repetition code can't satisfy the requirement that the error probability must be less than 10^{-15}. In fact, Figure 17b shows that as we reduce the error probability to 0, the rate also goes to 0 as well. Therefore, there is no positive rate that works for all error probability.

- However, because the channel capacity is 0.531 [bpcu], there must exist other encoding techniques which give better error probability than repetition code.
    - Although Shannon's result gives us the channel capacity, it does not give us any explicit instruction on how to construct codes which can achieve that value.

4.3 Information Channel Capacity

4.23. In Section 4.1, we have studied how to compute the value of mutual information $I(X;Y)$ between two random variables X and Y. Recall that, here, X and Y are the channel input and output, respectively. We have also seen, in Example 4.14, how to compute $I(X;Y)$ when the joint pmf matrix $\mathbf{P}$ is given. Furthermore, we have also worked on Example 4.15 in which the value of mutual information is computed from the prior probability vector $\underline{\mathbf{p}}$ and the channel transition probability matrix $\mathbf{Q}$. This second type of calculation is crucial in the computation of channel capacity. This kind of calculation is so important that we may write the mutual information $I(X;Y)$ as $I(\underline{\mathbf{p}},\mathbf{Q})$.

Definition 4.24. Given a DMC channel, we define its "information" channel capacity as

$$C = \max_{\underline{\mathbf{p}}} I\left(X;Y\right) = \max_{\underline{\mathbf{p}}} I\left(\underline{\mathbf{p}},\mathbf{Q}\right), \tag{34}$$

where the maximum is taken over all possible input pmfs $\underline{\mathbf{p}}$.

- Again, as mentioned in 4.20, Shannon showed that the "information" channel capacity defined here is equal to the "operational" channel capacity defined in Definition 4.19.
    - Thus, we may drop the word "information" in most discussions of channel capacity.

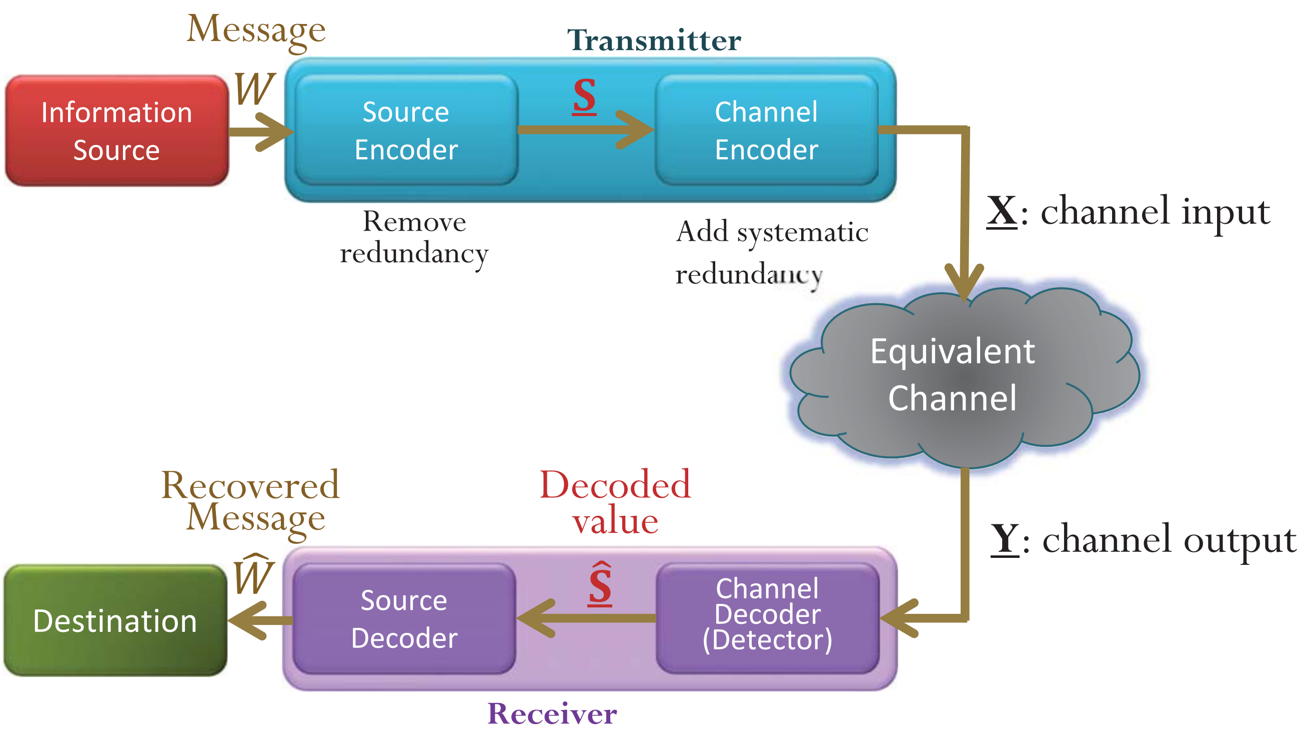
Digital Communication Systems
EES 452

Asst. Prof. Dr. Prapun Suksompong
prapun@siit.tu.ac.th

4. Mutual Information and Channel Capacity

4.3 Information Channel Capacity

System Model for Section 3.5



Channel Capacity

[Section 4.2]

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

→ Arbitrarily small error probability can be achieved.

Channel Capacity

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

[Section 4.3]

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

MATLAB

```
function H = entropy2s(p)
% ENTROPY2S 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;
```

$I(P,Q)$

Example 4.15. Find the mutual information $I(X;Y)$ between the two random variables X and Y whose $\underline{\mathbf{p}} = \left[\frac{1}{4}, \frac{3}{4}\right]$ and $\mathbf{Q} = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix}$.

Method 1: First, convert the given information into the joint pmf matrix.

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

$H(X) = H\left(\left[\frac{1}{4}\ \frac{3}{4}\right]\right) \approx 0.8113$

$H\left(\left[\frac{1}{16}\ \frac{3}{16}\ \frac{9}{16}\ \frac{3}{16}\right]\right) \approx 1.6226$

$$Q = \begin{bmatrix} 1/4 & 3/4 \\ 3/4 & 1/4 \end{bmatrix} \xrightarrow[\times 3/4]{\times 1/4} \begin{bmatrix} 1/16 & 3/16 \\ 9/16 & 3/16 \end{bmatrix} = P$$

$\Sigma \downarrow$ $10/16$, $\Sigma \downarrow$ $6/16$

$H(Y) = H\left(\left[\frac{5}{8}\ \frac{3}{8}\right]\right) \approx 0.9544$

Then, $I(X;Y) = H(X) + H(Y) - H(X,Y)$.

≈ 0.1432

Method 2: Use $I(X;Y) = H(Y) - H(Y|X)$.

(a) To find $H(Y)$, we need $q(y)$:

$$\underline{\mathbf{q}} = \underline{\mathbf{p}}\mathbf{Q} = \left[\frac{1}{4}, \frac{3}{4}\right] \begin{bmatrix} \frac{1}{4} & \frac{3}{4} \\ \frac{3}{4} & \frac{1}{4} \end{bmatrix} = \left[\frac{10}{16}, \frac{6}{16}\right] = \left[\frac{5}{8}, \frac{3}{8}\right].$$

This gives $H(Y) \approx 0.9544$.

(b) $H(Y|X) = \sum_x p(x) H(Y|x)$. So, we need $H(Y|x)$. Observe that each row of $\mathbf{Q}$ is $\left[\ \frac{1}{4}\ \ \frac{3}{4}\ \right]$. Therefore, ($X = x$)

$$H(Y|x) = H\left(\left[\ \tfrac{1}{4}\ \ \tfrac{3}{4}\ \right]\right) \approx 0.8113$$

for any x (for any row of $\mathbf{Q}$). This gives

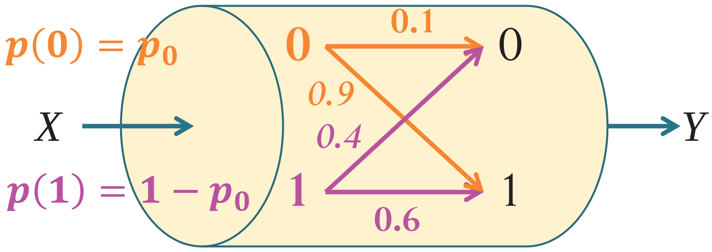
$$H(Y|X) = \sum_x p(x) H(Y|x) \approx \sum_x p(x) \times 0.8113$$
$$= 0.8113 \left(\sum_x p(x)\right) = 0.8113.$$

Finally,

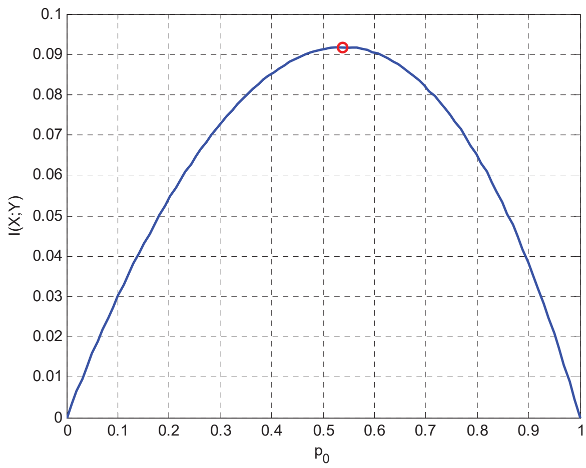
$$I(X;Y) = H(Y) - H(Y|X) \approx 0.1432.$$

[Ex. 4.25, Fig. 18]

Capacity calculation for BAC



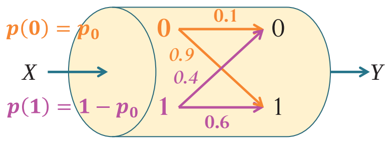
$$Q = \begin{bmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{bmatrix}$$



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

[Ex. 4.25, Fig. 18]

Capacity calculation for BAC



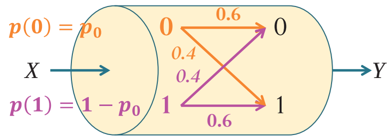
$$\mathbf{Q} = \begin{bmatrix} 0.1 & 0.9 \\ 0.4 & 0.6 \end{bmatrix}$$

```
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
```

[Ex. 4.25, Fig. 19]

Same procedure applied to BSC



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

```
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, Richard (1972), "Computation of channel capacity and rate-distortion functions", IEEE Transactions on Information Theory, 18 (4): 460–473

460 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-18, NO. 4, JULY 1972

Computation of Channel Capacity and Rate-Distortion Functions

RICHARD E. BLAHUT, MEMBER, IEEE

Abstract—By defining mutual information as a maximum over an appropriate space, channel capacities can be defined as double maxima and rate-distortion functions as double minima. This approach yields valuable new insights regarding the computation of channel capacities and rate-distortion functions. In particular, it suggests a simple algorithm for computing channel capacity that consists of a mapping from the set of channel input probability vectors into itself such that the sequence of probability vectors generated by successive applications of the mapping converges to the vector that achieves the capacity of the

Arimoto [13] used the first of the preceding expressions in an investigation of C, thereby obtaining Theorems 1 and 3 as well as Corollary 2 of this paper.[1]

This approach places the existing theory of C and $R(D)$ in a more transparent setting and suggests several new results. In particular, the approach in question results in algorithms for determining C and $R(D)$ by means of map-

Arimoto, Suguru (1972), "An algorithm for computing the capacity of arbitrary discrete memoryless channels", IEEE Transactions on Information Theory, 18 (1): 14–20

An Algorithm for Computing the Capacity of Arbitrary Discrete Memoryless Channels

SUGURU ARIMOTO

Abstract—A systematic and iterative method of computing the capacity of arbitrary discrete memoryless channels is presented. The algorithm is very simple and involves only logarithms and exponentials in addition to elementary arithmetical operations. It has also the property of monotonic convergence to the capacity. In general, the approximation error is at least inversely proportional to the number of iterations; in certain circumstances, it is exponentially decreasing. Finally, a few inequalities that give upper and lower bounds on the capacity are derived.

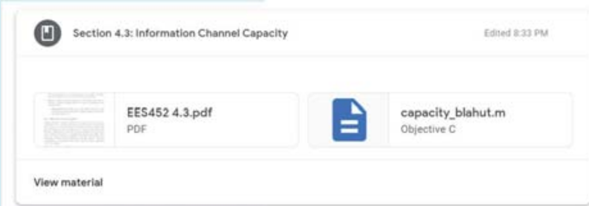
Manuscript received September 9, 1970.
The author is with the Faculty of Engineering Science, Osaka University, Osaka, Japan.

I. Introduction

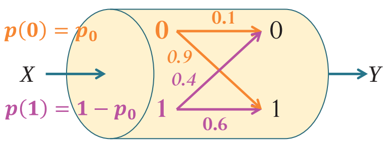
IT IS well known that the capacity of discrete memoryless channels that are symmetric from the input can easily be evaluated. Muroga [1] developed a method for straightforward evaluation of capacity, but unfortunately its usefulness is restricted to the case where 1) the channel

Blahut–Arimoto algorithm [4.26]

```
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 calculation for BAC: a revisit



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

```
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
```

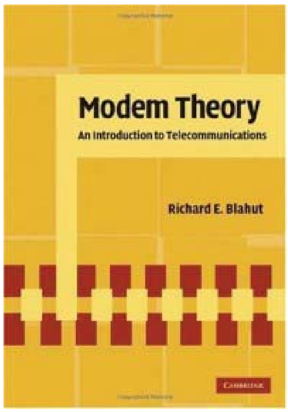
Prof. Toby Berger 80th Reunion Party

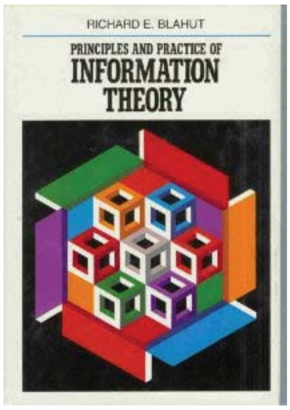
Toby Berger with Berger plaque

Richard Blahut

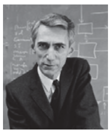
- Former chair of the Electrical and Computer Engineering Department at the University of Illinois at Urbana-Champaign
- Best known for **Blahut–Arimoto algorithm (Iterative Calculation of C)**

Toby Berger





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)
Gottfried Ungerboeck (2018)
Erdal Arıkan (2019)
Charles Bennett (2020)

[http://www.itsoc.org/honors/claude-e-shannon-award]
[https://en.wikipedia.org/wiki/Claude_E._Shannon_Award]

Example 4.25. The capacity of a BAC whose $Q(1|0) = 0.9$ and $Q(0|1) = 0.4$ can be found by first realizing that $I(X;Y)$ here is a function of a single variable: p_0. The plot of $I(X;Y)$ as a function of p_0 gives some rough estimates of the answers. One can directly solve for the optimal p_0 by simply taking derivative of $I(X;Y)$ and set it equal to 0. This gives the capacity value of 0.0918 bpcu which is achieved by $\underline{\mathbf{p}} = [0.5376,\ 0.4624]$.

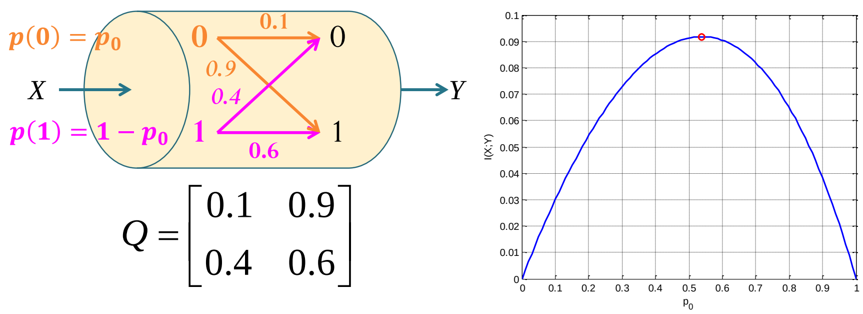


Figure 18: Maximization of mutual information to find capacity of a BAC channel. Capacity of 0.0918 bits is achieved by $\underline{\mathbf{p}} = [0.5376,\ 0.4624]$

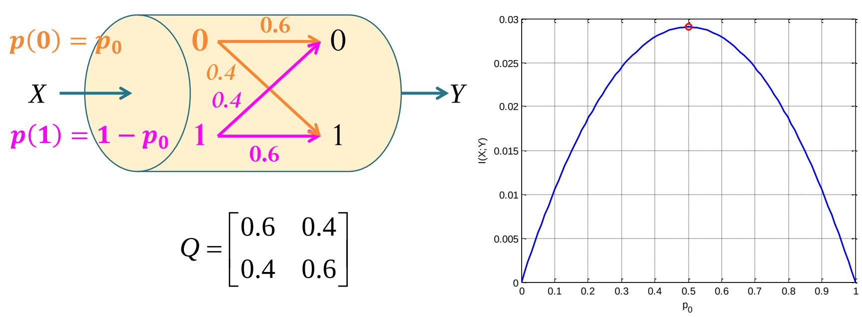


Figure 19: Maximization of mutual information to find capacity of a BSC channel. Capacity of 0.029 bits is achieved by $\underline{\mathbf{p}} = [0.5,\ 0.5]$

4.26. Blahut-Arimoto Algorithm [5, Section 10.8]: Alternatively, in 1972, Arimoto [1] and Blahut [2] independently developed an iterative algorithm to help us approximate the pmf $\underline{\mathbf{p}}^*$ which achieves capacity C. To do this, start with any (guess) input pmf $p_0(x)$, define a sequence of pmfs $p_r(x)$, $r = 0, 1, \ldots$ according to the following iterative prescription:

(a) $q_r(y) = \sum_x p_r(x) Q(y|x)$ for all $y \in \mathcal{Y}$.

(b) $c_r(x) = 2^{\left(\sum_y Q(y|x) \log_2 \frac{Q(y|x)}{q_r(y)}\right)}$ for all $x \in \mathcal{X}$.

(c) It can be shown that

$$\log_2\left(\sum_x p_r(x)\, c_r(x)\right) \le C \le \log_2\left(\max_x c_r(x)\right).$$

- If the lower-bound and upper-bound above are close enough. We take $p_r(x)$ as our answer and the corresponding capacity is simply the average of the two bounds.
- Otherwise, we compute the pmf

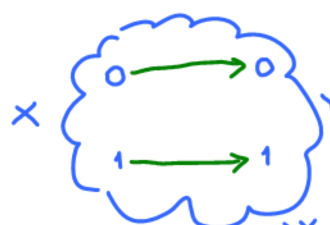
$$p_{r+1}(x) = \frac{p_r(x)\, c_r(x)}{\sum_x p_r(x)\, c_r(x)} \quad \text{for all } x \in \mathcal{X}$$

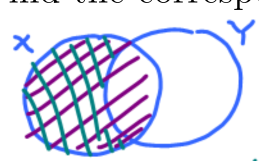
and repeat the steps above with index r replaced by $r+1$.

4.4 Special Cases for Calculation of Channel Capacity

In this section, we study special cases of DMC whose capacity values can be found (relatively) easily.

Example 4.27. Continue from Example 4.8 where we considered a noiseless binary channel. Find the corresponding channel capacity.





Example 4.28. Noisy Channel with Nonoverlapping Outputs: Find the channel capacity of a DMC whose

$$\mathbf{Q} = \begin{bmatrix} 1/8 & 7/8 & 0 & 0 \\ 0 & 0 & 1/3 & 2/3 \end{bmatrix}$$