

偏光を用いた2量子もつれの量子状態トモグラフィによる評価

62200139 青木 陽

March 17, 2025

Contents

1	原理	2
2	実験方法	3
3	実験結果	4
4	考察	5
A	ソースコード	6

Chapter 1

原理

Chapter 2

実験方法

Chapter 3

実験結果

Chapter 4

考察

付録 A

ソースコード

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import mpl_toolkits
4 from matplotlib import cm
5 from matplotlib.colors import Normalize
6
7 class HVDR:
8     def __init__(self, l):
9         self.l = l
10    def __mul__(self, other):
11        ret = list()
12        for x in self.l:
13            tmp = list()
14            for y in other.l:
15                tmp.append(x * y)
16            ret.append(tmp)
17        return ret
18
19 def calc_sigma_pow():
20     sigma_0 = HVDR([1, 1, 0, 0])
21     sigma_1 = HVDR([-1, -1, 2, 0])
22     sigma_2 = HVDR([-1, -1, 0, 2])
23     sigma_3 = HVDR([1, -1, 0, 0])
24     sigma = [sigma_0, sigma_1, sigma_2, sigma_3]
25     sigma_pow = [[None for _ in range(4)] for _ in range(4)]
26     for i in range(4):
27         for j in range(4):
28             sigma_pow[i][j] = sigma[i] * sigma[j]
29     return sigma_pow
30
31 def load_data(path):
32     polar_dict = {'H': 0, 'V': 1, 'D': 2, 'R': 3}
33     raw_data = np.loadtxt(path, dtype = "str", delimiter = ',')
34     data = [[None for _ in range(4)] for _ in range(4)]
35     for i in range(1, len(raw_data)):
36         data[polar_dict[raw_data[i][0]]][polar_dict[raw_data[i][1]]] = int(raw_data[i][2])
37     return data
38
39 def calc_uj(sigma_pow, data):
40     u = [[0 for _ in range(4)] for _ in range(4)]
41     for i1 in range(4):
42         for j1 in range(4):
43             for i2 in range(4):
44                 for j2 in range(4):
```

```

43         u[i1][j1] += sigma_pow[i1][j1][i2][j2] * data[i2][j2]
44     return u
45 def calc_sigma_matrix_tensor():
46     sigma_num_0 = np.array([[1, 0], [0, 1]])
47     sigma_num_1 = np.array([[0, 1], [1, 0]])
48     sigma_num_2 = np.array([[0, -1j], [1j, 0]])
49     sigma_num_3 = np.array([[1, 0], [0, -1]])
50     sigma_num = [sigma_num_0, sigma_num_1, sigma_num_2, sigma_num_3]
51     sigma_num_pow = [[None for _ in range(4)] for _ in range(4)]
52     for i in range(4):
53         for j in range(4):
54             sigma_num_pow[i][j] = np.kron(sigma_num[i], sigma_num[j])
55     return sigma_num_pow
56 def estimate_rho(u, sigma_num_pow):
57     rho = [[0 for _ in range(4)] for _ in range(4)]
58     rho_trace = 0
59     for i1 in range(4):
60         for j1 in range(4):
61             for i2 in range(4):
62                 for j2 in range(4):
63                     rho[i2][j2] += u[i1][j1] * sigma_num_pow[i1][j1][i2][j2]
64     for i in range(4): rho_trace += rho[i][i]
65     rho_norm = [[rho[i][j] / rho_trace for j in range(4)] for i in range(4)]
66     rho_norm = np.array(rho_norm)
67     return rho_norm
68 def make_graph(rho_norm, data_name):
69     rho_real_imag = [rho_norm.real, rho_norm.imag]
70     tmp_x = np.arange(4)
71     tmp_y = np.arange(4)
72     tmp_X, tmp_Y = np.meshgrid(tmp_x, tmp_y)
73     label_bra = [
74         r"$\left| \rm{HH} \right\rangle$",
75         r"$\left| \rm{VH} \right\rangle$",
76         r"$\left| \rm{HV} \right\rangle$",
77         r"$\left| \rm{VV} \right\rangle$"
78     ]
79     label_ket = [
80         r"$\left\langle \rm{HH} \right|$",
81         r"$\left\langle \rm{VH} \right|$",
82         r"$\left\langle \rm{HV} \right|$",
83         r"$\left\langle \rm{VV} \right|$"
84     ]
85     x = tmp_X.ravel()
86     y = tmp_Y.ravel()
87     z = np.zeros_like(x)
88     dx = dy = 0.5
89     for i in range(2):
90         fig = plt.figure()
91         ax = fig.add_subplot(111, projection="3d")
92         dz = rho_real_imag[i].ravel()
93         norm = Normalize(vmin=-1, vmax=1)
94         colors = cm.coolwarm_r(norm(dz))
95         alpha = 0.7
96         colors[:, 3] = alpha
97         ax.bar3d(x, y, z, dx, dy, dz, color=colors, shade=True)
98         ax.set_xticks(tmp_x + dx / 2)
99         ax.set_xticklabels(label_bra, ha="center")
100        ax.set_yticks(tmp_y + dy / 2)

```



```
101     ax.set_yticklabels(label_ket, ha="center")
102     ax.set_zlim(-0.75, 0.75)
103     ax.zaxis.set_tick_params(labelleft=False, labelright=False, labeltop=False, labelbottom=False)
104     mappable = cm.ScalarMappable(norm=norm, cmap="coolwarm_r")
105     mappable.set_array(dz)
106     fig.colorbar(mappable, ax=ax, shrink=0.6, aspect=10, label=r"$\hat{\rho}$")
107     title = data_name + "_" + "riemaalg"[i:: 2]
108     ax.set_title(title)
109     title += ".pdf"
110     plt.savefig(title, bbox_inches = "tight")
111 def main():
112     sigma_pow = calc_sigma_pow()
113     sigma_num_pow = calc_sigma_matrix_tensor()
114     datas = ["data1", "data2"]
115     for i in range(2):
116         data = load_data(datas[i] + ".csv")
117         u = calc_uj(sigma_pow, data)
118         rho_norm = estimate_rho(u, sigma_num_pow)
119         make_graph(rho_norm, datas[i])
120 if (__name__ == "__main__"):
121     main()
```
