MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science

6.441 Transmission of Information—Spring 2006

Scribe Notes

May 18, 2006

Contents

6.441 Transmission of Information February 7, 2006

Lecture 1

Lecturer: Madhu Sudan *Scribe: Elena Grigorescu*

1 Administrative Issues

Lecturer: Madhu Sudan, madhu@mit.edu
TA: Chung Chan, chungc@mit.edu
Website: http://theory.csail.mit.edu/∼madhu/ST06

Note: Visit the website in order to sign up for scribing and to fill up the questionnaire (if you didn't do it in class).

Class policy:

- 4 PSets. The first PSet is out today and will be due in about two weeks.
  Collaboration is encouraged, however the write-ups must be done separately and all the sources should be mentioned.
- 1 Midterm.
- 1 Project. The project consists in presenting one of the papers on the website, in teams of two. See more instructions online.
- 1 Scribe notes.

2 General Course Topics:

1. Mathematics of Information transmission
2. How to quantify information
3. How to quantify the 'capacity' of a communication channel
4. How to manipulate these quantities

We will use Probability Theory as a main mathematical tool in defining notions such as **information** and **entropy**.

3 Motivating Scenario

As an introductory example let us start with considering the following scenario. A space satellite is supposed to collect data and send it back to Earth. Let us assume that its sensor measures the surrounding temperature (say for now, as an integer) and the transmitter sends it to Earth at a rate 1 bit/time unit. However, the transmission is not perfectly accurate and therefore the received data is erroneous. The general question that we would like to answer is whether communication is feasible in this kind of settings. Let us model the above problem in a way that would allow us to make mathematical deductions.

3.1 A Simple Model

The Sensor:
Let $x_0, x_1, \ldots, x_t, x_{t+1} \ldots$ denote the measured temperatures at time $0, 1, \ldots$. For simplicity, we assume that each x_i is an integer. Since temperature is a continuous function, we do not expect drastic variations between consecutive measurements. Suppose that the following holds regarding consecutive measurements:

$$Pr[|x_{t+1} - x_t| \geq k] \leq 8^{-k}.$$

It follows that

$x_{t+1} = x_t$ w.p. $\frac{7}{8}$

$x_{t+1} = x_t + 1$ w.p. $\frac{7}{128}$

$x_{t+1} = x_t - 1$ w.p. $\frac{7}{128}$

$\vdots$

Our goal is to be able to transmit x_t for each i. Since we expect $|x_{t+1} - x_t|$ to be small, it is less costly to send $y_t = x_{t+1} - x_t$, rather than x_{t+1} at each time.

Therefore, we will need to send

$y_t = 0$ w.p. $\frac{7}{8}$

$y_t = +1$ w.p $\frac{7}{128}$

$y_t = -1$ w.p. $\frac{7}{128}$

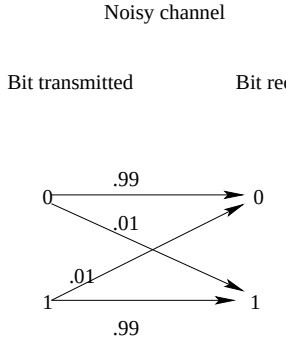
$y_t = +2$ w.p. etc.

$y_t = -2$ w.p. etc.

$\vdots$

All this data will be first encoded in some binary form (say, $0 \to 0$, $+1 \to 100$, $-1 \to 101$, $+2 \to 1100$, $-2 \to 1101$, etc.) and then sent through a noisy transmission channel.

The Transmission Channel:
Consider a transmission channel that flips a bit w.p .01, as described in the below figure. We will



moreover assume that the channel's decision to flip a bit at a certain moment in time is independent of its past or future behavior.

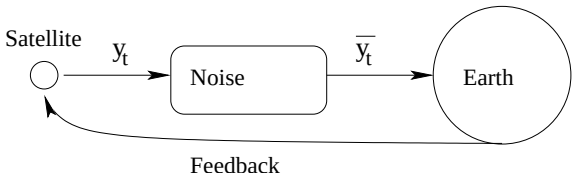
Let us first examine in what conditions data transmission across this channel is possible. A first question that we might ask is if we could send each y_i at one unit of time. This obviously cannot be done since the expected channel capacity is less than 1 bit/time, while the expected encoding length is greater than 1.

Instead, we could try and buffer the information that we get for say, 100 units of time and then send it. In analyzing the feasibility of this approach today we will be making crude assumptions and approximations, which however will be improved later in the course.

First, note that under the above premises, in the 100-bit sequence the expected number of 0's is 87 ($=\frac{7}{8}$.) Therefore, in order to specify the location of all these 0's we need $\log \binom{100}{87} \approx 53$ bits.

Similarly, the expected number of ± 1's is ≈ 11, and we need $\log \binom{13}{11} \approx 7$ bits to specify these positions, and 11 more bits to distinguish between $+1$ and -1. In addition, we should consider an expected additional cost of ≤ 3 per y_t.

Summing up, the total cost of about 77 bits seems feasible compared to the channel's capacity of 100 bits. However, we should take into account the fact that the channel is noisy and thus some sort of redundancies should be added in order to correctly decode from transmission error. For now, we will make some more simplifying assumptions about our model. Note that the expected error of the channel is of 1 bit, and assume that the channel actually makes exactly one error. Moreover, assume that the satellite can receive feedback from Earth with full accuracy, such that it can compute the position of the error and send it back to Earth. This will take $\log 100 \approx 7$ more bits. Even so, we are still in the limits of our capacity, which concludes the feasibility of this unrealistic model.



In future lectures we will see that the same results can be obtained however even when we do not make such optimistic assumptions as above.

4 Course highlights

Having introduced the motivating example of our topic, we can now be a bit more specific about the content of the course. The following are topics that we aim to approach.

- Review probability (today)
- Entropy and Information (next few lectures)
- Asymptotic Equipartition Property (the information theorists' Law of Large Numbers)
- Source coding (looks at the rate at which a source is producing information)
- Channel coding (looks at coding channels as the one described today: discrete channels)
- Continuous channels and Gaussian Error
- Network Information Theory (applications in other settings, such as stock markets, gambling, etc.)

References

1. Elements of Information Theory, T. Cover and J. Thomas - available on Reserve at the Barker Library and CSAIL Reading Room.

2. Course website, Scribe notes, Scribbled notes, Lectures notes from previous offerings.

5 Brief Review of Probability Theory

Probability Space: $(\Omega, \mathcal{F}, P)$, where Ω is an underlying ground set, $\mathcal{F}$ is the power set of Ω and P is a probability measure associated with events $E \in \mathcal{F}$.
To each probability space one can associate a **random variable** X distributed according to P. If Ω is a finite set, then $P(x) \geq 0$ for all $x \in \Omega$ and $\sum_{x \in \Omega} P(x) = 1$.
The **expectation** of a real valued random variable X is defined to be $E[X] = \sum_{x \in \Omega} xP(x)$.
The **indicator variable** of an event A is $1_A(x) = 1$ if $x \in A$ and $1_A(x) = 0$ otherwise. Thus, $E[1_A] = Pr[A]$.

The following are basic facts that we will be using extensively.

1. $Pr[E_1 \cup E_2] \leq Pr[E_1] + Pr[E_2]$.

2. $E[X_1 + X_2] = E[X_1] + E[X_2]$.

3. For a random variable $X \geq 0$, **Markov's inequality** states

$$Pr[X \geq kE[X]] \leq \frac{1}{k}.$$

Therefore,

$$Pr[(X - E[X])^2 \geq k^2 E[(X - E(X))^2]] \leq \frac{1}{k^2}.$$

The **variance** of X is defined as $Var[X] = E[X^2] - E[X]^2$. As an exercise, show that $Var[X] = E[(X - E(X))^2]$, and thus $Var[X] \geq 0$.

4. The latter inequality can be rewritten in a form known as **Chebychev's inequality**

$$Pr[|X - E[X]| \geq k\sqrt{Var[X]}] \leq \frac{1}{k^2}.$$

By definition, $\sigma[X] = \sqrt{Var[X]}$ is the **standard deviation** of X. As an exercise, figure out when $\sigma[X] = 0$.

5. **Conditional Probabilities:**

$$Pr[E_2|E_1] = \frac{Pr[E_1 \cap E_2]}{Pr[E_1]}.$$

6. One important concept in probability is that of **independence**. Two events E_1 and E_2 are independent if $Pr[E_2|E_1] = Pr[E_2]$.
Consider the following experiment called Random Decreasing Sequence. The sequence is such that, if the random number picked at index i was n_i then at index $i+1$ one picks a random number $n_{i+1} \leq n_i$. The sequence starts at $n_0 = 100$ and ends when $n_t = 1$ for some t. Let E_n be the event that the number n appears in this sequence. We can ask questions such as: What is $Pr[E_{10}]$ or $Pr[E_{11}]$? Are E_{10} and E_{11} independent? Give these questions some thought...

7. **Chernoff-Hoefding bound:** Given $X_1, \ldots, X_n$ identically and independently distributed (for short, i.i.d.), such that $X_i \in [0, 1]$ and $E[X_i] = \mu$, then

$$Pr[|\frac{\sum_{i=1}^{n} X_i}{n} - \mu| \geq \epsilon] \leq e^{\frac{-\epsilon^2 n}{2}}.$$

6.441 Transmission of Information Feb 7, 2005

Lecture 1

Lecturer: Madhu Sudan *Scribe: Guner Celik*

GENERAL COURSE INFORMATION

Lecturer:Madhu Sudan, madhu@mit.edu

TA: Chung Chan, chungc@mit.edu

Website: http://stellar.mit.edu/S/course/6/sp06/6.441/index.html

Text: Elements Of Information Theory by Cover and Thomas

- MIT Engineering Library
- CSAIL Reading Room
- Should you decide to buy, try the online retailers

Grading:

- 4 Problem Sets: First one out today, due approximately two weeks (encourage collaboration, but write-ups separate, mention all sources)
- 1 Midterm
- 1 Project
- 1 Script

Course Topics

- Mathematics of information transmission
- Quantify Information
- Quantify "capacity" of a channel
- How do you manipulate these quantities?

AN EXAMPLE OUTLINING THE CONCEPTS

- Satellite through space
- Sensor measuring temperature
- Transmitter beams back one bit/time (Transmission is not perfectly accurate)

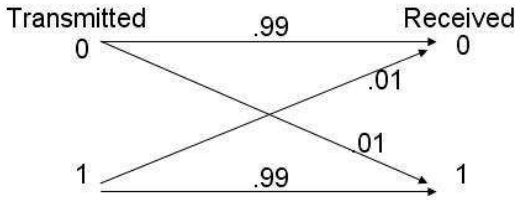
Is communication feasible?

Sensor: $X_0, X_1, X_2, ..., X_t, X_{t+1}$
X_t: Temperature at time t: integer

$Pr[|X_{t+1} - X_t| \geq k] \leq \delta^{-k}$

$$\begin{aligned} X_{t+1} \quad &= X_t \quad w.p.\ 7/8, \\ &= X_t + 1 \ w.p.\ 7/128, \\ &= X_t - 1 \ w.p.\ 7/128, \\ &\dots \end{aligned}$$

Transmission Channel:



Each time unit is independent of past and future.

$$\begin{aligned} Y_t = X_t - X_{t-1} \Rightarrow & \\ Y_t \quad &= \ 0 \ w.p.\ 7/8, \Rightarrow 0 \\ &= +1 \ w.p.\ 7/128 \Rightarrow 100 \\ &= -1 \ w.p.\ 7/128, \Rightarrow 101 \\ &= +2 \ w.p.\ ....., \Rightarrow 1100 \\ &= -2 \ w.p.\ ....., \Rightarrow 1101 \\ &\dots \end{aligned}$$

Then, $E[encoding\, length] > 1$ *and* $E[capacity] < 1\, bit/time$

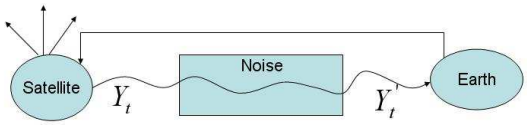
Idea1: Transmit Y_t each t : Didn't work
Idea2: Buffer information for 100 units of time

"Rate at which information is produced?"

-Expect to see 87 zeros $\Rightarrow log_2\binom{100}{87} \approx 53 bits$
-11 of these are +/- 1's $\Rightarrow log_2\binom{13}{11} \approx 7 bits$
-11 bits for +/- 1's.
-2 Symbols $Y_t's, |Y_t| \geq 2$
-Also Expected cost ≤ 3 bits per Y_t

$\Rightarrow$ Total Cost: 77 bits, seems good since we improved it from 100 bits.

Simplifying Assumption: Feedback
This feedback is assumed errorless, which is not a reasonable assumption. Since the probability of success in the channel is 0.99 we expect 1 bit error in every 100 bits (which is the same assumption as we did before, and it is also not very accurate) Therefore, there are 100 different possibilities for the



earth station to transmit this error back to satellite. As a result, at least $log_2 100$ bits are required for the feedback channel. Including the 77 bits from the previous analysis, we end up with approximately 84 bits which is lower than 100 bits, showing that the transmission with buffering of 100 bits may be successful as opposed to transmitting Y_t each t.

Under the light of this example, we can be more specific about the course topics.

Course Topics In More Detail

- Probability Theory Review (today)
- Entropy And Information (next few lectures)
- Asymptotic Equipartition Property (Information Theorists' Law of Large Numbers)
- Source Coding (Rate At Which A Source Is Producing Symbols)
- Channel Coding (Discrete Channels With Discrete Error)
- Continuous Channels (Gaussian Error)
- Network Information Theory (Application In Non-communication Setting; Stock Market, Gambling etc.)

REVIEW OF PROBABILITY THEORY

Probability Space: (Ω, F, P)
Underlying ground set:Ω
F: Power set of Ω
Events: $E \subseteq \Omega, E \in F$
If Ω is a finite set and X is a random variable $\sim$ (distributed according to) P then $P(X) \geq 0 \, for \, every \, x \in \Omega$, $\sum_{x \in \Omega} P(x) = 1$

Real Valued Random Variable X$\sim$P, then expected value of X is defined as: E(X)=$\sum_{x \in \Omega} xP(x)$

Expectation of Real Valued R.V. $\Leftrightarrow$ Probability Of Events

Indicator Random Variable

E: event$\Rightarrow$

$$\begin{aligned} 1_E(X) \quad &= 1 \;\; if X \in E \\ &= 0 \, otherwise \end{aligned}$$

$\Rightarrow E[1_E] = Pr[E]$

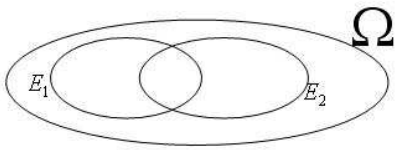
Manipulation Tools

1. $Pr(E_1 \bigcup E_2) \leq Pr(E_1) + Pr(E_2),\ \ E(X_1 + E_2) = E(X_1) + E(X_2)$
2. Expectation $\Rightarrow$ Probability :$X \geq 0$, $Pr[X \geq kE(X)] \leq \frac{1}{k}\ \ Markov's\, Inequality$
3. $Pr[(X - E(x))^2 \geq k^2 E[(X - E(X))^2] \leq \frac{1}{k^2}\ \Rightarrow$

   By definition, $Var(X) = E(X^2) - E(X)^2$ then

   $Pr[|X - E(X)| \geq k\sqrt{Var(X)}] \leq \frac{1}{k^2}\ also\ \sqrt{Var(X)} = \sigma(X) : StandartDeviation$

Conditional Probabilities



Event E_1 has occurred, how does this change the probability space?

$$\Omega \rightarrow E_1$$

$$Pr(E_2/E_1) = \frac{Pr(E_1 \bigcap E_2)}{Pr(E_1)}$$

Independence: E_1 and E_2 are independent if $Pr(E_1/E_2) = Pr(E_1)$

Example: Random Decreasing Sequence: 1,2,3,.....,100 → pick a number say 54, then 1,2,.....,54 → pick a number, say 27, then 1,2,....,27 → pick a number and decrease the sequence and so on.

E_{10} = Event that the number 10 appears in this sequence; Pr(E_{10})
E_{11} = Event that the number 11 appears in this sequence; Pr(E_{11})
Question: Are these events independent?

Joint Distributions on (X,Y) One can find the marginal distributions from the joint distribution.

Chernoff-Hoefding Bound

$X_1, X_2, ..., X_n$ are iid (independent identically distributed)

$$X_i \in [0,1]\ with\ E(x) = \mu$$

$$Pr[|\frac{\sum_{k=1}^{n} X_i}{n} - \mu| \geq \epsilon] \leq e^{\frac{-\epsilon^2 n}{2}} \qquad \epsilon = \frac{1}{n^{\frac{1}{3}}}$$

6.441 Transmission of Information	Feb 9, 2006
Lecture 2	
Lecturer: Madhu Sudan	*Scribe: Cy Chan*

1 Administrivia

- Questionnaire - get form from web and fill out
- Scribing - sign up on website
- Mailing List - if you haven't received an email already, tell staff
- Problem Set 1 - due 2/22/06

2 Introduction

- **Entropy** - associated with a random variable (RV) and quantifies the amount of uncertainty associated with that RV
- **Information** - associated with a pair of RVs:

$$I(X;Y) = \text{how much } Y \text{ informs us about } X$$

3 Entropy

3.1 Example

Let X, Y, and W be RVs where:

$$X = \begin{cases} 0 & \text{with probability } 1/2 \\ 1 & \text{with probability } 1/2 \end{cases}$$

$$Y = \begin{cases} 0 & \text{with probability } 7/8 \\ 1 & \text{with probability } 1/8 \end{cases}$$

$$W = \begin{cases} 0 & \text{with probability } 9/10 \\ 1 & \text{with probability } 1/20 \\ 2 & \text{with probability } 1/20 \end{cases}$$

Intuitively, X is more random than Y, but how do we make a comparison between Y and W? We need a way of quantifying the amount of randomness in each RV.

3.2 Derivation of Entropy $H(Z)$ for Bernoulli RV Z

Define Z to be a Bernoulli RV with parameter p:

$$Z = \begin{cases} 0 & \text{with probability } 1-p \\ 1 & \text{with probability } p \end{cases}$$

How many bits are required to convey the value of Z? If we only communicate a single instance of Z, we must send at least 1 bit, but if we are sending many instances, we can batch the values as in the previous lecture and achieve an average of less than 1 bit per value.

Suppose we have a sequence $Z_1, Z_2, \ldots, Z_n$ of n independent, identically distributed (IID) RVs each with the same distribution as Z above. We prescribe the following algorithm to encode a sequence $z_1, z_2, \ldots, z_n$ drawn from this distribution:

1. Send $k = \sum_{i=1}^{n} z_i$, which takes k bits (takes $\log_2 n$ bits)
2. Create a table T_k that describes every sequence with k 1's and $(n-k)$ 0's
3. Send the index in the table that describes the sequence $z_1, z_2, \ldots, z_n$ (takes $\log_2 \binom{n}{k}$ bits)

We can write the expected length of the resulting encoding as

$$l = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \log_2 \binom{n}{k} + \log_2 n.$$

To simplify, we make use of the law of large numbers, from which we get

$$\Pr\left[k \notin [(p-\epsilon)n, (p+\epsilon)n]\right] \leq 2^{-\epsilon^2 n}.$$

Then,

$$l = \sum_{k=(p-\epsilon)n}^{(p+\epsilon)n} \binom{n}{k} p^k (1-p)^{n-k} \log_2 \binom{n}{k} + \sum_{k \notin [(p-\epsilon)n, (p+\epsilon)n]} \Pr[k \text{ is such}] \binom{n}{k} + \log_2 n.$$

Since $\Pr\left[k \notin [(p-\epsilon)n, (p+\epsilon)n]\right] \binom{n}{k} \leq 2^{-\epsilon^2 n} n$, the second term becomes vanishingly small as n gets large. Similarly, the third term $\log_2 n$ vanishes when we divide by n when taking the average encoding length per value. In the first term we note the following:

$$\sum_{k=(p-\epsilon)n}^{(p+\epsilon)n} \binom{n}{k} p^k (1-p)^{n-k} \leq \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} = 1,$$

and for each term in the summation, $k \approx pn$, so

$$\sum_{k=(p-\epsilon)n}^{(p+\epsilon)n} \binom{n}{k} p^k (1-p)^{n-k} \log_2 \binom{n}{k} \leq 1 \cdot \log_2 \binom{n}{pn}.$$

Stirling's approximation for $n!$ implies

$$\binom{n}{pn} \approx \left(\frac{1}{p}\right)^{pn} \left(\frac{1}{1-p}\right)^{(1-p)n},$$

so for large n:

$$l \leq \log_2 \binom{n}{pn} \approx \log_2 \left[\left(\frac{1}{p}\right)^{pn} \left(\frac{1}{1-p}\right)^{(1-p)n}\right]$$

$$l \leq n \left[p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}\right].$$

The entropy $H(Z)$ is the average encoding length per value:

$$H(Z) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}.$$

3.3 Extensions to Non-Bernoulli discrete RVs

What if we have a RV that takes N values? Consider a RV Z that takes values in $\{1, 2, \ldots, N\}$, where $p_i = \Pr[Z = i]$. We define two new RVs, Z_1 and Z_2, where

$$Z_1 = \begin{cases} 0 & \text{if } Z = 1 \\ 1 & \text{otherwise} \end{cases}$$

$$\Pr[Z_1 = 0] = p_1, \text{ and } \Pr[Z_1 = 1] = 1 - p_1,$$

$$Z_2 = Z | \{Z_1 = 1\}$$

$$\Pr[Z_2 = i] = \frac{p_i}{1 - p_1}, \text{ for } i \in \{2, 3, \ldots, N\}.$$

We can show that

$$H(Z) = H(Z_1) + \Pr[Z_1 = 1] H(Z_2),$$

and by induction that

$$H(Z) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}.$$

Note that we would have gotten the same answer no matter how we partition the sequence and assign new random variables.

3.4 Properties of Entropy

The entropy function satisfies the following three properties:

1. $H(p_1, p_2, \ldots, p_N)$ is symmetric in its arguments
2. $H(p_1, p_2, \ldots, p_N) = H(p_1, 1 - p_1) + (1 - p_1) H(\frac{p_2}{1-p_1}, \frac{p_3}{1-p_1}, \ldots, \frac{p_N}{1-p_1})$
3. $H(p_1, p_2, \ldots, p_N) \leq \log_2 N$

In property 3, the inequality is strict unless $p_i = \frac{1}{N}$ for all i. In other words, maximum entropy occurs when the probability mass is evenly distributed. For probability functions with unbounded support, it is possible to have unbounded entropy. For example, over all densities on the real line, the density that maximizes entropy (indeed the differential entropy) for a given variance is a gaussian distribution. For densities over positive reals with a given mean, the entropy maximizing density is the exponential distribution. This is because the square of a zero mean Gaussian random variable is exponentially distribution with the mean equal to its variance.

Other functions may satisfy the above three requirements, but if we change property 3 to

3′. $H(\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}) = \log_2 N$,

then properties 1, 2, and 3′ imply our specific entropy function $H(Z) = \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i}$.

3.5 Joint and Conditional Entropy

We can extend our definition of entropy to include joint distributions of RVs. If we have a pair of RVs (X, Y) with density $P(X, Y)$ over $\Omega_x \times \Omega_y$, we define the joint entropy as:

$$H(X,Y) = \sum_{x \in \Omega_x, y \in \Omega_y} P(X = x, Y = y) \log_2 \frac{1}{P(X = x, Y = y)}.$$

We define conditional entropy $H(X|Y)$ as the average (over Y) entropy of X given Y:

$$H(X,Y) = \sum_{y \in \Omega_y} P_y(y)\ H(X|Y = y).$$

Intuitively, we sense that $H(X)$ should be no smaller than $H(X|Y)$, which we will prove next lecture. In the satellite example in the previous lecture, if X is the satellite transmission and Y is what Earth received, $H(X|Y)$ is the number of bits necessary to fix the errors.

4 Information

How much information does Y give about X (and vice versa)? First, we state the **chain rule of entropy**:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y).$$

Rearranging the terms, we define the quantity

$$I(X;Y) \triangleq H(X) - H(X|Y) = H(Y) - H(Y|X)$$

as the **mutual information** between X and Y. Since $H(X|Y) \leq H(X)$, the mutual information is always non-negative. As an example, consider tossing 10 coins and letting X be the values of the first 7 coins and Y be the value of the last 5 coins. Then

$$\begin{gathered} H(X) = 7 \text{ and } H(Y) = 5 \\ H(X|Y) = 5 \text{ and } H(Y|X) = 3 \\ I(X;Y) = I(Y;X) = 2. \end{gathered}$$

6.441 Transmission of Information Feb 14, 2006

Lecture 3

Lecturer: Madhu Sudan *Scribe: Daniel Kim (dskim116)*

1 Today's outline

- Property of information and entropy
- New notions: KL divergence, markov chains
- results: non-negativity of mutual information, data processing inequality, Fano's inequality

2 Lecture 2's Review

Let us define marginal and joint distributions. $p(x)$ denotes a marginal probability that $X = x$, $p(y)$ denotes a marginal probability that $Y = y$ and $p(x, y)$ denotes a joint probability that $X = x$ and $Y = y$.

- **Entropy**:
$$H(X) = -\sum_x p(x) \log p(x)$$
- **Conditional entropy**:
$$H(X|Y) = \sum_{y \in \Omega_y} p_y(y) H(X|Y = y) = \sum_{x,y} p(x, y) \log \frac{p_y(y)}{p(x, y)}$$
- **Mutual information**:
$$I(x, y) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x) \cdot p(y)} = I(y, x)$$
- **Chain rule**:
$$H(x, y) = H(x) + H(y|x)$$
Applying this iteratively, we derive:
$$\begin{aligned} H(x_1, x_2, \cdots, x_n) &= H(x_1) + H(x_2|x_1) + \cdots \\ &= \sum_{i=1}^{n} H(x_i|x_1, x_2, \cdots, x_{i-1}) \end{aligned}$$

3 Is $I(X, Y) \geq 0$?

Proving $I(X, Y) \geq 0$ is equivalent to proving that $H(X|Y) \leq H(X)$.

$$I(x, y) = \sum_{x,y} p(x, y) \log \frac{p(x, y)}{p(x) \cdot p(y)} = E\left[\log \frac{p(x, y)}{p(x) \cdot p(y)}\right] \geq 0$$

with equality when x and y are independent because:

$$p(x, y) = p(x) \cdot p(y) \Longrightarrow I(x, y) = 0$$

Before we prove Claim 3, let us define function convexity and state Jensen's Inequality.

Definition 1 *Function f is **convex** when either of following conditions holds:*

$$\begin{cases} f & : \mathbb{R} \to \mathbb{R} \text{ is convex if } f''(x) \geq 0 \ \forall x \\ f & : \mathbb{R} \to \mathbb{R} \text{ is strictly convex if } f''(x) > 0 \ \forall x \end{cases}$$

For example, x^2, e^x and $-\log x$ are convex functions.

Theorem 2 *Jensen's Inequality: $E[f(z)] \geq f[E[z]]$ provided f is convex.*

Now, here is the claim.

Claim 3 $E_{(x,y)\sim p}\left[\log \frac{p(x,y)}{q(x,y)}\right] \geq 0$ *with equality when $p(x,y) = q(x,y)$.*

Proof Let us define new variable $z = \frac{q(x,y)}{p(x,y)}$. Then,

$$\begin{aligned} E_{(x,y)\sim p}\left[\log \frac{p(x,y)}{q(x,y)}\right] &= E_z\left[\log \frac{1}{z}\right] \\ &= E\left[-\log z\right] \\ &\geq -\log E[z] (\because \text{Jensen's Inequality}) \\ &= -\log\left[E_{(x,y)}\left[\frac{q(x,y)}{p(x,y)}\right]\right] \\ &= -\log\left[\sum_{x,y} p(x,y)\frac{q(x,y)}{p(x,y)}\right] = -\log\left[\sum_{x,y} q(x,y)\right] = -\log 1 = 0. \end{aligned}$$

■

Here, note that $E\left[\log \frac{p(x,y)}{q(x,y)}\right]$ shows how much similarity $q(x,y)$ and $p(x,y)$ share.

4 Relative Entropy

Definition 4 *The **relative entropy** or **Kullback-Liebler distance** between two probability mass functions $p(z)$ and $q(z)$ is defined as:*

$$D(p||q) = \sum_z p(z) \log \frac{p(z)}{q(z)}.$$

4.1 Example

Let us consider the case when $x \in \{0, 1\}$ with following distributions:

$$p : X = \begin{cases} 0 & \text{with probability 1} \\ 1 & \text{with probability 0} \end{cases}$$

$$q : X = \begin{cases} 0 & \text{with probability } 1/2 \\ 1 & \text{with probability } 1/2 \end{cases}$$

Based on the above scenario, we get $D(p||q) = \log 2$ and $D(q||p) = \infty$.

4.2 Compression motivation example

Let us consider our satellite example with $x \sim p = (p_1, p_2, \cdots, p_N)$. Optimal compression should require $\left\lceil \log \frac{1}{p_i} \right\rceil$ bits long string. x with distribution q would require $\left\lceil \log \frac{1}{q} \right\rceil$ bits long string. By definition, average inefficiency of compressing by q when given distribution is p is $D(p||q)$.

4.3 Basic Property

- $D(p||q) \geq 0$ with equality only when $p = q$
- $I(X,Y) = D(p(x,y)||p(x) \cdot p(y)) \geq 0$
- $I(X,Y) = H(X) - H(X|Y) \geq 0$ ($\because$ conditioning reduces entropy)
- $H(X_1, X_2, \cdots, X_n) = H(X_1) + H(X_2|X_1) + H(X_3|(X1, X2)) + \cdots$
  Substituting the following:

$$\begin{aligned} &H(X_1) \leq H(X_1) \\ &H(X_2|X_1) \leq H(X_2) \\ &H(X_3|(X_1, X_2)) \leq H(X_3) \\ &\vdots \end{aligned}$$

  we can reduce it to:

$$\therefore H(X_1, X_2, \cdots X_n) \leq \sum_n H(X_n).$$

- $H(x) = \log(|\Omega_x|) - D(p||U)$ where U is uniform distribution on Ω_x. Because $D(p||q) \geq 0$, we derive that $H(x) \leq \log(|\Omega_x|)$.

4.4 Is entropy concave?

In order to prove whether entropy is concave or not, we need to show following:

$$H(\lambda p + (1-\lambda)q) \geq \lambda H(p) + (1-\lambda)H(q) \tag{1}$$

Proof Let us assume that $x \sim p$ and $y \sim q$ on set Ω. Also, let us define another variable b with following distribution.

$$b = \begin{cases} 0 & \text{with probability } \lambda \\ 1 & \text{with probability } 1-\lambda \end{cases}$$

Using these variables, let us define a new variable Z with following distribution :

$$Z : \text{if } b = 0 \text{ then } x; \text{ else } y.$$

Then, the left-hand side of Equation (1) is reduced to $H(Z)$ and the right-hand side of Equation (1) is reduced to $H(Z|b)$. Because conditioning reduces the uncertainty, $H(Z) \geq H(Z|b)$. This proves that the entropy is concave. ■

5 Data Processing Inequality (Markov Chain)

Let us consider three states, X, Y, and Z. $X \to Y \to Z$ forms a Markov chain if and only if X and Z are conditionally independent given Y. Let us put the definition into mathematical term. $X \to Y \to Z$ forms a Markov chain if and only if either of following conditions is true:

$$p_{Z|(X,Y)}(z|(x,y)) = p_{Z|Y}(z|y)$$

or

$$p_{(X,Z)|Y}((x,z)|y) = p_{X|Y}(x|y) \cdot p_{Z|Y}(z|y)$$

Also, $X \to Y \to Z \iff Z \to Y \to X$. Now let us consider the property of Markov chain.

Claim 5 *If $X \to Y \to Z$, then $I(X,Z) \leq I(X,Y)$.*

Proof

$$\begin{aligned} I(X,(Y,Z)) &= I(X,Z) + I((X,Y)|Z) \\ &= I(X,Y) + I((X,Z)|Y) \end{aligned}$$

Substituting the fact that $I((X,Z)|Y) = 0$ and $I((X,Y)|Z) \geq 0$, we get $I(x,z) \leq I(x,y)$. ■

6 Fano's Inequality

Let E be an event and let P_e denote the probability when $X \neq \widetilde{X}$.

Theorem 6 *When $H(X|Y)$ is large,*

$$P_e \geq \frac{H(X|Y) - 1}{\log |\Omega_x|}.$$

Proof

$$\begin{aligned} H((E,X)|Y) &= H(X|Y) + H(E|(X,Y)) \\ &= H(E|Y) + H(X|(E,Y)) \end{aligned}$$

Let us take a look at each term:

$$\begin{aligned} H(E|(X,Y)) &= 0 \\ H(E|Y) &\leq H(P_e) \end{aligned}$$

$$\begin{aligned} H(X|(E,Y)) &= P_e \cdot H(X|(E=1,Y)) + (1-P_e)H(X|(E=0,Y)) \\ &= P_e \cdot H(X|(E=1,Y)) \\ &\leq P_e \log(|\Omega_x| - 1) \end{aligned}$$

Substituting these into original equation, we prove the theorem. ■

6.441 Transmission of Information	Feb. 16, 2006
Lecture 4	
Lecturer: Madhu Sudan	*Scribe:* Costas Pelekanakis (gas)

1. Today's outline

a. Asymptotic Equipartition Property (A.E.P.)
b. Typical sets
c. Application to data compression

2. Review of last lecture

Notions:

- $H(x), H(x/y), I(x;y), I(x;y/z)$
- $D(p//q) = \sum_x p(x) \log \frac{p(x)}{q(x)}$ which is measure of the inefficiency of assuming that the distribution of x is q(x) when the true distribution is p(x).
- Markov chain: $X \to Y \to Z$ or equivalently $p_{x,y/z} = p_{x/y} p_{z/y}$

Results:

- $D(p//q) \geq 0$ with equality when $p(x) = q(x)$
- $I(x;y) = D(p_{x,y} // p_x p_y) \geq 0$
- $H(x/y) = H(x) - I(x;y) \leq H(x)$
- chain rule: $H(x_1,...,x_n) \leq \sum_{i=1}^{n} H(x_i)$
- If $X \to Y \to Z$ then $I(x;z) \leq I(x;y)$
- If $X \to Y \to \hat{X}$ then Fano's inequality: $\Pr(X \neq \hat{X}) = P_e \geq \frac{H(x/y) - 1}{\log|\Omega_x|}$

2.1. Review of Fano's inequality

We give two examples which show that Fano's inequality can be either weak or tight.

2.1.1. Example 1

x is uniformly distributed over the set of binary n-tuples and y takes values from the set of binary n/2-tuples. I claim no matter distribution I pick for y, $H(x/y) \geq n/2$.

We have:

$$H(x) = \log|\Omega_x| = \log_2 2^n = n$$
$$H(y) \leq \log|\Omega_y| = \log_2 2^{n/2} = n/2$$
$$H(x/y) = H(x) - I(x;y) = n - I(x;y)$$
$$I(x;y) = H(y) - H(y/x) \leq H(y) \leq n/2$$

Thus, $H(x/y) \geq n/2$.

Fano's inequality yields (assuming big n):

$$P_e \geq \frac{H(x/y)-1}{\log|\Omega_x|} \approx \frac{n/2}{n} = 0.5$$

A better bound on P_e in this case is:

$$P(\text{correct decoding}) \leq \frac{2^{n/2}}{2^n} \Rightarrow P(\text{error}) \geq 1 - \frac{2^{n/2}}{2^n}$$

In this example Fano's inequality is very weak!

2.1.2. Example 2

x,y are distributed as follows: with probability p, x=y, and x,y are uniformly distributed over the set of binary m-tuples and with probability 1-p, x is uniformly distributed over the set of binary n-tuples and y is a constant.

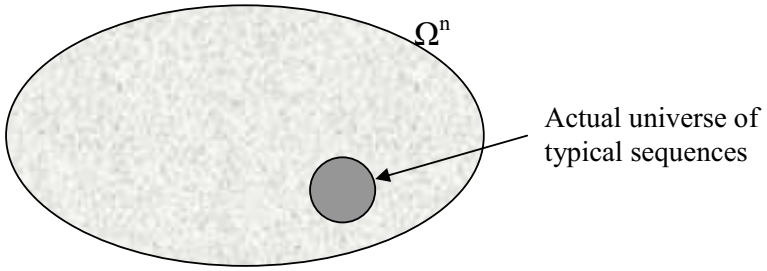
This is the picture of an erasure channel. The best strategy would decode as follows: observe y and assume that this is what it was sent. Obviously $P_e = p$ in this case. Fano's inequality yields:

$$H(x/y) = p \cdot H(x/y = const) + (1-p) \cdot H(x/y = x) = p \cdot n + 0 = p \cdot n$$
$$P_e \geq \frac{p \cdot n - 1}{n} \approx p$$

3. Typical sets

We want to answer the following question: if $x_1,...,x_n$ are iid and $x_i \sim p(x)$, i=1...n, what is the probability of the sequence $(x_1,...,x_n)$ to occur an n goes large? This will lead us to divide

the set of the sequences into two sets, the typical set, which contains the "highly likely to occur" sequences and the non-typical set which contains all the other sequences.



We will use the law of large numbers to answer the above question.

3.1. A.E.P. Lemma

If $x_1,...,x_n$ are i.i.d. according to p(x) then $-\frac{\log p(x_1...x_n)}{n} \to H(x)$ *in probability. In other words: for every ε>0, δ>0 there exists $n_o(\delta,\varepsilon)$ such that for every $n>n_o(\delta,\varepsilon)$ the* $\Pr\{H(x)-\varepsilon \le -\frac{\log p(x_1...x_n)}{n} \le H(x)+\varepsilon\} \ge 1-\delta$. *(Actually δ goes to 0 as $\exp(-n\varepsilon^2)$).*

Proof: Note that $p(x_1,...,x_n)$ is the probability of observing the sequence $(x_1,...,x_n)$. We have that x_i are iid so:

$$\frac{1}{n}\log p(x_1...x_n) = \frac{1}{n}\log\prod_{i=1}^{n} p(x_i) = \frac{1}{n}\sum_{i=1}^{n}\log p(x_i)$$

Let us call a new r.v. $z_i = -\log p(x_i)$. The z_i's are also i.i.d. and $E[z] = H(x)$. Applying the law of large numbers we have:

$$-\frac{1}{n}\sum_{i=1}^{n}\log p(x_i) = \frac{1}{n}\sum_{i=1}^{n} z_i \xrightarrow{LLN} E[z] = H(x)$$

Rewriting the above result we note that the probability to observe a sample sequence $(x_1,...,x_n)$ is bounded as: $2^{-n(H(x)+\varepsilon)} \le p(x_1,...,x_n) \le 2^{-n(H(x)-\varepsilon)}$. This motivates the definition of the typical set.

3.2. Definition: Typical set

$$A_\varepsilon^{(n)} = \{(x_1,...,x_n) \mid 2^{-n(H(x)+\varepsilon)} \le p(x_1,...,x_n) \le 2^{-n(H(x)-\varepsilon)}\}$$

3.3. Typical set theorem

i) $\Pr\{A_\varepsilon^{(n)}\} \ge 1-\delta$

ii) $\left|A_\varepsilon^{(n)}\right| \le 2^{n(H(x)+\varepsilon)}$

iii) $\left|A_\varepsilon^{(n)}\right| \ge (1-\delta)2^{n(H(x)-\varepsilon)}$

Proof:

i) A.E.P. Lemma

ii) $1 \ge \Pr\{A_\varepsilon^{(n)}\} \ge \left|A_\varepsilon^{(n)}\right| \cdot 2^{-n(H(x)+\varepsilon)} \Rightarrow \left|A_\varepsilon^{(n)}\right| \le 2^{n(H(x)+\varepsilon)}$

iii) $\left|A_\varepsilon^{(n)}\right| \cdot 2^{-n(H(x)-\varepsilon)} \ge \Pr\{A_\varepsilon^{(n)}\} \ge 1-\delta \Rightarrow \left|A_\varepsilon^{(n)}\right| \ge (1-\delta) \cdot 2^{n(H(x)-\varepsilon)}$

3.4. Example

Let $z = \begin{Bmatrix} 0 & \text{w.p. } 9/10 \\ 1 & \text{w.p. } 1/20 \\ -1 & \text{w.p. } 1/20 \end{Bmatrix}$

We expect the typical sequences $(z_1,...,z_n)$ to contain $\frac{9}{10}n(1\pm\varepsilon')$ "zeros", $\frac{1}{20}n(1\pm\varepsilon')$ "ones" and $\frac{1}{20}n(1\pm\varepsilon')$ "minus ones". Furthermore, $\left|A_\varepsilon^{(n)}\right| \approx 2^{n(H(z)\pm\varepsilon')}$

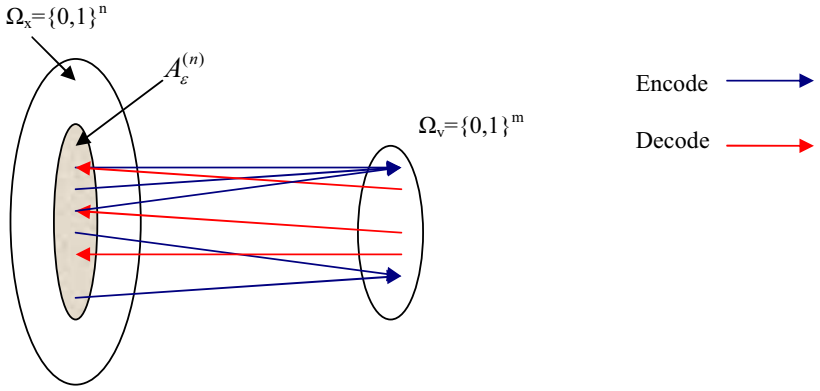
4. Application: Data compression

Compression is a mapping (function) of a higher dimensional space onto a lower one. Suppose we want to map the set Ω_x of binary n-tuples to the set Ω_y of the binary m-tuples with m«n. Obviously, the mapping is not "1-1" so errors will occur during decoding. We divide Ω_x into two sets: the $A_\varepsilon^{(n)}$ and its complement. We are computing the following quantity:

$$\Pr\{\text{decoding correctly}\} = \underbrace{\Pr\{\text{decoding correctly}/\Omega_x \setminus A_\varepsilon^{(n)}\}}_{\delta} + \Pr\{\text{decoding correctly}/A_\varepsilon^{(n)}\}$$

$$= \delta + \Pr\{(x_1,\ldots,x_n) \in A_\varepsilon^{(n)} \text{ and } (x_1,\ldots,x_n) \in \text{image of decoder}\}$$

$$\leq \delta + \left|\Omega_y\right| \max(\Pr\{(x_1,\ldots,x_n) \in A_\varepsilon^{(n)}\}) = \delta + 2^m \cdot 2^{-n(H(x)-\varepsilon)} \leq \delta + 2^m \frac{2^{2n\varepsilon}}{\left|A_\varepsilon^{(n)}\right|}$$



6.441 Transmission of Information February 23, 2006

Lecture 5

Lecturer: Madhu Sudan *Scribe: Hyun Sung Chang*

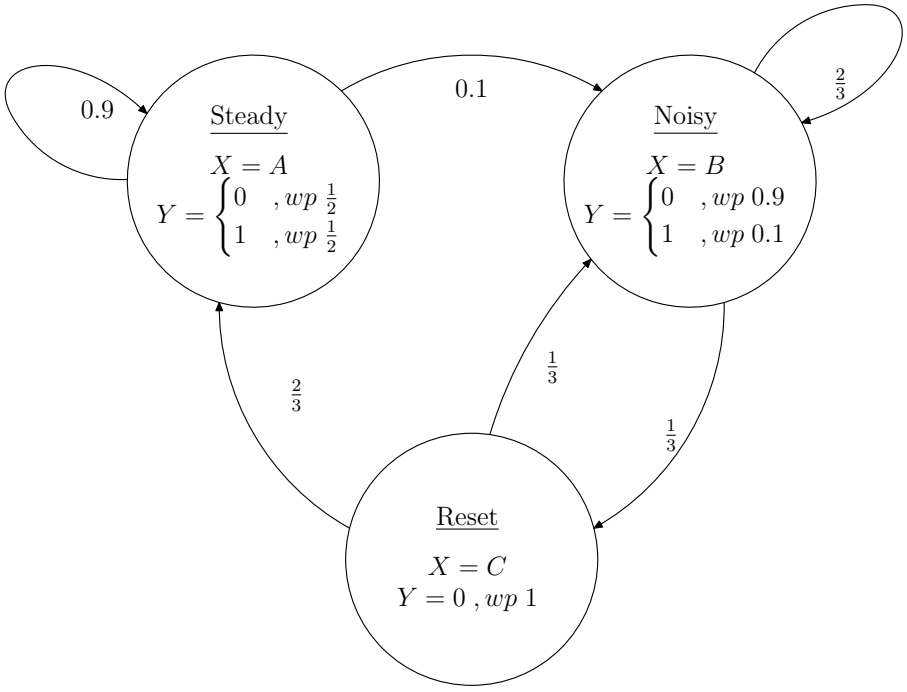
1 Introduction

1.1 Today's Topic

- Markov chains/processes
- Entropy rate of Markov chain

1.2 Motivating Example

Example 1: Let us start by considering the following example. What are the rates of X and Y?



2 Stochastic Process

A stochastic process can be viewed as an infinite sequence of random variables, e.g., X_{-n}, X_{-n+1}, $\cdots$, X_0, X_1, X_2, $\cdots$, X_n, $\cdots$, whose distribution may be expressed by

$$\Pr[X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n] \sim p(x_1, \cdots, x_n).$$

There are some meaningful and restricted classes of stochastic process.

Definition 1 (Stationary Process) *$\langle X_n \rangle_n$ is a stationary process if*

$$\Pr[X_1 = x_1, \cdots, X_n = x_n] = \Pr[\underbrace{X_{1+l} = x_1, \cdots, X_{n+l} = x_n}_{\text{time shift by } l}], \ \forall n, l, x_1, \cdots, x_n.$$

Definition 2 (Markov Process/Markov Chain) *$\langle X_n \rangle_n$ is a Markov chain if*

$$\Pr[X_n = x_n | X_1 = x_1, \cdots, X_{n-1} = x_{n-1}] = \Pr[X_n = x_n | X_{n-1} = x_{n-1}],\ \forall n, x_1, \cdots, x_n.$$

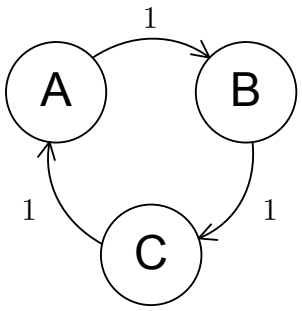
If $X_i \in \Omega$ and Ω is finite, then $\Pr[X_n = x_n | X_{n-1} = x_{n-1}]$ is just $|\Omega|^2$ entries for every n. But, can we describe it in finite terms? *No.*

Definition 3 (Time Invariant Markov Chain) *Markov Chain is time-invariant if*

$$\Pr[X_n = a | X_{n-1} = b] = \Pr[X_{n+l} = a | X_{n+l-1} = b],\ \forall n, l, a, b \in \Omega.$$

Time invariant Markov chain can be specified by distribution on X_0 and probability transition matrix $\boldsymbol{P} = [P_{ij}]$, where $P_{ij} = \Pr[X_2 = j | X_1 = i]$. Throughout the rest of lecture, time invariant Markov chain will be referred to simply as Markov chain (MC).

Example 2: Consider the following three-state MC. In this case, $\boldsymbol{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$.



With $X_0 = A$, the resulting sequence will be "$ABCABCABC\cdots$." Note that this is *not* stationary because $\Pr[X_0 = A, X_1 = B, X_2 = C] = 1$ but $\Pr[X_1 = A, X_2 = B, X_3 = C] = 0$. Instead, $\Pr[X_1 = B, X_2 = C, X_3 = A] = 1$

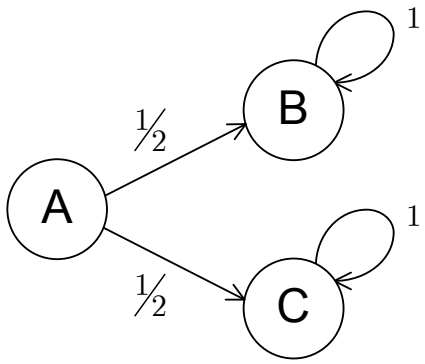
Fact 1 *For every MC, $\exists$stationary distribution $\boldsymbol{\mu}$ on X_0 such that $\boldsymbol{\mu}$ and $\boldsymbol{P}$ define a stationary process. In the example 2, $\boldsymbol{\mu} = \left[\frac{1}{3}\ \ \frac{1}{3}\ \ \frac{1}{3}\right]$.*

Because

$$\begin{aligned} &\Pr[X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n] \\ &\quad = \Pr[X_1 = x_1] \cdot \Pr[X_2 = x_2 | X_1 = x_1] \cdots \Pr[X_n = x_n | X_{n-1} = x_{n-1}] \\ &\quad = \Pr[X_1 = x_1] \cdot P_{x_1 x_2} \cdots P_{x_{n-1} x_n}, \end{aligned}$$

the overall distribution depends only on the distribution on X_1, which implies that the distribution $\boldsymbol{\mu}$ on X_0 is stationary if $\Pr[X_1 = i] = \mu_i (= \Pr[X_0 = i])$.

Example 3: Let us consider the following example:



In this case, $\mu_A = \mu_C = 0, \mu_B = 1$ is stationary, but $\mu_A = \mu_B = 0, \mu_C = 1$ is also stationary. More than one stationary distribution can be problematic, and this situation happens because the MC is reducible.

Definition 4 (Reducibility of Markov Chain) *1. Markov chain given by probability transition matrix $\boldsymbol{P}$ is reducible if $\boldsymbol{P}$ can be written as*

$$\left[\begin{array}{c|c} \boldsymbol{P_0} & \boldsymbol{P_1} \\ \hline \boldsymbol{0} & \boldsymbol{P_2} \end{array}\right],$$

where $\boldsymbol{P_0}, \boldsymbol{P_2}$ are square matrices.

2. MC is irreducible if it is not reducible.

In terms of graph structure, the "irreducible" and "aperiodic" characteristics can be interpreted as

- irreducible - strongly connected, $\exists$path from each state i to state j.
- aperiodic - greatest common divisor of cycle lengths is 1.

Theorem 2 (Perron-Frobenius's Theorem) *Every (aperiodic) irreducible Markov chain has a unique stationary distribution.*

For stationary distribution, the probability distribution on X_1 should be the same as $\boldsymbol{\mu}$, the probability distribution of X_0. $\Rightarrow \Pr[X_1 = j] = \sum_{i=1}^{N} \mu_i P_{ij} = \mu_i$, where $N = |\Omega|$ and $\Omega = \{1, 2, \cdots, N\}$. If we use vector-matrix notation,

$$[\ \boldsymbol{\mu}\ ] \left[\begin{array}{c} \\ \boldsymbol{P} \\ \\ \end{array}\right] = [\ \boldsymbol{\mu}\ ], \tag{1}$$

and $\boldsymbol{\mu}$ corresponds to an eigenvector. For the example 1,

$$\boldsymbol{P} = \left[\begin{array}{ccc} 0.9 & 0.1 & 0 \\ 0 & 2/3 & 1/3 \\ 2/3 & 1/3 & 0 \end{array}\right].$$

Theorem 2 implies that there exists a unique eigenvector with all entries non-negative. We can compute $\boldsymbol{\mu} = [\mu_1\ \mu_2\ \mu_3]$ using (1) and $\mu_1 + \mu_2 + \mu_3 = 1$. $\Rightarrow \boldsymbol{\mu} = [\frac{20}{32}\ \frac{9}{32}\ \frac{3}{32}]$.

3 Entropy Rate of Stochastic Process

There are two reasonable notions for measuring the uncertainty of $\mathscr{X} = \langle X_n \rangle_n$.

- Entropy rate:
$$H(\mathscr{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, \cdots, X_n) \text{ if the limit exists.}$$
- Entropy$'$ rate:
$$H'(\mathscr{X}) = \lim_{n\to\infty} H(X_n | X_1, \cdots, X_{n-1}) \text{ if the limit exists.}$$

Theorem 3 *Entropy rate of a stationary stochastic process exists and equals entropy$'$ rate.*
$$H(\mathscr{X}) = H'(\mathscr{X}).$$

Proof Idea The following inequality can be used for the proof of the existence of $H'(\mathscr{X})$.
$$H(X_n | X_1, \cdots, X_{n-1}) \le H(X_n | X_2, \cdots, X_{n-1}) = H(X_{n-1} | X_1, \cdots, X_{n-1}).$$
For complete proof, refer to pp.64-65 of *Cover*. ■

Theorem 4 *If irreducible MC has probability transition matrix $\boldsymbol{P}$ and stationary distribution $\boldsymbol{\mu}$,*
$$H(\mathscr{X}) = H'(\mathscr{X}) = -\sum_{i,j} \mu_i P_{ij} \log P_{ij}. \tag{2}$$

Proof
$$\begin{aligned} H'(\mathscr{X}) &= \lim_{n\to\infty} H(X_n | X_1, \cdots, X_{n-1}) \\ &= \lim_{n\to\infty} H(X_n | X_{n-1}) \\ &= H(X_2 | X_1) \\ &= \sum_i \Pr[X_1 = i] \cdot H(X_2 | X_1 = i) \\ &= -\sum_i \mu_i \sum_j P_{ij} \log P_{ij}. \end{aligned}$$

■

Using (2), $H(\mathscr{X})$ of the example 1 can be computed:
$$H(\mathscr{X}) = \frac{5}{8} H(0.9) + \frac{3}{8} H\left(\frac{2}{3}\right).$$

AEP for Markov Chain:
$$-\frac{1}{n} \log p(X_1, \cdots, X_n) \longrightarrow H(\mathscr{X}).$$

This doesn't follow from our law of large numbers because random variables may be dependent on each other.

Hidden Markov Model: Now, let us consider the rate of $\langle Y_n \rangle_n$ in the example 1. $H'(\mathscr{Y}) = \lim_{n\to\infty} H(Y_n | Y_1, \cdots, Y_{n-1})$, and is bounded by
$$H(Y_n | Y_1, \cdots, Y_{n-1}, X_1) \le H'(\mathscr{Y}) = \lim_{n\to\infty} H(Y_n | Y_1, \cdots, Y_{n-1}) \le H(Y_n | Y_1, \cdots, Y_{n-1}) \ \forall n.$$
(Try to prove the inequality at the left-hand side!) If we denote the interval between the upper and the lower bounds by ϵ_n,
$$\epsilon_n = H(Y_n | Y_1, \cdots, Y_{n-1}) - H(Y_n | Y_1, \cdots, Y_{n-1}, X_1) = I(X_1; Y_n | Y_1, \cdots, Y_{n-1}),$$
and
$$\sum_{n=1}^{M} \epsilon_n = \sum_{n=1}^{M} I(X_1; Y_n | Y_1, \cdots, Y_{n-1}) \le H(X_1).$$

6.441 Transmission of Information Feb 28, 2006

Lecture 6

Lecturer: Madhu Sudan *Scribe: Chung Chan*

1 Highlight of Previous Lectures

1.1 Lecture 1: Satellite Problem

In the first lecture, we are faced with the problem of transmitting the temperature measurements by a satellite back to the Earth through a noisy binary channel. We realized that encoding the temperature or the temperature difference one-by-one will lead to a binary sequence too long to be transmitted through the channel in real time, let alone the problem of data recovery in the presence of noise.

Intuitively, the problem of symbol-by-symbol encoding in this case is the integer constraint: we can only affort one bit per time unit or not assigning any bit at all. Perhaps we could solve this by encoding a sequence of symbols instead. If we assume expected number of occurrences of each temperature difference is the actual number of occurrences, we found that it is possible to encode 100 temperature changes in 77 bits. The remaining 33 bits is more than enough to correct error due to the noisy channel, given that we have perfect feedback of where the erroneous bit is, and that the actual number of error is its expected value.

Implicitly, we have broken the satellite communication problem into two: first, we try to compress the sequence of temperature changes by removing redundancy due to the integer constraint; second, we try to correct error due to the noise by injecting redundancy back into the sequence by resending the erroneous bits. Would this affect the overall optimality? Shall we compress by simply assuming that expectation is reality? The first question is to be answered by the source channel separation theorem, while the second will be taked in these two lectures.

1.2 Lecture 2: Entropy

In the second lecture, we are faced with the problem of quantifying the uncertainty associated with a random variable. Intuitively, a higher degree of randomness requires a longer description to completely resolve the uncertainty. We therefore constructed a coding scheme for a sequence of n i.i.d. Bernoulli(p) random variables $Z_1, \ldots, Z_n$, where p is unknown to the encoder. Such coding scheme is called universal because the source statistics is not perfectly known. Using the stirling approximation and Chernoff bound, the expected length of the codeword is approximately $-p \log p - (1-p) \log(1-p)$, which we take as the base case to induce the entropy formula by the grouping axiom that $H[p_1, \ldots, p_m] = H[p_1] + (1 - p_1) H[\frac{p_2}{1-p_1}, \ldots, \frac{p_m}{1-p_m}]$. The grouping axiom can be interpreted as the equivalence between asking which elements in $\{1, \ldots, m\}$ Z is and asking whether Z is 1 and if not, which elements in $\{1, \ldots, m\}$ Z is.

Although this derivation points out the close relationship of entropy and expected codeword length, we don't know how general the universal coding scheme it. More precisely, we have not proven whether the scheme is the best we could achieve in minimizing the expected length. Furthermore, the approximates and bound we use is tight only when n goes to infinity. That leaves us wonder what happens for finite n. Can we encode below entropy? The answer, as we will see in these two lectures, is yes, but not very much below entropy.

Although question 4 on p.42 of Cover indicates that we can replace this base case by the normalization condition $H_2[\frac{1}{2}, \frac{1}{2}] = 1$ and the continuity condition of $H[p, 1-p]$, this axiomatic formulation gives us less practical meaning of the entropy. In other words, the axiomatic formulation does not relate entropy to the expected codeword length at all.

1.3 Lecture 3: Properties of Entropy and Mutual Information

In the third lecture, we explored the mathematical properties of entropy and mutual information, in an attempt to justify our intuition of what information is. The conditioning does not increase entropy intuitively because additional knowledge can only help resolve ambiguity on average. Fano's inequality states that the error probability of estimating X from Y is bound to be large is the equivocation $H(X|Y)$, or the uncertainty in X after knowing Y remains large. Data processing theorem corresponds to the fact that further processing on an observation can only produce a second-hand information that is no better than the original in an estimation/detection problem.

Although these properties does not concretely define the practical meaning of entropy and mutual information, they agree with how we think randomness and information ought to behave.

1.4 Lecture 4: Asymptotic Equipartition Property (AEP)

The solution to Satellite problem assumed expectation is reality. Under what scenario is this assumption valid? By the law of large number for i.i.d. random variables (or the ergodic theory for the more general stationary ergodic process), reality indeed converges to expectation with probability one. More precisely, the n-sample probability converges to 2^{-nH} almost surely in the first order (in n) of the exponent. As a result, we can define a small ($\doteq 2^{nH}$) highly probable ($\Pr > 1 - e^{f(\epsilon)n}$ where $f(\epsilon) > 0$) set with an almost uniform probability distribution ($\doteq 2^{-nH}$).

As it will become clear later, typicality is an important concept in channel coding.

Typicality suggests a natural source coding scheme. If we encode just the typical set with $\lceil nH + \epsilon \rceil$ bits, the error probability will be less than $e^{f(\epsilon)n}$, meaning that the code will be asymptotically lossless as n increases. If we encode also the atypical set with $\lceil \log |\mathscr{X}| \rceil$ bits, we obtain a lossless code (not necessarily uniquely decodable[1]) with an expected length arbitrarily close to $\lceil nH + \epsilon \rceil$. The cost of going from lossy to lossless is therefore the weakening of the statement on the lengths of every codeword to the statement on the expected codeword length. But in both cases, we can draw a relationship between entropy and codeword length.

To make this relationship vigorous, we still have to prove whether typical set encoding is optimal. Intuitively, symmetry seems to support the use of fixed-length code for the typical sequences. But how could we argue that variable length could not do better? Furthermore, typical set encoding is just an asymptotic result for the the stochastic processes with AEP. What can we say about the case when n is finite, or when AEP does not hold? How general is AEP?

1.5 Lecture 5: Entropy Rate

To answer how general AEP is, we need to find the sufficient conditions for a stochastic process so that AEP holds. And to show that AEP holds for a stochastic process $\{X_i\}$, it is sufficient and necessary to prove that $-\frac{1}{n}p(X_1, \ldots, X_n)$ converges to its expectation $H(\mathscr{X}) := \lim_{n\to\infty} \frac{1}{n} H(X_1, \ldots, X_n)$, called entropy rate.

Stationarity of a process guarantees not only the existence of the entropy rate, but also that the rate converges to the limit $H'(\mathscr{X}) := \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_0)$. In the special case of a stationary Markov process generated by running a time-invariant, aperiodic and irreducible Markov chain with its unique stationary distribution μ_i, we showed that the entropy rate simplifies to $\sum_{i,j} \mu_i P_{ij} \log \frac{1}{P_{ij}}$ where $\mathbf{P}$ is the transition probability matrix.

Existence of entropy rate, however, does not imply that AEP holds. Consider the stationary process $\{X_i\}$ that is i.i.d. Bernoulli(1/2) with probability 1/2 and is deterministically 0.5 with probability 1/2. stationarity guarantees existence of the entropy rate. However, the sequence of 0.5 has probability 1/2 regardless of n while other sequences of 0's and 1's are equiprobable with a total probability accounting for the remaining 1/2. AEP does not hold for this process because 0.5 cannot be typical as it is the only sequence with probability 1/2, and the other binary sequences cannot be typical as their probability is only 1/2. It turns out that an additional constraint the the process being ergodic guarantees AEP.

[1] To make it uniquely decodable, we can add one additional bit to the front of every codeword to distinguish between the typical sequences and the atypical ones. Effectively, the bit makes the code prefix-free.

On the contrary, a Markov chain being periodic or reducible does not exclude the existence of entropy rate. For examples, the periodic Markov chain with $P_{01} = P_{10} = 1$ and the reducible Markov chain with $P_{01} = P_{00} = 1/2$ and $P_{11} = 1$ have a unique stationary distribution, from which the entropy rate of the corresponding stationary Markov processes can be calculated. AEP also holds for both cases because the processes are ergodic.

2 Data Compression/Source Coding

What is data compression/source coding? In the satellite problem, we compressed the sequence of temperature differences into a binary sequence short enough to be transmitted through a channel. More generally, data compression refers to the process of generating a compact representation of the input data to match some interface.

Why is it possible to compress data? As we concluded from the satellite problem, batching the input symbols allow us to remove redundancy due to the integer constraint, even though the input symbols are independent. Redundancy can also be due to patterns in the data. These patterns can be ascribed to the memory of the process. For example, the 2-state stationary Markov process with $P_{01} = P_{10} = 1$ has an alternating pattern. If we have to encode the process into a binary sequence symbol-by-symbol, one optimal way would be to encode 0 to 0 and 1 to 1 since the input symbols are equiprobable. However, if we batch the input symbols into sequences, we need only one bit to encode the two possible alternating pattern. From this example, we realize that redundancy can be due to the memory of the process rather than the integer constraint.

2.1 Introduction

A common (though not the most general) notion of code is a mapping from a countable support set of a discrete random variable to a D-ary sequence. More precisely,

Definition 1 (Code) *A code C of a random variable X taking values from $\mathscr{X}$ is the mapping $\mathscr{X} \mapsto \mathscr{D}^*$, where $\mathscr{D} := \{0, \dots, D-1\}$ is the D-ary alphabet set, and $\mathscr{D}^* := \bigcup_{n \in \mathbb{Z}^+} D^n$ is the set of all possible codeword lengths.*

2.2 Non-singular Code

The objective of data compression is usually to minimize the expected codeword length subjected to certain constraints on decodability. If $C(x)$ is a codeword with length $l(x)$ for an input symbol $x \in \mathscr{X}$, the expected length is $\sum_{x \in \mathscr{X}} p(x) l(x)$. A non-singular/lossless/decodable code C is defined as follows

Definition 2 (Non-singular code) *A code C is nonsingular if*

$$\exists Dec \, \forall x \in \mathscr{X} \; Dec(C(x)) = x$$

Dec is called the decompressor/decoder.

In many cases, we allow the code to be singular because the probability of error is small (e.g. the lossy typical set encoding), the error is neglegible (e.g. slightly distorted pixels sparsely distributed in an image), or it is simply infeasible to recover the input symbol error-free. The probability of error can be thought of as the expectation $E[1_{\{x \in \mathscr{X} : Dec(C(x)) = x\}}(X)]$, where 1_A is the indicator function $\mathscr{X} \mapsto \{0, 1\}$ that returns one when its argument is in A and zero otherwise. Another common choice is the mean squared error. These types of measurement is called distortion/fidelity. The studies of the relationship between distortion and rate is the rate distortion theory.

To construct an example of non-singular code, consider the random variable X which takes the value $1, 2, 3, 4$ with probability $p_1 = 1/2, p_2 = 1/4, p_3 = 1/8, p_4 = 1/8$ respectively. An optimal fixed-length

non-singular code would be,

$$C_1(1) = 00$$
$$C_1(2) = 01$$
$$C_1(3) = 10$$
$$C_1(4) = 11$$

with expected length $L_{FL} = 2$ bits.

The fixed-length requirement is so stringent that the optimal codeword lengths are determined by matching the cardinalities rather the probabilities of each symbol. If we allow variable length code, an optimal non-singular code would be,

$$C_2(1) = 0$$
$$C_2(2) = 1$$
$$C_2(3) = 00$$
$$C_2(4) = 11$$

with expected length $L_{NS} = 1.25$ bits. Optimality can be argued from the greedy algorithm of assigning the more probable symbols first to the shortest codeword not yet allocated.

2.3 Uniquely Decodable Code

Is there any weakness associated with the optimal non-singular code? To see this, let us define notion of an extended code,

Definition 3 (Extended code) *An extended code $C^{(k)}$ of a code C is a mapping $\mathscr{X}^k \mapsto \mathscr{D}^*$ such that*

$$C^{(k)}(x_1, \ldots, x_k) = \underbrace{C(1) \cdots C(k)}_{concatenated}$$

If $C^(k)$ is non-singular, it can be used to encode k symbols losslessly. This is more attractive then redesigning a non-singular code for the concatenated input symbols because doing so would generate a codebook with size exponential in k rather than constant with respect to k. For the optimal non-singular code example above, we have $C_2^{(2)}(13) = C_2^{(2)}(31) = 00$ and $C_2^{(3)}(113) = C_2^{(3)}(311) = 000$. Since the $\gcd(2, 3) = 1$, $C_2^{(k)}$ is non-singular for any $k > 1$ by induction. In otherwords, we cannot extend the optimal non-singular code to encode input sequences losslessly. How can we fix this?

Let us first define our desired code formally as follows,

Definition 4 (Uniquely decodable code) *A code C is uniquely decodable if its extended code $C^{(k)}$ is non-singular for all $k \geq 1$.*

It can be verified that this is equivalent to the condition that any extended codeword can be decoded even if the number of concatenations k is unknown. In other words, the union of all extended codes, called the extension $C^* : \mathscr{X}^* \mapsto D^*$, where $\mathscr{X}^* := \bigcup_{n \in \mathbb{Z}^+} \mathscr{X}^n$, is a non-singular code. Hence, it is unnecessary for the decoder to know the number encoded symbols apriori. It can figure that out directly by decoding the received codeword. For notational simplicity, we will often use $C(x_1, \ldots, x_k)$ instead of C^* or $C^{(k)}$ because the number of arguments tell us the order of the extension.

The reason why the optimal non-singular code C_2 is not uniquely decodable stems from the problem that $C_2(1)C_2(1) = C_2(3)$, which requires that $C_2(1)$ be a prefix of $C_2(3)$. To see it more clearly, consider the following code,

$$C_3(1) = 0$$
$$C_3(2) = 10$$
$$C_3(3) = 110$$
$$C_3(4) = 111$$

with the expected length $L_{PF} = 1.75 = H(X)$. This is called the prefix-free/instantaneous code because no codeword is a prefix of another codeword. This leads to the following definition,

Definition 5 (Prefix-free code) *A code C is prefix-free if no codeword is a prefix of another codeword.*

Intuitively, extended prefix-free code should be prefix-free, which implies non-singularity. If the codeword of the extended code has a prefix that is a codeword of the unextended code, the prefix cannot be a concatenation of more than one codeword because of the prefix-free condition. Thus, we can decode it immediately, remove it from the codeword, and repeat the process until the entire codeword is decoded. This explains not only why the code is uniquely decodable, but also why it is called instantaneous. Let's state this more precisely as a theorem,

Theorem 6 *Any prefix-free code is uniquely decodable.*

Proof Suppose C is prefix-free. If two distinct input sequences $a_1 \cdots a_m$ and $b_1 \cdots b_n$ have the same codeword $c_1 \cdots c_l$. i.e.

$$\begin{aligned} C^*(a_1, \ldots, a_m) &= c_1 \cdots c_l \\ &= C^*(b_1, \ldots, b_n) \end{aligned}$$

Then, we have $a_1 = b_1$ by the following proof of contradiction. If $a_1 \neq b_1$, $C(a_1) \neq C(b_1)$ by non-singularity, implying that $C(a_1)$ must be a prefix of $C(b_1)$ or vice versa in order to have the same codewords for the two input sequences. This immediately lead to a contradiction because C is prefix-free.

Suppose, without loss of generality, that $C(a_1) = C(b_1) = c_1 \cdots c_k$. Then,

$$\begin{aligned} C^*(a_2, \ldots, a_m) &= c_{k+1} \cdots c_l \\ &= C^*(b_2, \ldots, b_n) \end{aligned}$$

By induction, we have $m = n$ and $a_i = b_i$ for all $i \leq n$. Thus, any two distinct sequences must map to two distinct codewords, implying that extension of C is non-singular. ■

Is the prefix-free requirement too strong for unique decodability? There are certainly uniquely decodable codes that are not prefix-free. For instance, reversing the codewords in C_3 lead to the suffix-free code that is still uniquely decodable because its extended code consists of the reversed codewords of the extended prefix-free code that is non-singular. For example,

$$\begin{aligned} C_4(1) &= 0 \\ C_4(2) &= 10 \\ C_4(3) &= 110 \\ C_4(4) &= 111 \end{aligned}$$

is a uniquely decodable code that is suffix-free but not prefix-free. Can a suffix-free code be prefix-free? Indeed, it can be easily verified that all fixed length non-singular codes are both suffix-free and prefix-free. Can uniquely decodable code be neither suffix-free nor prefix-free? It's possible. For example,

$$\begin{aligned} C_4(1) &= 01 \\ C_4(2) &= 10 \\ C_4(3) &= 011 \\ C_4(4) &= 110 \end{aligned}$$

is uniquely decodable but $C_4(1)$ is a prefix of $C_4(3)$ and $C_4(2)$ is a suffix of $C_4(4)$. The proof of unique decodability for a code that is neither prefix-free nor suffix-free can be quite tricky. Should we concern ourselves with this type of codes? Does it gain us anything compared to the prefix-free code? We will see in the next lecture that prefix-free condition does not further increase the expected length compared to the unique decodability condition, and unique decodability does not increase the expected length by a significant amount compared to the non-singlarity condition.

6.441 Transmission of Information March 2, 2006

Lecture 7

Lecturer: Madhu Sudan *Scribe: Xiaomeng Shi*

1 Administrative Issues

- Pset 2 due next Wednesday, March 8, 2006.
- Midterm in three weeks (March 23). Project selection due on same day.

2 Today: Data Compression Continued

- Review L06
- Non-singular code
- Uniquely decodable code
- Entropy lower bound

3 Review of Lecture 6

AEP enables us to do fixed length typical sequence encoding. As the length of the sequence approaches infinity, the expected number of bits required per symbol approaches the entropy of the source. The next question we would like to ask is, would variable length encoding help us encode below entropy?

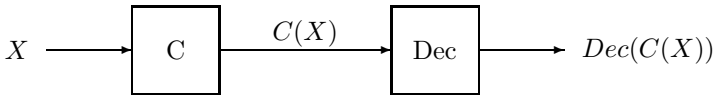


Figure 1: $(\forall X)\ Dec(C(X)) = X \implies C(X)$ is non-singular.

Non-Singular Codes Every elements of the range of X maps into a different string, thus X can be completely recovered from the code. In many situations, however, we don't need such stringent constraint on the code. For example, when compressing videos or pictures, as long as the errors are far apart and not clustered at one point, the effect is often negligible. Distortion theory gives theoretical bounds on the amount of compression that can be achieved under a given amount of distortion.

Non-singular extended codebook A more stringent constraint is to have the extended codebook also non-singular.

Definition 1 *An extended codebook from code C is C^* defined by $C^{(k)} : \mathscr{X}^{(k)} \to \mathscr{D}^{(k)}$, where $\mathscr{X} = \{1, \ldots, m\}$ is the set of source symbols, $\mathscr{D} = \{0, 1, \ldots, D-1\}$ is a D-ary alphabet of codeword symbols, $\mathscr{D}^* = \bigcup_{n \in \mathbb{Z}^+} \mathscr{D}^n$, and $C^{(k)}(x_1, \ldots, x_k) = C(x_1)C(x_2)\ldots C(x_k)$ is a concatenation of the individual codewords.*

Why extended codebooks? We could have simply concatenated sequences and assigned each a probability measure, but the resulting codebook size would be exponentially in sequence length. Through extension, the codebook is smaller in size. An equally important reason is that $C^{(k)}$ needs to be non-singular $(\forall k)$ for the concatenated sequence to be recoverable (uniquely decodable) at the receiver.

4 Best Non-Singular Code

What's the best non-singular code we can achieve, measured in terms of expected code length? Intuitively we would map the shortest codeword to symbols with the highest probability.

For the code to be non-singular, each possible code word must correspond to one unique node on the code tree, extended breath-first. Excluding the root node:

$$\underbrace{D + D^2 + \ldots + D^{l_i - 1}}_{\frac{D^{l_i-1}-1}{D-1}D} < i \leq \underbrace{D + D^2 + \ldots + D^{l_i}}_{\frac{D^{l_i}-1}{D-1}D} \implies l_i = \left\lceil \log_D \left(\frac{i(D-1)}{D} + 1 \right) \right\rceil$$

The expected length of this non-singular code is therefore:

$$L^*_{NS} = \sum_i p_i l_i \neq H(X)$$

This direct comparison with the source entropy is very complex. We will try solving the expected length inexactly to find a bound.

4.1 Bounds on the optimal codelength

How do we impose the condition of non-singularity mathematically?

Define $a(l)$ = number of distinct code words with length l, then

$$a(l) \leq D^l \,, \qquad l_1 \leq l_2 \leq \ldots \leq l_m \,.$$

The sum of probability of all possible code words is therefore

$$\sum_{i=1}^{m} D^{-l_i} = \sum_{l=1}^{l_m} D^{-l} a(l) \;\leq l_m \,.$$

The expected length of non-singular codes is ($*$ means optimal)

$$\begin{aligned} L^*_{NS} &= \sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m} -p_i \log_D D^{-l_i} = \sum_{i=1}^{m} -p_i \log_D \frac{D^{-l_i}}{p_i} + H(X) \\ (Jensen's\ \ ineq) \quad &\geq -\log_D \sum_{i=1}^{m} D^{-l_i} + H(X) \\ &\geq -\log_D l_m + H(X) \end{aligned}$$

Rather than using l_m, is there a bound that is independent of the codebook?

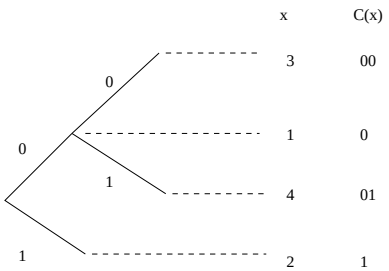
$$\begin{aligned} l_i = \left\lceil \log_D \left(\frac{i(D-1)}{D} + 1 \right) \right\rceil \implies l_m &\leq \lceil \log_D m \rceil \\ &\leq \log_D m + 1 \\ &\leq D \log_D m - \underbrace{(D-1)\log_D m}_{= \frac{D-1}{\ln D} \ln m \geq \frac{2-1}{\ln 2} \ln 2 = 1} + 1 \\ l_m &\leq D \log_D m \end{aligned}$$

Therefore,

$$\begin{aligned} L_{NS}^* &\geq -\log_D l_m + H(X) \\ &\geq -\log_D (D \log_D m) + H(X) \\ &\geq -\log_D D - \log_D \log_D m + H(X) \\ &\geq -\underbrace{\log_D \log_D m}_{\text{small}} - 1 + \underbrace{H(X)}_{\sim O(\log m)} \end{aligned}$$

Asymptotically, lengths of non-singular codes are constrained by entropy. Nonetheless, this is a bound that may be loose. Is it actually achievable? Can we really code below entropy? We show the answer is yes by an example.

Example: $p(1) = \frac{1}{2}, p(2) = \frac{1}{4}, p(3) = p(4) = \frac{1}{8}$



$E[|C(x)|] = 1.25 \ < \ H(X) = 1.75$

5 Kraft Inequality for Uniquely Decodable Codes

Claim 2 *Given a non-singular code C, there exist prefix-free codes with expected length $L_{NS} + O(\sqrt{L_{NS}})$*

What we are trying to show here is that the additional constraint of unique decodability does not worsen the expected code length much.

Proof First generate $C \to C_1$ by zero padding. Next divide C_1 into consecutive blocks of length $\lceil \sqrt{L_{NS}} \rceil$, and write this as $\lceil \sqrt{L_{NS}} \rceil \mid |C_1(x)|$. Insert 0 between each pair of blocks and append 1 at the end of the code word.

$$C_1(x) \quad \underbrace{\boxed{w_1}}_{\lceil \sqrt{L_{NS}} \rceil} \underbrace{\boxed{w_2}}_{\lceil \sqrt{L_{NS}} \rceil} \boxed{w_3} \quad \ldots \quad \boxed{w_k}$$

$$C_2(x) \quad \boxed{w_1} \boxed{0} \boxed{w_2} \boxed{0} \boxed{w_3} \ \ldots \ \boxed{w_k} \boxed{1}$$

$C_2(x)$ is prefix-free thus uniquely decodable.

$$\begin{aligned} E\left[|C_2(x)|\right] &= E\left[|C_1(x)| \frac{\lceil \sqrt{L_{NS}} \rceil + 1}{\lceil \sqrt{L_{NS}} \rceil}\right] \\ &= \frac{\lceil \sqrt{L_{NS}} \rceil + 1}{\lceil \sqrt{L_{NS}} \rceil} \underbrace{E\left[|C_1(x)|\right]}_{\leq E[|C(x)|] + \lceil \sqrt{L_{NS}} \rceil = L_{NS} + \lceil \sqrt{L_{NS}} \rceil} \\ &= L_{NS} + O\left(\left\lceil \sqrt{L_{NS}} \right\rceil\right) \end{aligned}$$

■

For this uniquely decodable code $C^{(k)}: \mathscr{X}^{(k)} \to \mathscr{D}^*$ constructed with k concatenations,

$$\begin{aligned} l(x_1,\ldots,x_k) =& \sum_{i=1}^{k} l(x_i) \\ \sum_{(x_1,\ldots,x_k)\in\mathscr{X}^k} D^{-l(x_1,\ldots,x_k)} =& \left(\sum_{i=1}^{m} D^{-l_i}\right)^k \leq k l_m \\ \sum_{i=1}^{m} D^{-l_i} \leq& (kl_m)^{1/k} = \exp\frac{\log k + \log l_m}{k} \to \qquad \exp(0) = 1 \text{ as } k\to\infty \\ \sum_{i=1}^{m} D^{-l_i} \leq& 1 \end{aligned}$$

This is the *Kraft Inequality for Uniquely Decodable Code.*

Next consider the expected length L^*_{UD}:

$$\begin{aligned} L^*_{UD} =& \sum_i p_i l_i \\ =& \sum_i -p_i \log_D \frac{D^{-l_i}}{p_i} + H(X) \\ (Jensen's\ ineq) \quad \geq& -\underbrace{\log_D \sum_i D^{-l_i}}_{\leq 1} + H(X) \\ \geq& H(x) \end{aligned}$$

Here the code is uniquely decodable, and its expected length is larger than the entropy. Since the AEP code achieves the entropy, this is a tight lower bound as $n \to \infty$. What happens when n is finite?

Recall that the Kraft Inequality is a necessary condition for unique decodability. It is indeed also a sufficient condition for the existence of a prefix-free code satisfying the length assignments. Sufficiency can be proved by examining the tree structure for prefix free codes. Assume $l_1 \leq l_2 \ldots \leq l_m$:

1. assign the first free node at depth l_i to i.
2. prune the subtree of the assigned node so that the descendents are not free to be assigned to any symbol.
3. repeat until l_m

With these assignments, it is feasible to get a prefix free code. We can prove this by contradiction as follows.

Proof If the assignment fails, there exists $k < m$ such than the tree become full[1] without any free nodes after assigning l_k. The fact that the tree is full implies that $\sum_{i=1}^{k} D^{-l_i} = 1 < \sum_{i=1}^{m} D^{-l_i}$, which is a contradiction to the assumption that the set of lengths satisfies the Kraft Inequality. ■

Consider the following length assignment,

[1] A D-ary tree is full iff all nodes have either 0 or D children.

Definition 3 (Shannon Code) $l_i = \lceil -\log_D p_i \rceil$.

which satifies the Kraft Inequality

$$\sum_i D^{-l_i} \leq \sum_i D^{\log_D p_i} = \sum_i p_i = 1$$

and therefore a prefix-free code exists with this length assignment. Its expected length L_{SH} is bounded as follows,

$$\log_D \frac{1}{p_i} \leq l_i \leq \log_D \frac{1}{p_i} \Longrightarrow H(X) \leq L^*_{UD} \leq L_{SH} < H(X) + 1$$

which therefore gives an upperbound on the expected length of the optimal uniquely decodable code.

What's the best optimal uniquely decodable code?

Huffman Code optimal prefix free code
To generate the Huffman Code,

1. add $r \in \{0, \ldots, D-1\}$ dummy nodes such that the total number of nodes is equal to $1 + k(D-1)$ for some k. $\Pr(dummy) = 0$.
2. group the D least probable symbols into one symbol; use one D-ary symbol to distinguish these D symbols.
3. repeat step 1 until only one node remains.

6 Summary

$$H(X) - \log_D \log_D m - 1 < L^*_{NS} \leq \underbrace{L^*_{UD}}_{\geq H(X)} < H(X) + 1$$

6.441 Transmission of Information March 7, 2005

Lecture 8

Lecturer: Madhu Sudan *Scribe: Anna Lee*

Today

- Quality of Huffman Codes
- Universal Coding
- Lempel Ziv Algorithm

Admin

- PS2 due tomorrow
- PS1 will be handed back Thursday

Review

$C : \Omega_{\{1,\ldots,n\}} \to D^*_{\text{often} D=\{0,1\}}$

- Kraft's Inequality: $l_i = |(C_i)|$, then $\sum_{i=1}^n D^{-l_i} \leq 1$ if code is uniquely decodable.
- if p_i is prob. of element i, we would like to minimize $E[l] = \sum_{i=1}^n p_i l_i$.
- Entropy inequality: $\frac{H(p_1,\ldots,p_n)}{\log D} \leq E[L]$

1. Kraft's inequality is tight
   if $l_i, \ldots, l_n$ satisfy $\sum_{i=1}^n D^{-l_i} \leq 1$ then $\exists C : \{i, \ldots, n\} \to D^*$ s.t. $|C(i)| = l_i$.
2. Shannon Coding Method
   - $l_i = \lceil \log_D \frac{1}{p_i} \rceil \leq \log_D \frac{1}{p_i} + 1 \Rightarrow E[L] \leq \frac{H(X)}{\log D} + 1$
   - should use to compress $\bar{X} = (X_1, \ldots, X_k)$ where $k \to \infty$, $X_1, \ldots, X_k$ i.i.d. $\sim X$
   - (from here, $D = 2$).
   - $kH(X) = H(\bar{X}) \leq E[\text{length compressing} \bar{X}] \leq H(\bar{X}) + 1 = kH(X) + 1$.
     $\to$ loss becomes $\frac{1}{k}$ per element.

Huffman Coding

- "optimal" prefix code for variable X
- $C_{\text{Huffman}} : \{i, \ldots, n\} \to \{0,1\}^*$
- Huffman code $(p_1, \ldots, p_n)$
  - if $n \leq 2$, ...
  - sort so that $p_1 \geq p_2 \geq \cdots \geq p_n$

– $C' \leftarrow$ Huffman Code$(p_1, p_2, \ldots, p_{n-2}, p_{n-1} + p_n)$

–

$$C[i] = \begin{cases} C'[i] & \text{, if } i \leq n-2, \\ C'[n-1]0 & \text{, if } i = n-1, \\ C'[n-1]1 & \text{, if } i = n. \end{cases}$$

Today

Claim: For any prefix-free code $C : \{1, \ldots, n\} \to \{0,1\}^*$ it is the case that $\sum_{i=1}^n p_i |C(i)| \geq \sum_{i=1}^n p_i |C_{\text{Huff}}(i)|$.

Prefix free:

- All codewords are leaves.
- in optimal tree, can always assume $p_i < p_j \Rightarrow l_i \leq l_j$
- in optimal tree, no nodes have only one child
- $\exists 2$ leaves at lowest level with the same parent and with the two lowest probabilities.
- $E[\text{length}(p_1, \ldots, p_n)] \geq E[\text{length}(p_1, \ldots, p_{n-2}, p_{n-1} + p_n)] + (p_{n-1} + p_n)1$

$X_1, X_2, \ldots, X_t$, X_i i.i.d. $\sim X$ then compressing with Huffman/Shannan is more realistic.

Markovian Source (Hidden Markov Chain or Ergodic Source)

- Finite State Space $\{1, \ldots, n\}$
- Transition prob. matrix $\{p_{ij}\}_{i,j=1,\ldots,n}$
- $(i,j) \to b_{ij} \in \{0,1\}$
- Build for English, but what happens if source switches to French?

Universal Coding

Goal: compress information produced by a Markovian Source

- must be efficient
- has no prior knowledge of source

Consider $X \in \{1, \ldots, n\}$, $p(X = i) = p_i$, $X_1, \ldots, X_t$ i.i.d. $\sim X$.
Compress $(\bar{X} = (X_1, \ldots, X_t))$

- let t_i be the number of occurences of i in $\bar{X}$
- send $(t_1, \ldots, t_n)$
- which of $\binom{t}{t_1 \ldots t_n}$ possible sequences was seen
- amount of communication $\rightsquigarrow$ negligible $+tH(X)$

AEP for Ergodic Markovian Source

if $(X_1, \ldots, X_L)$ elements drawn from finite Markovian (ergodic) source then

$$\frac{-\log \Pr(p(X_1, \ldots, X_L))}{L} \to H(X) \qquad \text{entropy rate of process.}$$

With probability $1 - \delta$, $2^{-H(X)L(1+\epsilon)} \leq p(X_1, \ldots, X_L) \leq 2^{-H(X)L(1-\epsilon)}$

Divide t-length sequences into blocks of length L.

Compression idea (L, k)

- $X_1, \ldots, X_t \rightsquigarrow Y_1, \ldots, Y_{\frac{t}{L}}$, $Y_i \in \{0,1\}^L$, $t' = \frac{t}{L}$
- (1) typical set: $w \in \{0,1\}^L$ s.t. w appears at least k times, send $w \leq 2^L$ bits
- (2) for each block:
  - "0" (typical) and index into set of elements sent in step 1 $\approx H(X)(L+1)t/L$ bits
  - "1" (nontypical) and $w \in \{0,1\}^L \approx \delta(L+1)t/L$ bits
- as $t \to \infty$, $2^L + H(X)(L+1)\frac{t}{L} + \delta(L+1)\frac{t}{L} \approx H(X)t$.

6.441 Transmission of Information Mar 9, 2005

Lecture 9

Lecturer: Madhu Sudan *Scribe: Jin Woo Shin*

Today we are going to continue talking about data compression; You can get more detail information of Lempel-Ziv algorithm at the lecture note of Gallager 2/7/1994 dated.

1 Today's topics

- Markov source
- Universal coding algorithm
- Lempel-Ziv algorithm

2 Markov source

Let's assume that there is a Markov process which has a finite state S and whose state transition matrix P is fixed. Also there is a function of output sequence X that only depends on current and one step before states. The sequence of S goes to produce the random source X and we can only observe the output sequence $X's$. This process is called as Markov source or Markovian process. The detailed description is as following:

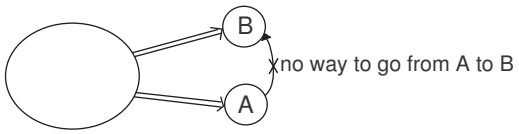
$$\begin{cases} S & = \{1, 2, \ldots, n\} \\ P & = \{P_{ij}\}_{i,j \in S} = \text{Prob. chain goes to state } j \text{ / state } i(\text{given}) \text{ in one step} \\ f & = S \times S \to \{0, 1\} \quad (\text{output function}) \end{cases}$$

- $(y_0, \ldots, y_t) \in S^t$ s.t. $y_0 \in S$: initial state
- $Pr[y_t = j | y_{t-1} = i \quad y_{t-1} \cdots y_0] = P_{ij}$
- $(x_0, \ldots, x_t) \in \mathscr{X}$ s.t. $x_t = f(y_{t-1}, y_t)$

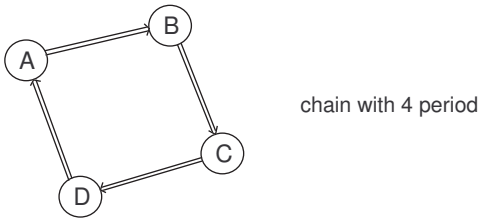
2.1 Notations for Markov source

There are several prime notations for Markov source.

- Source is **reducible** if $\exists i, j \in S$ with no path of prob from i to j, e.g.



- Source is **L periodic** if $\forall C$ paths from i back to i has length divisible by **L**



  We know where the state is after 4 steps.

- Source is **irreducible** if it is not reducible and aperiodic(:= if it is not L periodic for any $L \geq 2$

$$\boxed{\text{"}\textbf{irreducible + aperiodic}\text{"} \Leftrightarrow \text{ergodic}}$$

We don't get into the general concept of ergodic process. However, above relationship will help us set up what we want to do.

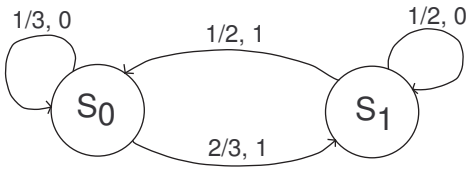
2.2 Entropy rate

We have already learned the definition of Entropy rate $H(\mathscr{X})$ as following:

$$H(\mathscr{X}) := \lim_{t\to\infty} H(X_t|X_{t-1}, \ldots, X_1)$$

This limit does exist in regular markov chain. However, in the case of markov source,it may not. $\{Y_i\}$ definitely follow markov chain and $Y_t|Y_{t-1}$ is fixed for $\forall Y_i$, but X_i does not construct markov chain because X_{t-1} does not contain all of past information of X_i.

Example



In the above Markov source, the observation 0 or 1 does not give us perfect information of states. We can also notice that more past sequence $(0, 1, \cdots, 1)$ gives us more information of states. Generally, it is very hard to calculate entropy rate in this type of problems because whenever we partially observe markov chain, we cannot find correct start state.

[Typical Set Theorem for ergodic Markov source $\mathscr{X}$]

$\forall t$, $\exists T_t \in \{0,1\}^t$

- $\lim_{t\to\infty} \frac{log|T_t|}{t} \to H(\mathscr{X})$
- For $\forall (x_1, \ldots, x_t) \in T_t$,
  $Pr_{(X_1,\ldots,X_t)}[(X_1, \ldots, X_t) = (x_1, \ldots, x_t)] \approx 2^{-H(\mathscr{X})(1\pm\epsilon)t}$
- $\lim_{t\to\infty} Pr_{(X_1,\ldots,X_t)}[(X_1, \ldots, X_t) \in T_t] \to 1$

With this theorem, we can achieve the fact that 1) in typical set, probability distribution of $\{X_i\}$ is almost uniform distribution and each prob. is $\approx 2^{-H(\mathscr{X})(1\pm\epsilon)t}$, 2) we don't need to think out of T_t.

Example Let's think of $\{X_i\}$ i.i.d and satisfies following property.

$$X_i = \begin{cases} 0 & \text{, with probability 0.9,} \\ 1 & \text{, with probability 0.1.} \end{cases}$$

Then, 1) the most probable sequence $(X_1, \ldots, X_t)$ is all 0 sequence, but this sequence is not contained in the typical set. 2) The typical set is roughly uniformly distributed large domain. 3) The size of typical set T_t is virtually lower bound of compression.

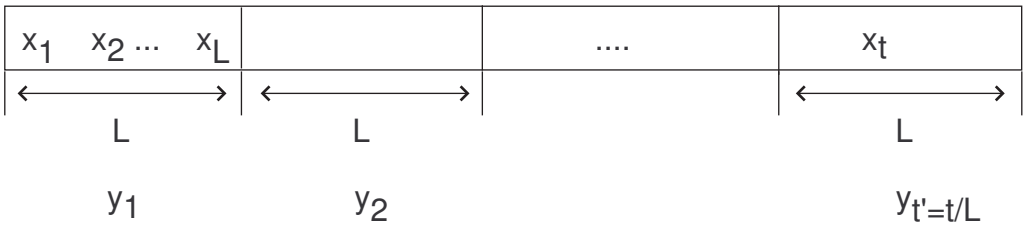
3 Universal Encoding

As we studied in the last lecture, preview the universal encoding. Huffman coding compresses and i.i.d source with a known distribution to its entropy entropy limit. But, what compression can be achieved if it is an unknown distribution? Universal Encoding starts from this idea, and our goal is finding such an uniquely decodable coding algorithm C that satisfies the following property.

$$\forall \mathscr{X},\ \lim_{t\to\infty} \frac{E_{X_i\sim\mathscr{X}}[|C(X_1, X_2, \ldots, X_t)|]}{t} = H(\mathscr{X})$$

I will introduce the 'shabby' encoding as such an example.

3.1 Shabby Algorithm



- At first, divide the data $(x_1, x_2, \ldots, x_t)$ into L sections, and each section is denoted by $y_i \in \{0,1\}^L$, $1 \le i \le t' = t/L$.
- As a first step of encoding, build a dictionary of frequent strings in $\{0,1\}^L$. The frequency constant k is what we will find later, and $I(w)$ indicates the index of w in the dictionary. This step can be formulated as follows,

  count $\leftarrow 0$
  For $w \in \{0,1\}^L$ do
  if $|j|y_j = w| \ge k$, then $Z_w \leftarrow 1$, count $\leftarrow$ count $+ 1, I(w) \leftarrow$ count
  else $Z_w \leftarrow 0$

- The second step of encoding is the real encoding step using the dictionary we built in the previous step. If the section data y_j is in the dictionary, encode it as the index of the dictionary. Otherwise, the encoded data is just the plain data. Also, we add one bit to the encoded data

to indicate its type. This step can be formulated as follows,

$$\begin{aligned}&\text{For } j = 1 \text{ to } t' \text{ do}\\ &\quad w \leftarrow y_j\\ &\quad \text{if } Z_w = 1, \text{ then } u_j \leftarrow (1, I(w))\\ &\quad \text{else } u_j \leftarrow (0, w)\end{aligned}$$

- Therefore, the encoded data is consisted of the dictionary$((Z_w)_{w\in\{0,1\}^L})$ and $u_1, u_2, \ldots, u_{t'}$.

Now, the remaining problem is to determine L, k to minimize the encoded data's length. The following theorem tells about that.

Theorem 1 *If* $k = \frac{t}{L}2^{-H(X)(1+\epsilon)L}$,

$$\lim_{\epsilon\to 0}\lim_{L\to\infty}\lim_{t\to\infty}\frac{E[|Shabby_{L,k}(X_1, X_2, \ldots, X_t|]}{t} \to H(\mathscr{X})$$

Proof Idea The size of dictionary$(\{Z_w\})$ is at most 2^L, because there are 2^L w's. And, the AEP says the size of the dictionary is bounded by $2^{H(\mathscr{X})(1+\epsilon)L}$. Therefore, if $Z_{u_j} = 1$, the length of u_j is $H(\mathscr{X})(1+\epsilon)L + 1$, and the total length of u_j, where $Z_{u_j} = 1$ is $\frac{t}{L}(H(\mathscr{X})(1+\epsilon)L + 1)$. Also, from AEP, we can know the total length of u_j, where $Z_{u_j} = 0$ is $\delta\frac{t}{L}(L+1)$. In sum, the length of the encoded data is at most $2^l + \frac{t}{L}(H(\mathscr{X})(1+\epsilon)L + 1) + \delta\frac{t}{L}(L+1)$. The dominant term of them is $\frac{t}{L}(H(\mathscr{X})(1+\epsilon)L + 1)$, and we can lead the result. ■

As we see the above theorem, this encoding scheme is not elegant and practical because it is not easy to find L, k from an unknown distribution.

3.2 LEMPEL-ZIV CODING

We now describe another more elegant and simple scheme for universal encoding. The algorithm defines simply as follows,

- (Parsing) Parse the source data $(x_1, x_2, \ldots, x_t)$ into t' sections $(y_1, y_2, \ldots, y_{t'})$ such that

$$\begin{aligned}&\forall j, \forall j' < j,\ y_j \neq y_{j'}\\ &\forall j, \exists j' < j,\ y_j = y_{j'}b\ (b \in \{0,1\})\end{aligned}$$

- (Encoding) Encode each section y_j $(1 \le j \le t')$ to (j', b).

We just touch the overview of analysis that this encoding scheme is a good compressor. There are two ideas to prove the statement.

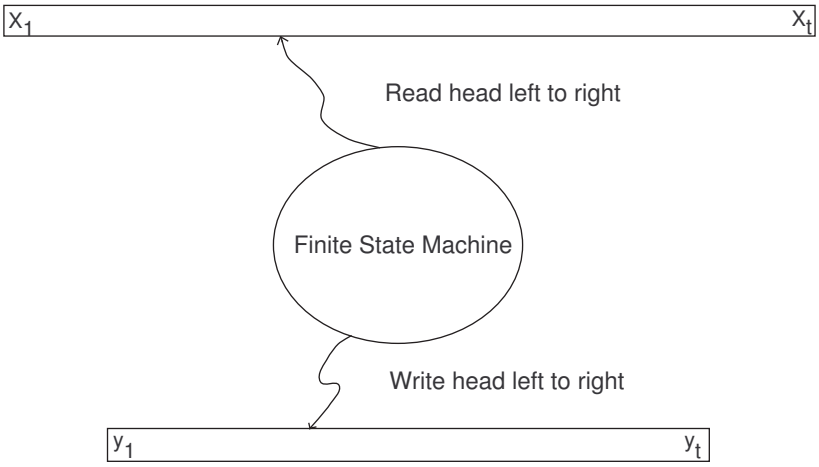
- Lempel Ziv compression is no worse than any finite-state compressor.

$$\forall S \quad \lim_{t\to\infty}\left\{\frac{LZ(\text{comp.})}{Cs(\text{comp.})}\right\} \le 1$$

- Finite state compressors are not too bad.

Sketch of Proof

1. What is finite state machine?



$$\begin{cases} S = \{1, \ldots, s\} \\ \delta(s, X_i) \to s' \\ W(s, X_i) \to y \in \{0,1\}^* \text{ (we can write nothing or many strings at a time.)} \end{cases}$$

Finite state machine consists of finite number states. In each state, machine reads the input sequences $\{X_i\}$, and based on the current state and input value, it determines next state and output sequences $\{Y_i\}$. All procedures in finite state machine are deterministic.

2. The finite state compression is not too bad. For example, 'Shabby' is $\approx 2^L$ state compressor, and we checked in the previous section that it is not too bad.

3. Lemple Ziv is no worse than any finite state compressor.

   Let $c(X_1, \ldots, X_t)$ be the maximum value of t' such that $\exists\ y_1, y_2, \ldots, y_{t'}, X_1X_2 \ldots X_t = y_1y_2 \ldots y_{t'}$ and y_i are all distinct. In other words, it can be interpreted as the largest number of distinct strings into which $X_1X_2 \ldots X_t$ can be parsed.

   Claim 2 *if* $c(X_1, \ldots, X_t) = t'$ *then* $t \geq t' \log\left(\frac{t'}{4}\right)$

   Claim 3 *if* $c(X_1, \ldots, X_t) = t'$ *then for every* S *state compressor* Cs*,*

   $$|Cs(X_1, \ldots, X_t)| \geq t' log\left(\frac{t'}{4S^2}\right) \textit{ (by the Jensen's inequality)}$$

   Claim 4 *if* $c(X_1, \ldots, X_t) = t'$ *then*

   $$|C_{LZ}(X_1, \ldots, X_t)| \leq t' log\left(t'(1 + o(1))\right)$$

■

Lempel-Ziv algorithm is practically very elegant, however analyzing it is very messy and complicate.

6.441 Transmission of Information	Mar 14, 2006
Lecture 10	
Lecturer: Madhu Sudan	*Scribe: Srujan Linga*

1 Last lecture

- Universal Coding
- Lempel-Ziv Algorithm

2 Today

- Channel capacity
- Sample channels and their capacities
- AEP for channels

3 Communication System Overview

The block diagram in Fig.1 shows an overview of the communication system.

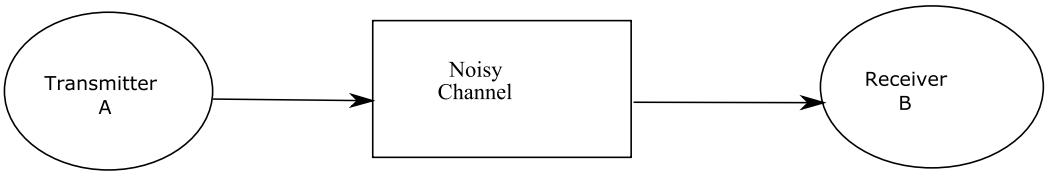


Figure 1: Block diagram of the communication system

The transfer of information from the transmitter to the receiver is a physical process and therefore is subject to noise and imperfections of the signalling process itself. Hence it is a property of physics that there are no perfect channels. A modified block diagram with an encoder and decoder introduced into the communication system is shown in Fig.2. Source symbols S from some finite alphabet Ω_S are mapped into some sequence of channel symbols X of alphabet Ω_X. The output sequence Y of the channel (input to the decoder) is random but has a distribution that depends on the input sequence X. From the output sequence, we attempt to recover the transmitted message.

3.1 Basic features of the channel

Considering a block of n channel symbols, let

1. $p_{\mathbf{X}^n}(\mathbf{x}^n)$ denote the probability distribution on an n-element sequence from Ω_X which is under the designer's control.
2. $p_{\mathbf{Y}^n|\mathbf{X}^n}(\mathbf{y}^n|\mathbf{x}^n)$ denote the probability that $\mathbf{y}^n$ is received given $\mathbf{x}^n$ was transmitted.
3. $\mathbf{P}_{\mathbf{Y}^n|\mathbf{X}^n}(\mathbf{y}^n|\mathbf{x}^n)$ denote an $|\Omega_X|^n \times |\Omega_Y|^n$ stochastic probability transmission matrix of the channel.

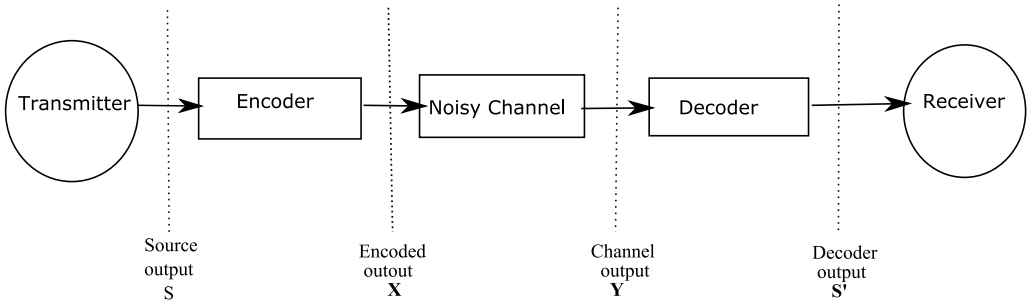


Figure 2: Block diagram with encoder and decoder

3.2 Channel Capacity

Let the capacity of the channel transmitting n-length sequences be given by $\mathbb{C}^{(n)}$, the n-fold capacity of the channel. Then,

$$\mathbb{C}^{(n)} = \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\frac{1}{n}\mathbf{I}(\mathbf{X}^n; \mathbf{Y}^n))$$

We would like to understand the behavior of $\mathbb{C}^{(n)}$ as $n \to \infty$.

3.3 Channel Classes

The classes of channels we want to consider today are Discrete Memoryless Channels (DMC's). These channels have the following properties:

1. Discreteness: Both Ω_X and Ω_Y are finite sets.
2. Memoryless: The behavior of the channel at time t is independent of the time and past inputs/outputs of the channel. More precisely,

$$p_{\mathbf{Y}^n|\mathbf{X}^n}(\mathbf{y}^n|\mathbf{x}^n) = \prod_{i=1}^{n} p_{\mathbf{Y}|\mathbf{X}}(y_i|x_i)$$

Roughly, outputs of memoryless channels capture the same idea as *i.i.d* outputs of the source.

3.4 Capacity of a Discrete Memoryless Channel

$$\mathbb{C}^{(n)} = \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\frac{1}{n}\mathbf{I}(\mathbf{X}^n;\mathbf{Y}^n)) \tag{1}$$

$$= \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\frac{1}{n}\mathbf{I}(\mathbf{Y}^n;\mathbf{X}^n)) \tag{2}$$

$$= \frac{1}{n} \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\mathbf{H}(\mathbf{Y}^n) - \mathbf{H}(\mathbf{Y}^n|\mathbf{X}^n)) \tag{3}$$

$$= \frac{1}{n} \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\sum_{i=1}^{n} \mathbf{H}(Y_i|Y_{i-1}, Y_{i-2}...Y_1) - \mathbf{H}(\mathbf{Y}^n|\mathbf{X}^n)) \tag{4}$$

$$\leq \frac{1}{n} \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\sum_{i=1}^{n} \mathbf{H}(Y_i) - \mathbf{H}(\mathbf{Y}^n|\mathbf{X}^n)) \tag{5}$$

$$= \frac{1}{n} \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\sum_{i=1}^{n} \mathbf{H}(Y_i) - \sum_{i=1}^{n} \mathbf{H}(Y_i|X_i)) \tag{6}$$

$$= \frac{1}{n} \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\sum_{i=1}^{n} (\mathbf{H}(Y_i) - \mathbf{H}(Y_i|X_i))) \tag{7}$$

$$= \max_{p_{\mathbf{X}^n}(\mathbf{x}^n)} (\sum_{i=1}^{n} \frac{\mathbf{I}(Y_i;X_i)}{n}) \tag{8}$$

where equation (2) arises due to symmetry of mutual information, equation (4) arises because of the chain rule of entropy, inequality (5) arises because conditioning only reduces the entropy and the property of discrete memoryless channel was used in (6) to expand $\mathbf{H}(\mathbf{Y}^n|\mathbf{X}^n)$. The result in (8) tells us that maximizing $\mathbf{I}(\mathbf{Y}^n;\mathbf{X}^n)$ over $p_{\mathbf{X}^n}(\mathbf{x}^n)$ is equivalent to maximizing $\mathbf{I}(Y_i;X_i)$ for each i from 1 to n. Therefore $p_{\mathbf{X}^n}(\mathbf{x}^n)$ may well be a product distribution, i.e. we may choose X_i to be $i.i.d$ random variables so that $p_{\mathbf{X}^n}(\mathbf{x}^n) = \prod_{i=1}^{n} p_{X_i}(x_i)$ and $p_{X_i}(x_i) = p_X(x)$ for each i. Now, since

$$\max_{p_X(x)} (\mathbf{I}(Y;X)) = \mathbb{C}^{(1)}$$

we have,

$$\mathbb{C}^{(n)} = \frac{\sum_{i=1}^{n} \mathbb{C}^{(i)}}{n} = \mathbb{C}^{(1)}$$

Therefore, n-fold usage of the channel is no greater than 1-fold usage in terms of the channel capacity. Note that if we choose $X_i's$ to be independent, $Y_i's$ are also independent due to the property of memoryless channel and hence the channel capacity can be achieved with equality. So, for independent X_i, $C^{(n)} = C^{(1)} = \max_{p_X(x)}(\mathbf{I}(Y;X))$.

4 Examples of Channel Capacity

4.1 Binary Erasure Channel (BEC)

Consider the Binary Erasure Channel shown in Fig.3. A BEC has two inputs 0 and 1 and a fraction p of the bits are erased. The receiver knows which of the bits have been erased. We calculate the capacity of the channel as follows,

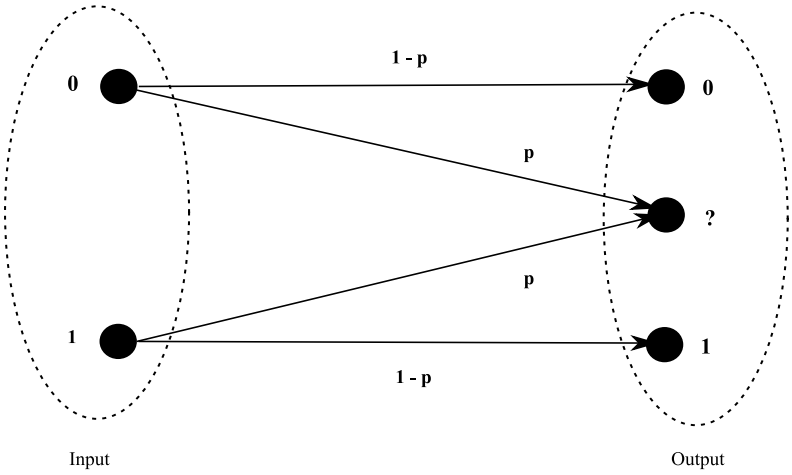


Figure 3: Binary Erasure Channel

$$
\begin{aligned}
\mathbb{C}^{(1)} &= \max_{p_X(x)} (\mathbf{I}(X;Y)) \\
&= \max_{p_X(x)} (\mathbf{H}(X) - \mathbf{H}(X|Y)) \\
&= \max_{p_X(x)} (\mathbf{H}(X) - \underbrace{\mathbf{H}(X|Y=?)}_{=\mathbf{H}(X)} \underbrace{Pr(Y=?)}_{=p}) \\
&= \max_{p_X(x)} (\mathbf{H}(X) - \mathbf{H}(X)p) \\
&= \max_{p_X(x)} (\mathbf{H}(X)(1-p)) \\
&= 1-p
\end{aligned}
$$

where the last equality was derived because maximum value of $H(X)$ is 1 bit. This result gives us the following intuition: *a)* One will be able to "push" through roughly $(1-p)$ bits every time unit; *b)* For every n bits one transmits, one would expect to see roughly $n(1-p)$ bits without erasures. This insight leads us to the following encoding scheme which asymptotically achieves zero error probability: If the encoder picks up source strings ($\mathbf{S}$) of length $n(1-p)$ and encodes them into channel symbols ($\mathbf{X}$) of length n, i.e. we add some form of redundancy to the source sequences, then by the above observations, the output sequence ($\mathbf{Y}$) has approximately $n(1-p)$ correct symbols which can then be decoded to $\mathbf{S}^{'}$ at the receiver. For this scheme it can be shown that, asymptotically, $Pr(\mathbf{S} = \mathbf{S}^{'}) = 1$ as $n \to \infty$. This scheme is depicted in Fig.4.

4.2 Binary Symmetric Channel (BSC)

Consider the Binary Symmetric Channel shown in Fig. 5. The capacity of a BSC can be calculated as follows:

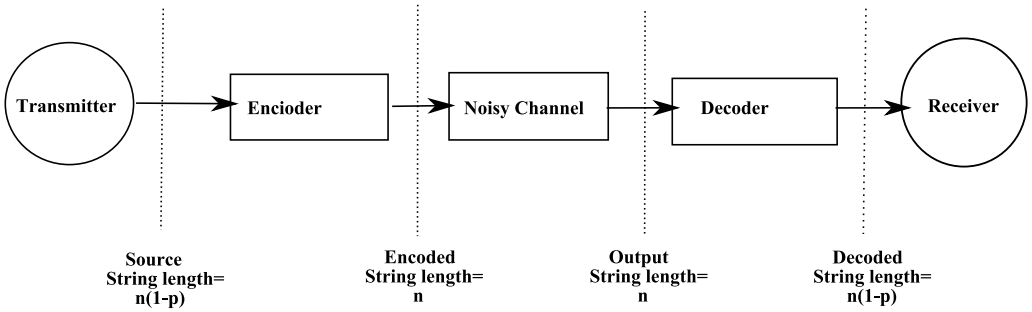


Figure 4: Proposed encoding scheme

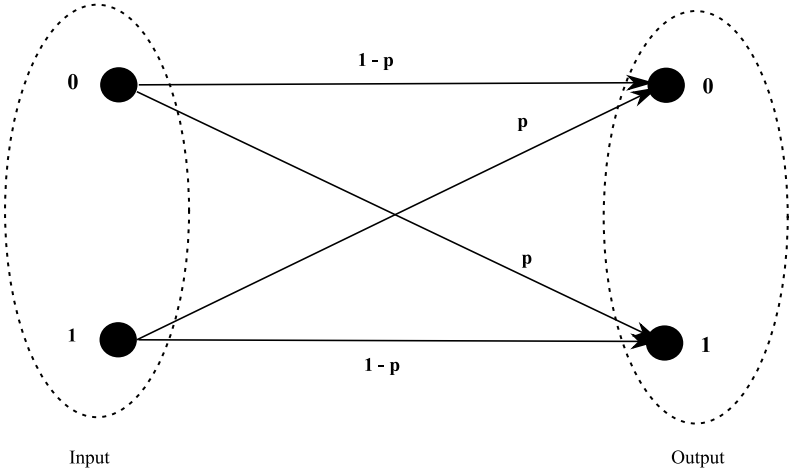


Figure 5: Binary Symmetric Channel

$$
\begin{aligned}
\mathbb{C}^{(1)} &= \max_{p_X(x)} (\mathbf{I}(X;Y)) \\
&= \max_{p_X(x)} (\mathbf{H}(Y) - \underbrace{\mathbf{H}(Y|X)}_{\mathbf{H}(p)}) \\
&= \max_{p_X(x)} (\mathbf{H}(Y) - \mathbf{H}(p)) \\
&= \max_{p_X(x)} (\mathbf{H}(Y)) - \mathbf{H}(p) \\
&\leq 1 - \mathbf{H}(p)
\end{aligned}
$$

where the final inequality is achieved if $p_X(x)$ is a uniform distribution.

4.3 Noisy Typewriter

In this case, the channel input is either received unchanged at the output with probability $\frac{1}{2}$ or transformed into the next letter with probability $\frac{1}{2}$. The channel transition probability matrix for such a channel is shown in Fig.6.

INPUT

OUTPUT

	A	B	C	D		Y	Z
A	½	½	0	0	.....	0	0
B	0	½	½	0	.....	0	0
C	0	0	½	½	.....	0	0
D	0	0	0	½	.....	0	0
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
Y	0	0	0	0	.....	½	½
Z	1/2	0	0	0	.....	0	½

Figure 6: Channel matrix for the noisy typewriter

The channel transition matrix, $\mathbf{P}_{\mathbf{Y}^n|\mathbf{X}^n}(\mathbf{y}^n|\mathbf{x}^n)$ is symmetric and has the following properties:

1. Every row of $\mathbf{P}_{\mathbf{Y}^n|\mathbf{X}^n}$ is a permutation of the first row.
2. Every column of $\mathbf{P}_{\mathbf{Y}^n|\mathbf{X}^n}$ is a permutation of the first column.

For such a channel,

$$
\begin{aligned}
\mathbb{C}^{(1)} &= \max_{p_X(x)} (\mathbf{I}(X;Y)) \\
&= \max_{p_X(x)} (\mathbf{H}(Y) - \mathbf{H}(Y|X)) \\
&\leq \max_{p_X(x)} (\mathbf{H}(Y)) - \min_{p_X(x)} (\mathbf{H}(Y|X)) \\
&= \max_{p_X(x)} (\mathbf{H}(Y)) - (\text{Entropy of the first row}) \\
&\leq \log(|\Omega_Y|) - (\text{Entropy of the first row})
\end{aligned}
$$

where the final inequality is satisfied if $p_X(x)$ is a uniform distribution. Therefore for the noisy typewriter, $\mathbb{C}^{(1)} \leq \log(26) - 1 = \log(13)$ bits.

6.441 Transmission of Information Mar 16, 2005

Lecture 11

Lecturer: Madhu Sudan *Scribe: Sang Joon Kim*

Today we will talk about coding theorem for symmetric channel.

1 Admin

- I will be out of town next week. Chung is in charge.
- Midterm is in 7 days.

2 Review

First, we review what we did last time.

2.1 DMC(Discrete Memoryless Channel)

DMC(Discrete Memoryless Channel) is the channel that can be repeatedly used for each transmitting information. The detail description is as following:

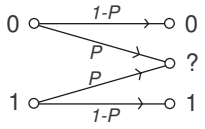
$$\begin{cases} \Omega_x - \text{ input alphabet} \\ \Omega_y - \text{ output alphabet} \\ P_{y|x}(y|x)_{y\in\Omega_y, x\in\Omega_x} - \text{ transition probability matrix} \end{cases}$$

Definition 1

$$\textit{Capacity} \triangleq \max_{X\sim P_X, Y\sim P_{Y|X}} I(X;Y)$$

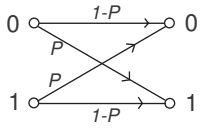
For maximizing the channel capacity, we choose the set X and determine the distribution of X,(P_X). Y is the received data set that is correlated random variable of X by $P_{Y|X}$.

2.2 BEC(Binary Erasure Channel)



BEC is like above and Capacity $= 1 - P$. If $P = 0.9$ then, we lose 90 % of information in this channel, i.e. for one successful transmission, we should try average 10 times transmission.

2.3 BSC(Binary Symmetric Channel)



Capacity of BSC $= 1 - H(P)$. BSC is less useful channel than BEC but more reasonable channel. For example, if $P = 0.1$ than, $H(0.1)$ is much larger than 0.1 and the channel capacity of BSC is less than that of BEC. Also, if we assume that $P = 0.5$ than, we don't need to send any information because we get no information of transmitted data from received data. One more example is that if $P = 0.49$ then, capacity $\approx 10^{-4}$. It means that we should retransmit 10000 times for one sending. This is obviously not very reliable situation and our question is that "how can we achieve a good capacity for this channel?".

2.4 Symmetric Channel

Symmetric channel is the channel that satisfies the following properties:

$$P_{Y|X} \text{ is } \begin{cases} \text{all rows are permutation of each other.} \\ \text{all columns are permutation of each other.} \end{cases}$$

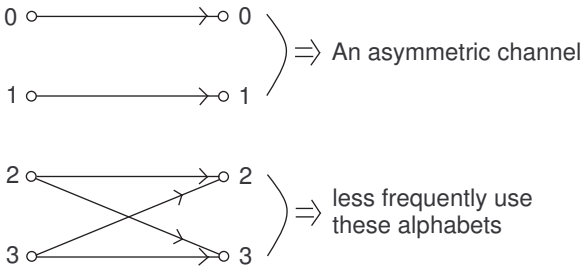
Example

$$P_{Y|X} = \begin{bmatrix} .21 & .21 & .21 & .21 & .16 \\ .16 & .21 & .21 & .21 & .21 \\ .21 & .16 & .21 & .21 & .21 \\ .21 & .21 & .16 & .21 & .21 \\ .21 & .21 & .21 & .16 & .21 \end{bmatrix}$$

i^{th} row represents the probability of Y given $X = x_i$.

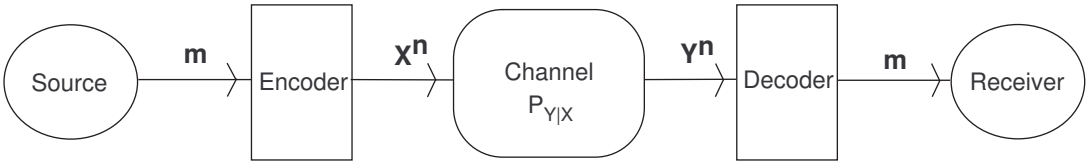
Capacity of symmetric channel $= \log |\Omega_y| - H(\text{first row})$. When X is uniformly distributed we achieve $H(Y) = \log |\Omega_y|$. We don't have a clue which message is transmitted if the row is uniformly distributed.

Example



In the above channel, if there are some x_i that are rarely transmitted then we don't use the uniform distribution. Also, there may be an asymmetric channel but we are only looking at symmetric case in the class.

3 Coding theorem



$$\begin{cases} E & : \{0,1\}^k \to \Omega_x^n \\ D & : \Omega_y^n \to \{0,1\}^k \end{cases}$$

$$m = D(\bar{Y}) \leftarrow \bar{Y} \leftarrow P_{Y|X}(\bar{X}) \leftarrow \bar{X} \leftarrow E(m)$$

The purpose of communication is not to guarantee absolutely the successful transmission, but to increase the reliability of the channel.

Definition 2

$$\textit{Decoding Error Rate} \triangleq Pr_{m\in\{0,1\}^k,\bar{Y}}\left[m \neq D(\bar{Y})\right]$$
$$\textit{where } \bar{Y} = \textit{channel}(E(m))$$

We assume that X is uniformly distributed because X is in the typical set. What we want to do is to make the decoding error rate go to 0 very fast and it is the same as increasing the reliability of the channel. If we have an enough time to decode $\bar{Y}$, looking all message and finding the best one - most likely probability is the optimum.

Optimal Decoding Algorithm D for fixed E

$$D(y^n) \rightarrow m_0 = arg\max_m \{Pr[E(m)] \cdot Pr[y^n|E(m)]\}$$
$$\text{(if we assume the uniform distribution for } E(m))$$
$$= arg\max_m \left\{\frac{1}{2^k}\prod_{i=1}^{n} P_{Y|X}(y_i|E(m)_i)\right\}$$

This is the same as maximum a posterior probability.

Our goal is to find the encoding function that allow to achieve following property:

$$\boxed{\frac{k}{n} \rightarrow \text{Capacity}}$$

3.1 BEC

In BEC, $Pr[\bar{Y} = ?^n] = P^n > 0$. This means that there is always probability that we fail to decode the received information.

$$m_1, \ldots, m_k \Rightarrow X^n = (x_1, \ldots, x_n) \Rightarrow (x_1, x_2, ?, x_4, ?, ?, x_7, \ldots, x_n)$$
$$\text{Channel sends} \{x_j\}_{j\in S}, \quad S \subseteq \{1, 2, \ldots, n\}$$

In above situation, message $\{m_i\}$ is encoded to X^n and transmitted. The received information has several ?s those are erased data. What we want to do is as following:

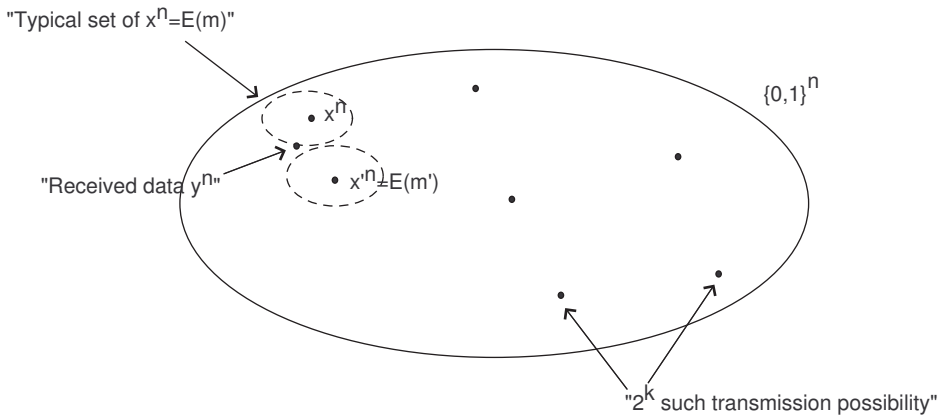
$$I(m_1, \ldots, m_k; \{x_j\}_{j\in S}) \approx k$$

That mutual information between $\{m_i\}$ and received data is approximately k bits means we can decode almost exactly whatever the received data is. This is our goal.

$$|S| \approx (1-P)n$$
$$\frac{k}{n} \approx \text{Capacity} \Leftrightarrow k \approx (1-P)n$$

Challenge : How to encode $m_1, \ldots, m_k \rightarrow x_1, \ldots, x_n$ so that almost every subset of size $(1+\epsilon)k$ give k bits of information about $m_1, \ldots, m_k$?

3.2 BSC



In BSC, if $m = (m_1, m_2, \ldots, m_k)$ is transmitted via the encoded data $x^n = (x_1, x_2, \ldots, x_n)$, the typical set of the received data $y^n = (y_1, y_2, \ldots, y_n)$ forms a geometric circle in $\{0,1\}^n$ space which has x as its origin. (The geometric distance between $x, y \in \{0,1\}^n$ is defined as the number of coordinate where they differ.) Therefore, our challenge is following.

Challenge : How to design the encoding function E so that $E(m)$ and $E(m')$ is far for most pair m, m'?

3.3 Shannon Encoding Function

Pick the Encoding Funtion $E : \{0,1\}^k \to \Omega_x^n$ as follows,

For every $m \in \{0,1\}^k$,
$E(m)$ is chosen uniformly from Ω_x^n and independently from $E(m')$ for all $m' \neq m$

Then, the following lemma holds.

Lemma 3 *If $k = R \cdot n$ for some $R < C$(capacity of the channel), then*

$$\lim_{n\to\infty} \Pr_{E,m,\bar{y}=channel(E(m))}[m \neq D(\bar{y})] = 0$$

The lemma implies the following.

$$\lim_{n\to\infty} \min_{E}\{\Pr_{m,\bar{y}=channel(E(m))}[m \neq D(\bar{y})]\} = 0$$

In other words, the lemma says all rates below capacity of the channel are achievable. Now, define the following typical set.

Definition 4 *For $\bar{x}^n \in \Omega_x^n$, define the set $A_{\epsilon,\bar{x}}^{(n)}$ as follows,*

$$A_{\epsilon,\bar{x}}^{(n)} \triangleq \{\bar{y}^n \in \Omega_y^n | \Pr[\bar{y} \text{ received } | \bar{x} \text{ transmitted }] \geq 2^{(-H(r)-\epsilon)n}\}$$

,where r is the first row of the transition matrix of the channel.

Then, we can easily check the following claim.

Claim 5 $\forall \bar{x}$, $|A_{\epsilon,\bar{x}}^{(n)}| \leq 2^{(H(r)+\epsilon)n}$.

Now, we think the following two events. The first event is the event of receiving $\bar{y} \notin A^{(n)}_{\epsilon,\bar{x}}$ when transmitting $\bar{x} = E(m)$. AEP tells the probability that this event happens goes to 0 as $n \to \infty$. The second event is the event of receiving $\bar{y} \in A^{(n)}_{\epsilon,E(m')}$ such that $\exists m' \neq m$ when transmitting $\bar{x} = E(m)$. The probability that this event happens is greatest when $\bar{y}$ distributes uniformly in Ω_y^n, because this channel is symmetric. Therefore, the probability that this event happens is

$$\begin{aligned}\sum_{m'} Pr(y \in A^{(n)}_{\epsilon,E(m')}) &\leq \sum_{m'} \frac{|A^{(n)}_{\epsilon,E(m')}|}{|\Omega_y|^n} \leq \sum_{m'} \frac{2^{(H(r)+\epsilon)n}}{|\Omega_y|^n} \\ &\leq 2^k 2^{(H(r)-\log|\Omega_y|)n+\epsilon n} = 2^k 2^{-Cn+\epsilon n} = 2^{-\epsilon n} \text{ if } k = Rn, \epsilon = \frac{C-R}{2}\end{aligned}$$

This implies that the probability that this event happens goes to 0 as $n \to \infty$.

6.441 Transmission of Information	March. 21, 2006
Lecture 12	
Lecturer: Madhu Sudan	*Scribe:* Costas Pelekanakis

1. Today's outline

a. Joint Typicality

b. Channel Coding Theorem for DMC

c. Achievability of $R < C = \max_{p(x)} \{I(x;y)\}$

d. Nonachievability of $R > C$

2. Definitions

- DMC : ($\underbrace{X}_{\text{finite set}}$, $\underbrace{P_{y/x}}_{\text{transition probability matrix}}$, $\underbrace{Y}_{\text{finite set}}$)
- N[th] extension of DMC: $(X^n, P_{y^n/x^n}, Y^n)$. Check for properties of DMC:
  - memoryless $\Leftrightarrow X_{i-1} \to X_i \to Y_i$
  - no feedback $\Leftrightarrow Y_1,...Y_{i-1} \to X_1,...,X_{i-1} \to X_i$
  - memoryless + no feedback $\Leftrightarrow (X_1,...X_{i-1}, Y_1,..., Y_{i-1}) \to X_i \to Y_i \Leftrightarrow P_{Y^n/X^n} = \prod_{i=1}^{n} P_{Y_i/X_i}$
- (M,n) code
  - message index set $\{1,...,M\}$
  - encoding function $f : W \to X^n$
  - codebook: $C = \begin{bmatrix} X_1(1) & X_2(1).....X_n(1) \\ X_1(2) & X_2(2).....X_n(2) \\ .... & \\ X_1(M) & X_2(M).....X_n(M) \end{bmatrix}$
  - decoding function $g : Y^n \to W$
  - For M uniformly distributed messages, Rate = bits/channel uses = $\log_2(M)/n$. If we fix rate $R = \log_2(M)/n \Rightarrow M = 2^{nR}$.
- Probability of error conditioned on i[th] message sent $\lambda_i = \Pr\left[g(Y^n) \neq i / X^n = f(i)\right]$

- Maximum probability of error: $\lambda^{(n)} = \max_i \{\lambda_i\}$
- Arithmetic average probability of error: $P_e^{(n)} = \frac{1}{M}\sum_{i=1}^{M} \lambda_i$. Obviously, $P_e^{(n)} \le \lambda^{(n)}$
- A rate is said to be *achievable* if there exists a sequence of $\left(\lfloor 2^{nR} \rfloor, n\right)$ codes such that $\lambda^{(n)}$ tends to zero as n increases.
- The capacity of a DMC is the supremum of all the achievable rates.

3. Jointly typical sequences

Recall that a random vector X^n with i.i.d. components according to p_x is a typical sequence if $2^{-n(H(X)+\varepsilon)} \le p(X^n = x^n) \le 2^{-n(H(X)-\varepsilon)}$. We extend the notion of typical sequences to jointly typical sequences. We have two random vectors X^n and Y^n with i.i.d. components according to p_x and p_y respectively. In addition, $(X_i, Y_i) \sim p_{x,y}$. Hence, $p(X^n = x^n, Y^n = y^n) = \prod_{i=1}^{n} p(X_i = x_i, Y_i = y_i)$.

3.1. Definition: Jointly typical sequences

The set A_ε^n of jointly typical sequences (x^n, y^n) is defined as:

$$A_\varepsilon^{(n)} = \left\{ \begin{array}{l} (x^n, y^n): \\ \qquad 2^{-n(H(X)+\varepsilon)} \le p(X^n = x^n) \le 2^{-n(H(X)-\varepsilon)} \\ \qquad 2^{-n(H(Y)+\varepsilon)} \le p(Y^n = y^n) \le 2^{-n(H(Y)-\varepsilon)} \\ \qquad 2^{-n(H(X,Y)+\varepsilon)} \le p(X^n = x^n, Y^n = y^n) \le 2^{-n(H(X,Y)-\varepsilon)} \end{array} \right\}$$

where

$$p(X^n = x^n, Y^n = y^n) = \prod_{i=1}^{n} p(X_i = x_i, Y_i = y_i)$$

3.2. Theorem: Joint AEP

Let (X^n, Y^n) *be a random sequence with i.i.d. pairs* $(X_i, Y_i) \sim p_{x,y}$ *and*

$p(X^n = x^n, Y^n = y^n) = \prod_{i=1}^{n} p(X_i = x_i, Y_i = y_i)$. *Then the following are true:*

1. $\Pr\left(\left(X^n, Y^n\right) \in A_\varepsilon^n\right) \to 1 \text{ as } n \to \infty.$
2. $\left|A_\varepsilon^n\right| \le 2^{n(H(X,Y)+\varepsilon)}$

3. *If* $\widetilde{X}^n$, $\widetilde{Y}^n$ *are independent with i.i.d. components and* $\widetilde{X}_i \sim p_x$ *and* $\widetilde{Y}_i \sim p_y$ *then for sufficient large n:* $\Pr\left(\left(\widetilde{X}^n, \widetilde{Y}^n\right) \in A_\varepsilon^n\right) \le 2^{-n(I(X;Y)-3\varepsilon)}$ and $\Pr\left(\left(\widetilde{X}^n, \widetilde{Y}^n\right) \in A_\varepsilon^n\right) \ge (1-\varepsilon)2^{-n(I(X;Y)+3\varepsilon)}$

Proof:

1 and 2 are derived from $2^{-n(H(X,Y)+\varepsilon)} \le p(X^n = x^n, Y^n = y^n) \le 2^{-n(H(X,Y)-\varepsilon)}$. We have shown a similar proof when we talked about typical sets. Here we give the proof for 3:

$$\Pr\left(\left(\widetilde{X}^n, \widetilde{Y}^n\right) \in A_\varepsilon^n\right) = \sum_{(x^n, y^n) \in A_\varepsilon^n} p(x^n) p(y^n) \le 2^{n(H(X/Y)+\varepsilon)} 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \le 2^{-n(I(X;Y)-3\varepsilon)}$$

$$\Pr\left(\left(\widetilde{X}^n, \widetilde{Y}^n\right) \in A_\varepsilon^n\right) = \sum_{(x^n, y^n) \in A_\varepsilon^n} p(x^n) p(y^n) \ge (1-\varepsilon) 2^{n(H(X/Y)-\varepsilon)} 2^{-n(H(X)+\varepsilon)} 2^{-n(H(Y)+\varepsilon)} \ge (1-\varepsilon)2^{-n(I(X;Y)+3\varepsilon)}$$

Comments: Not all pairs of typical X^n and typical Y^n are jointly typical since there exist approximately $2^{nH(X;Y)}$ typical pairs. Each typical X^n induces about $2^{nH(Y/X)}$ possible Y^n typical sequences, all of them equally likely. If we want to ensure that two different codewords will induce two disjoint sets of typical Y^n possible sequences then the total number of $2^{nH(Y)}$ typical Y^n must be divided into $2^{n(H(Y)-H(Y/X))} = 2^{nI(X;Y)}$ disjoint sets. Hence, we are allowed to send at most $2^{nI(X;Y)}$ different typical X^n sequences.

4. Channel Coding for the DMC

4.1. The Channel Coding Theorem

All rates below capacity C are achievable, namely, for every R<C there exists a sequence of $\left(2^{nR}, n\right)$ *codes with* $\lambda^{(n)} \overset{n\to\infty}{\to} 0$. *Conversely, any sequence of* $\left(2^{nR}, n\right)$ *codes with* $\lambda^{(n)} \overset{n\to\infty}{\to} 0$ must have R ≤ C.

Achievability:

Note that $\lambda^{(n)}$ is not easy to deal because it involves maximization which is a nonlinear operation while $P_e^{(n)}$ is something we can compute. But $P_e^{(n)} \le \lambda^{(n)}$, so let's try to generate a code $\aleph$ (*MxN* matrix of symbols) for which when $P_{e\aleph}^{(n)} \overset{n\to\infty}{\to} 0$ and show that there is a subcode $\aleph'$ of $\aleph$ (*TxN* submatrix of symbols, *T*<*M*) such that $\lambda_{\aleph'}^{(n)} \overset{n\to\infty}{\to} 0$. This is sufficient if $\lambda_{\aleph'}^{(n)} \le K \cdot P_{e\aleph}^{(n)}$. We do the following trick. We generate a code of *2M* codewords instead of *M*. Note that the new rate is $R_{2M} = \log_2(2M)/n = R_M + 1/n \overset{n\to\infty}{\to} R_M$. It is easy to show that $2P_{e2M}^{(n)} \ge \lambda_M^{(n)}$. The proof goes as follows:

assume that $\lambda_1 \le \lambda_2 \le ... \le \lambda_{2M}$. If this does not hold you can always swap the codewords in the codebook such that the first row has the smallest probability or error, the second row the second smallest probability of error, etc. Hence:

$$P_{e2M}^{(n)} = \frac{1}{2M}\sum_{i=1}^{2M}\lambda_i = \frac{1}{2M}\left(\sum_{i=1}^{M}\lambda_i + \sum_{i=M+1}^{2M}\lambda_i\right) \ge \frac{1}{2M}\sum_{i=M+1}^{2M}\lambda_i \ge \frac{1}{2M}M\lambda_M^{(n)} \ge \frac{\lambda_M^{(n)}}{2}.$$

Thus, using the code $\aleph$ and showing $P_{e\aleph}^{(n)} \overset{n\to\infty}{\to} 0$ it is equivalent as if we were using the subcode $\aleph'$ and showing $\lambda_{\aleph'}^{(n)} \overset{n\to\infty}{\to} 0$. In practice, we can throw away the worst *M* codewords and left with the rest *M* codewords (this is also called the expurgated code $\aleph'$ which has rate $R_{2M} - 1/n$).

Let's calculate the average probability of error averaged over all codewords in the codebook and averaged over all codebooks:

$$P\{error\} = \sum_{\aleph} \mathrm{P}\{\aleph\}P_e^{(n)}(\aleph) = \sum_{\aleph} \mathrm{P}\{\aleph\}\frac{1}{2^{nR}}\sum_{w=1}^{2^{nR}}\lambda_w(\aleph) = \frac{1}{2^{nR}}\sum_{w=1}^{2^{nR}}\sum_{\aleph} \mathrm{P}\{\aleph\}\lambda_w(\aleph)$$

by symmetry of the code construction, $P\{error\}$ does not depend on the particular message so:

$$P\{error\} = \sum_{\aleph} \mathrm{P}\{\aleph\}\lambda_1(\aleph) = P\{error / W = 1\}$$

So we calculate the average probability of error based on the scenario that the first codeword was transmitted. An error occurs if the following events happen:

1. $E_i = \{(X^n(i), Y^n) \in A_\varepsilon^{(n)}\},\ i = 2,...,2^{nR}$
2. $E_1^c = \{(X^n(1), Y^n) \notin A_\varepsilon^{(n)}\}$

Thus,

$$P\{error / W = 1\} = P\{E_1^C \cup E_2 \cup ... \cup E_{2^{nR}}\} \le P\{E_1^C\} + \sum_{i=2}^{2^{nR}} E_i \overset{n\to\infty}{\le} \varepsilon + \sum_{i=2}^{2^{nR}} 2^{-n(I(X,Y)-3\varepsilon)}$$

$$\le \varepsilon + (2^{nR}-1)2^{-n(I(X,Y)-3\varepsilon)} \le \varepsilon + 2^{-n(I(X,Y)-R-3\varepsilon)} \underset{n\to\infty}{\overset{R<I(X;Y)}{\to}} 0$$

To finish the proof, we find the capacity achieving input distribution to generate the codebooks so $C = I(X;Y)$. Hence, we can drive the average probability of error as close to zero as desired as long as $R < C$ for sufficient large n.

Converse:

Let *W* to be uniformly distributed over the set $\{1,2,...,2^{nR}\}$. We have:

$$nR = H(W) = H(W/Y^n) + I(W;Y^n) \overset{W\to X^n(W)\to Y^n}{\le} H(W/Y^n) + I(X^n(W);Y^n)$$

If $\lambda^{(n)} \overset{n\to\infty}{\to} 0$ then $P_e^{(n)} \overset{n\to\infty}{\to} 0$. Since $P_e^{(n)} = P\left(W \neq g(Y^n)\right)$ by Fano's inequality we have:

$$H(W/Y^n) \leq 1 + P_\varepsilon^{(n)} nR$$

Moreover,

$$I(X^n(W);Y^n) \overset{DMC}{\leq} \sum_{i=1}^{n} I(X_i;Y_i) \leq nC$$

thus,

$$nR \leq H(W/Y^n) + I(X^n(W);Y^n) \leq 1 + P_\varepsilon^{(n)} nR + nC \Rightarrow R \leq P_\varepsilon^{(n)} R + \frac{1}{n} + C$$

so as $n \to \infty$ the first two terms go to 0 and we get the desired result: $R \leq C$

4.2. Example

When may we encode above capacity and have zero probability error?

Assume the BSC with ε = 0. Obviously C = 1 bit/channel use. Consider the channel code

1 → 0 p=1/4
2 → 1 p=1/4
3 → 00 p=1/4
4 → 01 p=1/4

then $R = \frac{\log_2 4}{2 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2}} = \frac{4}{3} > 1$. Hence, we can have R > C and no errors at the receiver but note that the channel is not noisy anymore so the coding theorem does not hold anymore. Also note the code is not uniquely decodable.

6.441 Transmission of Information	April 4, 2005
Lecture 13	
Lecturer: Madhu Sudan	*Scribe: Shaili Jain*

The lecture intro has been taken from Chung Chan's notes.

1 Break Up of Things We've Seen

We briefly summarize what we have seen so far in this course.

1.1 Phase I: The Tools

- Entropy
- Mutual Information
- AEP

1.2 Phase II: Exercise in Compression

- Source Coding and AEP
- Kraft's Inequality
- Shannon, Huffman, Lempel-Ziv Coding

1.3 Phase III

- Channel Coding
- Channel Capacity
  - Conditional probability distribution between X and Y models the channel
  - We used the ideas of random codign and maximum likelihood decoding
  - Joint AEP lets us say that we can get arbitrarily close to the channel capacity
  - Coding Theorem: For any discrete memoryless channel with capacity C, $\forall R < C$, there exists an encoding from $\{0,1\}^{Rn} \to X^n$, such that $P_{err} = Pr_{m \in \{0,1\}^{Rn}, Channel}[D(Channel(E(m)) \neq m] \to 0$
  - Converse: If $R > C$, then $P_{err} \to 1$

2 Error Exponent

In this lecture we show how to compute the error exponent over a binary symmetric channel using a random code ensemble and using maximum likelihood decoding (the same as minimum distance decoding in this case).

The general idea behind this lecture is based on the following facts: The probability of error of a discrete memoryless channel decays exponentially with n for a fixed rate below capacity. As n becomes large, the error exponent is representative of the quality of the code. Computing the error exponent turns out to be easier and more insightful than computing the probability of error exactly.

For the binary symmetric channel that has capacity, $C = 1 - H(p)$, we can write the error probability as follows: $Pr[Y^n = w] \geq p^n = 2^{-n \log \frac{1}{p}}$

Our goal is to come up with a general expression, where we conclude for any transmission $P_{err} > 2^{-nE_c(R)}$, where $E_c(R)$ is the error exponent. We are interested in how the error exponent $E_c(R)$ behaves. If $E_c(R) = 0$, then we know that the channel is virtually useless. On the other hand, if $E_c(R) = \infty$, then the channel is perfect. We know that $E_c(R) \leq \log_c \frac{1}{p}$, $\forall R > 0$, $\forall$ encoding, $\forall$ decoding (for the binary symmetric channel).

Today we study lower bounds for the error exponent for a random code ensemble.

Random Code For every $m \in \{0,1\}^{Rn}$ pick $E(m)$ uniformly and independently at random from $\{0,1\}^n$.

Decoding Scheme We use Maximum Likelihood Decoding (MLD). Given y, output m that maximizes $Pr[y \text{ received}|E(m) \text{ is transmitted}]$.

The channel capacity for a binary symmetric channel is $C = 1 - H(p)$.
If $R < C$, then we know that $R < 1 - H(p)$
Hence $H(p) < 1 - R$ and $p < H^{-1}(1 - R)$.
Let $P_R = H^{-1}(1 - R)$.
Clearly, $P_R > p$, so we can choose a τ such that $p < \tau < P_R$.

Type I error: $\triangle(E(m), y) \geq \tau n$ (This is the number of errors in transmission).

Pr[Type I error] $\to 0$

Type II error: $\exists\ m' \neq m$ such that $\triangle(E(m'), y) < \tau \cdot n$, where $\triangle$ denotes distance. (This is the case where y is to "close" to a codeword $E(m')$).

Pr[Type II error] $\to 0$ provided that $H(\tau) + R < 1$

We want a formula of the form:

$$Pr[\text{Error with MLD of Random Code}] \leq 2^{-(\ldots)\cdot n}$$

2.1 Maximum Likelihood Decoding

$$Pr[y \text{ received}|E(m) \text{ transmitted}] = p^d(1-p)^{n-d}$$

Let $\triangle(y, x) =$ the number of coordinates where x and y differ. Let $d = \triangle(y, E(m))$. The maximum likelihood decoding method chooses the message m such that $\triangle(y, E(m))$ is minimized.

Clearly, the MLD algorithm is unsuccessful if $\exists m' \neq m$ such that $\triangle(y, E(m')) < \triangle(y, E(m))$

Now consider the following scenario: $\exists \tau, m'$ such that $\triangle(y, E(m')) \leq \tau n \leq \triangle(y, E(m))$. Notice that the first inequality denotes a Type II error and the second inequality denotes a Type I error.

$$\begin{aligned} max_\tau[Pr[\triangle(y, E(m')) \leq \tau n \leq \triangle(y, E(m)))]] &\leq Pr[\text{Decoding Error}] \\ &\leq \Sigma_{\tau n=0}^{n}[\triangle(y, E(m')) \leq \tau n \leq \triangle(y, E(m))] \\ &\leq (n+1) max_\tau\{Pr[\triangle(y, E(m')) \leq \tau n \leq \triangle(y, E(m))]\} \end{aligned}$$

Hence we analyze the expression: $Pr[\triangle(y, E(m')) \leq \tau n \leq \triangle(y, E(m)))]$.

$$Pr[\text{Type I error}] = Pr[\triangle(y, E(m)) \geq \tau n] = \Sigma_{i=\tau n}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$$\approx \binom{n}{\tau n} p^{\tau n}(1-p)^{n(1-\tau} \text{ assuming that } \tau > p$$

$$= 2^{-D(p||\tau)n}$$

$$Pr[\text{Type II error}] = 1 - (1 - \frac{2^{H(\tau)n}}{2^n})^{2^{Rn}-1}$$

$$\text{Fix } m', Pr[\triangle(E(m'), y) \leq \tau n] = \frac{2^{H(\tau)\cdot n}}{2^n}$$

$$Pr[\text{Type II error}] \approx 1 - 1 + 2^{Rn} \cdot \frac{2^{H(\tau)n}}{2^n}$$

since $R + H(\tau) < 1$. Thus we can write the probability of a Type II error as:

$$Pr[\text{Type II error } \approx 2^{-(1-H(\tau)-R)\cdot n}$$

$$Pr[\text{Type I and Type II error} = Pr[\text{Type I error}] \cdot Pr[\text{Type II error}]$$

$$\approx 2^{-[D(\tau||p+1-H(\tau)-R]\cdot n}$$

Notice that $1 - H(\tau) = D(\tau||\frac{1}{2})$, so we can write

$$Pr[\text{Type I and Type II error}] \approx 2^{-[D(\tau||p)+D(\tau||\frac{1}{2})-R]\cdot n}$$

Conclusion For a random code and maximum likelihood decoding, $P_{err} = 2^{-E_{RCE}(R)\cdot n}$, where E_{RCE} was derived to be:

$$E_{RCE}(R) = min_{p\leq\tau\leq P_R}\{D(\tau||p) + D(\tau||\frac{1}{2}) - R\}$$

Which choice of $\tau \in (p, P_R)$ minimizes $D(\tau||p) + D(\tau||\frac{1}{2})$?

We can find this by solving the following equation: $D'(\tau||p) = -D'(\tau||\frac{1}{2})$. We find that in this case, $\frac{\tau}{1-\tau} = \frac{\sqrt{p}}{\sqrt{1-p}}$. We find that when $\frac{\tau}{1-\tau} = \frac{\sqrt{p}}{\sqrt{1-p}}$, $D(\tau||p) + D(\tau||\frac{1}{2}) = 1 - \log(1 + 2\sqrt{p(1-p)}$.

A good reference for today's lecture is a paper by Barg and Forney entitled "Random codes: Minimum distances and error exponents" in IEEE Transactions on Information Theory in Sept 2002.

6.441 Transmission of Information April 6, 2006

Lecture 14

Lecturer: Madhu Sudan *Scribe: Adam McCaughan*

Today

- Feedback Capacity
- Joint Source Channel Coding
- Start Continuous Channels

Admin

- PS3 due Thursday (04/12)
- Tuesday 4:15pm in 32-155 – Venkat Guruswami "'Channel Coding..."'

Feedback Capacity

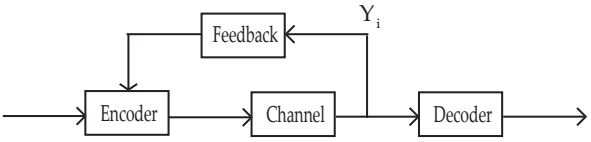
- Recall basic model of a channel

- In order to ask how well the channel performs, we apply an encoder and decoder

- Which more or less pins down exactly how the channel performs

$$C = p_x^{\max} I(X;Y)$$

How much capacity do we get with feedback?



- In other words, given $Y_1, \ldots, Y_n$ what is the maximal R such that the receiver can compute $W_1, \ldots, W_{k=Rn}$ (where $w_i \in \{0, 1\}$) with a $P_error \to 0$
- Denote this maximal R to be the 'feedback channel capacity', C_{FB}
- It's obvious that if you just construct an encoder with zero feedback, you're able achieve at least C, ie $C_{FB} \geq C$
- Now the question remains: Is it possible to improve capacity with feedback? Short answer: No. Proving this shows the strength of Shannon's coding theorem.

Lemma 1 ($C_{FB} \leq C$)

- $H(W) = Rn$ - The entropy of W is fairly large
- $H(W|Y^n) \leq 1 + P_{error}Rn$. Fano's Inequality
- If $H(W|Y^n)$ wasn't small we wouldn't be able to calculate W, given Y
- These two points imply that Y^n contains a lot of information

$$\begin{aligned} I(W; Y^n) &= H(W) - H(W|Y^n) \\ &\geq R_n - 1 - P_error Rn \end{aligned}$$

- $P_{error}Rn$ is vanishingly small

Question: Is $I(W; Y^n) \leq nC$ **?**

$$I(W; Y^n) = H(Y^n) - H(Y^n|W) \tag{1}$$

$$\leq \sum_{i=1}^{n} H(Y_i) - H(Y^n|W) \tag{2}$$

$Y_1 \ldots Y_{i-1}, W$ is enough to fully determine X_i:

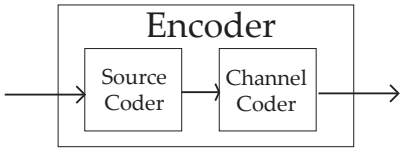
$$\begin{aligned} H(Y^n|W) &= \sum_{i=1}^{n} H(Y_i|Y_1 \ldots Y_{i-1}, W) \\ &= \sum_{i=1}^{n} H(Y_i|Y_1 \ldots Y_{i-1}, W, X_i) \\ &= \sum_{i=1}^{n} H(Y_i|X_i) \end{aligned}$$

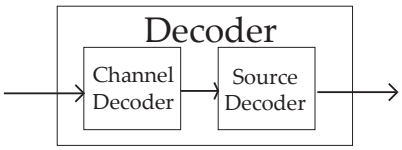
Then from (2)

$$\begin{aligned} I(W;Y^n) &\leq \sum_{i=1}^{n} H(Y_i) - \sum_{i=1}^{n} H(Y_i|X_i) \\ &= \sum_{i=1}^{n} I(Y_i;X_i) \\ &= nC \end{aligned}$$

Conclusion: Feedback doesn't contribute to capacity

- Previously we always looked at either
  - Uniform distributions on the source with a noisy channel
  - Clean channels with non-uniform sources
- We have now learned enough to combine non-uniform source with a noisy channel
- Simply need to look at the rate of the source, and the capacity of the channel. Then compare R and C
  - Apply compression algorithm to the source
  - Apply channel coding algorithm





Joint Source-Channel Coding Theorem

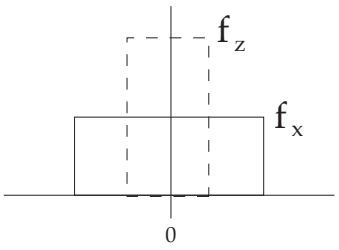
- If $W_1 \ldots W_k$ is produced by source $\mathscr{W}$ with entropy rate $H(\mathscr{W}) \rightarrow$ (source satisfies AEP)
- and if it's on a DMC with capacity C
  - Then communcation is possible with $P_{error} \rightarrow 0$ iff $H(\mathscr{W}) < C$
- After n steps of the source, it's producing on a uniform distribution of size $2^{H(\mathscr{W})n}$
- This concludes our discussion of the DMC

Continuous Channels

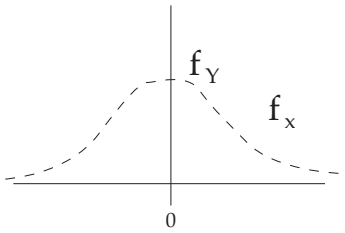
- We will begin by looking at a very simple channel
  - Input to channel X: $[-1, 1]$ (Real number)
  - Output of channel Y: Real number
  - Looking at the simplest case: noiseless channel (ie $X = Y$)
- Can't look at this in our typical manner because we can't define a finite alphabet to describe either input or output
- However, since $X = Y$, our channel capacity is apparently infinite
- What makes the channel capacity finite is the existance of noise
- **Adding noise Z to our model**
  - $Y = X + Z$
  - Z is uniform over $[-\epsilon, \epsilon]$ and independent of X
  - Divide input into intervals of 2ϵ – 'discretize it'
  - Then $C \geq \log(1/\epsilon)$
  - We will prove that the capacity is less than infinite in the future

Continuous Random Variables

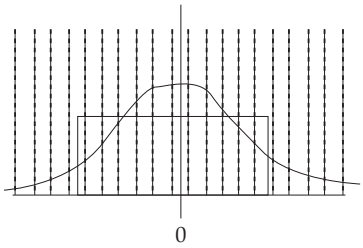
- X is a real-valued r.v.
- $f_X(x) \rightarrow$ Probability Density Function (PDF)
- $F_X(x) \rightarrow$ Cumulative Distribution Function (CDF)
  - $F_X(x) = \Pr[X \leq x] = \int_{-\infty}^{x} f_X(t)dt$
  - Monotonic, nondecreasing
- Given



- It is clear that just the pdf of the r.v. is not particularly revealing. However, comparing X and Z, one can certainly intuit that X is 'more random' than Z. How, then, do we quantitatively compute that?
- Because this is not as easy to interpret:



- X_ϵ: X discretized X by intervals of length ϵ, $Y_\epsilon \in \mathbb{Z}$



- $\lim_{\epsilon\to 0}\{H(X_\epsilon) - H(Y_\epsilon)\}$?
- Say we partition ϵ lots more: $\epsilon \to \frac{\epsilon}{2^l}$
- $\to H(\frac{X_\epsilon}{2^l}) \approx l + H(X_\epsilon)$
- and the same thing is happening to Y
- For $X+Y$, $\lim_{\epsilon\to 0}\{H(X_\epsilon)-H(Y_\epsilon)\}$ is well-behaved, but we want a quantity that only depends on X. For X alone what should we use as $H(X)$?

Differential Entropy

- $H(X) \triangleq \lim_{\epsilon\to 0}\{H(X_\epsilon) - f(\epsilon)\}$
- (We're going to be measuring against something like a baseline distribution)

$$f(\tfrac{\epsilon}{2^l}) = l + f(\epsilon) \to f(\epsilon) = \log(\tfrac{1}{\epsilon})$$
$$\text{then } H(X) = \lim_{\epsilon\to 0}\{H(X_\epsilon) + \log\epsilon\}$$

- Written in terms of the pdfs:

$$H(X) = -\int_{-\infty}^{\infty} f_X(x)\log[f_X(x)]dx$$

Examples

Example 1 - Entropy of the Uniform Distribution

$$X = \text{uniform}(a,b)$$
$$f_X(x) = \tfrac{1}{b-a} \text{ if } a \le x \le b, \text{ 0 otherwise}$$
$$H(X) = -\int_a^b \tfrac{1}{b-a}\log(\tfrac{1}{b-a})dx = \log(b-a)$$

- Not scale invariant.

Example 2 - Entropy of a Gaussian

$$X = N(0,\sigma^2)$$
$$f_X(x) = \tfrac{1}{\sqrt{2\pi\sigma^2}}\exp^{-\frac{x^2}{2\sigma^2}} \text{ if } a \le x \le b, \text{ 0 otherwise}$$
$$H(X) = -\int_{-\infty}^{\infty} f_X(x)\log f_X(x)dx$$
$$H(X) = \tfrac{1}{2}\log(2\pi\exp\sigma^2)$$

- Logarithmic in the variance

Future Lectures

- Try to understand how differential entropy behaves
- Look at AEP/LLN in this setting
- Continuous channels and how dif. entropy and mutual information play a role in determining capacity

6.441 Transmission of Information April 11, 2006

Lecture 15

Lecturer: Madhu Sudan *Scribe: Brandon Roy*

Today

- Differential entropy

  Conditional entropy, Joint entropy, Mututal information...

- Channel capacity

Admin

- PS3 due tomorrow
- Office hours, Thursday afternoon (send email)

Motivations from last time

Recall the "6.441 channel". We had input $X \in [-1, 1]$, noise $W \sim \text{Uniform}[-\epsilon, \epsilon]$, and output $Y = X + W$. We saw that

- If $\epsilon = 0$, channel has infinite capacity.
- If $\epsilon > 0$, channel has finite capacity.

Differential Entropy

Beginning with differential entropy, introduced last time, let us analyze this channel. We have X taking values in $\mathbb{R}$ with pdf $f = f_X$. Recall that we are working with X_ϵ, the ϵ-discretization of X. Then

$$h(X) \triangleq \lim_{\epsilon \to 0} \{H(X_\epsilon) + \log \epsilon\} = -\int_{-\infty}^{\infty} f_X(x) \log(f_X(x)) dx \quad \text{(if well behaved)}$$

Differential entropy is similar to "discrete" entropy but it is important not to draw too many conclusions from this similarity. For example, consider the following:

- $X \sim \text{Uniform}(a, b)$
- $h(X) = \log(b - a)$

- $h(aX) = h(X) + \log|a|$

Note that for some choices of a, goes to ∞, or if $b-a$ is very small, $\log(b-a) < 0$. So caution: $\exists X$ s.t. $h(X) < 0$ which is never true with $H(X)$ (when X is discrete)

Definitions

We now proceed to develop concepts for continuous random variables along the lines of those developed for discrete random variables. Consider a collection of random variables $X_1 \dots X_n$ (real-valued) with pdf $f(X_1, \dots, X_n)$.

Joint Entropy

$$h(X_1, \dots, X_n) = -\int_{X_1,\dots,X_n} f(x_1, \dots, x_n) \log f(x_1, \dots, x_n) dx_1 \dots dx_n$$

Conditional Entropy

Consider (X, Y) with joint distribution $f(X, Y)$, marginal distributions f_X,f_Y, and conditional distribution $f_{X|Y}(x|y)$. Then

$$\begin{aligned} h(X|Y) &= -\int_Y f_Y(y) \left[\int_X f_{X|Y}(x|y) \log f_{X|Y}(x|y) dx \right] dy \\ &= -\int \int_{X,Y} f(x, y) \log f_{X|Y}(x|y) dx dy \end{aligned}$$

Divergence

The divergence between pdf's f and g is

$$D(f||g) = \int_X f(x) \log \frac{f(x)}{g(x)} dx$$

Furthermore,

$$D(f||g) \geq 0 \qquad \text{(usual proof by Jensen's Inequality)}$$

Applying this,

$$(x, y) : D(f||f_X, f_Y) \geq 0 \implies h(X|Y) \leq h(X)$$

(Conditioning reduces entropy)
Note: when *comparing* entropies, any "$\log \epsilon$" terms show up on both sides and the comparison makes sense. Generally however, this is not true for the actual "values".

Mutual Information

$$I(X; Y) = h(X) - h(X|Y) \geq 0$$

If X and Y are "continuations" (opposite of discretizations) of discrete $\tilde{X}$, $\tilde{Y}$ then $I(X; Y) = I(\tilde{X}; \tilde{Y})$.

Chain Rule

$$h(X,Y) = h(X) + h(Y|X)$$

Maximum entropy distributions

Uniform distribution

Among random variables X taking values in $[0,1]$ the differential entropy is maximized by the $X \sim \text{Uniform}(0,1)$.

Proof 1

Let X be any r.v. taking values in $[0,1]$.
Let Y be any r.v. with distribution Uniform$(0,1)$, independent of X.
Let $Z = (X+Y) \mod 1$

Then
f_Z is Uniform$(0,1)$ (not hard to show)
$f_{Z|X}$ is Uniform$(0,1)$

$$\begin{aligned} h(Y,Z) &= h(X,Y) \\ &= h(X) + h(Y) \end{aligned}$$

$$h(Y,Z) \leq h(Y) + h(Z)$$

$$\implies h(X) \leq h(Z)$$

Proof 2 (Chung's proof)

$$\begin{aligned} h(X) &= E\left[\log \frac{1}{p(X)}\right] \\ &\leq \log\left[E\frac{1}{p(X)}\right] && \text{(Jensen's inequality)} \\ &= \log\left[\int_S p(x)\frac{1}{p(x)}dx\right] && (S \text{ is the support set}) \\ &= \log|X| \end{aligned}$$

which is the entropy of the uniform distribution.

So to conclude, among random variables taking values in $[0,1]$ the differential entropy is maximized by $X \sim \text{Uniform}(0,1)$.

Gaussian distribution

Furthermore, among (unbounded) random variables with mean 0 and variance 1, the differential entropy is maximized by $X \sim \text{Normal}(0,1)$. In other words, for any
X' distributed arbitrarily with mean 0 and variance 1
$X \sim \text{Normal}(0,1)$

$$D(X'||X) = h(X) - h(X') \geq 0$$

The Gaussian distribution has maximum entropy.

Entropy of the Gaussian distribution

Let $X \sim \text{Normal}(0, \sigma^2)$. Denote the pdf of X by $\Phi(X)$ Note that $\log \Phi(x) = a + bx^2$. Then

$$\begin{aligned} h(X) &= -\int \Phi(x) \log \Phi(x) dx \\ &= a \int \Phi(x) dx + b \int x^2 \Phi(x) dx \\ &= a + b\sigma^2 \end{aligned}$$

AEP Theorem

If $X_1, \ldots, X_n$ iid. X then

$$-\frac{1}{n} \log f(X_1, \ldots, X_n) \to h(X)$$

in probability

Typical set

$$A_\epsilon^{(n)} = \left\{ (x_1, \ldots, x_n) : \left| -\frac{1}{n} \log f(x_1, \ldots, x_n) - h(X) \right| \leq \epsilon \right\}$$

Also, define the "volume" of a set S as

$$Vol(S) = \int 1_S dx_1 \ldots dx_n$$

Then, $\forall \delta, \epsilon > 0$, $\exists n_0$ s.t. $\forall n \geq n_0$:

1. $\Pr(A_\epsilon^{(n)}) \geq 1 - \delta$
2. $Vol(A_\epsilon^{(n)}) \leq 2^{(h(X)+\epsilon)n}$
3. $Vol(A_\epsilon^{(n)}) \geq (1-\delta) 2^{(h(X)-\epsilon)n}$

Proofs

1:
$\Pr(A_\epsilon^{(n)}) \geq 1-\delta$. Follows from the LLN, applied to continuous random variables.

2:

$$\begin{aligned} 1 &= \int f(x_1, \ldots, x_n) dx_1, \ldots, dx_n \\ &\geq \int 1_{A_\epsilon^{(n)}} f(x_1, \ldots, x_n) dx_1, \ldots, dx_n \\ &\geq \int 1_{A_\epsilon^{(n)}} 2^{-(h(X)+\epsilon)n} dx_1, \ldots, dx_n \\ &= 2^{-(h(X)+\epsilon)n} \cdot Vol(A_\epsilon^{(n)}) \end{aligned}$$

$$\implies Vol(A_\epsilon^{(n)}) \leq 2^{(h(X)+\epsilon)n}$$

3:

$$\begin{aligned} 1-\delta &\leq \int 1_{A_\epsilon^{(n)}} f(x_1,\ldots,x_n)dx_1,\ldots,dx_n \\ &\leq \int 1_{A_\epsilon^{(n)}} 2^{-(h(X)-\epsilon)n} dx_1,\ldots,dx_n \end{aligned}$$

$$\implies Vol(A_\epsilon^{(n)}) \geq (1-\delta)2^{(h(X)-\epsilon)n}$$

Channel capacity

Now, back to the beginning. Recall our "6.441 channel": $Y = X + W$. Suppose $2\epsilon = \frac{1}{k}, k \in \mathbb{Z}$. We expected the "intuitive capacity" $\geq \log\lfloor 1 + \frac{2}{2\epsilon}\rfloor$.

Capacity

Define capacity as

$$C = \max_{f_X} \{I(X;Y)\}$$

Note that the maximization is over all distributions subject to constraints. But this is just a definition, let's see if it makes sense for our channel.

$$\begin{aligned} \max_{f_X} \{I(X;Y)\} &= \max_{f_X} \{h(Y) - h(Y|X)\} \\ &= \max_{f_X} \{h(Y) - h(X+W|X)\} \\ &= \max_{f_X} \{h(Y) - h(W|X)\} \\ &= \max_{f_X} \{h(Y) - h(W)\} \\ &\leq \log(2(1+\epsilon)) - \log(2\epsilon) \\ &= \log\left(\frac{1}{\epsilon} + 1\right) \end{aligned}$$

Wish to prove: operational capacity $\leq$ formal capacity. "Converse coding theorems" We want to find upper bound on R. The sequence of actions in transmission is

Choose $\underline{x} = (x_1,\ldots,x_n) \in$ set M of size 2^{nR}
Receiver gets $\underline{y} = (y_1,\ldots,y_n)$
We guess $\underline{\hat{x}} = (\hat{x_1},\ldots,\hat{x_n})$.

So we have the Markov chain $\underline{X} \to \underline{Y} \to \underline{\hat{X}}$ and use Fano's Inequality: $H(X|Y) \leq 1 + P_e \log|M|$

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &\geq H(X) - (1 + \log|M|P_e) \\ &\geq \log|M|(1-P_e) - 1 \\ &= nR(1-P_e) - 1 \end{aligned}$$

Note that above, we are using "discrete entropy" since X is "ϵ-discretized"

But we also have

$$\begin{aligned} I(X;Y) = h(Y) - h(Y|X) &\leq \sum_{i=1}^{n} h(y_i) - h(y_i|x_i) = \sum_{i=1}^{n} I(x_i;y_i) \\ &\leq nC \end{aligned}$$

and combining these two inequalities, we have

$$R \leq C$$

6.441 Transmission of Information Apr 13, 2006

Lecture 16

Lecturer: Madhu Sudan *Scribe: Zahi Karam*

Additive White Gaussian Noise (WGN) Channel

1 Mixing Discrete and Continuous r.v.s

Recall:
for a continuous r.v. $X \in \mathbb{R}$

$$\begin{aligned} X+a &\text{ - - -- >} h(X+a) = h(X) \\ aX &\text{ - - -- >} h(aX) = \log|a|.h(X) \end{aligned}$$

and for a continuous random vector $\underline{X} \in \mathbb{R}^n$ and A an nxm invertible matrix:

$$h(A\underline{X}) = h(\underline{X}) + \log|det(A)|$$

Then, assuming we have $\underline{Y} = \underline{X} + \underline{Z}$ and (using the above property) we get:

$$h(\underline{Y}|\underline{X}) = h(\underline{X} + \underline{Z}|\underline{X}) = h(\underline{Z}|\underline{X})$$

For continuous r.v.s. X and $Y \in \mathbb{R}$:

$$\begin{aligned} h(X) &= -\int f(X)\log f(X)dX \\ I(X;Y) &= h(X) - h(X|Y) \end{aligned}$$

But what if X is a discrete and Y is a continuous r.v.?
We can still say:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(X;Y) &= h(Y) - h(Y|X) \end{aligned}$$

However, does symetry hold, i.e. is

$$h(Y) - h(Y|X) = H(X) - H(X|Y)?$$

We can show this by using:

$$h(Y) - h(Y|X) = \lim_{\epsilon\to 0}\{(H(Y_\epsilon + \log\epsilon) - (H(Y_\epsilon|X) + \log\epsilon)\}$$

where $Y_\epsilon = \epsilon - discretization$ of Y, and using

$$H(X|Y) = \lim_{\epsilon \to 0}\{H(X|Y_\epsilon)\}$$

then $\forall \epsilon$:

$$\begin{aligned} H(X) - H(X|Y_\epsilon) &= H(Y_\epsilon) + \log\epsilon - (H(Y_\epsilon|X) + \log\epsilon) \\ \Rightarrow h(Y) - h(Y|X) &= H(X) - H(X|Y) \end{aligned}$$

2 Capacity Of Uniform $(-\epsilon, \epsilon)$ Error Channel (6.441 Channel) With $X \in (-1, 1)$

capacity of this channel is:

$$\log(1 + \lfloor \frac{1}{\epsilon} \rfloor) < Capacity \leq \log(1 + \frac{1}{\epsilon}) \tag{1}$$

This gap decreases as $\epsilon \to 0$, and is nonexistent if $\frac{1}{\epsilon}$ is an integer. The lower bound may be unrealistic, an example if $\epsilon = 1.5$. In this case the lower bound = 0 but we know we can acheive a capacity of at least 0.5 by modeling this channel as a binary eraser channel as shown in Figure 1.

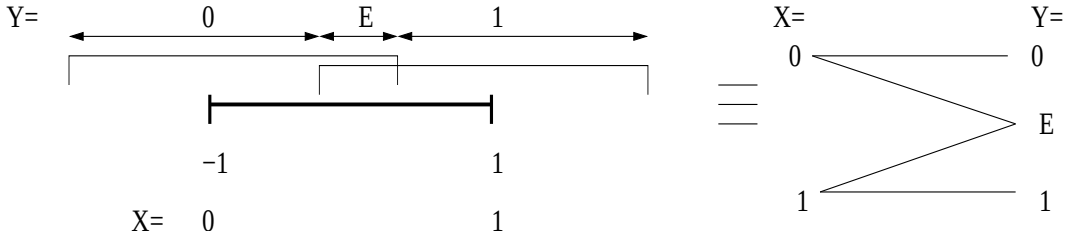


Figure 1: Model channel as a Discrete Binary Erasure Channel.

3 Capacity Of AWGN Channel

The channel is defined as: Input alphabet: $X \in \mathbb{R}$
Output alphabet: $Y \in \mathbb{R}$

$$\begin{aligned} Y &= X + Z \\ Z &\sim N(0, \sigma^2) \end{aligned}$$

We also need to have some constraint on X otherwise the capacity of the channel would be infinite.
We will constrain the channel by saying we can use the channel as many times as long

as we do not excede a certain power. This corresponds to:

$$\begin{aligned} var(X) &\leq P \\ E[X] &= 0 \end{aligned}$$

When thinking about variance constraint as Power constraint.
Therefore the channel is characterized by σ^2 (noise) and P (signal power). We expect that if the noise power was $2\sigma^2$ and the signal power was$2P$ then the capacity would not change. In other words the capacity of the channel is a function of the ratio $\frac{\sigma^2}{P}$.

So lets see if our assumption is valid:
Channel Capacity $= C = \max_{f_x}\{I(X;Y)\}$.
It is important to note that even though we have only proved this for the discrete case, it is essentially true for the continuous case as well. So we obtain the capacity as follows:

$$\begin{aligned} C &= \max\{h(Y) - h(Y|X)\} \\ &= \max\{h(Y) - h(X+Z|X)\} \\ &= \max\{h(Y) - h(Z)\} \\ &= \max\{h(Y)\} - h(Z) \end{aligned} \tag{2}$$

the last equalty holds because $h(Z)$ is independent of X.

We now take a moment to recall that for $W \sim N(0, \sigma^2)$

$$h(W) = \frac{1}{2}\log(2\pi e\sigma^2)$$

Also remember that if $Y = X + Z$ then

$$\begin{aligned} var(Y) &= var(X) + var(Z) \\ &\leq P + \sigma^2 \end{aligned}$$

Also recall in general that if

$$h(Y) \leq \max_{Y's.t.var(Y')\leq P+\sigma^2}(h(Y'))$$

And remember that the distribution with the largest entropy is $Y \sim N(0, \sigma^2)$. Which means that

$$h(Y) \leq \frac{1}{2}\log(2\pi e(P+\sigma^2))$$

So if we pick $X \sim N(0, P)$ and since $Z \sim N(0, \sigma^2)$ then $Y \sim N(0, P + \sigma^2)$ and therefore $max(h(Y))$ is acheived.

We now return to where we left off in our calculation of the capacity, in equation 2.

$$\begin{aligned} C &= \frac{1}{2}\log(2\pi e(P+\sigma^2) - \frac{1}{2}\log(2\pi e\sigma^2) \\ &= \frac{1}{2}\log(\frac{P+\sigma^2}{\sigma^2} \\ &= \frac{1}{2}\log(1 + \frac{P}{\sigma^2}) \end{aligned}$$

4 Outline Of An Encoding Scheme That Can Acheive Capacity

We will first outline a method that can actually achieve the computed capacity, allowing us to send $n\frac{1}{2}\log(1+\frac{P}{\sigma^2})$ bits of information via n uses of the channel
Consider the transmitted signal, $\underline{X} = X_1, X_2, ..., X_n$ where all the X_is are i.i.d. r.v. with $E[X_i] = 0$ and $var(X_i) = P$. The signal $\underline{X} \in \mathbb{R}^n$ has with high probability $\sum X_i^2 \approx nP$ which implies the signal lies in a ball with radius $\sqrt{np}$. Note that higher dimensional balls have the majority of their volume concentrated in the shell of the ball. Reffer to Figure 2 for an example of where $\underline{X}$ lies. Now, recall that the noise $\underline{Z} \in \mathbb{R}^n$ where $Z_1, Z_2, ..., Z_n$

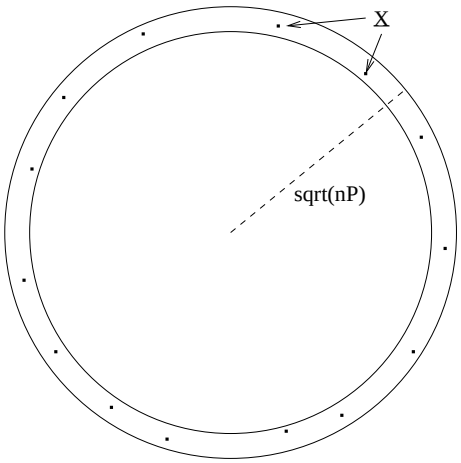


Figure 2: For i.i.d. X_i then the typical set of $\underline{X}$ is the shell of the ball.

are i.i.d. distributed $Z_i \sim N(0, \sigma^2)$, and therefore the $||\underline{Z}||_2 \sim \sqrt{n\sigma^2}$. Figure 3 is a sketch of how the noise affects the signal. Roughly we want all the balls to be disjoint so that we

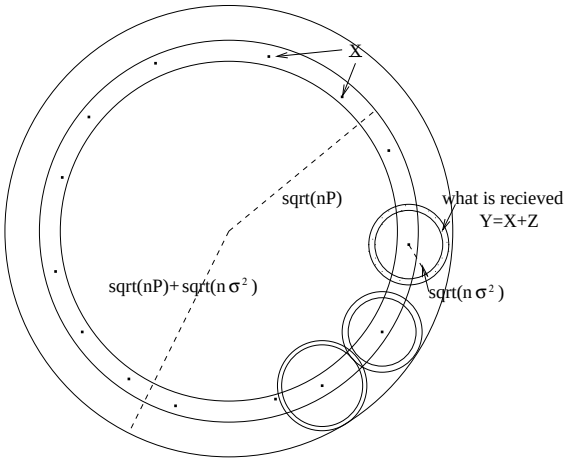


Figure 3: $X + Z$

can correctly decode the recieved message. Therefore the question becomes how many

disjoint balls small balls can we fit in the Y ball of radius $\sqrt{nP} + \sqrt{n\sigma^2}$.

$$\begin{aligned} NumSmallBalls &\leq \frac{Vol(BigBall)}{Vol(SmallBall)} \\ &= \frac{(\sqrt{nP} + \sqrt{n\sigma^2})^n}{(\sqrt{n\sigma^2})^n} \\ &\approx (\frac{\sqrt{nP}}{\sqrt{n\sigma^2}})^n \\ &= 2^{\frac{n}{2}\log\frac{P}{\sigma^2}} \end{aligned}$$

These calculations shows (roughly) how one could achieve the capacity we computed earlier. The next section shows a formal proof of this.

5 Formal Proof That The Computed Capacity Can Be Achieved

We wish to send the message $m \in \{1, ..., M\}$ where $M = 2^{nR}$
Encoding: $m \in \{1, ..., M\}$: $m \to X_1(m)X_2(m)...X_n(m)$ where the $X_i(m) \sim N(0, P - \epsilon)$ i.i.d.
Decoding: If $Y_1Y_2...Y_n$ is recieved then decode as follows:
if $\exists$ a unique m such that $\sum_{i=1}^n |X_i(m) - Y_i|^2 \leq n(\sigma^2 + \epsilon)$, choose nearest ball. Else declare an error.
Analyzing Probability of Error:

$$Pr[Encoding + DecodingError] \leq Pr[E_0] + Pr[E_1] + Pr[E_2]$$

E_0 = Event that $||\underline{X}(m)||^2 > nP$ which corresponds to too much power.
E_1 = Event that $||\underline{Z}||^2 > n(\sigma + \epsilon)^2$ which corresponds to too much error.
$E_2 = \bigcup_{m' \neq m} E_2(m') : \sum ||Y_i - X_i(m')||^2 \leq n(\sigma^2 + \epsilon)$
Errors E_0 and $E_1 \to 0$ as $n \to \infty$, however E_2 depends on the mutual information between X and Y. To analyize E_2 we first need to analyze $E_2(m')$ and to do that we need to first recall the following:
If we have (X, Y) and $\{(X_i, Y_i)\}_{i=1}^n$ are chosen i.i.d. from (X, Y) and $\{(\tilde{X}_i, \tilde{Y}_i)\}_{i=1}^n$ are chosen i.i.d. from (X, Y), then the typical set for $\{(X_i, Y_i)\}_{i=1}^n$ has size $2^{H(X,Y)n}$ and $Pr[\{(X_i, Y_i)\}_{i=1}^n$ is in the typical set$] \leq 2^{-I(X;Y)n}$.
Therefore:
$E_2(m') : Pr[\underline{X}(m')$ and Y are independent but jointly typical for the distribution $(X, Y)] \leq 2^{-I(X;Y)n}$.
Then using the union bound we obtain:

$$\bigcup_{m' \neq m} E_2(m') \leq 2^{Rn} 2^{-I(X;Y)n}$$

which proves the coding theorem.

6.441 Transmission of Information April 17, 2006

Lecture 16

Lecturer: Madhu Sudan *Scribe: Imad Jabbour*

1 Overview

In this lecture, we discuss the information-theoretic aspect of an Additive White Gaussian Noise (AWGN) channel. This channel is often used in communication theory to model many practical channels. We derive the capacity, and give an overview of the *Channel Coding Theorem* for AWGN channels. But first, we highlight some key facts from the previous lecture.

2 Review From Previous Lecture and Applications

In the previous lecture, we defined *differential entropy* $h(\cdot)$ and outlined some of its properties. We also defined *mutual information* for continuous random variables. In this section, we give a quick overview of the aforementioned material and illustrate some concepts with examples.

Definition 1 (Differential entropy of a continuous random variable) *The differential entropy* $h(X)$ *of a continuous random variable* X *with pdf* $f_X(x)$ *and support set* $\mathbb{R}$ *is defined as*

$$h(X) = -\int_{\mathbb{R}} f_X(x) \log f_X(x) dx \tag{1}$$

Definition 2 (Differential entropy of a continuous random vector) *The differential entropy* $h(\mathbf{X})$ *of a continuous random vector* $\mathbf{X}$ *with pdf* $f_{\mathbf{X}}(\mathbf{x})$ *and support set* $\mathbb{R}^n$ *is defined as*

$$h(\mathbf{X}) = -\int_{\mathbb{R}^n} f_{\mathbf{X}}(\mathbf{x}) \log f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \tag{2}$$

Theorem 1 (Differential entropy is invariant to translations) *The differential entropy of a continuous random variable* X *does not change if* X *is translated by a constant* c.

$$h(X + c) = h(X) \tag{3}$$

Theorem 2 (Scaling changes differential entropy) *The differential entropy of a continuous random variable* X *changes if* X *is scaled by a constant* a.

$$h(aX) = h(X) + \log |a| \tag{4}$$

Corollary:
More generally, for a continuous random vector $\mathbf{X}$ in $\mathbb{R}^n$, and for any invertible $n \times n$ matrix $\mathbf{A}$, we can write:

$$h(\mathbf{AX}) \quad = \quad h(\mathbf{X}) + \log |\mathbf{det}(\mathbf{A})|, \tag{5}$$

where $|\mathbf{det}(\mathbf{A})|$ denotes the absolute value of the determinant of $\mathbf{A}$.

Example (On the differential entropy of an additive-noise channel) Consider the channel: $Y = X + Z$, where the input X, the output Y and the additive noise Z are random variables distributed according to well-behaved pdf's. Furthermore, assume that X and Z are independent.
Now, let's consider the two-dimensional vector map

$$\begin{bmatrix} X \\ Z \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} X \\ X + Z \end{bmatrix}, \tag{6}$$

which translate into:

$$\mathbf{A}\begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} X \\ X+Z \end{bmatrix}, \tag{7}$$

and leads to $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Therefore, $|\mathbf{det}(\mathbf{A})| = 1$. By using Equation 5, we get:

$$h\left(\mathbf{A}\begin{bmatrix} X \\ Z \end{bmatrix}\right) = h\left(\begin{bmatrix} X \\ X+Z \end{bmatrix}\right) \tag{8}$$

$$= h\left(\begin{bmatrix} X \\ Z \end{bmatrix}\right) + \log(1) \tag{9}$$

$$= h\left(\begin{bmatrix} X \\ Z \end{bmatrix}\right), \tag{10}$$

which implies that $h\left(\begin{bmatrix} X \\ X+Z \end{bmatrix}\right) = h\left(\begin{bmatrix} X \\ Z \end{bmatrix}\right)$.
By the chain rule for entropy, we get: $h(X)+h(X+Z|X) = h(X)+h(Z|X)$, i.e. $h(X+Z|X) = h(Z|X)$. This means that

$$h(Y|X) = h(X+Z|X) = h(Z|X) = h(Z), \tag{11}$$

where the last equality follows from the fact that X and Z are independent. [1] This says that given X, the uncertainty remaining in Y is the same as the differential entropy of Z.

Definition 3 (Mutual information between continuous random variables) *The mutual information $I(X;Y)$ between two random variables X and Y, with joint density $f_{X,Y}(x,y)$, and marginal densities $f_X(x)$ and $f_Y(y)$ respectively, is defined as*

$$I(X;Y) = \int\int f_{X,Y}(x,y)\log\frac{f_{X,Y}(x,y)}{f_X(x)f_Y(y)}dxdy \tag{12}$$

From the definition, we can easily show that

$$I(X;Y) = h(X) - h(X|Y) = h(Y) - h(Y|X) \tag{13}$$

Theorem 3 (Relation of differential entropy to discrete entropy) *Consider a random variable X with a Riemann-integrable density $f_X(x)$. Suppose we divide the range of X into bins of length ϵ. Let $H(X_\epsilon)$ denote the entropy of the discretized version of X. Then*

$$h(X) = \lim_{\epsilon\to 0}[H(X_\epsilon) + \log\epsilon] \tag{14}$$

Example (On the mutual information between discrete and continuous r.v.'s) As a simple application of Eqs. 13 and 14, we would like to consider the case where X is a discrete-valued random variable, and Y is continuous-valued random variable with a Riemann integrable density $f_Y(y)$. Let Y_ϵ denote the ϵ-disretization of Y. The mutual information between X and Y can be written as

$$I(X;Y) = H(X) - H(X|Y) \tag{15}$$

But does the symmetry property of mutual information hold? That is, can we write $H(X) - H(X|Y) = h(Y) - h(Y|X)$? This turns out to be true, because of the following:

$$h(Y) - h(Y|X) = \lim_{\epsilon\to 0}[H(Y_\epsilon) + \log\epsilon] - \lim_{\epsilon\to 0}[H(Y_\epsilon|X) + \log\epsilon] \quad \text{By Eq. 14} \tag{16}$$

$$= \lim_{\epsilon\to 0}[H(Y_\epsilon) - H(Y_\epsilon|X)] \tag{17}$$

$$= I(X;Y_\epsilon) \tag{18}$$

$$= H(X) - \lim_{\epsilon\to 0} H(X|Y_\epsilon) \tag{19}$$

[1] This result can be generalized for the case of a input vector $\mathbf{X}^n$, noise vector $\mathbf{Z}^n$ and output vector $\mathbf{Y}^n$. In this case, $\mathbf{A}$ is a $2n \times 2n$ matrix whose determinant's absolute value is 1.

This says that the mutual information between X and Y is the limit of the mutual information between their quantized versions. Now, using the fact that $\lim_{\epsilon \to 0} H(X|Y_\epsilon) = H(X|Y)$, we get the desired result, i.e.

$$H(X) - H(X|Y) = h(Y) - h(Y|X) \tag{20}$$

Example (On the capacity of the *"6.441 Channel"*) Recall that the *"6.441 Channel"* is an additive-noise channel that has an input X distributed on the [-1, 1] interval, and a uniformly distributed noise on the (-ϵ, +ϵ) interval. It follows that the output Y is distributed between (-1 - ϵ) and (+1 + ϵ), and that the capacity C of this channel can be bounded as follows

$$\log\left(1 + \left\lfloor \frac{1}{\epsilon} \right\rfloor\right) \leq C \leq \log\left(1 + \frac{1}{\epsilon}\right) \text{ bits}, \tag{21}$$

with the upper and lower bound being equal if $\frac{1}{\epsilon}$ is an integer. However, the lower bound may be loose if, for instance, $\epsilon = 1.5$, which leads to $C \geq 0$. Yet, we know that we can achieve a capacity of at least $\frac{1}{2}$ bit if we represent the *"6.441 Channel"* as a binary erasure channel. Indeed, and as shown in Figure 1, the *"6.441 Channel"* can be thought of as a BER channel if we map a subset $\mathcal{S}_1$ from the support set of X to the input value 0 of the BER channel, and the complement of $\mathcal{S}_1$ to the input value 1 of the BER channel.

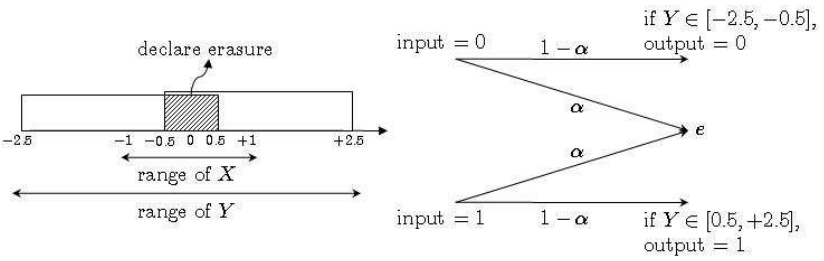


Figure 1: *"6.441 Channel"* and BER Channel

Under the above scenario, the erasure probability α is guaranteed to be at most 0.5, which means that the capacity of the *"6.441 Channel"* is at least 0.5 bit.

3 Capacity of the AWGN Channel

In this section, we derive the capacity of the AWGN channel. But before doing that, let's start by stating some facts about the AWGN channel.

3.1 What is an AWGN Channel?

An AWGN channel (see Figure 2) is a continuous-alphabet, time-discrete memoryless channel where, at each time unit, the output Y can be written as the sum of the input X and the noise Z

$$Y = X + Z \qquad Z \sim \mathcal{N}(0, \sigma^2) \tag{22}$$

The additive noise Z is assumed to be independent of the channel input X, and is represented a zero-mean Gaussian random variable with variance σ^2, and with density

$$f_Z(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{z^2}{2}} \tag{23}$$

A zero-mean Gaussian random variable is extensively used in the literature to model noise, since it serves as a good approximation to the cumulative effect of a large number of small random sources of noise (by the Central Limit Theorem). The term *white* is used to indicate that the noise's spectral density is flat over the frequency band of interest. In the time-domain, this says that the covariance function looks like a short duration pulse around $t = 0$. Roughly speaking, this means that the noise samples are mutually independent.

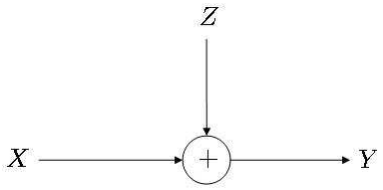


Figure 2: The AWGN Channel

3.2 Power Constraint

As we have previously mentioned, the noise Z is assumed to be independent of the signal X. But without further conditions, the capacity of this channel may be infinite, and this can happen if the noise variance is zero or if the input is unconstrained. This suggests that we need some sort of constraint on the channel input, and a good choice is the power constraint. As a result, an AWGN channel is usually specified by an upper-bound P on the signal

$$\mathbb{E}[X^2] \leq P, \tag{24}$$

which is equivalent to the constraint

$$Var(X) \leq P \quad \text{and} \quad \mathbb{E}[X] = 0 \tag{25}$$

Under these conditions, an AWGN channel is specified by a set of two parameters $\{\sigma^2, P\}$. Intuitively, if we double both the noise variance and the signal power, the capacity should remain unchanged; this suggests that the capacity of the channel should be a function of the ratio $\frac{P}{\sigma^2}$, an idea which we are going to formalize in what follows.

3.3 Information Capacity of an AWGN Channel

In this subsection, we define the information capacity of the AWGN channel as the maximum of the mutual information between the input and the output over all distributions of the input that satisfy the power constraint defined above.

Definition 4 (Information capacity of an AWGN channel) *The information capacity of the AWGN channel with power constraint P is defined as*

$$C = \max_{f(x):\mathbb{E}[X^2]\leq P} I(X;Y) \tag{26}$$

By expanding the mutual information, we get

$$\begin{aligned} I(X;Y) &= h(Y) - h(Y|X) && (27)\\ &= h(Y) - h(X+Z|X) && (28)\\ &= h(Y) - h(Z), && (29) \end{aligned}$$

where Eq. 29 follows from the result in Eq. 11. Recall that if $Z \sim \mathcal{N}(0, \sigma^2)$, then its differential entropy is: $h(Z) = \frac{1}{2}\log(2\pi e\sigma^2)$. We also remark that since X and Z are independent, and using the fact that $Var(X) \leq P$, then

$$\begin{aligned} Var(Y) &= Var(X) + Var(Z) && (30)\\ &\leq P + \sigma^2 && (31) \end{aligned}$$

Moreover, we use the fact that the Gaussian distribution maximizes the entropy for a given variance. Applying this fact to the received signal Y, whose variance is upper-bounded by $P + \sigma^2$, we get that

$h(Y) \leq \frac{1}{2}\log[2\pi e(P+\sigma^2)]$. This says that the input which maximizes this entropy is $X \sim \mathcal{N}(0,P)$. We are now ready to upper-bound the mutual information

$$\begin{aligned} I(X;Y) &= h(Y) - h(Z) && (32)\\ &\leq \frac{1}{2}\log[2\pi e(P+\sigma^2)] + \frac{1}{2}\log(2\pi e\sigma^2) && (33)\\ &= \frac{1}{2}\log\left(1+\frac{P}{\sigma^2}\right) && (34)\end{aligned}$$

By using Eq. 26, we finally get that the information capacity of the AWGN channel is

$$C = \max_{f(x):\mathbb{E}[X^2]\leq P} I(X;Y) = \frac{1}{2}\log\left(1+\frac{P}{\sigma^2}\right), \quad (35)$$

and this maximum is achieved when $X \sim \mathcal{N}(0,P)$, i.e. $f(x) = \frac{1}{\sqrt{2\pi P}}e^{-\frac{x^2}{2}}$. In communication theory, the ratio $\frac{P}{\sigma^2}$ is often called signal-to-noise ratio (SNR). In the next subsection, we show that the capacity that we just computed is also the supremum of all achievable rates of the channel, i.e. we give the above equation its *operational* meaning.

3.4 *Operational* Meaning of the Capacity of the AWGN Channel

In this subsection, we first show the capacity in Eq. 35 can be achieved using an argument based on the Weak Law of Large Numbers (WLLN) and the sphere-packing method. Then we formalize our proof and show that indeed, Eq. 35 is also the supremum of the achievable rates.

3.4.1 Sphere Packing method

The idea that will be raised in this paragraph is rather a plausibility argument rather than a formal proof. It emanates from the following question: *Given Eq. 35, and for n uses of the channel, will we be able to send* $\frac{n}{2}\log\left(1+\frac{P}{\sigma^2}\right)$ *bits with low probability of error?* The answer turns out to be yes, as it is outlined next.

Suppose that the sequence $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ is transmitted over n symbol durations, where the X_i's are i.i.d. $\sim \mathcal{N}(0,P)$. Using the WLLN, we can show that, with high probability, $||\mathbf{X}||^2 = \sum_{i=1}^n X_i^2 \approx nP$. This implies that, with high probability, the transmitted codeword $\mathbf{x}$ lies within an n-dimensional sphere of radius $\approx \sqrt{nP}$. Note that high-dimensional spheres have almost all their volume concentrated in their shell, which means that the typical set of $\mathbf{X}$ lies in the shell of the sphere of radius $\approx \sqrt{nP}$ (see Figure 3).

Furthermore, recall that the noise $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_n)$ has i.i.d. components Z_i that are drawn according to a zero-mean Gaussian distribution with variance σ^2, i.e. $Z_i \sim \mathcal{N}(0,\sigma^2)$. The WLLN asserts that, with high probability, $||\mathbf{Z}||^2 = \sum_{i=1}^n Z_i^2 \approx n\sigma^2$. This says that, given a specific codeword $\mathbf{x}$, the received signal lies on the shell of a sphere of radius $\sqrt{n\sigma^2}$, and centered at $\mathbf{x}$ (see Figure 3). Therefore, the received sequences $\mathbf{y}$ lie within a sphere of radius $\approx \sqrt{nP} + \sqrt{n\sigma^2}$.

When we encode our sequences, we want the "noise" spheres (i.e. the spheres centered around the codewords $\mathbf{x}$, and whose radius is approximately equal to $\sqrt{n\sigma^2}$) to be more or less disjoint, so that we can decode with low probability of error. The *sphere packing* or *Kepler conjecture* problem answers the following question: *How many such balls can we pack such that all of them are disjoint?* Roughly, the answer turns out to be

$$\text{Number of balls} \leq \frac{\text{Volume of big ball}}{\text{Volume of small ball}} \quad (36)$$

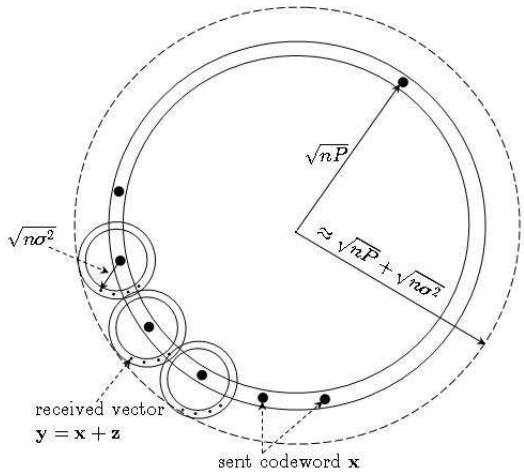


Figure 3: Sphere packing for the AWGN channel

$$= \frac{(\sqrt{nP} + \sqrt{n\sigma^2})^n}{(\sqrt{n\sigma^2})^n} \tag{37}$$

$$\approx \left(\frac{\sqrt{nP}}{\sqrt{n\sigma^2}}\right)^n \tag{38}$$

$$= 2^{\frac{n}{2}\log\frac{P}{\sigma^2}}, \tag{39}$$

where Eq. 38 uses the fact that the ratio $\frac{\sigma^2}{P}$ is assumed to be very small. In a nutshell, this very rough plausibility argument shows that the rate of the code is approximately $\frac{1}{2}\log\left(\frac{P}{\sigma^2}\right)$. Moreover, it indicates that we cannot hope to transmit data at rates greater than C with low probability of error. Yet, a more formal and cleaner proof of the operational meaning of capacity is derived in what follows.

3.4.2 Channel Coding Theorem for AWGN Channels

In this paragraph, we will formally prove that the capacity of an AWGN channel with power constraint P and noise variance σ^2 is the same as the information capacity defined in Eq. 35. But first, let's start by stating some definitions.

Definition 5 ((M,n) code for an AWGN channel) *A (M, n) code for the AWGN channel with power constraint P consists of the following:*

- *An message space $\{1, 2, \ldots, M\}$, where $M = 2^{nR}$, and R is the rate of the (M, n) code (in bits per transmission).* [2]
- *An encoding function $x : \{1, 2, \ldots, M\} \to \mathcal{X}^n$, yielding codewords $\mathbf{x}(m) = (x_1(m), x_2(m), \ldots, x_n(m))$. The codewords have i.i.d. components that satisfy the power constraint $X_i(m) \sim \mathcal{N}(0, P - \epsilon)$.*
- *A decoding function $g : \mathcal{Y}^n \to \{1, 2, \ldots, M\}$ that operates on the received sequence $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ as follows: if $\exists$ a unique m such that*

$$\sum_{i=1}^{n} |x_i(m) - y_i|^2 \leq n(\sigma^2 + \epsilon), \tag{40}$$

then output message m (i.e. choose nearest ball). Otherwise, declare an error.

[2] More precisely, $M = \lceil 2^{nR} \rceil$. However, we drop the ceiling function to simplify the notation.

Definition 6 (Achievable rate) [3] *A rate R is said to be achievable for an AWGN channel with power constraint P if there exists a sequence of $(2^{nR}, n)$ codes with codewords satisfying the power constraint, such that the maximal probability of error $\lambda^{(n)}$ tends to zero. The capacity of the channel is the supremum of the achievable rates*

Theorem 4 (Capacity of an AWGN Channel) *The capacity of an AWGN channel with power constraint P and noise variance σ^2 is*

$$C = \frac{1}{2}\log\left(1 + \frac{P}{\sigma^2}\right) \quad \textit{bits per tranmission} \tag{41}$$

Proof [Achievability] We begin by analyzing the probability of error. First, let's define the following events (assuming codeword m was transmitted):

$$\begin{aligned} E_0 &= ||\mathbf{X}(m)||^2 > nP \quad \text{(i.e. too much signal power)} && (42)\\ E_1 &= ||\mathbf{Z}||^2 > n(\sigma^2 + \epsilon) \quad \text{(i.e. too much noise variance)} && (43)\\ E_2 &= \bigcup_{m' \neq m} E_2(m') : \sum_i ||\mathbf{y}_i - \mathbf{x}_i(m')||^2 \leq n(\sigma^2 + \epsilon) && (44) \end{aligned}$$

By denoting $\mathbf{P}_e$ as the probability of error, we get

$$\mathbf{P}_e = \mathbf{P}(\text{encoding + decoding error}) = \mathbf{P}(E_0) + \mathbf{P}(E_1) + \mathbf{P}(E_2) \tag{45}$$

By the WLLN, $\mathbf{P}(E_0) \to 0$, and $\mathbf{P}(E_1) \to 0$, as $n \to \infty$. Hence, what remains is to analyze the probability of event E_2; more specifically, we want to analyze the probability of $E_2(m')$. We can define the probability that event $E_2(m')$ occurs as follows:

$\mathbf{P}(E_2(m')) = \mathbf{P}[\mathbf{X}(m')$ and $\mathbf{Y}$ are independent but jointly typical for the distribution of (X,Y)].

Now consider $\{(X_i, Y_i)\}_{i=1}^n$ and $\{(\widetilde{X}_i, \widetilde{Y}_i)\}_{i=1}^n$ each to be i.i.d. and drawn $\sim (X, Y)$. Then, it follows that the typical set for $\{(X_i, Y_i)\}_{i=1}^n$ has size $\approx 2^{nH(X,Y)}$, and that the probability that $\{(\widetilde{X}_i, \widetilde{Y}_i)\}_{i=1}^n$ is in the typical set is $\leq 2^{-nI(X;Y)}$. Therefore, $\mathbf{P}(E_2(m')) \leq 2^{-nI(X;Y)}$. By using the union bound, we get that

$$\begin{aligned} \mathbf{P}_e &\approx \mathbf{P}(E_2) && (46)\\ &= \mathbf{P}\left(\bigcup_{m' \neq m} E_2(m')\right) && (47)\\ &\leq 2^{nR} 2^{-nI(X;Y)} && (48)\\ &= 2^{-n(I(X;Y)-R)} && (49) \end{aligned}$$

For n sufficiently large, and $R < I(X;Y)$, the probability of error goes to zero, which proves the existence of a good $(2^{nR}, n)$ code. Therefore, the forward part of the theorem is proved. We will prove the converse part in the next lecture. ■

[3] Cf. Cover and Thomas, p. 242.

6.441 Transmission of Information Apr 20, 2006

Lecture 17

Lecturer: Madhu Sudan *Scribe: Matt Willis and Matt Willsey*

Gaussian Channels (continued)

1 Overview

In this lecture, we will continue our discussion of the All White Gaussian Noise Channel (AWGN). In particular, we review the Coding Theorem for the AWGN as discussed last lecture, and prove the converse to the Coding Theorem for the AWGN. Towards the end of our discussion, we will address Parallel Gaussian Channels, which is the case when there are n communication channels, each with independent (and possibly different) noise characteristics. We briefly comment on the generalization of this analysis to the Colored Gaussian Noise Model, where the noise properties of different channels may be linked, and are no longer independent.

2 Review From Previous Lecture

A Gaussian channel, with an input alphabet, $X \in \mathbb{R}^n$, output alphabet, $Y \in \mathbb{R}^n$, and subject to the power constraint is defined as follows:

$$\begin{aligned} Y &= X + Z \\ Z &\sim N(0, \sigma^2). \end{aligned}$$

As discussed in the previous lecture,we impose the following power constraints to maintain a finite capacity:

$$\begin{aligned} var(X) &\leq P \\ E[X] &= 0. \end{aligned}$$

Also from last lecture, we calculated the channel capacity, C, for a given input distribution $p(x)$ to be:

$$\begin{aligned} C &= \max_{p(x)} \{I(X;Y)\} \\ &= \frac{1}{2}\log(2\pi e(P+\sigma^2)) \; bits \; per \; transmission \end{aligned}$$

The mutual information is maximized when $X \sim N(0, P)$.

3 Coding Theorem

In this section we will prove both the coding theorem and the converse coding theorem.

3.1 Proof of the Coding Theorem

We will begin by defining an encoding function, E, that has messages in a set of size 2^{Rn} and maps it to n real numbers (since we used the channel n times) as shown below:

$$E\{1, 2, ..., 2^{nR}\} \rightarrow \mathbb{R}^n$$

We pick E such that it is chosen at random. In other words we will ensure that every symbol that we transmit achieves the following distribution:

$$(E(m))_i \quad \sim \quad N(0, P - \epsilon)$$

where m is the message and $(E(m))_i$ is i.i.d. over (m,i). Note that the variance of $(E(m))_i$ is $P - \epsilon$ so that we do not exceed the total power of nP in n transmissions. Next we will establish the following notation:

$\underline{X}$ denotes the transmitted sequence $\underline{X} = E(m)$
$\underline{Y}$ denotes the received sequence $\underline{Y} = \underline{X} + \underline{Z}$

We will now try to prove that if R is less than C, then the probability of error is very small.

Goal: if $R < I(x; y)$ then Pr(error) is small

Now there are three sources of decoding error when transmitting m. Generally speaking, the sources of error can depend on the encoding of m, the encoding of some other message m' (where $m' \neq m$), or the error introduced by the channel which is a random variable. The three sources of error correspond to the events below.

First let E_0 be when the power of a realized encoding, E(m), is too large:

$$||E(m)||_2^2 \geq nP.$$

Note that that the 2 in the subscript above indicates an $l^2 - norm$. Remembering that $Exp[(E(m))_i^2] = (P - \epsilon)$, the law of large numbers tells us that the likelihood that we exceed nP in the above equation is very small. Thus,

$$Pr[E_0] \rightarrow 0.$$

It is important to remember that this error is simply a violation of the power constraint.

Now let E_1 be when the noise causes Z to be too large.

$$||Z||_2^2 \geq n(\sigma^2 + \epsilon)$$

Once again the law of large numbers tells us that

$$Pr[E_1] \rightarrow 0.$$

since Z will converge to its mean.

Thus, the previous two errors, E_0 and E_1, simply discusses the likelihood that random

variables differ significantly from their expectation.

Now let $E_2(m')$ be defined as the event when probability of the encoding of some message, m', is too close to the encoding of the true message, m. In other words

$$||\underline{Y} - E(m')||_2^2 \leq n(\sigma^2 + \epsilon)$$

We claim that the probability of this event is

$$Pr[E_2(m')] \leq 2^{-I(X;Y)n},$$

which will be proved to follow from joint AEP.

Proof:

Let us consider two random variables, $(\underline{X}, \underline{Y})$, picked jointly according to our channel model (where $\underline{X} = E(m)$). Now let us also consider two additional random variables, $(\widetilde{\underline{X}}, \widetilde{\underline{Y}})$, picked jointly according to the channel model (where $\widetilde{\underline{X}} = E(m')$) and independent of the first two random variables. We now consider the following two subclaims:

1. $Pr[\widetilde{\underline{X}}, \underline{Y}$ *are jointly typical*$] \leq 2^{-I(X;Y)n}$
2. $E_2(m')$ *occurs if* $\widetilde{\underline{X}} = E(m')$ *and* $\underline{Y} = Channel(E[m])$ *are jointly typical.*

Generally speaking, subclaim 2 states that E_2 occurs when the encoding of E(m') is too close to E(m). Subclaim 1 then gives us $Pr[E_2]$.

Now let us draw a picture of this situation, which can be seen in Figure 1. For large

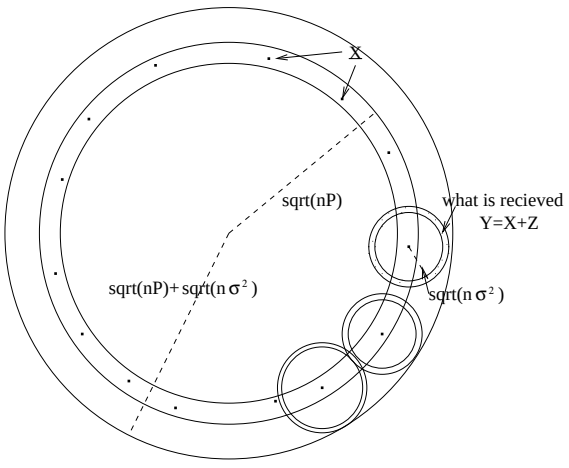


Figure 1: *Graphical Illustration Demonstrating* E_2

n, $\underline{X}$ will be located at a radius of $\sqrt{nP}$. A particular $\underline{Y}$ associated with a particular $\underline{X}$ will be located within a ball of radius $\sqrt{n\sigma^2}$ of $\underline{X}$ as shown in the figure. However, for large n, most of the volume for realizable values of $\underline{Y}$ will be located around a radius of

$\sqrt{nP + n\sigma^2}$, which is the outermost ring in the figure. E_2 occurs when $\underline{\widetilde{X}}$ falls within a small ball centered around $\underline{X}$ (which is the event that $\underline{\widetilde{X}}$ and $\underline{Y}$ are jointly typical).

Thus, if we pick an $\underline{\widetilde{X}}$ independent of $\underline{X}$, the probability of that $\underline{\widetilde{X}}$ and $\underline{Y}$ are jointly typical is roughly the volume of a small ball over the total volume of the biggest ball.

$$\begin{aligned} Pr[E_2(m')] &= \sqrt{n\sigma^2}^n / \sqrt{n(P+\sigma^2)}^n \\ &= [\sigma^2/(P+\sigma^2)]^{n/2} \\ &= 2^{-I(X;Y)n} \end{aligned}$$

The last equality holds since we have already shown that $I(X;Y) = log(1 + P/\sigma^2)^{1/2}$.

The above derivation is somewhat ad-hoc; however, now we shall formally prove subclaim 1. To determine the probability that $\underline{\widetilde{X}}$ and $\underline{Y}$ are typical, we integrate the joint probability over the jointly typical set. Since $\underline{\widetilde{X}}$ and $\underline{Y}$ are independent, we are able to write the joint probability as the product of the marginal probabilities of $\underline{\widetilde{X}}$ and $\underline{Y}$.

$$\begin{aligned} Pr[(\underline{\widetilde{X}}, \underline{Y}) \ joint \ typ.] &= \int_{joint \ typ. \ set} P_{\underline{\widetilde{X}}}(\underline{\widetilde{X}}) P_{\underline{Y}}(\underline{Y}) d_{\underline{\widetilde{X}}} d_{\underline{Y}} \\ &\leq Vol(joint \ typ. \ set) max_{\underline{\widetilde{X}}}[P_{\underline{\widetilde{X}}}(\underline{\widetilde{X}})] max_{\underline{Y}}[P_{\underline{Y}}(\underline{Y})] \\ &\leq 2^{h(x;y)n} \cdot 2^{-h(x)n} \cdot 2^{-h(y)n} \\ &= 2^{-I(X;Y)n} \end{aligned}$$

Note the second inequality hold since $\underline{\widetilde{X}}$ and $\underline{Y}$ are both contained in the jointly typical set ($Vol(joint \ typ. \ set) \approx 2^{h(x;y)n}$, $max_{\underline{\widetilde{X}}}[P_{\underline{\widetilde{X}}}(\underline{\widetilde{X}})] \approx 2^{-h(x)n}$, $and \ max_{\underline{Y}}[P_{\underline{Y}}(\underline{Y})] \approx 2^{-h(y)n}$). The above derivation formally proves subclaim 1, which is:

$$Pr[E_2(m')] = 2^{-I(X;Y)n}.$$

Since there are 2^{Rn} messages,

$$Pr[\exists \ m' \ s.t. \ E_2(m') \ occurs] = 2^{Rn} \cdot 2^{-I(X;Y)n}.$$

Therefore if $R < I(X;Y)$, then

$$Pr[\exists \ m' \ s.t. \ E_2(m') \ occurs] \to 0,$$

which proves the coding theorem.

3.2 Converse to the Coding Theorem

The goal of this section is to demonstrate that the probability of error approaching zero implies that the channel rate R is below capacity, i.e.:

$$p_{err} \to 0 \Longrightarrow R \leq C.$$

By assumption, for a given rate R we have an input alphabet containing messages M where

$$M \in \{1, 2, ..., 2^{Rn}\}$$

as well as an encoding function E:

$$E : M \to X^n.$$

Our channel is described mathematically as

$$Y^n = X^n + Z^n$$

We begin the proof by noting that M, X^n, and Y^n form a Markov chain ($M \to X^n \to Y^n$), which allows us to apply Fano's Inequality:

$$H(M|Y^n) \leq 1 + nRp_{err} = O(n),$$

where $O(n) \to 0$ as $p_{err} \to 0$ (this can also be seen by using the full-fledged Fano's Inequality, $H(p_{err}) + p_{err} \log(|X^n| - 1) \geq H(M|Y^n)$). Now consider the quantity

$$I(M; Y^n) = H(M) - H(M|Y^n) = nR - O(n)$$

where $H(M) = nR$ for a uniform input distribution of messages. Due to the Markov Chain ($M \to X^n \to Y^n$), the Data Processing Inequality yields:

$$I(X^n; Y^n) \geq I(M; Y^n) = nR - o(n)$$

We make use of the fact that $I(X^n; Y^n) \leq \sum_i I(X_i; Y_i)$ which can be seen from the following steps (see Cover and Thomas for details):

$$\begin{aligned} I(X^n; Y^n) &= h(Y^n) - h(Y^n|X^n) \\ &= h(Y^n) - h(Z^n) \\ &\leq \sum_i^n h(Y_i) - h(Z^n) \\ &= \sum_i^n h(Y_i) - \sum_i^n h(Z_i) \\ &= \sum_i^n I(X_i; Y_i). \end{aligned}$$

This substitution yields:

$$\sum_i^n I(X_i; Y_i) \geq nR - O(n)$$

Let us say that the i^{th} transmission contains power P_i, where by the power constraint, $\sum_i P_i \leq nP$. As we demonstrated last lecture, we can maximize each $I(X_i; Y_i)$ to be $\frac{1}{2}\log(1 + \frac{P_i}{\sigma^2})$ by choosing the normal distribution as input distribution. Therefore,

$$\sum_i^n I(X_i; Y_i) \geq nR - O(n)$$

$$\sum_i^n \frac{1}{2}\log(1 + \frac{P_i}{\sigma^2}) \geq nR - O(n).$$

Due to symmetry, the left hand side of the equation is maximized when each P_i of equal value. Therefore,

$$\sum_i^n \frac{1}{2}\log(1+\frac{P}{\sigma^2}) \geq nR - O(n)$$
$$nC \geq nR - O(n)$$
$$C \geq R - O(n)/n$$

where as explained above $O(n) \to 0$ as $p_{err} \to 0$, and this proves the converse.

4 Parallel Gaussian Channels

In previous sections, we discussed the use of one AWGN channel with noise characterized by $Z = N(0, \sigma^2)$. Now we consider the case of Parallel Gaussian Channels, where the user has n such channels at his disposal. Each channel is allowed to have its own noise characteristic ($Z_i = N(0, \sigma_i^2)$), which is independent from other channels. We still impose a power constraint, but now it states that the power used over all n channels must be limited. This is a fairly realistic model that might be used to describe a radio broadcasting station, where each channel represents a different broadcast frequency, and each frequency experiences a different atmospheric dispersion. In fact, we probably already have some intuition concerning how to use a parallel channel. By way of building up an intuition on how to use such a n channel system, consider the following two examples:

- **Example 1**
  In this example, we have n identical channels, with $\sigma_1^2 = \sigma_2^2 = ... = \sigma_n^2$. It is obvious that we would want to distribute the power equally to each channel, so that $P_i = \frac{P}{n}$.
- **Example 1**
  In this example, $\sigma_1^2 = \sigma_2^2 = ... = \sigma_k^2 = 1$, and $\sigma_{k+1}^2 = ... = \sigma_n^2 = \infty$. There is no reason to put any energy into an infinitely noisy channel, as we have no way of interpolating the input given the noisy output. Instead, we distribute the power evenly among the first k channels.

The intuition behind these two examples suggests that the effective way to use a Gaussian parallel channel is to weight the power distribution more heavily among the channels with better noise characteristics.

Now, for a more formal discussion: the i^{th} channel (where $i \in \{1, 2, ..., n\}$) of a Parallel Gaussian Channel is characterized as follows:

$$Y_i = X_i + Z_i$$
$$Z = N(0, \sigma_i^2)$$

For each channel with power P_i, the power constraint is:

$$\sum_i^n P_i \leq P$$

Or, in terms of channel values:

$$\text{Exp}[\sum_i^n X_i^2] \leq P$$

The quantity of interest, capacity C, is defined as:

$$C = \max_{p(x_1,...,x_n:\sum_i P_i \leq P} I(X_1, ..., X_n; Y_1, ..., Y_n)$$

Following the analysis of last section, we recognize that each channel is maximized (achieves channel capacity) with an input Gaussian distribution subject to the particular channel's power constraint, and therefore:

$$C \leq \sum_i^n \frac{1}{2} \log(1 + \frac{P_i}{\sigma_i^2})$$

The analysis leading up to this equation follows the same reasoning as the one channel case, but now we no longer fix σ, but instead we allow each σ_i to vary independently of the others. The task is to maximize the right side of the equation above subject to the power constraint $\sum_i^n P_i \leq P$,, or equivalently, to maximize the following expression:

$$C \leq \sum_i^n \frac{1}{2} \log(\frac{Q_i}{\sigma_i^2})$$
$$Q_i = P_i + \sigma_i^2,$$

subject to the constraint, $\sum_i^n Q_i \leq P + \sum_i^n \sigma_i^2$. In either case, this is an optimization problem subject to a power constraint; as pointed out in the text, it can be solved using the technique of Lagrange multipliers. First, we form the appropriate Lagrange multiplier expression:

$$J(P_1, P_2, ..., P_n) = \sum \frac{1}{2} \log(1 + \frac{P_i}{\sigma_i^2}) + \lambda(\sum_i P_i)$$

Then differentiate with respect to P_i:

$$\frac{1}{2} \frac{1}{P_i + \sigma_i^2} + \lambda = 0$$
$$\Rightarrow P_i = \nu - \sigma_i^2$$

Where ν can be solved for by substituting the solved P_i's into the power constraint. (It should be noted that for physical reasons, $P_i \geq 0$, and therefore you must bound each P_i below by zero).

The preceding discussion explains the mathematics behind it, but there is a more intuitive approach to understanding the optimization process, through the process of "water-filling." In this analogy, there is a finite amount of "water" (i.e., a limitation on the power constraint) that can be poured into these n channels. It is desirable to put more water into channels that are useful and have low noise characteristics, and less water

into noisier channels that have a lower capacity. So, how do we go about distributing the water? Refer to the Figure 10.4 in Cover and Thomas; the water will seek its own level, and naturally pool more deeply into the lower noise channels. A nice feature of the "water-filling" analogy is that it automatically takes into account the fact that $P_i \geq 0$; that is, if a channel is too noisy, it doesn't get negative power, rather, it simply gets no power at all.

5 General Colored Gaussian Channel

This section was only briefly covered in the last six minutes of lecture, but a brief summary is given.

A general colored gaussian channel can be characterized by three parameters. These parameters are the number of parallel channels, k, the total power constraint, P, and a $k \times k$ covariance matrix, K_z. If the K_z is diagonal, then we are dealing with the case explained in section 4 where the noise on each channel is independent from the noise on every other channel. However, in general K_z is not diagonal. Thus, we would then like to understand the capacitance of the total channel and how to relate it to the case shown in section 4. To do this we use linear algebra to diagonalize K_z as shown below.

$$K_z = Q \cdot \Lambda \cdot Q^T$$

In the above equation, $QQ^T = I$, and the diagonal matrix, Λ, is given as

$$\begin{aligned} \Lambda_{i,j} &= \lambda_{i,i}; \quad if\ i = j \\ &\quad\ 0; \quad else. \end{aligned}$$

Thus we can convince ourselves that the capacity is as the capacity in the parallel gaussian channel with independent channel noise, total power, P, and $\sigma_i^2 = \lambda_i$. It is important to note that any covariance matrix can be diagonalized. Thus we can extend the colored gaussian channel to the case explained in detail in section 4.

6.441 Transmission of Information April 25, 2006

Lecture 18

Lecturer: Madhu Sudan *Scribe: Xiaomeng Shi*

Review of Last Lecture

Gaussian Channel

- Noise $\sim \mathcal{N}(0, \sigma^2)$
- Input power constraint P
- Capacity achieving input is Gaussian with variance P
- Capacity: $\frac{1}{2}\log(1 + \frac{P}{\sigma^2})$.

Colored Gaussian Channels

- Blocks of n elements transmitted each time
- The additive noise $Z \in \mathbb{R}^n$ is multivariate gaussian with covariance matrix K_Z (noise with memory)
- Input signal $X \in \mathbb{R}^n$ has covariance matrix K_X
- Input power constraint is nP, ie. $\text{trace}(K_X) \leq nP$.
- Capacity (without feedback):
$$C_n = \frac{1}{2}\log\frac{|K_X + K_Z|}{|K_Z|}$$
- Unlike memoryless channels, in channels with memory, feedback may increase capacity by as much as $\frac{1}{2}$ bit. A colored Gaussian channel with feedback has capacity
$$C_{FB,n} = \max_{K_X:\text{tr}(K_X)\leq nP} \frac{1}{2}\log\frac{|K_{X+Z}|}{|K_Z|} \leq C_n + \frac{1}{2}$$

Network Information Theory

So far, we have only considered single transmitter single receiver communication systems as shown on the left side of Figure 1. More generally, practical communication systems are more complex and may contain multiple senders and/or multiple receivers in various configurations. The second plot in Figure 1 illustrates such a system. The channel is not dedicated to one communication link, but shared between multiple users. Network information theory studies problems in such settings. There are many unsolved problems in network information theory, but some special networks are better understood than others. One example is the multiple access (MA) channel.

The Multiple Access Channel

The multiple access has many (m) senders and one receiver:
An example of MA channels is the ethernet. A MA channel can be charaterized by its input alphabet $\Omega_{X_1}, \Omega_{X_2}, \ldots, \Omega_{X_m}$, output alphabet Ω_Y, and probability transition function $P_{Y|X_1,\ldots,X_m}$. One question we would like to ask is, suppose the sources generate information at rate $R_i, i \in \{1, \ldots, m\}$. Is it feasible to transmit all messages correctly? Next we look at some simple examples of multiple access channels to study what rates are feasible.

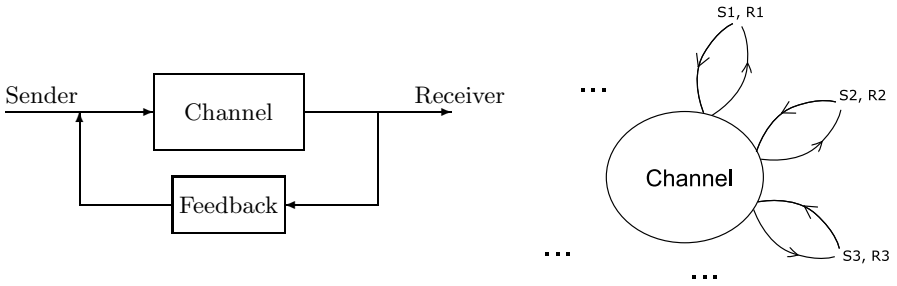


Figure 1: Single Channel vs. Network

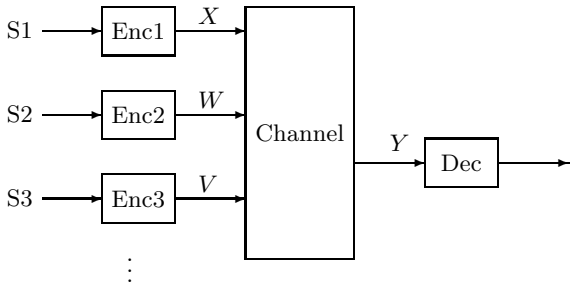


Figure 2: Multiple Access Channel

Examples of Multiple Access Channels

Parallel Channel: $Y = (X_1 + Z_1, X_2 + Z_2)$

Achievable Rates: $R_1 \leq C_1(X_1 \to X_1 + Z_1)$, $R_2 \leq C_2(X_2 \to X_2 + Z_2)$

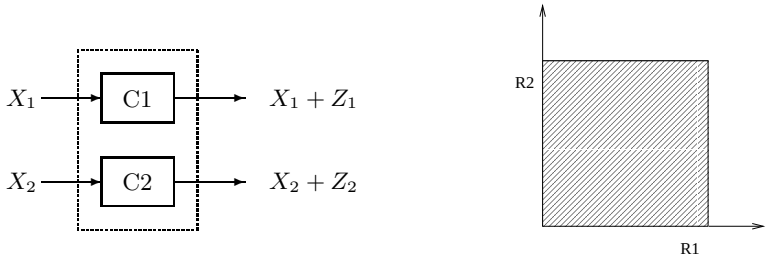


Figure 3: Parallel MA Channel

Binary Symmetric: $Y = X_1 + X_2 + Z(mod 2), X_1, X_2 \in \{0, 1\}, Z \sim Bern(p)$

- Setting $X_2 = 0$ achieves $R_1 = 1 - H(p)$
- Setting $X_1 = 0$ achieves $R_2 = 1 - H(P)$
- Time sharing between these two points gives a straight line $R_1 + R_2 = 1 - H(p)$.

Binary Erasure MA Channel: $Y = X_1 + X_2$

The binary erasure MA channel (first plot in Figure 5) adds its two inputs.
First, note:

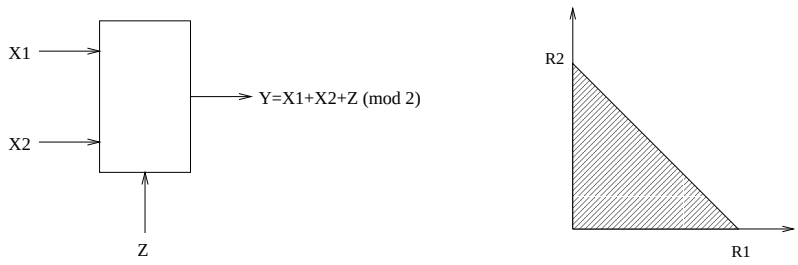


Figure 4: Binary Symmetric MA Channel

- Set $X_2 = 0 \Rightarrow$ noiseless channel with rate $R_1 \leq 1$.
- Set $X_1 = 0 \Rightarrow R_2 \leq 1$
- Time sharing gives a triangular shaped capacity region as in the symmetric channel $Y = X_1 + X_2 + Z (\text{mod } 2)$ case.

Can we do better?

The answer is yes:

- Assume $R_1 = 1$, ie., X_1 is always transmitted reliably.
- Decode X_2, regarding X_1 as noise (second plot in Figure 5), $X_1 \sim$Bern$(\frac{1}{2})$.
- The MA channel looks like a BEC for X_2 (last plot in Figure 5), $R_2 = \frac{1}{2}$.

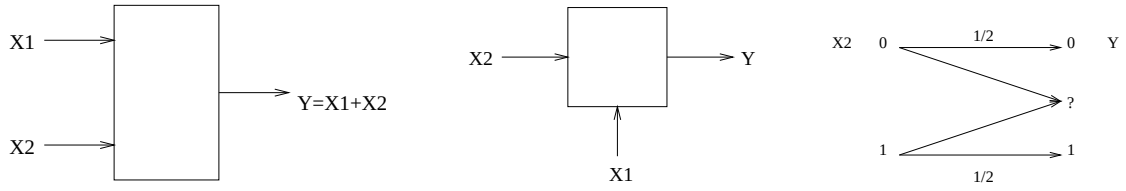


Figure 5: Binary Erasure MA Channel

$\therefore (1,0), (1,\frac{1}{2}), (\frac{1}{2},1), (0,1)$ are achievable rate pairs.

Time sharing then gives the following achievable rate region:

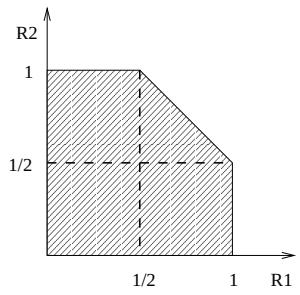


Figure 6: Achievable rate region of Binary Erasure MA Channel

Multiple Access Gaussian Channel

- $var(X_1) \leq P_1, var(X_2) \leq P_2, Z \sim \mathcal{N}(0, \sigma^2)$

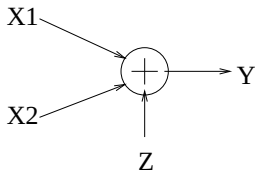


Figure 7: Multiple Access Gaussian Channel

- Set $X_2 = 0 \Rightarrow 0 \leq R_1 \leq \frac{1}{2}\ln(1 + \frac{P_1}{\sigma^2})$
- Set $X_1 = 0 \Rightarrow 0 \leq R_2 \leq \frac{1}{2}\ln(1 + \frac{P_2}{\sigma^2})$
- Decode one input regarding the other as noise $\Rightarrow R_1 + R_2 \leq \frac{1}{2}\ln(1 + \frac{P_1+P_2}{\sigma^2})$

The achievable region of a multiple access gaussian channel has the general shape same as Figure 6, except the vertices on the R_1, R_2 axis are located at $\left(0, \frac{1}{2}\ln(1 + \frac{P_2}{\sigma^2})\right), \left(\frac{1}{2}\ln(1 + \frac{P_1}{\sigma^2}), 0\right)$, and the slanted boundary line is $R_1 + R_2 \leq \frac{1}{2}\ln(1 + \frac{P_1+P_2}{\sigma^2})$.It can also be shown that instead of time-sharing, frequency division multiplexing can achieve the following capacity region:

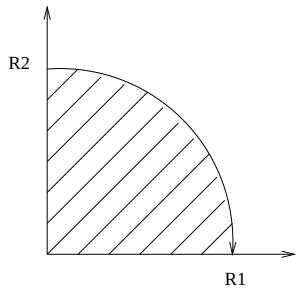


Figure 8: MA Gaussian Channel: Rate pairs achieved by FDM

Achievable Rate Pairs

For a multiple access channel, what does it mean exactly to have a achievable rate pair (R_1, R_2)?

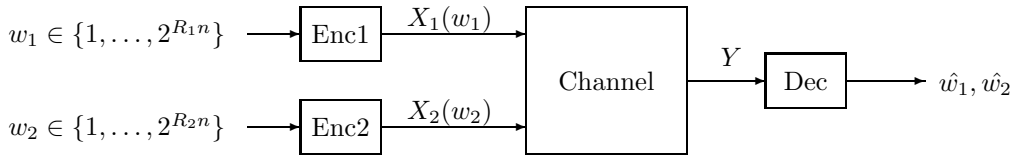
- (R_1, R_2) is achievable if there exist

  Encoding function: $X_1 : \{1, \ldots, 2^{R_1 n}\} \longrightarrow (\Omega_{X_1})^n$
  $X_2 : \{1, \ldots, 2^{R_2 n}\} \longrightarrow (\Omega_{X_2})^n$
  Decoding function: $Y : (\Omega_Y)^n \longrightarrow \{1, \ldots, 2^{R_1 n}\} \times \{1, \ldots, 2^{R_2 n}\}$

  such that decoding error probability approaches 0 when transmitting the messages w_1, w_2 independently generated (uniformly) on codebooks of size $2^{R_1 n}$ and $2^{R_2 n}$:

  $$w_1 \in \text{uniformaly on } \{1, \ldots, 2^{R_1 n}\} \qquad w_2 \in \text{uniformaly on } \{1, \ldots, 2^{R_2 n}\}$$

- As an illustration:



  If $(\hat{w_1}, \hat{w_2}) = (w_1, w_2)$ with probability $\to 1$, the rate pair (R_1, R_2) is achievable.

What rate pairs are achievable?

Theorem the rate pair $(\tilde{R_1}, \tilde{R_2})$ is achievable *iff* it is in the convex hull of points (R_1, R_2) such that there exist independent distributions P_{X_1}, P_{X_2} such that

$$\begin{aligned} 0 \leq R_1 &\leq I_1 = I(X; Y|W) \\ 0 \leq R_2 &\leq I_2 = I(W; Y|X) \\ R_1 + R_2 &\leq I_3 = I(X, W; Y) \end{aligned}$$

6.441 Transmission of Information April 27, 2006

Lecture 19

Lecturer: Madhu Sudan *Scribe: Mehmet Akçakaya*

1 Administrative Issues

- Project presentations in approximately 2 weeks from today.
- Report due in around 12 days.

2 Today

- Multiple Access Channels
- "Correlated Source Coding" a.k.a. Slepian-Wolf Theorem

3 Structure For Report/Presentation

- "Problem in English"
- Motivation - Why is this problem considered?
- Formal Model
- Theorem - Result - without going into the rigour at this point.
  At this point we've surpassed the attention span of most people in the audience.
- How? - Construction and Analysis (for the few who are still listening)

4 Multiple Access Channels

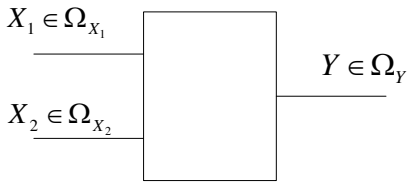


Figure 1: The Model

The multiple access channel is characterized by the input alphabets, Ω_{X_1} and Ω_{X_2}, the output alphabet, Ω_Y, and the transition probabilities, $p_{Y|(X_1,X_2)}$. We studied some specific channels in the last lecture, e.g. $Y = X_1 + X_2 + Z \bmod 2$, where all the alphabets were $\Omega = \{0, 1\}$.

Def (*Operational*): The rates (R_1, R_2) is achievable if $\exists$ encoding functions $X_1 : \{1, \ldots, 2^{nR_1}\} \to (\Omega_{X_1})^n$, $X_2 : \{1, \ldots, 2^{nR_2}\} \to (\Omega_{X_2})^n$ and decoding function $D : (\Omega_Y)^n \to \{1, \ldots, 2^{nR_1}\} \times \{1, \ldots, 2^{nR_2}\}$ such that $P_{\text{error}} \to 0$ as $n \to \infty$, meaning when the messages $W_1 \in_u \{1, \ldots, 2^{nR_1}\}$ and $W_2 \in_u \{1, \ldots, 2^{nR_2}\}$ are chosen independently, and we have $(W_1, W_2) \to (X_1(W_1), X_2(W_2)) \to Y \to (\widehat{W_1}, \widehat{W_2})$,

then $\mathbb{P}[(W_1, W_2) \neq (\widehat{W_1}, \widehat{W_2})] \to 0$ as $n \to \infty$.

Def (*Basic Achievable*): The rates (R_1, R_2) is basic achievable if $\exists$ distributions p_{X_1}, p_{X_2} with $(X_1, X_2) \sim p_{X_1} p_{X_2}$, such that

$$R_1 \leq I(X_1;\, Y|X_2) \tag{1}$$
$$R_2 \leq I(X_2;\, Y|X_2) \tag{2}$$
$$R_1 + R_2 \leq I(X_1, X_2;\, Y) \tag{3}$$

Thm(*Capacity*): (R_1, R_2) is achievable if and only if it lies in the convex hull of the basic achievable rates $(\tilde{R}_1, \tilde{R}_2)$.

Def (*Convex Hull*): Given $(R_1^{(1)}, R_2^{(1)}), \ldots, (R_1^{(k)}, R_2^{(k)})$, the convex hull of these points are the points, (R_1, R_2) that can be written as:

$$R_1 = \sum_{i=1}^{k} \lambda_i R_1^{(i)}$$
$$R_2 = \sum_{i=1}^{k} \lambda_i R_2^{(i)}$$

where $\{\lambda_1, \ldots, \lambda_k : \lambda_j \geq 0,\, \sum_j \lambda_j = 1\}$. Examples can be seen in Fig. 2

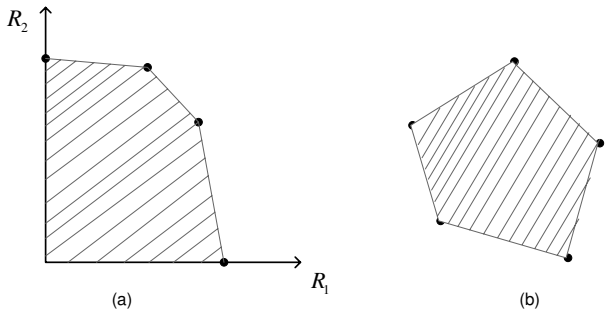


Figure 2: Examples of Convex Hulls: (a) with rates (b) in the plane

In other words, the theorem says (R_1, R_2) is achievable if and only if $\exists$ $(R_1^{(1)}, R_2^{(1)}), \ldots, (R_1^{(k)}, R_2^{(k)})$ basic achievable rates such that $R_1 = \sum_{i=1}^{k} \lambda_i R_1^{(i)}$ and $R_2 = \sum_{i=1}^{k} \lambda_i R_2^{(i)}$ with $\{\lambda_1, \ldots, \lambda_k : \lambda_j \geq 0,\, \sum_j \lambda_j = 1\}$.

Proof:
Achievability: We need

- Basic achievable pairs are achievable (shown via random coding and typical set decoding)
- Convex combinations are achievable (follows from a time-sharing argument)

Let $X_1(W_1)_i \sim p_{X_1}$ i.i.d. over W_1, i and $X_2(W_2)_i \sim p_{X_2}$ i.i.d. over W_2, i. Decoding function $D(Y)$ outputs (W_1, W_2) if $\exists!(W_1, W_2)$ such that $(X_1(W_1), X_2(W_2), Y)$ are jointly typical, else it outputs error.

When transmitting (W_1, W_2) a decoding error occurs when $(W_1', W_2') \neq (W_1, W_2)$ or the decoder outputs error:
- $(X_1(W_1), X_2(W_2), Y)$ is not jointly typical (by AEP the probability of this event $\to 0$ as $n \to \infty$).
- For $W_1' = W_1$, $W_2' \neq W_2$ (for fixed W_1, W_2), by joint AEP methods

$$\mathbb{P}[(X_1(W_1), X_2(W_2'), Y) \text{ is jointly typical}] \leq 2^{-nI(X_2;\,(X_1,Y))}$$

Thus the transmission will work if $R_2 \leq I(X_2;(X_1,Y)) = I(X_2;X_1)+I(X_2;Y|X_1) = I(X_2;Y|X_1)$. The last step follows since X_1 and X_2 are independent.

- Similar cases (i.e. $W_1' \neq W_1$, $W_2' = W_2$ and $W_1' \neq W_1$, $W_2' \neq W_2$) use similar inequalities.

Converse: Rigorous proof is omitted. But this follows from looking at the MAC in different ways: Looking at the MAC (Fig. 1) as a classical channel, i.e. point-to-point, we get $R_1 + R_2 \leq I(X_1, X_2; Y)$.

Alternatively we can look at it the other way (Fig. 3):

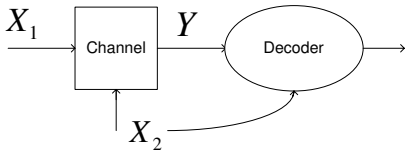


Figure 3: MAC viewed another way

In this case the decoder is more powerful than in the regular MAC case, since X_2 is available to it. We can view this as a point-to-point channel with additive noise X_2. Thus it follows reliable communication is possible only when $R_1 \leq I(X_1; Y|X_2)$. Since the decoder is more powerful than the MAC decoder, this will be an upper bound on the rate of communication possible with MAC.

5 Correlated Sources

The basic model is given in Fig. 4. Note that what makes this problem interesting is the fact that (X_1, X_2) are possibly dependent.

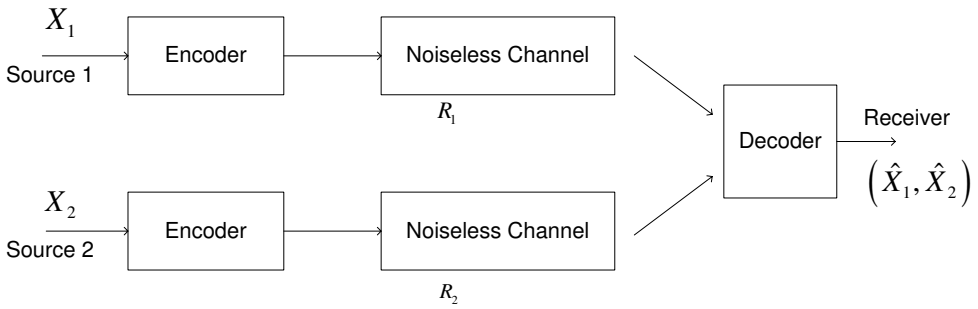


Figure 4: Correlated Sources Model

Ex: Let Z_0, Z_1, Z_2 be independent random variables with entropy H_0, H_1, H_2 respectively. Let $X_1 = (Z_0, Z_1)$ and $X_2 = (Z_0, Z_2)$ be the sources of interest. Note that $H(X_1) = H_0 + H_1$ and $H(X_2) = H_0 + H_2$.

It's easy to see that we can transmit at rates $R_1 = H_1$ and $R_2 = H_0 + H_2$, if we push all of Z_0 information through channel 2. Symmetrically we can transmit at $R_1 = H_0 + H_1$ and $R_2 = H_2$, if we push all of Z_0 information through cahhnel 1.

It follows naturally via time-sharing that we can transmit at the $R_1 = \alpha H_0 + H_1$ and $R_2 = (1-\alpha)H_0 + H_2$, for $0 \leq \alpha \leq 1$, by proportionately transmitting Z_0 information through channel 1

and channel 2.

Based on this example, we can hope (conjecture) that rates (R_1, R_2) are achievable if $R_1 \geq H(X_1|X_2)$, $R_2 \geq H(X_2|X_1)$ and $R_1 + R_2 \geq H(X_1, X_2)$. In fact this turns out to be the statement of our main theorem.

Thm(*Slepian-Wolf*): In the correlated sources model, rates (R_1, R_2) are achievable if and only if

$$R_1 \geq H(X_1|X_2) \tag{4}$$
$$R_2 \geq H(X_2|X_1) \tag{5}$$
$$R_1 + R_2 \geq H(X_1, X_2) \tag{6}$$

The idea is to transmit only the jointly typical sequences (X_1, X_2). This idea is illustrated in Fig. 5. Note that $H_1 = H(X_1)$, $H_2 = H(X_2)$, $I = I(X_1; X_2)$.

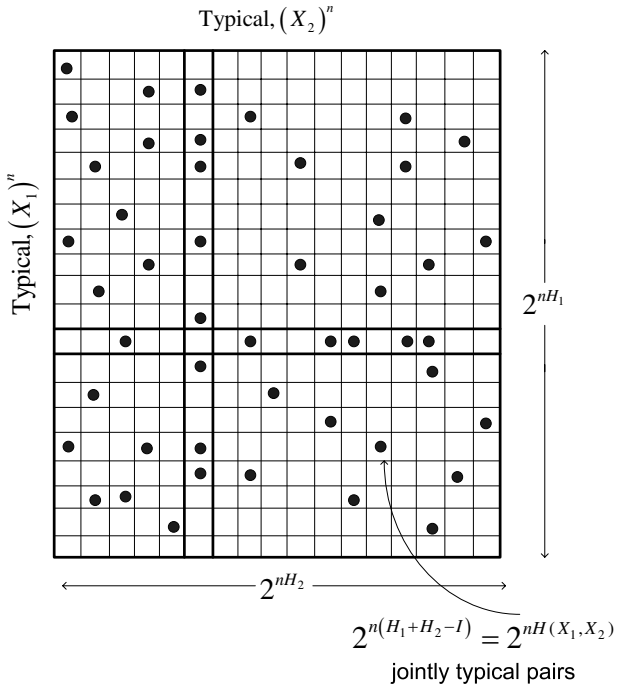


Figure 5: Slepian-Wolf Encoding

From the figure, we can infer the following:

$$\#\text{ dots per row} = \frac{\#\text{ dots}}{\#\text{ rows}} = \frac{2^{n(H_1+H_2-I)}}{2^{nH_1}} = 2^{n(H_2-I)} = 2^{nH(X_2|X_1)} \tag{7}$$

Similarly we'll have # dots per column $= 2^{n(H_1-I)} = 2^{nH(X_1|X_2)}$. The random coding argument goes as follows: We need to assign indices to each row, but we don't have 2^{nH_1} indices. Thus for each row, we pick an index randomly from $\{1, \ldots, 2^{nR_1}\}$. We do the same thing for the columns. Decoder will get "boxes" defined by the indices. If there's only one typical element in the box, then we output that element. If there's zero or more than one, then we declare an error. Formal proof will be given in the next lecture.

6.441 Transmission of Information May 2, 2006

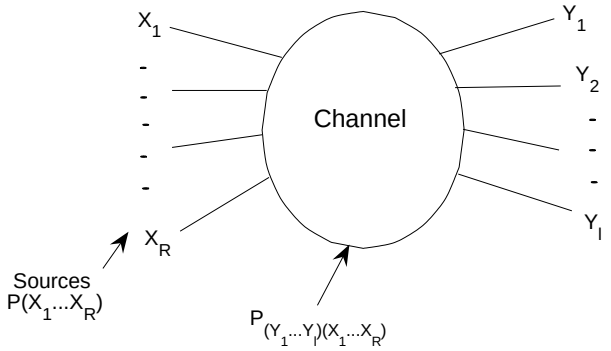
Lecture 20

Lecturer: Madhu Sudan *Scribe: Alaa Kharbouch*

1 Overview

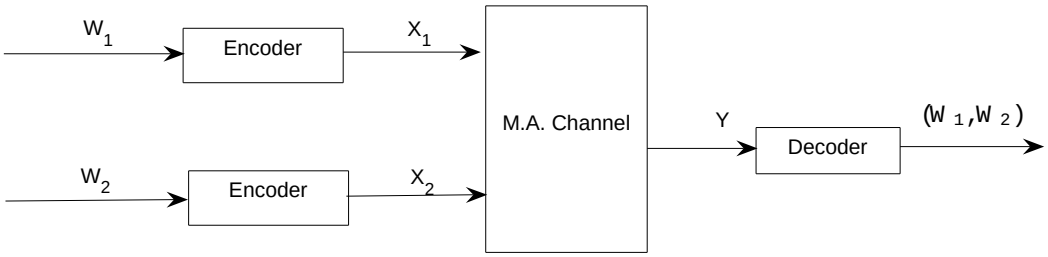
In this lecture, we will continue with the theme of network information theory.

- Correlated-Sources Coding
- Side Information (an aside)
- Broadcast Channel (We ran out of time and this will be covered in next lecture)



Channel is characterized by transition probabilities.
R_{ij} = requested rate from $X_i \to Y_j$

MULTIPLE ACCESS



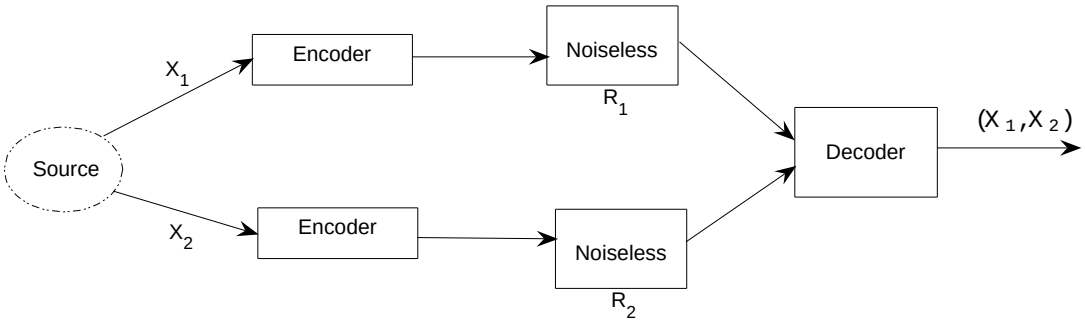
2 Correlated Source Coding

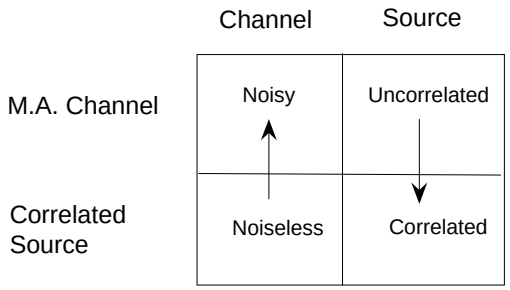
Where the arrows indicate how "hard" either case is (increasing in arrow direction).

Back to the Correlated Source Coding problem:

Definition: (R_1, R_2) achievable if $\exists\ f_1, f_2$

$$\begin{bmatrix} f_1 : \Omega_{X_1}^n \to \{1 \dots 2^{nR_1}\} \\ f_2 : \Omega_{X_2}^n \to \{1 \dots 2^{nR_2}\} \end{bmatrix}$$





$$g : \{1 \dots 2^{nR_1}\} \times \{1 \dots 2^{nR_2}\} \to \Omega_{X_1}^n \times \Omega_{X_2}^n$$

$(X_1, X_2) \longrightarrow (f_1(x), f_2(x)) \xrightarrow{\text{g}} (\hat{X}_1, \hat{X}_2)$

such that

$P_{err}^n = Pr[(X_1, X_2) \neq (\hat{X}_1, \hat{X}_2)] \underset{\text{n}}{\longrightarrow} 0$

X_1 is a source of entropy $H(X_1) = H_1$.
X_2 is a source of entropy $H(X_2) = H_2$
$I(X_1; X_2) = I$

3 Slepian-Wolf Theorem

(R_1, R_2) achievable iff

$$R_1 \geq H(X_1|X_2) = H_1 - I$$
$$R_2 \geq H(X_2|X_1) = H_2 - I$$
$$R_1 + R_2 \geq H(X_1, X_2) = H_1 + H_2 - I$$

ENCODING

Pick f_1 at random
Pick f_2 at random

Decoding$_{(f_1, f_2)}[Y_1, Y_2]$: if $\exists$ unique $(\hat{X}_1, \hat{X}_2)$ such that :

① $Y_1 = f_1(\hat{X}_1)$, $Y_2 = f_2(\hat{X}_2)$.
AND

② If $(\hat{X}_1, \hat{X}_2)$ are jointly typical then output $(\hat{X}_1, \hat{X}_2)$ else ERROR.

ANALYSIS:
Error Type 1 (X_1, X_2) is not jointly typical: $Pr \longrightarrow 0$ (LLN).

Error Type 2:

ⓐ $\exists \hat{X}_1 \neq X_1$, $\hat{X}_2 \neq X_2$ such that $(\hat{X}_1, \hat{X}_2)$ satisfy ① and ②.
To bound Pr[ⓐ]:
Fix $\hat{X}_1$, $\hat{X}_2$, X_1, X_2 with $\hat{X}_1 \neq X_1$ and $\hat{X}_2 \neq X_2$

$$\Pr_{f_1,f_2}[f_1(\hat{X}_1) = f_1(X_1) \text{and} f_2(\hat{X}_2) = f_2(X_2)] = 2^{-n(R_1+R_2)}$$

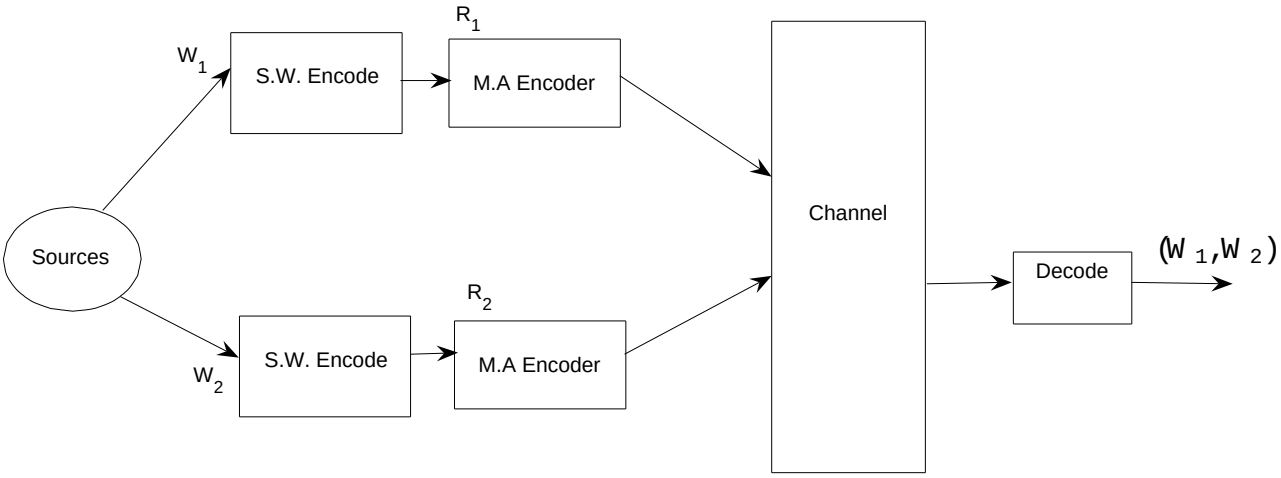
Union bound over $(\hat{X}_1, \hat{X}_2)$ jointly typical. Number of jointly typical $(\hat{X}_1, \hat{X}_2) \leq 2^{n(H(X_1,X_2)+\epsilon)}$.
If $R_1 + R_2 > H(X_1, X_2)$ then Pr[ⓐ]$\longrightarrow 0$.

ⓑ $\exists \hat{X}_1 \neq X_1$ such that $(\hat{X}_1, X_2)$ satisfy ① and ②.
Fix $\hat{X}_1$, X_1 with $\hat{X}_1 \neq X_1$

$$\Pr_{f_1,f_2}[f_1(\hat{X}_1) = f_1(X_1)] = 2^{-n(R_1)}$$

Union bound over $(\hat{X}_1, X_2)$ jointly typical. Number of $\hat{X}_1$ s.t. $(\hat{X}_1, X_2)$ jointly typical $= 2^{nH(X_1|X_2)}$.
Pr$[X_2] \leq 2^{-n(H(X_2)-\epsilon)}$, Pr$[\hat{X}_1, X_2] \geq 2^{-n(H(X_1,X_2)+\epsilon)}$. If $R_1 > H(X_1|X_2)$ then Pr[ⓑ]$\longrightarrow 0$.

ⓒ $\exists \hat{X}_2 \neq X_2$ such that $(X_1, \hat{X}_2)$ satisfy ① and ②. Similarly to ⓑ: If $R_2 > H(X_2|X_1)$ then Pr[ⓒ]$\longrightarrow 0$.



if $\exists\ R_1, R_2$

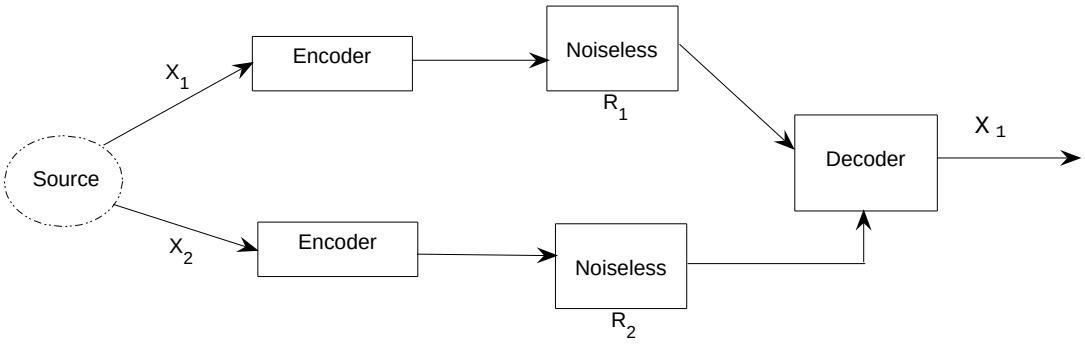
$$R_1 \geq H(W_1|W_2)$$
$$R_2 \geq H(W_2|W_1)$$
$$R_1 + R_2 \geq H(W_1, W_2)$$

and (R_1, R_2) are achievable for MA channel $(R_1, R_2) \in$ convex hull $(\widetilde{R}_1, \widetilde{R}_2)$ s.t. $P_{Y|(X_1,X_2)}$

$$\widetilde{R}_1 \leq I(X_1; Y|X_2)$$
$$\widetilde{R}_2 \leq I(X_2; Y|X_1)$$
$$\widetilde{R}_1 + \widetilde{R}_2 \leq I(Y; (X_1, X_2))$$

the transmission is feasible.

4 Side Information



$X_2 \longrightarrow$Decoder
Suffices if:

$$R_1 \geq H(X_1|X_2)$$
$$R_2 \geq H(X_2|X_1)$$
$$R_1 + R_2 \geq H(X_1, X_2)$$

For example: $X_1 = Z_1Z_2$ and $X_2 = Z_2Z_3$.
Then:

$$R_1 \geq H(X_1|X_2) = H(Z_1)$$
$$R_2 \geq H(X_2|X_1) = H(Z_3)$$
$$R_1 + R_2 \geq H(X_1, X_2) = H(Z_1) + H(Z_2) + H(Z_3)$$

Instead, since we don't care about X_2:

$$R_1 \geq H(Z_1)$$
$$R_2 \geq 0$$
$$R_1 + R_2 \geq H(Z_1) + H(Z_2) = H(X_1)$$

(R_1, R_2) suffices if $\exists\ \hat{X}_2$ s.t. $X_1 \longrightarrow X_2 \longrightarrow \hat{X}_2$
s.t.

$$R_1 \geq H(X_1|\hat{X}_2)$$
$$R_2 \geq H(\hat{X}_2|X_1)$$
$$R_1 + R_2 \geq H(X_1, \hat{X}_2)$$

What we want, or what would be nice to have is:

$$R_2 \geq H(\hat{X}_2|X_1) = 0$$
$$R_1 + R_2 \geq H(X_1, \hat{X}_2) = H(X_1)$$

OR

$$R_1 \geq H(X_1|\hat{X}_2)$$
$$R_2 \geq I(\hat{X}_2; X_1)$$

THEOREM:

Side information problem is realizable for (X_1, X_2) if (R_1, R_2) if $\exists\ \hat{X}_2$ s.t. $X_1 \longrightarrow X_2 \longrightarrow \hat{X}_2$

$$R_1 \geq H(X_1|\hat{X}_2)$$

$$R_2 \geq I(X_1; \hat{X}_2)$$

6.441 Transmission of Information May 4, 2006

Lecture 21

Lecturer: Madhu Sudan *Scribe: Vishal Doshi*

1 Review

- Recall the feasible rate region of the multiple access channel (two sources, one receiver) is the convex hull of rates (R_1, R_2) such that $\exists$ distributions p_{X_1} and p_{X_2} with

$$\begin{aligned} R_1 &\leq I(X_1; Y|X_2) \\ R_2 &\leq I(X_2; Y|X_1) \\ R_1 + R_2 &\leq I(X_1, X_2; Y) \end{aligned}$$

- Recall for encoding of two correlated sources, the rate region is given by

$$\begin{aligned} R_1 &\leq H(X_1|X_2) \\ R_2 &\leq H(X_2|X_1) \\ R_1 + R_2 &\leq H(X_1, X_2) \end{aligned}$$

- Thus, any (R_1, R_2) satisfying both sets of equations will be achievable in the general communications problem where we first source code and then channel code. However, this is not the best possible.

  Example. Consider a correlated source with $W_1 = W_2$ distributed as $Bern(\lambda)$ transmitted over an AWGN channel, where the noise has variance σ^2, and the power of each source is constrained to be $\leq P$. If we code X_1 and X_2 independently as channel coding requires, we would get a sum rate capacity of $R_1 + R_2 \leq \frac{1}{2}\log\left(1 + \frac{2P}{\sigma^2}\right)$. However, if we allow dependency in the channel coding, and let $X_1 = X_2$, then we can get sum rate $R_1 + R_2 \leq \frac{1}{2}\log\left(1 + \frac{4P}{\sigma^2}\right)$.

2 Broadcast Channel

- In the broadcast channel, we have a single center and multiple receivers with various rate requirements. The question we want to ask is what rate requirements are allowable.

- Formally, each subset (indexed by i) of receivers (except the null subset) is interested in a set of messages $S_i = \{1, \ldots, 2^{nR_i}\}$. The encoder takes a message tuple $(w_1, \ldots, w_{2^n-1}) \in S_1 \times \cdots \times S_{2^n-1}$ and produces some x to send over the channel. The channel is defined by a distribution $p_{Y_1,\ldots,Y_n|X}$. Thus the rates $(R_1, \ldots, R_n)$ are achievable if there exists an encoder and n decoders such that the probability of each receiver accurately receiving its relevant information is high.

2.1 Degraded Broadcast Channel

A degraded broadcast channel is one in which $X \to Y_1 \to Y_2$ holds. Note that not every channel has an equivalent degraded broadcast channel.

We think of a BSC degraded broadcast channel as a cascade of two BSC channels with the first output at the output of the first BSC channel and the second as the end of the cascade. Thus we can think of the output at the first as a BSC(p_1) and the second output as a BSC($p_1 * p_2$), where $p_1 * p_2 = p_1(1 - p_2) + p_2(1 - p_1)$.

For this channel with no joint rate requirements and a particular block length n, we can think of a good coding scheme in which we spread 2^{nR_1} messages so that the Hamming distance between any two of the messages allows us to satisfy our error requirement. Then for each message, we have associated a

ball that will decode (for receiver 1) to a given message. In each of these balls, we place smaller balls that will contain the information for receiver 2.

In this setup, for reliable communication to be possible, volume arguments show that if we send at $R_2 = 1 - H(p_1 * p_2)$, we need $2^{nR_1} \leq \frac{2^{nH(p_1 * p_2)}}{2^{nH(p_1)}}$ or $R_1 \leq H(p_1 * p_2) - H(p_1)$.

The proof of the following general result is omitted.

Theorem 1 (Capacity Theorem for Degraded Broadcast Channels) (R_1, R_2) *achievable iff* $\exists U$ *such that* $U \to X \to Y_1 \to Y_2$ *and*

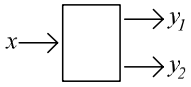
$$R_1 \leq I(X, Y_1|U)$$
$$R_2 \leq I(U, Y_2)$$

6.441 Transmission of Information May 9, 2006

Lecture 22

Lecturer: Madhu Sudan *Scribe: Jonathan Rohrs*

1 Broadcast Channels



This channel broadcasts messages to two receivers, and is characterized by the joint (marginal) probability mass function:

$$P_{(y_1,y_2|x)}$$

and alphabets for the inputs and two outputs.

1.1 Independent Information

Suppose one wishes to use such a channel to transmit some message
($W_1 \in \{1, \ldots, 2^{nR_1}\}$) to one receiver, and some other message ($W_2 \in \{1, \ldots, 2^{nR_2}\}$) to the other. This is done as follows

1. An encoder (E) is used to encode the two messages as one word:

$$(W_1, W_2) \xrightarrow{E} X^n$$

2. The channel (Ch) produces its two outputs words according to is probability distribution ($P_{(y_1,y_2|x)}$)

$$X^n \xrightarrow{Ch} (Y_1^n, Y_2^n)$$

3. Two decoders (D_1 and D_2) are used to decode the respective output words, producing the decoded messages ($\hat{W}_1$ and $\hat{W}_2$)

$$Y_1^n \xrightarrow{D_1} \hat{W}_1$$

$$Y_2^n \xrightarrow{D_2} \hat{W}_2$$

   The two receivers are not allowed to collude in decoding their messages.

1.2 Achievable Transmission Rates

As usual, we are interested in the possible transmission rates. We say that the *rate pair* (R_1, R_2) is *achievable* if there exists an encoder (E) and two decoders (D_1 and D_2) such that for any two messages (W_1 and W_2), the probability of decoding error goes to zero with n. That is,

$$P_{err} = \Pr(\,(W_1, W_2) \neq (\hat{W}_1, \hat{W}_2)\,) \longrightarrow 0$$

Unfortunately, useful results for this general case do not exist. Instead, we investigate special cases, namely degraded channels.

1.3 Stochastic Equivalence

We say two broadcast channels are *stochastically equivalent*

$$P_{(y_1,y_2)|x} =_{stoc} P'_{(y_1,y_2)|x}$$

if they share the same marginal distributions:

$$P_{y_1|x} =_{stoc} P'_{y_1|x}$$

$$P_{y_2|x} =_{stoc} P'_{y_2|x}$$

Achievable rate pairs depend only on the marginal distribution of the channel. This means that if two channels are stochastically equivalent:

$$P =_{stoc} P'$$

then they share the same set of achievable rate pairs:

$$(R_1, R_2) \text{ is achievable by } P \iff (R_1, R_2) \text{ is achievable by } P'$$

This can be useful, in that it sometimes allows one to find rate results for a broadcast channel by studying another, simpler but equivalent, channel instead.

1.4 Degraded Broadcast Channels

Degraded channels are a special case of broadcast channels. We study them in large part because they give clean results where results for general broadcast channels have not been found.

1.4.1 Definitions

Degraded broadcast channels come in two types:

- *Physically* degraded broadcast channels are those that form a Markov chain:

  $$X \to Y_1 \to Y_2$$

  It is as if the channel degrades the signal between the source and the first receiver, and then degrades it some more before the second receiver.

- *Stochastically* degraded broadcast channels are those that are stochastically equivalent to a physically degraded broadcast channel.

  The definitions of stochastic equivalence and physically degraded channels (together) imply that a channel (P) is stochastically degraded if there exists a distribution (P') such that

  $$p(y_2|x) = \sum_{y_1} p(y_1|x) \cdot p'(y_2|y_1)$$

Typically, one will prove rate results in terms of physically degraded channels, but the same results will apply to stochastically degraded channels, since they share the same set of achievable rate pairs.

2 Coding Theorem for Degraded Channel

Assume (without loss of generality) that we are given a physically degraded channel.

2.1 The Theorem

$\exists U$ s.t. $U \to X \to Y_1 \to Y_2$ and

$$\left.\begin{array}{c} R_2 \le I(U;Y_2) \\ \text{and} \\ R_1 \le I(X;Y_1|U) \end{array}\right\} \Longrightarrow (R_1, R_2) \text{ is achievable}$$

2.2 The Converse

$$(R_1, R_2) \text{ is achievable } \Longrightarrow \left\{\begin{array}{c} \exists U \text{ s.t. } U \to X \to Y_1 \to Y_2 \\ \text{and} \\ R_2 \le I(U;Y_2) \\ \text{and} \\ R_1 \le I(X;Y_1|U) \end{array}\right.$$

Furthermore, U takes values from a set of size

$$|\Omega_U| \le \min(\, |\Omega_X|,\, |\Omega_{Y_1}|,\, |\Omega_{Y_2}|\,)$$

2.3 Theorem Proof

Given some (W_1, W_2), use the following scheme for encoding:

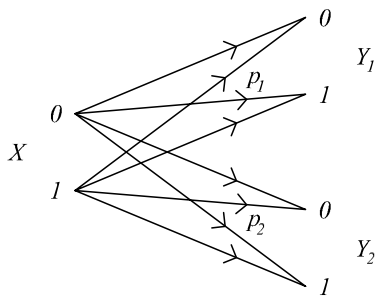
1. Map $W_2 \to U^n = (U_1, U_2, \ldots, U_n)$ s.t.
$$U_i \sim P_U$$
independent for each W_2 and i.
2. Map $(U^n, W_1) \to X^n = (X_1, X_2, \ldots, X_n)$ s.t.
$$X_i \sim P_{X|U}$$
independent for each W_1 and i.

Then use this scheme for decoding:

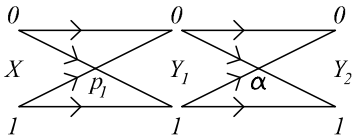
- Receiver 2: If there exists a unique $\hat{W}_2$ s.t.
$$(\, U(\hat{W}_2),\, Y_2^n \,) \text{ is jointly typical}$$
then output $\hat{W}_2$. Otherwise `error`.
- Receiver 1: If there exists a unique $(\hat{W}_1, \hat{W}_2)$ s.t.
$$(\, U(\hat{W}_2),\, X(U(\hat{W}_2), \hat{W}_1)\,,\, Y_1^n \,) \text{ is jointly typical}$$
then output $\hat{W}_1$. Otherwise `error`.

3 Example

Consider a pair of symmetric binary channels with bit-flip parameters p_1 and p_2, forming a broadcast channel as shown below:



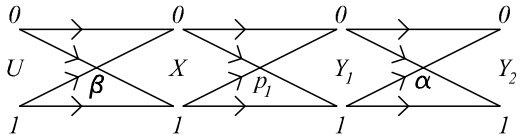
Since the bit-flips are independent, this does *not* form a physically degraded channel. However, it is stochastically equivalent to the physically degraded channel:



if α is chosen so that

$$p_2 = p_1 \star \alpha = p_1(1-\alpha) + \alpha(1-p_1)$$

For the purposes of applying the coding theorem for degraded channels, add an additional variable $U \sim Bern(\frac{1}{2})$ and a third symmetric binary channel stage (with parameter β):



In summary, alternately stated (using modulo 2 arithmetic),

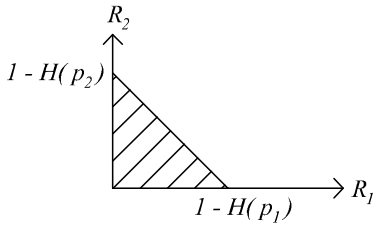
$$\begin{array}{lll} U \sim Bern(\frac{1}{2}) & & \\ X = U + Z_1 & \text{where} & Z_1 \sim Bern(\beta) \\ Y_1 = X + Z_2 & \text{where} & Z_2 \sim Bern(p_1) \\ Y_2 = Y_1 + Z_3 & \text{where} & Z_3 \sim Bern(\alpha) \end{array}$$

Applying the theorem: For fixed β, p_1, and p_2, we can achieve rates

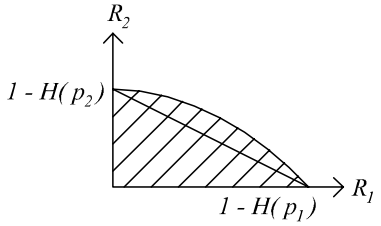
$$\begin{aligned} R_1 &= H(\beta \star p_1) - H(p_1) \\ R_2 &= 1 - H(\beta \star p_2) \end{aligned}$$

If $p_1 = p_2$, then

$$R_2 = 1 - H(p_1) - R_1$$

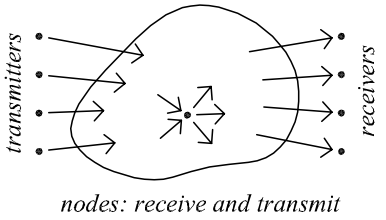


This is equivalent to time-sharing. By contrast, if $p_1 < p_2$ (and $R_1, R_2 \neq 0$), one can gain compared to time-sharing.



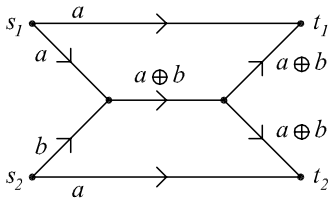
4 Network Information Theory

In general, a network may have many transmitters, receivers, and nodes (which both transmit and receive):



This may operate in a number of ways.

- *Store and Forward*: Nodes are not allowed to do calculations. Network capacity can be found in exponential time.
- *Recompute and Redistribute*: Nodes can compute as well. This may allow for an increase in capacity. For example:



  If each edge has capacity of 1, then the network shown can achieve a rate pair $(1, 1)$. If computation was not allowed (i.e. node could not calculate $a \oplus b$), one could only achieve $(\frac{1}{2}, \frac{1}{2})$.

Network information theory is an active area of research. See Li and Yeung for more information on what is known and what is still open.

6.441 Transmission of Information May 16, 2006

Lecture 23

Lecturer: Madhu Sudan *Scribe: Chung Chan*

1 Coding theory

1.1 Introduction

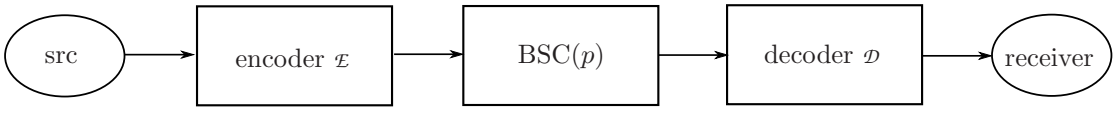


Figure 1: Notations

Consider the binary symmetric channel BSC(p) in Figure 1. Shannon's random coding scheme for any fixed rate $R = 1 - H(p) - \epsilon$ with small ϵ and dimension $k = \lfloor Rn \rfloor$ achieves asymptotically zero error probability $\Pr(\mathcal{E})$ that decays exponentially with the optimal error exponent $E_r(R) = E_{sp}(R) = D[\delta_{GV}(R) \parallel p] = o(\epsilon)$, where $E_r(R), E_{sp}(R)$ and $\delta_{GV}(R)$ are the random ocding exponent, sphere packing error exponent, and Gilbert-Varshamov distance respectively.[1] However, the theorem suggests neither an efficient way of finding the best codebook (other than exhaustive search) nor any special structure of the best code for efficient decoding (other than the computationally intensive ML decoding). Table 1 summarizes the complexity of the naive approach,

	storage	complexity
encoding	$2^k n \doteq 2^{Rn}$ (size of the codebook)	$2^{2^k n} \approx 2^{2^k}$ (number of codebooks for an exhaustive search of the best one)
decoding	$2^k n$ 2^n (size of the decoding table)	2^n (computing the Hamming distance from the observed sequence to each of the valid codeword) efficient (with a decoding table that maps every possible observation sequence to their optimal message hypotheses.)

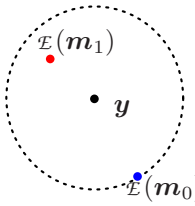
Table 1: Complexity of designing best channel code.

1.2 Smaller ensemble of good codes

In analyzing the probability of error for random code ensemble (RCE), we exploited the pairwise independence property among the randomly chosen codewords $\mathcal{E}(\boldsymbol{m}_0), \mathcal{E}(\boldsymbol{m}_1), \ldots, \mathcal{E}(\boldsymbol{m}_{2^k-1}) \in \{0,1\}^n$ although RCE guarantees a stronger property of mutual independence. Since the stronger property of mutual independence is not needed, we may be able to reduce complexity by relaxing the random code to have only pairwise but not necessarily mutually independent codewords.

How can we have pairwise independence but not mutual independence? Consider the following simpler task of having only onewise independence: choose $\mathcal{E}(M_0), \ldots, \mathcal{E}(M_{2^k-1})$ such that each codeword

[1] To show that the error exponent is $o(\epsilon)$ (as $\epsilon \to 0$), show that $\frac{\partial}{\partial d} D[d \parallel p]\Big|_{d=p} = 0$. For simplicity, we approximate the error exponent by a quadratic function $f(p)\epsilon^2$.



$\Pr[\mathcal{E}(\boldsymbol{m}_1)$ in ball pn around $\boldsymbol{y}|\boldsymbol{y}, \mathcal{E}(\boldsymbol{m}_0)]$ is independent of $\mathcal{E}(\boldsymbol{m}_0)$

Figure 2: How pairwise independence is used in computing $\Pr(\mathcal{E})$

is uniform over the 2^n possibilities. The simplest choice is to set $\mathcal{E}(M_0) = \mathcal{E}(M_1) = \cdots = \mathcal{E}(M_{2^k-1})$ and uniformly distributed.

To have pairwise independence, consider imposing the linearity/affine constraint:

$$\mathcal{E}(\boldsymbol{m}_i) = \boldsymbol{A}^{(k\times n)}\boldsymbol{m}_i^{(k\times 1)} + \boldsymbol{b}^{(n\times 1)}$$

for some randomly chosen $\boldsymbol{A}$ and $\boldsymbol{b}$ where the multiplication and addition are modulo two. If $\boldsymbol{A}$ and $\boldsymbol{b}$ are uniformly random, $\mathcal{E}(\boldsymbol{m}_i)$ and $\mathcal{E}(\boldsymbol{m}_j)$ are independent iff $\boldsymbol{m}_i \neq \boldsymbol{m}_j$. This is true even if $\boldsymbol{b}$ is an all-zero vector because every element of $\mathcal{E}(\boldsymbol{m}_i)$ can be thought of as the corresponding element of $\mathcal{E}(\boldsymbol{m}_j)$ corrupted by a BSC(0.5). If $\boldsymbol{A}$ is a toeplitz matrix, i.e.

$$\boldsymbol{A} = \begin{bmatrix} a_0 & a_{-1} & a_{-2} & \dots & a_{1-n} \\ a_1 & a_0 & a_{-1} & \dots & a_{2-n} \\ a_2 & a_1 & a_0 & \dots & a_{3-n} \\ \vdots & & & & \vdots \\ a_{k-1} & a_{k-2} & a_{k-3} & \dots & a_{k-n} \end{bmatrix}$$

whose entries $a_{1-n}, \dots, a_{k-1}$ are iid Bern(0.5), we also have the desired independence as long as $\boldsymbol{b}$ is uniformly random so that it breaks the dependence among different coordinates.[2]

With the linear random code $\boldsymbol{Am}$ or random affine code $\boldsymbol{Am} + \boldsymbol{b}$ with the Toeplitz matrix $\boldsymbol{A}$, we reduced the numbers of parameters or degrees of freedom to nk and $k+2n-1$ respectively. Effectively, the search space for the best code with the corresponding constraints reduced from doubly exponential 2^{2^k} for RCE to exponential (2^{nk} and 2^{k+2n-1} respectively) without affecting the error exponent *averaged* over the ensemble of codes.[3] The goal now is to further reduce the complexity to polynomial by eliminating parameters of less interest to us.

Consider the following approach,

1. divide the sequence $\boldsymbol{m}$ of k information bits into successive blocks of length $l = 10 \log n$.
2. encode each block separately by the *same* code.

The search space of the best code is polynomial $2^{\frac{l}{R}} = n^{10/R}$ but the probability of error is at least the probability of an error in the first block, which is also at least polynomial $\Pr(\mathcal{E}) \geq 2^{-E_r(R)l/R} = n^{-10E_{sp}(R)/R}$ by the sphere-packing upper bound on error exponent. Is there a way to make the error probability decay exponentially fast in n? The results from Reed Solomon and Peterson in 1960, and Forney in 1966 gives an affirmative answer.

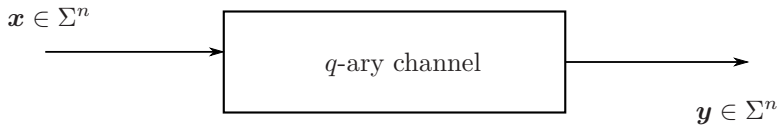
1.3 Concatenation code

Consider the q-ary channel in Figure 3, where Σ is some q-ary alphabet such that with the appropriate definition of addition and multiplication, Σ forms a finite field.[4] This allows us to define polynomials of

[2] To see this, consider the simple $k = n = 2$ case and compare $\mathcal{E}\left(\boldsymbol{m}_0 = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]\right)$ and $\mathcal{E}\left(\boldsymbol{m}_1 = \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]\right)$.

[3] The best codebooks under the different constraints need not be the same, and hence their error exponent need not be the same.

[4] This requires q to be some positive integral power of a prime number. i.e. $q = (\text{prime})^{(\text{positive integer})}$. In the simple case when q is prime, we can use modulo-q addition and multiplication and the resultant field is called the prime field.



Hamming distance: $\Delta(\boldsymbol{x}, \boldsymbol{y}) := |\{i : 1 \le i \le n, x_i \ne y_i\}|$

Figure 3: q-ary channel

$x \in \Sigma$ in the form of $f(x) := \sum_{i=0}^{k-1} f_i x^i$ (i.e. degree less than k) with $f_i \in \Sigma$ so that they satisfy the fundamental theorem of algebra that a degree $j < k$ polynomial have j roots that are not necessarily distinct. Given $\alpha \in \Sigma$, $f(\alpha) \in \Sigma$ denotes the evaluation of the polynomial f at the point α.

In Reed Solomon code,

1. n distinct points $\beta_1, \ldots, \beta_n \in \Sigma$ are chosen offline and known to both encoder and decoder.
2. The encoder represents the q-ary information k-sequence $f_0, \ldots, f_{k-1}$ as the polynomial $f(x) := \sum_{i=0}^{k-1} f_i x^i$ and then evaluates it at each of the n chosen points. The matrix representation of the evaluation procedure is,

$$\begin{bmatrix} f(\beta_1) \\ f(\beta_2) \\ f(\beta_3) \\ \vdots \\ f(\beta_n) \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & \beta_1 & \beta_1^2 & \cdots & \beta_1^{k-1} \\ 1 & \beta_2 & \beta_2^2 & \cdots & \beta_2^{k-1} \\ 1 & \beta_3 & \beta_3^2 & \cdots & \beta_3^{k-1} \\ \vdots & & & & \vdots \\ 1 & \beta_n & \beta_n^2 & \cdots & \beta_n^{k-1} \end{bmatrix}}_{\text{Vandermonde matrix}} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_k \end{bmatrix}$$

3. The evaluation sequence $\boldsymbol{x} = [\, f(\beta_1) \cdots f(\beta_n) \,]$ is transmitted through the q-ary channel.
4. The decoder estimates $f_0, \ldots, f_{k-1}$ from $\boldsymbol{y}$ (polynomial interpolation) successfully in polynomial time if $\Delta(\boldsymbol{x}, \boldsymbol{y}) \le \frac{n-k}{2}$.[Peterson 1960]

Roughly speaking, the error-correction capability of the code stems from the structure of polynomial evaluation or the transformation by a Vandermonde matrix. The polynomial decoding time is due to the efficient implementation of finite-field arithmetics. As a sanity check to see how the polynomial structure help recover the information, let us prove that the information sequence $f_0, \ldots, f_{k-1}$ is recoverable if there is no error and $n \ge k$. Suppose, for contradiction, that there exists $f' \ne f$ such that $f'(x) = f(x)$ at the n distinct points. In other words, $f'(x) - f(x)$ is a polynomial with $\deg < k \le n$ but n distinct roots $\beta_1, \ldots, \beta_n$. This contradicts the fundamental theorem of algebra and thus gives the desired result.

Going back to the BSC(p), we can improve the coding by a layered architecture in Figure 4: concatenating an outer 2^l-ary Reed-Solomon code with an inner binary code with the Toeplitz structure as follows

1. divide the k-bit information sequence $\boldsymbol{m}$ into k/l consecutive blocks of length $l := c \log n$ for some constant c.
2. the encoder treat the sequence as one block of k/l 2^l-ary symbol, and uses the Reed-Solomon code to encode it into a block of $(1+\delta)k/l$ 2^l-ary symbols.
3. the encoder now treat the sequence as $(1+\delta)k/l$ blocks of binary l-sequence and uses the same binary code to encode each block to a binary l/R-sequence of length l/R.
4. The entire binary n-sequence $\boldsymbol{x}$ is transmitted through the BSC(p).
5. The decoder receive the binary n-sequence $\boldsymbol{y}$, treat it as k/l blocks of binary l/R-sequence and uses the inner binary code to decode each block to a binary $(1+\delta)k/l$-sequence.

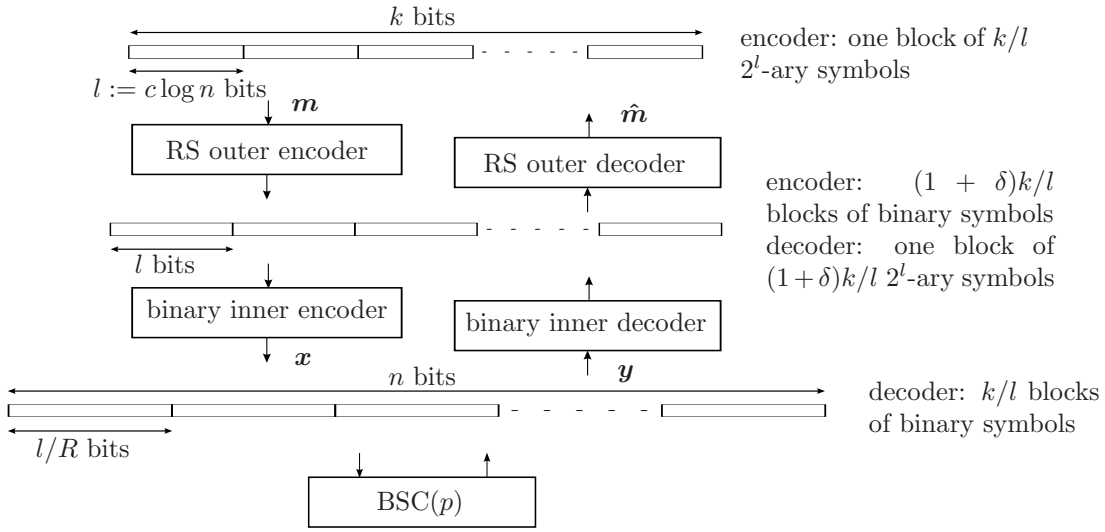


Figure 4: Concationation code

6. The decoder treat the sequence as one block of $(1+\delta)k/l$ 2^l-ary symbols and uses the Reed-Solomon outer code to decode it to an estimate of the original k-bit information sequence.

The Reed-Solomon code can correct up to $\delta k/2l$ errors in the block of $(1+\delta)k/l$ 2^l-ary symbol sequence while the probability of error in a particular block is $2^{-E_r(R)l/R} = n^{-cE_r(R)/R}$. By the union bound, the overall probability of error is upper bounded by $\binom{(1+\delta)k/l}{\delta k/2l}(2^{-E_r(R)l/R})^{\delta k/2l} \approx 2^{-n(E_r(R)\delta/2 - H(\delta/2(1+\delta)))}$.

6.441 Transmission of Information May 18, 2006

Lecture 24

Lecturer: Madhu Sudan *Scribe: Chung Chan*

1 Kolmogorov complexity

Shannon's notion of compressibility is closely tied to a probability distribution. However, the probability distribution of the source is often unknown to the encoder. Sometimes, we interested in compressibility of specific sequence. e.g. How compressible is the Bible? We have the Lempel-Ziv universal compression algorithm that can compress any string without knowledge of the underlying probability except the assumption that the strings comes from a stochastic source. But is it the best compression possible for each deterministic bit string? This notion of compressibility/complexity of a deterministic bit string has been studied by Solomonoff in 1964, Kolmogorov in 1966, Chaitin in 1967 and Levin.

Consider the following n-sequence

$$0100011011000001010011100101110111\cdots$$

Although it may appear random, it is the enumeration of all binary strings. A natural way to compress it is to represent it by the procedure that generates it: enumerate all strings in binary, and stop at the n-th bit. The compression achieved in bits is

$$|\text{compression length}| \leq 2\log n + O(1)$$

More generally, with the universal Turing machine, we can encode data to a computer program that generates it. The length of the smallest program that produces the bit string $\boldsymbol{x}$ is called the Kolmogorov complexity of $\boldsymbol{x}$.

Definition 1 (Kolmogorov complexity) *For every language $\mathcal{L}$, the Kolmogorov complexity of the bit string $\boldsymbol{x}$ with respect to $\mathcal{L}$ is*

$$K_{\mathcal{L}}(\boldsymbol{x}) = \min_{p:\mathcal{L}(p)=x} l(p)$$

where p is a program represented as a bit string, $\mathcal{L}(p)$ is the output of the program with respect to the language $\mathcal{L}$, and $l(p)$ is the length of the program, or more precisely, the point at which the execution halts.

But this notion of complexity depends on the particular language $\mathcal{L}$, which seems too specific to be useful. For example, we may have a language that prints the bible by a short program, say the ASCII string of "print the bible". The length of the program depends of the language so much that it does not seem to reflect our intuitive understanding of what compressibility is. Without fixing a particular language, however, the notion of complexity is ill-defined. Fortunately, we have the following theorem of the universal language which roughly says that the Kolmogorov complexity of $\boldsymbol{x}$ with respect to a universal language is well-defined up to a constant.

Theorem 2 (Universal language)

$$(\exists \textit{ universal language } \mathcal{U})(\forall \textit{ language } \mathcal{L})(\exists \textit{ finite constant } C_{\mathcal{L}})(\forall \textit{ bit string } \boldsymbol{x})$$
$$K_{\mathcal{U}}(\boldsymbol{x}) \leq K_{\mathcal{L}}(\boldsymbol{x}) + C_{\mathcal{L}}$$

Can we choose the best universal language $\mathcal{U}$ that minimizes $C_{\mathcal{L}}$ over all choices of $\mathcal{L}$? Such a minimum may not exist because $C_{\mathcal{L}}$, although finite, is unbounded.[1]

[1] Given any universal language $\mathcal{U}$ that one claims to be the best, we can find a finite bit string $\boldsymbol{x}$ that $\mathcal{U}$ compresses to more than 1 bit, and then give a universal language $\mathcal{U}'$ that compresses $\boldsymbol{x}$ to exactly 1 bit by storing $\boldsymbol{x}$ within the language manual.

To relate Kolmogorov complexity to Shannon's notion of compressibility, let us ask the following question: what is the probability model $P_{\mathcal{U}}$ under which the compression using the shortest program is good? Intuitively, sequences that can be generated by shorter programs should be more probable. i.e. $K_{\mathcal{U}}(\boldsymbol{x}) \approx \log \frac{1}{P_{\mathcal{U}}(\boldsymbol{x})}$ along the idea of entropy encoding. This can be satisfied with the following model for generating $\boldsymbol{X}$,

1. Fix a universal language $\mathcal{U}$.
2. Generate a random program p from an iid Bern(0.5) source.
3. Generate $\boldsymbol{X}$ as the output $\mathcal{U}(p)$.

The corresponding distribution $P_{\mathcal{U}}$ is called the universal probability, defined as follows,

Definition 3 (Universal probability)

$$P_{\mathcal{U}}(\boldsymbol{x}) = \sum_{p:\mathcal{U}(p)=\boldsymbol{x}} 2^{-l(p)}$$

1.1 Spectrum of research on data compression

Shannon's model assumes a known distribution, and easy optimal compression schemes are available. On the other hand, Kolmogorov model is robust under unknown distribution, but the compression, which involves searching for the shortest programs for strings, is incomputable. Fortunately, there are a variety of models between these two extremes.

Lempel Ziv Model assumes an finite-state Markov chains that may be unknown to the encoder. The compression algorithm is easy and is implemented in practice.

Keiffer-Yang uses grammars to compress strings. For example, a grammar may consists of the following rules with the associated probabilities:

$$\begin{array}{lll}
\text{sentence} \xrightarrow{.9} \text{subject,Verb,Object} & \text{subject} \xrightarrow{.9} \text{noun} & \\
\text{sentence} \xrightarrow{.1} \text{one word} & \text{subject} \xrightarrow{.1} \text{pronoun} & \\
\text{Verb} \xrightarrow{.2} \text{an} & \text{pronoun} \xrightarrow{.99} \text{I} & \text{object} \xrightarrow{.3} \text{me} \\
\vdots & \vdots & \vdots
\end{array}$$

Charikar, Sahai, Lehman etc. have come up with efficient grammars for polynomial-time compression algorithms.

Resource bounded Kolmogorov complexity $K_{\mathcal{U}}^{n^2}(\boldsymbol{x})$ is defined as the length of the smallest program p that produces $\boldsymbol{x}$ in time $l(\boldsymbol{x})^2$. The compression can be done within $2^{l(\boldsymbol{x})^2}$, basically by searching through all the possible programs that satisfies the resource constraint.

2 Summary of the course

Starting with the basic probability theory, we defined the entropy and mutual information as an intuitive measure of the uncertainty of a random variable and that of the information shared between two random variables. Then, we introduced the AEP, which comes in handy when we tackle large objects from small processes. In particular, we applied it in typical set source encoding and channel decoding. Random encoding is another important notion that simplified the error analysis when proving achievability results of source and channel coding. We then introduced the differential entropy as the appropriate measure of randomness of a continuous random variable, not in the absolute sense, but in the relative sense for the purpose of comparing to another random variable in the same coordinate system. Finally, we

introduced the network information theory for multiple access and broadcast channels, coding theory and Kolmogorov complexity. The need for information theory to address computational complexity is important for designing practical systems, such as channel codes with efficient encoding and decoding algorithms, and cryptographic systems that is computationally infeasible to break.

There are some topics that we wish to have covered. For example, the applications of information theory outside the communication settings such as Gambling and Stock Market, and the rate distortion theory.