EE5585 Data Compression	April 25, 2013
Lecture 25	
Instructor: Arya Mazumdar	*Scribe: Zhongyu He*

1 Wavelet Decomposition

Suppose we have a signal as the figure 1. What we want to do is to approximate the signal.

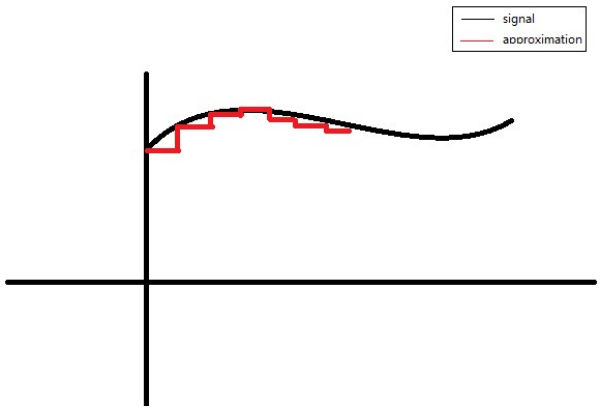

Figure 1: Approximated Signal.

One way of doing that is to take a function as figure 2 and use the shifted version of it. This is the simplest thing we can do. We can have this piecewise linear approximation as stated in red in figure 1. This also captures the quantization processes. The signal between two points will be quantized to a certain value.

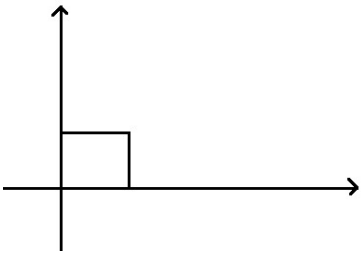

Figure 2: Block Wavelet

The signal in figure 2 is as following

$$\phi(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & \texttt{otherwise} \end{cases}$$

The shifted version of this signal will be

$$\phi(t-k), k = 0, 1, \cdots$$

Using the shifted version of the function in figure 2, we can approximate the signal in figure 1 in some resolution. However, this function has very low resolution. Between 0 and 1 we have only 1 value. If we want to approximate the signal better, we need to scale the function.

The shifted scaled function will be as follows:

$$\phi(2t-k) = \begin{cases} 1 & 0 \le 2t-k < 1 \\ 0 & \texttt{otherwise} \end{cases} = \begin{cases} 1 & \frac{k}{2} \le t < \frac{k+1}{2} \\ 0 & \texttt{otherwise} \end{cases}$$

Now, with the new function above, we double the resolution. We can improve the resolution by scaling the function more as above. From the Nyquist rule, once the frequency is twice of the maximum frequency of the original signal, the signal can be recovered without any lose.

Now this function defined as:

$$\phi_{j,k}(t) = \phi(2^j t-k) = \begin{cases} 1 & 0 \le 2^j t-k < 1 \\ 0 & \texttt{otherwise} \end{cases} = \begin{cases} 1 & \frac{k}{2^j} \le t < \frac{k+1}{2^j} \\ 0 & \texttt{otherwise} \end{cases}$$

Use the above function to approximate the signal as stated in red in figure 1.

Then we have the 0^{th} order approximation. The coefficient C are average value of f between k and k+1.

$$\begin{aligned} \phi_f^0(t) &= \sum_k C_{0,k}\phi_{0,k}(t) \\ C_{0,k} &= \frac{1}{1}\int f(t)\phi_{0,k}(t)dt \\ &= \int_k^{k+1} f(t)dt \end{aligned}$$

This is actually a very low resolution approximation of the signal we have. In order to get high resolution, we can do higher order approximation.

1^{st} order approximation:

$$\begin{aligned} \phi_f^1(t) &= \sum_k C_{1,k}\phi_{1,k}(t) \\ C_{1,k} &= \frac{1}{1/2}\int f(t)\phi_{1,k}(t)dt \\ &= 2\int_{k/2}^{(k+1)/2} f(t)dt \\ C_{1,2k} &= 2\int_k^{k+1/2} f(t)dt \\ C_{1,2k+1} &= 2\int_{k+1/2}^{k+1} f(t)dt \\ C_{1,2k} + C_{1,2k+1} &= 2\int_k^{k+1} f(t)dt = 2C_{0,k} \\ \Rightarrow C_{0,k} &= \frac{1}{2}[C_{1,2k} + C_{1,2k+1}] \end{aligned}$$

We can go further and further. 2^{nd} order approximation:

$$\phi_f^2(t) = \sum_k C_{2,k}\phi_{2,k}(t)$$
$$C_{2,k} = \frac{1}{1/4}\int f(t)\phi_{2,k}(t)dt$$
$$= 4\int_{k/4}^{(k+1)/4} f(t)dt$$
$$\vdots$$
$$\phi_f^j(t) = \sum_k C_{j,k}\phi_{j,k}(t)$$
$$C_{j,k} = 2^j \int_{k/2^j}^{(k+1)/2^j} f(t)dt$$

We can do the above procedures until we get the resolution we want. Suppose we are representing in 1^{st} order approximation.

$$\phi_f^1(t) = \sum_k C_{1,k}\phi_{1,k}(t)$$

Such representation can be split into two lower resolution parts:

$$\phi_f^0(t) = \sum_k C_{0,k}\phi_{0,k}(t), \quad C_{0,k} = \frac{1}{2}[C_{1,2k} + C_{1,2k+1}]$$
$$\texttt{and} \qquad \phi_f^1(t) - \phi_f^0(t)$$

We can see that higher resolution signal can be represented by lower resolution ones. Let's look at the interval from l to l+1.

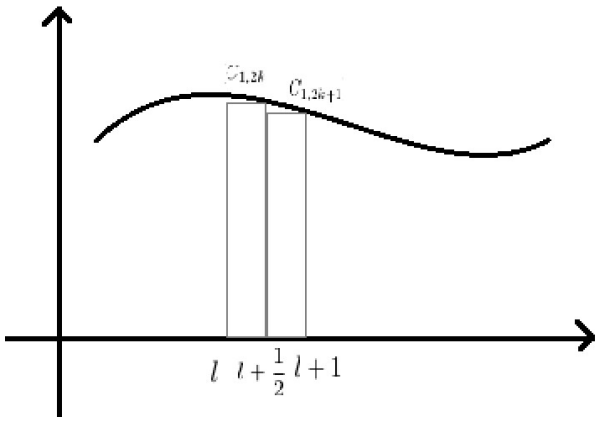

Figure 3: Interval from l to l+1.

We have:

$$\phi_f^1(t) - \phi_f^0(t) = \begin{cases} C_{1,2l} - C_{0,l} & l \le t < l + \frac{1}{2} \\ C_{1,2l+1} - C_{0,l} & l + \frac{1}{2} \le t < l + 1 \end{cases} = \begin{cases} \frac{1}{2}[C_{1,2l} - C_{1,2l+1}] & l \le t < l + \frac{1}{2} \\ -\frac{1}{2}[C_{1,2l} - C_{1,2l+1}] & l + \frac{1}{2} \le t < l + 1 \end{cases}$$

Define a function ψ:

$$\psi(t) = \begin{cases} 1 & 0 \le t < \frac{1}{2} \\ -1 & \frac{1}{2} \le t < 1 \end{cases}$$

Then we have

$$\begin{aligned} \phi_f^1(t) - \phi_f^0(t) &= \frac{1}{2}[C_{1,2l} - C_{1,2l+1}]\psi(t-l) \\ &= \sum b_{0,k}\psi_{0,k}(t) \\ \texttt{Where} \quad b_{0,k} &= \frac{1}{2}[C_{1,2k} - C_{1,2k+1}] \\ \psi_{0,k}(t) &= \psi(t-k) \end{aligned}$$

Now we can see that coefficients of lower resolution parts can be represented by coefficients of higher resolution parts. And the wavelet par $\psi(t)$ is called Haar Wavelet.

Depending on what kind of signal we want to approximate, blocks function may not be the best choice. Functions like figure 4 may be the optimal wavelet to approximate some signals.

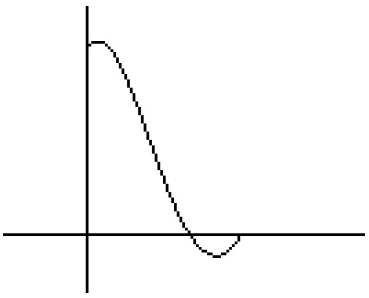

Figure 4: $\phi(t)$

Then we have shifted: $\phi(t-k)$, and scaled: $\phi(2^j t - k) = \phi_{j,k}(t)$. Then we will represent the signal with whatever resolution we need with these functions.

Again, as what we did above:

$$\begin{aligned} \phi_f^j(t) &= \sum C_{j,k}\phi_{j,k}(t) \\ C_{j,k} &= \frac{< f, \phi_{j,k} >}{< \phi_{j,k}, \phi_{j,k} >} \\ &= \frac{\int f(t)\phi_{j,k}(t)dt}{\int (\phi_{j,k}(t))^2 dt} \end{aligned}$$

Also $\phi_{j,k}(t)$ can be split into two parts. $\phi_{j-1,k}(t)$ and $\phi_{j,k}(t) - \phi_{j-1,k}(t)$. We can go further if $\phi(t)$ is known.

2 Data Compression Course Frame

What have we learn from this course? What to do with that? That is Data Compression. Data compression is not only compression but also recovery, because if you don't have a decoding algorithm, then you have no use to compress the data.

Therefore, there are two kinds of techniques base on the recovery that we can give. One is lossless recovery. The other one is lossy recovery.

2.1 Lossless Recovery

What is the limit of lossless compression? There are two parallel things.

Given a data string with unknown distribution, the best compression we can get is from ***Kolmogorov Complexity***. On the other hand, if the distribution of a random variable is known to us or we can try to model the data as a random variable, we can not go beyond the ***Entropy*** in compression.

So in some way, the Kolmogorov Complexity is more general than the Entropy.

Now there are several coding procedure that can take you to the value of Entropy. With these methods, we can get to the value of Entropy very closely, if not exactly.

2.1.1 variable length coding

1. Distribution is Known
Huffman Coding
Shannon Coding
Arithmetic Coding
In this part, we also have the notions of *Unique Decodability* and *Prefix free code*. All the codes are both unique decodable and prefix free code. We also discuss research problems on *fix free code*. 2. Universal Coding Lempel-Ziv Algorithm

2.1.2 Fixed to fix length coding

This is a special case when distortion is equal to zero.

2.2 Lossy Coding

More general case come from the Lossy coding when distortion is not zero. And the theory is given by ***Rate-Distortion***. We know this distortion can be the worst case and the average case. The Rate Distortion theory tells us that the compression algorithm we're using is lossy with some error. But given a certain amount of error, we can get the rate of compression we can achieve.

Then we look at ***Quantizers***. This is one important way to achieve this trade off between compression rate and rate distortion.

2.2.1 Scalar Quantizers

Uniform Quantizers
Optimal Quantizers.(We have to look at Lloyd's Algorithm)
In most of the cases, scalar quantizers are far away from what Rate Distortion theory promises. The solution is to go to Vector Quantizers.

2.2.2 Vector Quantizers

Basically in Vector Quantizers, we are doing ***clustering***.
We have multiple kinds of algorithms like ***Linde Buzo Gray*** and ***K-means-Clustering***.

There are two things the Vector Quantizers exploit.

1. Shape Gain

Most of the Vector Quantizers try to use rectangles to shape the decision region. This may not be the

best way to quantize but it's the most prominent way in Vector Quantizers.

2. Correlation between Data.

We saw the example of Height-Weight. This example consist of random variables but they are highly correlated as in figure 5. We can draw a regress line and turn it into 1-dimensional quantization.

Figure 5: Height-Weight

When you have correlated data, one way to do that is to rotate the axes. That gives you another tool to go the rate-distortion functions apart from the quantizers.

2.2.3 Transform coding

In transform coding, you basically rotate the axes by using a linear operator. In this case, just multiply the data by a transform matrix to transform the data to another orthonormal axes.

1. Karhunen-Loeve transform

If we want to minimize the distortion, the best thing to do is Karhunen-Loeve transform, which depends on the data. The process of doing this transform is called Principle Component Analysis

If we want to find some algorithm that is independent of the data, we can look at the following.

2. DCT or DFT.

These are standard transform coding techniques working well for a very large classes of signals.

Now that we use transform coding to get rid of many dimensionality, why do we need to acquire that many signal? This leads to the idea of sensing of the signals. That's ***Compressed Sensing***.

3. Compressed Sensing.

This time, the transform matrix is not a square matrix but a rectangular matrix. So we need to look at the ***sufficient condition*** for existence of such sampling schemes. And then we have one decoding algorithm called ***Basic Pursuit*** that does the recovery. And there is Restricted Isometry Property for this scheme.

If we want to deal with a larger width of signals, we need some more techniques as follows.

4.Predictive Coding

5.Differential Coding(DPCM)

6.Subband Coding
We will separate out this different signals into wavelets. The most prominent wavelet is Haar Wavelet.

Above are the basic building blocks of this course. Most of the applications will be a combination of multiple techniques. They are highly correlated with each other.

3 Homework 4 Solution

Problem 1 Consider the following matrix ϕ:

$$\begin{bmatrix} 1 & 0 & 0.5 \\ -2 & 1 & 1 \end{bmatrix}$$

We observe samples of an 1-sparse vector x

$$y = \phi x = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

Find out x.
Next, suppose we observe:

$$z = \phi x = \begin{bmatrix} 10.1 \\ -20.1 \end{bmatrix}$$

For a general vector x. Find out x using l1-minimization (basis pursuit).
Solution:
Question 1: Due to the linear independence, such problem has a unique solution. A universal way to solve is to assume:

$$x_1 = \begin{bmatrix} a \\ 0 \\ 0 \end{bmatrix}, x_2 = \begin{bmatrix} 0 \\ a \\ 0 \end{bmatrix}, x_3 = \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix}, \text{ Then we will get the solution } x = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$$

Question 2: assume

$$x = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$\begin{aligned} & min : \|x\|_{l_1} \\ & s.t. : \phi x = z \\ \Rightarrow & min : |a_1| + |a_2| + |a_3| \\ & s.t. : a_1 + 0.5a_3 = 10.1 \\ & \quad -2a_1 + a_2 + a_3 = -20.1 \\ \Rightarrow & min : |10.1 - 0.5a_3| + |0.1 - 2a_3| + |a_3| = f \end{aligned}$$

Case1: $a_3 > 20.2$

$$f = 0.5a_3 - 10.1 + 2a_3 - 0.1 + a_3 = 2.5a_3 - 10.2$$
$$\Rightarrow min.f = 2.5 \times 20.2 - 10.2 = 60.5, \quad when \quad a_3 = 20.2$$

Case2: $0.05 < a_3 < 20.2$

$$f = 10.1 - 0.5a_3 + 2a_3 - 0.1 + a_3 = 2.5a_3 + 10$$
$$\Rightarrow min.f = 2.5 \times 0.05 + 10 = 10.125, \quad when \quad a_3 = 0.05$$

Case2: $0 < a_3 < 0.05$

$$f = 10.1 - 0.5a_3 + 0.1 - 2a_3 + a_3 = -1.5a_3 + 10.2$$
$$\Rightarrow min.f = 10.125, \quad when \quad a_3 = 0.05$$

Case2: $a_3 < 0$

$$f = 10.1 - 0.5a_3 + 0.1 - 2a_3 - a_3 = -2.5a_3 + 10.2$$
$$\Rightarrow min.f = 10.2, \quad when \quad a_3 = 0$$

$$\Rightarrow x = \begin{bmatrix} 10.075 \\ 0 \\ 0.05 \end{bmatrix}$$

Problem 2 Following table shows height-weight data of 12 monkeys:

Height	Weight
18	29.5
28	39.2
36	54.5
25	36.0
17	25.0
31	43.8
21	30.4
35	56.1
24	36.0
22	29.9
18	26.9
32	48.2

Find out an orthonormal transform matrix to compress this data. Suppose you compress this two-dimensional data to one-dimensional. Show the data-compression procedure. What is the average error/distortion?

Solution:

The correlation of the data is y=1.5x. Then follow the procedure of transform coding to find angle θ and transform matrix

$$A = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix}$$

Then follow the procedure and can easily get the solution.