How small and how fast?

Shannon's Information Theoretic Limits for Data Compression and Communication

Faculty Development Program on Analog and Digital Communication
15th May - 24th May 2017

Prasad Krishnan
prasad.krishnan@iiit.ac.in

International Institute of Information Technology, Hyderabad

May, 2017

What is Information Theory?

What is Information Theory?

- Who is going to win the race?

What is Information Theory?

- Who is going to win the race?
- “If you don’t know where the stop line is, you cannot run the race effectively nor you can figure out how good you are.”

What is Information Theory

- Limits of communication and storage

What is Information Theory

- Limits of communication and storage
- **Compression :** “What is the minimum length of a code which represents a given source losslessly? ”

What is Information Theory

- Limits of communication and storage
- **Compression :** "What is the minimum length of a code which represents a given source losslessly? "
- **Channel capacity:** " What is the maximum rate of error-free transmission through a given channel?"

What is Information Theory

- Limits of communication and storage
- **Compression :** “What is the minimum length of a code which represents a given source losslessly? ”
- **Channel capacity:** “ What is the maximum rate of error-free transmission through a given channel?”
- ... and more.

Outline

Source material

- “A Mathematical Theory of Communication” - Claude Shannon, Bell System Technical Journey, 1948

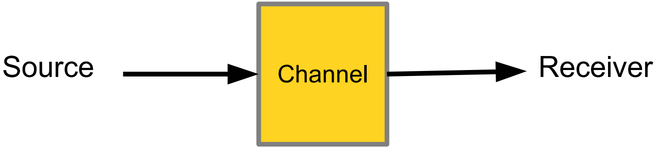
Outline

Digital Communication

Source Coding

Channel Coding

Communication System

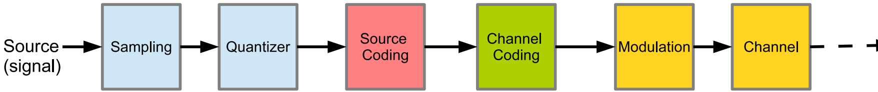


- Clear what the receiver and the (analog) source is.
- What is the channel?

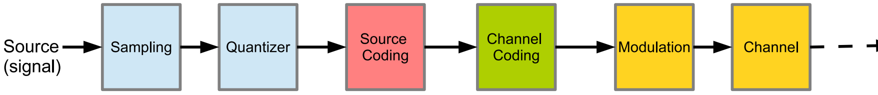
The Channel

- Given to us by nature (can optimise, but fundamental nature cannot be changed).
- Modelling noisy nature using probability (Models are not exact).
- Making appropriate assumptions are very important.
- AWGN : Typical Model for point to point (Noise signals are from a Gaussian Random Process)

The Digital Communication Model

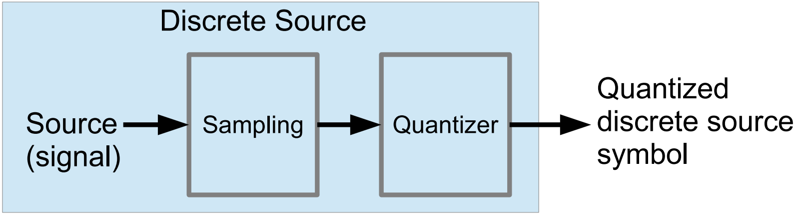


The Digital Communication Model

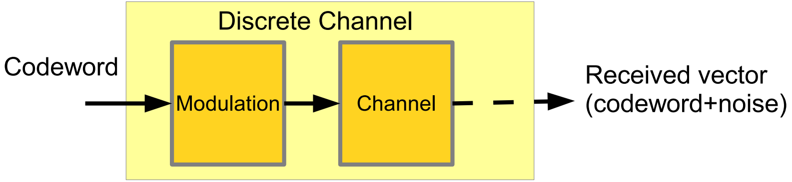


- Shannon did not explicitly deal with this model.
- Instead he merged together some of these blocks which helped him handle the problem probabilistically.

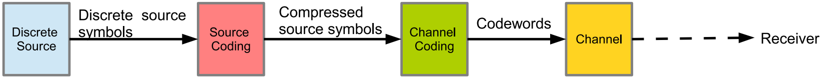
The Discrete Source



The Discrete Channel

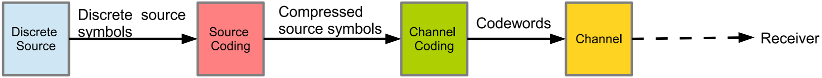


Shannon's Discrete (Digital) Communication Model



- Source coding $\rightarrow$ Remove inherent redundancy in the source, represent in minimal fashion.

Shannon's Discrete (Digital) Communication Model



- Source coding $\rightarrow$ Remove inherent redundancy in the source, represent in minimal fashion.
- Channel Coding $\rightarrow$ Add redundancy systematically to the source symbols to combat channel noise

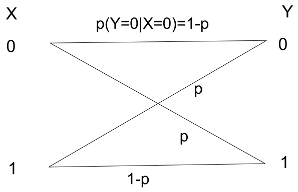
Discrete Sources and Channels - Examples

Binary Source

Sampler + Quantize + Convert each quantized sample to bits.

Binary Symmetric Channel (BSC)

- Analog Channel is AWGN
- Assume BPSK Modulation and Threshold Detector at receiver.



Outline

Source Coding

- $\mathcal{X}$ - Source Alphabet .
- X - Source Random Variable.
- We assume that there is a probability distribution $p(X)$ on the source.
- Want to compress this source - store it in the least space without loss of information.

Binary Source

- $X \in \mathcal{X} = \{0, 1\}$
- Source generates one binary symbol in each time unit.
- $p(X = 0) = p, \quad p(X = 1) = (1 - p). \; (X \sim Ber(p))$

The ‘least random’ and ‘most random’ source

- Every binary source can be stored using 1 bit representation.

The 'least random' and 'most random' source

- Every binary source can be stored using 1 bit representation.
- Consider that $p = 0$.
- Every time we run the source, we get $X = 1$.
- We are sure about the outcome.

The ‘least random’ and ‘most random’ source

- Every binary source can be stored using 1 bit representation.
- Consider that $p = 0$.
- Every time we run the source, we get $X = 1$.
- We are sure about the outcome.
- Least random source ($p = 0$ or $p = 1$)

The ‘least random’ and ‘most random’ source

- Every binary source can be stored using 1 bit representation.
- Consider that $p = 0$.
- Every time we run the source, we get $X = 1$.
- We are sure about the outcome.
- Least random source ($p = 0$ or $p = 1$)
- **Need 0 bits to represent this source!**

The 'least random' and 'most random' source

- Every binary source can be stored using 1 bit representation.
- Consider that $p = 0$.
- Every time we run the source, we get $X = 1$.
- We are sure about the outcome.
- Least random source ($p = 0$ or $p = 1$)
- **Need** 0 **bits to represent this source!**
- What is the most random binary source? **Ans:**

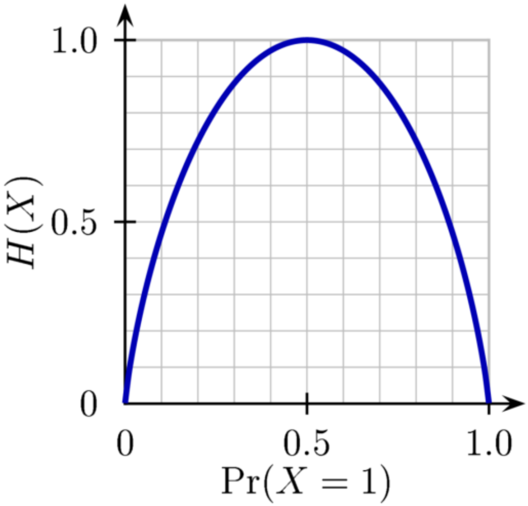
The 'least random' and 'most random' source

- Every binary source can be stored using 1 bit representation.
- Consider that $p = 0$.
- Every time we run the source, we get $X = 1$.
- We are sure about the outcome.
- Least random source ($p = 0$ or $p = 1$)
- **Need** 0 **bits to represent this source!**
- What is the most random binary source? **Ans:** $p = 0.5$.
- **Need** 1 **bit to represent this source. Maximum length!**
- *Minimum length of representation per source symbol* $=$ *Uncertainty/Randomness in the source*

The 'least random' and 'most random' source

- Every binary source can be stored using 1 bit representation.
- Consider that $p = 0$.
- Every time we run the source, we get $X = 1$.
- We are sure about the outcome.
- Least random source ($p = 0$ or $p = 1$)
- **Need 0 bits to represent this source!**
- What is the most random binary source? **Ans:** $p = 0.5$.
- **Need 1 bit to represent this source. Maximum length!**
- *Minimum length of representation per source symbol = Uncertainty/Randomness in the source*
- What is the most random M-ary source? How many bits is required?
- What about arbitrary p for binary source?

Binary Entropy



- To get the intermediate points, Shannon's idea was to 'let the source run' for some time.

Coin Toss - Bernouli RV

- Imagine a coin toss experiment with $p(heads) = p$, $p(Tails) = 1 - p$.
- Suppose we toss the coin N times with large N, how many heads and tails do we expect?

Coin Toss - Bernouli RV

- Imagine a coin toss experiment with $p(heads) = p$, $p(Tails) = 1 - p$.
- Suppose we toss the coin N times with large N, how many heads and tails do we expect?
- Roughly Np heads, $N(1 - p)$ tails.
- How many such N-length vectors are there?

Coin Toss - Bernouli RV

- Imagine a coin toss experiment with $p(heads) = p$, $p(Tails) = 1 - p$.
- Suppose we toss the coin N times with large N, how many heads and tails do we expect?
- Roughly Np heads, $N(1 - p)$ tails.
- How many such N-length vectors are there?
- $\begin{pmatrix} N \\ Np \end{pmatrix}$ vectors.

Coin Toss - Bernouli RV - Long vectors

- What happens as N increases?
- Applying Stirling's approximation $log_2(a!) = alog_2(a) - (log_2 e)a + O(log_2 a)$: we get

Coin Toss - Bernouli RV - Long vectors

- What happens as N increases?
- Applying Stirling's approximation $log_2(a!) = alog_2(a) - (log_2 e)a + O(log_2 a)$: we get
- Log_2(no of such vectors) $\approx$ $N(-plog(p) - (1-p)log(1-p)) = NH(X)$.
- $\boldsymbol{H(X) \triangleq -plog(p) - (1-p)log(1-p)}$ (Binary Entropy $H(p)$)

Coin Toss - Bernouli RV - Long vectors

- What happens as N increases?
- Applying Stirling's approximation $log_2(a!) = alog_2(a) - (log_2 e)a + O(log_2 a)$: we get
- Log_2(no of such vectors) $\approx$ $N(-plog(p) - (1-p)log(1-p)) = NH(X)$.
- $\boldsymbol{H(X) \triangleq -plog(p) - (1-p)log(1-p)}$ (Binary Entropy $H(p)$)
- What is the probability of each such vector? **Ans:** $\approx 2^{-NH(X)}$.
- Holds with equality as $N \to \infty$.

Compressing a binary RV

- Note that the source distribution begins to look like a uniform distribution for large N, with $2^{NH(X)}$ possible vectors, and Prob(any vector)$\approx 2^{-NH(X)}$.
- Already know how to represent uniform RV .

Compressing a binary RV

- Note that the source distribution begins to look like a uniform distribution for large N, with $2^{NH(X)}$ possible vectors, and Prob(any vector)$\approx 2^{-NH(X)}$.
- Already know how to represent uniform RV .
- Need length $log(2^{NH(X)}) = NH(X)$ bits.
- No. of bits required to represent one source symbol $= \frac{NH(X)}{N} = H(X)$.

Compressing a binary RV

- Note that the source distribution begins to look like a uniform distribution for large N, with $2^{NH(X)}$ possible vectors, and Prob(any vector)$\approx 2^{-NH(X)}$.
- Already know how to represent uniform RV .
- Need length $log(2^{NH(X)}) = NH(X)$ bits.
- No. of bits required to represent one source symbol $= \frac{NH(X)}{N} = H(X)$.
- Using less number of bits than this will cause loss of information about source!

Shannon's source coding theorem

Shannon's Source Coding Theorem

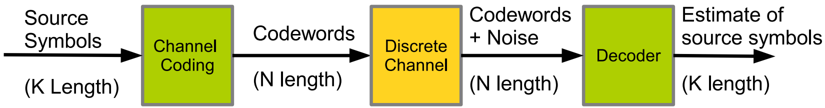
The minimum number of bits required to represent a source random variable X taking values from $\mathcal{X}$ with distribution $p(X)$ is

$$H(X) = \sum_{x \in \mathcal{X}} -p(x) log(p(x)).$$

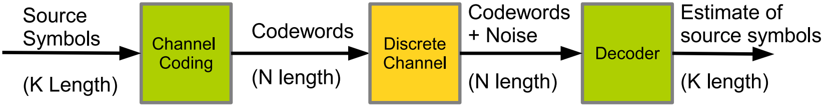
An explicit scheme exists which can achieve compression arbitrarily close to $H(X)$.

Outline

Channel Coding

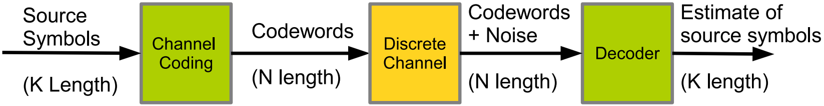


Channel Coding



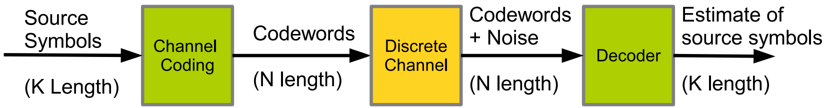
- Let Code alphabet be $\mathcal{X}$ with distribution $p(X)$ (Codeword vector is $\boldsymbol{x} \in \mathcal{X}^N$).

Channel Coding



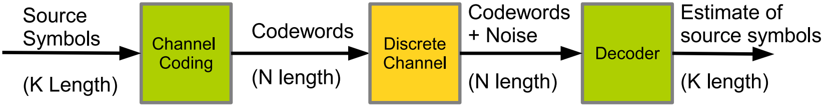
- Let Code alphabet be $\mathcal{X}$ with distribution $p(X)$ (Codeword vector is $\boldsymbol{x} \in \mathcal{X}^N$).
- Received alphabet be $\mathcal{Y}$ with distribution $p(Y)$, received vector is $\boldsymbol{y} \in \mathcal{Y}^N$.

Channel Coding



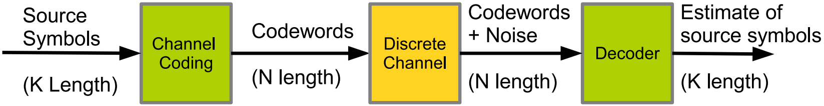
- Let Code alphabet be $\mathcal{X}$ with distribution $p(X)$ (Codeword vector is $\boldsymbol{x} \in \mathcal{X}^N$).
- Received alphabet be $\mathcal{Y}$ with distribution $p(Y)$, received vector is $\boldsymbol{y} \in \mathcal{Y}^N$.
- Channel is described according to conditional distribution $p(y|x)$, $y \in \mathcal{Y}, x \in \mathcal{X}$. (Assume $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} p(y_i|x_i)$).

Channel Coding



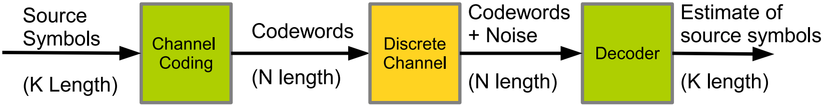
- Let Code alphabet be $\mathcal{X}$ with distribution $p(X)$ (Codeword vector is $\boldsymbol{x} \in \mathcal{X}^N$).
- Received alphabet be $\mathcal{Y}$ with distribution $p(Y)$, received vector is $\boldsymbol{y} \in \mathcal{Y}^N$.
- Channel is described according to conditional distribution $p(y|x)$, $y \in \mathcal{Y}, x \in \mathcal{X}$. (Assume $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} p(y_i|x_i)$).
- Note that $p(y) = \sum_{x \in \mathcal{X}} p(y|x)p(x)$.

Channel Coding



- Let Code alphabet be $\mathcal{X}$ with distribution $p(X)$ (Codeword vector is $\boldsymbol{x} \in \mathcal{X}^N$).
- Received alphabet be $\mathcal{Y}$ with distribution $p(Y)$, received vector is $\boldsymbol{y} \in \mathcal{Y}^N$.
- Channel is described according to conditional distribution $p(y|x)$, $y \in \mathcal{Y}, x \in \mathcal{X}$. (Assume $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} p(y_i|x_i)$).
- Note that $p(y) = \sum_{x \in \mathcal{X}} p(y|x)p(x)$.
- **Rate of the code** $= \frac{K}{N}$ (no. of msg symbols per codeword symbol).

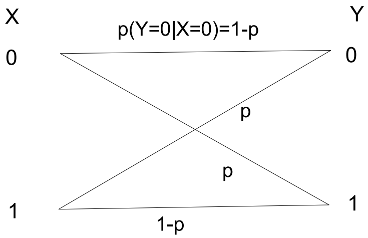
Channel Coding



- Let Code alphabet be $\mathcal{X}$ with distribution $p(X)$ (Codeword vector is $\boldsymbol{x} \in \mathcal{X}^N$).
- Received alphabet be $\mathcal{Y}$ with distribution $p(Y)$, received vector is $\boldsymbol{y} \in \mathcal{Y}^N$.
- Channel is described according to conditional distribution $p(y|x)$, $y \in \mathcal{Y}, x \in \mathcal{X}$. (Assume $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{N} p(y_i|x_i)$).
- Note that $p(y) = \sum_{x \in \mathcal{X}} p(y|x)p(x)$.
- **Rate of the code** $= \frac{K}{N}$ (no. of msg symbols per codeword symbol).
- **Question:** *What is the maximum rate possible to achieve for (almost) zero probability of error?*

Binary Source + Binary Symmetric Channel Setup

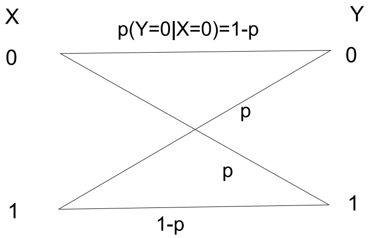
Binary Symmetric Channel:



- Assume source symbols are binary and channel is BSC.
- **Encoding:** K-length binary strings to N-length binary strings.
- Number of codewords is 2^K.

Binary Source + Binary Symmetric Channel Setup

Binary Symmetric Channel:



- Assume source symbols are binary and channel is BSC.
- **Encoding:** K-length binary strings to N-length binary strings.
- Number of codewords is 2^K.
- If X is channel input RV, we can also write channel output Y as

$$Y = X + Z,$$

where Z has a $Ber(p)$ distribution and is independent of X.

Channel coding on the BSC with binary inputs

Question: What is the best $\frac{K}{N} = \frac{log_2(\text{No. of codewords})}{N}$ with (almost) zero probability of error?

- Repetition code: $K = 1$, mapped to N-length string with same bits as the input.
- Probability of error goes to zero, what about rate?

Channel coding on the BSC with binary inputs

Question: What is the best $\frac{K}{N} = \frac{log_2(\text{No. of codewords})}{N}$ with (almost) zero probability of error?

- Repetition code: $K = 1$, mapped to N-length string with same bits as the input.
- Probability of error goes to zero, what about rate?
- Are there codes which have good rate, and low probability of error?

Channel coding on the BSC with binary inputs

Question: What is the best $\frac{K}{N} = \frac{log_2(\text{No. of codewords})}{N}$ with (almost) zero probability of error?

- Repetition code: $K = 1$, mapped to N-length string with same bits as the input.
- Probability of error goes to zero, what about rate?
- Are there codes which have good rate, and low probability of error?
- Shannon proved such codes exist ! (Trade off with complexity of encoding/decoding)

Channel coding on the BSC with binary inputs

- Let $\boldsymbol{x}$ be some N-length codeword input to the channel.
- Assuming large N, how does the received vector $\boldsymbol{y}$ look like?

Channel coding on the BSC with binary inputs

- Let $\boldsymbol{x}$ be some N-length codeword input to the channel.
- Assuming large N, how does the received vector $\boldsymbol{y}$ look like?
- **Ans:** We will have $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$, for some $\boldsymbol{z}$ containing roughly Np ones.

Channel coding on the BSC with binary inputs

- Let $\boldsymbol{x}$ be some N-length codeword input to the channel.
- Assuming large N, how does the received vector $\boldsymbol{y}$ look like?
- **Ans:** We will have $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$, for some $\boldsymbol{z}$ containing roughly Np ones.
- Number of such possible error vectors is $\begin{pmatrix} N \\ Np \end{pmatrix} \approx 2^{NH(Z)}$, and the probability of each is $\approx 2^{-NH(Z)}$.

Non-intersecting Hamming spheres

- Sphere of 'radius' Np around $\boldsymbol{x}$.
- The received vector could be anywhere in this sphere.
- Number of vectors in this sphere $\approx 2^{NH(Z)}$.

Non-intersecting Hamming spheres

- Sphere of ‘radius’ Np around $\boldsymbol{x}$.
- The received vector could be anywhere in this sphere.
- Number of vectors in this sphere $\approx 2^{NH(Z)}$.
- Suppose there was another codeword $\boldsymbol{x}'$ such that the spheres around $\boldsymbol{x}$ and $\boldsymbol{x}'$ intersect.
- Then there is decoding error

Sphere packing

- Any two codewords should be such that their spheres (of radius Np) don't intersect.

Sphere packing

- Any two codewords should be such that their spheres (of radius Np) don't intersect.
- Pick the maximum possible number of such codewords following above rule. (Sphere packing)

Sphere packing

- Any two codewords should be such that their spheres (of radius Np) don't intersect.
- Pick the maximum possible number of such codewords following above rule. (Sphere packing)
- Total possible received vectors $\approx 2^{NH(Y)}$.
- Number of vectors in each sphere $\approx 2^{NH(Z)}$.

Sphere packing

- Any two codewords should be such that their spheres (of radius Np) don't intersect.
- Pick the maximum possible number of such codewords following above rule. (Sphere packing)
- Total possible received vectors $\approx 2^{NH(Y)}$.
- Number of vectors in each sphere $\approx 2^{NH(Z)}$.
- Hence Maximum number of codewords

$$\frac{2^{NH(Y)}}{2^{NH(Z)}} = 2^{N(H(Y)-H(Z))}.$$

- Maximum rate of code (with output distribution $p(y)$) $= H(Y) - H(Z)$.

Capacity of BSC

- Note that $H(Y)$ is a function of $p(y)$ which is a function of $p(x)$.

Capacity of BSC

- Note that $H(Y)$ is a function of $p(y)$ which is a function of $p(x)$.
- The maximum rate of transmission through the BSC is thus

$$
\begin{aligned}
& max_{p(x)}(H(Y) - H(Z)) \\
&= max_{p(x)} H(Y) - H(p) \\
&= 1 - H(p)
\end{aligned}
$$

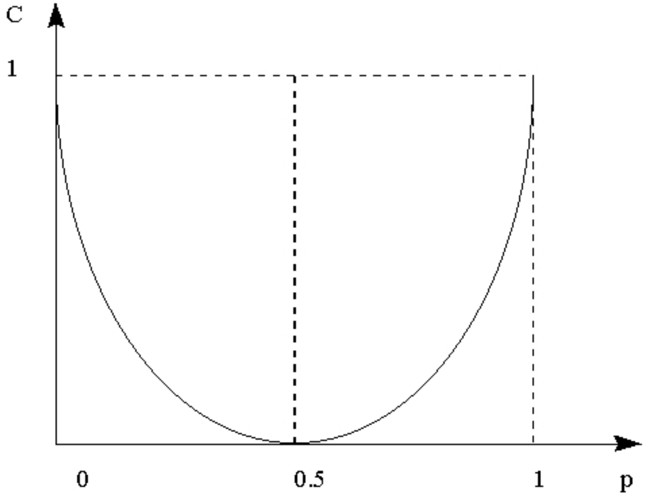
Capacity of BSC

- Note that $H(Y)$ is a function of $p(y)$ which is a function of $p(x)$.
- The maximum rate of transmission through the BSC is thus

$$
\begin{aligned}
& max_{p(x)}(H(Y) - H(Z)) \\
&= max_{p(x)} H(Y) - H(p) \\
&= 1 - H(p)
\end{aligned}
$$

- This is called the Capacity of the BSC.

Capacity of BSC



Shannon's Channel Capacity Theorem

Channel Capacity Theorem

For any discrete-memoryless channel with given $p(y|x)$, the rate of transmission R is always $\leq C$, where C is the channel capacity given as

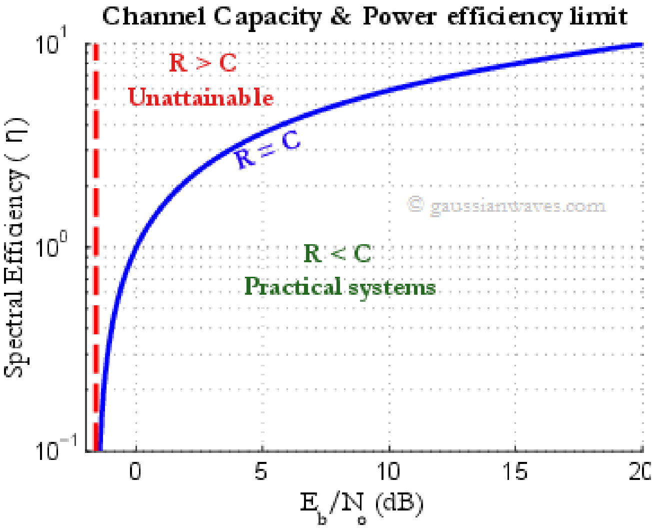
$$C \triangleq max_{p(x)} H(Y) - H(Y|X).$$

Also, there exists some encoding scheme by which any rate arbitrarily close to capacity is achievable.

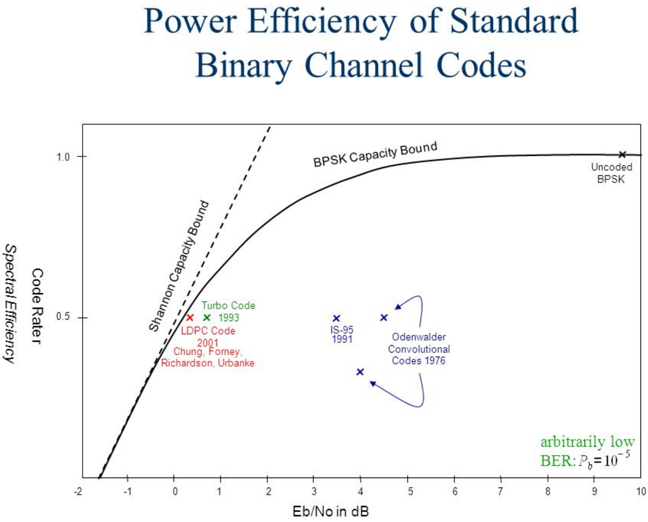
Capacity of AWGN

- For AWGN with bandwidth W:: Capacity = $\frac{1}{2} log(1 + SNR) = \frac{1}{2} log(1 + \frac{P}{N_0 W})$.
- **Only Existence of Good Codes is shown by Shannon.**
- Construction of 'good' codes has happened (for AWGN channels) over the last several decades since Shannon.

Capacity curve for AWGN



Turbo Codes and LDPC Codes along Shannon Capacity Curve



- Who is going to win the race?
- “If you don’t know where the stop line is, you cannot run the race effectively nor you can figure out how good you are.”

- Who is going to win the race?
- “If you don’t know where the stop line is, you cannot run the race effectively nor you can figure out how good you are.”
- Shannon tells us where the stop line is!

Thank You!