Solutions to Information Theory Exercise Problems 9–12

Exercise 9

Y and Z are two continuous random variables.

Y has an exponential probability density distribution $p(x)$ over $x \in [0, \infty]$: $p(x) = e^{-x}$.
Note that

$$\int_0^\infty e^{-x}\, dx = \left[-e^{-x}\right]_0^\infty = 1\,.$$

Z has a uniform probability density distribution: $p(x) = 1/\alpha$ for $x \in [0, \alpha]$, else $p(x) = 0$.

Calculate the differential entropies $h(Y)$ and $h(Z)$ for these two continuous random variables, and find the value of α for which these differential entropies are the same. Sketch these distributions.

Solution:

Differential entropy h for a probability density distribution $p(x)$ is $-\int_x p(x) \log_2 p(x)\, dx$, so:

$$h(Z) = -\int_0^\alpha \frac{1}{\alpha} \log_2 \left(\frac{1}{\alpha}\right) dx = \frac{\alpha}{\alpha} \log_2(\alpha) = \log_2(\alpha)$$

and

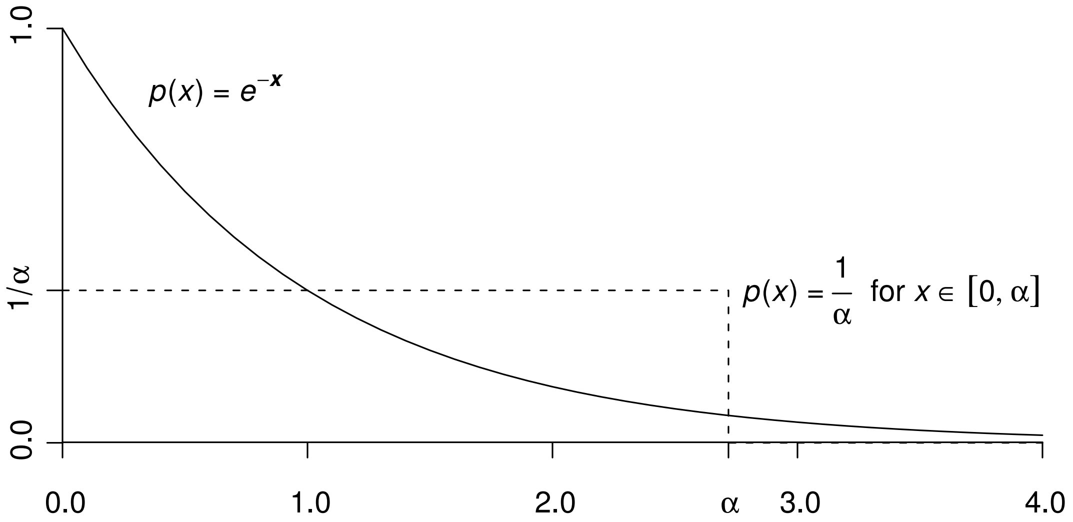
$$h(Y) = -\int_0^\infty e^{-x} \log_2(e^{-x})\, dx = \log_2(e) \int_0^\infty x e^{-x}\, dx$$

an integral which we can evaluate using integration-by-parts: $\int u\, dv = uv - \int v\, du$, with:
$u = x$, $dv = e^{-x}\, dx$, $v = -e^{-x}$, $du = dx$, so:

$$\int_0^\infty x e^{-x}\, dx = \left[-x e^{-x}\right]_0^\infty + \int_0^\infty e^{-x}\, dx = 1.$$

Thus we see that $h(Z) = h(Y)$ when $\log_2(\alpha) = \log_2(e)$, or in other words, $\alpha = e = 2.718...$

The following plot shows the two probability density distributions for Y and Z with equal differential entropies:



Exercise 10

(a) What does it mean for a function to be "self-Fourier?" Name three functions which are of importance in information theory and that have the self-Fourier property, and in each case mention a topic or a theorem exploiting it.

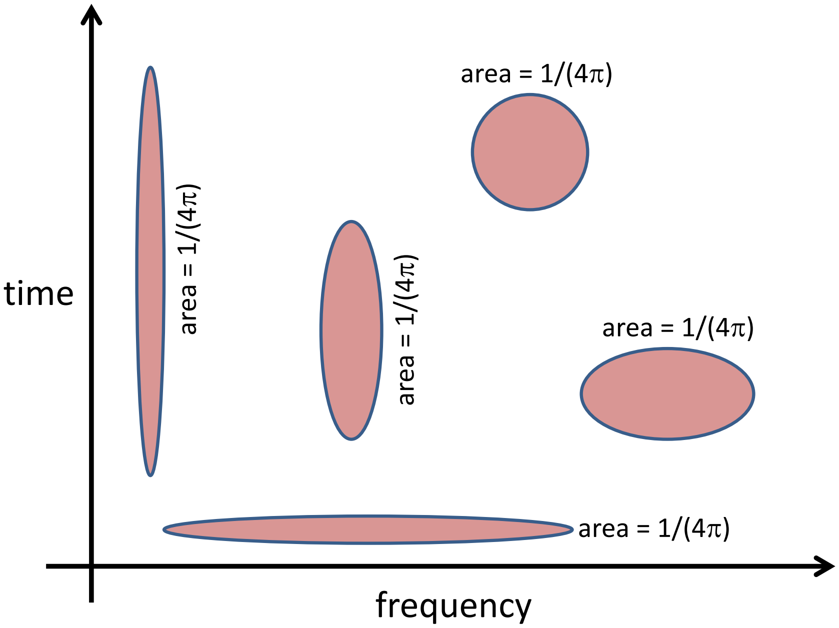
Solution:

(a) Functions are "self-Fourier" if they are identical in form to their own Fourier transforms. Some are continuous, some are discrete, some are periodic, some are aperiodic. Examples and their appearances in information theory include:

1. The comb function (a regular train of Dirac delta functions). The self-Fourier property of the comb function is critical for the Nyquist Sampling Theorem. (Slides 136 – 145.)

2. Gaussian functions, which describe the probability distribution $p(x)$ for a continuous signal's excursions x (from its mean) that gives it the maximum possible entropy $h(p(x))$ for any given variance σ^2. This is important both in characterising the AWGN (Additive White Gaussian Noise) channel, and in satisfying the requirement of "maximising the mutual information over all possible input distributions" when deriving the capacity C of such a channel in Shannon's *Noisy Channel Coding Theorem.*

3. Gabor wavelets, which as encoders optimise joint information resolution simultaneously in time and frequency under the fundamental (Heisenberg-Weyl-Gabor) Uncertainty Principle. Any given area of the time/frequency "Information Diagram" (see Figure below) is maximally resolved when populated by Gabor wavelets.



(b) Show that the set of all Gabor wavelets is closed under convolution, *i.e.* that the convolution of any two Gabor wavelets is just another Gabor wavelet. [HINT: This property relates to the fact that these wavelets are also closed under multiplication, and that they are also self-Fourier. You may address this question for just 1D wavelets if you wish.]

Solution:

(b) The functional form of a 1D Gabor wavelet is:

$$f(x) = e^{-(x-x_0)^2/\alpha^2} e^{i\omega_0(x-x_0)}$$

localised at "epoch" x_0, modulated with frequency ω_0, and having a size or spread constant α.

Such wavelets have Fourier transforms $F(\omega)$ with the same functional form, but with the parameters just interchanged:

$$F(\omega) = e^{-(\omega-\omega_0)^2\alpha^2} e^{-ix_0(\omega-\omega_0)}$$

It is algebraically obvious that the product of any two Gabor wavelets $f(x)$ will still have the functional form of a Gabor wavelet, because the contents of all the exponentials simply combine additively. Therefore the product's Fourier transform will also preserve this general form. Hence (using the convolution theorem of Fourier analysis), it follows that the family of Gabor wavelets are also closed under convolution.

(c) Show that the family of sinc functions (for any λ) used in the Nyquist Sampling Theorem,

$$\text{sinc}(x) = \frac{\sin(\lambda x)}{\lambda x}$$

is closed under convolution. Show further that when two different sinc functions are convolved, the result is simply whichever one of them had the lower frequency, *i.e.* the smaller λ.

Solution:

(c) When two functions are convolved together, their Fourier transforms are just multiplied together to give the Fourier transform of the result of the convolution. Conveniently, the Fourier transform of any sinc function with frequency parameter λ is a (zero-centred) rectangular pulse function $\text{rect}_\lambda(\omega)$ of width 2λ (i.e. $|\omega| \leq \lambda$) and 0 beyond. Multiplying two such pulse functions together just creates yet another pulse function:

$$\text{rect}_{\lambda_1}(\omega) \times \text{rect}_{\lambda_2}(\omega) = \text{rect}_{\lambda_3}(\omega)$$

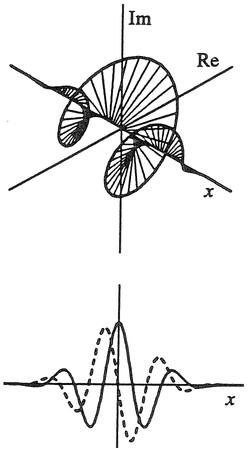
and therefore the result of the convolution is again just a sinc function. But specifically, because the product of the two pulse functions in frequency is simply whichever one of them was narrower, the resulting sinc function is just whichever one had the smaller λ (lower frequency):

$$\lambda_3 = \min\{\lambda_1, \lambda_2\}$$

Exercise 11

(a) An important class of complex-valued functions for encoding information with maximal resolution simultaneously in the frequency domain and the signal domain are Gabor wavelets. Using an expression for their functional form, explain:

1. their spiral helical trajectory as phasors, shown here with projections of their real and imaginary parts;
2. the Uncertainty Principle under which they are optimal;
3. the spaces they occupy in the Information Diagram;
4. some of their uses in pattern encoding and recognition.



Solution:

(a) Their functional form can be expressed as: $e^{-(x-x_0)^2/\alpha^2} e^{i\omega_0(x-x_0)}$ (okay to omit x_0).

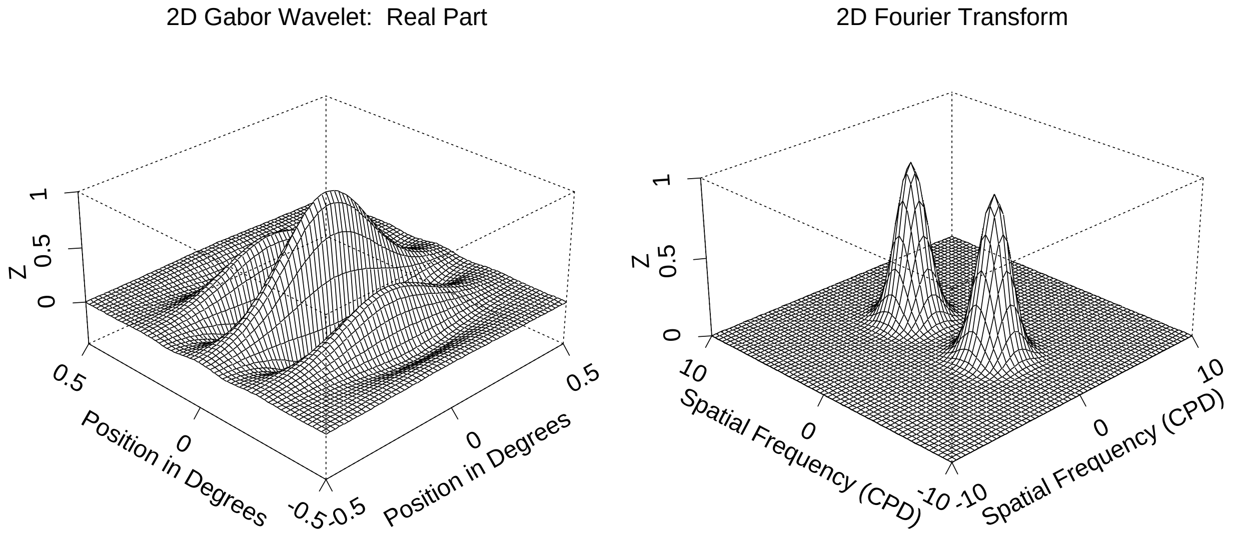
1. The second term is just a complex exponential with frequency ω_0, so it is a rotating phasor, but the first term puts a Gaussian envelope on its amplitude. Thus it is a kind of localised tone-burst with a well-defined epoch and frequency. In the complex coordinates shown it appears briefly as a spiral helix then is seen no more.

2. The Uncertainty Principle expresses a trade-off between how well a function may be specified in frequency and how well it may simultaneously be "localised" in the signal domain (say time or space). Gabor wavelets achieve the lower bound on joint uncertainty in these two domains, and in that sense they are optimal encoders of information.

3. The axes of the Information Diagram (as defined by Gabor) are time and frequency. The wavelets occupy elliptical areas in this plane, and they may be elongated along either axis or the other, but the area within these ellipses is always the same: $1/4\pi$. That area is the lower bound on joint uncertainty; hence their joint resolution is maximal.

4. Gabor wavelets are widely used for signal analysis, encoding, and recognition. Examples of such applications areas include audio "speech spectrograms" and, generalised to two-dimensional spatial form, image compression, visual encoding and pattern (e.g. iris) recognition.

(*b*) Explain why the real-part of a 2D Gabor wavelet has a 2D Fourier transform with two peaks, not just one, as shown in the right panel of the Figure below.



2D Gabor Wavelet: Real Part

2D Fourier Transform

Solution:

(*b*) The real-part of a 2D Gabor wavelet, as shown in the figure, has the functional form

$$f(x,y) = e^{-(x^2/\alpha^2 + y^2/\beta^2)} \cos(u_0 x + v_0 y)$$

But using the identity $\cos(\theta) = \frac{1}{2}(e^{i\theta} + e^{-i\theta})$ we see that the expression above is actually half the sum of two complex Gabor wavelets, namely

$$g(x,y) = e^{-(x^2/\alpha^2 + y^2/\beta^2)} e^{i(u_0 x + v_0 y)}$$

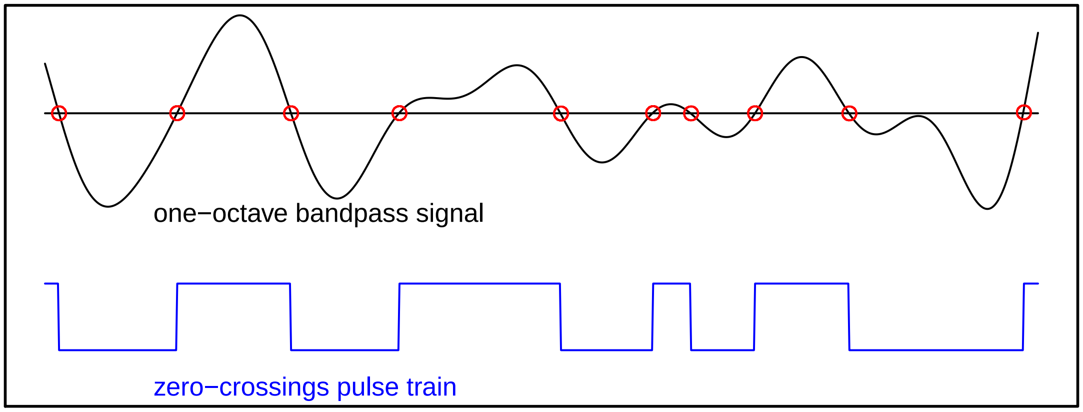
and

$$h(x,y) = e^{-(x^2/\alpha^2 + y^2/\beta^2)} e^{-i(u_0 x + v_0 y)}$$

These each have a 2D Gaussian Fourier transform, one centred at (u_0, v_0) and the other centred at $(-u_0, -v_0)$, added together as seen in the figure.

Exercise 12

(a) Explain Logan's Theorem about the richness of the zero-crossings in signals bandlimited strictly to one octave, such as illustrated in the figure below. Name one intended application, and at least one algorithmic difficulty, of Logan's Theorem.



Next, consider *instead* an amplitude-modulated signal such as $f(t) = [1 + a(t)]\, c(t)$, where $c(t)$ is a pure sinusoidal carrier wave and its modulating function is anything $[1 + a(t)] > 0$. What does Logan's Theorem say about the information contained in the zero-crossings of $f(t)$?

Solution

(a) Logan's theorem asserts that if a signal is bandpass limited to one octave, so that the highest frequency component W_H it contains is no higher than twice W_L its lowest frequency component, $W_H \leq 2W_L$, then merely listing the zero-crossings of this signal suffices for exact reconstruction of the signal (up to a scale factor in amplitude), even in the continuous intervals between those discrete zero-crossings.

An intended application was extreme compression of speech signals, to 1 bit, for telephony. The zero-crossings are where the MSB (sign bit) changes, since the waveform is bandpass; and indeed a pulse train of such "1-bit speech" is remarkably intelligible, albeit distorted.

Algorithmic difficulties include numerical instability: tiny changes in the position of an extracted zero-crossing have divergent effects on signal reconstruction. A further limitation is the inapplicability to computer vision. The zero-crossings of bandpass filtered *images* are not discrete and countable, as they are for a 1D signal, but they form continuous "snakes."

Obviously signals from (purely positive) amplitude-modulations of a carrier $c(t)$, of the form $f(t) = [1 + a(t)]c(t)$ where $[1 + a(t)] > 0$, will encode nothing about $a(t)$ in the zero-crossings of $f(t)$: they will remain always just the zero-crossings of the carrier $c(t)$. Thus, "pure AM" signals are excluded from Logan's theorem, by a condition that the signal must have no complex zeroes in common with its Hilbert transform.

(b) Compare and contrast the compression strategies deployed in the JPEG and JPEG-2000 protocols. Include these topics: the underlying transforms used; their computational efficiency and ease of implementation; artefacts introduced in lossy mode; typical compression factors; and their relative performance when used to achieve severe compression rates.

Solution

(b) The JPEG standard (published in 1994) uses the Discrete Cosine Transform (DCT) for image compression. It divides up an image into a regular uniform grid of non-overlapping small tiles, usually sized (8 x 8) pixels, and performs a DCT over each such tile. That operation in itself does not reduce the amount of data: (8 x 8) pixels would generate 64 DCT coefficients. However, compression arises because the DCT coefficients associated with high spatial frequencies can be quantised (resolved) much more coarsely than lower frequencies without incurring noticeable errors in image reconstruction. A *quantisation table* controls the relative severity of this truncation of coefficient resolution. Because many high frequency coefficients end up being 0 (after their quantisation into, say, just 4 possible values), their values are efficiently encoded by run-length (RLE) codes when high frequency coefficients are read out in a group, separate from lower frequencies. For most images (especially of natural scenes), compression factors of 10:1 are preceived as "almost lossless" (although they are in fact lossy). Both compression and decompression are easily implemented (the DCT is essentially just a subset of the FFT), and executed at video frame-rates. However, at more severe compression rates, blocking quantisation artefacts arise at the borders of each (8 x 8) tile which are very noticeable and objectionable. This is because the image structure within each DCT tile may be represented by just one, or just a few, chopped cosine waves.

JPEG-2000 by contrast uses the Discrete Wavelet Transform (DWT), with smoothly attenuating Daubechies wavelets. The absence of sharply truncated ("cliff edge") tile boundaries, and the overlapping, multi-scale nature of wavelet projections, avoids the block quantisation artefacts of the JPEG DCT. Therefore, JPEG-2000 performs very much better than JPEG at severe compression rates, such as 20:1 or 50:1. But despite the orthogonality of Daubechies wavelets, the implementation of the DWT in JPEG-2000 is more complex than the DCT, (for example, no closed-form analytic expression exists for the basis functions), and execution times are slower, both for compression and decompression. The use of the JPEG-2000 protocol is not yet commonplace, whereas JPEG is universally used.

(c) Define the Kolmogorov algorithmic complexity K of a string of data, and say whether or not it is computable. What relationship is to be expected between the Kolmogorov complexity K and the Shannon entropy H for a given set of data? Give a reasonable estimate of K for a fractal, and explain why it is reasonable. Discuss the following concepts in Kolmogorov's theory of pattern complexity: how writing a program that generates a pattern is a way of compressing it, and executing such a program decompresses it; Kolmogorov incompressibility, and patterns that are their own shortest possible description.

Solution:

(c) Kolmogorov defined the algorithmic (or descriptive) complexity K of a pattern (or sequence of data) as the length of the shortest binary program that could generate it or fully describe it. Thus, the data's Kolmogorov complexity is its "Minimal Description Length." Writing such a program amounts to "compressing" the pattern, if the program is shorter than the pattern; and executing that program "uncompresses" it, by recovering the pattern. A remarkable result is that Shannon entropy H and Kolmogorov complexity K for long random stings should be asymptotically roughly equal: $K \approx H$.

Because fractals can be generated by extremely short programs, namely iterations of simple recursive mappings, such patterns have Kolmogorov complexity of nearly $K \approx 0$ despite their apparently unlimited complexity across unlimited scales. Some patterns cannot be generated by programs except those that contain the pattern itself in a data statement; those are said to be Kolmogorov incompressible, and to be their own shortest possible description. A truly random sequence should be Kolmogorov incompressible. Unfortunately, finding the shortest possible description of a pattern is uncomputable (how could one be sure?).