0.1 Black StatisticalImage density of states comparisons

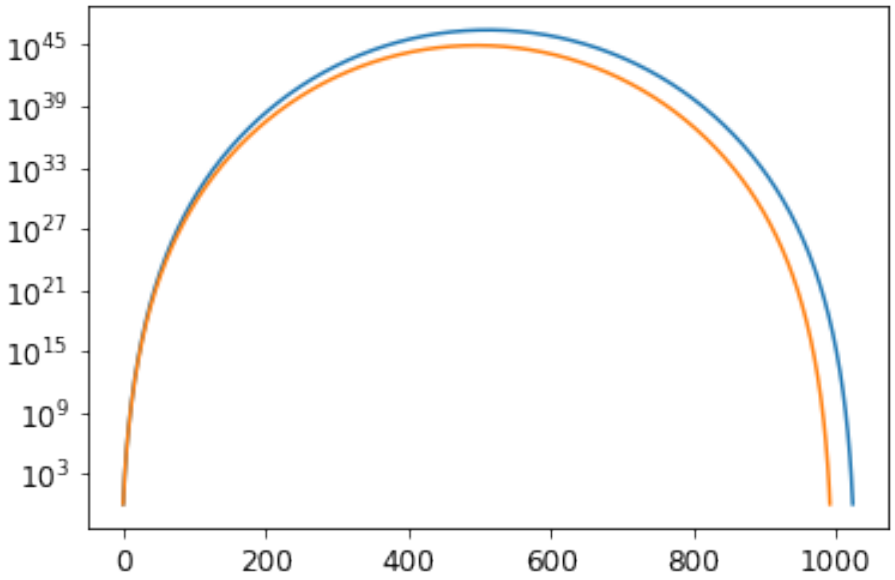
```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.size'] = 12
```

```
import sys
if 'src' not in sys.path: sys.path.append('src')
from statistical_image import bw_g, exact_bw_gs
```

$$\lim_{N\to\infty} p(E_1 = E' \mid E, N, M) = \lim_{N\to\infty} g(E - E'; N-1, M)^{-1} = \lim_{N\to\infty} g(E - E'; N, M)^{-1}$$

```
xEs, xgs = exact_bw_gs(32, 32)
xnEs, xngs = exact_bw_gs(32-1, 32)
```

```
plt.plot(xEs, xgs)
plt.plot(xnEs, xngs)
plt.yscale('log');
```

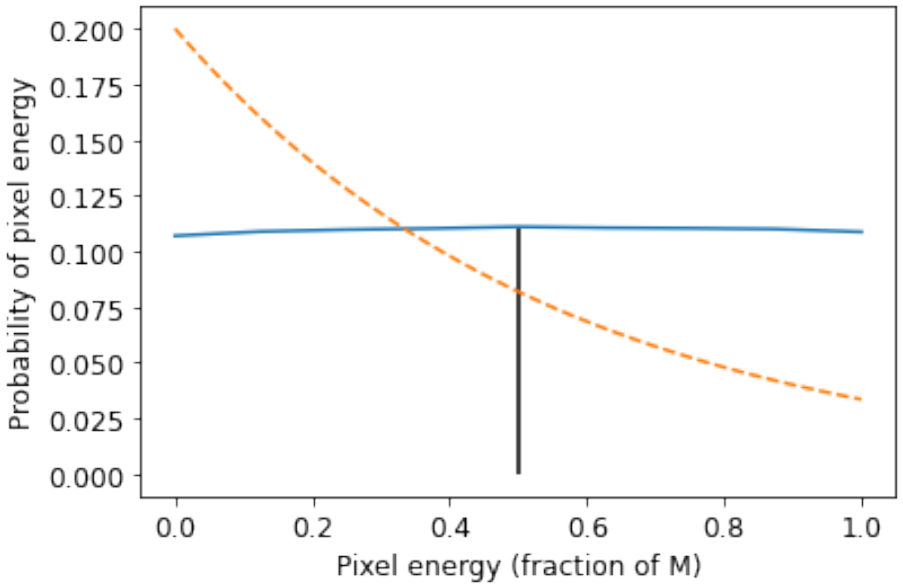



```
from functools import partial
```

```
def pixel_probs(E, N, M):
    xEs, xgs = exact_bw_gs(N - 1, M)
    E1s = np.arange(min(E, M) + 1)
    return E1s, np.vectorize(partial(bw_g, exact=False))(E - E1s, N-1, M) / bw_g(E, N, M, exact=False)
```

```
ε = 0.5
for N, M in [(64,8)]:
    E1s, pE1s = pixel_probs(ε*N*M, N, M)
    LEs = np.linspace(0, M)
```

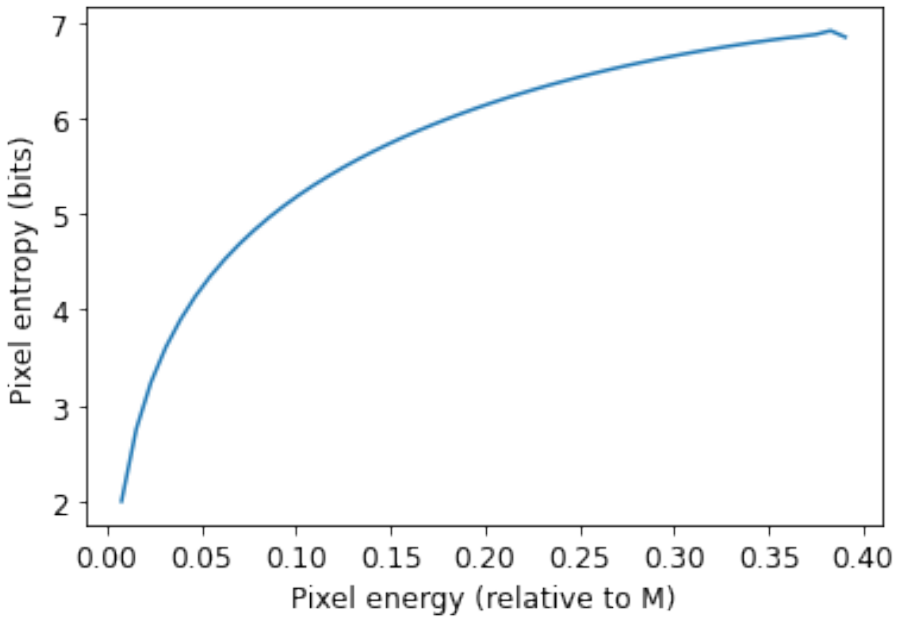
```
    μ = M*ε
    plt.plot(E1s / M, pE1s)
    plt.plot(LEs / M, (1 + (1/μ))**(- LEs) / ((1 + μ)), '--')
    plt.xlabel('Pixel energy (fraction of M)')
    plt.ylabel('Probability of pixel energy')
    plt.vlines(np.average(E1s, weights=pE1s) / M, 0, np.max(pE1s));
```




```
N, M = 128, 128
εs = np.arange(1/M, 0.5, 1/M)
Hs = []
for ε in εs:
    _, pE1s = pixel_probs(ε*N*M, N, M)
    Hs.append(-np.average(np.log2(pE1s), weights=pE1s))
```

```
<ipython-input-426-f64313f0add8>:6: RuntimeWarning: invalid value encountered in log2
  Hs.append(-np.average(np.log2(pE1s), weights=pE1s))
```

```
plt.plot(εs, Hs)
plt.xlabel('Pixel energy (relative to M)')
plt.ylabel('Pixel entropy (bits)');
```



Uniform

```
np.log2(M)
```

```
7.0
```

KL divergence

```
np.average(np.log2(M*pE1s), weights=pE1s)
```

```
4.500829346214116
```