CSE 250A HW1

Jiping Lin A15058075

1. (a) $LHS = P(X,Y|E) = \dfrac{P(X,Y,E)}{P(E)}$; $RHS = P(X|Y,E)P(Y|E)$

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

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

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

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

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

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

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

(c) $LHS = P(X|E) = \dfrac{P(X,E)}{P(E)}$, $RHS = \sum_y P(X,Y=y|E) = \sum_y \dfrac{P(X,Y=y,E)}{P(E)} = \dfrac{\sum_y P(X,Y=y,E)}{P(E)}$

So $LHS = RHS \Leftrightarrow P(X,E) = \sum_y P(X,Y=y,E)$,

where $\sum_y P(X,Y=y,E) = \sum_y P(X)P(E|X)P(Y=y|X,E)$ 「Product Rule」

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

$= P(X)P(E|X)$ 「Since $\sum_y P(Y=y|X,E)$ sums over all y given X,E so it is 1」

$= P(X,E)$.

2. $(1) \Rightarrow (2)$:

$(1) \Rightarrow P(X|E) = \dfrac{P(X,Y|E)}{P(Y|E)} = \dfrac{P(X|Y,E)P(Y|E)}{P(Y|E)}$ 「by 1(a)」

$= P(X|Y,E)$

$(2) \Rightarrow (3)$:

$(2) \Rightarrow P(X|Y,E)P(Y|E) = P(X|E)P(Y|E)$

$= P(X,Y|E)$ 「by 1(a)」

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

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

$(3) \Rightarrow (1)$:

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

$= P(Y,X|E)$ 「by 1(a)」 $= P(X,Y|E)$

3. (a) Let X denote whether a driver got pulled over,

Y denote whether the driver is drunk,

Z denote whether the driver is speeding.

then $P(X=1) < P(X=1|\underset{\text{Drunk}}{Y=1}) < P(X=1|\underbrace{Y=1, Z=1}_{\text{Drunk and speed}})$

(b) Keep X, Y same, change Z to whether he is driving in local

Assuming more police on freeway, we have

$$P(X=1|Y=1,Z=1) < P(X=1|Y=1)$$

(c) Suppose Abby and Bob live in the same town but do not know each other and have no interaction but working in the same place, let X be Abby arrives late for work, Y be Bob arrives late for work.

Then X, Y are not independent since if $X=1$ (Abby late), the cause for it might also affect Y (e.g. traffic or weather), let Z be whether the work is remote, then if $Z=1$, then X and Y are conditionally independent given $Z=1$, since $Z=1$ rules out all possibilities that can cause both $X=1$ and $Y=1$.

4. (a)

$$D \longrightarrow T$$

$P(D=0)=0.99$

$P(D=1)=0.01$

D	P(T=1 | D)
0	0.05
1	0.9

(b) We want $P(D=0 \mid T=0)$

Note
$$\begin{aligned} P(T=0) &= P(T=0, D=0) + P(T=0, D=1) \\ &= P(T=0 \mid D=0)P(D=0) + P(T=0 \mid D=1)P(D=1) \\ &= (1-0.05)0.99 + (1-0.9)0.01 \\ &= 0.9415, \end{aligned}$$

By Baye's Rule, we have
$$\begin{aligned} P(D=0 \mid T=0) &= \frac{P(T=0 \mid D=0)P(D=0)}{P(T=0)} \\ &= \frac{(1-0.05)\,0.99}{0.9415} \\ &= 99.89\% \end{aligned}$$

(c) We want $P(D=1 \mid T=1)$,
$$\begin{aligned} P(D=1 \mid T=1) &= \frac{P(T=1 \mid D=1)P(D=1)}{P(T=1)} \\ &= \frac{0.9 \times 0.01}{1-0.9415} = 15.38\% \end{aligned}$$

5. (a) Write $L(p_1, \cdots, p_n, \lambda) = -\sum_{i=1}^{n} p_i \log p_i + \lambda\left(\sum_{i=1}^{n} p_i - 1\right)$,

then we want
$$\frac{\partial L}{\partial p_i} = -\left(p_i \cdot \frac{1}{p_i} + \log p_i\right) + \lambda = 0, \quad \forall i = 1, \cdots, n,$$

and
$$\frac{\partial L}{\partial \lambda} = \sum_{i=1}^{n} p_i - 1 = 0 \iff \sum_{i=1}^{n} p_i = 1$$

we have $\lambda - 1 - \log p_i = 0 \Rightarrow \log p_i = \lambda - 1 \ \forall i = 1, \cdots, n$.

this implies $p_1 = \cdots = p_n$, since $\sum_{i=1}^{n} p_i = 1$,

then we must have $p_i = \frac{1}{n}$, $\forall i$.

(b) We first prove for any X_1, X_2, $X_1 \perp\!\!\!\perp X_2$, $H(X_1X_2) = H(X_1) + H(X_2)$:

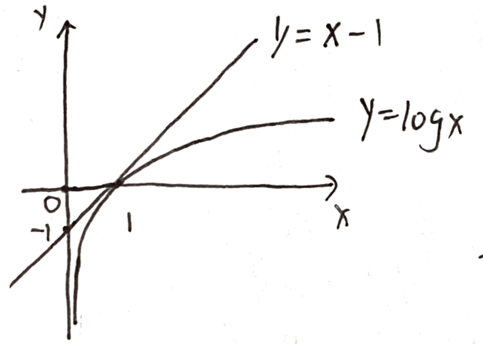
We have
$$
\begin{aligned}
H(X_1,X_2) &= -\sum_{x_1}\sum_{x_2} P(X_1,X_2)\log P(X_1,X_2) \\
&= -\sum_{x_1}\sum_{x_2} P(X_1)P(X_2)(\log P(X_1) + \log P(X_2)) \\
&= -\sum_{x_1}\sum_{x_2} P(X_1)P(X_2)\log P(X_1) - \sum_{x_1}\sum_{x_2} P(X_1)P(X_2)\log P(X_2) \\
&= -\sum_{x_1} P(X_1)\log P(X_1) \sum_{x_2} P(X_2) - \sum_{x_1} P(X_1) \sum_{x_2} P(X_2)\log P(X_2) \\
&= -\sum_{x_1} P(X_1)\log P(X_1) - \sum_{x_2} P(X_2)\log P(X_2) \\
&= H(X_1) + H(X_2).
\end{aligned}
$$

If we have $X_1, \cdots, X_n$ which are mutually independent,

then $H(X_1 \cdots X_n) = H((X_1 \cdots X_{n-1})X_n) = H(X_1 \cdots X_{n-1}) + H(X_n)$

$= \cdots = H(X_1) + \cdots + H(X_n)$ by induction.

6.(a)



From the graph we can see that $x-1 \geq \log x$, $\forall x > 0$, and $x - 1 = \log x \iff x = 1$.

Alternatively,

$\frac{\partial}{\partial x}[\log x - (x-1)] = \frac{1}{x} - 1 = 0 \iff x = 1$,

and $\frac{\partial^2}{\partial x}\Big|_{x=1} = -\frac{1}{x^2}\Big|_{x=1} = -1 < 0$,

so $x=1$ is the global min of $\log x - (x-1)$,

and since $x = 1 \iff \log x - (x-1) = 0$,

then $\log x \leq x - 1$ $\forall x$ and "=" holds only when $x=1$.

(b)
$$
\begin{aligned}
KL(p,q) &= \sum_i p_i \log\left(\frac{p_i}{q_i}\right) \\
&= \sum_i p_i \log\left(\left(\frac{q_i}{p_i}\right)^{-1}\right) \\
&= -\sum_i p_i \log\left(\frac{q_i}{p_i}\right) \qquad \left\ulcorner \log\frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1 \ (*) \right\lrcorner \\
&\geq -\sum_i p_i\left(\frac{q_i}{p_i} - 1\right) \quad (**) \\
&= -\sum_i (q_i - p_i) = -\sum_i q_i + \sum_i p_i = -1 + 1 = 0
\end{aligned}
$$

For equality holds in $(**)$, we must have equality holds in $(*)$.

By (a), this is true iff $\frac{q_i}{p_i} = 1$, i.e. $p_i = q_i$, $\forall i$.

(c) $KL(p,q) = -\sum_i p_i \log\left(\frac{q_i}{p_i}\right) = -\sum_i p_i 2\log\sqrt{\frac{q_i}{p_i}} = -2\sum_i p_i \log\sqrt{\frac{q_i}{p_i}} \geq -2\sum_i p_i\left(\sqrt{\frac{q_i}{p_i}} - 1\right)$

then $KL(p,q) \geq -2\sum_i \sqrt{p_i q_i} - p_i = -2\sum_i \sqrt{p_i q_i} + 2\sum_i p_i = 2 - 2\sum_i \sqrt{p_i q_i}$,

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

So $KL(p,q) \geq \sum_i (\sqrt{p_i} - \sqrt{q_i})^2$

(d) Let $p_1 = 0.8 \quad q_1 = 0.4$,
$p_2 = 0.2 \quad q_2 = 0.6$

then $KL(P, q) = 0.8 \log 2 + 0.2 \log \frac{1}{3} = 0.335$

$KL(q, P) = 0.4 \log \frac{1}{2} + 0.6 \log 3 = 0.382$

7. (a) Fix x to be a possible value of X,

then write $f_x(y) = \sum_y P(x,y) \log\left(\frac{P(x,y)}{P(x)P(y)}\right)$,

then $f_x(y) = \sum_y P(y|X=x) \log\left(\frac{P(y|X=x)}{c\,P(y)}\right)$ $\ulcorner P(x) = c > 0 \lrcorner$,

$\geq \sum_y P(y|X=x) \log\left(\frac{P(y|X=x)}{P(y)}\right)$ since $c \leq 1$,

$= KL(y|X=x, y) \geq 0$ by 6(b)

then $I(X,Y) = \sum_x f_x(y) \geq 0$.

(b) If $I(X,Y) = 0$, then $f_x(y) = 0$, $\forall x$, that means

$KL(y|X=x, y) = 0$, $\forall x$. By 6(b), $KL(y|X=x, y) = 0 \iff (y|X=x) = y$, $\forall x$.

So this implies $X \perp\!\!\!\perp Y$.

If $X \perp\!\!\!\perp Y$, then $P(x,y) = P(x)P(y) \Rightarrow I(X,Y) = \sum_x \sum_y P(x,y) \log 1 = 0$.

8. (a) First BN implies conditional independence of Y, Z given X;
while in second BN, Y, Z are dependent given (or not given) X.

(b) Second BN implies marginal independence of X, Z, but
in third BN, X, Z are also marginal independent.
So Second BN doesn't contain the desired independence.

(c) Third BN implies marginal independence of X, Z, while in
First BN, X, Z are dependent.

9. (a) Most: three seven eight would about,
their which after first fifty,
other forty years there sixty.

Least: troup ottis mapco caixa bosak,
yalom tocor serna paxon niaid,
foumy fabri cleft ccair

these make sense since top part words are frequently seen,
and I know nothing of the bot part words

(b) Missing parts are

E , 0.5394
O , 0.5340
E , 0.7715
E , 0.7127
R , 0.7454.

HW 1 Code

Jiping Lin A15058075

October 5, 2021

[1]:
```python
# CSE 250 HW1
# Jiping Lin A15058075
import string


def load_data():
    word_dic = {}
    with open("hw1_word_counts_05.txt", "r") as f:
        for line in f.readlines():
            line = line.strip().split(" ")
            word_dic[line[0]] = line[1]
    return word_dic


dic = load_data()


def create_prob_table():
    total = 0
    for count in dic.values():
        total += int(count)
    table = {}
    for key in dic.keys():
        table[key] = int(dic[key]) / total
    return table


prior_prob = create_prob_table()


# Print the most frequent/ least frequent words
def print_most(num):
    dic_sorted = sorted(dic.items(), key=lambda x: int(x[1]), reverse=True)
    for i in range(num):
        print(dic_sorted[i][0])
```

```python
def print_least(num):
    dic_sorted = sorted(dic.items(), key=lambda x: int(x[1]), reverse=True)
    for i in range(1, num + 1):
        print(dic_sorted[-i][0])


# return P(W=w)
def prob_w(word: str):
    # Prior Probability table
    return prior_prob[word]


class State:

    def __init__(self, content=None, out=None):
        if out is None:
            out = {}
            for k in range(5):
                out[k] = set()
        if content is None:
            content = [None] * 5

        self.content = content
        self.out = out

    def add_correct(self, letter: str, pos):
        if pos < 0 or pos > 4:
            return
        self.content[pos] = letter

    def add_false(self, letter):
        for i in range(5):
            self.out[i].add(letter)

    # P(E/W=w)
    def prob_ew(self, word: str) -> int:
        word = word.upper()
        appear_set = set()
        for i in range(len(self.content)):
            if self.content[i] is not None:
                appear_set.add(self.content[i])
        for i in range(len(self.content)):
            if self.content[i] is None:
                if word[i] in self.out[i] or word[i] in appear_set:
                    return 0
                else:
                    continue
```

```python
            if self.content[i] != word[i]:
                return 0
        return 1

    def get_bot(self):
        bot = 0
        for key in dic.keys():
            bot += self.prob_ew(key) * prob_w(key)
        return bot

    def posterior(self, word: str) -> float:
        bot = self.get_bot()
        pe = self.prob_ew(word)
        pw = prob_w(word)
        return pe * pw / bot

    def predictive_first(self, letter: str, word: str) -> int:
        for i in range(len(self.content)):
            if self.content[i] is None:
                if word[i] == letter:
                    return 1
        return 0

    def predictive(self, letter: str):
        prob = 0
        bot = self.get_bot()
        for key in dic.keys():
            pe = self.prob_ew(key)
            pw = prob_w(key)
            prob += self.predictive_first(letter, key) * pe * pw / bot
        return prob

    def get_next_guess(self):
        result = {}
        letters = []
        for let in string.ascii_uppercase:
            letters.append(let)
        for i in range(len(letters)):
            add = letters[i]
            result[add] = self.predictive(add)
        result_sorted = sorted(result.items(), key=lambda x: x[1], reverse=True)
        index = 0
        return result_sorted[index]


def print_guess(correct: list, false: set):
    current = State()
```

```python
    for i in range(len(correct)):
        if correct[i] is not None:
            current.add_correct(correct[i], i)
    for i in false:
        if i is not None:
            current.add_false(i)
    print(current.get_next_guess())
```

[2]:

```python
def main():
    letters = []
    for let in string.ascii_uppercase:
        letters.append(let)
    # 1.9a
    print_most(15)
    print()
    print_least(14)
    # 1.9b
    # First row
    print_guess([None, None, None, None, None], set())
    # Second row
    print_guess([None, None, None, None, None], {'E', 'A'})
    # Third row
    print_guess(['A', None, None, None, 'S'], set())
    # Fourth row
    print_guess(['A', None, None, None, 'S'], {'I'})
    # Fifth row
    print_guess([None, None, 'O', None, None], {'A', 'E', 'M', 'N', 'T'})
    # Sixth row
    print_guess([None, None, None, None, None], {'E', 'O'})
    # Seventh row
    print_guess(['D', None, None, 'I', None], set())
    # Eighth row
    print_guess(['D', None, None, 'I', None], {'A'})
    # Ninth row
    print_guess([None, 'U', None, None, None], {'A', 'E', 'I', 'O', 'S'})


if __name__ == '__main__':
    main()
```

```
THREE
SEVEN
EIGHT
WOULD
ABOUT
THEIR
WHICH
```

```
AFTER
FIRST
FIFTY
OTHER
FORTY
YEARS
THERE
SIXTY

TROUP
OTTIS
MAPCO
CAIXA
BOSAK
YALOM
TOCOR
SERNA
PAXON
NIAID
FOAMY
FABRI
CLEFT
CCAIR
('E', 0.5394172389647948)
('O', 0.5340315651557679)
('E', 0.7715371621621622)
('E', 0.7127008416220354)
('R', 0.7453866259829711)
('I', 0.6365554141009618)
('A', 0.8206845238095241)
('E', 0.7520746887966806)
('Y', 0.6269651101630528)
```

[ ]: