Channel Capacity

The Characterization of the Channel Capacity

Radu T. Trîmbiţaş

UBB

November 2012

Outline I

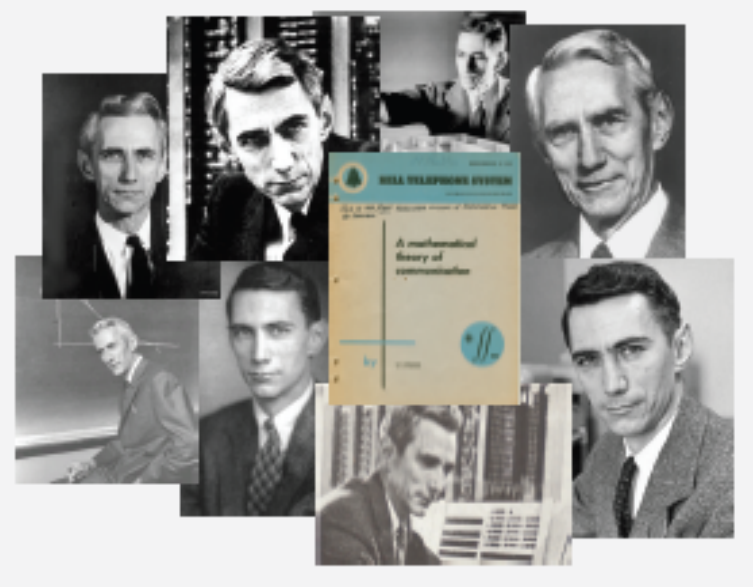
Outline II

Towards Channels I

- So far, we have been talking about compression. I.e., we have some source $p(x)$ with information $H(X)$ (the limits of compression) and the goal is to compress it down to H bits per source symbol in a representation Y.
- In some sense, the compressed representation has a "capacity" which is the total amount of bits that can be represented. I.e., with n bits, we can obviously represent no more than n bits of information.
- Compression can be seen as a process where we want to fully utilize the capacity in the compressed representation, i.e., if we have n bits of code word, we ideally (i.e., in an perfect compression scheme) would like there to be no less than n bits of information being represented. Recall efficiency: $H(X) = E(\ell)$.
- We now want to transmit information over a channel.

Towards Channels II

- From Claude Shannon:
  The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have meaning . . . [which is] irrelevant to the engineering problem



- Is there a limit to the rate of communication over a channel?
- If the channel is noisy, can we achieve (essentially) perfect error-free transmission at a reasonable rate?

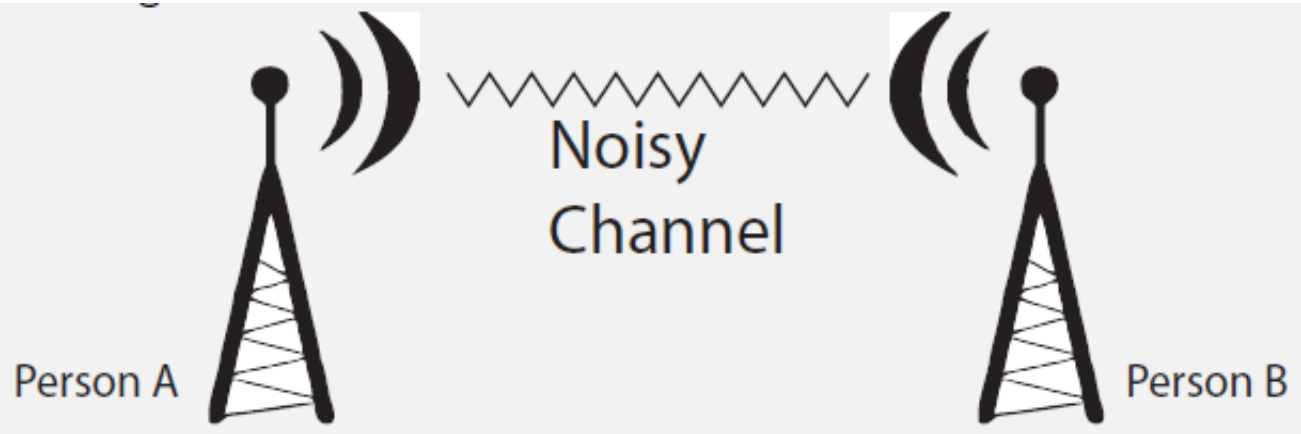
The 1930s America

- Stock-market crash of 1929
- The great depression
- Terrible conditions in textile and mining industries
- deflated crop prices, soil depletion, farm mechanization
- The rise of fascism and the Nazi state in Europe.
- Analog radio, and urgent need for secure, precise, and efficient communications
- Radio communication, noise always in data transmission (except for Morse code, which is slow).

From Wikipedia

Radio Communications I

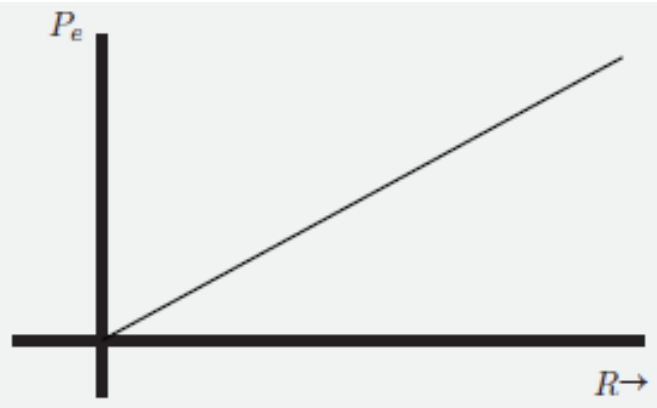
- Place yourself back in the 1930s.
- Analog communication model of the 1930s.



- Q: Can we achieve perfect communication with an imperfect communication channel?
- Q: Is there an upper bound on the information capable of being sent under different noise conditions?

Radio Communications II

- Key: If we increase the transmission rate over a noisy channel will the error rate increase?
- Perhaps the only way to achieve error free communication is to have a rate of zero.
- The error profile we might expect to see is the following:



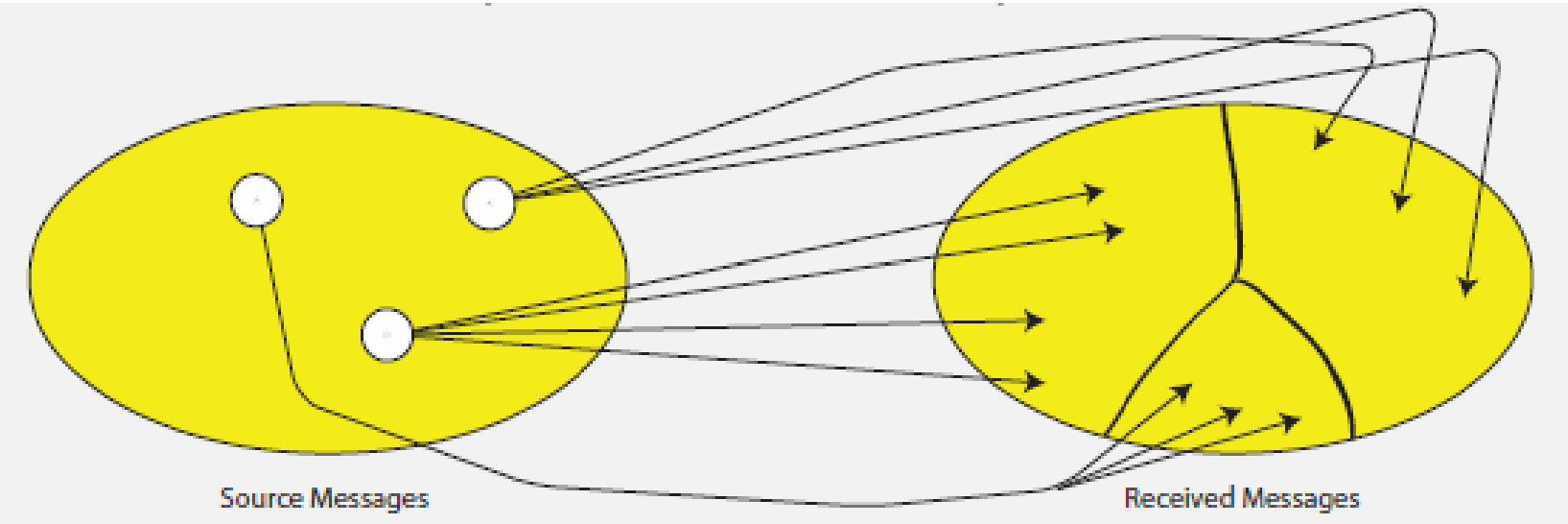
- Here, probability of error P_e goes up linearly with the rate R, with an intercept at zero.
- This was the prevailing wisdom at the time. Shannon was critical in changing that.

Simple Example

- Consider representing a signal by a sequence of numbers.
- We now know that any signal (either inherently discrete or continuous, under the right conditions) can be perfectly represented (or at least arbitrarily well) by a sequence of discrete numbers, and they can even be binary digits.
- Now consider speaking such a sequence over a noisy AM channel.
- Very possible one number will be masked by noise.
- In such case, each number we repeat k times, where k is sufficiently large to ensure we can "decode" the original sequence with very small probability of error.
- Rate of this code decreases but we can communicate reliably even if the channel is very noisy.
- Compare this idea to the figure on the following page.

A key idea

- If we choose the messages carefully at the sender, then with very high probability, they will be uniquely identifiable at the receiver.
- The idea is that we choose the source messages that (tend to) not have any ambiguity (or have any overlap) at the receiver end. I.e.,



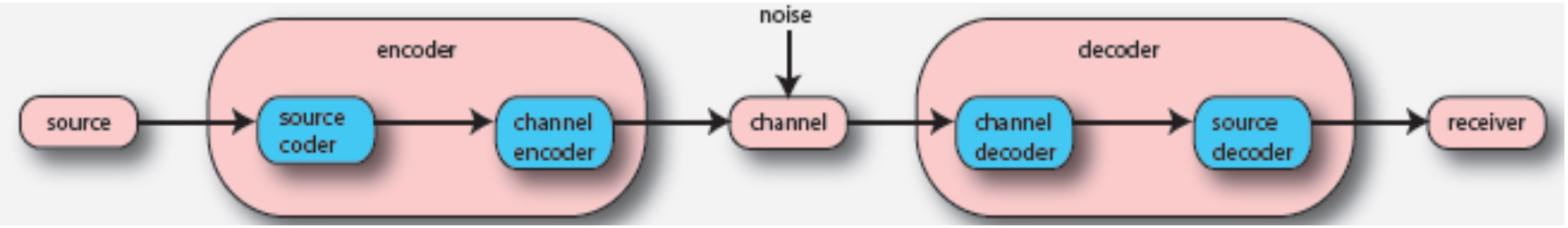
- This might restrict our possible set of source messages (in some cases severely, and thereby decrease our rate R), but if any message received in a region corresponds to only one source message, "perfect" communication can be achieved.

Discrete Channels

Definition 1

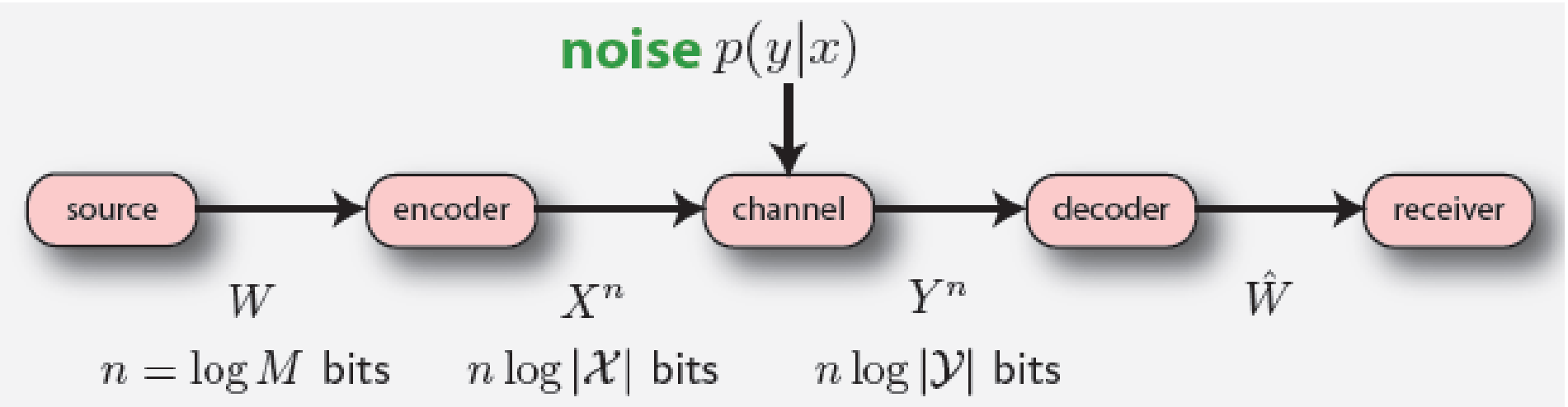
A *discrete channel* is a system consisting of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, and a distribution (probability transition matrix) $p(y|x)$ which is the probability of observing output symbol y given that we send the symbol x. A discrete channel is *memoryless* if the probability distribution of y_t, the output at time t, depends only on the input x_t and is independent of all previous inputs $x_{1:t-1}$.

- We will see many instances of discrete memoryless channels (or just DMC).
- Recall back from lecture 1 our general model of communications:



Model of Communication

- *Source* message W, one of M messages.
- *Encoder* transforms this into a length-n string of source symbols X^n
- *Noisy channel* distorts this message into a length-n string of receiver symbols Y^n.
- *Decoder* attempts to reconstruct original message as best as possible, comes up with $\widehat{W}$, one of M possible sent messages.



Rates and Capacities I

- So we have a source X governed by $p(x)$ and channel that transforms X symbols to Y symbols and which is governed by the conditional distribution $p(y|x)$
- These two items $p(x)$ and $p(y|x)$ are sufficient to compute the mutual information between X and Y. That is, we compute

$$I(X;Y) = I_{p(x)}(X,Y) = \sum_{x,y} \underbrace{p(x)p(y|x)}_{p(x,y)} \log \frac{p(y|x)}{p(y)} \tag{1}$$

$$= \sum_{x,y} p(x)p(y|x) \log \frac{p(y|x)}{\sum_{x'} p(y|x')p(x')} \tag{2}$$

- We write this as $I(X;Y) = I_{p(x)}(X;Y)$, meaning implicitly the MI quantity is a function of the entire distribution $p(x)$, for a given fixed channel $p(y|x)$.

Rates and Capacities II

- We will often be optimizing over the input distribution $p(x)$ for a given fixed channel $p(y|x)$.

Definition 2 (Information flow)

The rate of *information flow* through a channel is given by $I(X;Y)$, the mutual information between X and Y, in units of *bits per channel use*.

Definition 3 (Capacity)

The *information capacity of a channel* is the maximum information flow

$$C = \max_{p(x)} I(X;Y), \tag{3}$$

where the maximum is taken over all possible input distributions $p(x)$.

Rates and Capacities III

Definition 4 (Rate)

The *rate* R of a code is measured in the number of bits per channel use.

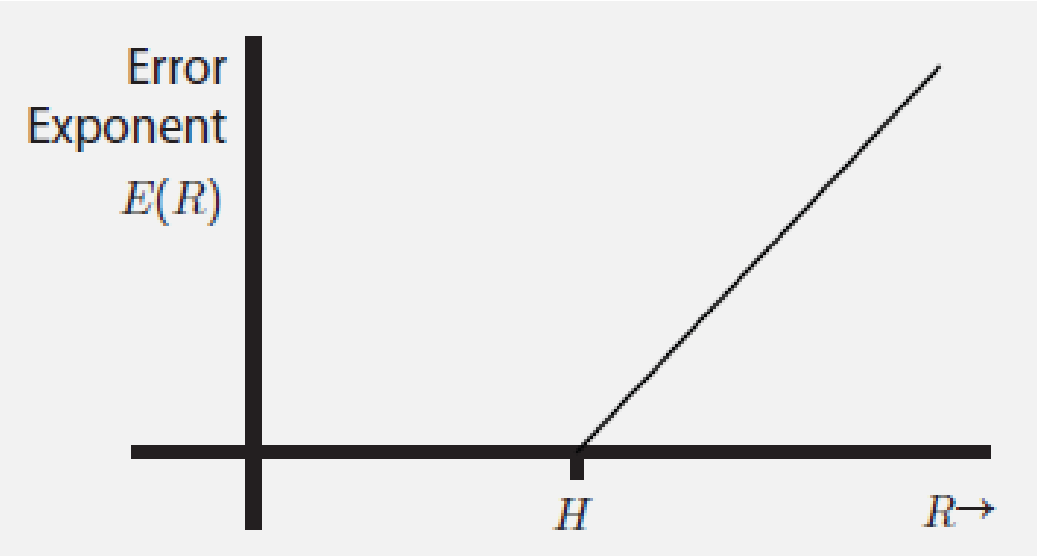
- We shall soon give an operational definition of channel capacity as the highest rate in bits per channel use at which information can be sent with arbitrarily low probability of error.
- Shannon's second theorem establishes that the information channel capacity is equal to the operational channel capacity.
- There is a duality between the problems of data compression and data transmission.
  - During compression, we remove all the redundancy in the data to form the most compressed version possible
  - During data transmission, we add redundancy in a controlled fashion to combat errors in the channel.

Rates and Capacities IV

- We show that a general communication system can be broken into two parts and that the problems of data compression and data transmission can be considered separately.

Fundamental Limits of Compression

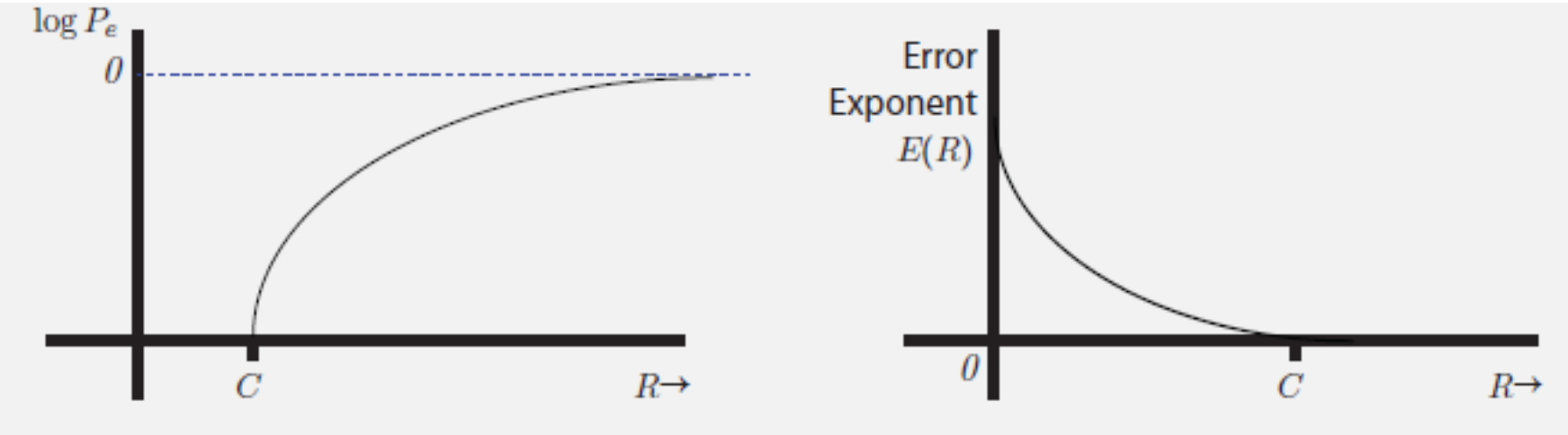
- For compression, if error exponent is positive, then error $\to 0$ exponentially fast as block length $\to 1$. Note, $P_e \sim e^{-nE(R)}$.
- That is,



- Only hope of reducing error was if $R > H$. Something "funny" happens at the entropy rate of the source distribution. Can't compress below this without incurring error.

Fundamental Limits of Data Transmission/Communication

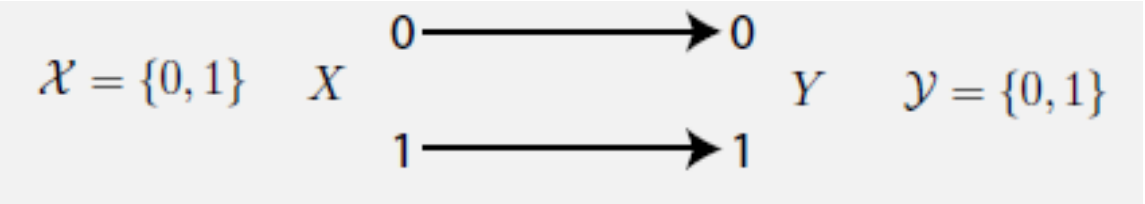
- For communication, lower bound on probability of error becomes bounded away from 0 as the rate of the code R goes above a fundamental quantity C. Note, $P_e \sim e^{-nE(R)}$.
- That is,



- only possible way to get low error is if $R < C$. Something funny happens at the point C, the capacity of the channel.
- Note that C is not 0, so can still achieve "perfect" communication over a noisy channel as long as $R < C$.

Noiseless Binary Channel

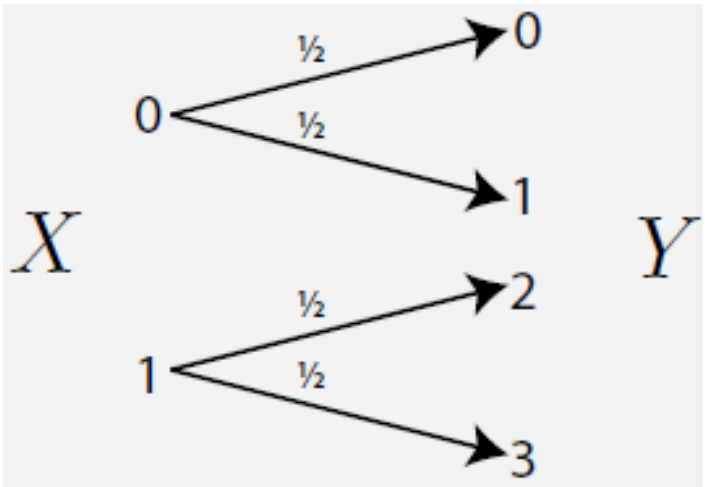
- Noiseless binary channel, diagram shows $p(y|x)$



- So, $p(y = 0|x = 0) = 1 = 1 - p(y = 1|x = 0)$ and $p(y = 1|x = 1) = 1 = 1 - p(y = 0|x = 1)$, so channel is just an input copy.
- One bit sent at a time is received without error, so capacity should be 1 bit (intuitively, we can reliably send one bit per channel usage).
- $I(X; Y) = H(X) - H(X|Y) = H(X)$ in this case, so $C = \max_{p(x)} I(X; Y) = \max_{p(x)} H(X) = 1$.
- Clearly, $p(0) = p(1) = 1/2$ achieves this capacity.
- Also, $p(0) = 1 = 1 - p(1)$ has $I(X; Y) = 0$, so achieves zero information flow.

Noisy Channel with non-overlapping outputs

This channel has 2 possible outputs corresponding to each of the 2 inputs. The channel appears to be noisy, but really is not.



- Here, $p(y = 0|x = 0) = p(y = 1|x = 0) = 1/2$ and $p(y = 2|y = 1) = p(y = 3|x = 1) = 1/2$.
- If we receive a 0 or 1, we know 0 was sent. If we receive a 2 or 3, a 1 was sent.
- Thus, $C = 1$ since only two possible error free messages.
- Same argument applies

$$I(X;Y) = H(X) - \underbrace{H(Y|X)}_{=0} = H(X)$$

- Again uniform distribution $p(0) = p(1) = 1/2$ achieves the capacity.

Permutation Channel

- Here, $p(Y = 1|X = 0) = p(Y = 0|X = 1) = 1$.
- So output is a binary permutation (swap) of input.
- Thus, $C = 1$; no information lost.

- In general, for alphabet of size $k = |\mathcal{X}| = |\mathcal{Y}|$, let be a permutation, so that $Y = \sigma(X)$.
- Then $C = \log k$.

Asside: on the optimization to compute the value C

- To maximize a given function $f(x)$, it is sufficient to show that $f(x) \leq \alpha$ for all x, and then find an x^* such that $f(x^*) = \alpha$.
- We'll be doing this over the next few slides when we want to compute $C = \max_{p(x)} I(X;Y)$ for fixed $p(y|x)$.
- The solution $p^*(x)$ that we find that achieves this maximum won't necessarily be unique.
- Also, the solution $p^*(x)$ that we find won't necessarily be the one that we end up, say, using when we wish to do channel coding.
- Right now C is just the result of a given optimization.
- We'll see that C, as computed, is also the critical point for being able to channel code with vanishingly small error probability.
- The resulting $p^*(x)$ that we obtain as part of the optimization in order to compute C won't necessarily be the one that we use for actual coding (example forthcoming).

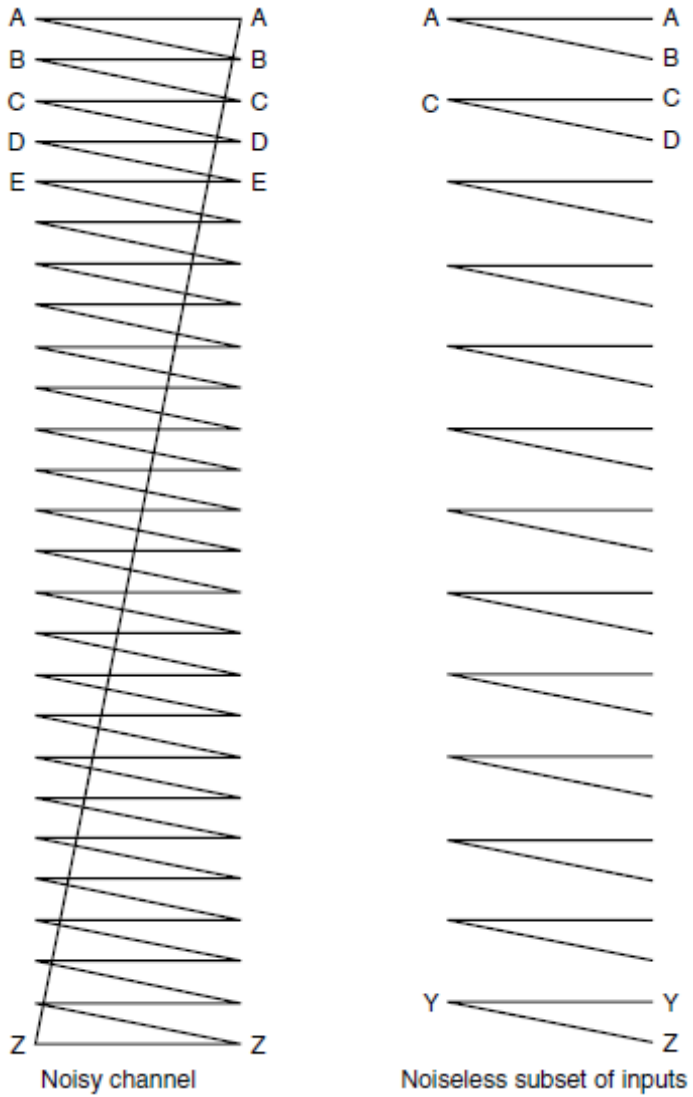


Figure: Noisy Typewriter, $C = \log 13$ bits

Noisy Typewriter I

- In this case the channel input is either received unchanged at the output with probability $1/2$ or is transformed into the next letter with probability $1/2$ (Figure 1).
- So 26 input symbols, and each symbol maps probabilistically to itself or its lexicographic neighbor.
- I.e., $p(A \to A) = p(A \to B) = 1/2$, etc.
- Each symbol always has some chance of error, so how can we communicate without error?
- Choose subset of symbols that can be uniquely disambiguated on receiver side. Choose every other source symbol, A, C, E, etc.
- Thus $A \to \{A; B\}$, $C \to \{C; D\}$, $E \to \{E; F\}$, etc. so that each received symbols has only one unique source symbol.
- Capacity $C = \log 13$
- Q: what happens to C when probabilities are not all $1/2$?

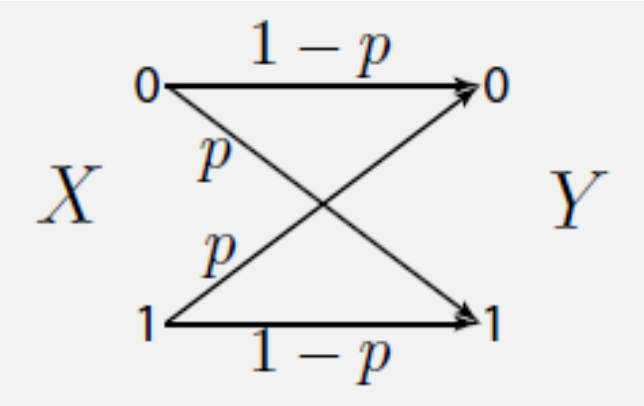
Noisy Typewriter II

- We can also compute the capacity more mathematically.
- For example:

$$\begin{aligned} C &= \max_{p(x)} I(X;Y) = \max_{p(x)} \left(H(X) - H(Y|X)\right) \\ &= \max_{p(x)} H(Y) - 1 \qquad //\text{for } X = x \;\exists\; 2 \text{ choices} \\ &= \log 26 - 1 = \log 13 \end{aligned}$$

- The $\max_{p(x)} H(Y) = \log 26$ can be achieved by using the uniform distribution for $p(x)$, for which when we choose any x symbol, there is equal likelihood of two Ys being received.
- An alternatively $p(x)$ would put zero probability on the alternates (B, D, F, etc.). In this case, we still have $H(Y) = \log 26$
- So the capacity is the same in each case ($\exists$ two $p(x)$ that achieved this) but only one is what we would use, say, for error free coding.

Binary Symmetric Channel (BSC) I



- A bit that is sent will be flipped with probability p.
- $p(Y=1|X=0) = p = 1 - p(Y=0|X=0)$. $p(Y=0|X=1) = p = p(Y=1|X=1)$.

- BSC is the simplest model of channel with errors, yet it captures most of the complexity of the general problem
- Q: can we still achieve reliable ("guaranteed" error free) communication with this channel? A: Yes, if $p < 1/2$ and if we do not ask for too high a transmission rate (which would be $R > C$), then we can. Actually, any $p \neq 1/2$ is sufficient.
- Intuition: think about AEP and/or block coding.

Binary Symmetric Channel (BSC) II

- But how to compute C, the capacity?

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) = H(Y) - \sum p(x) H(Y|X=x) \\ &= H(Y) - \sum p(x) H(p) = H(Y) - H(p) \leq 1 - H(p). \end{aligned}$$

- When is $H(Y) = 1$? Note that

$$\begin{aligned} P(Y=1) &= P(Y=1|X=1)P(X=1) \\ +P(Y=1|X=0)P(X=0) & \\ &= (1-p)P(X=1) + pP(X=1) \\ &= P(X=1) \end{aligned}$$

- So $H(Y) = 1$ if $H(X) = 1$.
- Thus, we get that $C = 1 - H(p)$ which happens when X is uniform.
- If $p = 1/2$ then $C = 0$, so if it randomly flips bits, then no information can be sent.

Binary Symmetric Channel (BSC) III

- If $p \neq 1/2$, then we can communicate, albeit potentially slowly. E.g., if $p = 0.499$ then $C = 2.8854 \times 10^{-6}$ bits per channel use. So to send one bit, need to use the channel quite a bit.
- If $p = 0$ or $p = 1$, then $C = 1$ and we can get maximum possible rate (i.e., the capacity is one bit per channel use).

Decoding

- Lets temporarily look ahead to address this problem.
- We can "decode" the source using the received string, source distribution, and the channel model $p(y|x)$ via Bayes rule. I.e.

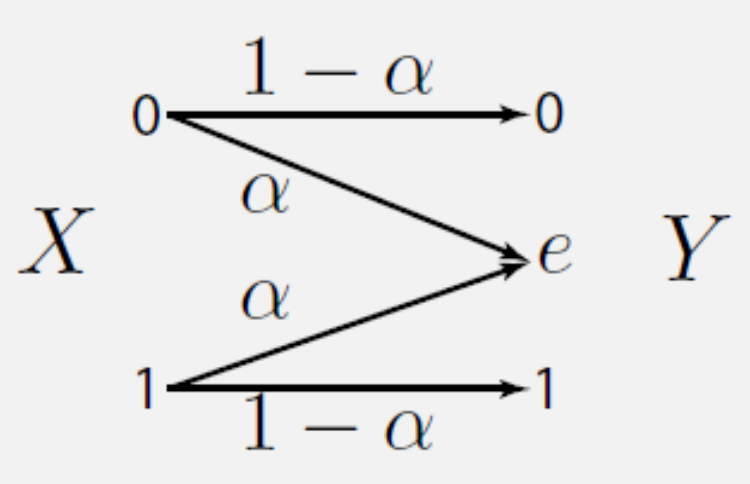
$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} = \frac{P(Y|X)P(X)}{\sum_{x'} P(Y|X' = x')Pr(X' = x')}$$

- If we get a particular y, we can compute $p(x|y)$ and make a decision based on that. I.e., $\widehat{x} = argmax_x p(x|y)$.
- This is optimal decoding in that it minimizes the error.
- Error if $x \neq \widehat{x}$, and $P(error) = P(x \neq \widehat{x})$.
- This is minimal if we chose $argmax_x p(x|y)$ since the error $1 - P(\widehat{x}|y)$ is minimal.

Minimum Error Decoding

- Note: Computing quantities such as $P(x|y)$ is a task of probabilistic inference.
- Often this problem is difficult (NP-hard). This means that doing minimum error decoding might very well be exponentially expensive (unless $P = NP$).
- Many real world codes are such that computing the exact computation must be approximated (i.e., no known fast algorithm for minimum error or maximum likelihood decoding).
- Instead we do approximate inference algorithms (e.g., loopy belief propagation, message passing, etc.). These algorithms tend still to work very well in practice (achieve close to the capacity C).
- But before doing that, we need first to study more channels and the theoretical properties of the capacity C.

Binary Erasure Channel I



- e is an erasure symbol, if that happens we don't have access to the transmitted bit.
- The probability of dropping a bit is then α.
- We want to compute capacity. Obviously, $C = 1$ if $\alpha = 0$.

$$C = \max_{p(x)} I(X, Y) = \max_{p(x)} \left(H(Y) - H(X|Y)\right)$$
$$= \max_{p(x)} H(Y) - H(\alpha).$$

- So while $H(Y) \leq \log 3$, we want actual value of the capacity.

Binary Erasure Channel II

- let $E = \{Y = e\}$. Then

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

- Let $\pi = P(X = 1)$. Then

$$H(Y) = H\left(\overbrace{(1-\pi)(1-\alpha)}^{\text{if } Y=0}, \overbrace{\alpha}^{\text{if } Y=e}, \overbrace{\pi(1-\alpha)}^{\text{if } Y=1}\right)$$
$$= H(\alpha) + (1-\alpha)H(\pi).$$

- This last equality follows since $H(E) = H(\alpha)$, and

$$H(Y|E) = H(Y|Y = e) + (1-\alpha)H(Y|Y \neq e) = \alpha \cdot 0 + (1-\alpha)H(\pi).$$

Binary Erasure Channel III

- Then we get

$$\begin{aligned} C &= \max_{p(x)} H(Y) - H(\alpha) \\ &= \max_{\pi} \left((1-\alpha) H(\pi) + H(\alpha) \right) - H(\alpha) \\ &= \max_{\pi} (1-\alpha) H(\pi) = 1 - \alpha \end{aligned}$$

- Best capacity when $\pi = 1/2 = P(X = 1) = P(X = 0)$.
- This makes sense, loose $\alpha\%$ of the bits of original capacity.

Symmetric Channels I

Definition 5

A channel is *symmetric* if rows of the channel transition matrix $p(y|x)$ are permutations of each other, and columns of this matrix are permutations of each other. A channel is *weakly symmetric* if every row of the transition matrix $p(.|x)$ is a permutation of every other row, and all column sums $\sum_x p(y|x)$ are equal.

Theorem 6

For a weakly symmetric channel,

$$C = \log |\mathcal{Y}| - H(r) \tag{4}$$

where r is the row of transition matrix. This is achieved by a uniform distribution on the input alphabet.

Symmetric Channels II

Proof.

$$I(X;Y) = H(Y) - H(Y|X) = H(Y) - H(r) \leq \log|Y| - H(r) \tag{5}$$

with equality if the output distribution is uniform. But $p(x) = 1/|\mathcal{X}|$ achieves a uniform distribution on Y, as seen from

$$p(y) = \sum_{x\in\mathcal{X}} p(y|x)p(x) = \frac{1}{|\mathcal{X}|}\sum p(y|x) = c\frac{1}{|\mathcal{X}|} = \frac{1}{|\mathcal{Y}|}, \tag{6}$$

where c is the sum of the entries in one column of the transition matrix. □

Symmetric Channels III

Example 7

The channel with transition matrix

$$p(y|x) = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.5 & 0.3 & 0.2 \\ 0.2 & 0.5 & 0.3 \end{bmatrix}$$

is symmetric. Its capacity is
$C = \log 3 - H(0.5, 0.2, 0.3) \ = 8.8818 \times 10^{-16}$.

Example 8

$Y = X + Z \pmod{c}$, where $\mathcal{X} = \mathcal{Z} = \{0, 1, \ldots, c-1\}$, and X and Z are independent.

Symmetric Channels IV

Example 9

The channel with transition matrix

$$p(y|x) = \begin{bmatrix} \frac{1}{3} & \frac{1}{6} & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{6} \end{bmatrix}$$

is weakly symmetric, but not symmetric.

Properties of Channel Capacity

1. $C \geq 0$ since $I(X;Y) \geq 0$.
2. $C \leq \log |\mathcal{X}|$ since $C = \max_{p(x)} I(X;Y) \leq \max H(X) = \log |\mathcal{X}|$.
3. $C \leq \log |\mathcal{Y}|$ for the same reason. Thus, the alphabet sizes limit the transmission rate.
4. $I(X;Y) = I_{p(x)}(X;Y)$ is a continuous function of $p(x)$.
5. $I(X;Y)$ is a concave function of $p(x)$ for fixed $p(y|x)$.
6. Thus, $I_{\lambda p_1 + (1-\lambda)p_2}(X;Y) \geq \lambda I_{p_1}(X;Y) + (1-\lambda) I_{p_2}(X;Y)$. Interestingly, since concave, this makes computing something like the capacity easier. I.e., a local maximum is a global maximum, and computing the capacity for a general channel model is a convex optimization procedure.
7. Recall also, $I(X;Y)$ is a convex function of $p(y|x)$ for fixed $p(x)$.

Shannon's 2nd Theorem I

- One of the most important theorems of the last century.
- We'll see it in various forms, but we state it here somewhat informally to start acquiring intuition.

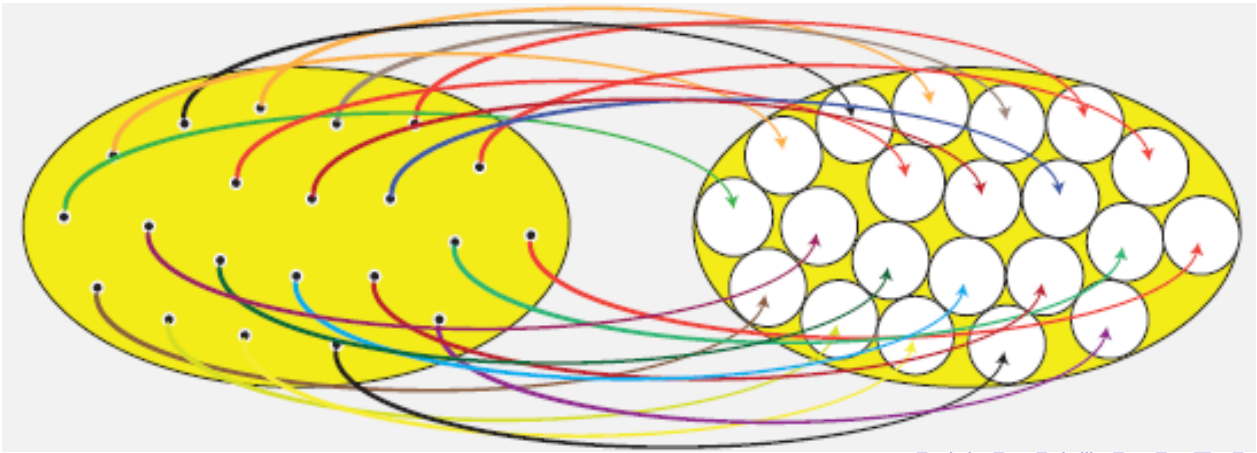
Theorem 10 (Shannon's 2nd Theorem)

C is the maximum number of bits (on average, per channel use) that we can transmit over a channel reliably.

- Here, "reliably" means with vanishingly small and exponentially decreasing probability of error as the block length gets longer. We can easily make this probability essentially zero.
- Conversely, if we try to push $> C$ bits through the channel, error quickly goes to 1.

Shannon's 2nd Theorem II

- Intuition of this we've already seen in the noisy typewriter and the region partitioning.
- Slightly more precisely, this is a sort of bin packing problem.
- We've got a region of possible codewords, and we pack as many smaller non-overlapping bins into the region as possible.
- The smaller bins correspond to the noise in the channel, and the packing problem depends on the underlying "shape"
- Not really a partition, since there might be wasted space, also depending on the bin and region shapes.



Shannon's 2nd Theorem III

- Intuitive idea: use typicality argument.
- There are $\approx 2^{nH(X)}$ typical sequences, each with probability $2^{-nH(X)}$ and with $p(A_\varepsilon^{(n)}) \approx 1$, so the only thing with "any" probability is the typical set and it has all the probability.
- The same thing is true for conditional entropy.
- That is, for a typical input X, there are $2^{nH(Y|X)}$ output sequences.
- Overall, there are $2^{nH(Y)}$ typical output sequences, and we know that $2^{nH(Y)} \geq 2^{nH(Y|X)}$.

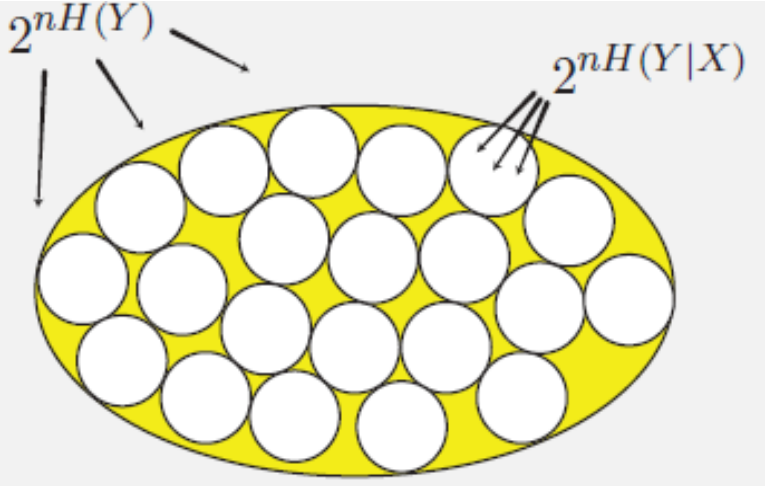
Shannon's 2nd Theorem: Intuition I

- Goal: find a non-confusable subset of the inputs that produce disjoint output sequences (as in picture).
- There are $\approx 2^{nH(Y)}$ (typical) outputs (i.e., the marginally typical Y sequences).
- There are $2^{nH(Y|X)}$ (X-conditionally typical Y sequences) outputs. $\equiv$ the average possible number of outputs for a possible input, so this many could be confused with each other. I.e., on average, for a given $X = x$, this is approximately how many outputs there might be.
- So the number of non-confusable inputs is

$$\leq \frac{2^{nH(Y)}}{2^{nH(Y|X)}} = 2^{n(H(Y)-H(Y|X))} = 2^{nI(X;Y)} \tag{7}$$

Shannon's 2nd Theorem: Intuition II

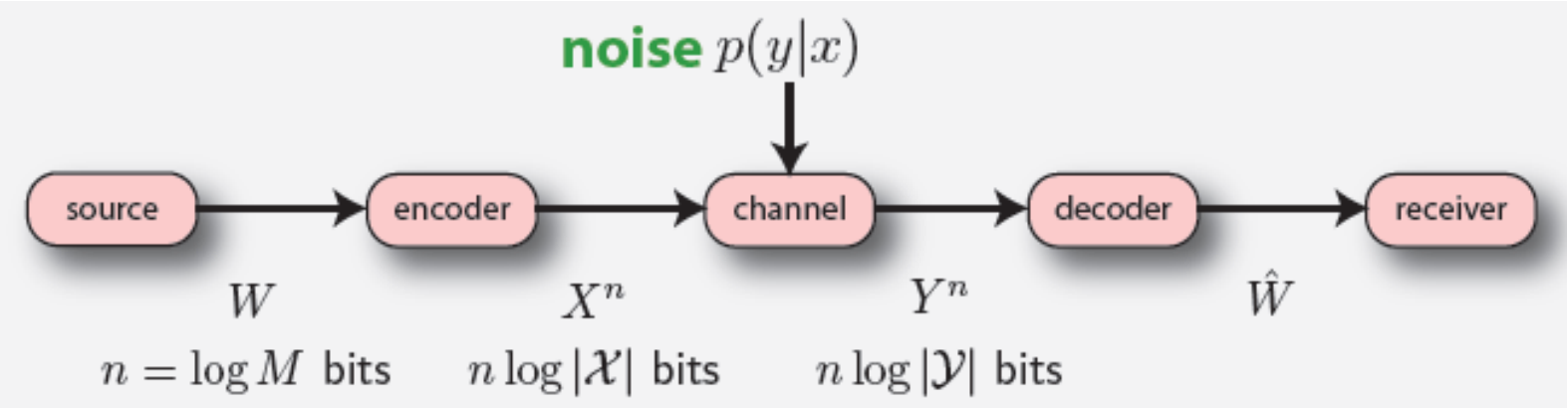
- Note, in non-ideal case, there could be overlap of the typical Y-given-X sequences, but the best we can do (in terms of maximizing the number of non-confusable inputs) is when there is no overlap on the output. This is assumed in the above.
- We can view the number of non-confusable inputs (7) as a volume. $2^{nH(Y)}$ is the total number of possible slots, while $2^{nH(Y|X)}$ is the number of slots taken up (on average) for a given source word. Thus, the number of source words that can be used is the ratio.



Shannon's 2nd Theorem: Intuition III

- To maximize the number of non-confusable inputs (7), for a fixed channel $p(y|x)$, we find the best $p(x)$ which gives $I(X;Y) = C$, which is the log of the maximum number of inputs possible to use.
- This is the capacity.

Definitions I



Definitions 11

- *Message* $W \in \{1, \ldots, M\}$ requiring $\log M$ bits per message.
- *Signal* sent through channel $X^n(W)$, a random codeword.
- *Received signal* from channel $Y \sim p(y^n|x^n)$
- *Decoding* via guess $\widehat{W} = g(Y^n)$.
- *Discrete memoryless channel (DMC)* $(\mathcal{X}; p(y|x); \mathcal{Y})$

Definitions II

Definitions 12

- *n-th extension to channel* is $(\mathcal{X}^n; p(y^n|x^n); \mathcal{Y}^n)$, where

$$p\left(y_k|x^k, y^{k-1}\right) = p(x_k|y_k)$$

- *Feedback* if x_k can use both previous inputs and outputs.
- No feedback if $p(x_k|x_{1:k-1}; y_{1:k-1}) = p(x_k|x_{1:k-1})$. We'll analyze feedback a bit later.

Remark. If the channel is used without feedback, the channel transition function reduces to

$$p\left(y^n|x^n\right) = \prod_{i=1}^{n} p(y_i|x_i). \tag{8}$$

(M,n) code

Definition 13 ((M, n) code)

An (M, n) *code* for channel $(\mathcal{X}; p(y|x); \mathcal{Y})$ is

1. An index set $\{1, 2, \ldots, M\}$.
2. An encoding function $X^n : \{1, 2, \ldots, M\} \to \mathcal{X}^n$ yielding codewords $X^n(1)$, $X^n(2)$, $X^n(3)$, $\ldots$, $X^n(M)$. Each source message has a codeword, and each codeword is n code symbols.
3. Decoding function, i.e., $g : \mathcal{Y}^n \to \{1, 2, \ldots, M\}$ which makes a "guess" about original message given channel output.

Remark. In an (M, n) code, $M =$ the number of possible messages to be sent, and $n =$ number of channel uses by the codewords of the code.

Error I

Definition 14 (Probability of Error λ_i for message $i \in \{1, \ldots, M\}$)

$$\lambda_i := P\left(g(Y^n) \neq i | X^n = x^n(i)\right) = \sum_{y^n} p\left(y^n | x^n(i)\right) I\left(g\left(y^n\right) \neq i\right) \quad (9)$$

$I(\cdot)$ is the indicator function; the conditional probability of error given that index i was sent.

Definition 15 (Max probability of Error $\lambda^{(n)}$ for (M, n) code)

$$\lambda^{(n)} := \max_{i \in \{1,2,\ldots,M\}} \lambda_i. \quad (10)$$

Error II

Definition 16 (Average probability of error $P_e^{(n)}$)

$$P_e^{(n)} = \frac{1}{M}\sum_{i=1}^{M}\lambda_i = P(Z \neq g(Y^n))$$

where Z is a r.v. with probability $P(Z = i) \sim U(M)$ (discrete uniform)

$$P_e^{(n)} = E(I(Z \neq g(Y^n))) = \sum_{i=1}^{M} P\left(g(Y^n) \neq i | X^n = X^n(i)\right) p(i),$$

where $p(i) = \frac{1}{M}$.

Remark. A key Shannon's result is that a small average probability of error means we must have a small maximum probability of error!

Rate

Definition 17

The *rate* R of an (M, n) code is

$$R = \frac{\log M}{n} \qquad \text{bits per transmission}$$

Definition 18

A rate R is *achievable* if there exists a sequence of $\left(\left\lceil 2^{nR} \right\rceil, n\right)$ codes such that $\lambda^{(n)} \to 0$ as $n \to \infty$.

Remark. To simplify the notation we write $\left(2^{nR}, n\right)$ instead $\left(\left\lceil 2^{nR} \right\rceil, n\right)$.

Capacity

Definition 19

The *capacity* of a channel is the supremum of all achievable rates.

- So the capacity of a DMC is the rate beyond which the error won't tend to zero with increasing n.
- Note: this is a different notion of capacity that we encountered before.
- Before we defined $C = \max_{p(x)} I(X, Y)$.
- Here we are defining something called the "capacity of a DMC".
- We have not yet compared the two (but of course we will).

Jointly Typical Sequences

Definition 20

The set of jointly typical sequences with respect to the distribution $p(x, y)$ is defined by

$$A_\varepsilon^{(n)} = \{(x^n, y^n) \in \mathcal{X}^n \times \mathcal{Y}^n :$$

$$\left| -\frac{1}{n} \log p(x^n) - H(X) \right| < \varepsilon \tag{11}$$

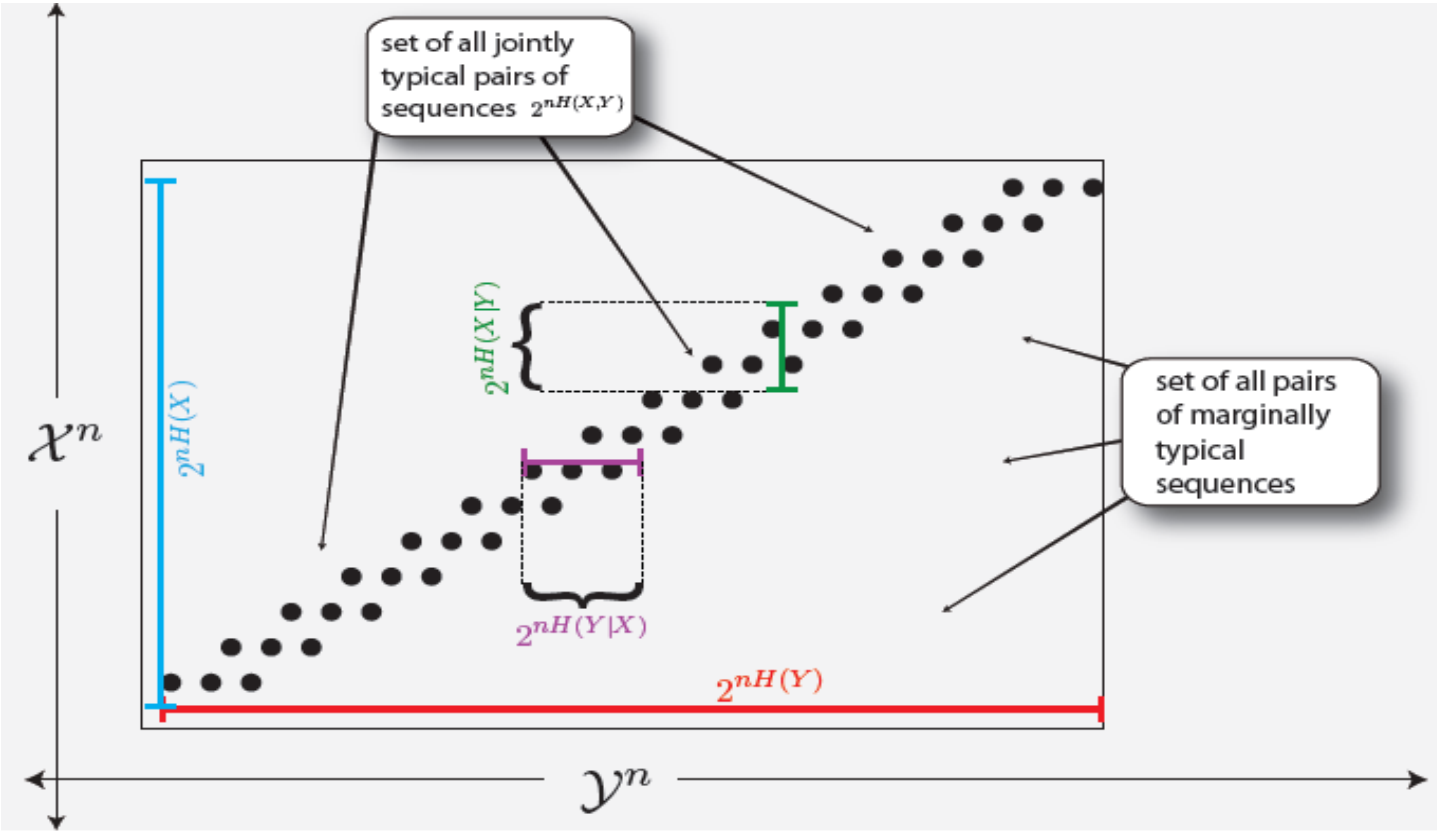
$$\left| -\frac{1}{n} \log p(y^n) - H(Y) \right| < \varepsilon \tag{12}$$

$$\left| -\frac{1}{n} \log p(x^n, y^n) - H(X, Y) \right| < \varepsilon \Bigg\}, \tag{13}$$

where

$$p(x^n, y^n) = \prod_{i=1}^{n} p(x_i, y_i). \tag{14}$$

Jointly Typical Sequences: Picture



Intuition for Jointly Typical Sequences

- So intuitively,

$$\frac{\text{num. jointly typical seqs.}}{\text{num ind. chosen typical seqs.}} = \frac{2^{nH(X,Y)}}{2^{nH(X)}2^{nH(Y)}}$$
$$= 2^{n(H(X,Y)-H(X)-H(Y))}$$
$$= 2^{-nI(X,Y)}$$

- So if we independently at random choose two (singly) typical sequences for X and Y, then the chance that it will be an (X, Y) jointly typical sequence decreases exponentially with n, as long as $I(X, Y) > 0$.
- to decrease this chance as much as possible, it can become 2^{-nC}.

Theorem 21 (Joint AEP)

Let $(X^n, Y^n) \sim p(x^n, y^n) = \prod_{i=1}^n p(x_i, y_i)$ *i.i.d. Then:*

1. $P\left((X^n, Y^n) \in A_\varepsilon^{(n)}\right) \to 1$ *as* $n \to \infty$.
2. $(1-\varepsilon)2^{n(H(X,Y)-\varepsilon)} \leq \left|A_\varepsilon^{(n)}\right| \leq 2^{n(H(X,Y)+\varepsilon)}$
3. *If* $(\widetilde{X}^n, \widetilde{Y}^n) \sim p(x^n)p(y^n)$, *are drawn independently, then*

$$P\left((\widetilde{X}^n, \widetilde{Y}^n) \in A_\varepsilon^{(n)}\right) \leq 2^{-n(I(X;Y)-3\varepsilon)}. \tag{15}$$

Also, for sufficiently large n,

$$P\left((\widetilde{X}^n, \widetilde{Y}^n) \in A_\varepsilon^{(n)}\right) \geq (1-\varepsilon)\, 2^{-n(I(X;Y)+3\varepsilon)} \tag{16}$$

Key property: we have bound on the probability of independently drawn sequences being jointly typical, falls off exponentially fast with n, if $I(X;Y) > 0$.

Joint AEP proof

Proof of $P\left((X^n, Y^n) \in A_\varepsilon^{(n)}\right) \to 1$.

- We have, by the w.l.l.n.s,

$$-\frac{1}{n}\log P\left(X^n\right) \to -E\left(\log P(X)\right) = H(X) \tag{17}$$

so $\forall \varepsilon > 0\ \exists m_1$ such that for $n > m_1$

$$P\left(\underbrace{\left|-\frac{1}{n}\log P(X^n) - H(X)\right| > \varepsilon}_{S_1}\right) < \frac{\varepsilon}{3} \tag{18}$$

- So, S_1 is a non-typical event.

. . .

Joint AEP proof

Proof of $P\left((X^n, Y^n) \in A_\varepsilon^{(n)}\right) \to 1$.

- Also $\exists m_2, m_3$ such that $\forall n > m_2$ we have

$$P\left(\underbrace{\left|-\frac{1}{n}\log P(Y^n) - H(Y)\right| > \varepsilon}_{S_2}\right) < \frac{\varepsilon}{3} \tag{19}$$

and $\forall n > m_3$ we have

$$P\left(\underbrace{\left|-\frac{1}{n}\log P(X^n, Y^n) - H(X, Y)\right| > \varepsilon}_{S_3}\right) < \frac{\varepsilon}{3} \tag{20}$$

- So all events S_1, S_2 and S_3 are non-typical events.

Joint AEP proof

Proof of $P\left((X^n, Y^n) \in A_\varepsilon^{(n)}\right) \to 1$.

- For $n > \max(m_1, m_2, m_3)$, we have that

$$P\left(S_1 \cup S_2 \cup S_3\right) \leq 3\varepsilon/3$$

- So, non-typicality has probability $< \varepsilon$, meaning $P(\overline{A_\varepsilon^{(n)}}) \leq \varepsilon$ giving $P(A_\varepsilon^{(n)}) \geq 1 - \varepsilon$, as desired.

. . .

Joint AEP proof

Proof of $(1-\varepsilon)2^{n(H(X,Y)-\varepsilon)} \leq \left|A_\varepsilon^{(n)}\right| \leq 2^{n(H(X,Y)+\varepsilon)}$.

- We have

$$1 = \sum_{x^n,y^n} p(x^n,y^n) \geq \sum_{(x^n,y^n)\in A_\varepsilon^{(n)}} p(x^n,y^n) \geq \left|A_\varepsilon^{(n)}\right| 2^{-n(H(X,Y)+\varepsilon)}$$

$$\Longrightarrow \left|A_\varepsilon^{(n)}\right| \leq 2^{n(H(X,Y)+\varepsilon)}$$

- Also, $P\left(A_\varepsilon^{(n)}\right) > 1-\varepsilon$,

$$1-\varepsilon \leq \sum_{(x^n,y^n)\in A_\varepsilon^{(n)}} p(x^n,y^n) \leq \left|A_\varepsilon^{(n)}\right| 2^{-n(H(X,Y)-\varepsilon)}$$

$$\Longrightarrow \left|A_\varepsilon^{(n)}\right| \geq (1-\varepsilon)2^{n(H(X,Y)-\varepsilon)}$$

. . .

Joint AEP proof

Proof of two indep. sequences are likely not jointly typical.

We have the following two derivations:

$$
\begin{aligned}
P\left((\widetilde{X}^n, \widetilde{Y}^n)\right) &= \sum_{(x^n,y^n)\in A_\varepsilon^{(n)}} p(x^n)p(y^n) \\
&\leq 2^{n(H(X,Y)+\varepsilon)}2^{-n(H(X)-\varepsilon)}2^{-n(H(Y)-\varepsilon)} \\
&= 2^{-n(I(X;Y)-3\varepsilon)} \\
P\left((\widetilde{X}^n, \widetilde{Y}^n)\right) &\geq (1-\varepsilon)2^{n(H(X,Y)-\varepsilon)}2^{-n(H(X)+\varepsilon)}2^{-n(H(Y)+\varepsilon)} \\
&= (1-\varepsilon)2^{-n(I(X;Y)+3\varepsilon)}.
\end{aligned}
$$

More Intuition I

- There are $\approx 2^{nH(X)}$ typical X sequences
- There are $\approx 2^{nH(Y)}$ typical Y sequences.
- The total number of independent typical pairs is $\approx 2^{nH(X)}2^{nH(Y)}$, but not all of them are jointly typical. Rather only $\approx 2^{nH(X;Y)}$ of them are jointly typical.
- The fraction of independent typical sequences that are jointly typical is:

$$\frac{2^{nH(X,Y)}}{2^{nH(X)}2^{nH(Y)}} = 2^{n(H(X,Y)-H(X)-H(Y))} = 2^{-nI(X,Y)}$$

  and this is essentially the probability that a randomly chosen pair of (marginally) typical sequences is jointly typical.
- So if we use typicality to decode (which we will) then there are about $2^{nI(X;Y)}$ pairs of sequences before we start using pairs that will be jointly typical and chosen randomly.

More Intuition II

- Ex: if $p(x) = 1/M$ then we can choose about M samples before we see a given x, on average.

Channel Coding Theorem (Shannon 1948[2])

- The basic idea is to use joint typicality.
- Given a received codeword y^n, find an x^n that is jointly typical with y^n.
- This x^n will occur jointly with y^n with probability 1, for large enough n.
- Also, the probability that some other $\widehat{x^n}$ is jointly typical with y^n is about $2^{-nI(X;Y)}$,
- so if we use $< 2nI(X;Y)$ codewords, then some other sequence being jointly typical will occur with vanishingly small probability for large n.

Channel Coding Theorem (Shannon 1948): more formally

Theorem 22 (Channel coding theorem)

For a discrete memoryless channel, all rates below capacity C are achievable. Specifically, for every rate $R < C$, there exists a sequence of $\left(2^{nR}, n\right)$ codes with maximum probability of error $\lambda^{(n)} \to 0$ as $n \to \infty$. Conversely, any $\left(2^{nR}, n\right)$ sequence of codes with $\lambda^{(n)} \to 0$ as $n \to \infty$ must have that $R < C$.

Channel Theorem I

- Implications: as long as we do not code above capacity we can, for all intents and purposes, code with zero error.
- This is true for all noisy channels representable under this model. We're talking about discrete channels now, but we generalize this to continuous channels in the coming lectures.
- We could look at error for a particular code and bound its errors.
- Instead, we look at the average probability of errors of all codes generated randomly.
- We then prove that this average error is small.
- This implies $\exists$ many good codes to make the average small.
- To show that the maximum probability of error also small, we throw away the worst 50% of the codes.
- Recall: idea is, for a given channel $(X, p(y|x), Y)$ come up with a $(2^{nR}, n)$ code of rate R which means we need:

Channel Theorem II

1. Index set $\{1, \ldots, M\}$
2. Encoder: $X^n : \{1, \ldots, M\} \to \mathcal{X}^n$ maps to codewords $X^n(i)$
3. Decoder: $g : \mathcal{Y}^n \to \{1, \ldots, M\}$.

- Two parts to prove: 1) all rates $R < C$ are achievable (exists a code with vanishing error). Conversely, 2) if the error goes to zero, then must have $R < C$.

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

- Given $R < C$, assume use of $p(x)$ and generate 2^{nR} random codewords using $p(x^n) = \prod_{i=1}^{n} p(xi)$.
- Choose $p(x)$ arbitrarily for now, and then change it later to get C.
- set of random codewords (the codebook) can be seen as a matrix:

$$\mathcal{C} = \begin{bmatrix} x_1(1) & x_2(1) & \cdots & x_n(1) \\ \vdots & \vdots & \ddots & \vdots \\ x_1\left(2^{nR}\right) & x_2\left(2^{nR}\right) & \cdots & x_n\left(2^{nR}\right) \end{bmatrix} \tag{21}$$

- So, there are 2^{nR} codes each of length n generated via $p(x)$.
- To send any message $\omega \in \left\{1, 2, \ldots, M = 2^{nR}\right\}$, we send codeword $x_{1:n}(\omega) = \left\{x_1(\omega), x_2(\omega), \ldots, x_n(\omega)\right\}$.

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

- Can compute probabilities of a given codeword for ω ...

$$P\left(x^n\left(\omega\right)\right) = \prod_{i=1}^{n} p(x_i(\omega)), \qquad \omega \in \{1, 2, \ldots, M\}.$$

- . . . or even the entire codebook:

$$P(C) = \prod_{\omega=1}^{2^{nR}} \prod_{i=1}^{n} p(x_i(\omega)) \tag{22}$$

...

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

Consider the following encoding/decoding scheme:

1. Generate a random codebook as above according to $p(x)$
2. Codebook known to both sender/receiver (who also knows $p(y|x)$).
3. Generate messages W according to the uniform distribution (we'll see why shortly), $P(W = \omega) = 2^{-nR}$, for $\omega = 1, \ldots, 2^{nR}$.
4. Send $X^n(\omega)$ over the channel.
5. Receiver receives Y^n according to distribution

$$Y^n \sim P\left(y^n | x^n(\omega)\right) = \prod_{i=1}^{n} p\left(y_i | x_i\left(\omega\right)\right). \tag{23}$$

6. The signal is decoded using typical set decoding (to be described).

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

Typical set decoding: Decode message as $\hat{\omega}$ if

1. $(x^n(\hat{\omega}), y^n)$ is jointly typical
2. $\nexists\ k$ such that $(x^n(k), y^n) \in A_\varepsilon^{(n)}$ (i.e., $\hat{\omega}$ is unique)

Otherwise output special invalid integer "0" (error). Three types of errors might occur (type A, B, or C).

A. $\exists k \neq \hat{\omega}$ such that. $(x^n(k), y^n) \in A_\varepsilon^{(n)}$ (i.e., > 1 possible typical message).

B. no $\hat{\omega}$ s.t. $(x^n(\hat{\omega}), y^n)$ is jointly typical.

C. if $\hat{\omega} \neq \omega$, i.e., wrong codeword is jointly typical.

Note: maximum likelihood decoding is optimal, but typical set decoding is not, but this will be good enough to show the result. ...

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

three types of quality measures we might be interested in.

1. Code specific error

$$P_e^{(n)}(\mathcal{C}) = P(\hat{\omega} \neq \omega | \mathcal{C}) = \frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} \lambda_i$$

where

$$\lambda_i = P(g(y^n) \neq i | X^n = x^n(i)) = \sum_{y^n} p\left(y^n | x^n(i)\right) I(g(y^n) \neq i)$$

but we would like something easier to analyze.

2. Average error over all randomly generated codes (avg. of avg.)

$$P(\mathcal{E}) = \sum_{\mathcal{C}} Pr(\mathcal{C}) Pr(\widehat{W} \neq W | \mathcal{C}) = \sum_{\mathcal{C}} P(\mathcal{C}) P_e(\mathcal{C})$$

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

three types of quality measures we might be interested in.

3. Max error of the code, ultimately what we want to use

$$P_{\mathcal{C},\max}(\mathcal{C}) = \max_{i \in \{1,\ldots,M\}} \lambda_i$$

We want to show that if $R < C$, then exists a codebook $\mathcal{C}$ s.t. this error $\to 0$ (and that if $R > C$ error must $\to 1$).

Our method is to:

1. Expand average error (bullet 2 above) and show that it is small.
2. deduce that $\exists$ at least one code with small error
3. show that this can be modified to have small maximum probability of error.

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

$$P(\mathcal{E}) = \sum_{\mathcal{C}} P(\mathcal{C}) P_e^{(n)}(\mathcal{C}) = \sum_{\mathcal{C}} P(\mathcal{C}) \frac{1}{2^{nR}} \sum_{\omega=1}^{2^{nR}} \lambda_\omega(\mathcal{C}) \tag{24}$$

$$= \frac{1}{2^{nR}} \sum_{\omega=1}^{2^{nR}} \sum_{\mathcal{C}} P(\mathcal{C}) \lambda_\omega(\mathcal{C}) \tag{25}$$

but

$$\sum_{\mathcal{C}} P(\mathcal{C}) \lambda_w(\mathcal{C}) = \sum_{\mathcal{C}} \underbrace{P\left(g\left(Y^n\right) \neq \omega | X^n = x^n(\omega)\right) \overbrace{P\left(x^n(1), \ldots, x^n\left(2^{nR}\right)\right)}^{\prod_{i=1}^{2^{nR}} P(x^n(i))}}_{T}$$

$$= \sum_{x^n(1), \ldots, x^n(2^{nR})} T$$

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

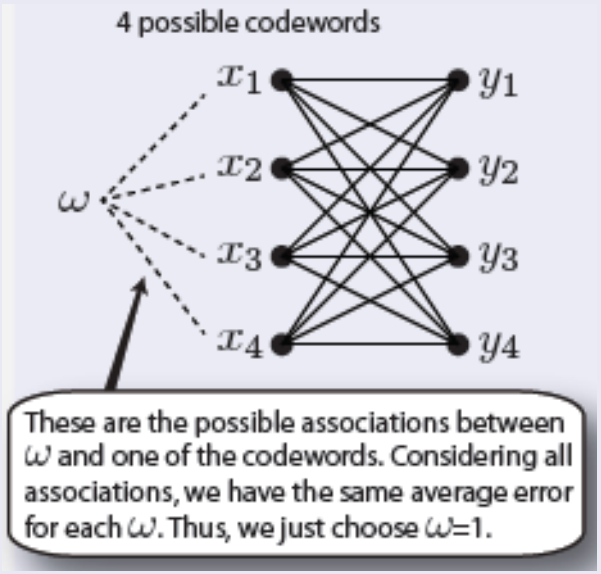
$$
\begin{aligned}
&P(\mathcal{C})\lambda_\omega(\mathcal{C}) \\
&= \underbrace{\sum \prod_{i \neq \omega} P(x^n(i))}_{1} \sum_{x^n(\omega)} P\left(g\left(Y^n\right) \neq \omega | X^n = x^n(\omega)\right) P\left(x^n(\omega)\right) \\
&= \sum_{x^n(\omega)} P\left(g\left(Y^n\right) \neq \omega | X^n = x^n(\omega)\right) P\left(x^n(\omega)\right) \\
&= \sum_{x^n(\omega)} P\left(g\left(Y^n\right) \neq 1 | X^n = x^n(1)\right) P\left(x^n(1)\right) = \sum_{\mathcal{C}} P(\mathcal{C})\lambda_1\left(\mathcal{C}\right) = \beta
\end{aligned}
$$

Last sum is the same regardless of ω, call it β. Thus, we can can arbitrarily assume that $\omega = 1$.

...

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.



So error is equal to:
prob. of choosing x_1 for ω and not choosing y_1
+prob. of choosing x_2 for ω and not choosing y_2
$+\ldots$
this is just the same for all $\omega \in \{1, \ldots M\}$ so we may just pick $\omega = 1$

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

So we get

$$P(\mathcal{E}) = \sum_{\mathcal{C}} P(\mathcal{C}) P_e^{(n)}(\mathcal{C}) = \frac{1}{2^{nR}} \sum_{\omega=1}^{2^{nR}} \beta = \sum_{\mathcal{C}} P(\mathcal{C}) \lambda_1(\mathcal{C}) = P(\mathcal{E}|W=1)$$

- Next, define the random events (again considering $\omega = 1$):

$$E_i := \left\{ (X^n(i), Y^n) \in A_\varepsilon^{(n)} \right\}, \qquad i \in \left\{ 1, 2, \ldots, 2^{nR} \right\}.$$

- Assume that input is $x^n(1)$ (i.e., first message sent).
- Then the no error event is the same as: $E_1 \cap \overline{(E_2 \cup E_3 \cup \cdots \cup E_M)} = E_1 \cap (\overline{E}_2 \cap \overline{E}_3 \cap \cdots \cap \overline{E}_M)$.

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

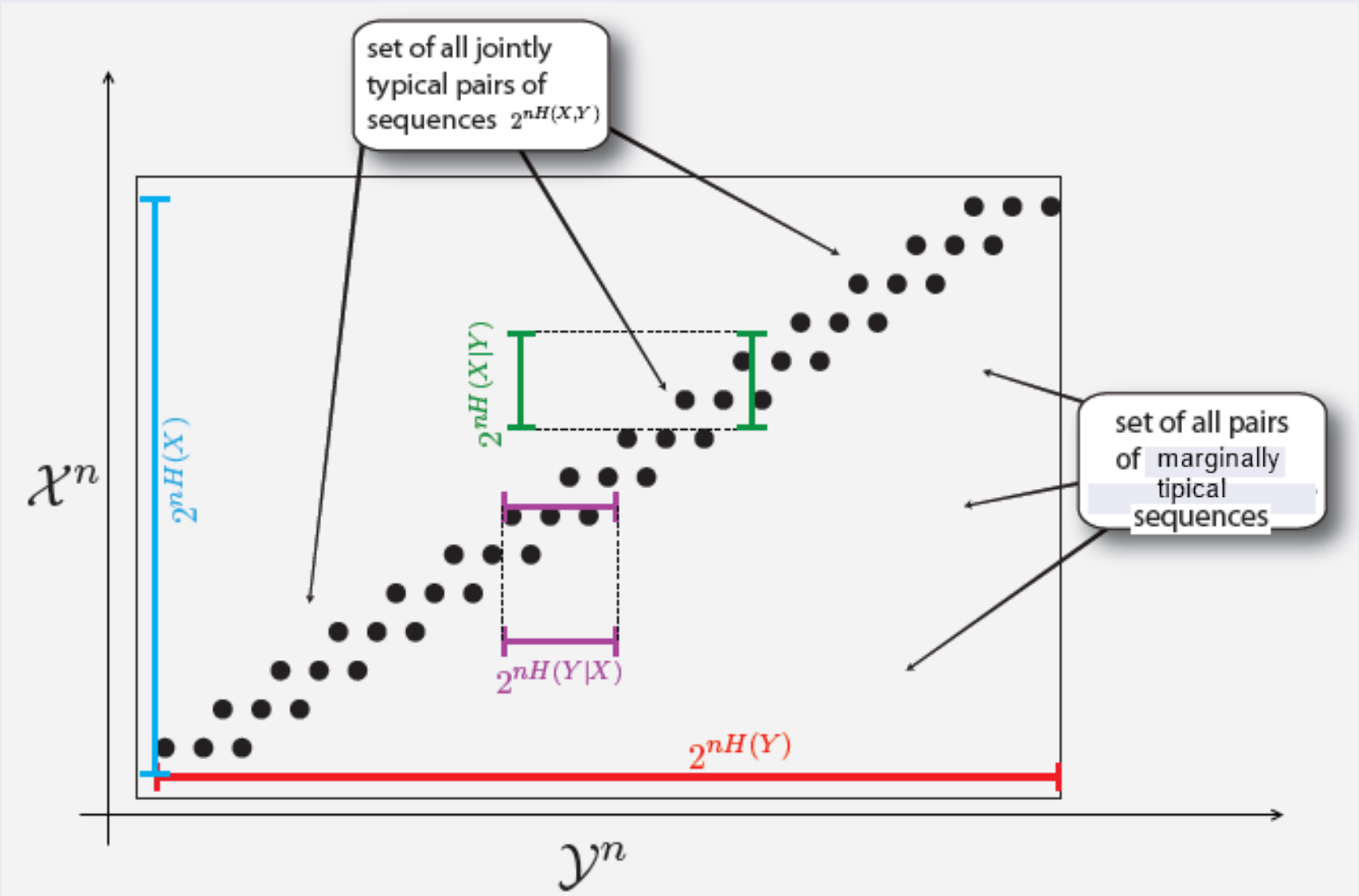
Various flavors of error

- E_1 means that the transmitted and received codeword are not jointly typical (this is error type B from before).
- $E_2 \cup E_3 \cup \cdots \cup E_{2^{nR}}$. This is either:
  - Type C: wrong codeword is jointly typical with received sequence
  - Type A: greater than 1 codeword is jointly typical with received sequence

so this is type C and A both.
Our goal is to bound the probability of error, but lets look at some figures first. ...

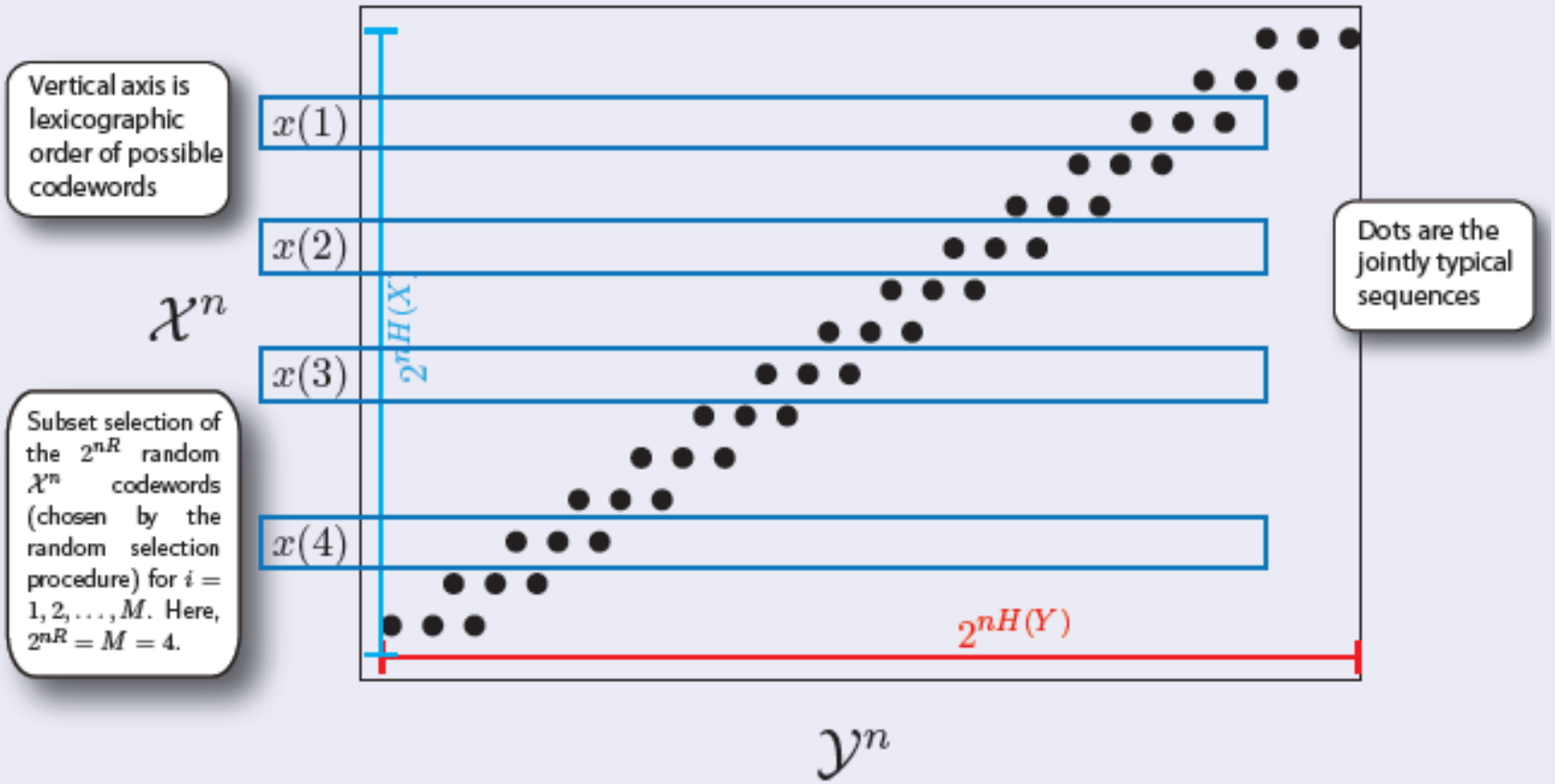
All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.



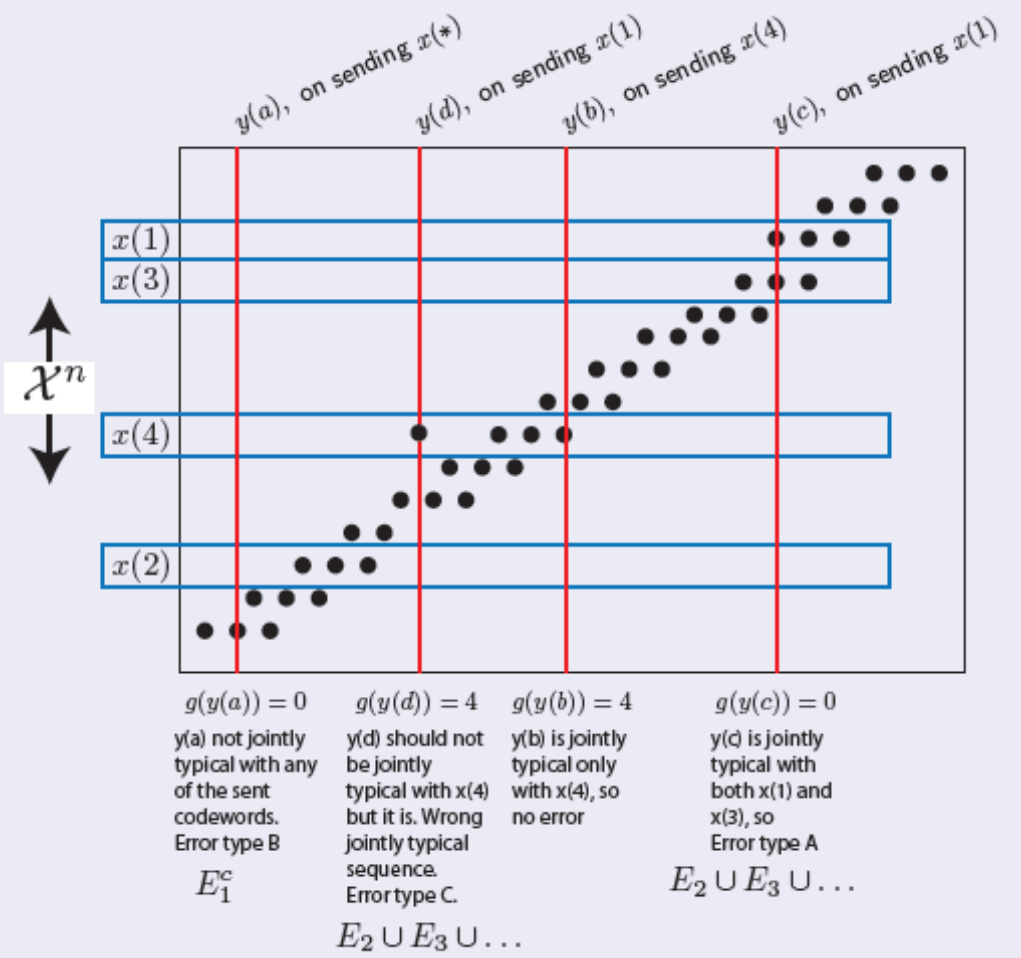
All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.



All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.



All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

Goal: bound the probability of error:

$$P(\mathcal{E}|W=1) = P\left(\overline{E}_1 \cup E_2 \cup E_3 \cup \dots\right)$$
$$\leq P\left(\overline{E}_1\right) + \sum_{i=2}^{2^{nR}} P(E_i)$$

We have that

$$P\left(\overline{E}_1\right) = P\left(\overline{A_\varepsilon^{(n)}}\right) \to 0 \qquad (n \to \infty)$$

So, $\forall \varepsilon > 0 \exists n_0$ such that

$$P\left(\overline{E}_1\right) \leq \varepsilon, \ \forall n > n_0$$

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

- Also, because of random code generation process (and recall, $\omega = 1$), $X^n(1)$ and $X^n(i)$ i.r.v., hence $X^n(i)$ and Y^n i.r.v for $i \neq 1$
- This gives, for $i \neq 1$,

$$P\left(\underbrace{(X^n(i), Y^n) \in A_\varepsilon^{(n)}}_{\textit{indep. events}}\right) \leq 2^{-n(I(X;Y)-3\varepsilon)}$$

  by the joint AEP.
- This will allow us to bound the error, as long as $I(X;Y) > 3\varepsilon$.

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

Consequently,

$$
\begin{aligned}
P(\mathcal{E}) = P(\mathcal{E}|W=1) &\leq P\left(\overline{E}_1|W=1\right) + \sum_{i=2}^{2^{nR}} P(E_i|W=1) \\
&\leq \varepsilon + \sum_{i=2}^{2^{nR}} 2^{-n(I(X;Y)-3\varepsilon)} \\
&= \varepsilon + \left(2^{nR}-1\right) 2^{-n(I(X;Y)-3\varepsilon)} \\
&\leq \varepsilon + 2^{3n\varepsilon} 2^{-n(I(X;Y)-R)} \\
&= \varepsilon + 2^{-n[(I(X;Y)-3\varepsilon)-R]} \\
&\leq 2\varepsilon
\end{aligned}
$$

The last statement is true only if $I(X;Y) - 3 > R$. ...

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

- So if we chose $R < I(X;Y)$ (strictly), we can find an ε and n so that the average probability of error $P(\mathcal{E}) \leq 2\varepsilon$, can be made as small as we want.
- But, we need to get from an average to a max probability of error, and bound that.
- First, choose $p(x) = argmax_{p(x)} I(X;Y)$ rather than uniform $p(x)$, to change the condition from $R < I(X;Y)$ to $R < C$. Thus, this gives us higher rate limit.
- If $P(\mathcal{E}) \leq 2\varepsilon$, the bound on the average error is small, so there must exist some specific code, say $\mathcal{C}^*$ s.t.

$$P_e^{(n)}(\mathcal{C}^*) \leq 2\varepsilon.$$

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

- Lets break apart this error probability.

$$P_e^{(n)}(\mathcal{C}^*) = \frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} \lambda_i(\mathcal{C}^*)$$
$$= \frac{1}{2^{nR}} \sum_{i:\lambda_i < 4\varepsilon} \lambda_i(\mathcal{C}^*) + \frac{1}{2^{nR}} \sum_{i:\lambda_i \geq 4\varepsilon} \lambda_i(\mathcal{C}^*)$$

- Now suppose more than half of the indices had error $\geq 4\varepsilon$ (i.e., suppose $|\{i : \lambda_i \geq 4\varepsilon\}| \geq 2^{nR}/2 = 2^{nR-1}$). Under this assumption

$$\frac{1}{2^{nR}} \sum_{i:\lambda_i \geq 4\varepsilon} \lambda_i(\mathcal{C}^*) \geq \frac{1}{2^{nR}} \sum_{i:\lambda_i \geq 4\varepsilon} 4\varepsilon = \frac{1}{2^{nR}} |\{i : \lambda_i \geq 4\varepsilon\}| 4\varepsilon > 2\varepsilon.$$

. . .

All rates $R < C$ are achievable

Proof that all rates $R < C$ are achievable.

- Can't be since these alone would be more than our 2ε upper bound.
- Hence, at most half the codewords can have error $\geq 4\varepsilon$, and we get

$$|\{i : \lambda_i \geq 4\varepsilon\}| \geq 2^{nR}/2 \Longrightarrow |\{i : \lambda_i < 4\varepsilon\}| \geq 2^{nR}/2$$

- Create a new codebook that eliminates all bad codewords (i.e., those in with index $\{i : \lambda_i \geq 4\varepsilon\}$). There are at most half of them.
- The remaining codewords are of size $\geq 2^{nR}/2 = 2^{nR-1} = 2^{n(R-1/n)}$(at least half of them). They all have max probability $\leq 4\varepsilon$.
- We now code with rate $R\prime = R - 1/n \to R$ as $n \to \infty$, but for this new sequence of codes, the max error probability (n) 4 , which can be made as small as we wish.

Discussion I

- To summarize, random coding is the method of proof to show that if $R < C$, there exists a sequence of $(2^{nR}, n)$ codes with $\lambda^{(n)} \to 0$ as $n \to \infty$.
- This might not be the best code, but it is sufficient. It is an existence proof.
- Huge literature on coding theory. We'll discuss Hamming codes.
- But many good codes exist today: Turbo codes, Gallager (or low-density-parity-check) codes, and new ones are being proposed often.
- But we have yet to prove the converse ...
- We next need to show that any sequence of $(2^{nR}, n)$ codes with $(2^{nR}, n)$ must have that $R < C$.
- First lets consider the case if $P_e^{(n)} = 0$, in such case it is easy to show that $R < C$.

Zero-Error Codes I

- $P_e^{(n)} = 0 \Longrightarrow H(W|Y^n) = 0$ (no uncertainty)
- For the sake of an easy proof, assume $H(W) = nR = \log M$ (i.e., uniform distribution over $\{1, 2, \ldots, M\}$.
- First lets consider the case if $P_e^{(n)} = 0$, in such case it is easy to show that $R < C$. Then we get

$$\begin{aligned} nR = H(W) &= H(W|Y^n) + I(W; Y^n) = I(W; Y^n) \\ &\leq I(X^n; Y^n) \qquad // \text{ Since } W \to X^n \to Y^n \text{ Markov chain and DP in} \\ &\leq \sum_{i=1}^{n} I(X_i; Y_i) \qquad //\text{Fano's lemma, follows next} \\ &\leq nC \qquad //\text{definition of capacity} \end{aligned}$$

- Hence

$$R \leq C.$$

Fano's Lemmas I

Lemma 23

For a DMC with a codebook $\mathcal{C}$ and the input message W uniformly distributed over 2^{nR}

$$H(W|\widehat{W}) \leq 1 + P_e^{(n)} nR \tag{26}$$

Proof.

W uniformly distributed $\Longrightarrow P_e^{(n)} = P(W \neq \widehat{W})$. We apply Fano's inequality

$$1 + P_e \log |\mathcal{X}| \geq H(X|Y);$$

for W and an alphabet of length 2^{nR}. □

Next lemma shows that the capacity per transmission is not increased if we use a DMC many times.

Fano's Lemmas II

Lemma 24

Let Y^n be the result of passing X^n through a memoryless channel of capacity C. Then

$$I\left(X^n; Y^n\right) \leq nC, \qquad \forall p(x^n). \tag{27}$$

Fano's Lemmas III

Proof.

$$
\begin{aligned}
I\left(X^n; Y^n\right) &= H\left(Y^n\right) - H\left(Y^n | X^n\right) \\
&= H\left(Y^n\right) - \sum_{i=1}^{n} H(Y_i | Y_1, \ldots, Y_{i-1}, X^n) \\
&= H\left(Y^n\right) - \sum_{i=1}^{n} H(Y_i | X_i) \qquad //\text{def. of DMC}
\end{aligned}
$$

. . .

Proof - continuation.

$$
\begin{aligned}
I\left(X^n; Y^n\right) &= H\left(Y^n\right) - \sum_{i=1}^{n} H(Y_i|X_i) \\
&\leq \sum_{i=1}^{n} H(Y_i) - \sum_{i=1}^{n} H(Y_i|X_i) \\
&= \sum_{i=1}^{n} \underbrace{I(X_i; Y_i)}_{\leq C} \leq nC.
\end{aligned}
$$

Converse to the Channel Coding Theorem

The converse states: any sequence of $(2^{nR}; n)$ codes with $\lambda^{(n)} \to 0$ must have that $R < C$.

Proof that $\lambda^{(n)} \to 0$ as $n \to \infty \Rightarrow R < C$.

- Average prob. goes to zero if max probability does: $\lambda^{(n)} \to 0 \Rightarrow P_e^{(n)} \to 0$, where $P_e^{(n)} = \frac{1}{2^{nR}} \sum_{i=1}^{2^{nR}} \lambda_i$
- Set $H(W) = nR$ for now (i.e., W uniform on $\left\{1, 2, \ldots, M = 2^{nR}\right\}$). Again, makes the proof a bit easier and
- doesn't affect relationship between R and C).
- So, $P\left(\widehat{W} \neq W\right) = P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i$

. . .

Converse to the Channel Coding Theorem I

Proof that $\lambda^{(n)} \to 0$ as $n \to \infty \Rightarrow R < C$.

$$
\begin{aligned}
nR = H(W) &= H\left(W|\widehat{W}\right) + I\left(W;\widehat{W}\right) && //\text{uniformity of } W \\
&\leq 1 + P_e^{(n)} nR + I\left(W;\widehat{W}\right) && //\text{by Fano's Lemma 23} \\
&\leq 1 + P_e^{(n)} nR + I\left(X^n;Y^n\right) && //\text{DP inequality} \\
&\leq 1 + P_e^{(n)} nR + nC && //\text{Lemma 24} \\
R &\leq P_e^{(n)} R + C + \frac{1}{n} && (28)
\end{aligned}
$$

Now as $n \to \infty$, $P_e^{(n)} \to 0$, and $1/n \to 0$ as well. Thus $R < C$. ...

Proof that $\lambda^{(n)} \to 0$ as $n \to \infty \Rightarrow R < C$.

- We rewrite (28) as
$$P_e^{(n)} \geq 1 - \frac{C}{R} - \frac{1}{nR}.$$
- if $n \to \infty$ and $R > C$, then error lower bound is strictly positive, and depends on $1 - C/R$.
- Even for small n, $P_e^{(n)} > 0$, since otherwise, if $P_e^{(n0)} = 0$ for some code, we can concatenate code to get large n same rate code, contradicting $P_e > 0$.
- Hence we cannot achieve an arbitrarily low probability of error at rates above capacity

- This coverse is called *weak converse*
- *strong converse*: if $R > C$, $P_e^{(n)} \to 1$

Equality in the Converse to the Channel Coding Theorem I

What if we insist on $R = C$ and $P_e = 0$. In such case, what are the requirements of any such code.

$$
\begin{aligned}
nR &= H(W) = H(X^n(W)) \\
&= \underbrace{H(W|\widehat{W})}_{=0,\text{ since } P_e=0} + I(W;\widehat{W}) = I(W;\widehat{W}) \\
&= H(Y^n) - H(Y^n|X^n) \\
&= H(Y^n) - \sum_{i=1}^{n} (Y^n|X_i) \\
&= \sum_i H(Y_i) - \sum_i H(Y_i|X_i) \qquad //\text{if all } Y_i \text{ indep.} \\
&= \sum_i I(X_i;Y_i) \\
&= nC \qquad //p^*(x) \in \arg\max_{p(x)} I(X_i;Y_i)
\end{aligned}
$$

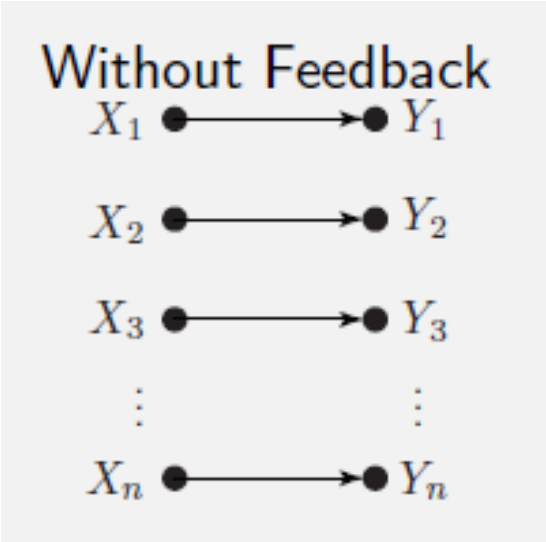
Equality in the Converse to the Channel Coding Theorem II

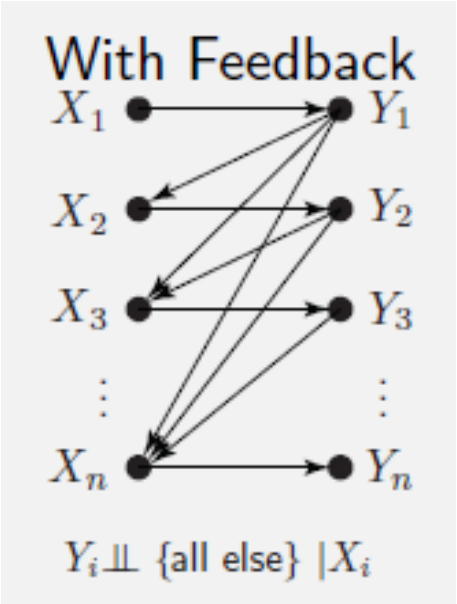
So there are 3 conditions for equality, $R = C$, namely

1. all codewords must be distinct
2. Y_i's are independent
3. distribution on x is $p^*(x)$, a capacity achieving distribution.

Feedback Capacity

Consider a sequence of channel uses.





Another way of looking at it is:

Can this help, i.e., can this increase C?

Does feedback help for DMC

- A: No.
- Intuition: w/o memory, feedback tells us nothing more than what we already know, namely $p(y|x)$.
- Can feedback made decoding easier? Yes, consider binary erasure channel, when we get $Y = e$ we just re-transmit.
- Can feedback help for channels with memory? In general, yes.

Feedback for DMC

Definition 25

A $(2^{nR}, n)$ *feedback code* is is the encoder $X_i(W; Y_{1:i-1})$, a decoder $g : \mathcal{Y}^n \to \{1, 2, \ldots, 2^{nR}\}$ and $P_e^{(n)} = P\left(g\left(Y^n\right) \neq W\right)$, for $H(W) = nR$ (uniform).

Definition 26 (Capacity with feedback)

The *capacity with feedback* C_{FB} of a DMC is the supremum of all rates achievable by feedback codes.

Theorem 27 (Feedback capacity)

$$C_{FB} = C = \max_{p(x)} I(X; Y). \tag{29}$$

Feedback codes for DMC

Proof.

- Clearly, $C_{FB} > C$, since FB code is a generalization.
- Next, we use W instead of X and bound R.
- We have

$$\begin{aligned} H(W) &= H(W|\widehat{W}) + I(W;\widehat{W}) \\ &\leq 1 + P_e^{(n)} nR + I(W;\widehat{W}) \qquad //\text{by Fano} \\ &\leq 1 + P_e^{(n)} nR + I(W;Y^n) \qquad //\text{DP ineq.} \end{aligned}$$

- We next bound $I(W;Y^n)$

. . .

Feedback codes for DMC

... proof continued.

$$I(W;Y^n) = H(Y^n) - H(Y^n|W) = H(Y^n) - \sum_{i=1}^{n} H\left(Y_i|Y_{1:i-1}, W\right)$$

$$= H(Y^n) - \sum_{i=1}^{n} H\left(Y_i|Y_{1:i-1}, W, X_i\right) \qquad //X_i = f\left(W, Y_{1:i-1}\right)$$

$$= H(Y^n) - \sum_{i=1}^{n} H\left(Y_i|X_i\right) \leq \sum_{i} \left[H(Y_i) - H\left(Y_i|X_i\right)\right]$$

$$= \sum_{i} I(X_i;Y_i) \leq nC$$

...

Feedback codes for DMC

... proof continued.

Thus we have

$$nR = H(W) \leq 1 + P_e^{(n)} nR + nC$$
$$\Longrightarrow R \leq \frac{1}{n} + P_e^{(n)} R + C \Longrightarrow R \leq C < C_{FB}.$$

Thus, feedback does not help. □

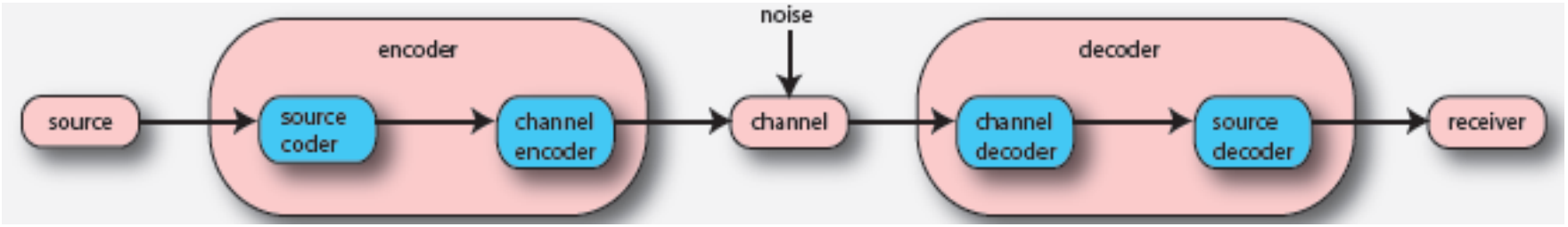
Joint Source/Channel Theorem

- **Data compression**: We now know that it is possible to achieve error free compression if our average rate of compression, R, measured in units of bits per source symbol, is such that $R > H$ where H is the entropy of the generating source distribution.
- **Data Transmission**: We now know that it is possible to achieve error free communication and transmission of information if $R < C$, where R is the average rate of information sent (units of bits per channel use), and C is the capacity of the channel.
- **Q**: Does this mean that if $H < C$, we can reliably send a source of entropy H over a channel of capacity C?
- This seems intuitively reasonable.

Joint Source/Channel Theorem: process

The process would go something as follows:

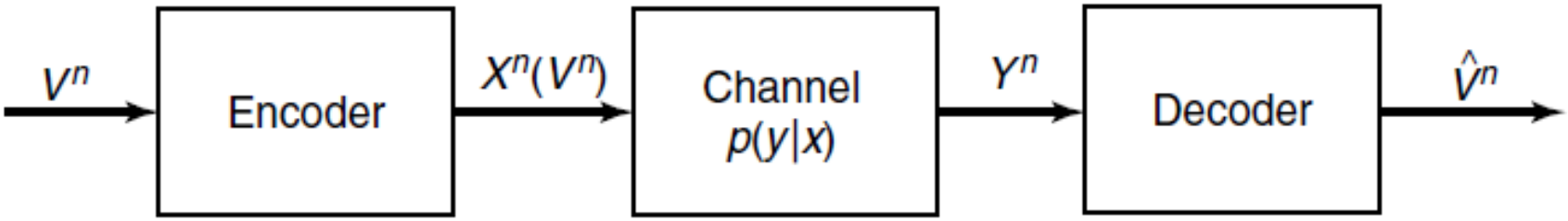
1. Compress a source down to its entropy, using Huffman, LZ, arithmetic coding, etc.
2. Transmit it over a channel.
3. If all sources could share the same channel, would be very useful.
4. I.e., perhaps the same channel coding scheme could be used regardless of the source, if the source is first compressed down to the entropy. The channel encoder/decoder need not know anything about the original source (or how to encode it).
5. Joint source/channel decoding as in the following figure:



6. Maybe obvious now, but at the time (1940s) it was a revolutionary idea!

Source/Channel Separation Theorem

- Source: $V \in \mathcal{V}$ that satisfies AEP (e.g., stationary ergodic).
- Send $V_{1:n} = V_1, V_2, \ldots, V_n$ over channel, entropy rate $H(V)$ of stochastic process (if i.i.d., $H(V) = H(V_i), \forall i$).



- Error probability and setup:

$$P_e^{(n)} = P\left(V_{1:n} \neq \widehat{V}_{1:n}\right)$$
$$= \sum_{y_{1:n}} \sum_{v_{1:n}} P(v_{1:n}) P\left(y_{1:n} | X^n(v_{1:n})\right) I\left\{g\left(y_{1:n}\right) \neq v_{1:n}\right\}$$

where I indicator function, g decoding function

Source/Channel Coding Theorem

Theorem 28 (Source-channel coding theorem)

If $V_{1:n}$ satisfies AEP and if $H(\mathcal{V}) < C$, then $\exists$ a sequence of $(2^{nR}; n)$ codes with $P_e^{(n)} \to 0$. Conversely, if $H(\mathcal{V}) > C$, then $P_e^{(n)} > 0$ for all n and cannot send the process with arbitrarily low probability of error.

Proof.

- If V satisfies AEP, then $\exists$ a set $A_\varepsilon^{(n)}$ with $\left|A_\varepsilon^{(n)}\right| < 2^{n(H(\mathcal{V})+\varepsilon)}$ ($A_\varepsilon^{(n)}$ has all the probability).
- We only encode the typical set, and signal an error otherwise. This ε contributes to P_e.
- We index elements of $A_\varepsilon^{(n)}$ as $\left\{1, 2, \ldots, 2^{n(H+\varepsilon)}\right\}$, so need $n(H+\varepsilon)$ bits.
- This gives a rate of $R = H(\mathcal{V}) + \varepsilon$. If $R < C$ then the error $< \varepsilon$, which we can make as small as we wish.

Source/Channel Coding Theorem

. . . proof continued.

- Then

$$
\begin{aligned}
P_e^{(n)} &= P(V_{1:n} \neq \widehat{V}_{1:n}) \\
&\leq P(V_{1:n} \notin A_\varepsilon^{(n)}) \underbrace{+P\left(g\left(Y^n\right) \neq V^n | V^n \in A_\varepsilon^{(n)}\right)}_{<\varepsilon, \text{ since } R<C} \\
&\leq \varepsilon + \varepsilon = 2\varepsilon,
\end{aligned}
$$

- and the first part of the theorem is proved.
- To show the converse, show that $P_e^{(n)} \to 0 \Rightarrow H(\mathcal{V}) < C$ for source channel codes.

. . .

Source/Channel Coding Theorem

... proof continued.

- Define:

$$X^n(V^n) : \mathcal{V}^n \to \mathcal{X}^n \qquad //\text{encoder}$$
$$g_n(Y^n) : \mathcal{X}^n \to \mathcal{V}^n \qquad //\text{decoder}$$

- Now recall, original Fano says $H(X|Y) \leq 1 + P_e \log |\mathcal{X}|$.
- Here we have

$$H(V^n|\hat{V}^n) \leq 1 + P_e^{(n)} \log |\mathcal{V}^n| = 1 + nP_e \log |\mathcal{X}|$$

...

Source/Channel Coding Theorem

... proof continued.

- We get the following derivation

$$
\begin{aligned}
H(\mathcal{V}) &\leq \frac{H(V_1, V_2, \ldots, V_n)}{n} = \frac{H(V_{1:n})}{n} \\
&= \frac{1}{n} H(V_{1:n}|\widehat{V}_{1:n}) + \frac{1}{n} I(V_{1:n}; \widehat{V}_{1:n}) \\
&\leq \frac{1}{n}\left(1 + P_e^{(n)} \log |\mathcal{V}^n|\right) + \frac{1}{n} I(V_{1:n}; \widehat{V}_{1:n}) \quad //\text{by Fano} \\
&\leq \frac{1}{n}\left(1 + P_e^{(n)} \log |\mathcal{V}^n|\right) + \frac{1}{n} I(X_{1:n}; Y_{1:n}) \quad //V \to X \to Y \to \widehat{V} \\
&\leq \frac{1}{n} + P_e^{(n)} \log |\mathcal{V}| + C \quad //\text{memoryless}
\end{aligned}
$$

- Letting $n \to \infty$, $1/n$ and $P_e \to 0$ which leaves us with $H(\mathcal{V}) \leq C$.

Coding and codes

- Shannon's theorem says that there exists a sequence of codes such that if $R < C$ the error goes to zero.
- It doesn't provide such a code, nor does it offer much insight on how to find one.
- Typical set coding is not practical. Why? Exponentially large sized blocks.
- In all cases, we add enough redundancy to a message so that the original message can be decoded unambiguously

Physical Solution to Improve Coding

- It is possible to communicate more reliably by changing physical properties to decrease the noise (e.g., decrease p in a BSC).
- Use more reliable and expensive circuitry
- improve environment (e.g., control thermal conditions, remove dust particles or even air molecules)
- In compression, use more physical area/volume for each bit.
- In communication, use higher power transmitter, use more energy thereby making noise less of a problem.
- These are not IT solutions which is what we want.

Repetition Repetition Repetition Code Code Code

- Rather than send message $x_1 x_2 \dots x_k$ we repeat each symbol K times redundantly.
- Recall our example of repeating each word in a noisy analog radio connection.
- Message becomes $\underbrace{x_1 x_1 \dots x_1}_{K\times} \underbrace{x_2 x_2 \dots x_2}_{K\times} \dots$
- For many channels (e.g., BSC($p < 1/2$)), error goes to zero as $k \to 1$.
- Easy decoding: when K is odd, take a majority vote (which is optimal for a BSC)
- On the other hand, $R\alpha 1/K \to 0$ as $K \to \infty$
- This is really a pre-1948 way of thinking code.
- Thus, this is not a good code.

Repetition Code Example

- (From D. Mackay [2]) Consider sending message $s = 0010110$
- One scenario

s	0	0	1	0	1	1	0
t	000	000	111	000	111	111	000
n	000	001	000	000	101	000	000
r	000	001	111	000	010	111	000

- Another scenario

s		0	0	1	0	1	1	0
t		000	000	111	000	111	111	000
n		000	001	000	000	101	000	000
r		000	001	111	000	010	111	000
$\widehat{s}$		0	0	1	0	0	1	0
corrected errors			*					
undetected errors						*		

- Thus, can only correct one bit error not two.

Simple Parity Check Code

- Binary input/output alphabets $\mathcal{X} = \mathcal{Y} = \{0, 1\}$.
- Block sizes of $n-1$ bits: $x_{1:n-1}$.
- nth bit is an indicator of an odd number of 1 bits in.
- I.e., $x_n \leftarrow \text{mod}\left(\sum_{i=1}^{n-1} x_i, 2\right)$
- Thus a necessary condition for valid code word is: $\text{mod}\left(\sum_{i=1}^{n-1} x_i, 2\right) = 0$.
- Any any instance of an odd number of errors (bit swaps) won't pass this condition (although an even number of errors will pass the condition).
- Quite perfect: can not correct errors, and moreover only detects some of the kinds of errors (odd number of swaps).
- On the other hand, parity checks form the basis for many sophisticated coding schemes (e.g., low-density parity check (LDPC) codes, Hamming codes etc.).
- We study Hamming codes next.

(7; 4; 3) Hamming Codes I

- Best illustrated by an example.
- Let $\mathcal{X} = \mathcal{Y} = \{0, 1\}$.
- Fix the desired rate at $R = 4/7$ bit per channel use.
- Thus, in order to send 4 data bits, we need to use the channel 7 times.
- Let the four data bits be denoted $x_0, x_1, x_2, x_3 \in \{0, 1\}$.
- When we send these 4 bits, we are also going to send 3 additional parity or redundancy bits, named x_4, x_5, x_6.
- Note: all arithmetic in the following will be mod 2, i.e. $1 + 1 = 0$, $1 + 0 = 1$, $1 = 0 - 1 = -1$, etc.
- Parity bits determined by the following equations:

$$\begin{aligned} x_4 &\equiv x_1 + x_2 + x_3 \quad \mod 2 \\ x_5 &\equiv x_0 + x_2 + x_3 \quad \mod 2 \\ x_6 &\equiv x_0 + x_1 + x_3 \quad \mod 2 \end{aligned}$$

(7; 4; 3) Hamming Codes II

- I.e., if $(x_0, x_1, x_2, x_3) = (0110)$ then $(x_4, x_5, x_6) = (011)$ and complete 7-bit codeword sent over channel would be (0110011).
- We can also describe this using linear equalities as follows (all mod 2).

$$\begin{array}{cccccccc} & x_1 & +x_2 & +x_3 & +x_4 & & & = 0 \\ x_0 & & +x_2 & +x_3 & & +x_5 & & = 0 \\ x_0 & +x_1 & & +x_3 & & & +x_6 & = 0 \end{array}$$

- Or alternatively, as $Hx = 0$ where $x^T = (x_1, x_2, \ldots, x_7)$ and

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

- Notice that H is a column permutation of all non-zero length-3 column vectors.

(7; 4; 3) Hamming Codes III

- H is called *parity-check matrix*
- Thus the code words are defined by the null-space of H, i.e., $\{x : Hx = 0\}$.
- Since the rank of H is 3, the null-space dimension is 4, and we expect there to be $16 = 2^4$ binary vectors in this null space.
- These 16 vectors are:

0000000	0100101	1000011	1000011
0001111	0101010	1001100	1001100
0010110	0110011	1010101	1010101
0011001	0111100	1011010	1011010

Hamming Codes : weight

- Thus, any codeword is in $C = \{x : Hx = 0\}$.
- Thus, if $v_1, v_2 \in C$ then $v_1 + v_2 \in C$ and $v_1 - v_2 \in C$ due to linearity (codewords closed under addition and subtraction).
- *weight* of a code is 3, which is minimum number of ones in any non-zero codeword.
- Why? Since columns of H are all different, sum of any two columns is non-zero, so can't have any weight-$2v$ (summing two columns is never zero).
- Minimum weight is 3 since sum of two columns will equal another column, and sum of two equal column vectors is zero.

Hamming Codes : Distance

- Thus, any codeword is in $C = \{x : Hx = 0\}$.
- minimum distance of a code is also 3, which is minimum number of differences between any two codewords.
- Why? Given $v_1, v_2 \in C \Rightarrow (v_1 - v_2) \in C$. Can't have difference (or sum, and 1+1 = 1-1) of any two columns equaling zero, so $v_1 - v_2$ can't differ in only two places.
- Another way of saying this: if $v_1, v_2 \in C$ then $d_H(v_1, v_2) \geq 3$ where $d_H(\cdot, \cdot)$ is the *Hamming distance*.
- In general, codes with large minimum distance is good because then it is possible to correct errors, i.e., if $\hat{v}$ is received codeword, then we can find $i \in argmin_i d_H(\hat{v}, v_i)$ as the decoding procedure.

Hamming Codes : BSC I

- Now a BSC(p) (crossover probability p) will chance some of the bits (noise), meaning a 0 might change to a 1 and vice verse.
- So if $x = (x_0, x_1, \ldots, x_6)$ is transmitted, what is received is $y = x + z = (x_0 + z_0; x_1 + z_1, \ldots, x_6 + z_6)$, where $z = (z_0, z_1, \ldots, z_6)$ is the noise vector.
- Receiver knows y but wants to know x. We then compute

$$s = Hy = H(x + z) = \underbrace{Hx}_{=0} + Hz = Hz$$

- s is called the syndrome of y. $s = 0$ means that all parity checks are satisfied by y and is a necessary condition for a correct codeword.

Hamming Codes : BSC II

- Moreover, we see that s is a linear combination of columns of H

$$s = z_0 \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + z_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + z_2 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \cdots + z_6 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

- Since $y = x + z$, we know y, so if we know z we know x.
- We only need to solve for z in $s = Hz$, 16 possible solutions.
- Ex: Suppose that $y^T = 0111001$ is received, then $s^T = (101)$ and the 16 solutions are:

0100000	0010011	0101111	1001001
1100011	0001010	1000110	1111010
0000101	0111001	1110101	0011100
0110110	1010000	1101100	1011111

- 16 is better than 128 (possible z vectors) but still many.

Hamming Codes : BSC III

- What is the probability of each solution? Since we are assuming a BSC(p) with $p < 1/2$, the most probable solution has the leastweight. Any solution with weight k has probability p_k.
- Notice that there is only one possible solution with weight 1, and this is most probable solution.
- In previous example, most probable solution is $z^T = (01000000)$ and in $y = x + z$ with $y^T = 0111001$ this leads to codeword $x = 0011001$ and information bits 0011.
- In fact, for any s, there is a unique minimum weight solution for z in $s = Hz$ (in fact, this weight is no more than 1)!
- If $s = (000)$ then the unique solution is $z = (0000000)$.
- For any other s, then s must be equal to one of the columns of H, so we can generate s by flipping the corresponding bit of z on (giving weight 1 solution).

Hamming Decoding Procedure

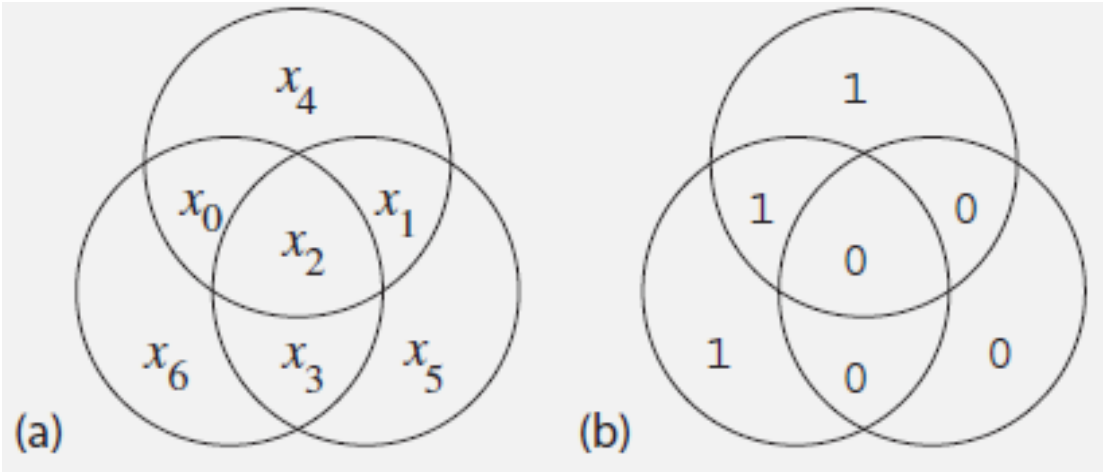
Here is the final decoding procedure on receiving y:

1. Compute the syndrome $s = Hy$.
2. If $s = (000)$ set $z \leftarrow (0000000)$ and goto 4.
3. Otherwise, locate unique column of H equal to s form z all zeros but with a 1 in that position.
4. Set $x \leftarrow y + z$.
5. output (x_0, x_1, x_2, x_3) as the decoded string.

This procedure can correct any single bit error, but fails when there is more than one error.

Hamming Decoding: Venn Diagrams

- We can visualize the decoding procedure using Venn diagrams

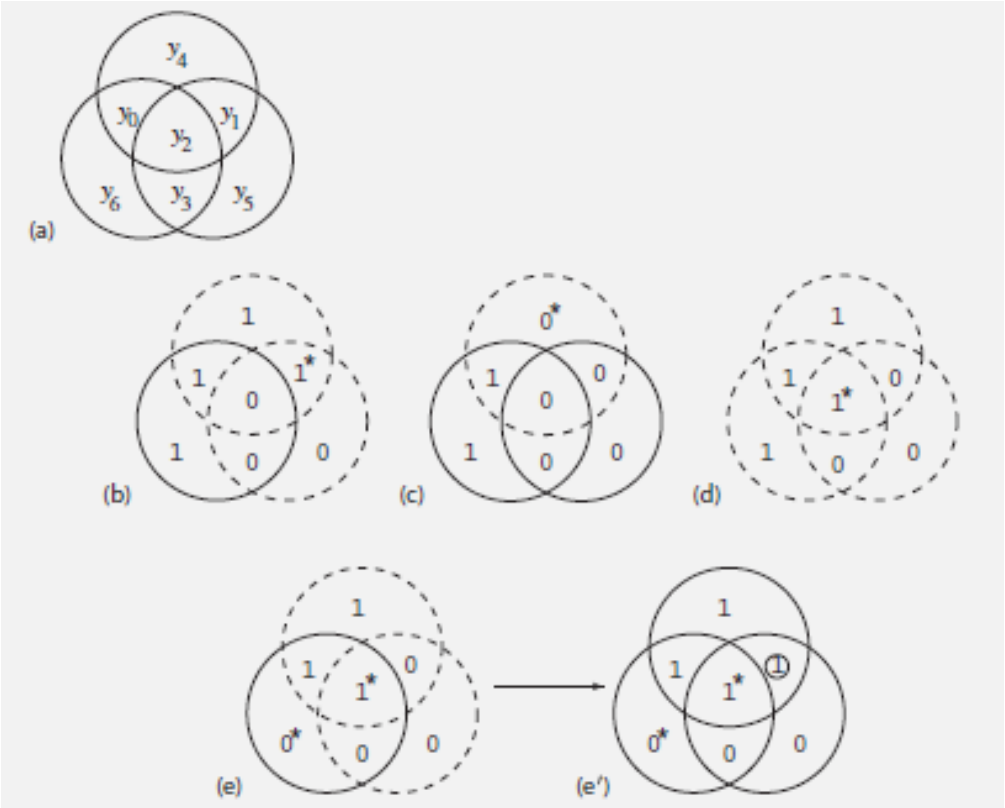


- Here, first four bits to be sent (x_0, x_1, x_2, x_3) are set as desired and parity bits (x_4, x_5, x_6) are also set. Figure shows $(x_0; x_1, \ldots, x_6) = (1, 0, 0, 0, 1, 0, 1)$ with parity check bits:

$$x_4 \equiv x_0 + x_1 + x_2 \mod 2$$
$$x_5 \equiv x_1 + x_2 + x_3 \mod 2$$
$$x_6 \equiv x_0 + x_2 + x_3 \mod 2$$

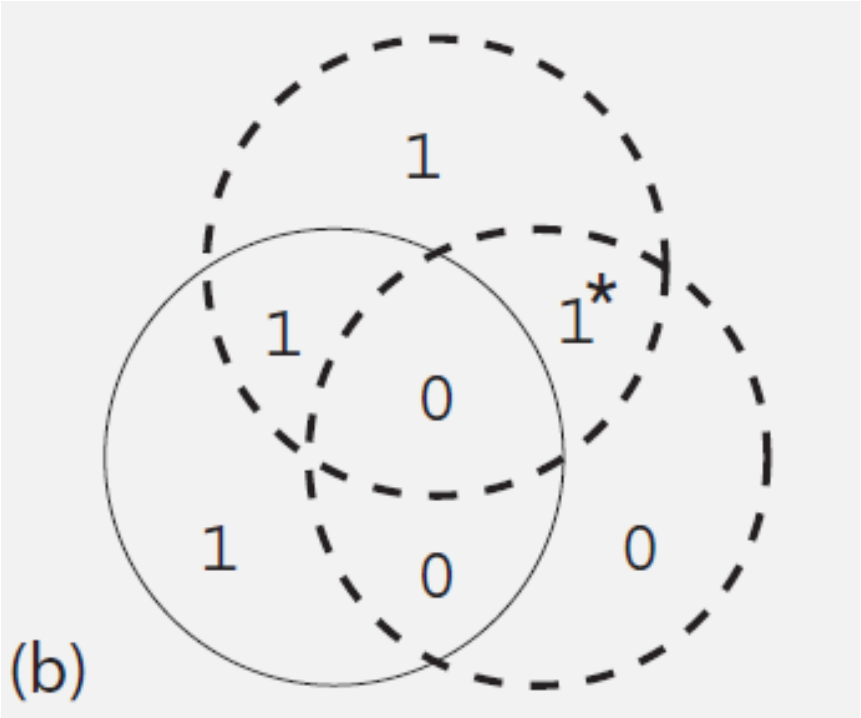
Hamming Decoding: Venn Diagrams

- The syndrome can be seen as a condition where the parity conditions are not satisfied.
- Above we argued that for $s \neq (0,0,0)$ there is always a one bit flip that will satisfy all parity conditions.



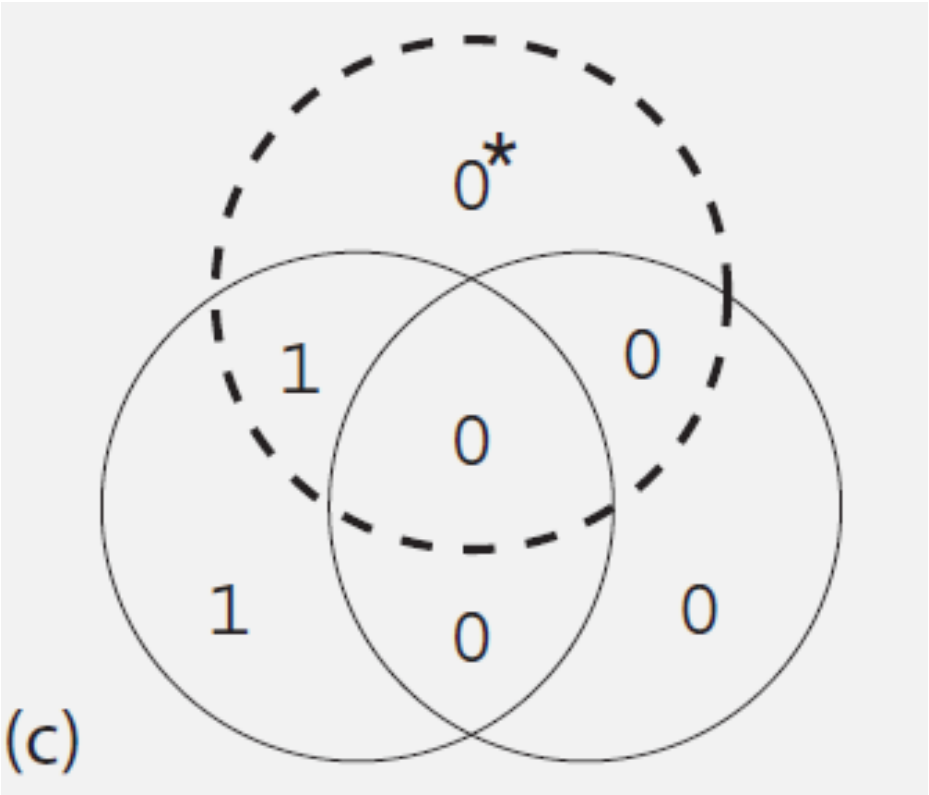
Hamming Decoding: Venn Diagrams

- Example: Here, z_1 can be flipped to achieve parity.



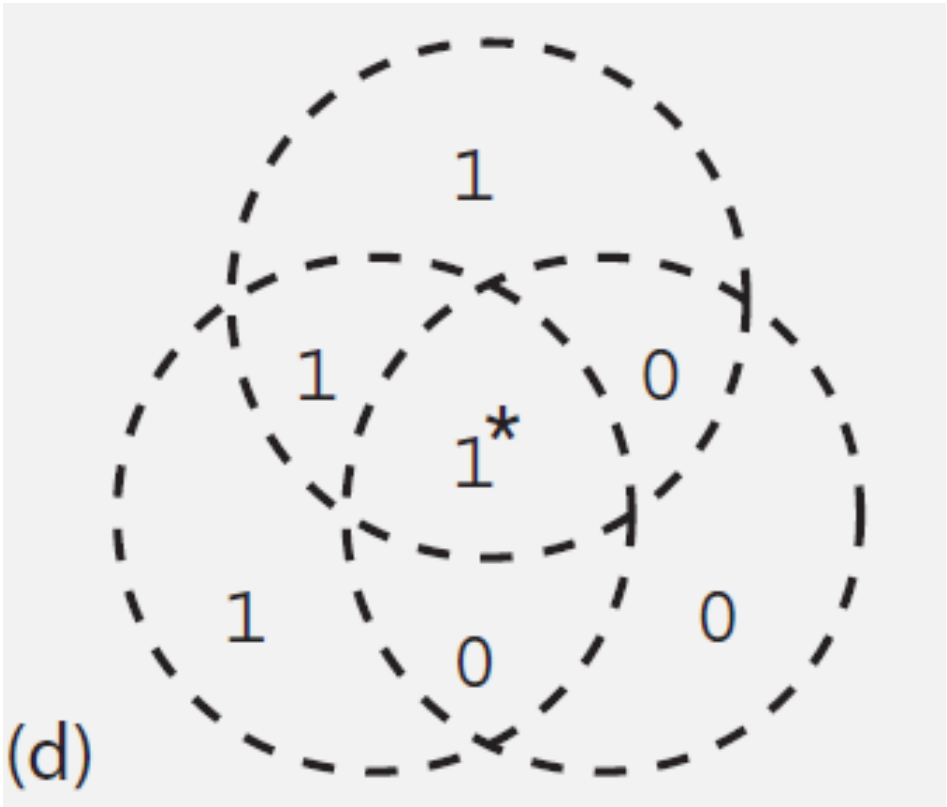
Hamming Decoding: Venn Diagrams

- Example: Here, z_4 can be flipped to achieve parity.



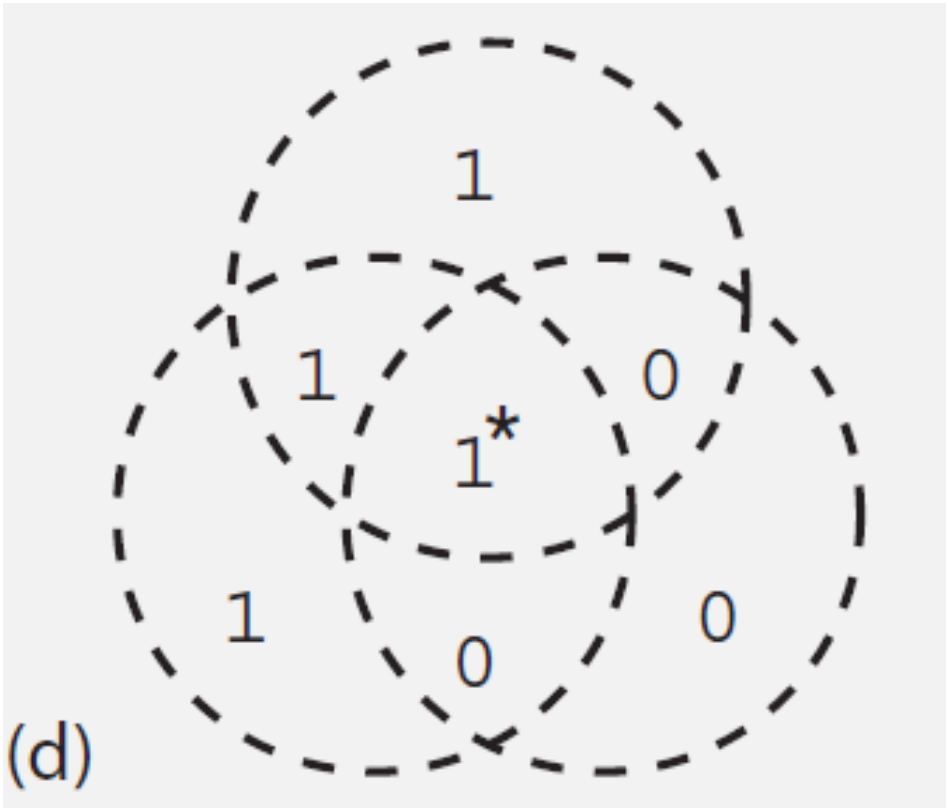
Hamming Decoding: Venn Diagrams

- Example: And here, z_2 can be flipped to achieve parity.



Hamming Decoding: Venn Diagrams

- Example: And here, there are two errors, y_6 and y_2 (marked with a *).



- Flipping y_1 will achieve parity, but this will lead to three errors (i.e., we will switch to a wrong codeword, and since codewords have minimum Hamming distance of 3, we'll get 3 bit errors).

Coding

- Many other coding algorithms.
- Reed Solomon Codes (used by CD players).
- Bose, Ray-Chaudhuri, Hocquenghem (BCH) codes.
- Convolutional codes
- Turbo codes (two convolutional codes with permutation network)
- Low Density Parity Check (LDPC) codes.
- All developed on our journey to find good codes with low rate that achieve Shannon's promise.

References


Thomas M. Cover, Joy A. Thomas, *Elements of Information Theory*, 2nd edition, Wiley, 2006.


David J.C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.


Robert M. Gray, *Entropy and Information Theory*, Springer, 2009


D. A. Huffman, A method for the construction of minimum redundancy codes, *Proc. IRE*, 40: 1098–1101,1952


C. E. Shannon, A mathematical theory of communication, *Bell System Technical Journal*, 1948.


R. V. Hamming, Error detecting and error correcting codes, *Bell System Technical Journal*, 29: 147-160, 1950