
STOCHASTIC PROCESSES

HW04

Hanie Hatami(99100614)

Stochastic processes course

Contents

1	Abstract	3
2	Results	3
2.1	Binning:	3
2.1.1	$D^{(1)}$	3
2.1.2	$D^{(2)}$	3
2.1.3	$D^{(4)}$	4
2.2	Interaction Matrix:	4
2.2.1	$D^{(1)}$	4
2.2.2	$D^{(2)}$	5
2.2.3	$D^{(4)}$	5
3	Conclusion	6
4	Code	7

1 Abstract

In this exercise, we want to plot $D^{(1)}$, $D^{(2)}$ and $D^{(4)}$ in terms of x once with binning and with error bars, and once with interaction matrix.

2 Results

2.1 Binning:

2.1.1 $D^{(1)}$

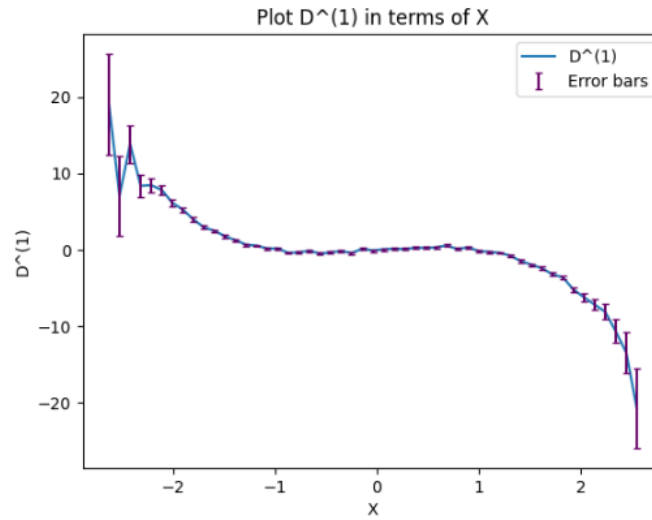


Figure 1: $D^{(1)}$ in terms of X

2.1.2 $D^{(2)}$

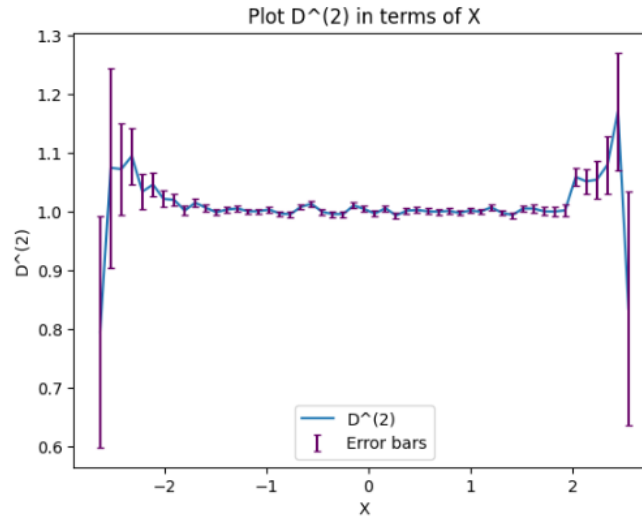


Figure 2: $D^{(2)}$ in terms of X

2.1.3 $D^{(4)}$

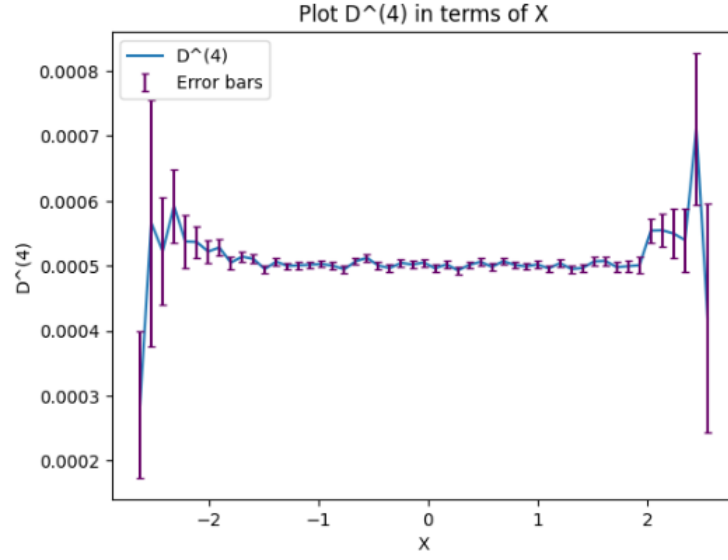


Figure 3: $D^{(4)}$ in terms of X

2.2 Interaction Matrix:

2.2.1 $D^{(1)}$

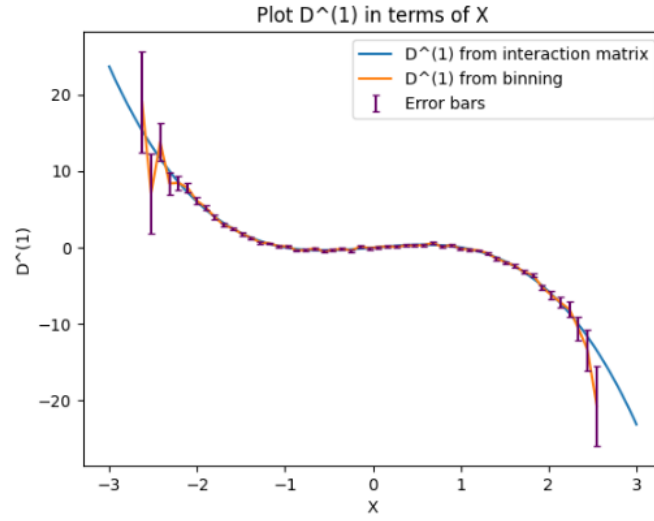


Figure 4: $D^{(1)}$ in terms of X

2.2.2 $D^{(2)}$

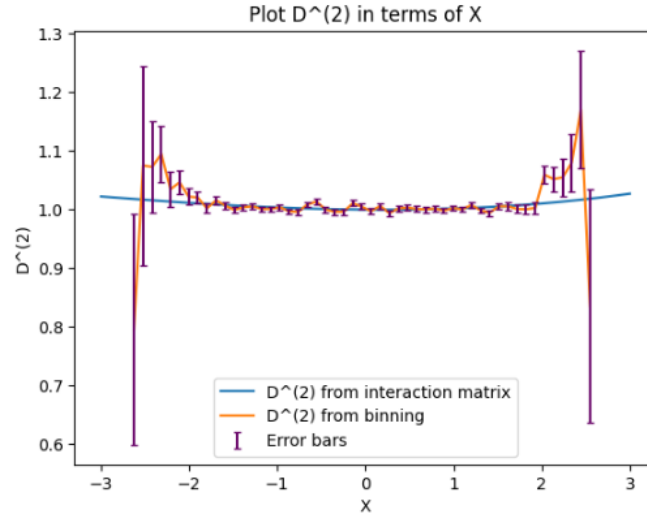


Figure 5: $D^{(2)}$ in terms of X

2.2.3 $D^{(4)}$

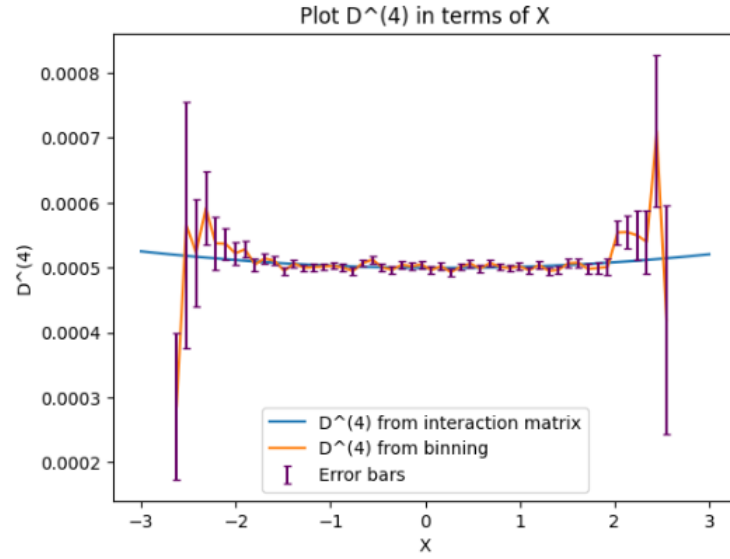


Figure 6: $D^{(4)}$ in terms of X

3 Conclusion

$D^{(1)}$ represents our trend. Plot $D^{(1)}$ shows us that the data tends to go up in very negative X 's and go down in very positive X 's, but in small and medium X 's, the probability of increasing and decreasing X is the same, and the data between There has no difference between going up and down.

$D^{(2)}$ shows the intensity of these differences and movements. As if it shows the slope of the data. As we can see in the $D^{(2)}$ plot, in very large and very small X 's, the value of $D^{(2)}$ is lower, which indicates that the intensity of the differences is less, but in the middle X 's, we see the highest value of $D^{(2)}$ and in the value of 2, which indicates that Jumps are more intense.

Since our $D^{(4)}$ is not zero, we have jump in the data and it is not continuous.

As we can see, the results we obtained from the interaction matrix part are very consistent and similar to the results we obtained by binning, and we can conclude from.

4 Code

```
1 # Import needed libraries
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Mount drive to Read data and plot it
7 # Load get data
8 from google.colab import drive
9 drive.mount('/content/drive')
10
11 # Read data and plot it
12 # Load data
13 df = np.loadtxt('/content/drive/MyDrive/data/0-001.txt')
14
15 #Plot the data
16 plt.plot(df)
17 bins = 51
18
19 # Calculating needed values
20 hist_values, bin_edges = np.histogram(df, bins=bins)
21 max = np.max(df)
22 min = np.min(df)
23
24 # Calculation differences
25 tool = len(df)
26 differences = []
27 binmid_list = []
28 for i in range(bins):
29     binmid = (bin_edges[i] + bin_edges[i+1]) / 2
30     binmid_list.append(binmid)
31     diff = []
32     for j in range(tool):
33         if df[j] >= bin_edges[i] and df[j] < bin_edges[i+1] and j != tool-1:
34             ekhtelaf = df[j+1] - df[j]
35             diff.append(ekhtelaf)
36     differences.append(diff)
37
38 # Calculating  $D^{(1)}$ 
39 D1_dt = []
40 errors_dt = []
41 for j in range(bins):
42     d1 = (sum(differences[j])) / len(differences[j])
43     res = pd.Series(differences[j]).var()
44     sem = (res / len(differences[j])) ** (1/2)
45     errors_dt.append(sem)
46     D1_dt.append(d1)
47
48 D1 = [i * 1000 for i in D1_dt]
49 errors = [i * 1000 for i in errors_dt]
50 plt.plot(binmid_list, D1, label='D^(1)')
51 plt.errorbar(binmid_list, D1, yerr=errors, fmt='none', color='#660066',
52             capsize=2, label='Error bars')
53 plt.legend()
54 plt.xlabel("X")
```

```

54 plt.ylabel("D^(1)")
55 plt.title('Plot D^(1) in terms of X')
56 plt.show()
57
58 # Calculating D^(2)
59 D2_dt = []
60 errors_dt = []
61 for j in range(bins):
62     double_list = np.power(differences[j], 2)
63     d1 = (sum(double_list))/len(differences[j])
64     res = pd.Series(double_list).var()
65     sem = (res/len(double_list))**(1/2)
66     errors_dt.append(sem)
67     D2_dt.append(d1)
68
69 D2 = [i * (1000/2) for i in D2_dt]
70 errors = [i * (1000/2) for i in errors_dt]
71 plt.plot(binmid_list, D2, label='D^(2)')
72 plt.errorbar(binmid_list, D2, yerr=errors, fmt='none', color='#660066',
73             capsize=2, label='Error bars')
74 plt.legend()
75 plt.xlabel("X")
76 plt.ylabel("D^(2)")
77 plt.title('Plot D^(2) in terms of X')
78 plt.show()
79
80 # Calculating D^(4)
81 D4_dt = []
82 errors_dt = []
83 for j in range(bins):
84     power4_list = np.power(differences[j], 4)
85     d1 = (sum(power4_list))/len(differences[j])
86     res = pd.Series(power4_list).var()
87     sem = (res/len(power4_list))**(1/2)
88     errors_dt.append(sem)
89     D4_dt.append(d1)
90
91 D4 = [i * (1000/24) for i in D4_dt]
92 errors = [i * (1000/24) for i in errors_dt]
93 plt.plot(binmid_list, D4, label='D^(4)')
94 plt.errorbar(binmid_list, D4, yerr=errors, fmt='none', color='#660066',
95             capsize=2, label='Error bars')
96 plt.legend()
97 plt.xlabel("X")
98 plt.ylabel("D^(4)")
99 plt.title('Plot D^(4) in terms of X')
100 plt.show()
101
102 data = df
103 diff = []
104 for i in range(len(data)-1):
105     diff = data[i+1]-data[i]
106     diff.append(diff)
107
108 y1 = [map(lambda x, y: x * y**i, diff, data) for i in range(4)]

```



```

109
110 tau=1
111 y = data[tau:] - data[: -tau]
112 data1 = np.ones(len(data))
113 ys = np.zeros(4)
114 for i in range(4):
115     ys[i] = np.mean(y)
116     y *= data[: -1]
117
118
119 mmnts = np.ones(7)
120 xk = np.ones(len(data))
121 for i in range(7):
122     mmnts[i] = xk.mean()
123     xk *= data
124
125 A = np.zeros((4,4))
126 for i in range(4):
127     A[i] = np.roll(mmnts, -i)[:4]
128
129 A_inv = np.linalg.inv(A)
130 phis = A_inv@ys
131 phis = phis/0.001
132 x = np.linspace(-3,3, 10001)
133
134 plt.plot(x,np.array([phis[i]*x**i for i in range(4)]).sum(0), label='D^(1)
    from interaction matrix')
135 plt.plot(binmid_list,D1, label='D^(1) from binning')
136 plt.errorbar(binmid_list, D1, yerr=errors, fmt='none', color='#660066',
    capsize=2, label='Error bars')
137 plt.legend()
138 plt.xlabel("X")
139 plt.ylabel("D^(1)")
140 plt.title('Plot D^(1) in terms of X')
141 plt.show()
142
143 tau=1
144 y = data[tau:] - data[: -tau]
145 y = y**2
146 ys2 = np.zeros(4)
147 for i in range(4):
148     ys2[i] = np.mean(y)
149     y *= data[: -1]
150
151
152 phis2 = A_inv@ys2
153 phis2 = phis2/0.002
154 x = np.linspace(-3,3, 10001)
155
156 plt.plot(x,np.array([phis2[i]*x**i for i in range(4)]).sum(0), label='D^(2)
    from interaction matrix')
157 plt.plot(binmid_list,D2, label='D^(2) from binning')
158 plt.errorbar(binmid_list, D2, yerr=errors, fmt='none', color='#660066',
    capsize=2, label='Error bars')
159 plt.legend()
160 plt.xlabel("X")
161 plt.ylabel("D^(2)")

```

```

162 plt.title('Plot  $D^{(2)}$  in terms of  $X$ ')
163 plt.show()
164
165 tau=1
166 y = data[tau:] - data[:-tau]
167 y = y**4
168 ys4 = np.zeros(4)
169 for i in range(4):
170     ys4[i] = np.mean(y)
171     y *= data[:-1]
172
173
174 phis4 = A_inv@ys4
175 phis4 = phis4/0.024
176 x = np.linspace(-3,3, 10001)
177
178 plt.plot(x,np.array([phis4[i]*x**i for i in range(4)]).sum(0), label='D(4)
    from interaction matrix')
179 plt.plot(binmid_list,D4, label='D(4) from binning')
180 plt.errorbar(binmid_list, D4, yerr=errors, fmt='none', color='#660066',
    capsize=2, label='Error bars')
181 plt.legend()
182 plt.xlabel("X")
183 plt.ylabel("D(4)")
184 plt.title('Plot  $D^{(4)}$  in terms of  $X$ ')
185 plt.show()

```