# 67800 | Probabilistic Methods in Artificial Intelligence | Ex 1

Guy Lutsker 207029448

## Q1 - Warmup

(a) What is the general expression for the joint probability defined by our MN (like the one in Fig. 1)?

$$Solution:$$

$$P(\mathcal{Y}) = \frac{1}{Z}\prod_{i}\phi_{Y_i}\cdot\prod_{j}\phi_{P_j}$$

(b) What are the different factors and their parameters definition?

$$Solution:$$

Our factors are $\phi_{Y_i}(Y_i), \phi_{P_j}(Y_j)$ with $Y_j \subset \mathcal{Y}$

(c) Formally (as a mathematical expression) define the message decoding task.

$$Solution:$$

$$\hat{Y} = \underset{Y}{\text{argmax}} P(Y)$$

$$\text{s.t } HY = 0$$

(d) If the probability for an error in each bit is 5%, what is the chance of receiving a correct 128-bit message $\widetilde{Y}$ ?

$$Solution:$$

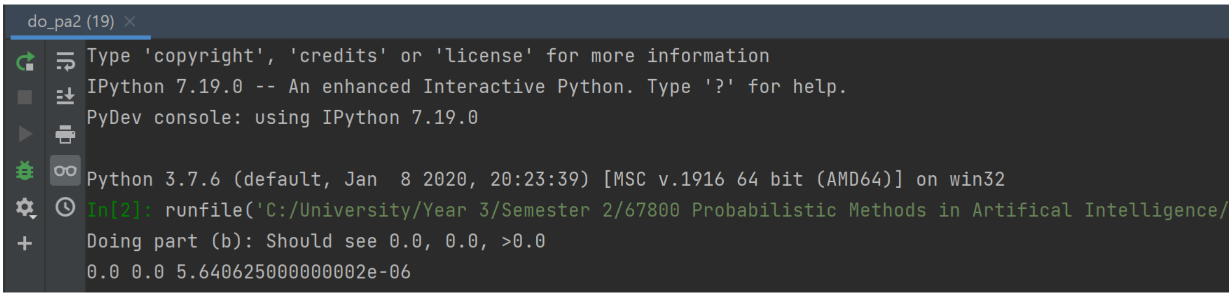
$$0.95^{128} = 0.0014$$

(approximately)

## Q2

I have implemented the code, and chose to run:

**graph.evaluateWeight(np.array([1, 0, 1, 0, 1, 0])),**
**graph.evaluateWeight(np.array([0, 1, 0, 0, 0, 0])),**
**graph.evaluateWeight(np.array([1, 0, 0, 0, 0, 1]))**



```
Type 'copyright', 'credits' or 'license' for more information
IPython 7.19.0 -- An enhanced Interactive Python. Type '?' for help.
PyDev console: using IPython 7.19.0

Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)] on win32
In[2]: runfile('C:/University/Year 3/Semester 2/67800 Probabilistic Methods in Artifical Intelligence/
Doing part (b): Should see 0.0, 0.0, >0.0
0.0 0.0 5.640625000000002e-06
```

Figure 1: Output of part b

As we can see the output was as expected by the algorithm.

## Q3

Implemented.

## Q4

Run your loopy belief propagation code for 50 iterations. After the final iteration, plot the estimated posterior probability (conditioned on the received, noisy message) that each codeword bit equals one:

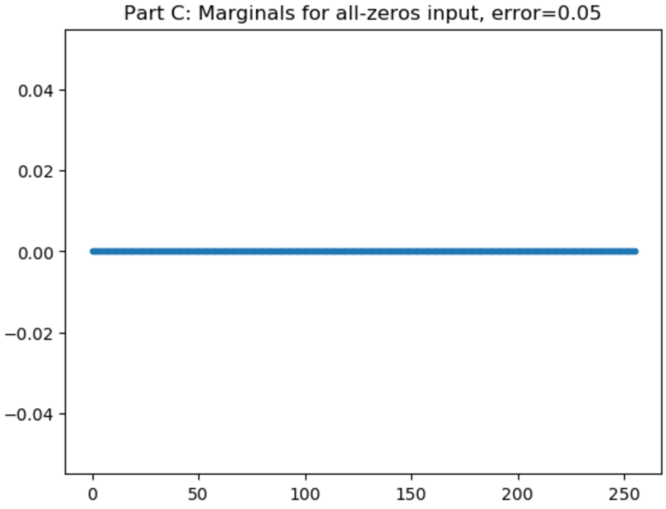


Figure 2: Output of part c

If we decode by setting each bit to the maximum of its corresponding marginal, would we find the right codeword?

Answer: We would still be able to finf the right answer even if we decodes by setting each bit to the maximum of its corresponding marginal since the argmax value would be 0 for each character of the word.

## Q5:

On a single plot, display 10 curves showing Hamming distance versus iteration for each trial:

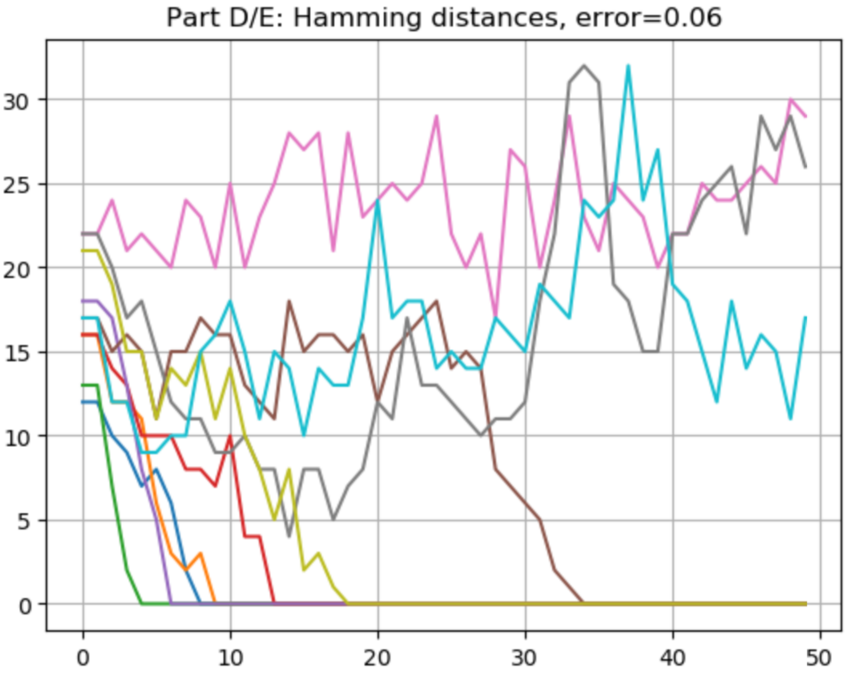


Figure 3: Output for Q5

Is BP a reliable decoding algorithm?

Answer: It would seem as though BP is not a reliable decoding algorithm as some runs did not converge to the right answer and so its successes rate is not great.

## Q6

Repeat part (d) with two higher error probabilities, $\varepsilon = 0.08$ and $\varepsilon = 0.10$:

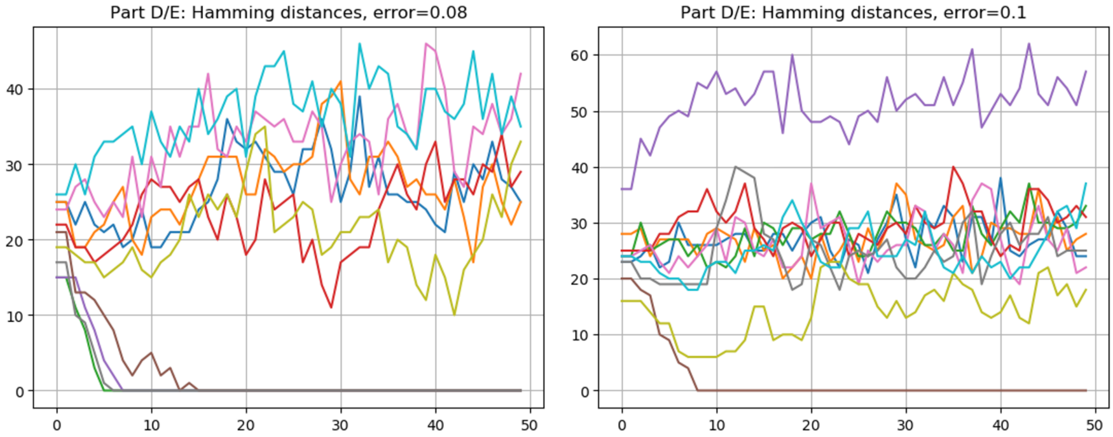


Figure 4: Output for Q6

As we can see, as $\varepsilon$ increases more runs are unable to reach the right answer, and tend to fail. This makes seance as $\varepsilon$ represents the probability of error. In addition the runs that do fail have a larger Hamming distance as $\varepsilon$ increases - $\varepsilon = 0.06$ had a maximal distance of 30, $\varepsilon = 0.08$ had 40, and $\varepsilon = 0.1$ had 60, which too makes seance.

## Q7

Plot first the noisy input image and then images showing the output of the sum-product decoder after 0, 1, 2, 3, 5, 10, 20, and 30 iterations:

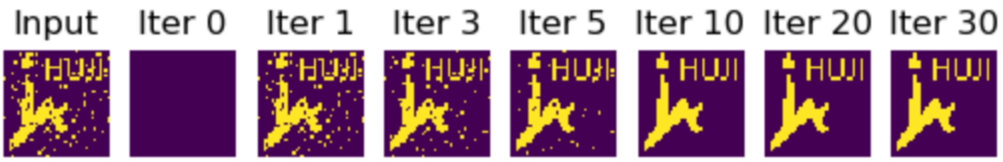


Figure 5: Output for Q7