



UNIVERSIDADE ESTADUAL PAULISTA
“JULIO DE MESQUITA FILHO”
INSTITUTO DE GEOCIÊNCIAS E CIÊNCIAS
EXATAS



Trabalho de Conclusão de Curso
Curso de Graduação em Física

**Técnicas de Aprendizado de Máquina para a
Classificação e Previsão de Estados Quânticos via
Efeito Stern-Gerlach Óptico.**

Gustavo Del Duque

Ricardo Paupitz Barbosa dos Santos

Rio Claro-SP
2024

UNIVERSIDADE ESTADUAL PAULISTA
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

GUSTAVO DEL DUQUE

Técnicas de Aprendizado de Máquina para a
Classificação e Previsão de Estados Quânticos via Efeito Stern-Gerlach
Óptico.

Trabalho de Conclusão de Curso apresentado ao Instituto de Geociências e Ciências Exatas - Câmpus de Rio Claro, da Universidade Estadual Paulista Júlio de Mesquita Filho, para obtenção do grau de Bacharel em Física.

Rio Claro (SP)
2024

D946t

Duque, Gustavo Del

Técnicas de Aprendizado de Máquina para a Classificação e Previsão de Estados Quânticos via Efeito Stern-Gerlach Óptico. / Gustavo Del Duque. -- Rio Claro, 2024

43 p.

Trabalho de conclusão de curso (Bacharelado - Física) - Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro

Orientador: Ricardo Paupitz Barbosa dos Santos

Coorientador: Daniel Zini Rossatto

1. Aprendizado de Máquina. 2. Efeito Stern-Gerlach óptico. 3. Classificação de estados quânticos. I. Título.

GUSTAVO DEL DUQUE

Técnicas de Aprendizado de Máquina para a
Classificação e Previsão de Estados Quânticos via Efeito
Stern-Gerlach Óptico.

Trabalho de Conclusão de Curso apresentado ao Instituto de Geociências e Ciências Exatas - Câmpus de Rio Claro, da Universidade Estadual Paulista Júlio de Mesquita Filho, para obtenção do grau de Bacharel em Física

Comissão Examinadora

Prof. Dr. Ricardo Paupitz Barbosa dos Santos

Prof. Dr. Daniel Zini Rossatto

Prof. Dr. Lucas Fugikawa Santos

Prof. Dr. Makoto Yoshida (Suplente)

Rio Claro, 08 de novembro de 2024.

Assinatura da(o) aluna(o)

Assinatura da(o) orientadora(o)

Agradecimentos

O seguinte trabalho foi fomentado no processo nº2022/15394-3, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP). As opiniões, hipóteses e conclusões ou recomendações expressas neste material são de responsabilidade do(s) autor(es) e não necessariamente refletem a visão da FAPESP.

Gostaria de expressar minha sincera gratidão a todos que contribuíram para a realização deste trabalho. Em primeiro lugar, ao meu orientador Prof. DR. Ricardo Paupitz, pela paciência e sabedoria, e ao meu co-orientador, Prof. Dr. Daniel Zini Rossatto, que esteve comigo desde meu segundo ano do curso, sendo figura principal do meu desenvolvimento acadêmico, e que apesar da distância entre as instituições foi um excelente mentor.

A todos os professores e amigos, em especial Giovanna Borim Trindade que emprestou seu notebook por um tempo para eu finalizar este trabalho, e meu colega de casa, Pedro Lopez Henrique Nanzer por todo o apoio e ajuda ao longo da graduação, a minha namorada Isabelle que esteve do meu lado nos piores e melhores dias da minha graduação, a minha família que, mesmo longe estiveram sempre torcendo por mim e me dando apoio independente de qualquer coisa. Este trabalho só foi possível graças ao suporte e ao ambiente acadêmico que tive a honra de fazer parte.

Resumo

Nos últimos anos, as aplicações de técnicas de aprendizado de máquina têm sido estudadas ativamente e de forma cada vez mais relevante. Para adquirir informações sobre um sistema quântico, é essencial compreender seu estado de maneira geral, uma tarefa que pode se tornar desafiadora devido à complexidade associada ao aumento da dimensão do espaço de Hilbert. Durante a realização deste trabalho, foram utilizados algoritmos de máquinas de vetores de suporte para classificar e prever características como o número médio de fótons do estado do campo eletromagnético de um modo de uma cavidade óptica. Para isso, a máquina foi submetida a um processo de treinamento e alimentação baseado nas alturas e distâncias dos picos do padrão de espalhamento observado em um feixe atômico ao atravessar a cavidade em um estado específico (conhecido como efeito Stern-Gerlach óptico). Os estados para a cavidade considerados neste estudo foram Fock, coerente, gato e comprimido. Empregamos 100 amostras para cada tipo de estado, utilizando 60% dos dados para treinamento e reservando 40% para teste, para os quais a máquina retornou acurácias maiores ou iguais a 96% dependendo do estado do átomo quando este atravessa a cavidade e da tarefa da máquina.

Palavras-chave: Stern-Gerlach Óptico, Aprendizado de Máquina, classificação de Estados, Estados Quânticos, Cavidade Óptica.

Abstract

In recent years, the application of machine learning techniques has been actively and increasingly studied. Acquiring information about a quantum system requires a comprehensive understanding of its state, a task that becomes challenging due to the complexity associated with the growing dimension of the Hilbert space. In this work, support vector machine algorithms were utilized to classify and predict features such as the average photon number in the electromagnetic field state of an optical cavity mode. The machine was trained and fed with data derived from the heights and distances of peaks in the scattering pattern observed in an atomic beam traversing the cavity in a specific state (known as the optical Stern-Gerlach effect). The cavity states considered in this study include Fock, coherent, cat, and squeezed states. For each type of state, 100 samples were employed, with 60% used for training and 40% for testing. The machine demonstrated accuracies of 96% or higher, depending on the atom's state as it traverses the cavity and the task assigned to the machine.

Keywords: Stern-Gerlach Effect, Machine learning, Quantum States, State Classification, Optical Cavity Mode.

Lista de Figuras

1	Esquema do experimento proposto na Referência. [1]. Um campo clássico prepara os átomos de dois níveis em uma superposição coerente (zona de Ramsey). Depois, o feixe atômico passa por uma fenda estreita que o confina em um região em torno do nó do campo eletromagnético estacionário dentro da cavidade. Após uma interação ressonante com o campo da cavidade, os átomos viajam até um anteparo distante. A distribuição espacial nesse anteparo reflete a distribuição de momento dos átomos logo após sua interação com o campo.	11
2	Em 2a temos a visualização de dois possíveis hiperplanos que conseguiria classificar as duas amostras, e em 2b temos qual seria a escolha do melhor hiperplano.	18
3	Em 3a temos a visualização dos dados das duas amostras, e em 3b é usado $z = x^2 + y^2$ para facilitar a análise.	19
4	Análise do hiperplano via <i>kernel</i> = <i>linear</i> , <i>RBF</i> e <i>Poly</i> , respectivamente da esquerda para direita. Imagens tiradas de [2].	20
5	Análise do hiperplano via $C = 0.1, 1, 10$, respectivamente da esquerda para direita. Imagens tiradas de [2].	20
6	Análise do hiperplano via $\gamma = 0.1, 1, 10$, respectivamente da esquerda para direita. Imagens tiradas de [2].	20
7	Distribuição do momento atômico, que representa a imagem de distribuição espacial gerada no anteparo distante, considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado $ 0\rangle$ (painéis superiores) e para o campo no estado $ 2\rangle$ (painéis inferiores).	22
8	Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado coerente de amplitude $\alpha = 1$ (painéis superiores) e de $\alpha = 3$ (painéis inferiores).	23
9	Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado de gato par (painéis superiores) e ímpar (painéis inferiores) assumindo $\alpha = 3$	24
10	Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado de gato par (painéis superiores) e ímpar (painéis inferiores) assumindo $\alpha = 5$	24
11	Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo da cavidade no estado de comprimido assumindo $\xi = 1$	25

- 12 Matriz de confusão considerando $c_g = 0$ com 97% de acurácia, $c_g = \frac{1}{\sqrt{2}}$ com 96% de acurácia e $c_g = 1$ com 98% de acurácia, respectivamente. Os rótulos 0, 1, 2, 3 e 4 correspondem, respectivamente, ao estado de Fock, estado coerente, estado de gato par, estado de gato ímpar e estado de vácuo comprimido. 27
- 13 Regressão para o número médio de fótons, da esquerda para a direita $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, todas com 99% de acurácia. 29
- 14 Exemplo de Matriz Confusão 43

Sumário

1	Introdução	9
2	Efeito Stern-Gerlach Óptico	10
2.1	Estado Inicial	11
2.2	Evolução Temporal do Estado	12
2.3	Distribuição de Momento dos Átomos	13
2.4	Distribuição do Estado de fock	14
2.5	Estado Coerente	15
2.6	Estados de Gato Óptico	16
2.7	Distribuição do Estado de Vácuo Comprimido	16
3	Aprendizado de Máquina	18
3.1	Máquina de Vetores de Suporte	18
3.1.1	Hiperparâmetros	19
4	Visualização da Distribuição do Momento dos Átomos	22
4.1	Distribuição do Momento dos Átomos para o Estado de Fock	22
4.2	Distribuição do Momento dos Átomos para o Estado Coerente	23
4.3	Distribuição do Momento dos Átomos para o Estado de Gato Óptico	23
4.4	Distribuição do Momento dos Átomos para o Estado de Vácuo Comprimido	24
5	Treinamento da Máquina	25
5.1	Aprendizado Supervisionado	25
5.2	Validação Cruzada	26
5.3	Classificação e previsão dos estados quânticos	26
5.4	Regressão por Vetores de Suporte (SVR)	28
	Referências	31
A	Algoritmo de classificação	33
B	Algoritmo de Regressão	38
C	Análise da Matriz Confusão	43

1 Introdução

Nos últimos anos, o avanço das tecnologias quânticas trouxe uma nova perspectiva para o processamento de informações e a manipulação de estados quânticos. A capacidade de controlar e medir esses estados com alta precisão tem impulsionado pesquisas em áreas como a computação quântica, criptografia e informação quântica [3, 4]. Um dos fenômenos fundamentais para entender a natureza de partículas quânticas é o Efeito Stern-Gerlach, amplamente conhecido por sua demonstração da quantização do momento angular e por ser um dos experimentos-chave para a compreensão do comportamento das partículas [5].

Embora o efeito Stern-Gerlach clássico envolva partículas com momento angular intrínseco (spin) ao atravessarem um campo magnético, recentes avanços em ótica quântica têm permitido expandir esse conceito para o campo da luz, utilizando sistemas ópticos como base para experimentos que imitam o comportamento de partículas massivas. O uso de fótons e outros quanta de luz em experimentos análogos ao Stern-Gerlach abre novas oportunidades para controlar e medir estados quânticos com alta precisão, essencial para o desenvolvimento de tecnologias quânticas [6, 7].

Paralelamente, o surgimento de técnicas de aprendizado de máquina tem se mostrado um divisor de águas na capacidade de processamento de grandes volumes de dados e na identificação de padrões complexos em sistemas físicos [8, 9, 10]. Em particular, a combinação entre aprendizado de máquina e experimentos quânticos tem o potencial de revolucionar a forma como medimos, interpretamos e classificamos estados quânticos [11]. Em um cenário onde o número de estados quânticos possíveis cresce exponencialmente com o número de partículas envolvidas, as técnicas convencionais de análise se tornam ineficazes. O uso de *machine learning* permite, então, um tratamento mais eficiente desses dados, promovendo uma melhor compreensão e controle dos sistemas quânticos.

Este trabalho teve como objetivo investigar a aplicação de técnicas de aprendizado de máquina na classificação de estados quânticos gerados via um experimento Stern-Gerlach óptico. A relevância deste estudo reside no fato de que a classificação correta de estados quânticos é um passo crítico para diversas áreas da física e da tecnologia, como o desenvolvimento de algoritmos de computação quântica, simulações de sistemas quânticos e protocolos de comunicação segura [12].

Revisão Bibliográfica

2 Efeito Stern-Gerlach Óptico

O efeito Stern-Gerlach é um experimento icônico da física quântica, responsável por demonstrar a quantização do momento angular e a existência do spin em partículas. No experimento original, um feixe de átomos é enviado através de um campo magnético não homogêneo, resultando na separação espacial das componentes do spin. Este experimento revolucionou a compreensão da mecânica quântica, confirmando a quantização do momento angular e introduzindo uma nova maneira de estudar propriedades intrínsecas de partículas, como o spin [5] .

Nos anos 1990, esse conceito foi estendido para o campo da óptica, dando origem ao efeito Stern-Gerlach óptico, onde átomos interagem com um campo eletromagnético confinado em cavidades ópticas em vez de um campo magnético. A versão óptica deste efeito introduz novas oportunidades para a medição e manipulação de estados quânticos, especialmente em cavidades ressonantes, onde a interação entre átomos e o campo eletromagnético confinado produz padrões de deflexão atômica que contêm informações sobre o estado quântico do campo [6, 13].

Nesse cenário, o campo eletromagnético na cavidade é tratado como um campo quântico, e a interação com os átomos resulta em um deslocamento de momento, gerando um padrão de deflexão. Como ilustrado na Figura. 1, o feixe de átomos é espalhado ao passar por uma cavidade ressonadora, e a distribuição espacial dos átomos defletidos reflete a distribuição de momento dos átomos logo após a interação. Este padrão espacial carrega informações valiosas sobre o estado do campo na cavidade, permitindo a realização de uma tomografia de estados quânticos [1].

Ao longo dos anos, diversas técnicas foram propostas para a reconstrução do estado do campo na cavidade a partir dos dados obtidos pelo padrão de deflexão atômica. Métodos destrutivos [14] e não destrutivos [15] têm sido desenvolvidos para capturar essas informações, com avanços permitindo análises mais detalhadas e eficientes. Esses métodos possibilitam a extração de informações quânticas relevantes, como a coerência do estado e as correlações entre os modos do campo.

Além disso, o uso de múltiplas cavidades em experimentos mais recentes expandiu o alcance do efeito Stern-Gerlach óptico. A inclusão de duas cavidades cruzadas, por exemplo, permitiu a investigação de interações entre modos de campos eletromagnéticos distintos, abrindo portas para a criação de novas tecnologias quânticas, como litografia atômica quântica bidimensional e a detecção de emaranhamento quântico [7, 16].

No contexto deste trabalho, o efeito Stern-Gerlach óptico é explorado como uma ferramenta para gerar dados experimentais ricos em informações sobre estados quânticos, o que levanta desafios na análise e reconstrução desses estados a partir dos padrões de deflexão atômica. A quantidade e complexidade dos dados gerados por esses experimentos

requerem técnicas avançadas para sua análise, sendo aqui que o aprendizado de máquina se torna uma ferramenta essencial.

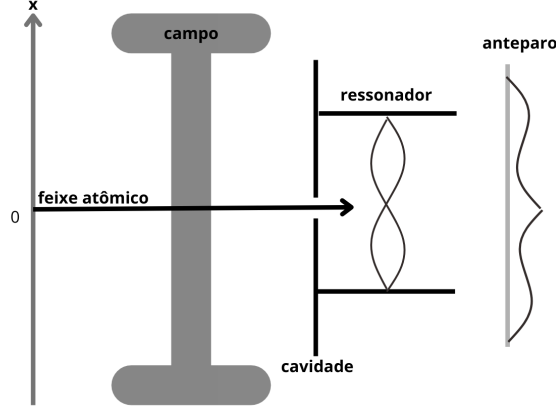


Figura 1: Esquema do experimento proposto na Referência. [1]. Um campo clássico prepara os átomos de dois níveis em uma superposição coerente (zona de Ramsey). Depois, o feixe atômico passa por uma fenda estreita que o confina em uma região em torno do nó do campo eletromagnético estacionário dentro da cavidade. Após uma interação ressonante com o campo da cavidade, os átomos viajam até um anteparo distante. A distribuição espacial nesse anteparo reflete a distribuição de momento dos átomos logo após sua interação com o campo.

Neste trabalho, consideramos a descrição quântica de um sistema atômico acoplado a um campo de cavidade óptica, em que o acoplamento é descrito pelo hamiltoniano de Jaynes-Cummings. O processo segue o regime de Raman-Nath¹, e o Hamiltoniano de interação é dado por

$$H = -\mu\varepsilon_0 kx(\sigma_+ a + a^\dagger \sigma_-), \quad (1)$$

onde μ é o momento de dipolo atômico, ε_0 é o campo elétrico efetivo por fóton, $k = 2\pi/\lambda$ é o número de onda do campo, e a ($a^\dagger$) é o operadores de aniquilação (criação) do modo de cavidade. Os operadores atômicos $\sigma_+ = |e\rangle\langle g|$ e $\sigma_- = |g\rangle\langle e|$ atuam nos estados quânticos do átomo, onde $|g\rangle$ e $|e\rangle$ são os estados fundamental e excitado, respectivamente.

2.1 Estado Inicial

O estado inicial do sistema consiste em uma superposição do estado atômico $|g\rangle$ e $|e\rangle$, enquanto o campo de cavidade está em uma superposição de estados de número de fóton $|m\rangle$.

¹Nesse regime, despreza-se a energia cinética do átomo, assumindo que o deslocamento do átomo na cavidade é pequeno quando comparado ao comprimento de onda do campo.

$$\begin{aligned}
|\phi(x, t=0)\rangle &= \langle x | \text{atom} \rangle \otimes |\text{field}\rangle \\
&= f(x)(c_g |g\rangle + c_e |e\rangle) \otimes \sum_{m=0}^{\infty} c_m |m\rangle,
\end{aligned} \tag{2}$$

onde c_g e c_e são as amplitudes dos estados atômicos $|g\rangle$ e $|e\rangle$, respectivamente, e c_m é a amplitude de encontrar o campo no estado com m fótons. A função espacial $f(x)$ descreve a distribuição espacial dos átomos ao entrarem na cavidade, com:

$$f(x) = (\pi \Delta x^2)^{-1/4} e^{(-x^2/2\Delta x^2)}, \tag{3}$$

onde Δx é a largura da distribuição espacial.

2.2 Evolução Temporal do Estado

A evolução temporal do sistema é governada pelo hamiltoniano de Jaynes-Cummings. Para calcular o estado do sistema após um tempo de interação τ ($|\phi(x, \tau)\rangle = e^{-iH\tau} |\phi(x, t=0)\rangle$), instante no qual o átomo atravessa a cavidade e se torna uma partícula livre novamente, decompomos o estado inicial em termos dos autoestados do Hamiltoniano H . Os autoestados (ou estados vestidos) são dados por:

$$|g, 0\rangle = |g\rangle \otimes |m\rangle \quad \text{com} \quad E_0 = 0; \tag{4}$$

$$|\pm, m\rangle = \frac{1}{\sqrt{2}} (|g\rangle \otimes |m\rangle \pm |e\rangle \otimes |m-1\rangle) \quad \text{com} \quad E_{\pm} = \pm g(x)\sqrt{m} \quad (m \geq 1), \tag{5}$$

onde $g(x) = -\mu\epsilon_0 kx$ é o acoplamento atômico-campo dependente da posição. Assim:

$$\begin{aligned}
|\phi(x, t=0)\rangle &= f(x)c_0c_g |g, 0\rangle + \frac{f(x)}{\sqrt{2}} \sum_{m=1}^{\infty} [c_m c_g (|+, m\rangle + |-, m\rangle) + c_{m-1} c_e (|+, m\rangle - |-, m\rangle)] \\
&= f(x)c_0c_g |g, 0\rangle + \frac{f(x)}{\sqrt{2}} \sum_{m=1}^{\infty} [(c_m c_g + c_{m-1} c_e) |+, m\rangle + (c_m c_g - c_{m-1} c_e) |-, m\rangle].
\end{aligned} \tag{6}$$

Com isso, a evolução temporal do estado inicial $|\phi(x, t=0)\rangle$ após um tempo τ é dada por:

$$\begin{aligned}
|\phi(x, \tau)\rangle &= e^{-iH\tau} |\phi(x, t=0)\rangle \\
&= f(x) c_0 c_g |g, 0\rangle + \frac{f(x)}{\sqrt{2}} \sum_{m=1}^{\infty} [(c_m c_g + c_{m-1} c_e) e^{-iE_+ \tau} |+, m\rangle + (c_m c_g - c_{m-1} c_e) e^{-iE_- \tau} |-, m\rangle] \\
&= f(x) c_0 c_g |g, 0\rangle + \frac{f(x)}{\sqrt{2}} \sum_{m=1}^{\infty} [(c_m c_g + c_{m-1} c_e) e^{i\sqrt{m}\kappa k x} |+, m\rangle + (c_m c_g - c_{m-1} c_e) e^{-i\sqrt{m}\kappa k x} |-, m\rangle],
\end{aligned} \tag{7}$$

onde $\kappa = \mu \varepsilon_0 \tau$.

2.3 Distribuição de Momento dos Átomos

Após o tempo de interação, a distribuição de momento dos átomos $W(p)$ é uma quantidade observável importante onde $W(p) = \langle \phi(p, \tau) | \phi(p, \tau) \rangle^2$. A função de onda no espaço de momento é dada pela transformada de Fourier da função de onda no espaço de posição.

$$|\phi(p, \tau)\rangle = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dx e^{-ipx} |\phi(x, \tau)\rangle. \tag{8}$$

Substituímos a expressão para o estado $|\phi(x, \tau)\rangle$ e introduzindo a distribuição de momento adimensional $\wp = p/k$. Temos

$$\begin{aligned}
|\phi(\wp, \tau)\rangle &= \sqrt{2} \sum_{m=1}^{\infty} \hat{f}(\wp - \sqrt{m}\kappa) (c_m c_g + c_{m-1} c_e) |+, m\rangle \\
&\quad + \sqrt{2} \sum_{m=1}^{\infty} \hat{f}(\wp + \sqrt{m}\kappa) (c_m c_g - c_{m-1} c_e) |-, m\rangle \\
&\quad + 2\hat{f}(\wp) c_0 c_g |g, 0\rangle,
\end{aligned} \tag{9}$$

onde $\hat{f}(\wp \pm \sqrt{m}\kappa)$ é a transformada de Fourier da função de distribuição espacial $f(x)$.

$$\begin{aligned}
\hat{f}(\wp \pm \sqrt{m}\kappa) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(u/k) e^{-i(\wp \pm \sqrt{m}\kappa)u} du \\
&= \frac{1}{\sqrt{k}} \left[\frac{\pi}{4} \left(\frac{\Delta x}{\lambda} \right)^2 \right]^{1/4} \exp \left[-2\pi^2 \left(\frac{\Delta x}{\lambda} \right)^2 \left(\frac{\wp}{\kappa} \pm \sqrt{m} \right)^2 \right].
\end{aligned} \tag{10}$$

Finalmente, a distribuição de momento $W(\wp)$ é obtida como:

$$\begin{aligned}
W(\wp) = & 2 \sum_{m=0}^{\infty} \left| \sqrt{k} \hat{f}(\wp - \sqrt{m}\kappa) \right|^2 |c_m c_g + c_{m-1} c_e|^2 \\
& + 2 \sum_{m=1}^{\infty} \left| \sqrt{k} \hat{f}(\wp + \sqrt{m}\kappa) \right|^2 |c_m c_g - c_{m-1} c_e|^2 \\
& + 2 |c_0 c_g|^2 \left| \hat{f}(\wp) \right|^2.
\end{aligned} \tag{11}$$

Essa Expressão nos diz como que o Momento dos átomos será distribuída em um anteparo distante do ressonador em um tempo $T = \tau$, formando um padrão de espalhamento atômico, que mais a frente será usado para o o treino e teste da máquina programada,

Em particular, consideramos o modo do campo da cavidade, que interage com a transição do átomo de dois níveis, em um estado de Fock, estado coerente, estado de gato e estado de vácuo comprimido [17, 18, 19, 20]. A respeito do estado inicial atômico, consideramos para os valores de $c_g = \{0, 1/\sqrt{2}, 1\}$. Ademais, assumiu-se $\kappa = 20$ e $\Delta x = 0.1\lambda$. Nas figuras a seguir, algumas distribuições de momento para alguns exemplos desses estados são ilustradas.

2.4 Distribuição do Estado de fock

O estado de Fock é um estado quântico caracterizado por um número bem definido de partículas, como fótons no caso de sistemas ópticos. Em uma cavidade óptica, o estado de Fock é representado como $|n\rangle$, onde n indica o número exato de fótons presentes no estado. Esses estados são ortogonais entre si, ou seja, a probabilidade de encontrar um número diferente de fótons é zero. O estado de Fock é útil quando queremos estudar sistemas com um número fixo de partículas, sem flutuações, o que torna a descrição do sistema precisa. Diferentemente dos estados coerentes, que possuem características mais clássicas e variáveis, o estado de Fock não tem superposição de números de fótons e é frequentemente usado para estudar sistemas que não interagem com o ambiente de forma significativa.

Para entender por que o coeficiente c_m para um estado de Fock $|n\rangle$ é representado pelo delta de Kronecker $\delta_{m,n}$, precisamos analisar como funcionam os estados quânticos de um sistema e por que eles são ortogonais. Vamos explorar isso passo a passo.

Os estados de Fock $|n\rangle$ representam estados de um sistema quântico onde há exatamente n partículas ou n quantas de energia, como em um oscilador harmônico quântico (que pode ser visualizado como uma "caixa" onde "vibra" essas quantas de energia). Esses estados são fundamentais porque representam os possíveis resultados "contáveis" de medição no sistema.

Os estados de Fock $|n\rangle$ e $|m\rangle$ são ortogonais entre si quando n é diferente de m . Ser ortogonal significa que, se fizermos a sobreposição (produto interno) entre dois estados

diferentes $|m\rangle$ e $|n\rangle$, ela será zero:

$$\langle m|n\rangle = 0 \quad \text{para} \quad m \neq n$$

E se $m = n$, a sobreposição será 1:

$$\langle n|n\rangle = 1$$

Essa propriedade de ortogonalidade e normalização é representada pelo delta de Kronecker $\delta_{m,n}$, que tem as seguintes características: - $\delta_{m,n} = 1$ quando $m = n$; - $\delta_{m,n} = 0$ quando $m \neq n$.

Imagine que você tenha uma base de estados quânticos e quer representar um estado específico $|n\rangle$ dessa base. Se tentássemos descrever $|n\rangle$ como uma combinação de outros estados de Fock $|m\rangle$, onde m pode ser diferente de n , encontraríamos que o único termo que "sobrevive" nessa combinação é aquele que corresponde ao próprio estado $|n\rangle$.

Portanto, o coeficiente c_m para descrever o estado $|n\rangle$ terá valor 1 apenas quando $m = n$ e será zero para todos os outros $m \neq n$. Isso é exatamente o que o delta de Kronecker $\delta_{m,n}$ representa: - Ele "seleciona" apenas o termo correspondente ao estado específico $|n\rangle$; - Ignora todos os outros termos diferentes de n .

Quando falamos em estados de Fock como uma base ortogonal, isso significa que cada estado é único e independente dos outros. Essa independência implica que, se estamos em um estado $|n\rangle$, nenhum outro estado $|m\rangle$ com $m \neq n$ pode ter qualquer "semelhança" ou "contribuição" para o estado $|n\rangle$. Portanto, se tentamos ver a "parte" de $|m\rangle$ que poderia "existir" em $|n\rangle$, encontramos que essa "parte" é zero para $m \neq n$.

O coeficiente c_m que multiplica cada estado $|m\rangle$ em uma soma só é igual a 1 quando $m = n$. Isso garante que a "presença" de um estado de Fock $|n\rangle$ só existe nele mesmo e não se "espalha" para outros estados. Essa propriedade é representada matematicamente pelo delta de Kronecker $\delta_{m,n}$. Assim, quando dizemos que o coeficiente c_m para o estado de Fock é o delta de Kronecker, estamos expressando que o estado $|n\rangle$ só "responde" quando comparado a ele mesmo, e não a outros estados.

2.5 Estado Coerente

O estado coerente é um estado quântico que minimiza as incertezas de posição e momento de maneira balanceada, tornando-se o estado quântico mais próximo do comportamento clássico. Representado como $|\alpha\rangle$, onde α é um número complexo, esse estado é uma superposição de estados de Fock. Um estado coerente possui flutuações bem definidas, que seguem uma distribuição de Poisson para o número de fótons. Isso o torna ideal para descrever sistemas que apresentam comportamento quase clássico, como lasers, em que o número de fótons flutua em torno de um valor médio. Ao contrário do estado de Fock, que possui um número fixo de partículas, o estado coerente permite flutuações,

mas de forma controlada, mantendo a relação de incerteza de Heisenberg entre posição e momento.

Os estados coerentes são obtidos aplicando o operador de deslocamento $\hat{D}(\alpha)$ no estado de vácuo $|0\rangle$, onde α é um número complexo característico do estado. A expressão para um estado coerente $|\alpha\rangle$ é dada por:

$$|\alpha\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{m=0}^{\infty} \frac{\alpha^m}{\sqrt{m!}} |m\rangle \quad (12)$$

Considerando o modo da cavidade no estado coerente $|\alpha\rangle = e^{\alpha a^\dagger - \alpha^* a} |0\rangle = e^{-\frac{\alpha^2}{2}} \sum_{m=0}^{\infty} \frac{\alpha^m}{m!} |m\rangle$, obtém-se

$$c_m = \frac{e^{-\frac{|\alpha|^2}{2}} \alpha^m}{\sqrt{m!}}, \quad (13)$$

2.6 Estados de Gato Óptico

Os estados gato de óptico é um exemplo paradigmático de superposição quântica, caracterizado pela coexistência de dois estados macroscopicamente distintos. Geralmente representado como $|\alpha\rangle + |-\alpha\rangle$. Ao contrário dos estados de Fock e coerentes, que podem ser descritos por números bem definidos ou amplitudes, o estado gato manifesta uma forte não-clássica.

A superposição de estados coerentes de mesma amplitude, porém de fases diametralmente opostas forma o chamado formam os estados de gato ópticos, em referência ao estado de gato de Schrödinger. Em particular, define-se como estado de gato par e ímpar as superposições simétricas e antissimétricas, respectivamente [21], $|\psi_{\pm}\rangle = N_{\pm}(|\alpha\rangle \pm |-\alpha\rangle)$, onde $N_{\pm} = [2(1 \pm e^{-2|\alpha|^2})]^{-\frac{1}{2}}$ é um fator de normalização. Muito parecido com o estado coerente, para o campo no estado de gato, c_m assume a expressão

$$c_m = N_{\pm} \frac{e^{-\frac{|\alpha|^2}{2}} \alpha^m}{\sqrt{m!}} [1 \pm (-1)^m]. \quad (14)$$

2.7 Distribuição do Estado de Vácuo Comprimido

O estado comprimido é um estado quântico no qual uma das variáveis conjugadas, como posição ou momento, tem incerteza reduzida abaixo dos limites do estado coerente, ao custo de aumentar a incerteza na variável conjugada oposta. Representado por uma operação de compressão no estado coerente, o estado comprimido pode ser visto como uma manipulação dos estados de Fock e coerentes, onde, por exemplo, a incerteza no número de fótons é aumentada enquanto a incerteza em sua fase é reduzida. Ao contrário do estado coerente, que tenta manter um equilíbrio entre as incertezas, o estado comprimido oferece uma vantagem para medições de precisão, mas é mais sensível à decoerência.

O estado de vácuo comprimido é um tipo de estado quântico que resulta de uma

compressão nas flutuações quânticas do campo eletromagnético. Essa compressão pode ser entendida como uma manipulação das incertezas associadas às componentes do campo elétrico e magnético, conforme descrito pelo princípio da incerteza de Heisenberg. Em um estado de vácuo comprimido, a incerteza de uma das variáveis (por exemplo, o campo elétrico) é reduzida, enquanto a incerteza da variável conjugada (o campo magnético) é aumentada, mantendo o produto das incertezas no limite permitido pelo princípio da incerteza.

Consideramos o modo do campo da cavidade em um estado de vácuo comprimido $|\xi\rangle = e^{\frac{1}{2}(\xi^* a^2 - \xi a^{\dagger 2})} |0\rangle$, onde $\xi = r e^{i\theta}$, com r e θ sendo o fator e o ângulo de compressão, respectivamente. Para este caso tem-se:

$$c_m = \frac{[\tanh(r)e^{i\theta}]^{\frac{m}{2}}}{2^{\frac{m}{2}}[m! \cosh(r)]^{\frac{1}{2}}} H_m(0); \quad (15)$$

Onde $H_m(0)$ é o m-ésimo polinômio de Hermite de argumento nulo.

3 Aprendizado de Máquina

O uso de Aprendizado de Máquina (AM) tem se expandido rapidamente em diversas áreas do conhecimento, oferecendo soluções eficientes para problemas complexos de análise de dados. Na física quântica, a aplicação (AM) se tornou uma ferramenta fundamental para lidar com o crescente volume de dados e a complexidade intrínseca de sistemas quânticos. Com Isso, técnicas de aprendizado supervisionado têm sido exploradas para classificar estados quânticos com precisão, superando desafios tradicionais da tomografia quântica [22, 23].

Neste trabalho, a classificação dos estados quânticos do campo foi realizada utilizando um algoritmo de Máquina de Vetores de Suporte (MVS). As MVS são algoritmos amplamente utilizados para tarefas de classificação, sendo particularmente eficazes em situações onde os dados apresentam complexidade ou se encontram em espaços de alta dimensionalidade.

3.1 Máquina de Vetores de Suporte

O princípio fundamental por trás de uma MVS é a busca pelo melhor hiperplano que separa os dados em diferentes classes, maximizando a margem entre os grupos de dados.

Imagine que em um plano estão distribuídos os dados de duas amostras diferentes. O algoritmo de MVS analisa esses dados de entrada e gera o melhor hiperplano para separar essas duas amostras em dois conjunto de dados diferentes, mas é fácil notar que pode existir infinitos hiperplanos que conseguiriam separar essas amostras com sucesso, dessa forma qual seria o ideal?

O algoritmo de SVM vai encontrar os pontos mais próximos do hiperplano e vai maximizar essa distância, assim estabelecendo um limite de decisão onde a separação das duas amostras seja a mais ampla possível, como visto na Figura 2.



Figura 2: Em 2a temos a visualização de dois possíveis hiperplanos que conseguiria classificar as duas amostras, e em 2b temos qual seria a escolha do melhor hiperplano.

Note que, no exemplo mostrado, um hiperplano conseguiria separar adequadamente

as amostras, mas o que é feito caso elas sejam distribuídas de uma forma que isso não seja possível? O MVS introduz um recurso adicional para resolver esse problema, como mostra a Figura 3, onde $z = x^2 + y^2$.



Figura 3: Em 3a temos a visualização dos dados das duas amostras, e em 3b é usado $z = x^2 + y^2$ para facilitar a análise.

Perceba que a ideia principal desse ajuste foi conseguir separar melhor as amostras, além de os valores de z sempre serem positivos, pois é a soma quadrática dos termos x e y , e seria fácil encontrar manualmente um hiperplano linear que consiga separar com sucesso as duas amostras. Entretanto o MVS tem uma ferramenta que consegue encontrar a função z que melhor separa as amostras, transformando um espaço de entrada dimensional baixo num espaço dimensional mais alto, ou seja, transforma o um problema não separável em um problema separável.

A função que realiza esse tipo de tarefa se chama *Kernel*, e é usada principalmente por problemas de separação não linear, além dela existem outras duas funções muito importantes para a realização desse trabalho, a função C e a função *gamma*. Elas são chamadas de hiperparâmetros e servem para ajustar melhor o hiperplano conforme a complexidade dos dados.

3.1.1 Hiperparâmetros

1. **Kernel** define a função de mapeamento dos dados para um espaço dimensional mais elevado, onde se torna mais fácil separá-los linearmente. Utilizamos o *kernel Radial Basis Function* (RBF), que é uma escolha comum em problemas de classificação não linear, já que pode lidar com fronteiras de decisão complexas.

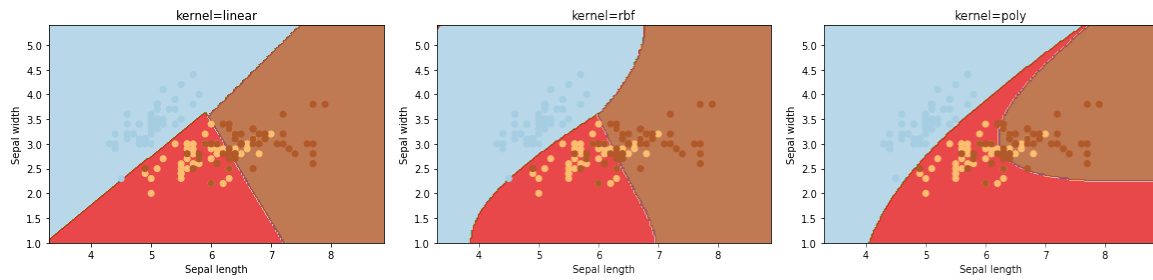


Figura 4: Análise do hiperplano via $kernel = linear$, RBF e $Poly$, respectivamente da esquerda para direita. Imagens tiradas de [2].

2. C controla a penalização por erros de classificação durante o treinamento. Valores menores de C permitem um modelo mais flexível, que se ajusta mais aos dados de treinamento, mas isso pode resultar em *overfitting* – quando o modelo se ajusta demais aos dados de treino, perdendo sua capacidade de generalização para novos dados. Valores maiores de C , por outro lado, favorecem soluções mais simples, com maior capacidade de generalização, ao impor penalidades mais severas para erros no conjunto de treinamento.

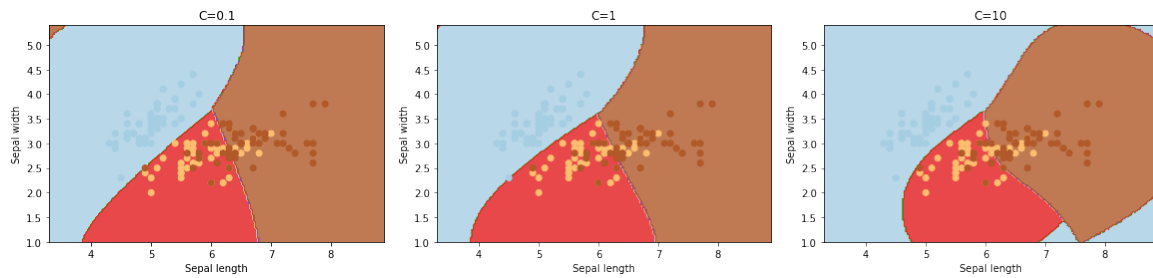


Figura 5: Análise do hiperplano via $C = 0.1, 1, 10$, respectivamente da esquerda para direita. Imagens tiradas de [2].

3. ***Gamma*** define a amplitude da influência de cada exemplo de treinamento no espaço de decisão do MVS. Valores baixos de *gamma* resultam em uma influência mais ampla, fazendo com que os exemplos distantes tenham impacto no modelo, enquanto valores elevados restringem a influência a exemplos de treinamento mais próximos, o que pode melhorar a capacidade de separação de classes em certos casos.

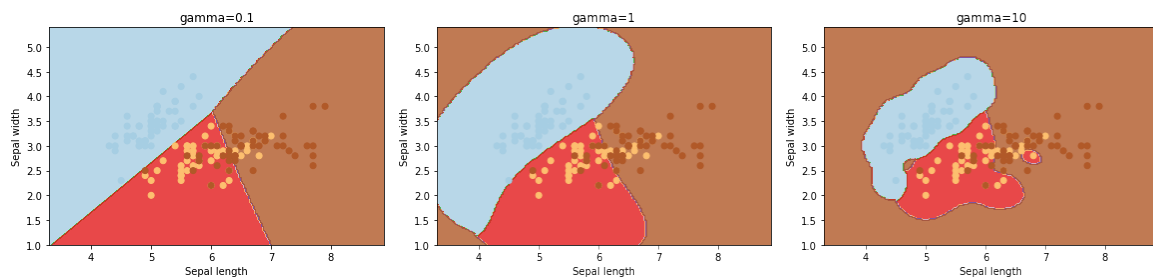


Figura 6: Análise do hiperplano via $gamma = 0.1, 1, 10$, respectivamente da esquerda para direita. Imagens tiradas de [2].

A *scikit-learn* oferece a função *Gridsearchcv* para realizar uma busca exaustiva por hiperparâmetros ideais. Através da validação cruzada, o *Gridsearchcv* testa várias combinações de hiperparâmetros para selecionar aqueles que oferecem o melhor desempenho. O código abaixo demonstra o uso dessa função:

```
from sklearn.model_selection import GridSearchcv
# Definindo a grade de hiperparâmetros
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [0.001, 0.01, 0.1, 1]}
# Criando uma instância de GridSearchcv
grid = GridSearchcv(SVC(kernel='rbf'), param_grid, cv=5)
# Treinando o modelo com GridSearchcv
grid.fit(X_train, y_train)
# Melhores hiperparâmetros encontrados
best_params = grid.best_params_
```

O processo de ajuste de hiperparâmetros garante que o modelo final tenha o melhor desempenho possível, evitando tanto o *underfitting* (modelo simples) quanto o *overfitting* (modelo complexo).

Metodologia e Resultados

4 Visualização da Distribuição do Momento dos Átomos

Wolfram Mathematica é uma plataforma computacional poderosa amplamente utilizada em diversos campos da ciência, engenharia e matemática. Com uma interface amigável e um vasto conjunto de ferramentas, permite a execução de cálculos complexos, manipulação simbólica, e a visualização de dados e gráficos de maneira eficiente. A versatilidade da plataforma a torna uma escolha popular para pesquisadores e acadêmicos que necessitam de uma solução robusta para análise numérica e visualização de resultados [24].

No contexto deste trabalho, o Wolfram Mathematica foi responsável pela visualização da distribuição do momento dos átomos para cada estado quântico estudado.

4.1 Distribuição do Momento dos Átomos para o Estado de Fock

Na Figura 7 são ilustradas as distribuições de momento dos átomos para o campo no estado de vácuo ($M = 0$ – painéis superiores) e no estado de dois fótons ($M = 2$ – painéis inferiores).

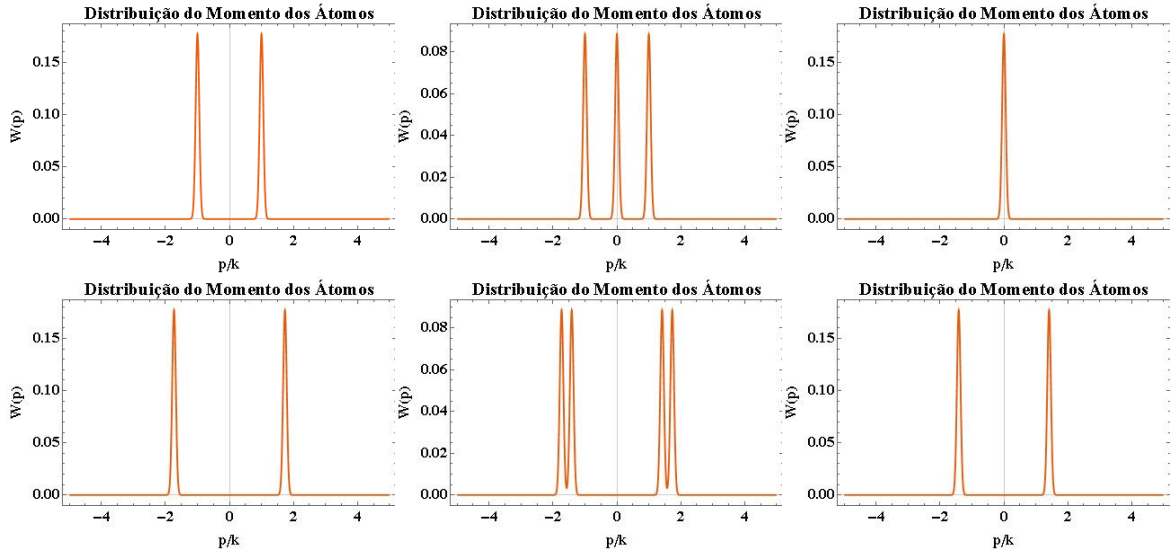


Figura 7: Distribuição do momento atômico, que representa a imagem de distribuição espacial gerada no anteparo distante, considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado $|0\rangle$ (painéis superiores) e para o campo no estado $|2\rangle$ (painéis inferiores).

4.2 Distribuição do Momento dos Átomos para o Estado Coerente

As distribuições de momento atômico para o estado coerente considerando $\alpha = 1$ (painéis superiores) e $\alpha = 3$ (painéis inferiores), para os diferentes valores de c_g são mostradas na Figura 8

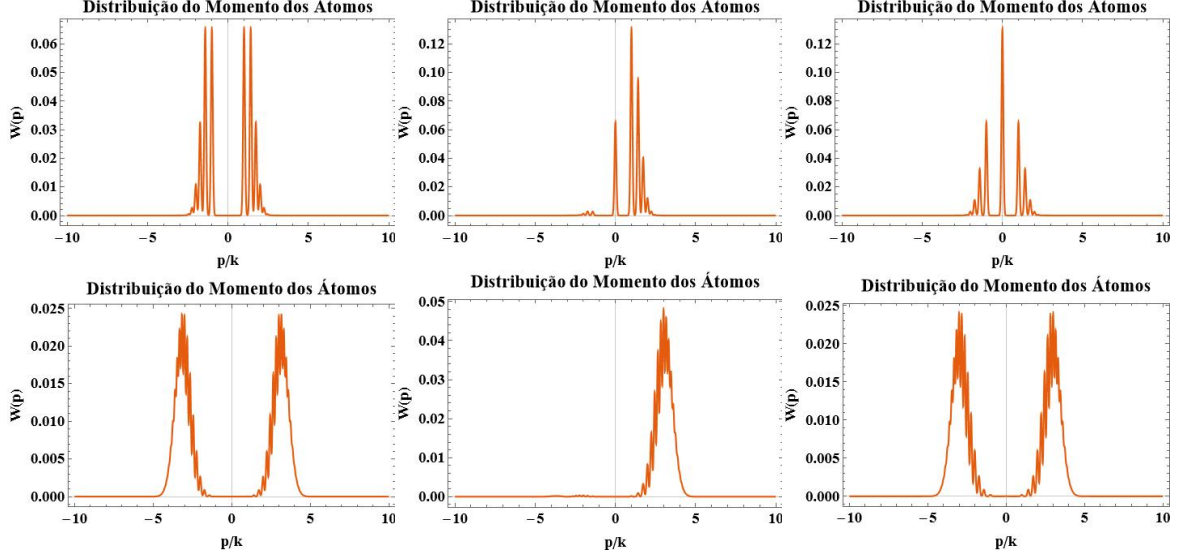


Figura 8: Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado coerente de amplitude $\alpha = 1$ (painéis superiores) e de $\alpha = 3$ (painéis inferiores).

4.3 Distribuição do Momento dos Átomos para o Estado de Gato Óptico

As distribuições de momento atômica considerando o modo do campo da cavidade no estado de gato par (painéis superiores) e ímpar (painéis inferiores) para os diferentes valores de c_g , assumindo $\alpha = 3$ e $\alpha = 5$ são ilustradas nas Figuras 9 e 10, respectivamente.

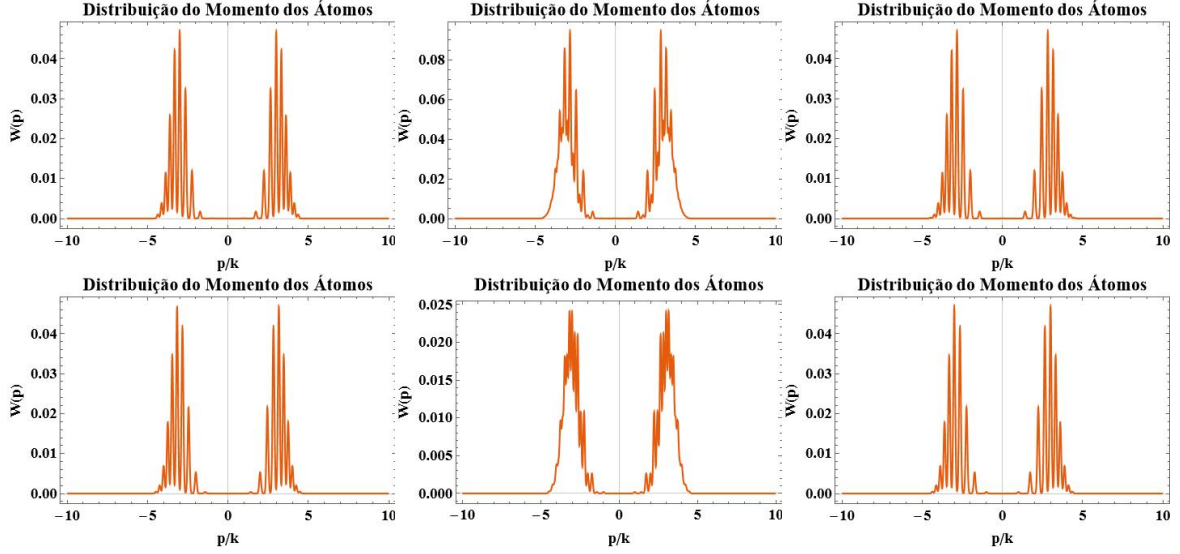


Figura 9: Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado de gato par (painéis superiores) e ímpar (painéis inferiores) assumindo $\alpha = 3$.

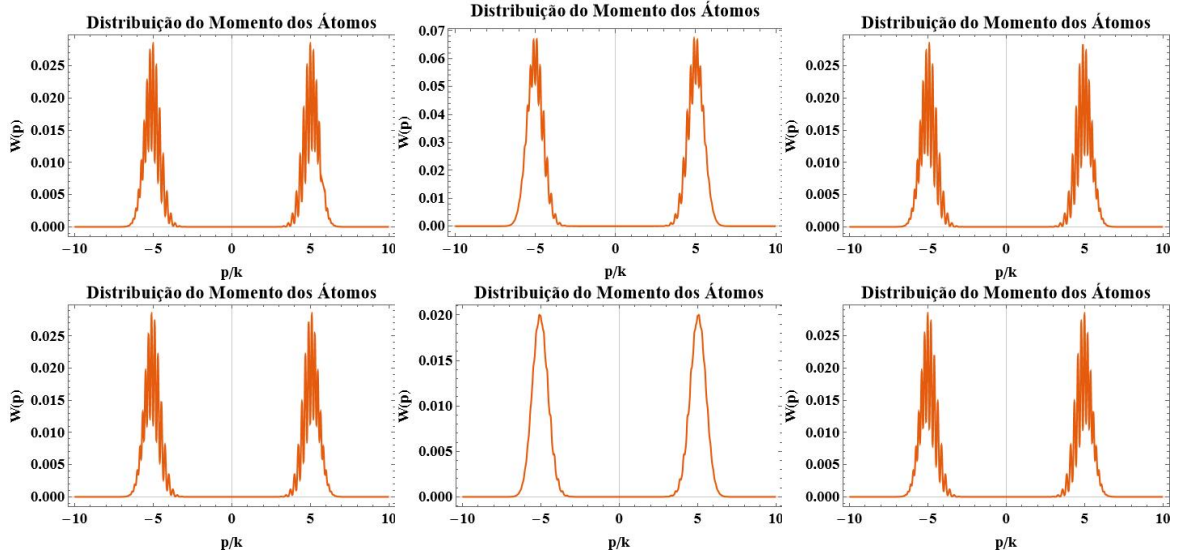


Figura 10: Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo no estado de gato par (painéis superiores) e ímpar (painéis inferiores) assumindo $\alpha = 5$.

4.4 Distribuição do Momento dos Átomos para o Estado de Vácuo Comprimido

Fixando $\theta = 0$ e $r = 1$, a Figura 11 mostra as distribuições de momento atômico para o campo nesse estado considerando distintos valores de c_g .

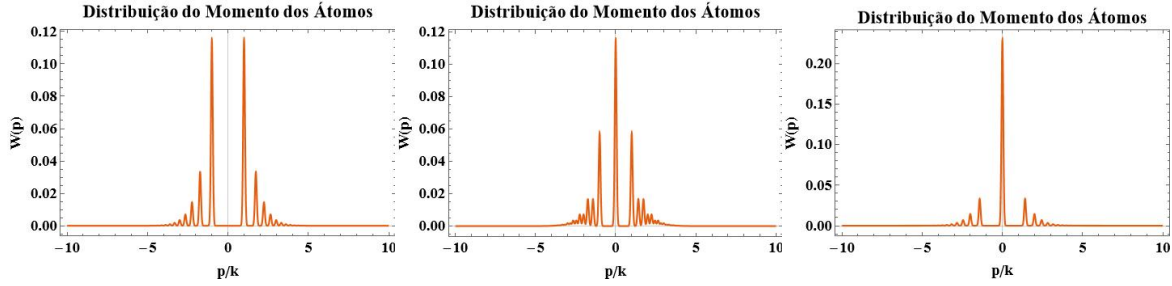


Figura 11: Distribuição do momento atômico considerando $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, para o campo da cavidade no estado de comprimido assumindo $\xi = 1$.

Com esses exemplos, é notável a dependência do padrão de espalhamento atômica com o estado em que se encontra o campo da cavidade. Dessa forma, da estrutura das Eqs. (10) e (11) temos que a largura dos picos está relacionada com o tempo de interação entre átomo e campo, enquanto a altura do pico e sua distância da origem dependem da projeção do estado do campo em $|m\rangle$. Assim, a largura dos picos não seria uma boa característica para alimentar a máquina querendo que ela nos retorne informações do campo, já que está relacionado com o tempo de interação. Por outro lado, o número de picos, suas alturas e suas distâncias com a origens estão diretamente ligadas com a estrutura do estado do campo.

Mas como a máquina foi treinada?

5 Treinamento da Máquina

5.1 Aprendizado Supervisionado

O aprendizado supervisionado é uma técnica em que o algoritmo aprende a partir de um conjunto de dados rotulados. No caso da classificação de estados quânticos, os dados de entrada consistem em padrões de deflexão do momento dos átomos obtidos experimentalmente, enquanto as etiquetas correspondem a diferentes tipos de estados quânticos. O objetivo é treinar o modelo para que, ao receber novos dados (padrões de deflexão não rotulados), ele consiga inferir corretamente o estado quântico correspondente [10, 25].

Com a *scikit-learn*, a implementação da MVS para classificação é bastante simples. O processo envolve:

1. Importar a classe **SVC** da *scikit-learn*.
2. Ajustar os dados de treinamento ao modelo com o método `fit()`.
3. Prever os rótulos para novos dados com o método `predict()`.

```
from sklearn.svm import SVC
# Criando uma instância do classificador SVM
svm = SVC(kernel='rbf', C=1.0, gamma=0.01)
```

```
# Treinando o modelo com os dados de treinamento
svm.fit(X_train, y_train)
# Fazendo previsões com novos dados
predictions = svm.predict(X_test)
```

5.2 Validação Cruzada

Para garantir que os modelos de aprendizado supervisionado, como as SVMs, sejam capazes de generalizar corretamente para novos dados, é essencial utilizar técnicas de validação cruzada. Na validação cruzada *k-fold*, o conjunto de dados é dividido em *k* subconjuntos (ou "*folds*") de tamanhos aproximadamente iguais. O modelo é treinado utilizando *k* − 1 desses subconjuntos e testado no subconjunto restante. Esse processo é repetido *k* vezes, de modo que cada subconjunto é utilizado uma vez como conjunto de teste. A métrica de desempenho do modelo, como acurácia ou erro, é então calculada como a média dos resultados em cada rodada.

A *scikit-learn* oferece a função `cross_val_score()` para realizar validação cruzada de forma eficiente. Por exemplo, para realizar uma validação cruzada com 5 folds em um classificador SVM, podemos utilizar o seguinte código:

```
from sklearn.model_selection import cross_val_score
# Realizando validação cruzada 5-fold
scores = cross_val_score(svm, X_train, y_train, cv=5)
# Calculando a média da acurácia
mean_score = scores.mean()
```

A vantagem da validação cruzada é que ela permite uma melhor utilização dos dados disponíveis, garantindo que o modelo seja avaliado em diferentes subconjuntos de teste, minimizando o risco de *overfitting*. Isso é particularmente importante no contexto de dados quânticos, onde a quantidade de dados pode ser limitada e a complexidade do problema requer uma avaliação cuidadosa do modelo.

Para melhor visualização de como as técnicas de aprendizado supervisionado e de validação cruzada foram aplicadas, veja o Apêndice A

5.3 Classificação e previsão dos estados quânticos

Para classificar os estados quânticos do campo entre os diferentes estados estudados, como mostrado no Apêndice A, utilizamos um conjunto de dados gerados a partir dos padrões de espalhamento atômico via o efeito Stern-Gerlach óptico. Esses padrões, discutidos em detalhes na seção anterior, exibem picos característicos localizados em torno de $\varphi = \pm\sqrt{m}\kappa$ (com $m = 0, 1, \dots$), cujas alturas e distâncias são utilizadas como características (features) para a classificação. Esses picos estão relacionados aos diferentes

estados do campo, como o estado de Fock, estado coerente, e estados de gato Óptico e de vácuo comprimido. O algoritmo MVS foi treinado para distinguir entre essas diferentes configurações de estados com base na análise dessas características.

O processo de pré-processamento dos dados é um passo crucial para a efetividade do MVS. Realizamos a normalização das variáveis, o que é essencial para garantir a consistência e eficiência do algoritmo, já que o desempenho do MVS é sensível à escala das variáveis de entrada. Em seguida, os dados foram divididos em conjuntos de treinamento e teste, sendo o primeiro utilizado para treinar o modelo e o segundo para avaliar sua capacidade de generalização.

Após o treinamento inicial, o modelo MVS alcançou uma acurácia de aproximadamente 75% na tarefa de classificação dos estados quânticos. No entanto, observou-se que a máquina tinha dificuldades em distinguir os estados de gato de Schrödinger, uma vez que seus padrões de espalhamento atômico são bastante semelhantes (ver Figuras 9 e 10). Para melhorar o desempenho, foi realizado um estudo mais aprofundado do comportamento do modelo em relação aos hiperparâmetros C e γ . Verificou-se que, para melhorar a acurácia e evitar o *overfitting*, era necessário ajustar cuidadosamente esses parâmetros.

A melhor performance foi obtida com $C = 100$ e $\gamma = 5$, utilizando o *kernel* RBF. A máquina foi treinada com 60% do total de dados, e os 40% restantes foram reservados para teste. Após essa otimização, a acurácia do modelo aumentou significativamente, atingindo 96%, conforme a matriz de confusão apresentada na Figura 12. Essa matriz exibe a porcentagem de predições corretas (diagonal principal) e a taxa de confusão entre diferentes estados quânticos (elementos fora da diagonal).

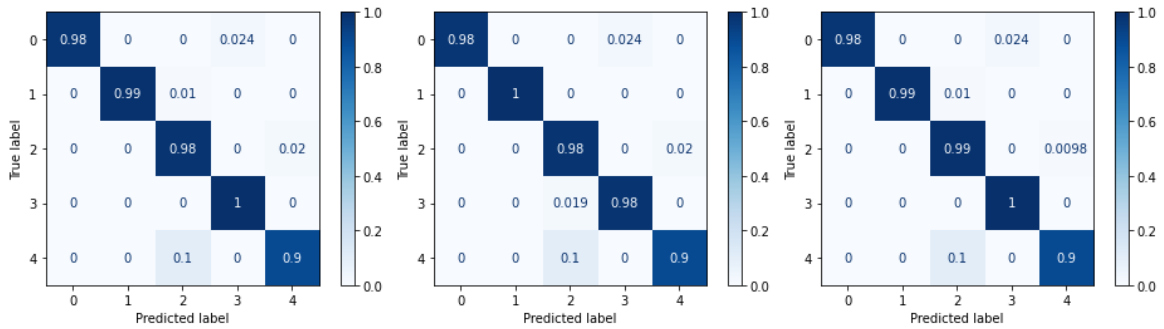


Figura 12: Matriz de confusão considerando $c_g = 0$ com 97% de acurácia, $c_g = \frac{1}{\sqrt{2}}$ com 96% de acurácia e $c_g = 1$ com 98% de acurácia, respectivamente. Os rótulos 0, 1, 2, 3 e 4 correspondem, respectivamente, ao estado de Fock, estado coerente, estado de gato par, estado de gato ímpar e estado de vácuo comprimido.

A matriz de confusão obtida, representada na Figura 12, revela que o algoritmo MVS foi capaz de classificar corretamente os estados em, no mínimo, 96% dos casos, dependendo do valor de c_g , que representa a amplitude do estado atômico inicial. Esse resultado demonstra a eficiência do MVS em lidar com a tarefa de classificação de estados quânticos em um cenário complexo e com múltiplas classes.

Para melhor compreensão de como se analisa uma matriz confusão, veja o Apêndice C

5.4 Regressão por Vetores de Suporte (SVR)

O método de Classificação por Vetores de Suporte (SVC) pode ser estendido para resolver problemas de regressão. Esse método é chamado Regressão por Vetores de Suporte (SVR).

O modelo produzido pela classificação por vetores de suporte (como descrito acima) depende apenas de um subconjunto dos dados de treinamento, pois a função de custo para construir o modelo não considera os pontos de treinamento que estão além da margem. De forma análoga, o modelo produzido pela Regressão por Vetores de Suporte depende apenas de um subconjunto dos dados de treinamento, pois a função de custo ignora amostras cuja previsão esteja próxima de seu valor-alvo.

Existem três implementações diferentes de Regressão por Vetores de Suporte: SVR, NuSVR e LinearSVR. O LinearSVR oferece uma implementação mais rápida do que o SVR, mas considera apenas o kernel linear, enquanto o NuSVR implementa uma formulação ligeiramente diferente do SVR e do LinearSVR. Devido à sua implementação em *liblinear*, o LinearSVR também regulariza o intercepto, se considerado. Esse efeito pode, no entanto, ser reduzido ajustando cuidadosamente o parâmetro `intercept_scaling`, o que permite que o termo de intercepto tenha um comportamento de regularização diferente dos demais recursos. Portanto, os resultados da classificação e o score podem diferir dos outros dois classificadores.

Assim como nas classes de classificação, o método `fit` usará como argumento os vetores X e y , sendo que, neste caso, espera-se que y contenha valores de ponto flutuante em vez de valores inteiros.

```
from sklearn import svm
# Definindo o conjunto de dados de entrada (X) com duas amostras,
# cada uma com duas variáveis.
X = [[0, 0], [2, 2]]
# Definindo os valores-alvo (y) correspondentes a cada amostra em X.
# Esses valores são os que o modelo de regressão tentará prever.
regr = svm.SVR()
y = [0.5, 2.5]
# Criando um objeto do modelo de Regressão por Vetores de Suporte (SVR)
regr = svm.SVR()
# Treina o modelo SVR nos dados de entrada (X) e valores-alvo (y).
regr.fit(X, y)
# Usa o modelo treinado para prever o valor-alvo de uma nova amostra
# de entrada, [1, 1].
```

```
regr.predict([[1, 1]])
```

Com isso foi aplicado um Regressor de Vetores de Suporte (RVS) para prever a estatística do campo intracavidade, onde esse algoritmo busca encontrar uma função ideal para prever a variável dependente a partir das variáveis independentes. Essa função é determinada otimizando um modelo matemático que minimiza os erros de predição. Os pontos-chave no espaço de entrada, chamados de vetores de suporte, são cruciais para definir a função de regressão. A avaliação da qualidade do modelo é realizada com dados de teste, e métricas como o coeficiente de determinação (R^2) indicam a precisão da predição. Para o nosso caso, treinamos a máquina para prever o número médio de fótons do estado do campo considerando os estados e condições do efeito de Stern-Gerlach trabalhados. O código para a realização da tarefa de regressão pode ser analisada no (Apêndice B).

Para ser possível a construção do algoritmo e testá-lo, são necessárias as expressões para o número médio de fótons ($\bar{n} = [a^\dagger a]$).

1. Fock: $|N\rangle \rightarrow \bar{n} = N$
2. Coerente: $|\alpha\rangle \rightarrow \bar{n} = |\alpha|^2$
3. Gato par: $|\psi_+\rangle \rightarrow \bar{n} = |\alpha|^2 \tanh |\alpha|^2$
4. Gato ímpar: $|\psi_-\rangle \rightarrow \bar{n} = |\alpha|^2 \coth |\alpha|^2$
5. Comprimido: $|\xi\rangle \rightarrow \bar{n} = \sinh^2(r)$

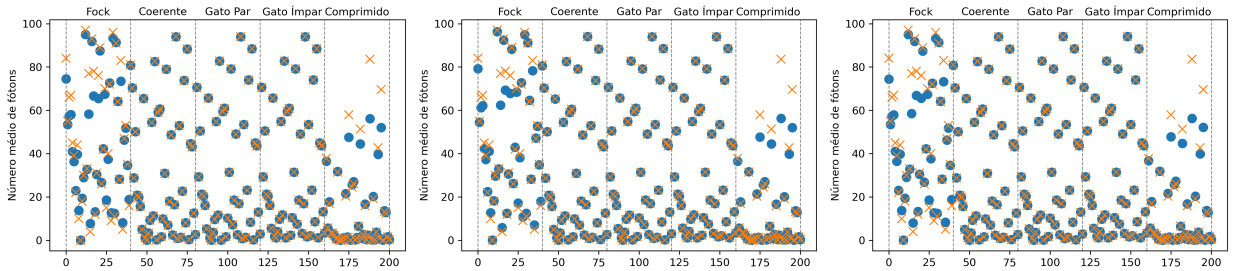


Figura 13: Regressão para o número médio de fótons, da esquerda para a direita $c_g = 0$, $c_g = \frac{1}{\sqrt{2}}$ e $c_g = 1$, respectivamente, todas com 99% de acurácia.

A Figuras 13 compara os valores preditos (círculos azuis) com os valores reais ('x' laranja) para $\bar{n}$, respectivamente. Os intervalos no eixo das abscissas correspondem às amostras de dados de teste relativos aos estados quânticos analisados, de 0 a 40 (Fock), 40 a 80 (coerente), 80 a 120 (gato par), 120 a 160 (gato ímpar) e 160 a 200 (vácuo comprimido). No algoritmo da máquina, foram empregados os parâmetros de regularização $C = 5000$, $\gamma = 2$ e $\text{Kernel} = \text{rbf}$, e é observado que o algoritmo retornou uma acurácia da ordem de 99% para a tarefa de regressão do número médio de fótons.

Conclusão

Em suma, o uso de MVS se mostrou uma abordagem eficaz para a classificação e previsão de estados quânticos do campo em um contexto experimental envolvendo o efeito Stern-Gerlach óptico.

Os resultados obtidos, refletidos nas matrizes de confusão, demonstraram um desempenho notável na acurácia nas tarefas de classificação e regressão, atingindo até 99% de acurácia em algumas configurações. Esta alta taxa de acerto não apenas valida a eficácia das abordagens de aprendizado de máquina na análise de estados quânticos, mas também destaca a importância das técnicas utilizadas na extração de características relevantes, permitindo que os algoritmos aprendam a distinguir entre os diferentes estados quânticos de maneira precisa e prever características dos mesmos.

Além disso, a análise da matriz de confusão proporcionou *insights* valiosos sobre a eficácia do modelo, evidenciando quais estados foram classificados corretamente e onde ocorreram os erros. Esses dados são cruciais para aprimorar os métodos de classificação e entender as limitações dos modelos. A capacidade de discernir estados quânticos complexos é fundamental para diversas aplicações em óptica quântica, computação quântica e informação quântica.

As implicações deste trabalho são significativas, não apenas para a pesquisa em física quântica, mas também para o desenvolvimento de novas tecnologias. A integração de aprendizado de máquina em sistemas quânticos abre novas perspectivas para a automação de experimentos e para a análise de dados em larga escala. Potencializando a pesquisa e a inovação nessas áreas.

Referências

- 1 FREYBERGER, M.; HERKOMMER, A. M. Probing a quantum state via atomic deflection. *Phys. Rev. Lett.*, American Physical Society, v. 72, p. 1952–1955, Mar 1994. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.72.1952>>.
- 2 PEDREGOSA, F. et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <<http://jmlr.org/papers/v12/pedregosa11a.html>>.
- 3 NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information*. [S.l.]: Cambridge University Press, 2000.
- 4 ZHANG, Y.; NI, Q. Recent advances in quantum machine learning. *Quantum Engineering*, v. 2, n. 1, p. e34, 2020. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/que2.34>>.
- 5 SAKURAI, J. J.; NAPOLITANO, J. *Mecânica quântica moderna*. [S.l.]: Bookman, 2013.
- 6 SLEATOR, T. et al. Experimental demonstration of the optical stern-gerlach effect. *Phys. Rev. Lett.*, American Physical Society, v. 68, p. 1996–1999, Mar 1992. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.68.1996>>.
- 7 MÁXIMO, C. E. et al. Quantum atomic lithography via cross-cavity optical stern-gerlach setup. *J. Opt. Soc. Am. B*, The Optical Society, v. 31, n. 10, p. 2480, set. 2014. Disponível em: <<https://doi.org/10.1364/josab.31.002480>>.
- 8 JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, American Association for the Advancement of Science, v. 349, n. 6245, p. 255–260, 2015. Disponível em: <<https://science.sciencemag.org/content/349/6245/255>>.
- 9 CARLEO, G. et al. Machine learning and the physical sciences. *Rev. Mod. Phys.*, American Physical Society, v. 91, p. 045002, Dec 2019. Disponível em: <<https://link.aps.org/doi/10.1103/RevModPhys.91.045002>>.
- 10 DUNJKO, V.; BRIEGEL, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, IOP Publishing, v. 81, n. 7, p. 074001, jun 2018. Disponível em: <<https://doi.org/10.1088/2F1361-6633/81/7/074001>>.
- 11 LI, Z. et al. Experimental realization of a quantum support vector machine. *Phys. Rev. Lett.*, American Physical Society, v. 114, p. 140504, Apr 2015. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.114.140504>>.
- 12 SUPPORT vector machines on the D-Wave quantum annealer. *Computer Physics Communications*, v. 248, p. 107006, 2020. ISSN 0010-4655. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S001046551930342X>>.
- 13 MEYSTRE, P.; SCHUMACHER, E.; STENHOLM, S. Atomic beam deflection in a quantum field. *Optics Communications*, Elsevier BV, v. 73, n. 6, p. 443–447, nov. 1989. Disponível em: <[https://doi.org/10.1016/0030-4018\(89\)90458-6](https://doi.org/10.1016/0030-4018(89)90458-6)>.

- 14 HERKOMMER, A. M.; AKULIN, V. M.; SCHLEICH, W. P. Quantum demolition measurement of photon statistics by atomic beam deflection. *Phys. Rev. Lett.*, American Physical Society, v. 69, p. 3298–3301, Dec 1992. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.69.3298>>.
- 15 HOLLAND, M. J.; WALLS, D. F.; ZOLLER, P. Quantum nondemolition measurements of photon number by atomic beam deflection. *Phys. Rev. Lett.*, American Physical Society, v. 67, p. 1716–1719, Sep 1991. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.67.1716>>.
- 16 MÁXIMO, C. E. et al. Entanglement detection via atomic deflection. *J. Opt. Soc. Am. B*, The Optical Society, v. 34, n. 12, p. 2452, nov. 2017. Disponível em: <<https://doi.org/10.1364/josab.34.002452>>.
- 17 STECK, D. A. *Quantum and Atom Optics*. [s.n.], 2023. Disponível em: <<https://atomoptics.uoregon.edu/~dsteck/teaching/quantum-optics/>>.
- 18 LAMBROPOULOS, D. P. P. *Fundamentals of quantum optics and quantum information*. 10. ed. [S.l.]: Springer, 2007.
- 19 SCULLY, M. O.; ZUBAIRY, M. S. *Quantum Optics*. [S.l.]: Cambridge University Press, 1997.
- 20 DODONOV, V.; MALKIN, I.; MAN'KO, V. Even and odd coherent states and excitations of a singular oscillator. *Physica*, Elsevier BV, v. 72, n. 3, p. 597–615, mar. 1974. Disponível em: <[https://doi.org/10.1016/0031-8914\(74\)90215-8](https://doi.org/10.1016/0031-8914(74)90215-8)>.
- 21 DODONOV, V. V.; MAN'KO, V. I. *Theory of nonclassical states of light*. [S.l.]: CRC Press, 2003.
- 22 TORLAI, G. et al. Neural-network quantum state tomography. *Nature Physics*, v. 14, n. 5, p. 447–450, May 2018. ISSN 1745-2481. Disponível em: <<https://doi.org/10.1038/s41567-018-0048-5>>.
- 23 CARLEO, G.; TROYER, M. Solving the quantum many-body problem with artificial neural networks. *Science*, American Association for the Advancement of Science, v. 355, n. 6325, p. 602–606, 2017. ISSN 0036-8075. Disponível em: <<https://science.sciencemag.org/content/355/6325/602>>.
- 24 RESEARCH, W. *Mathematica: The World's Most Powerful Computational Tool*. 2021. Disponível em: <<https://www.wolfram.com/mathematica/>>.
- 25 MEHTA, P. et al. A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, v. 810, p. 1 – 124, 2019. ISSN 0370-1573.

A Algoritmo de classificação

```

import numpy as np
import matplotlib.pyplot as plt
import math
import datetime

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn import svm, preprocessing
from sklearn.model_selection import cross_val_score
from sklearn.metrics import ConfusionMatrixDisplay

startTime = datetime.datetime.now()

#Distribuicao de momento
def Wpp(p,cm,cg,ce,kappa,dx,Nmax):
    W = 4*gau2p(p,dx,0,kappa)*np.abs(cg*cm[0])**2
    for m in range(1,Nmax):
        (W += 2*gau2m(p,dx,m,kappa)*np.abs(cg*cm[m]+ce*cm[m-1])**2
         + 2*gau2p(p,dx,m,kappa)*np.abs(cg*cm[m]-ce*cm[m-1])**2)
    return W

(Calcula a distribuição de momento W com base em parâmetros
como momento (p),coeficientes (cm), probabilidades do átomo
entrar na cavidade nos estados fundamental (cg), e excitado (ce),
largura da fenda (dx), e número máximo de fótons (Nmax).)

#Gaussianas
def gau2p(p,dx,m,kappa):
    return ((0.25*np.pi*dx**2)**(0.5) *
            np.exp(-(2*np.pi*dx*kappa*(p+np.sqrt(m))**2)))

def gau2m(p,dx,m,kappa):
    return ((0.25*np.pi*dx**2)**(0.5) *
            np.exp(-(2*np.pi*dx*kappa*(p-np.sqrt(m))**2)))

(Funções auxiliares para calcular as gaussianas.)

kappa = 20.0
cg = 1/np.sqrt(2)
ce = np.sqrt(1-cg**2)
dx = 0.1

```

```

Nmax = 151

#distancias dos picos em relacao a origem
pM = 144
plist = np.linspace(-pM,pM,2*pM+1)
plist = np.sign(plist)*np.sqrt(np.abs(plist))

#Xc = []

X = [] #features
y = [] #labels

#Fock
for N in range(0,101):
    cmFock = [0.]*Nmax
    cmFock[N] = 1.

    XF = []
    for p in plist:
        XF.append(Wpp(p,cmFock,cg,ce,kappa,dx,Nmax))

    X.append(XF)
    y.append(0)

#Coerente
for alpha in np.linspace(np.sqrt(0.01),np.sqrt(51),251):
    # alpha = np.sqrt(N+1.)
    cmalpha = [0.]*Nmax
    for m in range(0,Nmax):
        cmalpha[m] = (np.exp(-0.5 * np.abs(alpha)**2) *
            alpha**m / (math.factorial(m)**0.5))

    XC = []
    for p in plist:
        XC.append(Wpp(p,cmalpha,cg,ce,kappa,dx,Nmax))

    X.append(XC)

```

```

y.append(1)

#Gato par
# alpha = np.sqrt(N+1.)
cmp = [0.]*Nmax
NP = (2*(1 + np.exp(-2*np.abs(alpha)**2)))*(-0.5)
for m in range(0,Nmax):
    cmp[m] = (NP * (np.exp(-0.5*np.abs(alpha)**2)*alpha**m /
        (math.factorial(m)**0.5 ) * (1 + (-1)**m))

XGP = []
for p in plist:
    XGP.append(Wpp(p,cmp,cg,ce,kappa,dx,Nmax))

X.append(XGP)
y.append(2)

#Gato impar
# alpha = np.sqrt(N+1.)
cmm = [0.]*Nmax
NM = (2*(1 - np.exp(-2*np.abs(alpha)**2)))*(-0.5)
for m in range(0,Nmax):
    cmm[m] = (NM * (np.exp(-0.5*np.abs(alpha)**2)*alpha**m /
        (math.factorial(m)**0.5 ) * (1 - (-1)**m))

XGI = []
for p in plist:
    XGI.append(Wpp(p,cmm,cg,ce,kappa,dx,Nmax))

X.append(XGI)
y.append(3)

#Comprimido
for r in np.linspace(0.1,3.0,201):
    theta = 0.0
    cms = [0.]*Nmax
    for m in range(0,int(Nmax/2)+1):
        cms[2*m] = ((np.cosh(r))**(-0.5) * (-0.5 * np.tanh(r))**m *

```

```

    (math.factorial(2*m)/math.factorial(m)**2 )**0.5)

XSQ = []
for p in plist:
    XSQ.append(Wpp(p,cms,cg,ce,kappa,dx,Nmax))

X.append(XSQ)
y.append(4)

```

(Geração de estados quânticos para Fock, coerente, gato par, gato ímpar, e comprimido. Cada estado é caracterizado por uma lista de coeficientes (cmFock, cmalpha, cmp, cmm, cms) e é representado em termos de sua distribuição de momento (Wpp). X é a Lista de features, onde cada feature é uma distribuição de momento. y é a Lista de rótulos, onde cada rótulo representa a categoria do estado quântico (0 para Fock, 1 para coerente, 2 para gato par, 3 para gato ímpar, 4 para comprimido).

```

X_normalized = preprocessing.normalize(X, norm='l2')

```

(Normalização das features)

```

# Divisão dos dados em treino e teste
X_train, X_test, y_train, y_test = \
train_test_split(X_normalized, y, test_size=0.4, random_state=0)

```

(Divisão dos dados normalizados em conjuntos de treino e teste)

```

# Ajuste de parâmetros do SVC usando GridSearchCV
parameters = {'C': [0.1, 1, 10, 100],
'gamma': [0.001, 0.01, 0.1, 5],
'kernel': ['rbf']}
grid_search = GridSearchCV(svm.SVC(), parameters, cv=5)
grid_search.fit(X_train, y_train)

best_params = grid_search.best_params_
print(f"Melhores parâmetros: {best_params}")

```

(Utilização do GridSearchCV para encontrar os melhores parâmetros do SVM (Support Vector Machine) através da busca em uma grade de valores para C, gamma, e o tipo de kernel.)

```

# Treinamento do modelo com os melhores parâmetros
clf = svm.SVC(**best_params).fit(X_train, y_train)

```

```
# Avaliação do modelo
score = clf.score(X_test, y_test)
print(f"Score: {score}")

# Validacao cruzada
scores = cross_val_score(clf, X_test, y_test, cv=5)
print(scores)
print("%.2f accuracy with a standard deviation of %.2f" \
      % (scores.mean(), scores.std()))

# Matriz de Confusão
disp = ConfusionMatrixDisplay.from_estimator(
    clf,
    X_test,
    y_test,
    normalize='true',
    cmap=plt.cm.Blues,
)
plt.show()

endTime = datetime.datetime.now()
print(f"Total runtime: {endTime - startTime}")
```

B Algoritmo de Regressão

```
import numpy as np
import math
import datetime
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.svm import SVR

startTime = datetime.datetime.now()

# Função de teste Wpp
def Wpp(p, cm, cg, ce, kappa, dx, Nmax):
    W = 4 * gau2p(p, dx, 0, kappa) * np.abs(cg * cm[0])**2
    for m in range(1, Nmax):
        W += 2 * gau2m(p, dx, m, kappa) * np.abs(cg * cm[m] + \
            ce * cm[m-1])**2 + 2 * gau2p(p, dx, m, kappa) * \
            np.abs(cg * cm[m] - ce * cm[m-1])**2
    return W

# Funções Gaussianas
def gau2p(p, dx, m, kappa):
    return ((0.25 * np.pi * dx**2)**0.5 * \
        np.exp(-(2 * np.pi * dx * kappa * (p + np.sqrt(m))**2)))

def gau2m(p, dx, m, kappa):
    return (0.25 * np.pi * dx**2)**0.5 * \
        np.exp(-(2 * np.pi * dx * kappa * (p - np.sqrt(m))**2)))

# Parâmetros
kappa = 20.0
cg = 1.0
ce = np.sqrt(1 - cg**2)
dx = 0.1
Nmax = 151

# Distâncias dos picos em relação à origem
```

```

pM = 144
plist = np.linspace(-pM, pM, 2*pM+1)
plist = np.sign(plist) * np.sqrt(np.abs(plist))

# Fock
X_Fock = []
y_Fock = []

for N in range(0, 101):
    cmFock = [0.] * Nmax
    cmFock[N] = 1.

    XF = [Wpp(p, cmFock, cg, ce, kappa, dx, Nmax) for p in plist]

    X_Fock.append(XF)
    y_Fock.append(N)

# conjuntos de treino e teste (60% treino, 40% teste)
X_train_Fock, X_test_Fock, y_train_Fock, y_test_Fock = \
train_test_split(X_Fock, y_Fock, test_size=0.4, random_state=42)

# Coerente
X_Coerente = []
y_Coerente = []

for alpha in np.linspace(0, 10, 100):
    cmalpha = [0.]*Nmax
    for m in range(0,Nmax):
        cmalpha[m] = np.exp(-0.5 * np.abs(alpha)**2) * alpha**m \
        / (math.factorial(m))**0.5

    XC = [Wpp(p, cmalpha, cg, ce, kappa, dx, Nmax) for p in plist]

    X_Coerente.append(XC)
    y_Coerente.append(alpha**2)

# conjuntos de treino e teste (60% treino, 40% teste)
X_train_Coerente, X_test_Coerente, y_train_Coerente, y_test_Coerente = \
train_test_split(X_Coerente, y_Coerente, test_size=0.4, random_state=42)

```

```

# Gato Par
X_GatoPar = []
y_GatoPar = []

for alpha in np.linspace(np.sqrt(0.01), 10, 100):
    cmp = [0.] * Nmax
    NP = (2 * (1 + np.exp(-2 * np.abs(alpha)**2)))*(-0.5)
    for m in range(0, Nmax):
        cmp[m] = NP * (np.exp(-0.5 * np.abs(alpha)**2) * alpha**m \
            / (math.factorial(m)**0.5) * (1 + (-1)**m)

    XGP = [Wpp(p, cmp, cg, ce, kappa, dx, Nmax) for p in plist]

    X_GatoPar.append(XGP)
    y_GatoPar.append(np.abs(alpha)**2 * np.tanh(np.abs(alpha)**2))

# conjuntos de treino e teste (60% treino, 40% teste)
X_train_GatoPar, X_test_GatoPar, y_train_GatoPar, y_test_GatoPar = \
    train_test_split(X_GatoPar, y_GatoPar, test_size=0.4, random_state=42)

# Gato Ímpar
X_GatoImpar = []
y_GatoImpar = []

for alpha in np.linspace(np.sqrt(0.01), 10, 100):
    cmm = [0.] * Nmax
    NM = (2 * (1 - np.exp(-2 * np.abs(alpha)**2)))*(-0.5)
    for m in range(0, Nmax):
        cmm[m] = NM * (np.exp(-0.5 * np.abs(alpha)**2) * alpha**m \
            / (math.factorial(m)**0.5) * (1 - (-1)**m)

    XGI = [Wpp(p, cmm, cg, ce, kappa, dx, Nmax) for p in plist]

    X_GatoImpar.append(XGI)
    y_GatoImpar.append((np.abs(alpha))**2 * (1/(np.tanh(alpha))**2))

# conjuntos de treino e teste (60% treino, 40% teste)
X_train_GatoImpar, X_test_GatoImpar, y_train_GatoImpar, y_test_GatoImpar = \
    train_test_split(X_GatoImpar, y_GatoImpar, test_size=0.4, random_state=42)

```

```

# Vácuo Comprimido
X_VacuComprimido = []
y_VacuComprimido = []

for r in np.linspace(0, 3, 100):
    theta = 0.0
    cms = [0.] * Nmax
    for m in range(0, int(Nmax/2)+1):
        cms[2*m] = (np.cosh(r))**(-0.5) * (-0.5 * np.tanh(r))**m * \
            (math.factorial(2*m)/math.factorial(m)**2)**0.5

    XSQ = [Wpp(p, cms, cg, ce, kappa, dx, Nmax) for p in plist]

    X_VacuComprimido.append(XSQ)
    y_VacuComprimido.append(np.sinh(r)**2)
(Ate é essa parte do código é seguida a mesma lógica do Anexo A)

# conjuntos de treino e teste (60% treino, 40% teste)
X_train_VacuComprimido, X_test_VacuComprimido, y_train_VacuComprimido, \
y_test_VacuComprimido = train_test_split\
(X_VacuComprimido, y_VacuComprimido, test_size=0.4, random_state=42)

# Combine todos os estados
X_train_combined = np.concatenate([X_train_Fock, X_train_Coerente, \
X_train_GatoPar, X_train_GatoImpar, X_train_VacuComprimido])
y_train_combined = np.concatenate([y_train_Fock, y_train_Coerente, \
y_train_GatoPar, y_train_GatoImpar, y_train_VacuComprimido])

X_test_combined = np.concatenate([X_test_Fock, X_test_Coerente, \
X_test_GatoPar, X_test_GatoImpar, X_test_VacuComprimido])
y_test_combined = np.concatenate([y_test_Fock, y_test_Coerente, \
y_test_GatoPar, y_test_GatoImpar, y_test_VacuComprimido])
(Combinação dos conjuntos de treino e teste para todos os estados
quânticos. Isso cria um conjunto de dados mais amplo, representando
os diferentes tipos de estados quânticos.)

model_combined = SVR(kernel='rbf', C=5000, gamma=2)
model_combined.fit(X_train_combined, y_train_combined)

```

```
y_pred_combined = model_combined.predict(X_test_combined)

r2_combined = r2_score(y_test_combined, y_pred_combined)
print(f"Coeficiente de Determinação (R2) - Conjunto Total: \
{r2_combined}")
(Cálculo do coeficiente de determinação
para avaliar o desempenho do modelo.)

fig, ax = plt.subplots(1, 1, dpi=300, figsize=(6, 4))
ax.plot(y_pred_combined, marker='o', \
linestyle='none', markersize=8)
ax.plot(y_test_combined, marker='x', \
linestyle='none', markersize=8)

points = [0, 40, 80, 120, 160, 200]
labels = \
['Fock', 'Coerente', 'Gato Par', 'Gato Ímpar', 'Comprimido']

for i in range(len(points) - 1):
    interval_label = f"{labels[i]}"
    interval_middle = (points[i] + points[i + 1]) / 2
    ax.axvline(x=points[i], \
color='gray', linestyle='--', linewidth=0.8)
    ax.text(interval_middle, \
ax.get_ylim()[1] * 1.02, interval_label, \
rotation=0, color='black', ha='center')
    plt.ylabel('Número médio de fótons')

ax.axvline(x=points[-1], color='gray', linestyle='--', linewidth=0.8)

plt.show()

endTime = datetime.datetime.now()
print(f"Total runtime: {endTime - startTime}")
```

C Análise da Matriz Confusão

Para analisar a acurácia com que a máquina conseguiu diferenciar todos os estados quânticos trabalhados é utilizada uma ferramenta computacional chamada Matriz Confusão vista na Figura.

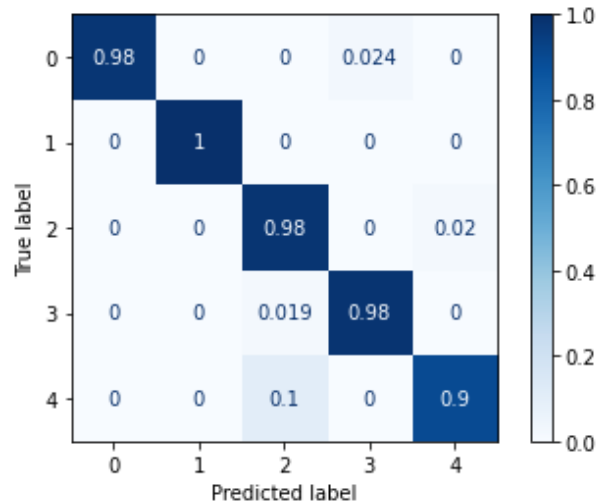


Figura 14: Exemplo de Matriz Confusão

- Elemento M_{ij} : o valor no elemento M_{ij} da matriz representa o número de vezes que o estado i foi classificado como o estado j pelo modelo. Por exemplo, M_{02} indicaria a quantidade de vezes que o estado de Fock (rótulo 0) foi erroneamente classificado como estado de gato par (rótulo 2).
- Diagonal principal: os elementos na diagonal principal M_{ii} representam as classificações corretas, isto é, o número de vezes que o modelo previu corretamente o estado i como i . A soma desses valores reflete a quantidade total de classificações corretas.
- elementos fora da diagonal: indicam as classificações incorretas. Esses valores são úteis para identificar padrões de confusão específicos entre certos estados.