Digital Communication Systems
ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

4. *Mutual Information* and Channel Capacity

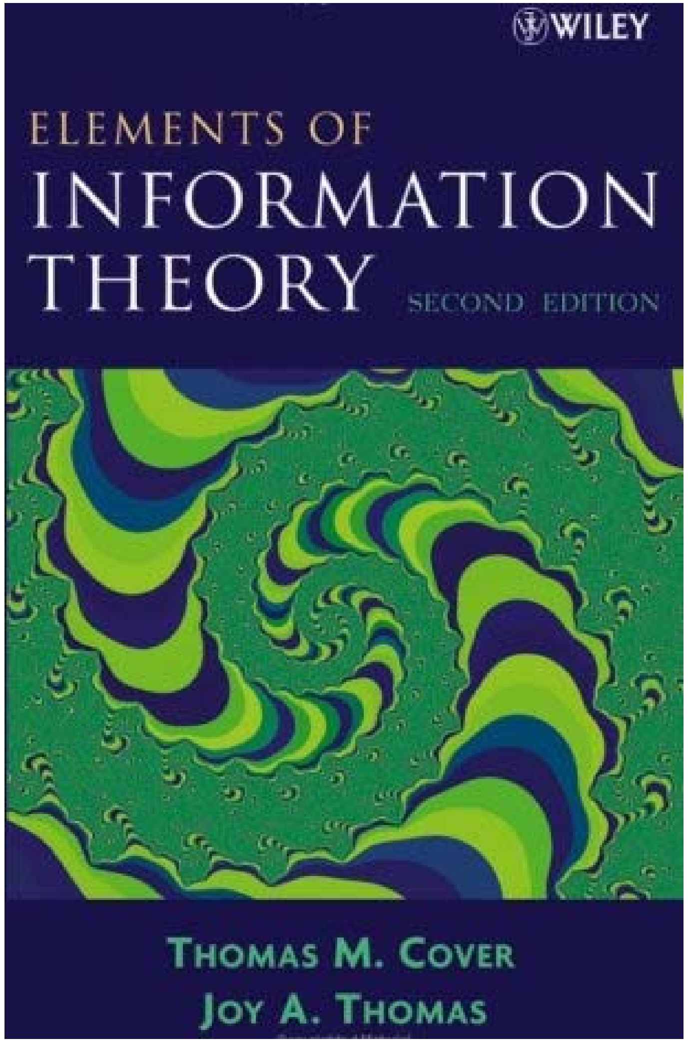
Office Hours:

BKD, 6th floor of Sirindhralai building

Monday	10:00-10:40
Tuesday	12:00-12:40
Thursday	14:20-15:30

Reference for this chapter

- Elements of Information Theory
- By Thomas M. **Cover** and Joy A. **Thomas**
- 2nd Edition (Wiley)
- Chapters 2, 7, and 8
- 1st Edition available at SIIT library: Q360 C68 1991



Digital Communication Systems ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Information-Theoretic Quantities

Conditional Entropies

Amount of randomness in Y

$$H(Y) \equiv -\sum_{y\in\mathcal{Y}} q(y)\log_2 \overbrace{q(y)}^{P[Y=y]} \equiv H(\underline{\mathbf{q}})$$

Amount of randomness still remained in Y when we know that $X = x$.

$$\overbrace{H(Y|X=x)}^{\text{given a particular value } x} \equiv H(Y|x) \equiv -\sum_{y\in\mathcal{Y}} Q(y|x)\log_2 \overbrace{Q(y|x)}^{P[Y=y|X=x]}$$

Apply the entropy calculation to a row from the $\mathbf{Q}$ matrix

The **average** amount of randomness still remained in Y when we know X

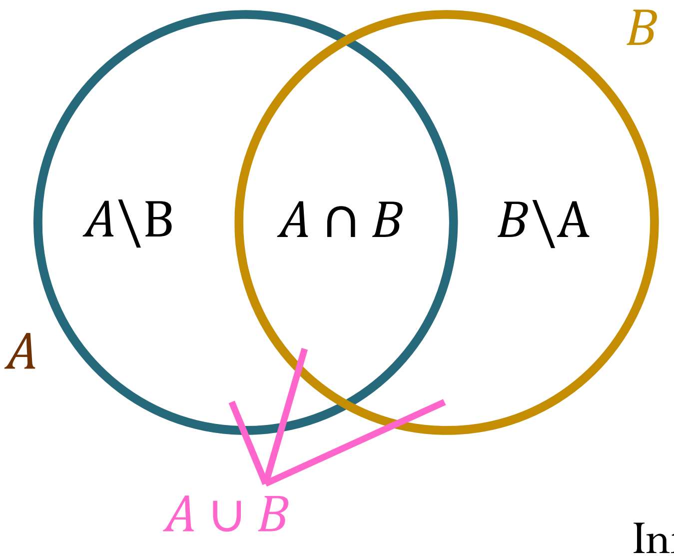
$$H(Y|X) \equiv \sum_{x\in\mathcal{X}} p(x)H(Y|x)$$

$$= H(X,Y) - H(X)$$

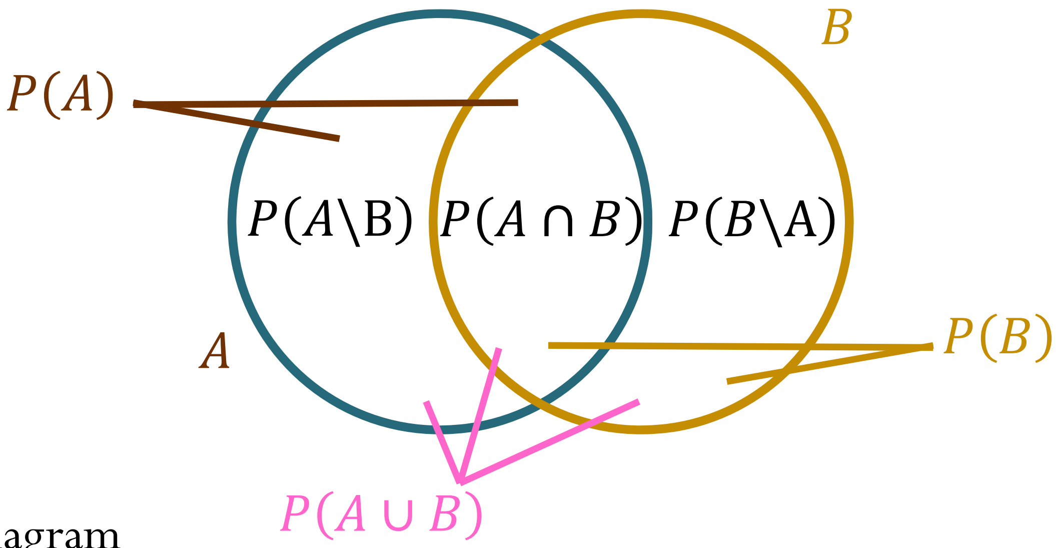
$$= H(Y) - I(X;Y)$$

Diagrams [Figure 16]

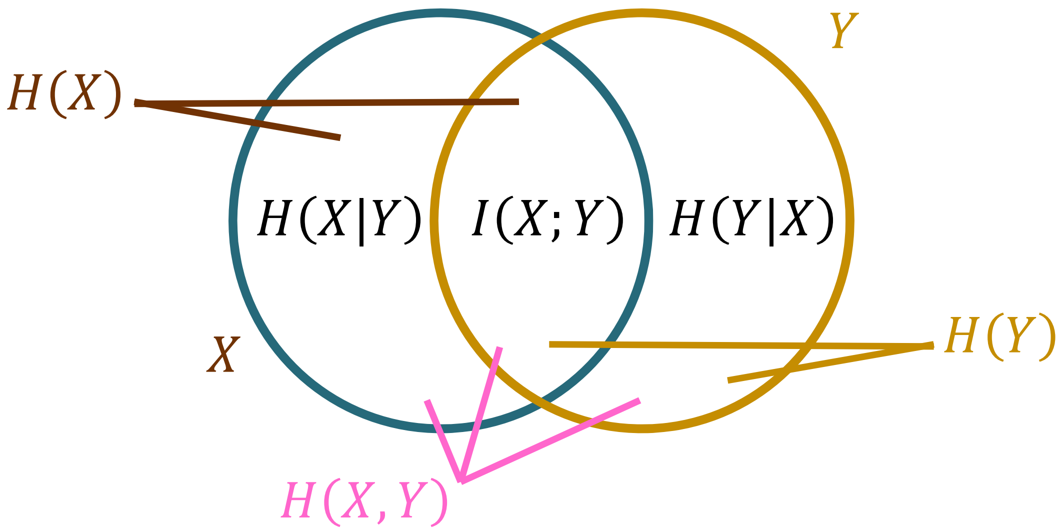
Venn Diagram



Probability Diagram

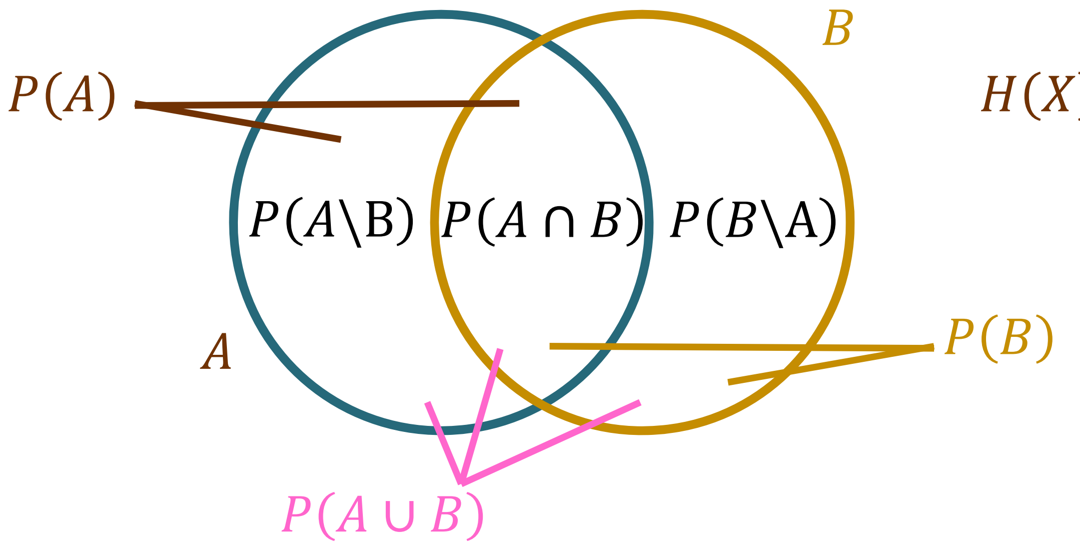


Information Diagram

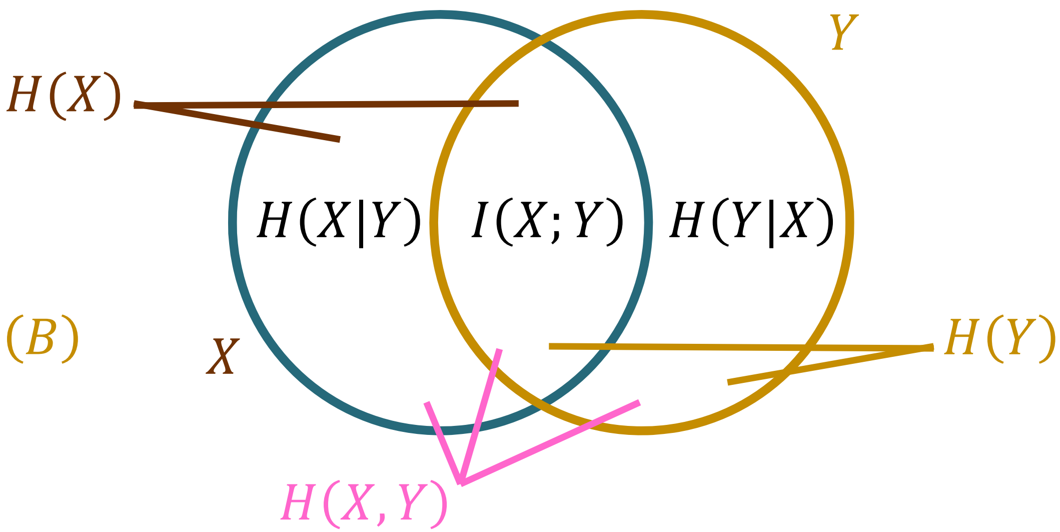


Diagrams [Figure 16]

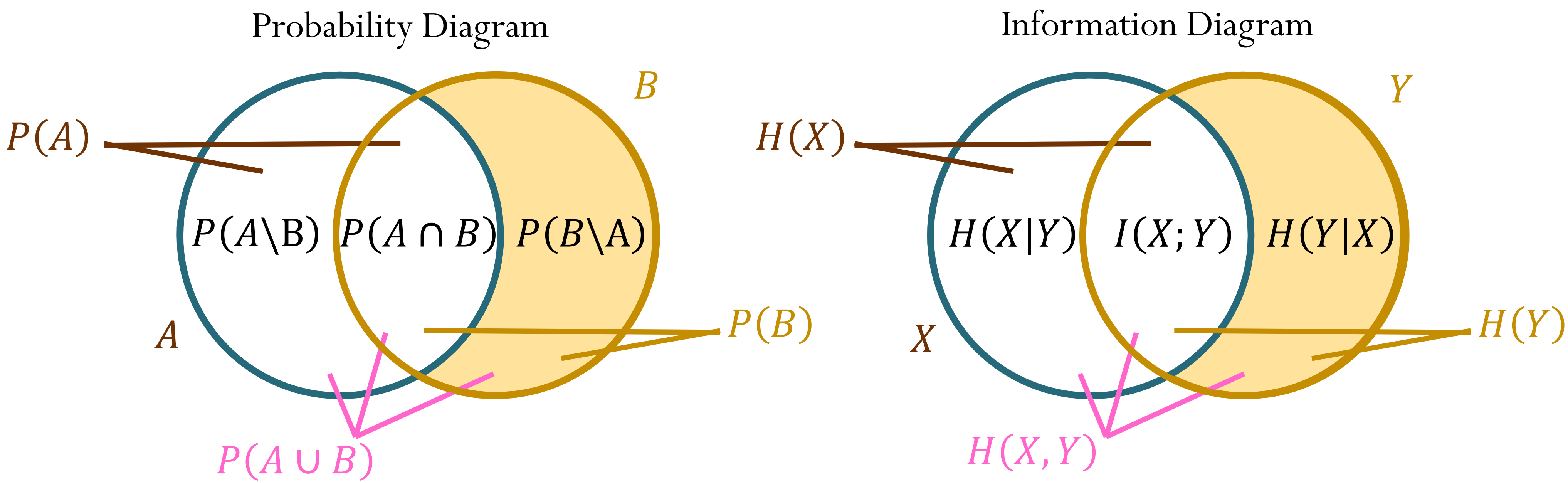
Probability Diagram



Information Diagram



Diagrams



$$P(B \setminus A) = P(A \cup B) - P(A) \qquad H(Y|X) = H(X,Y) - H(X)$$

Digital Communication Systems ECS 452

Asst. Prof. Dr. Prapun Suksompong

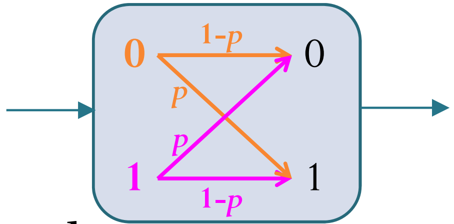
prapun@siit.tu.ac.th

Operational Channel Capacity

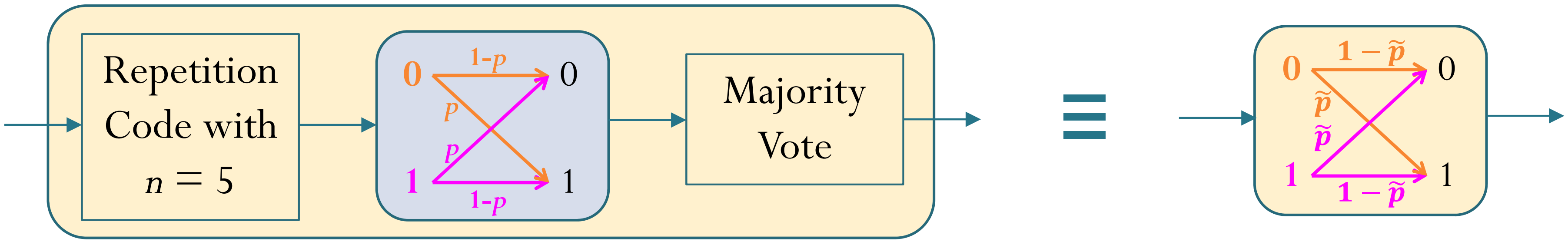
Example: Repetition Code

[Figure 14]

- Original Equivalent Channel:



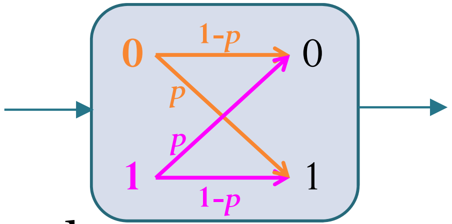
  - BSC with crossover probability $p = 0.01$
- New (and Better) Equivalent Channel:



  - Use repetition code with $n = 5$ at the transmitter
  - Use majority vote at the receiver
  - New BSC with $\tilde{p} = \binom{5}{3}p^3(1-p)^2 + \binom{5}{4}p^4(1-p)^1 + \binom{5}{5}p^5(1-p)^0 \approx 10^{-5}$

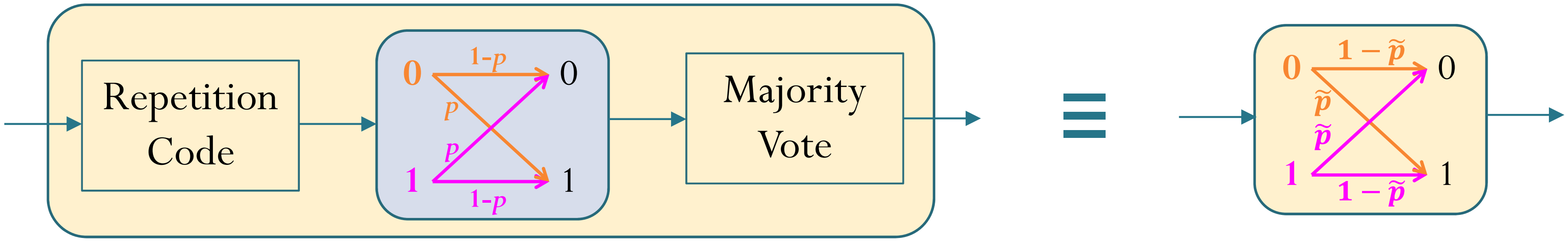
Example: Repetition Code

- Original Equivalent Channel:



  - BSC with crossover probability $p = 0.1$

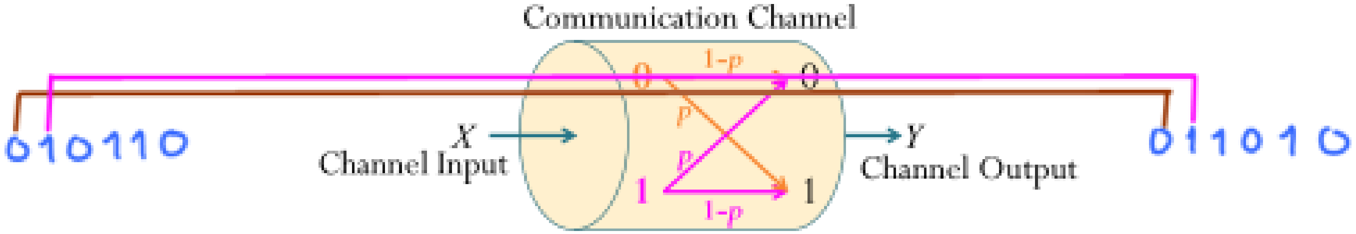
- New (and Better) Equivalent Channel:



  - Use repetition code at the transmitter
  - Use majority vote at the receiver
  - New BSC with new crossover probability $\tilde{p}$

Reminder [From ECS315]

Example 6.58. ***Digital communication over unreliable channels***: Consider a digital communication system through the **binary symmetric channel** (BSC) discussed in Example 6.18. We repeat its compact description here.



Suppose you put 010110 into this channel. What is the probability that you get 011010 as output?

$(1-p)(1-p)pp(1-p)(1-p)$

$= (1-p)^4 p^2$

85

$P(\mathcal{E}) = P(\mathcal{E} | [X=0]) P[X=0] + P(\mathcal{E} | [X=1]) P[X=1]$

$= p \times P[X=0] + p\, P[X=1] = p\,(\underbrace{P[X=0] + P[X=1]}_{1}) = p$

Again this channel can be described as a channel that introduces

One method of reducing the error rate is to use error-correcting codes:

[From ECS315]

A simple error-correcting code is the ***repetition code***. Exam-

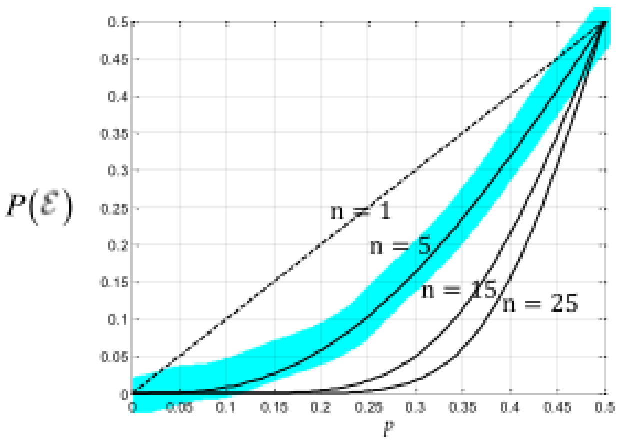
Two ways to calculate the probability of error:

(a) (transmission) error occurs if and only if the number of bits in error are ≥ 3.

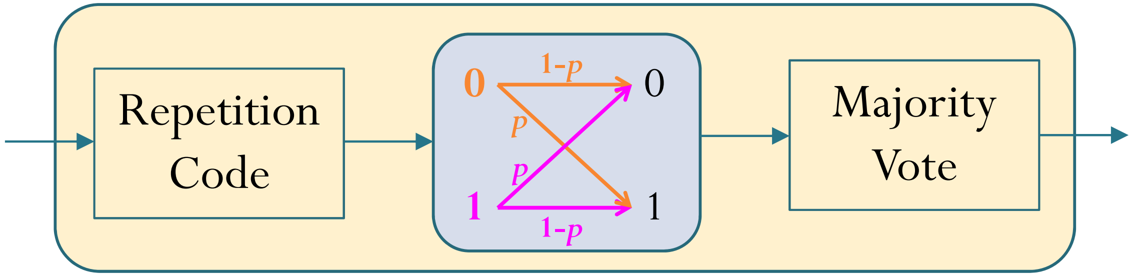
$$\left(P[\hat{M} \neq M] = \right) P(\mathcal{E}) = \binom{5}{3}p^3(1-p)^2 + \binom{5}{4}p^4(1-p) + \binom{5}{5}p^5(1-p)^0$$

(b) (transmission) error occurs if and only if the number of bits *not* in error are ≤ 2.

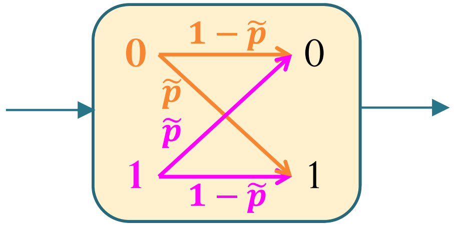
$$P(\mathcal{E}) = \binom{5}{0}(1-p)^0p^5 + \binom{5}{1}(1-p)^1p^4 + \binom{5}{2}(1-p)^2p^3$$



Example: Repetition Code



≡



n	$\tilde{p}$
1	$p = 0.1$
3	$\binom{3}{2}p^2(1-p)+\binom{3}{3}p^3 \approx 0.0280$
5	$\binom{5}{3}p^3(1-p)^2+\binom{5}{4}p^4(1-p)^1+\binom{5}{5}p^5 \approx 0.0086$
7	≈ 0.0027
9	$\approx 8.9092\times10^{-4}$
11	$\approx 2.9571\times10^{-4}$

Digital Communication Systems ECS 452

Asst. Prof. Dr. Prapun Suksompong

prapun@siit.tu.ac.th

Information Channel Capacity

Channel Capacity

Channel Capacity

"**Operational**": max rate at which **reliable** communication is possible

Arbitrarily small error probability can be achieved.

"**Information**": $\max_{\underline{\mathbf{p}}} I(X;Y)$ [bpcu]

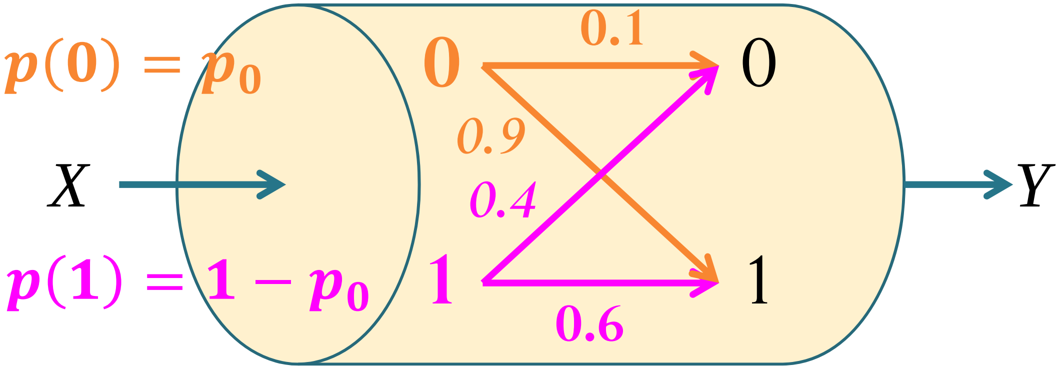
Shannon [1948] shows that these two quantities are actually the same.

MATLAB

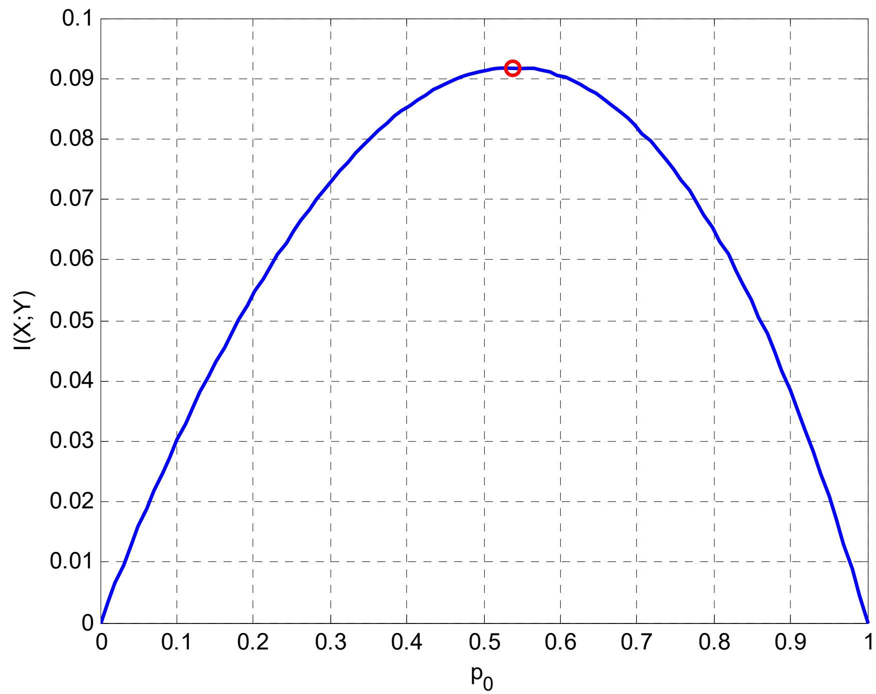
```
function H = entropy2s(p)
% ENTROPY2 accepts probability mass function
% as a row vector, calculate the corresponding
% entropy in bits.
p=p(find(abs(sort(p)-1)>1e-8)); % Eliminate 1
p=p(find(abs(p)>1e-8)); % Eliminate 0
if length(p)==0
    H = 0;
else
    H = simplify(-sum(p.*log(p))/log(sym(2)));
end
```

```
function I = informations(p,Q)
X = length(p);
q = p*Q;
HY = entropy2s(q);
temp = [];
for i = 1:X
    temp = [temp entropy2s(Q(i,:))];
end
HYgX = sum(p.*temp);
I = HY-HYgX;
```

Capacity calculation for BAC

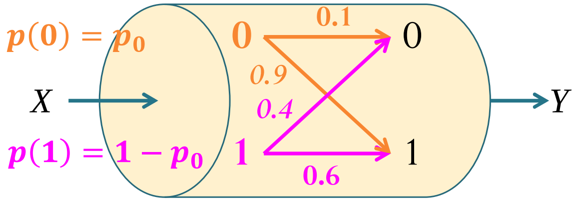


$$Q = \begin{bmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{bmatrix}$$



Capacity of 0.0918 bits is achieved by $\underline{p} = [0.5380, \ 0.4620]$

Capacity calculation for BAC

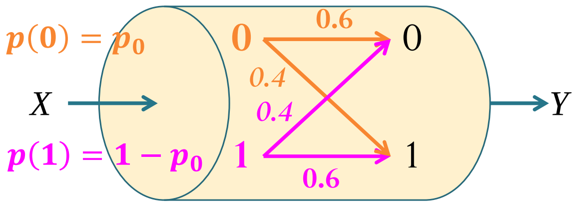


$$Q = \begin{bmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{bmatrix}$$

```
close all; clear all;
syms p0
p = [p0 1-p0];
Q = [1 9; 4 6]/sym(10);

I = simplify(informations(p,Q))

p0o = simplify(solve(diff(I)==0))

po = eval([p0o 1-p0o])

C = simplify(subs(I,p0,p0o))

eval(C)
```

```
>> Capacity_Ex_BAC
I =
(log(2/5 - (3*p0)/10)*((3*p0)/10 - 2/5) - log((3*p0)/10 + 3/5)*((3*p0)/10 + 3/5))/log(2) + (log((5*2^(3/5)*3^(2/5))/6)*(p0 - 1))/log(2) + (p0*log((3*3^(4/5))/10))/log(2)
p0o =
(27648*2^(1/3))/109565 - (69984*2^(2/3))/109565 + 135164/109565
po =
    0.5376    0.4624
C =
(log((3*3^(4/5))/10)*((27648*2^(1/3))/109565 - (69984*2^(2/3))/109565 + 135164/109565))/log(2) - (log((104976*2^(2/3))/547825 - (41472*2^(1/3))/547825 + 16384/547825)*((104976*2^(2/3))/547825 - (41472*2^(1/3))/547825 + 16384/547825) + log((41472*2^(1/3))/547825 - (104976*2^(2/3))/547825 + 531441/547825)*((41472*2^(1/3))/547825 - (104976*2^(2/3))/547825 + 531441/547825))/log(2) + (log((5*2^(3/5)*3^(2/5))/6)*((27648*2^(1/3))/109565 - (69984*2^(2/3))/109565 + 25599/109565))/log(2)
ans =
    0.0918
```

Same procedure applied to BSC



$$Q = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{bmatrix}$$

```
close all; clear all;
syms p0
p = [p0 1-p0];
Q = [6 4; 4 6]/sym(10);

I = simplify(informations(p,Q))

p0o = simplify(solve(diff(I)==0))

po = eval([p0o 1-p0o])

C = simplify(subs(I,p0,p0o))

eval(C)
```

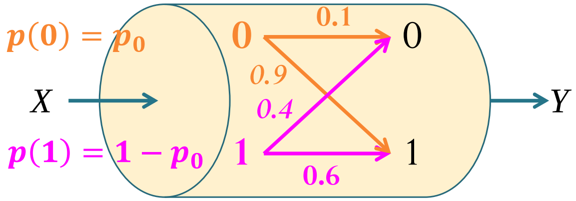
```
>> Capacity_Ex_BSC
I =
(log((5*2^(3/5)*3^(2/5))/6)*(p0 - 1))/log(2) -
(p0*log((5*2^(3/5)*3^(2/5))/6))/log(2) - (log(p0/5 +
2/5)*(p0/5 + 2/5) - log(3/5 - p0/5)*(p0/5 -
3/5))/log(2)
p0o =
1/2
po =
   0.5000   0.5000
C =
log((2*2^(2/5)*3^(3/5))/5)/log(2)
ans =
   0.0290
```

Blahut–Arimoto algorithm

```
function [ps C] = capacity_blahut(Q)
% Input:      Q  = channel transition probability matrix
% Output:     C  = channel capacity
%             ps = row vector containing pmf that achieves capacity

tl = 1e-8; % tolerance (for the stopping condition)
n = 1000; % max number of iterations (in case the stopping condition
          % is "never" reached")
nx = size(Q,1); pT = ones(1,nx)/nx; % First, guess uniform X.
for k = 1:n
    qT = pT*Q;
    % Eliminate the case with 0
    % Column-division by qT
    temp = Q.*(ones(nx,1)*(1./qT));
    %Eliminate the case of 0/0
    l2 = log2(temp);
    l2(find(isnan(l2) | (l2==-inf) | (l2==inf)))=0;
    logc = (sum(Q.*(l2),2))';
    CT = 2.^(logc);
    A = log2(sum(pT.*CT)); B = log2(max(CT));
    if((B-A)<tl)
        break
    end
    % For the next loop
    pT = pT.*CT;     % un-normalized
    pT = pT/sum(pT); % normalized
    if(k == n)
        fprintf('\nNot converge within n loops\n')
    end
end
ps = pT;
C = (A+B)/2;
```

[capacity_blahut.m]

Capacity calculation for BAC: a revisit



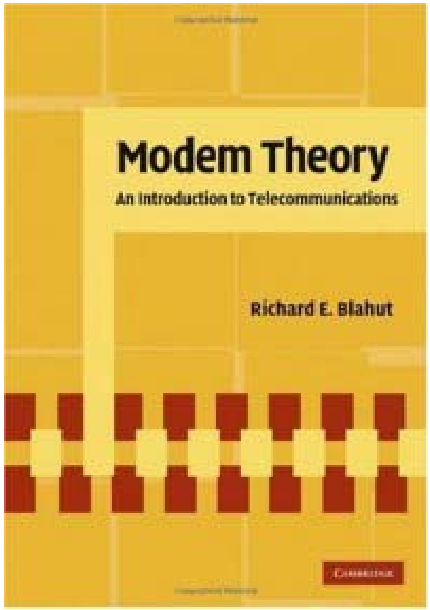
$$Q = \begin{bmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{bmatrix}$$

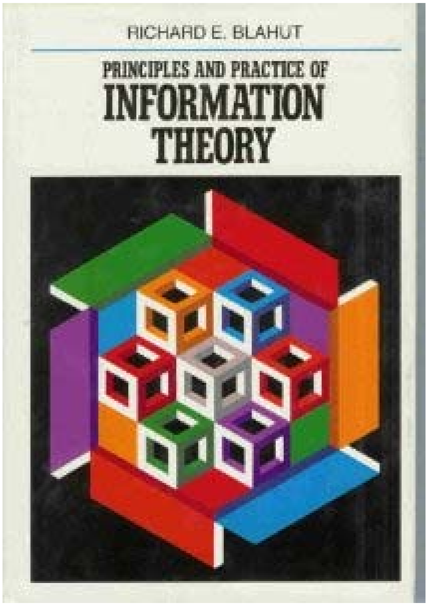
```
close all; clear all;

Q = [1 9; 4 6]/10;

[ps C] = capacity_blahut(Q)
```

```
>> Capacity_Ex_BAC_blahut
ps =
   0.5376   0.4624
C =
   0.0918
```

Richard Blahut

- Former chair of the Electrical and Computer Engineering Department at the University of Illinois at Urbana-Champaign
- Best known for **Blahut–Arimoto algorithm (Iterative Calculation of C)**





Claude E. Shannon Award

Claude E. **Shannon** (1972)
David S. Slepian (1974)
Robert M. Fano (1976)
Peter Elias (1977)
Mark S. Pinsker (1978)
Jacob Wolfowitz (1979)
W. Wesley Peterson (1981)
Irving S. Reed (1982)
Robert G. Gallager (1983)
Solomon W. Golomb (1985)
William L. Root (1986)
James L. Massey (1988)
Thomas M. Cover (1990)
Andrew J. Viterbi (1991)

Elwyn R. Berlekamp (1993)
Aaron D. Wyner (1994)
G. David Forney, Jr. (1995)
Imre Csiszár (1996)
Jacob Ziv (1997)
Neil J. A. Sloane (1998)
Tadao Kasami (1999)
Thomas Kailath (2000)
Jack Keil Wolf (2001)
Toby **Berger** (2002)
Lloyd R. Welch (2003)
Robert J. McEliece (2004)
Richard Blahut (2005)
Rudolf Ahlswede (2006)

Sergio Verdu (2007)
Robert M. Gray (2008)
Jorma Rissanen (2009)
Te Sun Han (2010)
Shlomo Shamai (Shitz) (2011)
Abbas El Gamal (2012)
Katalin Marton (2013)
János Körner (2014)
Arthur Robert Calderbank (2015)
Alexander S. Holevo (2016)
David Tse (2017)

[http://www.itsoc.org/honors/claude-e-shannon-award]

Berger plaque

Raymond Yeung

- BS, MEng and PhD degrees in electrical engineering from **Cornell** University in 1984, 1985, and 1988, respectively.