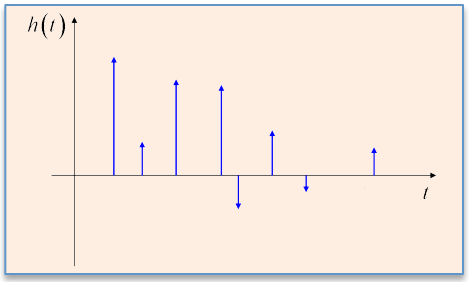
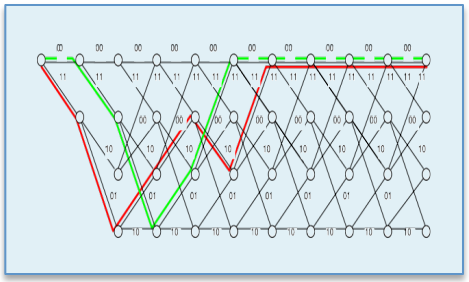
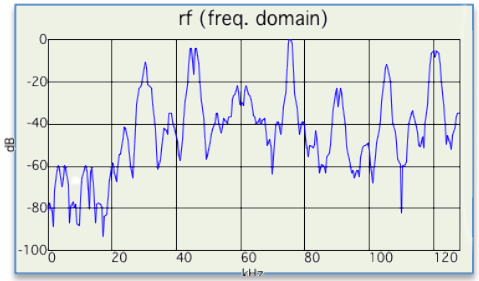
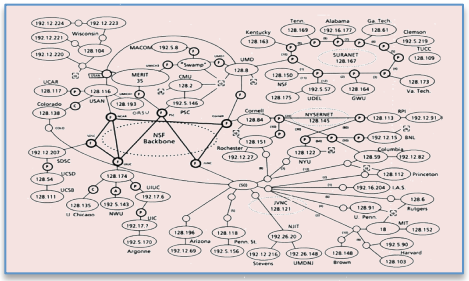








INTRODUCTION TO EECS II

DIGITAL COMMUNICATION SYSTEMS

6.02 Fall 2011
Lecture #9

- Claude E. Shannon
- Mutual information
- Channel capacity
- Transmission at rates up to channel capacity, and with asymptotically zero error

First a quick review of what we know about information & entropy

…

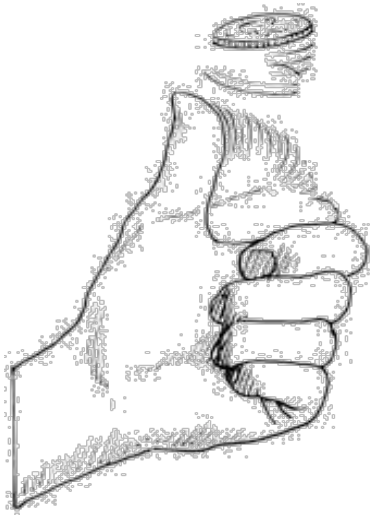
Measuring Information

We've seen Shannon's (and Hartley's) definition of the information obtained on being told the outcome x_i of a probabilistic experiment X:

$$I(\{X = x_i\}) = \log_2\left(\frac{1}{p_X(x_i)}\right)$$

where $p_X(x_i)$ is the probability of the event $\{X = x_i\}$.

The unit of measurement (when the log is base-2) is the **bit** (**b**inary **i**nformation uni**t**).



1 bit of information corresponds to $p_X(x_i) = 0.5$. So, for example, when the outcome of a *fair* coin toss is revealed to us, we have received 1 bit of information.

"*Information is the resolution of uncertainty*"
Shannon

Expected Information as Uncertainty or Entropy

Consider a discrete random variable X, which may represent the set of possible messages to be transmitted at a particular time, taking possible values $x_1, x_2, \ldots, x_m$, with respective probabilities $p_X(x_1), p_X(x_2), \ldots, p_X(x_m)$.

The *entropy* $H(X)$ of X is the expected (or mean or average) value of the information obtained by learning the outcome of X:

$$H(X) = \sum_{i=1}^{m} p_X(x_i) I(\{X = x_i\}) = \sum_{i=1}^{m} p_X(x_i) \log_2 \left(\frac{1}{p_X(x_i)} \right)$$

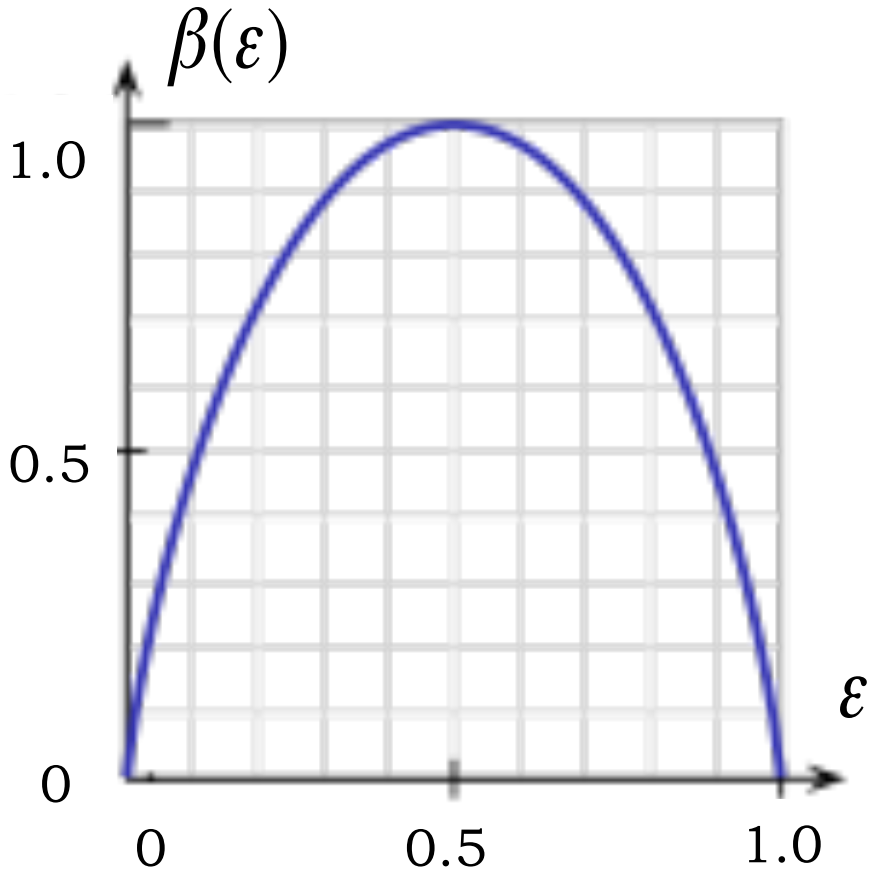
When all the $p_X(x_i)$ are *equal* (with value $1/m$), which corresponds to the case of maximum uncertainty about the outcome, then

$$H(X) = \log_2 m \qquad \text{or} \qquad m = 2^{H(X)}$$

e.g., Binary entropy function $\beta(\varepsilon)$

$B = 1$ with probability ε,

$B = 0$ with probability $1 - \varepsilon$,



$$H(B) = -\varepsilon \log_2 \varepsilon - (1-\varepsilon)\log_2(1-\varepsilon) = \beta(\varepsilon)$$

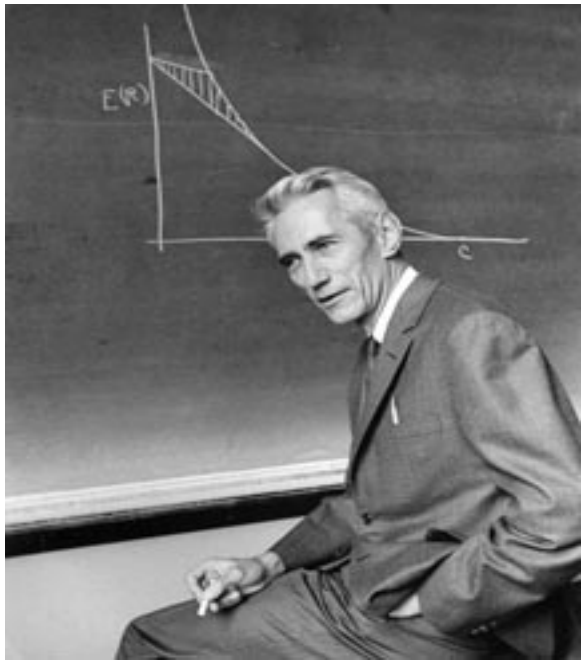
Claude Elwood Shannon

mathematician, electrical engineer, cryptographer, informatician, professor, juggler, unicyclist, puzzler, gadgeteer, rhymster, …!

Claude E. Shannon, 1916-2001

1937 Masters thesis, EE Dept, MIT
A symbolic analysis of relay and switching circuits
Introduced application of Boolean algebra to logic circuits, and vice versa.
Very influential in digital circuit design.
"Most important Masters thesis of the century"

1940 PhD, Math Dept, MIT
An algebra for theoretical genetics
To analyze the dynamics of Mendelian populations.



MIT faculty
1956-1978

Joined Bell Labs
"A mathematical theory of cryptography"
1945/1949

"A mathematical theory of communication"
1948

Letter from Shannon to *Scientific American* Editor, Dec 1981

Dear Dennis:

You probably think I have been fritterin', I say fitterin', away my time while my juggling paper is languishing on the shelf. This is only half true. I have come to two conclusions recently:

1) I am a better poet than scientist.
2) Scientific American should have a poetry column.

You may disagree with both of these, but I enclose "A Rubric on Rubik Cubics" for you.

Sincerely,

Claude E. Shannon

P.S. I am still working on the juggling paper.

http://blogs.scientificamerican.com/cross-check/2011/03/28

A Rubric on Rubik Cubics

Strange imports come from Hungary:
Count Dracula, and ZsaZsa G.,
Now Erno Rubik's Magic Cube
For PhD or country rube.

This fiendish clever engineer
Entrapped the music of the sphere.
It's sphere on sphere in all 3D—
A kinematic symphony!

Ta! Ra! Ra! Boom De Ay!
One thousand bucks a day.
That's Rubik's cubic pay.
He drives a Chevrolet.

Forty-three quintillion plus
Problems Rubik posed for us.
Numbers of this awesome kind
Boggle even Sagan's mind. [… lots more!]

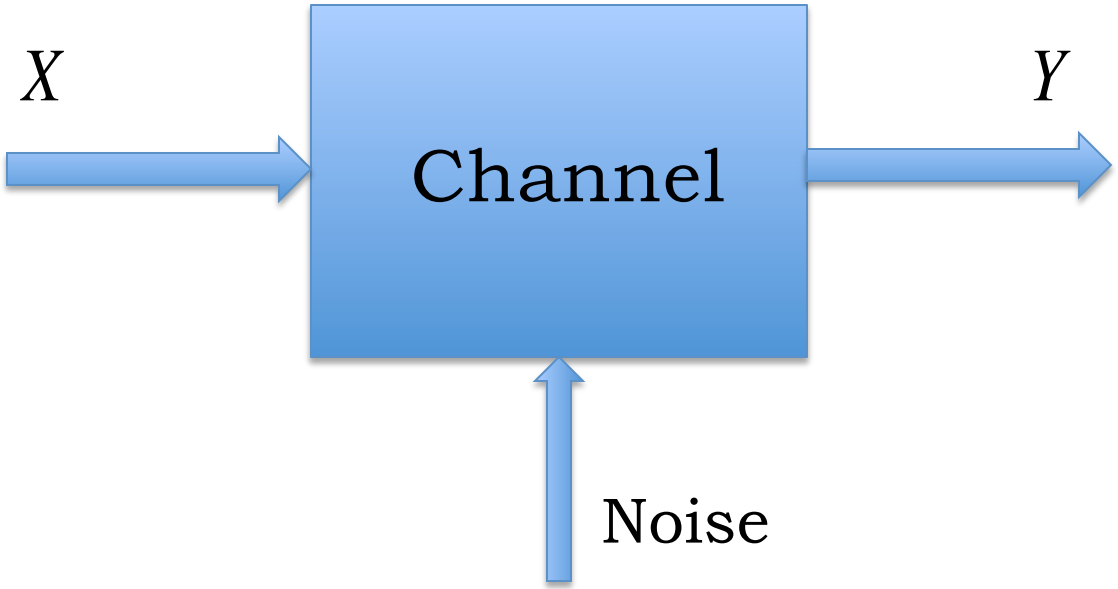
Back to information & entropy
(but now the part of the story that
nobody before Shannon
had any inkling of!)
…

Mutual information

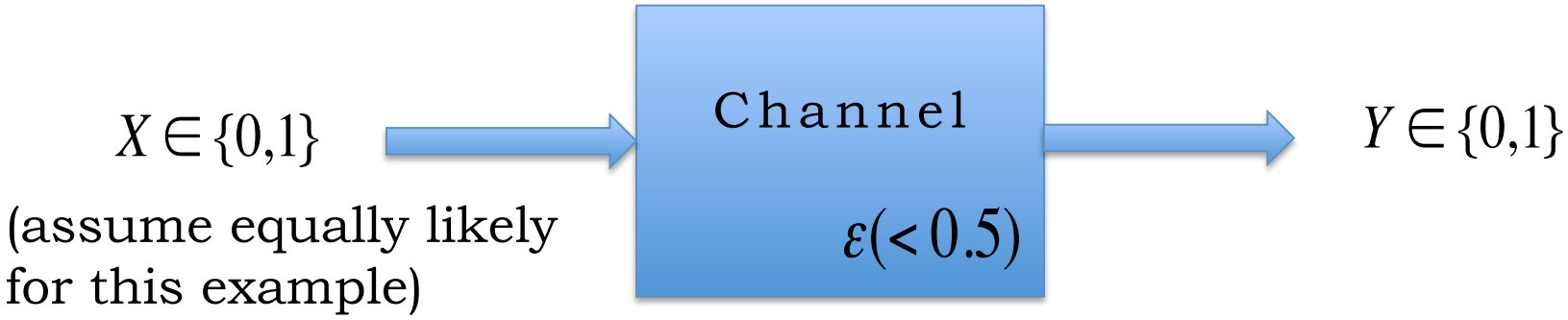
$$I(X;Y) = H(X) - H(X \mid Y)$$

How much is our uncertainty about X reduced by knowing Y ?

Evidently a central question in communication or, more generally, inference.



e.g., Mutual information between input and output of binary symmetric channel (BSC)



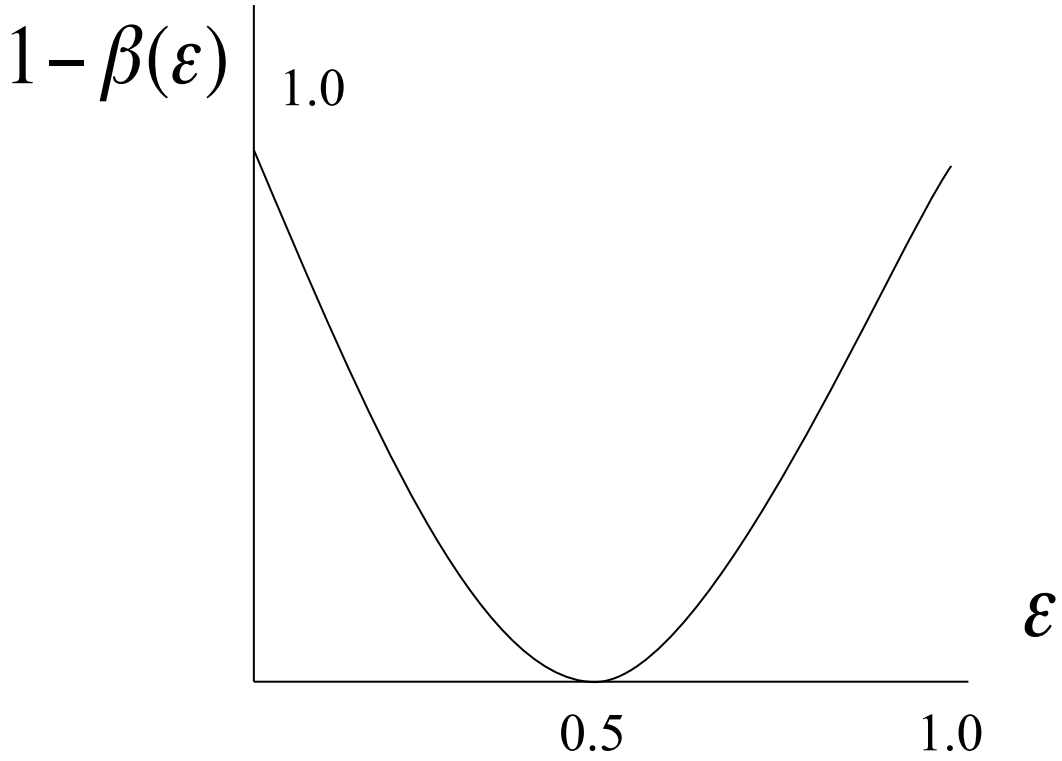
(assume equally likely for this example)

With probability $\varepsilon(<0.5)$ the input binary digit gets flipped before being presented at the output.

It turns out (see later slide) that $I(X;Y) = I(Y;X)$ (really?!) so

$$I(X;Y) = I(Y;X) = H(Y) - H(Y \mid X)$$
$$= 1 - H(Y \mid X = 0)p_X(0) - H(Y \mid X = 1)p_X(1)$$
$$= 1 - \beta(\varepsilon)$$

So mutual information between input and output of the BSC looks like this:



For low-noise channel, significant reduction in uncertainty about the input after observing the output.

For high-noise channel, little reduction.

Evaluating mutual information and conditional entropy in general

$$H(X,Y) = H(X) + H(Y \mid X)$$

$$= H(Y) + H(X \mid Y)$$

because

$$p(x_i, y_j) = p(x_i)p(y_j \mid x_i)$$

$$= p(y_j)p(x_i \mid y_j)$$

so

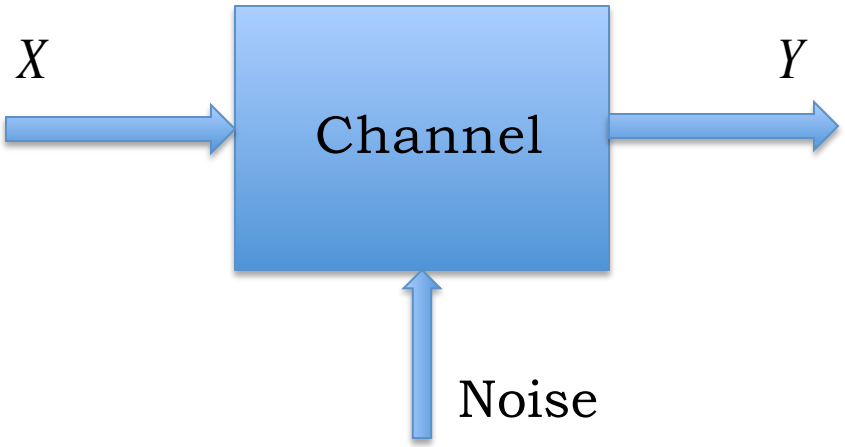
$$I(X;Y) = I(Y;X)$$ → mutual information is symmetric

To compute conditional entropy:

$$H(X \mid Y = y_j) = \sum_{i=1}^{m} p(x_i \mid y_j) \log_2 \left(\frac{1}{p(x_i \mid y_j)} \right)$$

$$H(X \mid Y) = \sum_{i=1}^{m} H(X \mid Y = y_j) p(y_j)$$

Channel capacity



To characterize the *channel*, rather than the input and output, define

$$C = \max I(X;Y) = \max\left\{H(X) - H(X \mid Y)\right\}$$

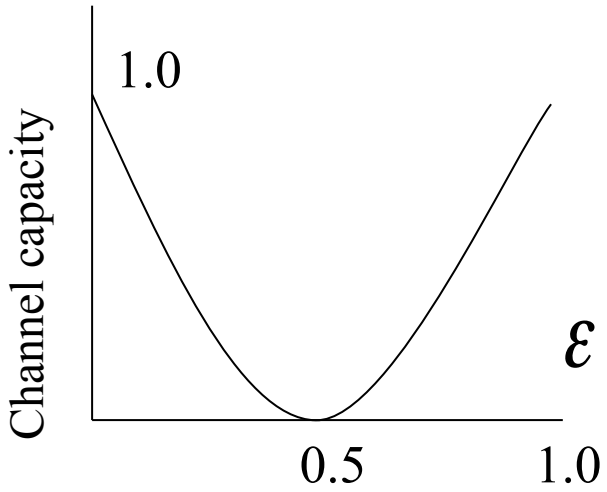
where the maximization is over all possible distributions of X.

This is the most we can expect to reduce our uncertainty about X through knowledge of Y, and so must be *the most information we can expect to send through the channel on average, per use of the channel.*

e.g., capacity of the binary symmetric channel

Easiest to compute as $C = \max\{H(Y) - H(Y \mid X)\}$, still over all possible probability distributions for X. The second term doesn't depend on this distribution, and the first term is maximized when 0 and 1 are equally likely. But this is exactly what we assumed in our mutual information example earlier. So:

➔ $C = 1 - \beta(\varepsilon)$



Information rate and bit error rate

The memoryless binary symmetric channel (BSC) abstraction takes a stream of bits at its input, and puts out essentially the same stream of bits, except each input bit is flipped (from 0 to 1, or from 1 to 0) with probability ε.

With no coding at the input, we have a message transmission rate of 1 bit/s, but the BER for message bits is ε.

With a replication code at the input (decided by majority vote at the output), and repeating each input message bit n times, the effective BER on message bits is reduced to order $\varepsilon^{(n+1)/2}$, but the message transmission rate is now only $1/n$.

We know we can do better with smarter codes, but how much better? The notion of channel capacity tells us, as we outline next.

What channel capacity tells us about how fast and how accurately we can communicate …

The magic of asymptotically error-free transmission at any rate $R < C$

Shannon showed that one can theoretically transmit information (i.e., message bits) at an average rate $R < C$ with arbitrarily low error.

(He also showed the converse, that transmission at an average rate $R \geq C$ incurs an error probability that is lower-bounded by some positive number.)

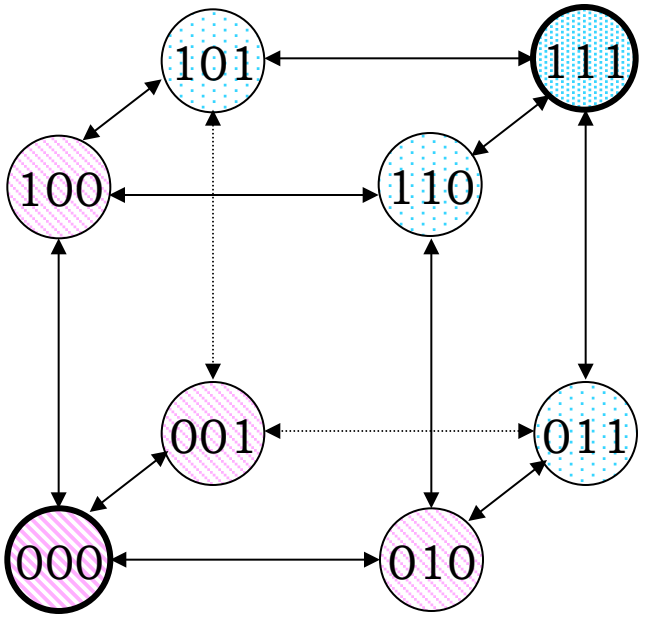
The secret: Encode blocks of k message bits into n-bit codewords, so $R = k/n$, with k and n very large.

We've already seen hints of this in our earlier discussion of coding. Let's work through an *intuitive argument* for the case of the BSC, but recognizing that a rigorous argument takes more work.

Two special things about LARGE n

1. Of the 2^n points in the space of n-component binary vectors, only a small fraction $n / 2^n$ lie Hamming distance 1 away from a particular codeword; only a fraction $n(n-1) / 2^{(n+1)}$ lie Hamming distance 2 away from a codeword; etc. So there's lots of space to put in codewords without their bumping into each other, even with errors.

Things are tighter for $n = 3$



And secondly …

2. The *law of large numbers* tells us that for an n-bit codeword at the input of the BSC, we can expect very close to $n\varepsilon$ bit errors in the received codeword. Moreover, all ways of getting $n\varepsilon$ errors are equally likely.

Since the output entropy for a given n-bit input sequence is

$$nH(Y \mid X = x_i) = n\beta(\varepsilon)$$

because the channel is used independently n times, we know that each input codeword produces one of approximately $2^{n\beta(\varepsilon)}$ equally likely corrupted codewords at the output, each at a Hamming distance of essentially $n\varepsilon$ from the correct codeword.

Another way to arrive at this:

Invoking the law of large numbers as earlier, the number of possible corrupted codewords at the output of the BSC for an n-bit input codeword is approximately

$$\binom{n}{n\varepsilon} = \frac{n!}{(n-n\varepsilon)!(n\varepsilon)!}$$

Using Stirling's approximation

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

and some algebra again shows that the number of possible corrupted codewords at the output is approximately $2^{n\beta(\varepsilon)}$

What all this implies for high-rate, low-error transmission

As in the figure on Slide 20, but now working in n dimensions, we want to select 2^k corners of the unit cube, out of the 2^n available corners, to constitute our codewords. This corresponds to having k message bits in a codeword of length n, so a rate $R = k/n$. To maximize this rate, we want k as large as possible, i.e., select as many corners as possible to be codewords.

However, to avoid decoding errors, we want to be sure the $2^{n\beta(\varepsilon)}$ neighbors of any particular codeword at Hamming distance $n\varepsilon$ from it are distinct from the the corresponding neighbors of all other codewords. A necessary condition for this is:

$$2^k 2^{n\beta(\varepsilon)} \leq 2^n - 2^k < 2^n$$

(packing the unit cube with as many Hamming $n\varepsilon$ -neighborhoods as possible)

which simplifies to

$$R < [1 - \beta(\varepsilon)] = C$$

Random coding

The problem with the preceding calculation is it only yields a necessary condition for high-rate low-error transmission. It doesn't show whether an actual choice of 2^k corners is possible that maintains a sufficient distance between the codewords to have the probability of error be as small as desired.

Shannon instead showed that a *random* choice of 2^k codewords, subject to the necessary constraint $R < C$, will in fact lead to a probability of error that reduces exponentially as n increases. Specifically, the probability of error on a given transmission is the probability that the received corrupted codeword is also one of the corrupted codewords that could have been obtained from the transmission of one of the $2^k - 1$ other codewords. This probability may be estimated as

$$(2^k - 1)2^{n\beta(\varepsilon)} / 2^n \approx 2^{k-n+n\beta(\varepsilon)} = 2^{-(C-R)n}$$

which indeed tends exponentially to 0 with increasing n.

Some final comments

1. The penalty paid for low error rates in this framework is **latency** and **computational cost**: we have to wait till all n bits of the codeword are received, and then do the necessary computations, before we can report any bits of the message.

2. Shannon's result says nothing about what codes are efficient, computationally tractable, etc.

3. A huge amount of work has gone into developing good codes --- ones that provide substantial error protection with manageable computational effort and tolerable latency, while coming close to the Shannon limit.

4. Excellent codes are known for a variety of settings. Among these is the class of *low density parity check (LDPC)* codes developed by Prof. Robert Gallager of EECS in his 1960 PhD thesis at MIT, but which have only now become computationally tractable.