

# Information Theory

submit on Gradescope before May 2nd for drops

# Preparation

- Read the csv file in (header = None!!!!)

```
dictionary = pd.read_csv("Assignment10-WordFrequencies.csv", header = None)
```

- Calculate the frequency

```
SumOfWordFrequencies = np.sum(dictionary[1])
```

```
dictionary.insert(2, "P(word)", dictionary[1]/SumOfWordFrequencies)
```

```
dictionary.head()
```

	0	1	P(word)
0	i	152884	0.039609
1	and	128635	0.033327
2	the	123648	0.032035
3	you	106042	0.027473
4	it	99341	0.025737

```
dictionary.head()
```


	0	1
0	i	152884
1	and	128635
2	the	123648
3	you	106042
4	it	99341

# Problem 1: Entropy


$$H[X] = \sum_x P(x) \log \frac{1}{P(x)}$$


- 2 based log:  $\log(1/p, 2)$  or  $\text{np.log2}()$


☒ Resolved ☐ Unresolved


 **Anonymous** 2 hours ago  
hello, my peers

are people getting an entropy of 8.938?

 **Anonymous** 1 hour ago yes

 **Zoe Ferguson** 1 hour ago yes!

 **Christiann Savage** 37 minutes ago Shoot. I'm getting 2.742. ;-( Am I the only one?

 **zhifeng** (anon. to classmates) Just now I got 6.195...

Reply to this followup discussion

## Problem 2: 20 questions

$$H[X] = \sum_x P(x) \log \frac{1}{P(x)}$$

- Don't need code. This one is interesting.
- What is the entropy for yes/no?
- What does entropy actually means in this scenario?
  - (a) if the word being guessed is chosen according to frequency,
  - (b) if the word is chosen uniformly (e.g. with equal probability for each word).

**Entropy says on average how many bits of information do I need to convey a given outcome?**

# Problem 3: Conditional Entropy

- Find the first/last character, and first vowel.

$$H[X|d] = \sum_x P(x | d) \log \frac{1}{P(x | d)}$$

What is  $P(x|d)$ ?

- (the frequency of word  $x$ ) / (sum of frequency with **first letter  $d$** )
- (the frequency of word  $x$ ) / (sum of frequency with **last letter  $d$** )
- (the frequency of word  $x$ ) / (sum of frequency **with vowel  $d$** )



**Duc Nguyen** (anon. to classmates) 15 hours ago

For this one, what is  $p(\text{word} | \text{a specific first letter})$ ?

Like, is it = frequency of that word / frequency of all words starting with that letter

OR =  $1 / \text{number of words starting with that letter}$ ?



How should we handle cases where a word doesn't contain a vowel?



**Mugdha Bhusari** 5 hours ago It's entropy will have 0 contribution

```
In [52]: dictionary.iloc[3, 0][0]
```

```
Out[52]: 'y'
```

Finding the last letter of the word

```
In [54]: dictionary.iloc[3, 0][-1]
```

```
Out[54]: 'u'
```

Finding the first vowel of the word

```
In [55]: vowels = ['a', 'e', 'i', 'o', 'u']
```

```
In [58]: for i in dictionary.iloc[3, 0]:  
          print(i)  
          if i in vowels:  
              first_vowel = i  
              break
```

# Problem 4: Conditional Probability

$D = \{\text{first letter, second letter, vowel}\}$

What is  $P(d)$  ?

- **Conditioning on a particular value:**

$$H[X|d] = \sum_x P(x | d) \log \frac{1}{P(x | d)}$$

- **Conditioning on a random variable (where  $d$  are possible values in  $D$ ):**

$$H[X | D] = \sum_d P(d) H[X | d]$$



**i Sam Cheyette** 2 hours ago You want to calculate how much you should expect your uncertainty to decrease after learning, e.g., the first letter of the word. I.e., calculate  $H[\text{words}] - H[\text{words}|\text{first letter}]$ . To calculate  $H[\text{words}|\text{first letter}]$ , you will need to figure out what the entropy after learning each possible first letter is; and then take the weighted sum over those values, where the weight corresponds to how likely each letter is to be the first letter of a word. So if  $L$  is the set of all possible first letters, then  $H[\text{words}|\text{first letter}] = \sum_{x \in L} H[\text{words}|\text{starts with } x] p(\text{starts with } x)$ .



**i Sam Cheyette** 2 hours ago You're trying to calculate how much your entropy over words is reduced after (e.g.) learning the first letter of the word. I.e., calculate  $H[\text{words}] - H[\text{words}|\text{first letter}]$ .

# Problem 5: Word Length v.s. Surprisal

- Find the first/last character, and first vowel.
  - **$\log(1/p) = -\log(p)$  is called the surprisal of an event**
- Average Surprisal:

$p(x) = 1/N$ . Please see Sam's Clarification on Piazza!



# Shannon's Sense

- **By definition, codewords are decodable into events (e.g. you can't just assign everything a codeword “1”).**  
(These correspond to labeled binary trees).
- **Higher probability events have shorter codewords**  
(= less information required to convey that they happened)
- **Lower probability events have longer codewords**  
(= more information required to convey that they happened)

## Problem 6: Extra Credit

# Some Fun stuff: Zipf's Law

<https://drive.google.com/file/d/1--gTdDXkF6fjtfQUdh5T0PG11yhMJgg/view?usp=sharing>

Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. Thus the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, etc.