Shirui Ye

PS5

CS542

Cover

To achieve the best result, I decide to use hand to write down the written assignment part. (Typing in math symbols and equations in LaTex really interrupts my logic)

Shirui Ye

PS5

1.(a) Bishop 8.3

we have 3 binaries a, b, c, we want to show $P(a,b) \neq P(a)P(b)$, but when c coming in, $P(a,b|c) = P(a|c)P(b|c)$ for $c=0$ and $c=1$.

For first part we follow the question and show by direct evaluation

b	a	$p(a,b)$	$p(a)p(b)$
1	0	.264	.244800
0	1	.256	.236800
0	0	.336	.356600
1	1	.144	.162800

$\Rightarrow$ we can see directly from these that a, b are marginally dependent which $p(a,b) \neq P(a)P(b)$

For second part, now we have a condition c, we know that

$$p(a,b|c) = \frac{P(a,b,c)}{\sum_{a\in\{0,1\}}\sum_{b\in\{0,1\}} P(a,b,c)}$$ we can get $P(a|c)$ and $P(b|c)$ the same way

Thus, we can have below:

$$\frac{P(a,b,c)}{\sum_{a\in\{0,1\}}\sum_{b\in\{0,1\}}P(a,b,c)} = \frac{\sum_{b\in\{0,1\}}P(a,b,c)}{\sum_{a\in\{0,1\}}\sum_{b\in\{0,1\}}P(a,b,c)} \; \frac{\sum_{a\in\{0,1\}}P(a,b,c)}{\sum_{a\in\{0,1\}}\sum_{b\in\{0,1\}}P(a,b,c)}$$

$$\Downarrow \qquad\qquad \Downarrow \qquad\qquad \Downarrow$$

$$p(a,b|c) = P(a|c) \quad P(b|c)$$ $\Rightarrow$ following the question

According to table 2, we can tell that when $c=0, c=1$, above equation is valid so that we successfully show when c involved, $P(a,b|c) = P(a|c)P(b|c)$ for $c=0,1$.

Shirui Ye
PS5

1.(b) Bishop 8.4

In this question, we are asked to use Table 8.2 again, we can still show by direct evaluation.

$p(a) = \sum_{b\in\{0,1\}} \sum_{c\in\{0,1\}} p(a,b,c)$ $\quad p(b|c) = \frac{\sum_{a\in\{0,1\}} p(a,b,c)}{\sum_{a\in\{0,1\}} \sum_{b\in\{0,1\}} p(a,b,c)}$ $\quad p(c|a) = \frac{\sum_{b\in\{0,1\}} p(a,b,c)}{\sum_{b\in\{0,1\}} \sum_{c\in\{0,1\}} p(a,b,c)}$

$p(a,b,c) = p(a)\,p(c|a)\,p(b|c)$

a	p(a)
1	0.6
0	0.4

a	c	p(c|a)
0	0	0.4
1	1	0.4
0	1	0.6
1	0	0.6

b	c	p(b|c)
0	0	0.8
1	1	0.6
1	0	0.2
0	1	0.4

Now Let's calculate then compare to information from Table 8.2

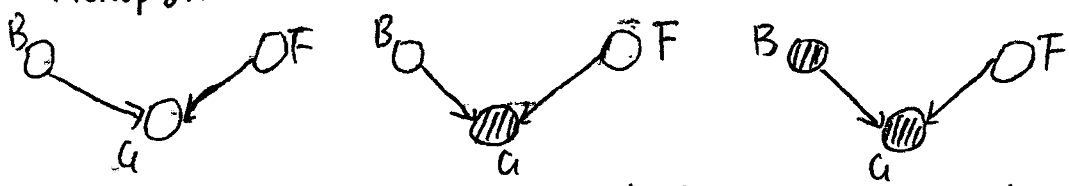
a	b	c	p(a,b,c)	our calculated p(a,b,c)
0	0	0	0.192	0.192
0	0	1	0.144	0.144
0	1	0	0.048	0.048
0	1	1	0.216	0.216
1	0	0	0.192	0.192
1	0	1	0.064	0.064
1	1	0	0.048	0.048
1	1	1	0.096	0.096

$\Rightarrow$ Thus, we prove that $p(a,b,c) = p(a)p(c|a)p(b|c)$

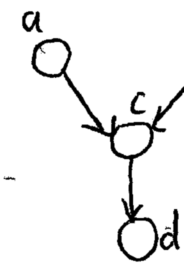
Shirui Ye

PS5

1. (c) Bishop 8.11



This is Figure 8.21 system, instead of observing G directly, we have a D here, so the situation should be like 8.54



First, we need to evaluate the Prob of tank is empty when D=0. Simply use Bayes' Theorem.

$$\text{Prob}(\text{Empty tank} \mid D=0) = \frac{P(D=0 \mid \text{Empty tank})\, P(\text{empty tank})}{P(D=0)}$$

$P(D=0) \rightarrow \sum_{F,G,B} P(D=0|G)\,P(B)\,P(F)\,P(G|B,F) = 0.352$

$P(D=0 \mid \text{Empty tank}) = \sum_{B,G} P(D=0|G)\,P(B)\,P(G|B,\text{Empty tank}) = 0.748$

$\therefore$ overall $\text{Prob}(\text{Empty tank} \mid D=0) = \underline{\underline{0.21}}$

Then we want to calculate when battery is flat which is $B=0$, we still use Bayes' Theorem and can get $\text{Prob}(\text{Empty tank} \mid B=0, D=0) = \underline{\underline{0.11}}$

The <u>intuition behind the results is below</u>:

Firstly 0.21 is less than 0.257 which is from 8.32, this shows that driver is not really dependable, then 0.11 is also less than 0.111 which is from 8.33. Just like what explained in the textbook: "This accords with our intuition that finding out that the battery is flat explains away the observation that the fuel gauge reads empty."

Shirui Ye

PS5

1. (d) Bishop 8.14

We have a particular case based on 8.42 $E(X,y)=h\sum_i x_i-\beta\sum_{\{i,j\}} x_i x_j-\eta\sum_i x_i y_i$ which now $\beta=h=0$. We want to show that the most probable configuration of the latent variables is given by $x_i=y_i$ for all i.

It is pretty clear to show so for this particular case. We know most probable configuration means lowest energy configuration. In 8.42 we know η is a positive constant and we have the special case $\beta=h=0$, then $x_i y_i \in \{-1,+1\}$ it is obtained when $x_i=y_i$ where $i=1,\dots,D$. Thus, we successfully prove that the most probable configuration of the latent variables is given by $x_i=y_i$ for all i.

Shirui Ye

PS5 Programming Report

Source code:

Part A

```
%read in data and preprocess
source=imread('Bayesnoise_textbook.png');
%extract
extract_s=source(:,:,1);
extract_s=int8(extract_s);
%get greyscale for source
[r,c]=size(extract_s);
for i=1:r
    for j=1:c
        if extract_s(i,j)<119
            extract_s(i,j)=-1;
        else
            extract_s(i,j)=1;
        end
    end
end
gc=extract_s;
or=gc;
s=size(gc);
n=7;
yd=s(1);
xd=s(2);
h=-0.01;
fp=1;
c=0;
b=5;
while (fp)
    c=c+1;
    fp=0;
    for i=2:xd-1
        for j=2:yd-1
            fpe=(-gc(j,i))*(h-(b*(gc(j,i+1)+gc(j,i-
1)+gc(j+1,i)+gc(j-1,i)))-(n*gc(j,i)));
            nfpe=gc(j,i)*(h-(b*(gc(j,i+1)+gc(j,i-
1)+gc(j+1,i)+gc(j-1,i)))-(n*gc(j,i)));
            if nfpe>fpe
                gc(j,i)=-gc(j,i);
                fp=1;
            end
```

```
        end
    end
end
%correction read in and process
correction=imread('Bayes_textbook.png');
corr_coe=int8(correction(:,:,1));
%get greyscale for correction
[r,c]=size(corr_coe);
for i=1:r
    for j=1:c
        if corr_coe(i,j)<119
            corr_coe(i,j)=-1;
        else
            corr_coe(i,j)=1;
        end
    end
end
corr_b=corr_coe;
[r,c]=size(corr_b);
sum=r*c;
comparison=0;
for i=1:r
    for j=1:c
        if corr_b(i,j)==gc(i,j)
            comparison=comparison+1;
        end
    end
end
%report recovery rate
recovery=(comparison/sum)*100;
fprintf('The recovery is %.4f \n', recovery)
%get image
imshow(uint8(gc)*255);
figure();
imshow(uint8(or)*255);
```

Part B

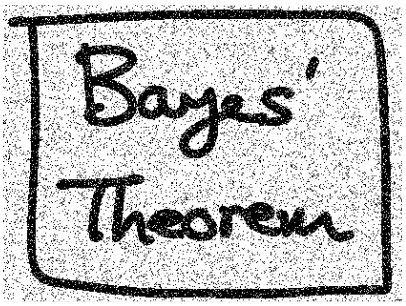
```
%read in data and preprocess
source=imread('Lenanoise.png');
source=int16(source);
src=source;
s=size(source);
yd=s(1);
xd=s(2);
form=@(x,N)(mod(x-1,N)+1);
d_lam=1;
```

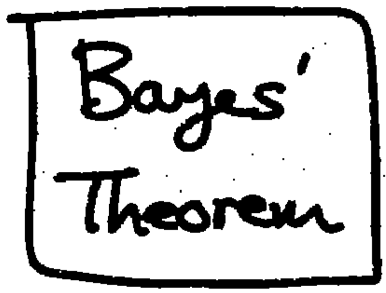
```
lam_s=1;
check=true;
procedure=1;
while(check)
    check=false;
    for i=1:xd
        for j=1:yd
            %1st case
            minu=(-d_lam*abs(max(0,source(j,i)-procedure)-
src(j,i)))-(lam_s*(abs(max(0,source(j,i)-procedure)-
source(form(j-1,yd),i))+abs(max(0,source(j,i)-procedure)-
source(j,form(i+1,xd)))+abs(max(0,source(j,i)-procedure)-
source(j,form(i-1,xd)))+abs(max(0,source(j,i)-procedure)-
source(form(j+1,yd),i))));
            %2nd case
            plus=(-d_lam*abs(min(255,source(j,i)+procedure)-
src(j,i)))-(lam_s*(abs(min(255,source(j,i)+procedure)-
source(form(j-1,yd),i))+abs(min(255,source(j,i)+procedure)-
source(j,form(i+1,xd)))+abs(min(255,source(j,i)+procedure)-
source(form(j+1,yd),i))+abs(min(255,source(j,i)+procedure)-
source(j,form(i-1,xd)))));
            %3rd case
            same=(-d_lam*abs(source(j,i)-src(j,i)))-
(lam_s*(abs(source(j,i)-source(form(j-1,yd),i))+abs(source(j,i)-
source(j,form(i+1,xd)))+abs(source(j,i)-
source(form(j+1,yd),i))+abs(source(j,i)-source(j,form(i-
1,xd)))));
            %variable and compare
            xi=source(j,i);
            if plus>same
                source(j,i)=min(255,xi+procedure);
                check=true;
            end
            if same<minu
                source(j,i)=max(0,xi-procedure);
                check=true;
            end
        end
    end

end
recover=imread('Lena.png');
%get image
imshow(uint8(source));
figure();
imshow(uint8(src));
```

Report:

Part A - the optimum values I have for h, β, η are 0.01, 5 and 7. The accuracy I get with these values is 99.2742%.






```
>> Part_A
The recovery is 99.2742
fx >>
```

Please run my part A to see the exact image outcomes and accuracy above.

The clean image is gotten from the noisy image. The image cannot be recovered exactly, but we have a pretty good result. Markov Random Frields are used.

Noise y_i is in {-1,1} orginal x_i is in {-1,1}

I write the Energy function:

E(x,y)=h$h\sum_i x_i - B\sum_{\{i,j\}} x_i x_j - n\sum_i x_i y_i$

and correction in my Part A so that they can be used directly in the file. Then I implement Coordinate-descent algorithm.

{x_i} (x_i=y_i)

For x_i if -x →E(x,y) decreases x=-x

I started with values 0.03, 15, 8. The accuracy started from around 94, then I adjust these values step by step and finally get to 99.2742% accuracy.

Part B – We still cannot recover Lena exactly. This part is harder than recovering the image in Part A.

Graph Model:

P(X|Y, lamda(d), lamda(s))=$\frac{1}{z}\exp\{lamda(d)\ \sum_i p(x_i - y_i) - lamda(s)\sum_{i,j\ is\ in\ \varepsilon} p(x_i - x_j)\}$

X=output clean, Y=input noise, 2 lambdas are weights

L_1 norm:P(z)=|z|, L_2norm P(z)=$|z|^2$

Max-sum alg→MAP→X

Argmax$_{\text{lamda(d), lamda(s)}}$p(X|Y,lamda(d),lamda(s))

Please see images below:

You can run my code to see the results above. The model is the extension of what we talked about.

p(X|Y, ,λ_d,λ_d)=$\frac{1}{Z}\exp\{-\lambda_d \sum_i p(x_i - y_i) - \lambda_s \sum_{(i,j) is\ in\ \varepsilon} p(x_i - x_j)\}$

X is to restore, Y is noisy Lena. Max sum is used to get MAP solution. I started from small values from 32 to 256. Then get multiple restore results. Then I chose the best case to report. Run my code to see the result.