4-1 Maximum likehood estimation in belief networks

(a)

$$P_{ML}(X_1 = x) = \frac{\text{count}(x_1 = x)}{T}$$

$$P_{ML}(X_{i+1} = x' \mid X_i = x) = \frac{\text{count}(X_{i+1} = x', X_i = x)}{\text{count}(X_i = x)}$$

(b)

$$P_{ML}(X_n = x) = \frac{\text{count}(X_n = x)}{T}$$

$$P_{ML}(X_i = x \mid X_{i+1} = x') = \frac{\text{count}(X_{i+1} = x', X_i = x)}{\text{count}(X_{i+1} = x')}$$

(c)

$$P_1^{ML}(X_1 = x_1, X_2 = x_2 \dots X_n = x_n) = P_{ML}(X_1 = x_1) \prod_{i=1}^{n-1} P_{ML}(X_{i+1} = x_{i+1} \mid X_i = x_i)$$

$$= \frac{\text{count}(X_1 = x_1)}{T} \prod_{i=1}^{n-1} \frac{\text{count}(X_{i+1} = x_{i+1}, X_i = x_i)}{\text{count}(X_i = x_i)}$$

$$= \frac{1}{T} \prod_{i=1}^{n-1} \text{count}(X_{i+1} = x_{i+1}, X_i = x_i) \cdot \prod_{i=1}^{n-1} \frac{\text{count}(X_1 = x_1)}{\text{count}(X_i = x_i)}$$

$$= \frac{1}{T} \prod_{i=1}^{n-1} \text{count}(X_{i+1} = x_{i+1}, X_i = x_i) \cdot \prod_{i=2}^{n-1} \frac{1}{\text{count}(X_i = x_i)} \quad \#$$

$$P_2^{ML}(X_1 = x_1, X_2 = x_2 \dots X_n = x_n) = P^{ML}(X_n = x_n) \prod_{i=1}^{n-1} P^{ML}(X_i = x_i \mid X_{i+1} = x_{i+1})$$

$$= \frac{\text{count}(X_n = x_n)}{T} \prod_{i=1}^{n-1} \frac{\text{count}(X_{i+1} = x_{i+1}, X_i = x_i)}{\text{count}(X_{i+1} = x_{i+1})}$$

$$= \frac{1}{T} \prod_{i=1}^{n-1} \text{count}(X_{i+1} = x_{i+1}, X_i = x_i) \cdot \prod_{i=1}^{n-1} \frac{\text{count}(X_n = x_n)}{\text{count}(X_{i+1} = x_{i+1})}$$

$$= \frac{1}{T} \prod_{i=1}^{n-1} \text{count}(X_{i+1} = x_{i+1}, X_i = x_i) \prod_{i=2}^{n-1} \frac{1}{\text{count}(X_i = x_i)} \quad \#$$

(d) The maximum likehood CPT for G_3 would not give rise the same joint distribution. Note that X_3 in G_3 has two parents X_2 & X_4. Hence $P(X_2 = x_2 \mid X_3 = x_3, X_4 = x_4)$ not necessarily equal to $P(X_2 = x_2 \mid X_3 = x_3)$.

4-2 Statistical Learning model

(a)

```
MILLION 0.002072759168154815
MORE 0.0017088989966186725
MR. 0.0014416083492816956
MOST 0.0007879173033190295
MARKET 0.0007803712804681068
MAY 0.0007298973156289532
M. 0.0007034067394618568
MANY 0.0006967290595970209
MADE 0.0005598610827336895
MUCH 0.0005145971758110562
MAKE 0.0005144626437991272
MONTH 0.00044490959363187093
MONEY 0.00043710673693999306
MONTHS 0.00040576077781605526
MY 0.0004003183467688823
MONDAY 0.00038198530259784006
MAJOR 0.00037089252670515475
MILITARY 0.00035204581485220204
MEMBERS 0.00033606096579846475
MIGHT 0.00027358919153183117
MEETING 0.0002657374141083427
MUST 0.0002665079156312084
ME 0.0002635726717345725
MARCH 0.0002597935452176646
MAN 0.0002528834918776787
MS. 0.0002389900041002911
MINISTER 0.00023977273580605944
MAKING 0.00021170446604452378
MOVE 0.0002099555498894477
MILES 0.00020596851026319035
```

(b)

```
('THE', '<UNK>'): 0.6150198100055118
('THE', 'U.'): 0.013372499432610317
('THE', 'FIRST'): 0.011720260675031612
('THE', 'COMPANY'): 0.011658788055636611
('THE', 'NEW'): 0.009451480076516552
('THE', 'UNITED'): 0.008672308141231398
('THE', 'GOVERNMENT'): 0.006803488635995202
('THE', 'NINETEEN'): 0.006650714911000876
('THE', 'SAME'): 0.006287066757449016
('THE', 'TWO'): 0.006160749602827221
```

(c)

```
Log Likelihood of the unigram model of sentence1 -64.50944034364878
Log Likelihood of the bigram model of sentence1 -40.91813213378977
```

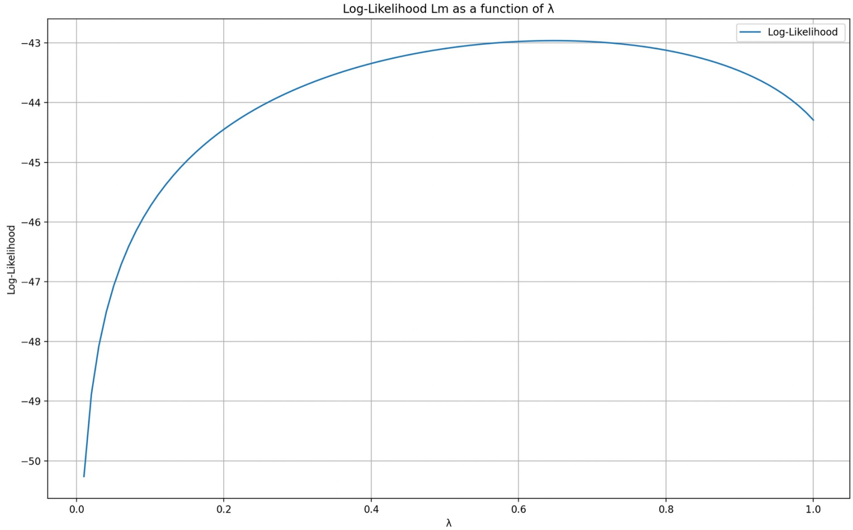
Since $L_u < L_b$, the bigram yields the higher log-likelihood

(d)

```
Log Likelihood of the unigram model of sentence2 -44.291934473132606
Log Likelihood of the bigram model of sentence2 = -inf
```

Since $L_u > L_b$, the unigram yields the higher log-likelihood
The bigrams "SIXTEEN OFFICIALS" and "SOLD FIRE" are not observed in the training set

(e)



The optimal value is $\lambda = 0.646$, log-likelihood value $= -42.964$

```python
import numpy as np
import matplotlib.pyplot as plt
##read data from txt.file
def load_file(file_voc,file_uni,file_big):
    with open(file_voc, 'r') as file:
        vocb= file.readlines()
    with open(file_uni, 'r') as file:
        unigram= file.readlines()
    with open(file_big, 'r') as file:
        bigram= file.readlines()
    return vocb,unigram,bigram

## P_u(w) from letters start with M or m
def unigram_M(vocb,unigram):
    vocbs=[v.strip() for v in vocb]
    unigrams=np.array([int(u.strip()) for u in unigram])
    prob_u={}

    for i in range(len(vocb)):
        prob_u[vocbs[i]]=unigrams[i]/np.sum(unigrams)
        if(vocbs[i][0]=="M" or vocbs[i][0]=="m"):
            print(vocbs[i]+ " " +str(prob_u[vocbs[i]]))
    return prob_u

## Print out the most likely words to follow the word "The"
import numpy as np

def bigram_The(bigram):
    prob_b = {}
    the_bigrams = {}
    partial_sums = {}
    vocbs = [v.strip() for v in vocb]
    bigrams = [b.strip().split('\t') for b in bigram]
    bigrams = np.array(bigrams).astype(np.uint32)

    for b in bigrams:
        index1 = int(b[0])
        index2 = int(b[1])
        if index1 not in partial_sums:
            partial_sums[index1] = np.sum(bigrams[bigrams[:, 0] == index1, 2])
        key = (vocbs[index1 - 1], vocbs[index2 - 1])
        value = int(b[2])
        prob_b[key] = value / partial_sums[index1]


    the_bigrams = {k: v for k, v in prob_b.items() if k[0] == "THE"}
```

```python
    sorted_the_bigrams = sorted(the_bigrams.items(), key=lambda item: item[1], reverse=True)
    for bigram, prob in sorted_the_bigrams[:10]:
        print(f"{bigram}: {prob}")

    return prob_b

## Compute log-likelihood "THE STOCK MARKET FELL BY ONE HUNDRED POINTS LAST WEEK"
def log_likelihood_u1(prob_u):
    sentence1="THE STOCK MARKET FELL BY ONE HUNDRED POINTS LAST WEEK"
    sentence1=sentence1.split(" ")
    L1=0
    for s in sentence1:
        if s in prob_u:
            L1 += np.log(prob_u[s])
    print("Log Likelihood of the unigram model of sentence1",L1)
    return L1
def log_likelihood_b1(prob_b):
    sentence1="<s> THE STOCK MARKET FELL BY ONE HUNDRED POINTS LAST WEEK"
    sentence1=sentence1.split(" ")
    L2=0
    for i in range(len(sentence1) - 1):
        bi = (sentence1[i], sentence1[i + 1])
        if bi in prob_b:
            L2+=np.log(prob_b[bi])
    print("Log Likelihood of the bigram model of sentence1",L2)

## Compute log-likelihood "THE SIXTEEN OFFICIALS SOLD FIRE INSURANCE"
def log_likelihood_u2(prob_u):
    sentence2="THE SIXTEEN OFFICIALS SOLD FIRE INSURANCE"
    sentence2=sentence2.split(" ")
    L3=0
    for s in sentence2:
        if s in prob_u:
            L3 += np.log(prob_u[s])
    print("Log Likelihood of the unigram model of sentence2",L3)
def log_likelihood_b2(prob_b):
    sentence2="<s> THE SIXTEEN OFFICIALS SOLD FIRE INSURANCE"
    sentence2=sentence2.split(" ")
    L4=0
    for i in range(len(sentence2) - 1):
        bi_2 = (sentence2[i], sentence2[i + 1])
        try:
            L4+=np.log(prob_b[bi_2])
        except:
            print("Log Likelihood of the bigram model of sentence2 = -inf")
def mixture_model(prob_u,prob_b,lam):
```

```
    L5= 0
    sentence3= "<s> THE SIXTEEN OFFICIALS SOLD FIRE INSURANCE"
    sentence3=sentence3.split(" ")
    for i in range(len(sentence3) - 1):
        word=sentence3[i + 1]
        pre_word=sentence3[i]
        m1= (pre_word, word)
        if m1 in prob_b:
            prob_m = lam * prob_u[word] + (1 - lam) * prob_b[m1]
        else:
            prob_m = lam * prob_u[word]

        L5 += np.log(prob_m)

    return L5

vocb, unigram, bigram = load_file('hw4_vocab.txt', 'hw4_unigram.txt', 'hw4_bigram.txt')
prob_u = unigram_M(vocb, unigram)
prob_b=bigram_The(bigram)
u1=log_likelihood_u1(prob_u)
b1=log_likelihood_b1(prob_b)
u2=log_likelihood_u2(prob_u)
b2=log_likelihood_b2(prob_b)


lambda_values = np.linspace(0, 1, 100)
log_likelihoods = [mixture_model(prob_u, prob_b, lam) for lam in lambda_values]
optimal_lam=lambda_values[np.argmax(log_likelihoods)]
print(np.max(log_likelihoods))
print(optimal_lam)
plt.plot(lambda_values, log_likelihoods, label="Log-Likelihood ")
plt.xlabel("")
plt.ylabel("Log-Likelihood")
plt.title("Log-Likelihood Lm as a function of ")
plt.legend()
plt.grid()
plt.show()
```

4-3 Markov modeling

(a) Unigram model

T	a	b	c	d
$P_1(T)$	0.25	0.25	0.25	0.25

$a: \frac{4}{16} = 0.25$

$b: \frac{4}{16} = 0.25$

$c: \frac{4}{16} = 0.25$

$d: \frac{4}{16} = 0.25$

(b) Bigram model

T'

T	$P_2(T'\|T)$	a	b	c	d
	a	$\frac{1}{2}$	$\frac{1}{4}$	0	$\frac{1}{4}$
	b	0	$\frac{3}{4}$	$\frac{1}{4}$	0
	c	0	0	$\frac{2}{3}$	$\frac{1}{3}$
	d	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{2}$

T T'	T T'
(a, a)	(d, d)
(a, b)	(d, a)
(b, b)	(a, a)
(b, b)	(a, d)
(b, b)	(d, d)
(b, c)	(d, c)
(c, c)	(c, c)
(c, d)	

(c) Likelihoods

(1)

$P_U(S) = P_U(T_1)$

(2)

$P_U(S) = P_U(T_2)$

(3)

$P_U(S) = P_U(T_3)$

(4)

$P_B(T_1) < P_B(S)$

(5)

$P_B(T_2) < P_B(S)$

(6)

$P_B(T_3) = P_B(T_2)$

(7)

$P_U(S) < P_B(S)$

(8)

$P_U(T_1) = P_B(T_1)$

(9)

$P_U(T_2) > P_B(T_2)$

(10)

$P_U(T_3) > P_B(T_3)$

(d)

S = "aabbbbccddaaddcc" => D

T_1 = "adadadadadadadad" => A

T_2 = "aaaaddddcccc<u>c</u>bbbb" => C

T_3 = "<u>bdbdbdbdbdbdbdbd</u>" => B