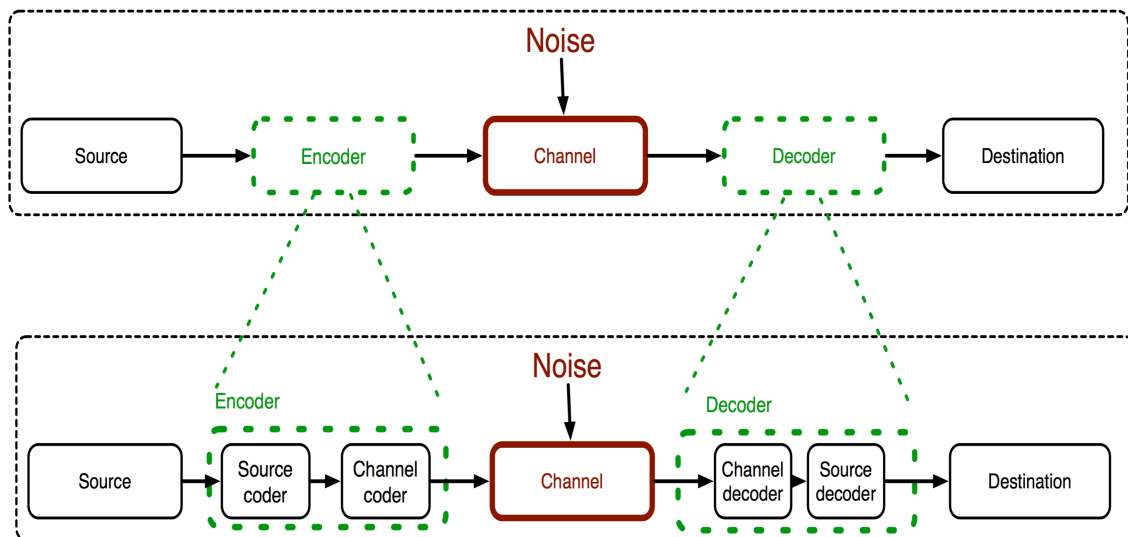# Source-Channel Coding

In this final part of the course, we want to study the "ultimate problem" of transmitting a source of information over a noisy channel.

(Image credit: **ECE 534** by Natasha Devroye, thanks a lot!)

Recall that a source can be modeled as P_V and a noisy channel by a conditional probability distribution $P_{Y|X}$. The problem of **source-channel coding** studies if there exists an encoding procedure mapping n iid samples V^n from the source P_V to channel inputs X^n, such that the resulting channel outputs Y^n can be decoded back to $\hat{V}^n$ in such a way that $P[V^n \neq \hat{V}^n]$ vanishes for $n \to \infty$.

One way of building such a source-channel encoder and decoder coder is sketched in the lower part of the figure above. We can use our knowledge from Module 03 of this course to build an optimal source coder that compresses the source information (with vanishing error) down to a rate $R > H(V)$ bits per source symbol. Then, we can use the results of this model to build a channel coder which will work with vanishing error as long as the rate $R < C$ is smaller than the capacity. From this intuition, it seems that requiring $H(V) < C$, i.e. that the entropy of the source we want to transmit is strictly smaller than the capacity of the channel, is a **necessary and sufficient** condition for transmitting a source over a channel.

We will see on the following pages that under certain conditions, that is indeed the case. However, let me try to convince (and confuse) you here that it is *a priori* not clear that separating the source and channel coder is such a good idea:

1. Recall the example from the **very first video (starting at 2:31)** of English text sent over an erasure channel. It seems that for this kind of source (English text) and channel (erasure channel), it works pretty well to simply not encode the text at all, and then try to recover (using some language model, in your brain or on a computer) from the erasures by filling in the missing letters. That is a very different source-channel coder than the one suggested above, which would first compress the text (say using zip) to get rid of redundancies, and then insert again some redundancy (using some error-correcting code) to survive the erasure channel.

2. Consider the **cocktail-party effect.** Our brains seems to be extremely good at extracting individual conversations in very noisy situations (such as at cocktail parties). You can experience that yourself by repeatedly watching the following video, trying to focus to understand either the left or right speaker:

   Again, in such a scenario, it seems rather wasteful to first compress the audio information to (basically) random noise, and then re-encoding it to survive the noisy environment, and then undo these steps to decode. Speaking English with a human voice already seems to be a very good encoder of information for this scenario, especially combined with a very powerful human-brain decoder.

created: 2018-12-12