Margot Wagner
A53279875
CSE 250A Principles of AI
HW 1
Due: 2pm 10/13/20

1.1 Conditioning on Background Evidence

(a) Product Rule

From the definition of conditional probability

$$P(X,Y|E) = \frac{\boxed{P(X,Y,E)}}{P(E)}$$

$$\boxed{P(X,Y,E)} = \boxed{P(Y,E)}\,P(X|Y,E) \qquad \text{product rule}$$

$$\boxed{P(Y,E)} = P(E)\,P(Y|E) \qquad \text{product rule}$$

$$P(X,Y|E) = \frac{\cancel{P(E)}\,P(Y|E)\,P(X|Y,E)}{\cancel{P(E)}} \qquad \text{subst \& simplify}$$

$$P(X,Y|E) = P(X|Y,E)\,P(Y,E)$$

(b) Bayes' Rule

Starting with the LHS

$$P(X|Y,E) = \frac{P(X,Y,E)}{P(Y,E)} \qquad \text{conditional def}$$

$$= \frac{P(Y,E|X)\,P(X)}{P(Y,E)} \qquad \text{product rule}$$

$$= \frac{P(Y,E|X)\,P(X)\,P(E)}{P(Y|E)} \qquad \text{product rule}$$

$$= \frac{P(X,Y,E)\,P(X)\,P(E)}{P(X)\,P(Y|E)} \qquad \text{conditional prod def}$$

$$= \frac{P(Y|X,E)\,P(X,E)\,P(E)}{P(Y|E)} \qquad \text{product rule}$$

$$= \frac{P(Y|X,E)\,P(E)\,P(X|E)}{P(Y|E)\,P(E)} \qquad \text{product rule}$$

$$P(X|Y,E) = \frac{P(Y|X,E)\,P(X|E)}{P(Y|E)} \qquad \text{simplify}$$

(c) MARGINALIZATION

Show $P(X|E) = \sum_y P(X, Y=y|E)$

$P(X,E) = P(E)P(X|E)$ product rule

$= \sum_y P(X,E,Y=y)$ marginalization

$= \sum_y P(E)P(X,Y=y|E)$ conditional prob

$= P(E)\sum_y P(X,Y=y|E)$ remove $P(E)$ from $\sum_y$

$P(X|E)P(\cancel{E}) = P(\cancel{E})\sum_y P(X,Y=y|E)$ product rule simplify

$P(X|E) = \sum_y P(X,Y=y|E)$

1.2 Conditional Independence

Looking at the LHS of (2), we can express it using the conditionalized form of Bayes rule derived in 1.1(b)

$$P(X|Y,E) = \frac{P(Y|X,E)\,P(X|E)}{P(Y|E)}$$

If (1) is true, we can substitute $P(Y|X,E) = P(Y|E)$:

$$P(X|Y,E) = \frac{\cancel{P(Y|E)}\,P(X|E)}{\cancel{P(Y|E)}}$$

$$P(X|Y,E) = P(X|E)$$

Thus (1) imlies (2)

From the conditionalized version of the product rule proved in 1.1a,

$$P(X,Y|E) = P(X|Y,E)P(Y|E)$$

If (2) is true, we can rewrite this as

$$P(X,Y|E) = P(X|E)P(Y|E)$$

Thus (2) implies (3)

From the product rule proved in 1.1a

$$P(X,Y|E) = P(Y|X,E)P(X|E)$$

If (3) is true, this can be substituted on the RHS:

$$(3)\quad P(X,Y|E) = P(X|E)P(Y|E)$$

$$P(Y|X,E)\cancel{P(X|E)} = \cancel{P(X|E)}\,P(Y|E)$$

$$P(Y|X,E) = P(Y|E)$$

Therefore, (3) implies (1)

We have now proved all the statements are equivalent as $(1) \rightarrow (2) \rightarrow (3) \rightarrow (1)$

1.3 Creative Writing

(a) Cumulative evidence

$$P(Z=1) > P(Z=1 \mid X=1) > P(Z=1 \mid X=1, Y=1)$$

Z : get an "A" in 250A (0 : not A ; 1 : A)

X : not attending class (0 : attend ; 1 : not attend)

Y : not doing PAs (0 : do HW ; 1 : don't)

The probability of getting an A is higher than the probability of getting an A given you did not go to class which is higher than if you both did not go to class and did not do HW.

(b) Explaining away

$$P(X=1 \mid Z=1) > P(X=1)$$

$$P(X=1 \mid Z=1, Y=1) < P(X=1 \mid Z=1)$$

X : having a headache
Z : drank last night
Y : drank lots of water

The probability of headache in general is less than having a headache if you drank last night, but having a headache after drinking is more likely than having a headache after drinking given you also drank a lot of water.

(c) Conditional independence

$$P(Y=1, Z=1) \neq P(Y=1)P(Z=1)$$
$$P(Y=1, Z=1 \mid X=1) = P(Y=1 \mid X=1)P(Z=1 \mid X=1)$$

Y + Z independent given X (otherwise dependent)

Y : pet allergies
Z : Scratched furniture
X : cat

The pet allergies and scratched furniture are directly caused by the cat, but neither has a direct affect on the other.

1.4 Bayes Rule

C : has COVID $\in \{0,1\}$

T : test positive $\in \{0,1\}$

(a)

$$C \longrightarrow T$$

$$\boxed{\begin{matrix} P(C=1)=0.01 \\ P(C=0)=0.99 \end{matrix}}$$

,

C	$P(T=1 \mid C)$	$P(T=0 \mid C)$
0	0.05	0.95
1	0.98	0.02

(b) $P(C=0 \mid T=1)$

$$P(Y \mid X) = \frac{P(X \mid Y)P(Y)}{P(X)}$$

$$P(C \mid T) = \frac{P(T \mid C)P(C)}{P(T)}$$

$$P(C=0 \mid T=1) = \frac{P(T=1 \mid C=0)P(C=0)}{P(T=1)}$$

We know

$$P(T=1 \mid C=0) = 0.05$$
$$P(C=0) = 0.99$$

We need to know

$$P(T=1)$$

$$P(T=1) = \sum_c P(C=c, T=1) \quad \text{marginalization}$$

$$= \sum_c P(C=c)P(T=1 \mid C=c) \quad \text{product rule}$$

$$= P(C=0)P(T=1 \mid C=0) + P(C=1)P(T=1 \mid C=1)$$

$$= 0.99(0.05) + (0.01)(0.98)$$

$$= 0.0593$$

$$P(C=0 \mid T=1) = \frac{(0.05)(0.99)}{0.0593} =$$

$$= 0.835$$

(c) $P(C=1|T=0)$

$$P(C=1|T=0) = \frac{\overset{0.02}{P(T=0|C=1)}\,\overset{0.01}{P(C=1)}}{P(T=0)}$$

$$P(T=0) = \sum_c P(C=c, T=0)$$

$$= \sum_c P(C=c)\,P(T=0|C=c)$$

$$= P(C=0)P(T=0|C=0) + P(C=1)P(T=0|C=1)$$

$$= 0.99(0.95) + 0.01(0.02)$$

$$= 0.9704$$

$$P(C=1|T=0) = \frac{(0.02)(0.01)}{0.9704}$$

$$= 2.13 \times 10^{-4}$$

1.5 Entropy

(a) Maximal entropy

$$\mathcal{H}[X] = -\sum_{i=1}^{n} p_i \log p_i \quad \text{where} \quad \sum_i p_i = 1$$

$$\mathcal{L}(x,\lambda) = f(x) - \lambda g(x)$$

$$\frac{\partial}{\partial p_i}\left[-\sum_{i=1}^{n} p_i \log p_i - \lambda\left(\sum_{i=1}^{n} p_i - 1\right)\right]$$

$$= \sum_{i=1}^{n}(-\ln(p_i) - 1) - \lambda \sum_{i=1}^{n}(1)$$

$$= \sum_{i=1}^{n}(-\ln(p_i) - 1) - \lambda n$$

$$0 = -\sum_{i=1}^{n}\ln(p_i) - n - \lambda n$$

$$0 = -\sum_{i=1}^{n}\ln(p_i) - (1+\lambda)n$$

$$(1+\lambda)n = -\sum_{i=1}^{n}\ln(p_i)$$

$$\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} e^{-(1+\lambda)n} = 1$$

$$n e^{-(1+\lambda)n} = 1$$

$$e^{-(1+\lambda)n} = 1/n$$

$$\text{since} \quad e^{-(1+\lambda)n} = p_i$$

$$p_i = 1/n$$

(b) Joint entropy

Show that $H(X_1, X_2 \dots X_n) = \sum_{i=1}^{n} H(X_i)$

$$P(x_1, x_2, \dots x_n) = \prod_{i=1}^{n} P(x_i)$$

$$\mathcal{H}(X_1, X_2, \dots X_N) = -\sum_{x_1}\sum_{x_2}\cdots\sum_{x_n} P(x_1, x_2 \dots x_n) \log P(x_1, x_2 \dots x_n)$$

$$= -\sum_{x_1}\cdots\sum_{x_n} P(x_1 \dots x_{n-1}) P(x_n) \log P(x_1 \dots x_{n-1}) P(x_n) \qquad \text{(indep)}$$

$$= -\sum_{x_1}\cdots\sum_{x_n} P(x_1 \dots x_{n-1}) P(x_n) \log P(x_1 \dots x_{n-1}) + \log(P(x_n)) \qquad \text{(log)}$$

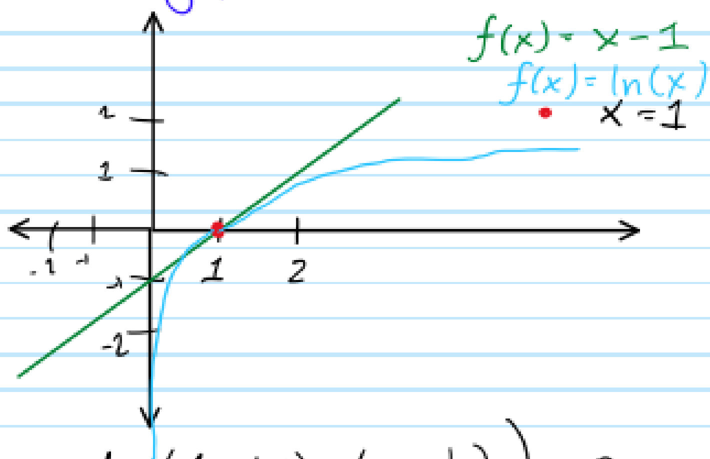
$$= -\underbrace{\sum_{x_1} \cdots \sum_{x_{n-1}} P(x_1 \ldots x_{n-1}) \log P(x_1 \ldots x_{n-1})}_{H(x_1 \ldots x_{n-1})} \underbrace{\sum_{x_n} P(x_n)}_{1}$$

$$= H(x_1 \ldots x_{n-1}) + H(x_n)$$

We can use the same logic to prove complete independance $\sum_{i=1}^{n} H(X_i)$

1.6 Kullback-Leibler distance

(a) Sketch graphs



$$\frac{d}{dx}\left(\log(x) - (x-1)\right) = 0$$

$$= \frac{1}{x} - 1 = 0$$

$$\frac{1}{x} = 1$$

$$x = 1 \checkmark$$

(b) Prove $KL(p,q) \geq 0$

From 1.6a $\ln(x) \leq x - 1$

$-\ln(x) \geq 1 - x$ mult by (−)

let $x = \frac{q_i}{p_i} \rightarrow -\ln(\frac{q_i}{p_i}) \geq 1 - \frac{q_i}{p_i}$

$KL(p,q) = \sum p_i \log(p_i/q_i)$ mult by (−)

$= -\sum_i p_i \log(q_i/p_i)$ subst from above

$KL(p,q) \geq -\sum_i p_i (1 - \frac{q_i}{p_i})$

$= \sum_i p_i - \sum_i q_i = 0$

$\sum_i p_i \ln(\frac{p_i}{q_i}) \geq 0$

(c) Derive tighter lower bound

$\ln x \leq x - 1$ (a)

$2 \ln\sqrt{x} \leq x - 1$ subst

$2\ln\sqrt{x} \geq 2(1 - \frac{1}{\sqrt{x}})$ subst

$$\sum_i \sqrt{p_i}^2 + \sum_i \sqrt{q_i}^2 = 2 = 2\sum_i p_i$$ def + bound

$$\sum_i p_i \ln\left(\frac{p_i}{q_i}\right) \geq \sum_i 2p_i\left(1-\sqrt{\frac{q_i}{p_i}}\right) = 2\sum_i p_i - \sqrt{p_i q_i}$$ subst inequality

$$= \sum_i \sqrt{p_i}^2 + \sum_i \sqrt{q_i} - 2\sum_i \sqrt{p_i q_i}$$

$$= \sum_i (\sqrt{p_i} - \sqrt{q_i})^2$$

$$\therefore \boxed{\sum_i p_i \ln\left(\frac{p_i}{q_i}\right) \geq \sum_i (\sqrt{p_i} - \sqrt{q_i})^2}$$

(d) Symmetry Counterexample

For a distribution where

	1	2
P	0.1	0.9
q	0.7	0.3

$$KL(p,q) = \sum_i p_i \log(p_i/q_i)$$

$$= 0.1\log\left(\frac{0.1}{0.7}\right) + 0.9\log\left(\frac{0.9}{0.3}\right)$$

$$= 0.345$$

$$KL(q,p) = \sum_i q_i \log(q_i/p_i)$$

$$= 0.7\log(0.7/0.1) + 0.3\log(0.3/0.9)$$

$$= 0.448$$

$$0.345 \neq 0.448$$

1.7 Mutual Information

$$I(X,Y) = \sum_x \sum_y P(x,y) \log\left[\frac{P(x,y)}{P(x)P(y)}\right]$$

(a) Nonnegative

$$\ln(x) \le x - 1 \qquad \text{1.6a}$$
$$\ln(x) \ge 1 - \frac{1}{x} \qquad \text{rearrange}$$

$$\sum_x \sum_y P(x,y) \ln\left[\frac{P(x,y)}{P(x)P(y)}\right] \ge \sum_x \sum_y P(x,y)\left(1 - \frac{P(x)P(y)}{P(x,y)}\right)$$

$$\ge \sum_x \sum_y P(x,y) - P(x)P(y) = 0$$

$$\therefore \sum_x \sum_y P(x,y) \ln\left[\frac{P(x,y)}{P(x)P(y)}\right] \ge 0$$

(b) MI vanishes iff X + Y independent

If X and Y are independent then

$$\frac{P(x,y)}{P(x)P(y)} = 1 \qquad \text{by definition}$$

Thus

$$I(X,Y) = \sum_x \sum_y P(x,y) \ln\left[\frac{P(x,y)}{P(x)P(y)}\right]$$

$$= \sum_x \sum_y P(x,y) \ln(1) = 0$$

This is the only circumstance where this would occur

1.8 Compare & Contrast

(a) Yes, belief network 1 has a conditional dependence of $P(Z|X)$ that is not present in 2. If Y and Z are conditioned on X then they become conditionally independent in belief network 1 but not 2.

(b) No additional independences

(c) Yes, again in #1, Y and Z are conditionally independent when conditioned on X. This independence is not present in #3

1.9 Hangman

In [1]:
```
import pandas as pd
import string
```

In [2]:
```
# Read in word counts file
word_counts = pd.read_csv('./hw1_word_counts_05.txt', sep=' ',
                          header=None, names=['word', 'count'])

# Compute probabilities of words
word_counts['P(W=w)'] = word_counts['count'] / word_counts['count'].sum(
axis=0)

word_counts.head()
```

Out[2]:

	word	count	P(W=w)
0	AARON	413	0.000054
1	ABABA	199	0.000026
2	ABACK	64	0.000008
3	ABATE	69	0.000009
4	ABBAS	290	0.000038

(a) Print out 15 most frequent and 14 least frequent 5-letter words.

Compute the prior probabilities $P(w) = COUNT(w)/COUNT_{total}$. As a sanity check, print out the 15 most frequent 5-letter words, as well as the 14 least frequent words.

In [3]:
```
top_15 = word_counts.sort_values(by='P(W=w)', ascending=False)
top_15.head(15)
```

Out[3]:

	word	count	P(W=w)
5821	THREE	273077	0.035627
5102	SEVEN	178842	0.023333
1684	EIGHT	165764	0.021626
6403	WOULD	159875	0.020858
18	ABOUT	157448	0.020542
5804	THEIR	145434	0.018974
6320	WHICH	142146	0.018545
73	AFTER	110102	0.014365
1975	FIRST	109957	0.014346
1947	FIFTY	106869	0.013943
4158	OTHER	106052	0.013836
2073	FORTY	94951	0.012388
6457	YEARS	88900	0.011598
5806	THERE	86502	0.011286
5250	SIXTY	73086	0.009535

In [4]:
```
last_14 = word_counts.sort_values(by='P(W=w)')
last_14.head(14)
```

Out[4]:

	word	count	P(W=w)
3554	MAPCO	6	7.827935e-07
712	BOSAK	6	7.827935e-07
895	CAIXA	6	7.827935e-07
4160	OTTIS	6	7.827935e-07
5985	TROUP	6	7.827935e-07
1107	CLEFT	7	9.132590e-07
2041	FOAMY	7	9.132590e-07
977	CCAIR	7	9.132590e-07
5093	SERNA	7	9.132590e-07
6443	YALOM	7	9.132590e-07
5872	TOCOR	7	9.132590e-07
3978	NIAID	7	9.132590e-07
4266	PAXON	7	9.132590e-07
1842	FABRI	7	9.132590e-07

Do your results make sense?
Yes. The most common words are ones that are often used including numbers. The least common words are very specific and not used in daily speech.

(b) The Best Next Guess

Consider the following stages of the game. For each of the following, indicated the best next guess -- namely, the letter l that is most probable to be among the missing letters. Also report the probability $P(L_i = l$ for some i in $1, 2, 3, 4, 5 | E)$ for your guess l. Your answers should fill in the last two columns of this table.

(b) Chart

correctly guessed	incorrectly guessed	best next guess ℓ	$P(L_i=\ell$ for some $i \in \{1,2,3,4,5\} \mid E)$
– – – – –	{}	E	0.5394
– – – – –	{A, I}	E	0.6214
A – – – R	{}	T	0.9816
A – – – R	{E}	O	0.9913
– – U – –	{O, D, L, C}	T	0.7045
– – – – –	{E, O}	I	0.6366
D – – I –	{}	A	0.8207
D – – I –	{A}	E	0.7521
– U – – –	{A, E, I, O, S}	Y	0.6270

(c)

In [22]:
```
def p_e_given_w(correct , incorrect):
    '''
    Function to give P(E|W) for each word. Checks if the guessed correct
ly/incorrectly
    matches with each word.

    correct:   list of correct guesses
    incorrect: list of incorrect guesses

    p_e_w: list of probabilities P(E|W) for each word
    '''
    # Initialize P(E|W) for each word
    p_e_w = [None]*len(word_counts.index)

    # Compare words to correct/incorrect guesses to determine P(E|W)
    for i in range(len(word_counts.index)):
        for letter, guess in zip(word_counts['word'][i], correct):
            # P(E|W) = 0 if the word contains a letter that is "incorrec
t"
            if letter in incorrect:
                p_e_w[i] = 0
                break

            # P(E|W) = 0 if the letter is in the guessed spot but the
            # word doesn't have the same letter in that spot.
            elif (guess != None) and (letter != guess):
                p_e_w[i] = 0
                break

            # P(E|W) = 0 if a letter has been guessed and is in the word
            # but not in the same positions as the word in question.
            elif (letter in correct) and (guess == None):
                p_e_w[i] = 0
                break

            else:
                p_e_w[i] = 1

    return p_e_w
```

In [23]:

```
def p_l_given_w(letter):
    '''
    Function that gives the probabilites of a certain letter given a wor
d.

    letter: letter of interest from alphabet

    p_l_w: probability of that letter for each word
    '''
    # Initialize P(L|W) for each word
    p_l_w = [None]*len(word_counts.index)

    # Check if letter is anywhere in word
    for i in range(len(word_counts.index)):
        if letter in list(word_counts['word'][i]):
            p_l_w[i] = 1
        else:
            p_l_w[i] = 0

    return p_l_w
```

In [31]:

```
def best_guess(correct, incorrect):
    '''
    Gives the best next guess l and P(L=l|E)

    correct: list of correct guesses in order
    incorrect: list of incorrect guesssses
    '''

    # Compute P(E|W=w)
    word_counts['P(E|W=w)'] = p_e_given_w(correct, incorrect)

    # Compute P(W=w|E)
    word_counts['P(E|W=w)*P(W=w)'] = word_counts['P(E|W=w)'] * word_coun
ts['P(W=w)']
    word_counts['P(W=w|E)'] = word_counts['P(E|W=w)*P(W=w)'] / word_coun
ts['P(E|W=w)*P(W=w)'].sum(axis=0)
    del word_counts['P(E|W=w)*P(W=w)']

    # Compute P(L=l|W=w)
    for letter in string.ascii_uppercase:
        word_counts['P(L={}|W=w)'.format(letter)] = p_l_given_w(letter)

    # Compute P(L=l|W=w)*P(W=w|E) for each letter and word
    for letter in string.ascii_uppercase:
        word_counts['P(L={}|W=w)*P(W=w|E)'.format(letter)] = p_l_given_w
(letter) * word_counts['P(W=w|E)']

    # Compute P(L=l|E)
    p_l_e = [None]*len(string.ascii_lowercase)
    for letter, i in zip(string.ascii_uppercase, range(len(string.ascii_
lowercase))):
        p_l_e[i] = word_counts['P(L={}|W=w)*P(W=w|E)'.format(letter)].su
m(axis=0)

    p_letter_e = pd.DataFrame(p_l_e, columns=['P(L=l|E)'], index=list(st
ring.ascii_uppercase))

    best_guess = p_letter_e.loc[p_letter_e[p_letter_e['P(L=l|E)'] < 0.99
99999].idxmax()].index.values[0]
    max_p_l_e = p_letter_e.loc[p_letter_e[p_letter_e['P(L=l|E)'] < 0.999
9999].idxmax()].values[0][0]

    print('For correct guesses', correct, 'and incorrect guesses {}:'.fo
rmat(incorrect))
    print('Your best next guess is', best_guess, 'with a probability P(L
={}|E) of'.format(best_guess),round(max_p_l_e, 4),'\n')
```

In [32]:

```
# Check against given solutions
correct = [None] * 5
incorrect = ['E', 'O']
best_guess(correct, incorrect)

correct = ['D', None, None, 'I', None]
incorrect = []
best_guess(correct, incorrect)

incorrect = ['A']
best_guess(correct, incorrect)

correct = [None, 'U', None, None, None]
incorrect = ['A', 'E', 'I', 'O', 'S']
best_guess(correct, incorrect)
```

```
For correct guesses [None, None, None, None, None] and incorrect guesse
s ['E', 'O']:
Your best next guess is I with a probability P(L=I|E) of 0.6366

For correct guesses ['D', None, None, 'I', None] and incorrect guesses
[]:
Your best next guess is A with a probability P(L=A|E) of 0.8207

For correct guesses ['D', None, None, 'I', None] and incorrect guesses
['A']:
Your best next guess is E with a probability P(L=E|E) of 0.7521

For correct guesses [None, 'U', None, None, None] and incorrect guesses
['A', 'E', 'I', 'O', 'S']:
Your best next guess is Y with a probability P(L=Y|E) of 0.627
```

In [33]:
```python
# New solutions
correct = [None] * 5
incorrect = []
best_guess(correct, incorrect)

incorrect = ['A','I']
best_guess(correct, incorrect)

correct = ['A', None, None, None, 'R']
incorrect = []
best_guess(correct, incorrect)

incorrect = ['E']
best_guess(correct, incorrect)

correct = [None, None, 'U', None, None]
incorrect = ['O', 'D', 'L', 'C']
best_guess(correct, incorrect)
```

```
For correct guesses [None, None, None, None, None] and incorrect guesse
s []:
Your best next guess is E with a probability P(L=E|E) of 0.5394

For correct guesses [None, None, None, None, None] and incorrect guesse
s ['A', 'I']:
Your best next guess is E with a probability P(L=E|E) of 0.6214

For correct guesses ['A', None, None, None, 'R'] and incorrect guesses
[]:
Your best next guess is T with a probability P(L=T|E) of 0.9816

For correct guesses ['A', None, None, None, 'R'] and incorrect guesses
['E']:
Your best next guess is O with a probability P(L=O|E) of 0.9913

For correct guesses [None, None, 'U', None, None] and incorrect guesses
['O', 'D', 'L', 'C']:
Your best next guess is T with a probability P(L=T|E) of 0.7045
```