

# P2\_FilipeMarinho

May 19, 2020

## 1 Análise e Reconhecimento de Padrões - Filipe Antunes Marinho - 10438866

Implementar 4 dos seguintes conjuntos de medidas e testar sobre sinais gerados por autômatos probabilísticos como os já vistos em aula, incluindo o autômato da Figura 2 do CDT-23:

- Média, desvio padrão, entropia e evenness do número de bursts, e respectivos tamanhos nos split signals;
- Média, desvio padrão, entropia e evenness do número de distâncias intersímbolos e respectivos valores nos split signals;
- Média e desvio padrão das magnitudes do espectro de potência da transformada de Fourier discreta dos split signals (pode usar rotina Para FFT);
- Média e desvio do grau e coeficiente de aglomeração de grafos dos sinais, obtidos pelo método de visibilidade, que deve ser implementado;
- Coeficiente alpha do DFA da integral do sinal.

```
[7]: from P2aux import *

#Automatos usados
A = np.array([[0.9, 0.1], [0.9, 0.1]])
B = np.array([[0.2, 0.8], [0.2, 0.8]])
C = np.array([[0.5, 0.5], [0.5, 0.5]])
D = np.array([[0.9, 0.1, 0, 0, 0, 0], [0.882, 0.098, 0.02, 0, 0, 0], [0, 0, 0.2, 0,
→ 8, 0, 0], [0, 0, 0.194, 0.776, 0.03, 0], [0, 0, 0, 0, 0.5, 0.5], [0.01, 0, 0, 0, 0.495, 0.495]])
E = np.array([[0.5, 0.5, 0, 0], [0, 0.1, 0.9, 0], [0, 0, 0.6, 0.4], [0.7, 0, 0, 0.3]])

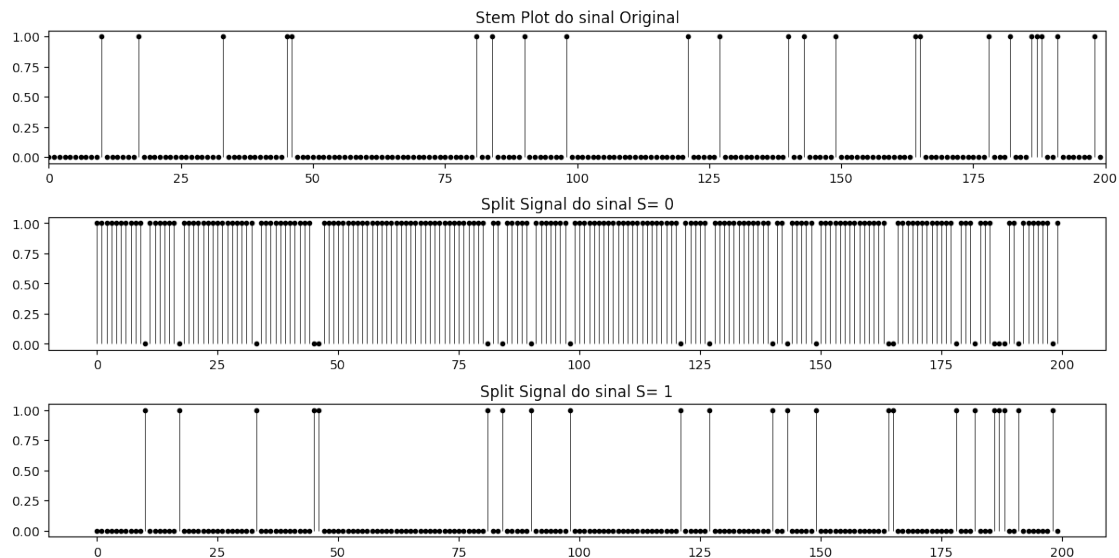
automato_a = New_automato(A, 200)
automato_b = New_automato(B, 200)
automato_c = New_automato(C, 200)
automato_d = New_automato(D, 200)
automato_e = New_automato(E, 200)
```

## 1.1 Média, desvio padrão, entropia e evenness do número de bursts, e respectivos tamanhos nos split signals

```
[8]: splits_a = Splits_all(automato_a)

means, std, entropy, eveness = [], [], [], []

for split_signal in splits_a:
    means.append(split_signal.mean())
    std.append(split_signal.std())
    entropy.append(Entropy(Relative_freq(split_signal)))
    eveness.append(Evenness(Relative_freq(split_signal)))
data = {"Média":means, "Desvio": std, "Entropia": entropy, "Eveness":eveness}
print(pd.DataFrame(data))
```



	Média	Desvio	Entropia	Eveness
0	0.885	0.319022	0.514816	1.428812
1	0.115	0.319022	0.514816	1.428812

---

## 1.2 Média, desvio padrão, entropia e evenness do número de distâncias inter-símbolos e respectivos valores nos split signals

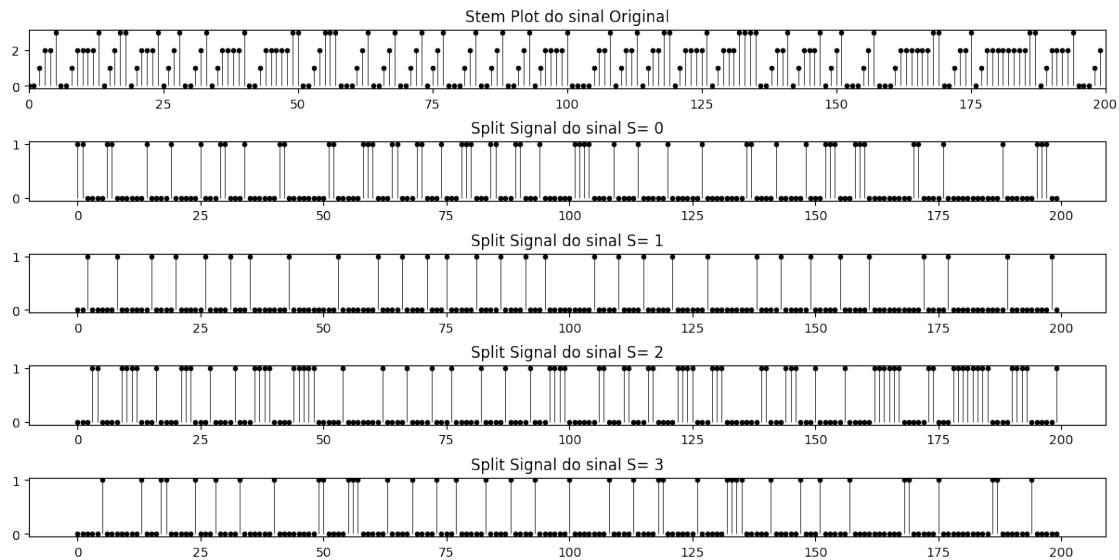
```
[10]: splits_e = Splits_all(automato_e)

means, std, entropy, eveness = [], [], [], []
for split_signal in splits_e:
```

```

means.append(Intersymbol(split_signal).mean())
std.append(Intersymbol(split_signal).std())
entropy.append(Entropy(Relative_freq(Intersymbol(split_signal))))
eveness.append(Evenness(Relative_freq(Intersymbol(split_signal))))
data = {"Média":means, "Desvio": std, "Entropia": entropy, "Eveness":eveness}
print(pd.DataFrame(data))

```



	Média	Desvio	Entropia	Eveness
0	4.766667	2.027861	2.401118	5.282125
1	5.419355	2.225573	2.812889	7.026901
2	4.064516	1.134090	1.930048	3.810678
3	5.166667	1.771691	2.489246	5.614846

### 1.3 Média e desvio padrão das magnitudes do espectro de potência da transformada de Fourier discreta dos split signals

```

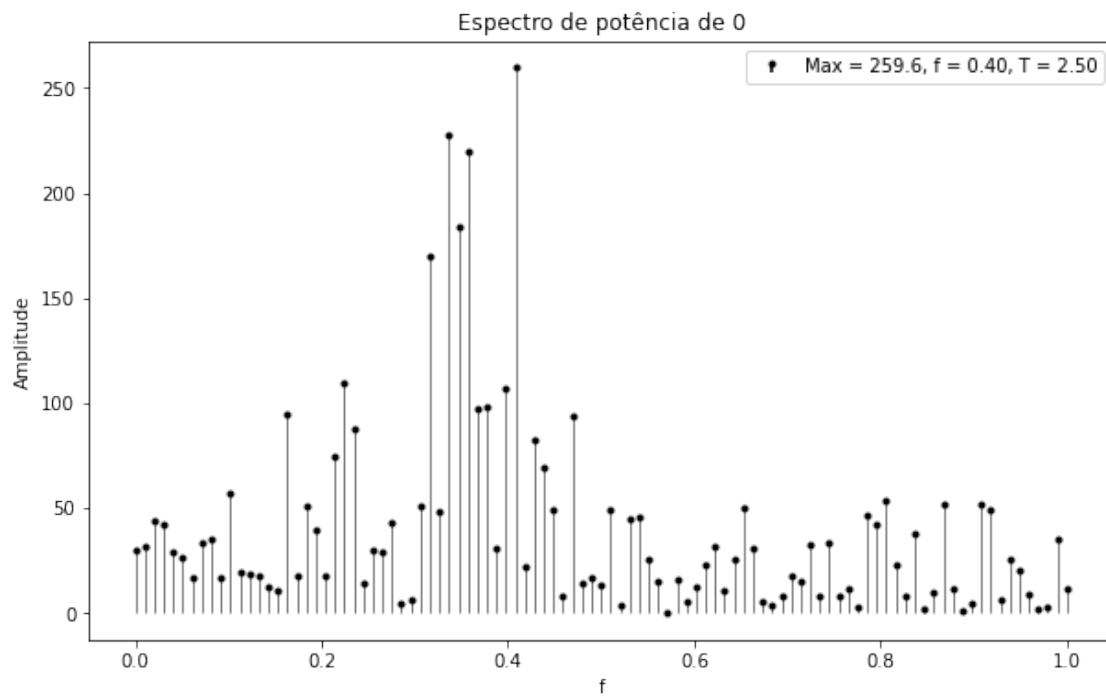
[11]: means, std, entropy, eveness, count = [], [], [], [], 0

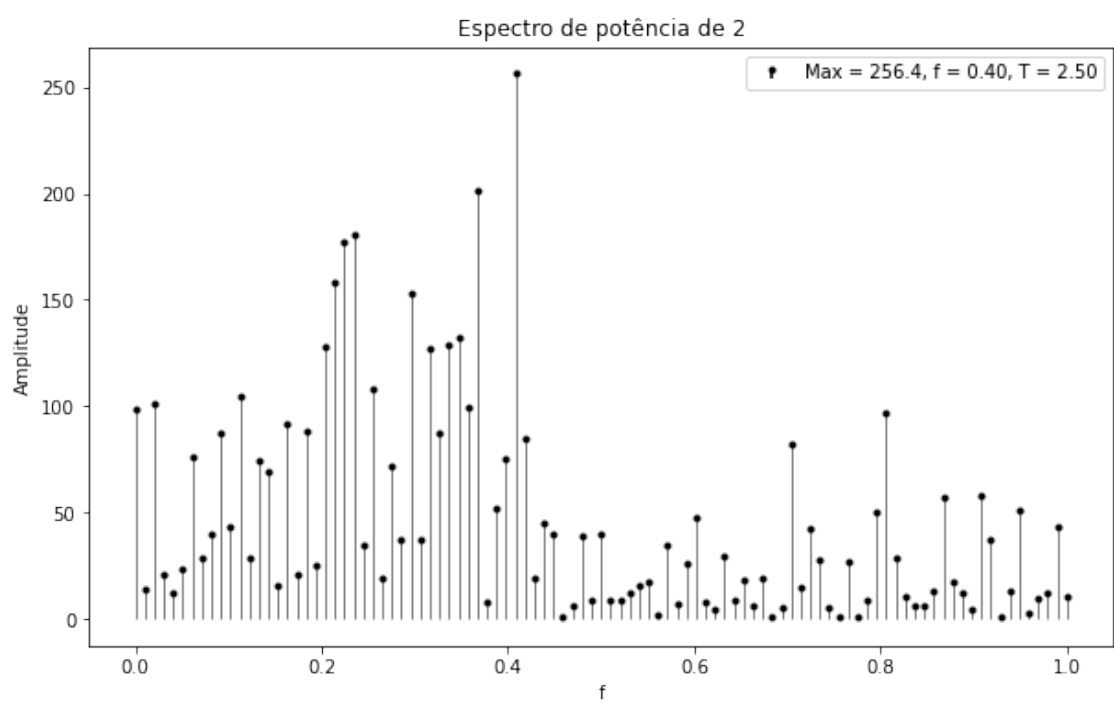
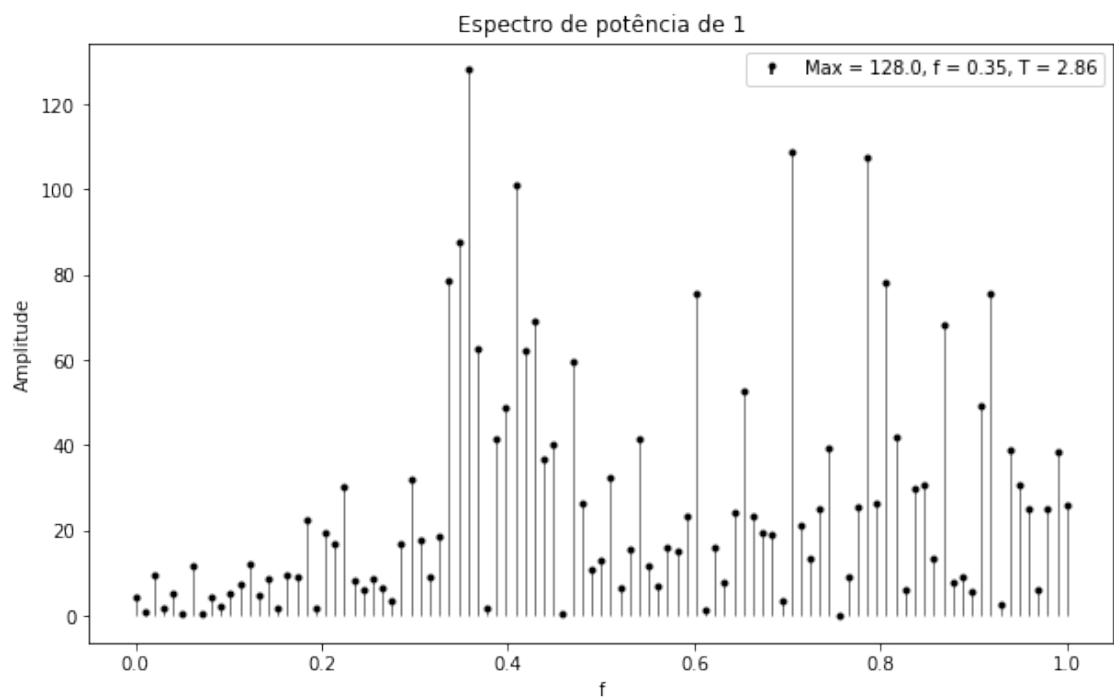
for split_signal in splits_e:
    p =PowerSpec(split_signal)
    plot_power(p, str(count))
    count+=1
    means.append(split_signal.mean())
    std.append(split_signal.std())
    entropy.append(Entropy(Relative_freq(split_signal)))
    eveness.append(Evenness(Relative_freq(split_signal)))
data = {"Média":means, "Desvio": std, "Entropia": entropy, "Eveness":eveness}

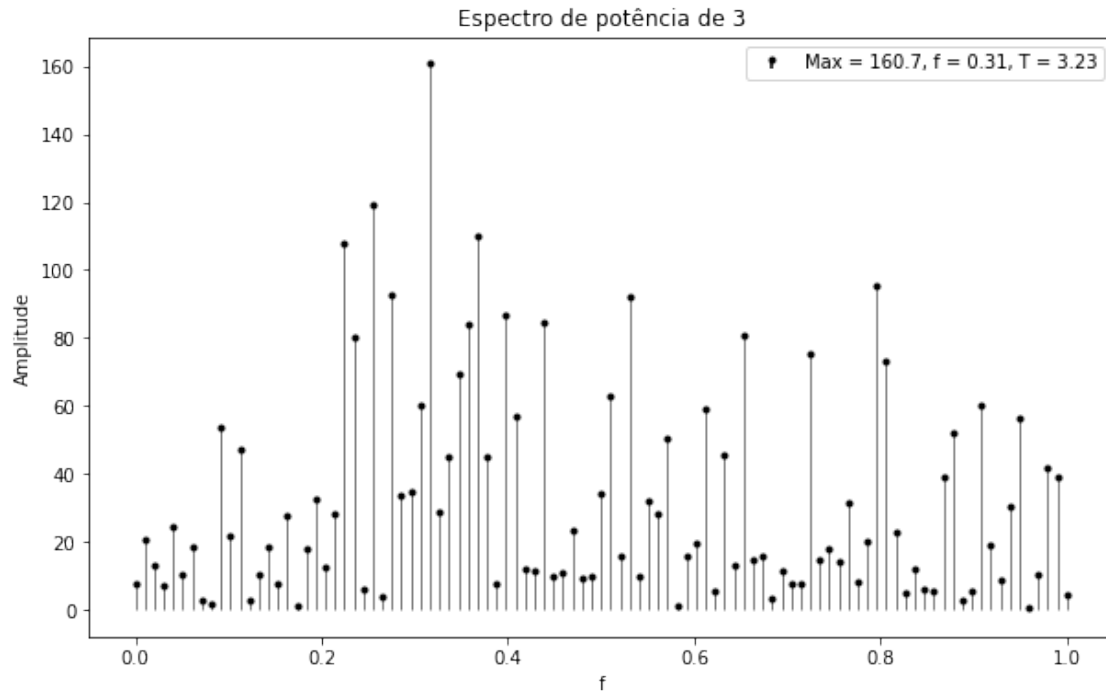
```

```
print(pd.DataFrame(data))
print()
plt.legend()
plt.show()
```

	Média	Desvio	Entropia	Eveness
0	0.275	0.446514	0.848548	1.800688
1	0.155	0.361905	0.622213	1.539234
2	0.370	0.482804	0.950672	1.932773
3	0.200	0.400000	0.721928	1.649385







#### 1.4 Média e desvio do grau e coeficiente de aglomeração de grafos dos sinais, obtidos pelo método de visibilidade

```
[12]: M_e = Visibility(automato_e)
      _ = Average_Degree(automato_e, M_e)
      _ = Clustering_Coefficient(M_e)
```

Grau:

Média: 2.58 Desvio padrão: 0.11

Coeficiente de aglomeração: 0.03

Matriz de Adjacência

