Artificial Intelligence

Homework 3

Ning Ma

3.1

$$P(Y_1, Y_2, Y_3|X) = P(Y_1|X)P(Y_2|X)P(Y_3|X) \tag{1}$$

$$P(Z_1|Y_1, Y_2, Y_3) = P(Z_1|Y_1, Y_2) \tag{2}$$

$$P(Z_2|Y_1, Y_2, Y_3) = P(Z_2|Y_2, Y_3) \tag{3}$$

Thus, the CPTs for the polytree is the following:

Table 1: CPTs for the polytree

Y_1	Y_2	Y_3	Y	$P(Y\|X=0)$	$P(Y\|X=1)$	$P(Z_1=1\|Y)$	$P(Z_2=1\|Y)$
0	0	0	1	0.021	0.08	0.8	0.1
1	0	0	2	0.049	0.08	0.7	0.1
0	1	0	3	0.009	0.32	0.6	0.2
0	0	1	4	0.189	0.02	0.8	0.3
1	1	0	5	0.021	0.32	0.5	0.2
1	0	1	6	0.441	0.02	0.7	0.3
0	1	1	7	0.081	0.08	0.6	0.4
1	1	1	8	0.189	0.08	0.5	0.4

3.2

(a)

$$P(X) = \frac{C(x)}{T} \tag{4}$$

$$P(Y|X) = \frac{C(x,y)}{C(x)} \tag{5}$$

$$P(Z|Y) = \frac{C(x,y)}{C(y)} \tag{6}$$

(b)

$$P(Z) = \frac{C(z)}{T} \tag{7}$$

$$P(Y|Z) = \frac{C(y,z)}{C(z)} \tag{8}$$

$$P(X|Y) = \frac{C(x,y)}{C(y)} \tag{9}$$

(c) For left DAG, we have

$$P_l(X,Y,Z) = P(X)P(Y|X)P(Z|Y) = \frac{C(x,y)C(y,z)}{TC(y)} \tag{10}$$

For the right DAG, we have

$$P_r(X,Y,Z) = P(Z)P(Y|Z)P(X|Y) = \frac{C(y,z)C(x,y)}{TC(y)} = P_l(X,Y,Z) \tag{11}$$

Thus, different DAGs give the same joint distribution

(d) No. The only conditional independent relation is $P(X|Y,Z) = P(X|Y)$. This relation is implied by both DAGs.

3.3

(a) See the table printed out.

MILLION	0.002072759
MORE	0.001708899
MR.	0.001441608
MOST	0.000787917
MARKET	0.000780371
MAY	0.000729897
M.	0.000703407
MANY	0.000696729
MADE	0.000559861
MUCH	0.000514597
MAKE	0.000514463
MONTH	0.000444910
MONEY	0.000437107
MONTHS	0.000405761
MY	0.000400318

MONDAY	0.000381985
MAJOR	0.000370893
MILITARY	0.000352046
MEMBERS	0.000336061
MIGHT	0.000273589
MEETING	0.000265737
MUST	0.000266508
ME	0.000263573
MARCH	0.000259794
MAN	0.000252883
MS.	0.000238990
MINISTER	0.000239773
MAKING	0.000211704
MOVE	0.000209956
MILES	0.000205969

(b) See the table printed out.

HUNDRED	0.209060511
<UNK>	0.124304048
.POINT	0.099952347
OF	0.073947061
THOUSAND	0.068654364
MILLION	0.031831530
,COMMA	0.031621777
-HYPHEN	0.030479336
HALF	0.029139361
.PERIOD	0.024376138

(c) I got the following message from Matlab:
Logrithm of unigram is $L_u = -57.19869441$
Logrithm of bigram is $L_b = -38.09758270$
Thus, the bigram model yields the highest log-likelihood.

(d) I got the following results from Matlab:
Logrithm of unigram is $L_u = -44.23642999$
Pairs of adjacent words OFFICIALS follows FOURTEEN are not observed.
Pairs of adjacent words FIRE follows SOLD are not observed.
Since the condictional probability corresponding to these two adjacent words are zero, L_b goes to negative infinite.

(e) I got the following results from Matlab:
maximum value is -42.95806624 with lemda 0.34100000. Thus, the maximum log-likelihood is -42.95806624 with the λ=0.34. The plot of the log-likelihood as function of λ is

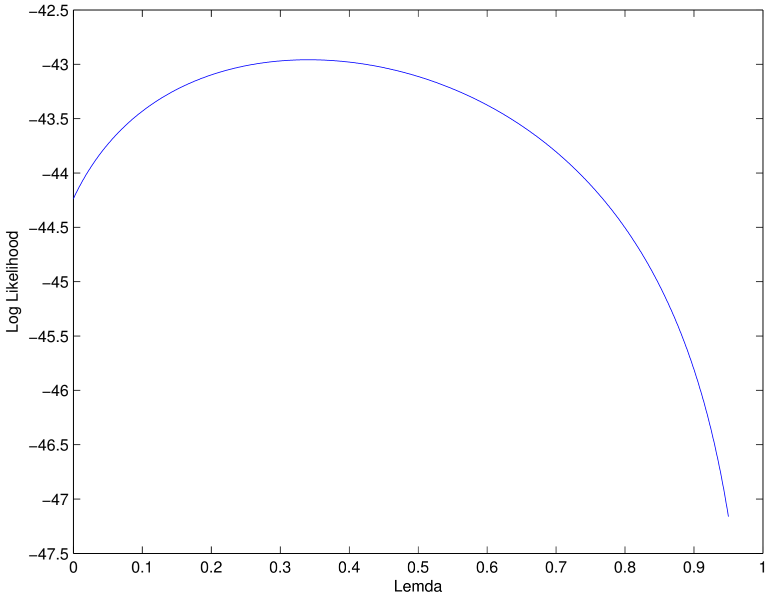

Figure 1: Log-likelihood as fuction of λ

The following is the Matlab source code

```
%------------------The load_data function ---------------------------------------
function [a,b,c]=load_data()
%function [a,b,c]=load_data()
%a will get vocab, b will get unigram, c will get bigram

a = importdata('vocab.txt');%load the words data from txt file
b = importdata('unigram.txt');%load the unigram counts from txt file
c = importdata('bigram.txt');% load the bigram counts form txt file
```

```
%----------------The source code for the problem 3.3(a)
%Compute the maximum likehood estimate of the unigram distribution

[vocab_string,unigramdata,bigramdata] = load_data;%load the words data,
%and bigram counts form txt file.

totalwords = sum(unigramdata);%total words appears in the journal
wordcatagory = size(vocab_string,1);% the number of words in the vocabul
FID=fopen('outputtable_a', 'w+'); %create a file where I will wirte the
% words start with 'M' and their correspoing unigram probability

for i=1:wordcatagory
string_i = vocab_string(i);%check the ith word
char_i = char(string_i);% covert the element of cell to char
if char_i(1) == 'M' % check the first letter of ith word
Pui = unigramdata(i)/totalwords; %calculate the unigram distribution
% of the word starting with letter 'M'
fprintf(FID, '%-10s  %-1.9f \n', char(string_i), Pui);
end
end
fclose(FID);


%----------------The souce code for the problem 3.3(b)--------------------
%Compute the maximum likehood estimate of the bigram distribution

[vocab_string,unigramdata,bigramdata] = load_data;%load the words data,
%and bigram counts form txt file.
FID=fopen('outputtable_b', 'w+');

TF=strcmp('ONE',vocab_string);%find the index for 'ONE' in the vacabular
r=find(TF == 1);
ONE_FIRST=find(bigramdata(:,1) == r);%find the postion(indices) of the w
sum_ONE = sum(bigramdata(ONE_FIRST,3));%the total counts that 'ONE' appr

[sorted_wordnumber,IX] = sort(bigramdata(ONE_FIRST,3),'descend');%sort t
top_10_words = vocab_string(bigramdata(ONE_FIRST(1) - 1 + IX(1:10),2));%
%like words w' to follow 'ONE'
top_10_probability = sorted_wordnumber(1:10,1)/sum_ONE;% calculate the m
%bigram distrituion for the 10 most like words w'
```

```
for i = 1:10
    fprintf(FID, '%-10s  %-1.9f \n', char(top_10_words(i)), top_10_proba
end

fclose(FID);


%----------------The souce code for the problem 3.3(c)-----------------------
[vocab_string,unigramdata,bigramdata] = load_data;%load the words data,
%and bigram counts form txt file.

totalwords = sum(unigramdata);%total words appears in the journal
sentence = {'<s>','THE', 'MARKET', 'FELL', 'BY', 'ONE', 'HUNDRED', 'POINTS', 'L
n = size(sentence,2);
PU = 1;
PB = 1;

%compute Lu
for i = 2:n
    TF = strcmp(sentence(i),vocab_string);
    r = find(TF == 1);%find the index for ith word in the vacabulary
    count_i = unigramdata(r);%counts this word appreas
    pu(i) = count_i/totalwords;
    PU=PU*pu(i);
end
Lu = log(PU);
fprintf('Logrithm of unigram is Lu = %2.8f\n',Lu);

%compute Lb
for i = 1:n-1
    TF = strcmp(sentence(i), vocab_string);%find the ith word in the sen
    ri = find(TF == 1);

    TF = strcmp(sentence(i+1), vocab_string);%find the i+1th word in the
    rj = find(TF == 1);

    word_i = find(bigramdata(:,1) == ri);% the indices of ith word in th
    sum_i = sum(bigramdata(word_i,3));% the total counts that ith words
    indiceij = find(bigramdata(word_i,2) == rj);%find the location where
    if(isempty(indiceij))
            fprintf('Pairs of adjacent words %10s and %10s are not obser
    else
            wordij = bigramdata(word_i(1) - 1 + indiceij,3);%the counts
            Pij=wordij/sum_i;
           %printf('P%d%d = %f',i,i+1,Pij);
           PB=PB*Pij;
    end

end
 Lb = log(PB);
 fprintf('logrithm of bigram is LB = %2.8f\n',Lb);



%————————————The souce code for the problem 3.3(d)————————————————
[vocab_string,unigramdata,bigramdata] = load_data;%load the words data,
%and bigram counts form txt file.

totalwords = sum(unigramdata);%total words appears in the journal
sentence = {'<s>','THE','FOURTEEN','OFFICIALS','SOLD','FIRE','INSURANCE'
n = size(sentence,2);
PU = 1;
PB = 1;

%compute Lu
for i = 2:n
    TF = strcmp(sentence(i),vocab_string);
    r = find(TF == 1);%find the index for ith word in the vacabulary
    count_i = unigramdata(r);%counts this word appreas
    pui = count_i/totalwords;
    PU=PU*pui;
end
Lu = log(PU);
fprintf('Logrithm of unigram is Lu = %2.8f\n',Lu);

%compute Lb
for i = 1:n-1
    TF = strcmp(sentence(i), vocab_string);%find the ith word in the sen
    ri = find(TF == 1);
    TF = strcmp(sentence(i+1), vocab_string);%find the i+1th word in the
    rj = find(TF == 1);

    word_i = find(bigramdata(:,1) == ri);% the indices of ith word in th
    sum_i = sum(bigramdata(word_i,3));% the total counts that ith words
    indiceij = find(bigramdata(word_i,2) == rj);
    %find the location where i+1th word follows ith word
    if(isempty(indiceij))
            fprintf('Pairs of adjacent words %-9s follows %-9s are not o
            Pij=1;%Can't find ith word i followed  by (i+1)th word, so i
    else
            wordij = bigramdata(word_i(1) - 1 + indiceij,3);%the counts
            Pij=wordij/sum_i;
           %size(Pij)
           PB=PB*Pij;
    end

end
 Lb = log(PB);
 fprintf('Logrithm of bigram is Lb = %2.8f\n',Lb);



 %------------------The souce code for the problem 3.3(e)--------------------------
N=1001;
lemda=linspace(0,1,N);
[vocab_string,unigramdata,bigramdata] = load_data;%load the words data,
%and bigram counts form txt file.
totalwords = sum(unigramdata);%total words appears in the journal
sentence = {'<s>','THE','FOURTEEN','OFFICIALS','SOLD','FIRE','INSURANCE'
n = size(sentence,2);
PM=1;
PMLEMDA=zeros(1,N-1);

for k=1:N

    PM=1;
    for i = 1:n-1
```

```
        %compute pu
        TF = strcmp(sentence(i+1),vocab_string);
        r = find(TF == 1);%find the index for i+1 th word in the vacabul
        counti_next = unigramdata(r);%counts this word appreas
        pui_next = counti_next/totalwords;

        %compute pb
        TF = strcmp(sentence(i), vocab_string);%find the ith word in the
        ri = find(TF == 1);

        TF = strcmp(sentence(i+1), vocab_string);%find the i+1th word in
        rj = find(TF == 1);
        word_i = find(bigramdata(:,1) == ri);% the indices of ith word i
        sum_i = sum(bigramdata(word_i,3));% the total counts that ith wo
        indiceij = find(bigramdata(word_i,2) == rj);
        %find the location where i+1th word follows ith word
        if(isempty(indiceij))%if can not find ith and i+1th words as adj
            pbij=0;
        else
            wordij = bigramdata(word_i(1) - 1 + indiceij,3);%the counts
            pbij = wordij/sum_i;
        end

        Pmij=(1-lemda(k))*pui_next+lemda(k)*pbij;
        PM=PM*Pmij;
    end
    PMLEMDA(k)=log(PM);
end
plot(lemda(1:N-50),PMLEMDA(1:N-50));%lemda ranges from 0 to 0.95
[maxvalue,I] = max(PMLEMDA(1:N-50));
fprintf('maximum value is %2.8f with lemda %2.8f', maxvalue, lemda(I));
xlabel('Lemda');
ylabel('Log Likelihood');
```