THE STATE UNIVERSITY OF NEW JERSEY
RUTGERS

College of Engineering
Department of Electrical and Computer Engineering

QUANTIZATION NOTES
C. Rose

1 Optimal Quantization

We start with a signal $x(t)$ and *perfectly* sample it (at the Nyquist Rate or better) to obtain a sequence of real numbers $\{x_k\}$. We take this sequence and *quantize* it to obtain the approximate values $\{\tilde{x}_k\} = \{Q(x_k)\}$.

Quantization is necessary because we cannot store sampled values with infinite precision. Please note that this fact has little to do with digital computers and wordlengths, etc. The physical fact is that no matter what the measurement/sampling procedure, there is some uncertainty. And this uncertainty (usually called the granularity) limits the precision with which we can specify the measured values.

For example, do you weigh 50kg or 50.0000000001kg? It is usually sufficient to measure the weight of people to the nearest kg or so.

Now we need a way to specify the quantization function $Q(x)$. More precisely, we need a way to measure the "goodness" of any given $Q(x)$ for any given set of samples $\{x_k\}$. To this end we introduce the quantization error function (the usual square difference) $e_k^2 = (x_k - Q(x_k))^2$ and then average over a large number of samples to obtain the mean square error

$$\bar{e^2} = \frac{1}{N}\sum_{k=1} N(x_k - Q(x_k))^2$$

Well, for N large enough, the summation begins to look like an expectation with respect to the variable x_k. The basic idea is to form a "relative frequency function" (aka probability function) for the variable x_k. We call this function $f_X(x)$ and our mean square error becomes

$$\bar{e^2} = E_X[(x - Q(x)^2] = \int_{-\infty}^{\infty} f_X(x)((x - Q(x))^2 dx$$

What's next? Well we're interested in finding $Q(x)$ which minimizes the mean square error. To do this we note that a quantization function takes values of x and "bins them". That is, it takes values in a given range, (say $a_1 < x < a_2$) and maps those values to a single value (q_2). There are a finite number of quantization levels (if this were not so, then how could we represent and store these values?).

So, we formally define an N-level quantizer $Q(x)$ as follows:

$$Q(x) = \begin{cases} q_1 & x < a_1 \\ q_i & a_{i-1} < x \leq a_i \\ q_N & x > a_{N-1} \end{cases}$$

We now have $2N-1$ parameters to choose in our minimization of the mean square error: $\{a_1, ...a_{N-1}\}$ and $\{q_1, ...q_N\}$.

In general, multivariate optimization is a difficult task. However, we're going to close our eyes and assume that simply finding where the first partials are zero with respect to our variables will provide a solution to the error minimization. That is, we seek a set of parameters $\{a_1, ...a_{N-1}\}$ and $\{q_1, ...q_N\}$ such that

$$\frac{\partial e^2}{\partial a_i} = 0$$

$i = 1, 2, ...N-1$ and

$$\frac{\partial e^2}{\partial q_i} = 0$$

$i = 1, 2, ...N$

Since e^2 is an integral, we'll need Liebnitz' formula:

$$\frac{d}{dx}\int_{w(x)}^{z(x)} g(x,t)dt = \frac{dz(x)}{dx}g(x,z(x)) - \frac{dw(x)}{dx}g(x,w(x)) + \int_{w(x)}^{z(x)} \frac{\partial g(x,t)}{\partial x}dt$$

After taking derivatives we end up with:

$$(a_i - q_i)^2 = (a_i - q_{i+1})^2$$

Expanding both sides and rearranging we obtain

$$a_i = (q_i + q_{i+1})/2$$

In words – the bin cutoff a_i is the AVERAGE of the surrounding quantization levels. Lovely, isn't it?

We also have:

$$-2\int_{-\infty}^{a_1}(x - q_1)f_X(x)dx = 0$$

$$-2\int_{a_i}^{a_{i+1}}(x - q_{i+1})f_X(x)dx = 0$$

and

$$-2\int_{a_{N-1}}^{-\infty}(x - q_N)f_X(x)dx = 0$$

But we note that these can be rewritten as

$$\int_{-\infty}^{a_1} x f_X(x)dx = q_1\int_{-\infty}^{a_1} f_X(x)dx$$

$$\int_{a_i}^{a_{i+1}} x f_X(x)dx = q_{i+1}\int_{a_i}^{a_{i+1}} f_X(x)dx$$

and

$$\int_{a_{N-1}}^{-\infty} x f_X(x)dx = q_N\int_{a_{N-1}}^{-\infty} f_X(x)dx$$

Then we note that $f(x|x \in (c,d)) = f(x)/\text{Prob}(x \in (c,d))$. We then have:

$$\int_{-\infty}^{a_1} x f_X(x)dx = q_1\text{Prob}(x \in (-\infty, a_1))$$

$$\int_{a_i}^{a_{i+1}} x f_X(x) dx = q_{i+1} \text{Prob}(x \in (a_i, a_{i+1}))$$

and

$$\int_{a_{N-1}}^{-\infty} x f_X(x) dx = q_N \text{Prob}(x \in (a_{N-1}, \infty))$$

Rearranging we then have

$$q_1 = E_X[x|x < a_1]$$

$$q_{i+1} = E_X[x|a_i < x \leq a_{i+1}]$$

and

$$q_N = E_X[x > a_{N-1}]$$

The quantization levels are the CONDITIONAL MEANS of x in the quantization interval!!!!!

The two conditions, (conditional means for the q_i and the mean relationship between a_i and q_i, q_{i+1} are together called the Loyd-Max conditions. These conditions are necessary (BUT NOT SUFFICIENT) for any solution to the mean square quantization error minimization problem. The reason for insufficiency is that without evaluating a nasty expression involving the cross partials, we have no way of knowing whether the extremal point we find when setting first partials to zero is 1) a max or a min, and 2) whether it's unique.

But we usually just accept on faith that if we can find sets $\{q_i\}$ and $\{a_i\}$ which satisfy Loyd-Max we've found the minimum.

You should try to prove on your own that "Loyd-Max" implies if the probability distribution (relative frequency of samples) of a function is symmetric about zero, then the quantizer function levels have odd symmetry about zero and that the quantizer bins sizes are symmetric about zero. How about if the function $f_X(x)$ is symmetric about it's mean? Is $Q(x)$ symmetric? Where?

2 Quantization Noise

Let's take a peek at how dynamic range (amplitude range over which the signal $x(t)$ runs) affects quantization. Let's assume for simplicity that our signal $x(t)$ is bounded in amplitude between $\pm A$. Now we assume that our quantizer bins are all the same size Δ. This is called a uniform quantizer, by the way. With both bin size and quantizer step size $\Delta q = q_{i+1} - q_i \; \forall i$, it's also called a uniform quantizer but this latter one is "more" uniform the the previous.

Well, let's assume that our quantizer bins are small enough that GIVEN x falls into a particular bin (a_i, a_{i+1}), the probability distribution of x over that bin is essentially uniform

Well, that means that the difference between x and it's quantized value $Q(x)$ is going to be a continuous uniform random random variable on $(-\Delta/2, \Delta/2)$. That is, for sample x_k, the error e_k is a uniformly distributed random variable on the interval $(-\Delta/2, \Delta/2)$.

So, now let's look at the energy/power in our signal x_k. This is $E[x_k^2] \equiv P_x$. Now consider the energy/power in the error signal. This will be $E[e_k^2]$, but since e_k is uniformly distributed with zero mean we have $E[e_k^2] = \Delta^2/12$.

Now, the signal $x(t)$ runs between $\pm A$ and let's assume we have $N = 2^b$ quantization bins. This means $\Delta = 2A/2^b$. We can consider v the number of quantization bits in our A/D, if you like.

Well, a useful measure of fidelity between the quantized signal and the original signal is signal power divided by the quantization noise power ($E[e_k^2]$). This is usually called the signal to quantization noise ratio or SQNR for short.

We have

$$\text{SQNR} = P_x/\Delta^2/12 = \frac{3 \cdot 2^{2b} P_x}{A^2}$$

Here are some punchlines we can take from this expression: We'll assume fixed signal power, P_x

- For given amplitude variation (A), also called dynamic range, if you increase the number of bins in your quantizer (increase b) your SQNR goes up quickly (geometrically in b).
- For fixed b, if the signal is "peaky" (large dynamic range A but mostly stays small so that power does not exceed P_x) then your SQNR gets shot in the foot (as A^2) as the dynamic range increases.

So, that's why people kept striving to get wider (more bits) A/D converters. The SQNR drops quickly. This is also why music signals are devilish. You're going along listening to a soft flute and then a cymbal crash smashes in on you or the horns come in.

3 Companding

Uniform quantizers are easier to design and build than non-uniform ones. But most signals (especially music and voice signals) do not have uniform distributions on amplitude. They cover a wide range of amplitudes but dwell mostly at moderate values. But we'd still like to use uniform quantizers because they're cheap and plentiful.

The way we usually get around this problem is by optimally "companding" signals. For music, we COMPRESS the signal where our ears don't care much and make sure they're very carefully represented in ranges where we do care. That is, the EXACT reproduction of that infrequent cymbal crash is probably not as important as to have a crystal clear sound for the normal amplitude levels (i.e., strings, but not LOUD electric guitar riffs). We then apply a uniform quantizer to the compressed signal. This is EXACTLY equivalent to optimally designing a quantizer if the compression is optimally designed.

To recover the signal the inverse quantizer (D/A) is applied and then the inverse of the compression (EXPANSION) is applied – thus the name "companding". Some typical companding functions for music and voice follow. We always assume the that maximum amplitude of the signal is known and the waveform is normalized by the amplitude. Thus, $|x| < 1$.

- μ-law

  $$g(x) = \frac{\log(1+\mu|x|)}{\log(1+\mu)}\text{sgn}(x)$$

  where μ is the companding factor. For small signal amplitudes $g(x)$ is basically linear in x but saturates as x grows.

- A-law

  $$g(x) = \begin{cases} \dfrac{A|x|}{\log(1+A)}\text{sgn}(x) & 0 \le |x| \le 1/A \\ \dfrac{1+\log(A|x|)}{\log(1+A)}\text{sgn}(x) & 1/A \le |x| \le 1 \end{cases}$$

  where A is a factor similar to μ in μ-law. For A-law, the linear region is precisely defined.

For larger A and μ, the compression functions are similar in shape, but for smaller values of A and μ ($A = 2$ A-law has a sharp breakpoint.