

Course notes for Data Compression - 3
Basics of signal processing

Fall 2005

Peter Bro Miltersen

November 2, 2005

Version 1.1

1 Signals

We now begin our study of lossy data compression. Although one could in principle study lossy compression in a completely general way on arbitrary domains, the by far most practically important topic is the compression of *signals*.

In general, a signal is a map from a subset of \mathbf{R}^k to \mathbf{C}^l . For example, an analog audio signal can be represented as a waveform $f : \mathbf{R} \rightarrow \mathbf{R}$, where $f(t)$ is the amplitude of the audio signal at time t . A square color image can be represented by a map $f : [0, 1]^2 \rightarrow \mathbf{R}^3$, where $f(x, y)$ is the RGB-decomposition of the color value at position (x, y) of the image. Note that we allow signals to take complex values. This is not motivated by examples but will prove convenient for us when considering frequency domain representations of signals. We shall distinguish between two types of signals, *continuous* signals, where the domain of definition is all of \mathbf{R}^k or the Cartesian product of k intervals and *discrete* signals, where the domain of definition is \mathbf{Z}^k or a finite subset thereof. A signal is said to be of *finite duration*, if it has compact support, i.e., if it is 0 outside $[-m; m]^k$ for some value of m .

The object of lossy data compression of signals is to devise representations of signals as finite bit strings so that the original signals can be approximately reconstructed from the representations. Note that there is no hope of precisely reconstructing every given signal or even precisely reconstructing the value of every given signal at a single point, as we have only countably many finite bit strings but over-countably many real numbers. Thus, the lossiness of our compression schemes is not just a choice we make to save space but an inherent property of any scheme. Indeed, the “raw” data files that will usually be our starting point are already infinitely(!) compressed versions of the original data they are meant to represent. For instance, a raw WAV data file representing a continuous time audio signal $f : [0, 1] \rightarrow \mathbf{R}$ really holds a discrete sequence of values $(\tilde{f}(j))_{j=0,1,\dots,\lfloor 1/T \rfloor}$ for some value T , where $\tilde{f}(j)$ is only approximately equal to $g(j) = f(jT)$. Thus, before we even get our hand at the data, two lossy transformations have been made by the device capturing the signals:

- *Sampling*. The continuous time signal $f : [0, 1] \rightarrow \mathbf{R}$ has been replaced with the discrete time signal $g : \{0, 1, \dots, \lfloor 1/T \rfloor\} \rightarrow \mathbf{R}$. The value T is called the *sample rate* and the value $1/T$ is called the (rotational) *sample frequency*.
- *Quantization*. Each value $g(j)$ has been replaced with an approximate

or *quantized* value $\tilde{g}(j)$.

Even though these two transformations are usually outside our control, we shall begin by gaining an understanding of them. Such an understanding will prove useful for our task of further lossily compressing the raw (or arguably not so raw) signal \tilde{g} . In particular, we may want to redo the transformations with coarser parameters to achieve a better compression.

2 Sampling and the Frequency domain

For convenience, let us first concentrate on one-dimensional signals $f : \mathbf{R} \rightarrow \mathbf{R}$. When is such a signal f adequately represented by the discrete signal $g : \mathbf{Z} \rightarrow \mathbf{R}$ given by $g(i) = f(iT)$ for some value T ? Ideally, we should have that the original signal can be uniquely reconstructed from f (i.e., ideally the sampling should be lossless). Here, we shall state a theorem that gives us a sufficient condition for this property to be true.

To state the theorem, we need to recall the notion of transforming a signal to the frequency domain using the *Fourier transform* (we assume that the intuition behind the transform is known from previous courses).

Definition 1 Given integrable $f : \mathbf{R} \rightarrow \mathbf{C}$, its *Fourier transform* or *spectrum* $\hat{f} = \text{FT}[f]$ is the map $\hat{f} : \mathbf{R} \rightarrow \mathbf{C}$ given by

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t)e^{-it\omega} dt$$

The value i is the imaginary number¹ $i = \sqrt{-1}$.

Conversely, given integrable $\hat{f} : \mathbf{R} \rightarrow \mathbf{C}$, the inverse Fourier transform $f = \text{IFT}[\hat{f}]$ is the map $f : \mathbf{R} \rightarrow \mathbf{C}$ given by

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \hat{f}(\omega)e^{it\omega} d\omega.$$

Remark: In the signal processing community the constant $\frac{1}{\sqrt{2\pi}}$ in the definition of FT is often replaced with the constant 1 and the constant $\frac{1}{\sqrt{2\pi}}$ in the

¹For this reason, we will try very hard not to use the symbol “ i ” as an index to a sequence in these section, though it is very hard not too! To add to the confusion, the signal processing community often use the symbol “ j ” rather than “ i ” to denote the imaginary unit $\sqrt{-1}$.

definition of IFT consequently replaced with the constant $\frac{1}{2\pi}$. This scaling issue is of course of no importance as long as one remembers which definition one is using! We prefer the definitions above for two reasons: they make the transforms norm-preserving and they are easier to remember!

For nice² functions f , we have $f = \text{IFT}[\text{FT}[f]]$. If $\hat{f} = \text{FT}[f]$ and $\hat{f}(\omega) = v$ we say that the value of the spectrum of f at *angular frequency* ω (radians) is v . We may prefer to use *rotational* frequencies h rather than angular frequencies ω and convert between the two using the formula $\omega = 2\pi h$. Thus, the value of the spectrum of f at rotational frequency h is $\hat{f}(2\pi h)$.

We are now ready to state the *Shannon-Nyquist sampling theorem*.

Theorem 2 *Let f be a nice function with spectrum \hat{f} satisfying $\hat{f}(\omega) = 0$ for $\omega \notin [-\pi/T; \pi/T]$, i.e., the value of the spectrum of f at all rotational frequencies of absolute value bigger than $\frac{1}{2T}$ is 0. Then, f can be uniquely reconstructed from the discrete values $\{f(kT)\}_{k \in \mathbf{Z}}$ using the formula*

$$f(t) = \sum_{k \in \mathbf{Z}} f(kT) \frac{\sin(2\pi(\frac{t}{2T} - \frac{k}{2}))}{2\pi(\frac{t}{2T} - \frac{k}{2})} \quad (1)$$

with the convention $\sin(0)/0 = 1$.

The elegant proof of the theorem is beyond the scope of these notes. The theorem tells us two things:

1. A signal with maximum absolute rotational frequency $1/T$ should be sampled at a rate of at most $T/2$. This is called the *Nyquist rule*.
2. An apparatus synthesising a signal $f : \mathbf{R} \rightarrow \mathbf{R}$ from a sequence of samples $f(iT)$ should ideally do so using equation (1).

Unfortunately, a real-life analog signal is likely to contain components of arbitrarily high frequency or at least too high frequency for the theorem to be directly applicable. If a signal containing a rotational frequency of absolute value bigger than $1/T$ is sampled at rate below $T/2$ and reconstructed using equation 1, we get the unfortunate phenomenon of *aliasing*: The high-frequency component appears at a low-frequency component in the reconstructed signal which can be seen by a direct calculation. In particular,

²We shall not be rigorous about what we mean by nice in this section. Functions f which are continuous and integrable are certainly nice, but we shall also take the transform on some non-continuous functions without worrying too much about it.

the rotational frequency $1/T + \epsilon$ for a small value ϵ in the original signal will appear as a frequency $1/T - \epsilon$ with the same amplitude in the reconstructed signal. This is a very real phenomenon which will be noticeable in almost any sampling situation if explicit measures are not taken to avoid it. An example is the wheel of a rolling wagon appearing to be going backwards in old Western movies as seen on TCM on Sunday afternoons: The explanation is that the frame rate of the movie is too low compared to the rotational frequency of the wheel. In the next section, we see how to avoid such phenomena.

3 Anti-Aliasing and filters

Suppose that we are bound to sample at rate T . To avoid aliasing we need to remove all high frequency components from the signal before sampling. Formally, if our original signal is f and the spectrum of f is \hat{f} , we should define $\hat{g}(\omega) = \hat{f}(\omega)\hat{h}(\omega)$ where h is the function so that $\hat{h}(\omega)$ is 1 inside $[-1/2T, 1/2T]$ and 0 outside $[-1/2T, 1/2T]$. We should then compute the inverse Fourier transform g of \hat{g} and replace f with g . A nice fact about the Fourier transform is that if we let h be the inverse Fourier transform of \hat{h} , we have that $\hat{g} = \hat{f}\hat{h}$ is equivalent to $g = f * h$ where $f * h$ denotes *convolution* of f and h , i.e.,

$$f * h(t) = \int_{-\infty}^{\infty} f(t - u)h(u) du.$$

Thus, we can compute g by first computing $h = \text{IFT}(\hat{h})$ and then convoluting f with h . As the computation is to be done before sampling, it has to be carried out by an analog device. Note that since $\hat{h}(\omega) = \hat{h}(-\omega)$ for all ω , we have that h and hence g is a real-valued signal - which is an important sanity check for the entire procedure.

We refer to the process of replacing f by $f * h$ as a *filtering* of f and when using a real function h in this way, we call h a *filter*. In this case, h is a filter that allows exactly the low-frequency components of f to pass through. Such a filter is called a *low-pass filter*. By letting \hat{h} be the indicator function for an arbitrary set of the form $[-v, -u] \cup [u, v]$ for non-negative values $u < v$, we could have similarly constructed a filter h letting any particular absolute range of frequencies pass through. Such a filter is called a *band-pass filter*. A filter letting exactly the high-frequency components of f pass through is called a *high-pass filter*.

The filters above are *ideal* in the sense that they let exactly the desired range of frequencies pass through. However, they have a somewhat annoying property: Even if the original signal f is a finite-duration signal, the filtered signal

g may not be. Indeed, with \hat{h} being the indicator function on $[-1/2T; 1/2T]$, h is not finite-duration and the convolution of f and h is hence quite unlikely to be finite-duration. Also, we can't compute the value of the convolution at a given point without knowing the function f at all points. A filter h without compact support is called an *infinite impulse response filter* or IIR filter. A filter with compact support (and hence turning finite duration signals into finite duration signals) is called a *finite impulse response filter* or FIR filter. An FIR filter cannot be ideal, but we may want to use it rather than an ideal filter because of the FIR property. A low-pass FIR filter h could not have \hat{h} being exactly the indicator function for the low-range frequencies but it is possible to construct filters h for which \hat{h} is a good approximation to this function and with finite (and even short) impulse response, meaning that the duration of the filtered signal will only be slightly longer than the original signal. Also, we can find the value of the filtered signal at a point without knowing more than a neighborhood of the original signal at the same point.

4 Sampling and filtering in higher dimensions

The notions of sampling, spectra and filtering carries over to higher dimensions in a fairly obvious way. In particular, we can define the Fourier transform of a higher dimensional signal $f : \mathbf{R}^k \rightarrow \mathbf{C}$ in the following way.

Definition 3 Given integrable $f : \mathbf{R}^k \rightarrow \mathbf{C}$, its *Fourier transform* or *spectrum* $\hat{f} = \text{FT}[f]$ is the map: $\hat{f} : \mathbf{R}^k \rightarrow \mathbf{C}$ given by

$$\hat{f}(\omega) = \frac{1}{(2\pi)^{k/2}} \int_{\mathbf{R}^k} f(x) e^{-it\langle\omega, x\rangle} dx,$$

where $\langle\omega, x\rangle$ is the inner product of the vectors ω and x . The inverse transform is defined analogously.

It is easy to see that we may compute the higher dimensional Fourier transform of a two-dimensional signal $f(x, y)$ by first applying the one-dimensional transform in the x -direction, i.e., letting $g_y(x) = f(x, y)$ and finding $\hat{g}_y = \text{FT}(g_y)$ and then applying the one-dimensional transform in the y -direction, i.e., letting $h_\omega(y) = \hat{g}_y(\omega)$ and finding $\hat{h}_\omega = \text{FT}(h_\omega)$. Then, the two-dimensional transform of f is then $\hat{f}(\omega, \sigma) = \hat{h}_\omega(\sigma)$. This generalizes to higher dimensions in the obvious way. The order in which the one-dimensional transforms are taken doesn't matter.

With this definition, we get a higher dimensional version of the Shannon-Nyquist theorem completely analogous to the one-dimensional one: If $f : \mathbf{R}^k \rightarrow \mathbf{C}$ has the support of \tilde{f} contained within $[-T/2; T/2]^k$, we can completely reconstruct f from the values $f(T\mathbf{Z}^k)$. If not, we should apply a suitable k -dimensional low-pass filter to f before sampling.

5 Quantization and distortion measures

Let us assume that we have sampled our continuous signal $f : [0, 1] \rightarrow \mathbf{R}$ and obtained a discrete, finite-duration signal $g : \{0, \dots, p-1\} \rightarrow \mathbf{R}$ in a satisfactory way. To construct a raw data file representing the signal, we are allowed to use a fixed number b of bits (say, $b = 8$ or $b = 16$) to represent each sample $g(j)$. This is done using a *scalar quantization scheme*. A scalar quantization scheme is simply a sequence $y_0 < y_1 < \dots < y_{2^b-1}$ of 2^b real numbers, called the *reconstruction levels* of the quantization scheme. The reconstruction levels are the only values that samples in the raw file may have, each being represented by a unique b -bit pattern. The actual values $g(j)$ thus have to be rounded, or *quantized* to the nearest y_k . This quantized value is denoted $\tilde{g}(j)$. Doing this for all samples $g(j)$ we obtain the *quantization* \tilde{g} of the signal g which we can then finally store on a digital computer.

The signal $d = \tilde{g} - g$ is called the *quantization noise* or *quantization distortion*. In general, we want to make this noise as small as possible, so we need a way of measuring it. We do this by the *signal-to-noise ratio*, defined by

$$\text{SNR} = \frac{\sum_{j=0}^{p-1} (g(j) - \bar{g})^2}{\sum_{j=0}^{p-1} d(j)^2}$$

where \bar{g} is the average value of the entries $g(j)$. In general, we hope $d(j) \ll |g(j) - \bar{g}|$ so the signal-to-noise ratio will usually be a real number much bigger than one, and the bigger the better. The measure is invariant under affine transformations of the signal. The convention is to measure its magnitude on a logarithmic scale, in the unit *decibels* or dB, using the formula

$$\text{SNR (in dB)} = 10 \log_{10} \frac{\sum_{j=0}^{p-1} (g(j) - \bar{g})^2}{\sum_{j=0}^{p-1} d(j)^2}.$$

The definition of SNR above can be used to measure the distortion introduced when replacing any signal g with any approximation \tilde{g} . Indeed, as we usually

do not have access to the original signal but only the “raw” file representing it, we shall hardly ever use it with g being the original sampled signal and \tilde{g} being the “raw” quantized data file. Instead, we shall use it to measure the distortion introduced when further compressing the raw data file. Thus, we shall use the definition with g being the raw (quantized) data file and \tilde{g} the result of compressing and decompressing g using a lossy compression scheme. In this setting, an alternative, *peak-signal-to-noise ratio* is often seen used:

$$\text{PSNR} = \frac{p \cdot y_{2^b-1}^2}{\sum_{j=0}^{p-1} d(j)^2}$$

where y_{2^b-1} is the reconstruction value of biggest magnitude. For instance, in a gray scale image such as Lena where the reconstruction levels are $\{0, 1, \dots, 255\}$, $y_{2^b-1} = 255$. Note that the SNR penalizes noise in low-intensity images higher than noise in high-intensity images, while the PSNR does not. As the SNR, the PSNR is usually measured in dB.

$$\text{PSNR (in dB)} = 10 \log_{10} \frac{p \cdot y_{2^b-1}^2}{\sum_{j=0}^{p-1} d(j)^2}.$$