Digital Communication Systems
EES 452

Asst. Prof. Dr. Prapun Suksompong

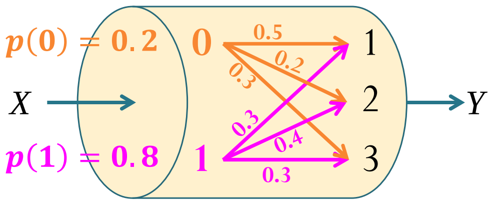
prapun@siit.tu.ac.th

3 An Introduction to Digital Communication Systems Over Discrete Memoryless Channel

3.3 Optimal Decoder

[Ex. 3.32]

Searching for the Optimal Detector



$$\mathbf{Q}=\begin{array}{c} x\backslash y \\ 0 \\ 1 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left[\begin{array}{ccc} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{array}\right] \end{array} \begin{array}{c} \\ \xrightarrow{\times 0.2} \\ \xrightarrow{\times 0.8} \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \left[\begin{array}{ccc} 0.10 & 0.04 & 0.06 \\ 0.24 & 0.32 & 0.24 \end{array}\right] \end{array}\!\!\begin{array}{c} y/x \\ 0 \\ 1 \end{array} = \mathbf{P}$$

y	$\hat{x}_{\text{Optimal}}(y)$
1	?
2	?
3	?

[DMC_decoder_ALL_demo.m]

[Ex. 3.32]

Searching for the Optimal Detector

```
% Channel Input
S_X = [0 1]; S_Y = [1 2 3];
%p_X = [0.6 0.4];
p_X = [0.2 0.8];
% Channel Characteristics
Q = [0.5 0.2 0.3; 0.3 0.4 0.3];

%% Construct all possible "reasonable" decoders
Decoder_Table_ALL = S_X';
for k = 2:length(S_Y);
    l = size(Decoder_Table_ALL,1);
    t = [];
    for i = 1:length(S_X)
        t = [t; [S_X(i)*ones(l,1) Decoder_Table_ALL]];
    end
    Decoder_Table_ALL = t;
end
```

[DMC_decoder_ALL_demo.m]

[Ex. 3.32]

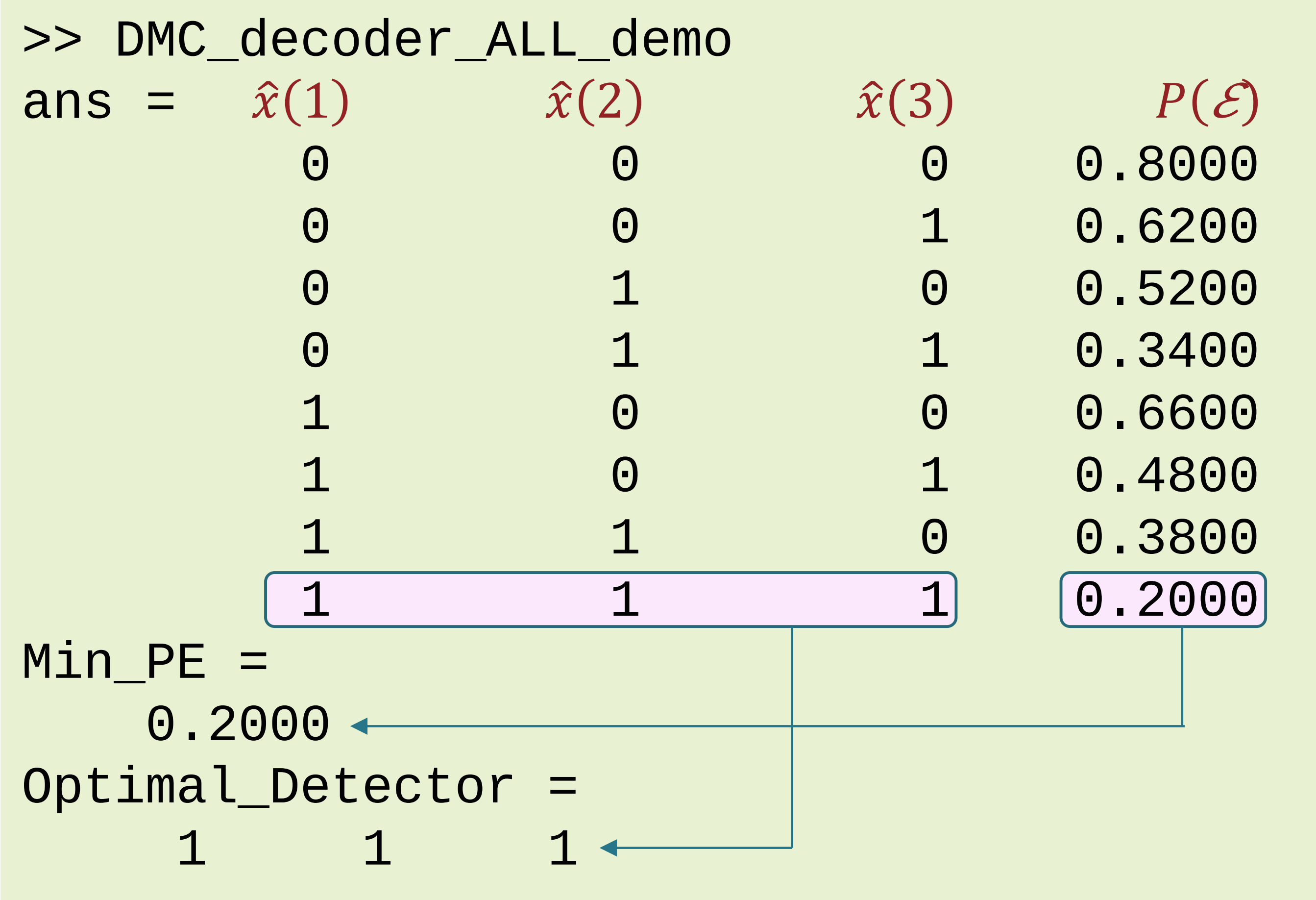
Searching for the Optimal Detector

```
%% Calculate the error probability for each of the decoder
PE_ALL = [];
for k = 1:size(Decoder_Table_ALL,1)
    Decoder_Table = Decoder_Table_ALL(k,:);
    PC = 0;
    for k = 1:length(S_X)
        I = (Decoder_Table == S_X(k));
        Q_row = Q(k,:);
        PC = PC + p_X(k)*sum(Q_row(I));
    end
    PE_theretical = 1-PC;
    PE_ALL = [PE_ALL; PE_theretical];
end

%% Display the results
[Decoder_Table_ALL PE_ALL]

%% Find the optimal detectors
Min_PE = min(PE_ALL)
I = [abs(PE_ALL-Min_PE)<1e-10];
Optimal_Detector = Decoder_Table_ALL(I,:)
```

[DMC_decoder_ALL_demo.m]

[Ex. 3.32]

Searching for the Optimal Detector



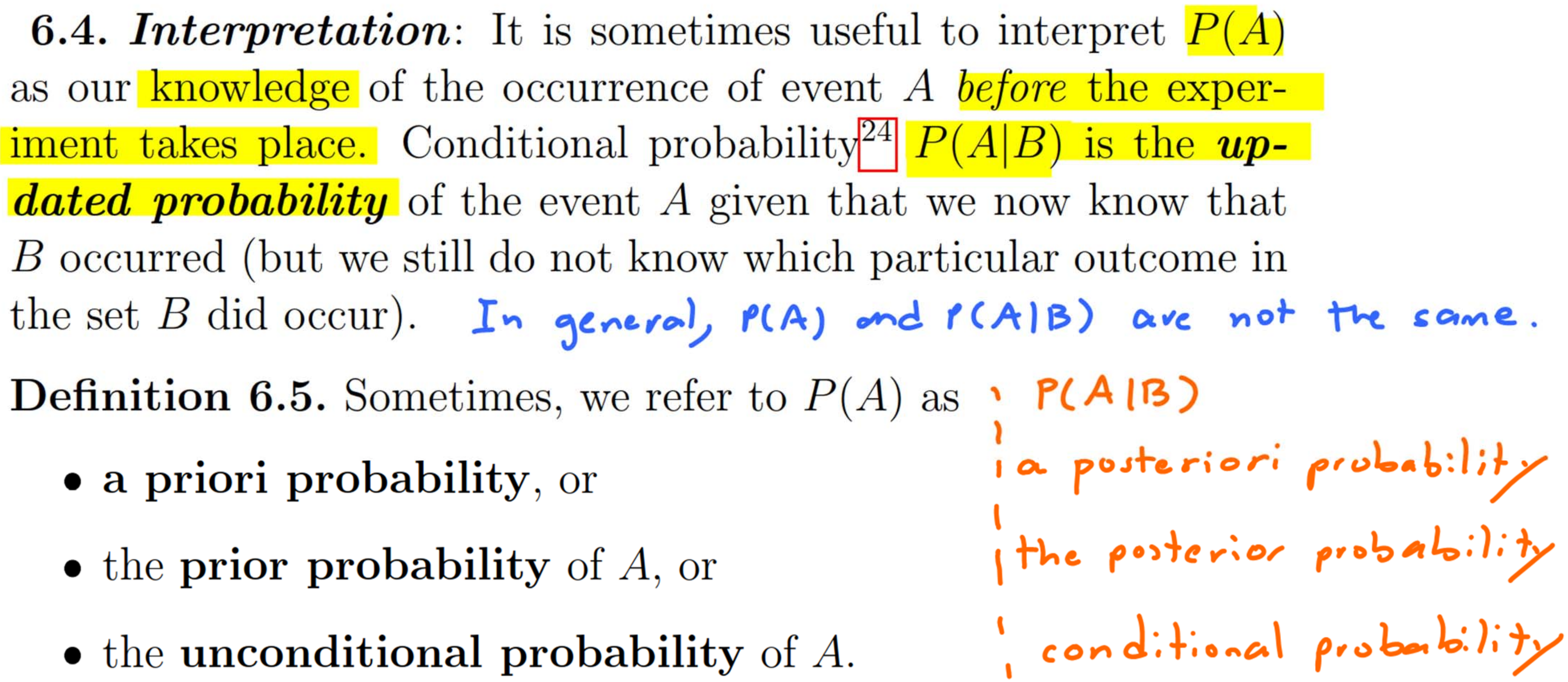
```
>> DMC_decoder_ALL_demo
ans =    x̂(1)        x̂(2)          x̂(3)       P(ℰ)
           0           0            0      0.8000
           0           0            1      0.6200
           0           1            0      0.5200
           0           1            1      0.3400
           1           0            0      0.6600
           1           0            1      0.4800
           1           1            0      0.3800
           1           1            1      0.2000
Min_PE =
    0.2000
Optimal_Detector =
     1     1     1
```

Ex. 3.26

Ex. 3.27

[DMC_decoder_ALL_demo.m]

Review: EES315 (2020)



6.4. ***Interpretation***: It is sometimes useful to interpret $P(A)$ as our knowledge of the occurrence of event A *before* the experiment takes place. Conditional probability[24] $P(A|B)$ is the ***updated probability*** of the event A given that we now know that B occurred (but we still do not know which particular outcome in the set B did occur). In general, $P(A)$ and $P(A|B)$ are not the same.

Definition 6.5. Sometimes, we refer to $P(A)$ as

- **a priori probability**, or
- the **prior probability** of A, or
- the **unconditional probability** of A.

$P(A|B)$

a posteriori probability

the posterior probability

conditional probability

[http://www2.siit.tu.ac.th/prapun/ees315/EES315%206.1%20u4.pdf]

Summary: MAP vs. ML Decoders

$$\begin{aligned}\hat{x}_{\text{MAP}}(y) &= \arg\max_x p_{X,Y}(x,y) \\ &= \hat{x}_{\text{optimal}}(y) \\ &= \arg\max_x \overbrace{P\left[X = x \middle| Y = y\right]}^{\text{a posteriori probability}} \\ &= \arg\max_x Q(y|x)\overbrace{p(x)}^{\text{prior probability}}\end{aligned}$$

- Decoder is derived from the **P** matrix
- Select (by circling) the maximum value in each column (for each value of y) in the **P** matrix.
- The corresponding x value is the value of $\hat{x}_{\text{MAP}}(y)$.

$$\hat{x}_{\text{ML}}(y) = \arg\max_x \overbrace{Q(y|x)}^{\text{likelihood funtion}}$$

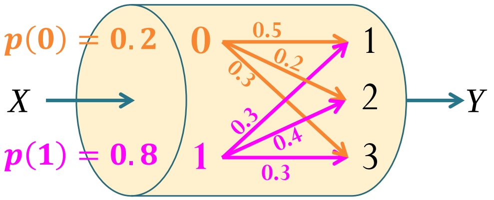
Optimal at least when $p(x)$ is uniform (the channel inputs are equally likely)

Can be derived without knowing the channel input probabilities.

- Decoder is derived from the **Q** matrix
- Select (by circling) the maximum value in each column (for each value of y) in the **Q** matrix.
- The corresponding x value is the value of $\hat{x}_{\text{ML}}(y)$.

- Once the decoder (the decoding table) is derived $P(\mathcal{C})$ and $P(\mathcal{E})$ are calculated by adding the corresponding probabilities in the **P** matrix.

Example: MAP Decoder [Ex. 3.41]



y	$\hat{x}_{\text{MAP}}(y)$
1	1
2	1
3	1

$$\mathbf{Q} = \begin{array}{cc} x\backslash y & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array} \begin{array}{c} \xrightarrow{\times 0.2} \\ \xrightarrow{\times 0.8} \end{array} \begin{array}{cc} \begin{array}{ccc} 1 & 1 & 1 \\ 1 & 2 & 3 \end{array} & \begin{array}{c} \hat{x}_{\text{MAP}}(y) \\ y\diagup x \end{array} \\ \begin{bmatrix} 0.10 & 0.04 & 0.06 \\ \boxed{0.24} & \boxed{0.32} & \boxed{0.24} \end{bmatrix} & \begin{array}{c} 0 \\ 1 \end{array} \end{array} = \mathbf{P}$$

$$P(\mathcal{E}) = 1 - P(\mathcal{C}) = 1 - (0.24 + 0.32 + 0.24) = 0.2$$

MAP Decoder: Two Approaches

[3.37-3.39]

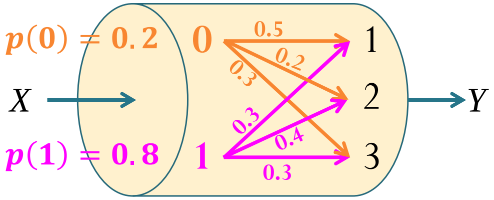
$$\begin{aligned}\hat{x}_{\text{MAP}}(y) &= \arg\max_{x} p_{X,Y}(x,y) \\ &= \hat{x}_{\text{optimal}}(y) \\ &= \arg\max_{x} \overbrace{P\left[X = x \middle| Y = y\right]}^{\text{a posteriori probability}} \\ &= \arg\max_{x} Q(y|x)\overbrace{p(x)}^{\text{prior probability}}\end{aligned}$$

[3.40]

1) Decoder is derived from the **P** matrix

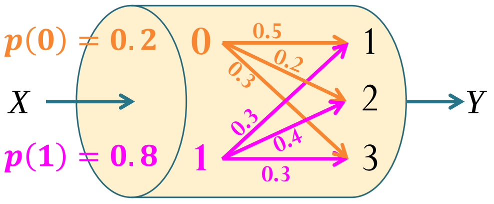
2) Select (by circling) the maximum value in each column (for each value of y) in the **P** matrix.

3) The corresponding x value is the value of $\hat{x}_{\text{MAP}}(y)$.



$$\mathbf{Q} = \begin{array}{cc} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ x \backslash y & \\ \begin{array}{c} 0 \\ 1 \end{array} & \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.4 & 0.3 \end{bmatrix} \end{array} \begin{array}{c} \xrightarrow{\times 0.2} \\ \xrightarrow{\times 0.8} \end{array} \begin{array}{cc} \begin{array}{ccc} 1 & 1 & 1 \end{array} & \hat{x}_{\text{MAP}}(y) \\ \begin{array}{ccc} 1 & 2 & 3 \end{array} & y \diagup x \\ \begin{bmatrix} 0.10 & 0.04 & 0.06 \\ 0.24 & 0.32 & 0.24 \end{bmatrix} & \begin{array}{c} 0 \\ 1 \end{array} \end{array} = \mathbf{P}$$

Example: ML Decoder [Ex. 3.47]



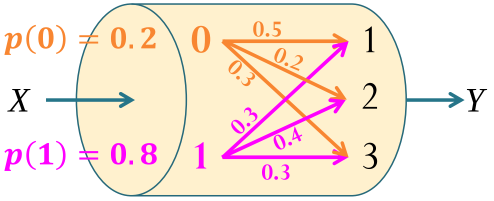
[Same as Ex. 3.26]

y	$\hat{x}_{\mathrm{ML}}(y)$
1	0
2	1
3	0

Sol 1:

$$\mathbf{Q} = \begin{array}{c|ccc} x\backslash y & 1 & 2 & 3 \\ \hline 0 & \boxed{0.5} & 0.2 & \boxed{0.3} \\ 1 & 0.3 & \boxed{0.4} & 0.3 \end{array} \begin{array}{c} \xrightarrow{\times 0.2} \\ \xrightarrow{\times 0.8} \end{array} \begin{array}{ccc|c} 0 & 1 & 0 & \hat{x}_{\mathrm{ML}}(y) \\ 1 & 2 & 3 & y/x \\ \hline \boxed{0.10} & 0.04 & \boxed{0.06} & 0 \\ 0.24 & \boxed{0.32} & 0.24 & 1 \end{array} = \mathbf{P}$$

$$P(\mathcal{E}) = 1 - P(\mathcal{C}) = 1 - (0.10 + 0.32 + 0.06) = 0.52$$



y	$\hat{x}_{\mathrm{ML}}(y)$
1	0
2	1
3	1

Sol 2:

$$\mathbf{Q} = \begin{array}{c|ccc} x\backslash y & 1 & 2 & 3 \\ \hline 0 & \boxed{0.5} & 0.2 & 0.3 \\ 1 & 0.3 & \boxed{0.4} & \boxed{0.3} \end{array} \begin{array}{c} \xrightarrow{\times 0.2} \\ \xrightarrow{\times 0.8} \end{array} \begin{array}{ccc|c} 0 & 1 & 1 & \hat{x}_{\mathrm{ML}}(y) \\ 1 & 2 & 3 & y/x \\ \hline \boxed{0.10} & 0.04 & 0.06 & 0 \\ 0.24 & \boxed{0.32} & \boxed{0.24} & 1 \end{array} = \mathbf{P}$$

[Agree with the complete table in Ex. 3.32]

$$P(\mathcal{E}) = 1 - P(\mathcal{C}) = 1 - (0.10 + 0.32 + 0.24) = 0.34$$

MAP Decoder

```
%% MAP Decoder
P = diag(p_X)*Q; % Weight the channel transition probability by the
                 % corresponding prior probability.
[V I] = max(P); % For I, the default MATLAB behavior is that when there are
                % multiple max, the index of the first one is returned.
Decoder_Table = S_X(I) % The decoded values corresponding to the received Y
```

```
%% Decode according to the decoder table
x_hat = y; % preallocation
for k = 1:length(S_Y)
    I = (y==S_Y(k));
    x_hat(I) = Decoder_Table(k);
end

PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
```

```
%% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    I = (Decoder_Table == S_X(k));
    Q_row = Q(k,:);
    PC = PC+ p_X(k)*sum(Q_row(I));
end
PE_theretical = 1-PC
```

[DMC_decoder_MAP_demo.m]

ML Decoder

```
%% ML Decoder
[V I] = max(Q); % For I, the default MATLAB behavior is that when there are
                % multiple max, the index of the first one is returned.
Decoder_Table = S_X(I) % The decoded values corresponding to the received Y
```

```
%% Decode according to the decoder table
x_hat = y; % preallocation
for k = 1:length(S_Y)
    I = (y==S_Y(k));
    x_hat(I) = Decoder_Table(k);
end

PE_sim = 1-sum(x==x_hat)/n % Error probability from the simulation
```

```
%% Calculation of the theoretical error probability
PC = 0;
for k = 1:length(S_X)
    I = (Decoder_Table == S_X(k));
    Q_row = Q(k,:);
    PC = PC+ p_X(k)*sum(Q_row(I));
end
PE_theretical = 1-PC
```

[DMC_decoder_ML_demo.m]

Different Decoders

For naïve decoder, $\hat{x}_{\text{naïve}}(y) = y$. Therefore, we can simply copy the values in the y-column into its decoding table.

For DIY decoder, the decoding table is provided. (Alternatively, some equation may be given; in which case, we can simply plug-in each of the possible y values to get $\hat{x}_{\text{DIY}}(y)$.)

y	$\hat{x}_{\text{naïve}}(y)$	$\hat{x}_{\text{DIY}}(y)$	$\hat{x}_{\text{MAP}}(y)$	$\hat{x}_{\text{ML}}(y)$
1	1	1	1	0
2	2	1	1	1
3	3	0	1	0
	[Ex. 3.25]	[Ex. 3.27]	[Ex. 3.35]	[Ex. 3.46]

Deriving the MAP and ML decoder follows almost the same recipe: select the max value in each column and read the corresponding x-value. The difference is that the MAP decoder uses the **P** matrix but the ML decoder uses the **Q** matrix.

$$\mathbf{Q} = \begin{array}{cccc} x \backslash y & 1 & 2 & 3 \\ 0 & \enclose{circle}{0.5} & 0.2 & \enclose{circle}{0.3} \\ 1 & 0.3 & \enclose{circle}{0.4} & 0.3 \end{array} \quad \begin{array}{c} \xrightarrow{\times 0.2} \\ \xrightarrow{\times 0.8} \end{array} \quad \begin{array}{cccc} x \backslash y & 1 & 2 & 3 \\ 0 & 0.10 & 0.04 & 0.06 \\ 1 & \enclose{circle}{0.24} & \enclose{circle}{0.32} & \enclose{circle}{0.24} \end{array} = \mathbf{P}$$

Finding $P(\mathcal{E})$

- Once a decoder $\hat{x}(y)$ is defined, we can find its corresponding $P(\mathcal{E})$ easily from the **P** matrix:
  - Write $\hat{x}(y)$ values on top of the y values for the **P** matrix.
  - For each column y in the **P** matrix, circle the element whose corresponding x value is the same as $\hat{x}(y)$.

y	$\hat{x}_{\textbf{naïve}}(y)$	$\hat{x}_{\textbf{DIY}}(y)$	$\hat{x}_{\textbf{MAP}}(y)$	$\hat{x}_{\textbf{ML}}(y)$
1	1	1	1	0
2	2	1	1	1
3	3	0	1	0
$P(\mathcal{E})$	0.76	0.38	0.20	0.52
	[Ex. 3.25]	[Ex. 3.27]	[Ex. 3.35]	[Ex. 3.46]

- $P(\mathcal{C})$ = the sum of the circled probabilities.
- $P(\mathcal{E}) = 1 - P(\mathcal{C})$.

$$\mathbf{P} = \begin{array}{c|ccc} \hat{x}_{\text{naive}}(y) & 1 & 2 & 3 \\ x \backslash y & 1 & 2 & 3 \\ 0 & 0.10 & 0.04 & 0.06 \\ 1 & \boxed{0.24} & 0.32 & 0.24 \end{array} \qquad \begin{array}{c|ccc} \hat{x}_{\text{DIY}}(y) & 1 & 1 & 0 \\ x \backslash y & 1 & 2 & 3 \\ 0 & 0.10 & 0.04 & \boxed{0.06} \\ 1 & \boxed{0.24} & \boxed{0.32} & 0.24 \end{array} \qquad \begin{array}{c|ccc} \hat{x}_{\text{MAP}}(y) & 1 & 1 & 1 \\ x \backslash y & 1 & 2 & 3 \\ 0 & 0.10 & 0.04 & 0.06 \\ 1 & \boxed{0.24} & \boxed{0.32} & \boxed{0.24} \end{array} \qquad \begin{array}{c|ccc} \hat{x}_{\text{ML}}(y) & 0 & 1 & 0 \\ x \backslash y & 1 & 2 & 3 \\ 0 & \boxed{0.10} & 0.04 & \boxed{0.06} \\ 1 & 0.24 & \boxed{0.32} & 0.24 \end{array}$$