev. Lett.*, 88:017901, Dec 2001.

[45] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[46] Marco Piani, Paweł Horodecki, and Ryszard Horodecki. No-local-broadcasting theorem for multipartite quantum correlations. *Physical review letters*, 100(9):090502, 2008.

[47] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[48] Haleema Sadia Qureshi, Shakir Ullah, and Fazal Ghafoor. Hierarchy of quantum correlations using a linear beam splitter. *Scientific reports*, 8(1):1–10, 2018.

[49] Ali Rahimi and Benjamin Recht. Uniform approximation of functions with random bases. In *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 555–561. IEEE, 2008.

[50] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: replacing minimization with randomization in learning. In *Nips*, pages 1313–1320. Citeseer, 2008.

[51] Ali Rahimi, Benjamin Recht, et al. Random features for large-scale kernel machines. In *NIPS*, volume 3, page 5. Citeseer, 2007.

[52] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[53] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.

[54] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[55] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.

[56] Alessio Serafini. Multimode uncertainty relations and separability of continuous variable states. *Physical review letters*, 96(11):110402, 2006.

[57] R Simon, N Mukunda, and Biswadeb Dutta. Quantum-noise matrix for multimode systems: U (n) invariance, squeezing, and normal forms. *Physical Review A*, 49(3):1567, 1994.

[58] Rajiah Simon. Peres-horodecki separability criterion for continuous variable systems. *Physical Review Letters*, 84(12):2726, 2000.

[59] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[60] Steven J van Enk and Christopher A Fuchs. The quantum state of a propagating laser field. *arXiv preprint quant-ph/0111157*, 2001.

[61] V Vedral. Classical correlations and entanglement in quantum measurements. *Physical review letters*, 90(5):050401, 2003.

[62] DF Walls and GJ Milburn. Quantum optics springer-verlag. *New York*, 1994.

[63] Paul Werbos. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974.

[64] John Williamson. On the algebraic problem concerning the normal forms of linear dynamical systems. *American journal of mathematics*, 58(1):141–163, 1936.

[65] CM Wilson, JS Otterbach, Nikolas Tezak, RS Smith, AM Polloreno, Peter J Karalekas, S Heidel, M Sohaib Alam, GE Crooks, and MP da Silva. Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers. *arXiv preprint arXiv:1806.08321*, 2018.