CS182 Assignment 5

Charles Liu
cliu02@g.harvard.edu
November 13, 2015

1: Assuming a uniform prior distribution, calculate the condition probability table (CPT) of $Pr(X = x|E = e)$ for all x and e

---

x	e	$P(E = e\|X = x)P(X = x)$	$P(X = x\|E = e)$
A	A	.125	.5
B	A	.0625	.25
C	A	.0625	.25
D	A	0	0
A	B	.0625	.25
B	B	.125	.5
C	B	0	0
D	B	.0625	.25
A	C	.0625	.25
B	C	0	0
C	C	.125	.5
D	C	.0625	.25
A	D	0	0
B	D	.0625	.25
C	D	.0625	.25
D	D	.125	.5

2: Now let the prior distribution be:

x	$P(X = x)$
A	**0.4**
B	**0.2**
C	**0.1**
D	**0.3**

Calculate the CPT $P(X = x|E = e)$ for all x and e

---

x	e	$P(E=e \mid X=x)P(X=x)$	$P(X=x \mid E=e)$
A	A	.2	$.\overline{72}$
B	A	.05	$.\overline{18}$
C	A	.025	$.\overline{18}$
D	A	0	0
A	B	.1	$.\overline{36}$
B	B	.1	$.\overline{36}$
C	B	0	0
D	B	.075	$.\overline{27}$
A	C	.1	$.\overline{4}$
B	C	0	0
C	C	.05	$.\overline{2}$
D	C	.075	$.\overline{3}$
A	D	0	0
B	D	.05	$.\overline{2}$
C	D	.025	$.\overline{1}$
D	D	.15	$.\overline{6}$

3: We next consider a time series of observations. Our transition model will be (rows X_{i-1}):

$P(X_i \mid X_{i-1})$

	A	**B**	**C**	**D**
Begin	**1**	**0**	**0**	**0**
A	**0.5**	**0.5**	**0**	**0**
B	**0.0**	**0.5**	**0.5**	**0**
C	**0.5**	**0**	**0**	**0.5**
D	**0.25**	**0.25**	**0.25**	**0.25**

For this problem we are concerned with true (hidden) sequences, as opposed to observations. What is the probability under this model of the sequence of letters "A B B C D"? How about "A A B A"? What is $P(X_3 = x \mid X_1 = A, X_2 = B)$ for all x?

$$\begin{aligned}
ABBCD &= P(X_1=A)P(X_2=B|X_1=A)P(X_3=B|X_2=B)P(X_4=C|X_3=B)P(X_5=D|X_4=C) \\
&= 1*0.5*0.5*0.5*0.5 \\
&= .0625 \\
AABA &= P(X_1=A)P(X_2=A|X_1=A)P(X_3=B|X_2=A)P(X_4=A|X_3=B) \\
&= 1*0.5*0.5*0.0 \\
&= 0
\end{aligned}$$

$P(X_3 = x|X_1 = A, X_2 = B) = P(X_3 = x|X_2 = B)$ which corresponds to row B of the table:

	A	B	C	D
B	0.0	0.5	0.5	0

4: Finally we consider the full filtering problem $P(X_n|E_1,...,E_n)$. Let "A B B C D" be the sequence of observed key strokes. What is the current belief state of the model? That is compute $P(X_n = x|E_1 = A, E_2 = B, E_3 = B, E_4 = C, E_5 = D)$ for all x.

We build a beliefs table below (row i corresponds to belief b_i):

i	A	B	C	D
1	1.0	0.0	0.0	0.0
2	0.333333333333	0.666666666667	0.0	0.0
3	0.142857142857	0.857142857143	0.0	0.0
4	0.0769230769231	0.0	0.923076923077	0.0
5	0.0	0.04	0.0	0.96

Table produced by following Python code:

```
#Print latex table row
def printBelief(i, b):
    print "{} & {} & {} & {} & {} \\\\".format(i, b["A"], b["B"], b["C"], b["D"])
#Initial distribution
def initProb(x):
    if x == "A":
        return 1.0
    return 0.0
#Pr(E_i = e | X_i = x)
def E_cond_X(e, x):
    if e == x:
        return 0.5
    if (e == "A" and x == "D") or (e == "D" and x == "A") or \
       (e == "B" and x == "C") or (e == "C" and x == "B"):
           return 0.0
    return 0.25

#Pr(X_i = x | X_i-1 = x_1)
def X_cond_X_1(x, x_1):
    if x_1 == "A":
        if x == "A":
            return 0.5
        if x == "B":
            return 0.5
        if x == "C":
            return 0.0
        if x == "D":
            return 0.0
    if x_1 == "B":
        if x == "A":
            return 0.0
        if x == "B":
            return 0.5
        if x == "C":
            return 0.5
        if x == "D":
```

```
            return 0.0
    if x_1 == "C":
        if x == "A":
            return 0.5
        if x == "B":
            return 0.0
        if x == "C":
            return 0.0
        if x == "D":
            return 0.5
    if x_1 == "D":
        if x == "A":
            return 0.25
        if x == "B":
            return 0.25
        if x == "C":
            return 0.25
        if x == "D":
            return 0.25

#Normalize beliefs
def normalize(b):
    total = sum(b.values())
    for k in b:
        b[k] /= total

if __name__ == '__main__':
    states = ["A", "B", "C", "D"]
    E = ["A", "B", "B", "C", "D"]
    b = {}
    for x in states:
        b[x] = initProb(x)*E_cond_X(E[0], x)
    normalize(b)
    printBelief(1, b)
    for i in range(1, len(E)):
        temp = {}
        for x in states:
            temp[x] = 0.0
            for x_1 in states:
                temp[x] += E_cond_X(E[i], x)*X_cond_X_1(x, x_1)*b[x_1]
        normalize(temp)
        b = temp
        printBelief(i+1, b)
```