The Search for Sparsity

As we have discussed, typical signals reside in **very** high-dimensional spaces, in the sense that the size of the tensors is huge.

However, most possible tensors are not realistic data. Most matrices are not an image. Most vectors don't represent real sounds. In fact, the realistic or typical data for any given application occupy a vanishingly small part of that space. There is an information theoretic proof of that, but it is perhaps beyond us here.

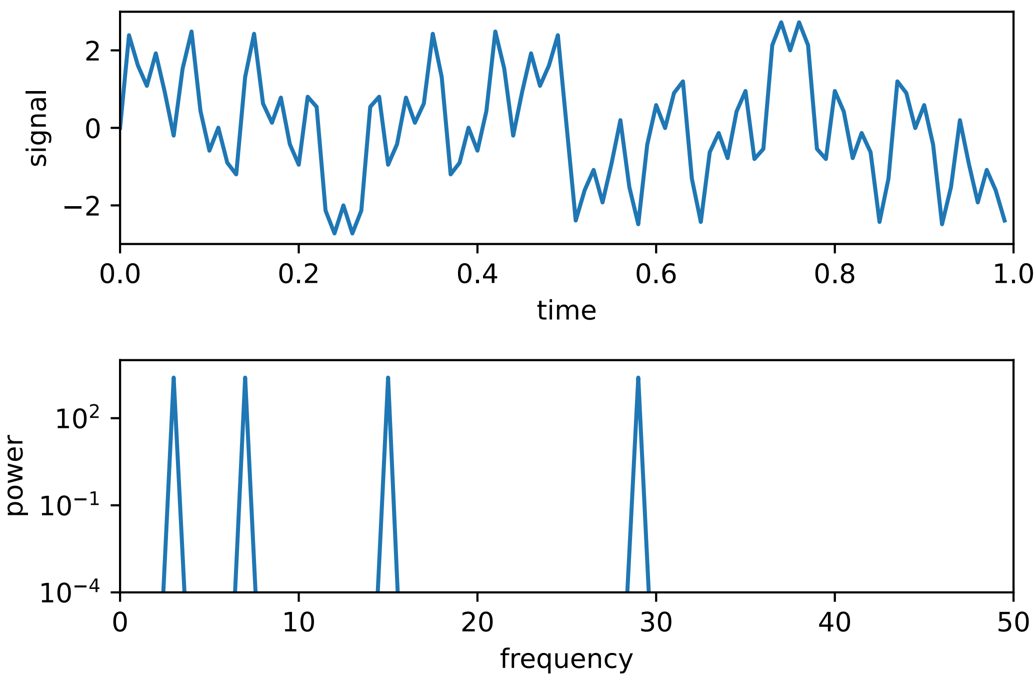
If we can find a means to describe that space, and represent our signal in the reduces then we automatically have a huge advantage because we can (at least to some extent) avoid the curse of dimensionality.

Also, if the data lives in a smaller space than the noise, compressing into that space can remove or at least reduce the noise. It can even allow us to reconstruct missing parts of the data by inferring where the point should lie in the reduced space.

Moreover, we can *understand* sparse signals in a way we don't their more complicated cousins. In some sense, all AI is the search for sparse (or simple or concise) ways of describing complicated ensembles of signals.

One approach is to search for a way to represent our signal such that it is sparse, $i.e,$ we might preserve the original dimension of the space, but require that most signals (we care about) have only a few non-zero values.

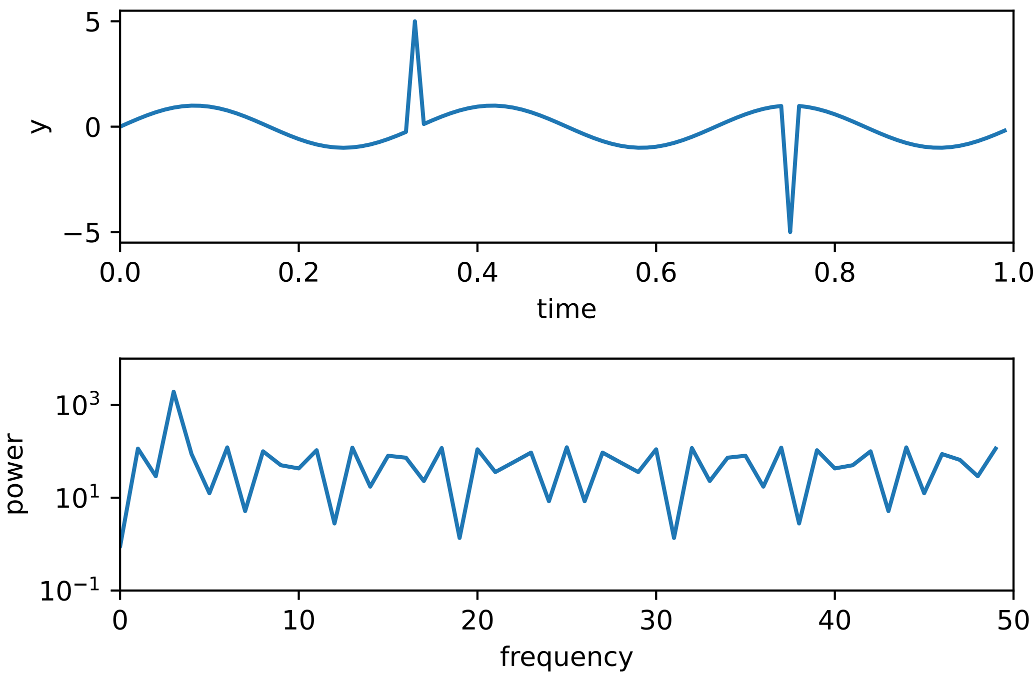
The Fourier transform is one approach. We saw, last week, simple examples such as:



This simple linear transform have moved us into a space where the complicated signal can be represented by only four non-zero numbers.

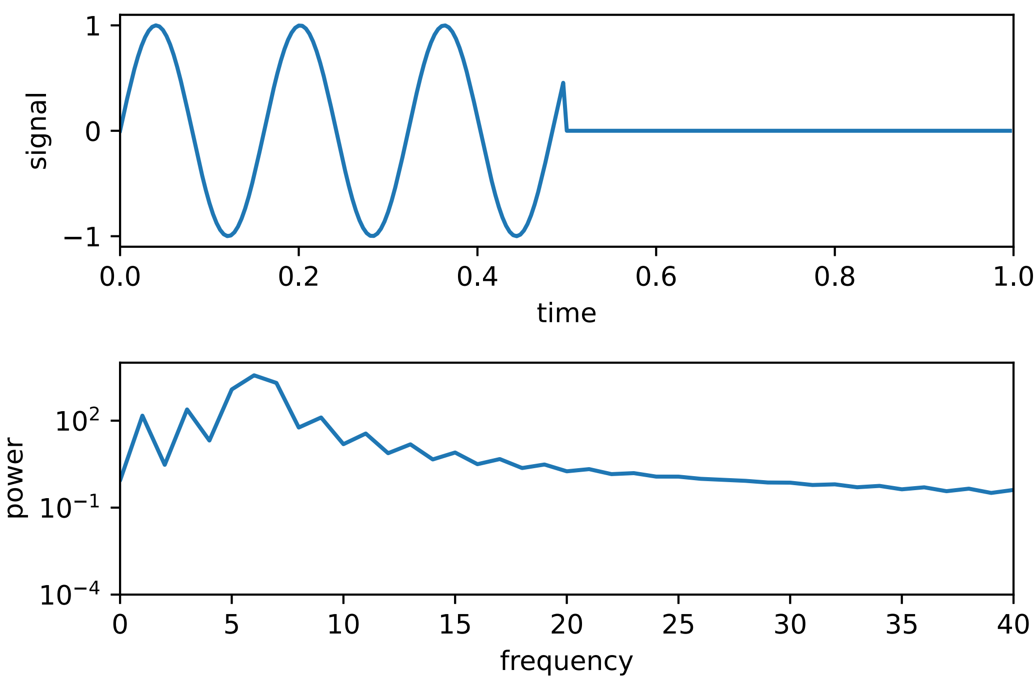
The Fourier transform is useful to find sparsity in many signals (and for other reasons as well). However it has some limitations. Primarily it a fixed transform. The transform is ignorant of the data. That means that we just have to hope that our data is sparse in the Fourier domain. Many data sets are, but some are not.

A simple example is that of a sinusoid plus two spikes (delta functions). We know this has a sparse representation (I just described it) but we can't get to it either in the time domain or the Fourier domain.



In the Fourier domain we still see evidence of the sine wave (the spike at $f = 4$) but the noise generated by the spikes in the original make the signal anything but sparse.

Another example is the simple case where there is a sine wave in part of the signal.



Note that now:

- the signal's Fourier representation is more complicated than just a single spike, and that
- the Fourier spectrum looses any notion of temporal locality, *i.e.,* it can't show that the signal is only present in the 1st half of the time.

Many (simple) signals are not sparsely represented in the Fourier or time domain. There are plenty of alternative transforms. Wavelets are a whole class of transforms that aim to provide some time and some frequency information, but the standard wavelet transforms still work with a fixed basis (though you have some choice over which basis because wavelets are a class of transform not a single transform).

So we might want an approach that adapts to the signal. That's is where we are heading today.

Such techniques are sometimes called dimension reduction techniques because ultimately they allow us to map the data into a space where it is sparse, and hence we don't care about most of the dimensions.

One more note before we proceed: many signals are only approximately represented as sparse. They might, for instance, have very few zero elements, but many elements that are small, and only a few elements that are significant. We call such vectors or matrices (or tensors) *compressible* and note that for real data, this is the goal as true sparsity is too idealised.

Subtopics

1. From diagonalisation to SVD
2. From SVD to PCA
3. Beyond PCA

Notes and links

- Code for these examples is in `simple_example.ipynb`
-