

Soft Decision Combined Source-Channel Coding

Design Chapter

Tamra Baxter

Ken Edwards

Tim Evans

February 4, 2005

Supervised by:

Dr. Fady Alajaji

Dr. Firouz Behnamfar

2 Design

2.1 Engineering Design

The goal of this project is to investigate the performance of various decoding schemes for transmissions over noisy channels. Performance can be measured in complexity, cost, and bandwidth. Visual evaluation can sometimes conflict with the previous quantitative measures, as human visual perception can often mask certain types of distortion. The following discusses the main areas of design within this project.

The simulation of noisy channels has already been done so the project will not focus on this simulation. However, there are aspects of the design as to how these noisy channels are simulated in our C models such as implementation, storage, precision, etc. The binary symmetric channel is great for initially verifying the correctness of code. The Gaussian channel is better used to simulate real world performance. The Rayleigh fading channel is an even better approximation to wireless channel behaviour and will be enforcement to the validity of our simulations.

This project will focus on how decoding is accomplished on noisy transmissions. Hard-decision decoding is a very simple decoding scheme. Received vectors/codewords generally include noise, so to determine what codeword was actually sent involves choosing the closest codeword from the codebook. Here we define closest codeword, y_i , as having the highest probability that y_i was sent given our received codeword. A decision region is the region where all received vectors/codewords lie that will be mapped to a given codeword from our codebook. Complexity is minimal with this decoding scheme, small errors will be corrected while large errors will be decoded as incorrect codewords. If a received codeword included enough noise to put it slightly in an incorrect region then decoding will increase the perceived noise and decrease the signal to distortion ratio (SDR). We want to see if a different decoding scheme will provide better results.

Soft-decision decoding, as opposed to hard-decision decoding, offers a different way of decoding that has the potential to improve SDR as well as perceived image quality. In soft-decision decoding, we will have intermediate regions that received codewords may be decoded into. Of course this method of decoding has its tradeoffs. While there is the opportunity that we decrease SDR com-

pared with hard-decision decoding for the case of uncorrected decoding, we can also increase SDR when decoding received codewords in an intermediate region that would otherwise be corrected with hard-decision decoding.

2.2 Mathematical and Software Models

2.2.1 Binary Symmetric Channel

The binary symmetric channel (BSC) is one in which each bit in the stream has an equal probability of being flipped. If the probability of one bit being flipped is ϵ then the probability of a message containing n bits being sent over a BSC with k bits being flipped is:

$$\binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k}$$

2.2.2 Additive White Gaussian Noise Channel

The additive white Gaussian noise (AWGN) channel is a simple and effective general channel model. In an AWGN channel the source vector \mathbf{x} may have an error introduced by the addition of an error vector \mathbf{n} . The only error that is introduced in an AWGN channel is additive. Multiplicative noise, which models such things as fading, is not modeled in this channel. The whiteness of the channel refers to the fact that noise is added at equal power across the frequency spectrum. The noise that is added to the source vector has zero mean and an average of σ^2 , and hence the noise is Gaussian.

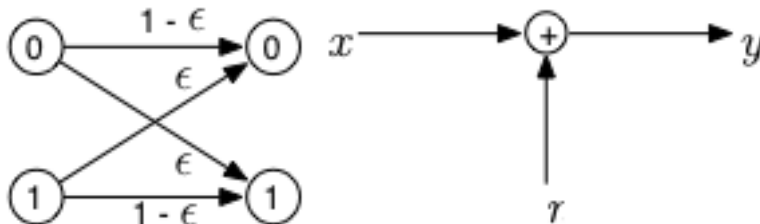


Figure 1: BSC and AWGN Channel Models

2.3 Vector Quantization

In vector quantization (VQ) we take a string of bits and assign them one codeword. There is an iterative algorithm called the Lloyd algorithm used to choose the optimal codewords for a set of vectors using two conditions: the nearest neighbour condition and the centroid condition.

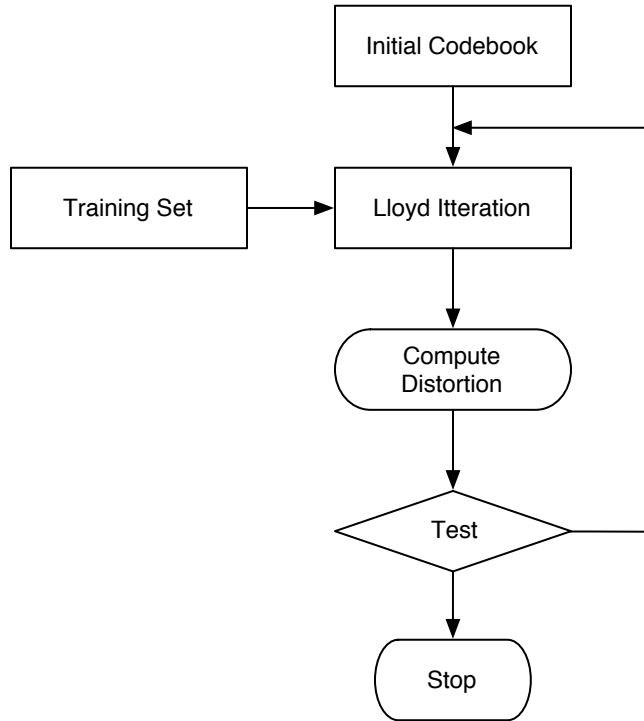


Figure 2: Lloyd Algorithm Flow Chart

2.3.1 Nearest Neighbour Condition

Each R_i consists of all vectors \mathbf{x} which have less distortion when reproduced with code vector \mathbf{c}_i than with any other code vector. The nearest neighbour condition is how we divide up the whole space of possible inputs into decision regions. In this case we use the mean square error (MSE) as our distortion measure, yielding the following equation:

$$R_i = \{\mathbf{x} : \|\mathbf{x} - \mathbf{c}_i\| \leq \|\mathbf{x} - \mathbf{c}_j\|, j = 1, \dots, N\} \forall i = 1, \dots, N$$

where R_i is the i^{th} decision region, \mathbf{x} is an input vector, \mathbf{c}_i is the i^{th} codevector, and N is the number of codevectors.

2.3.2 Centroid Condition

Once the decision regions, R_i have been determined by the nearest neighbour condition, the centroid condition decides what codevector should act as the representative for this region. In the case that we use MSE as our distortion measure, the centroid condition reduces to finding the average codevector in the region:

$$\mathbf{c}_i = \frac{\sum_{m:\mathbf{x}_m \in R_i} \mathbf{x}_m}{\sum_{m:\mathbf{x}_m \in R_i} 1}$$

The two steps of applying the nearest neighbour condition and the centroid condition are repeated until the difference between two consecutive iterations falls below a predetermined threshold value. After the threshold has been reached we call the codebook $\mathcal{C} = \{\mathbf{c}_i\}_{i=1}^N$

2.4 Channel Optimized Vector Quantization

Once the quantizer must be optimized for a particular channel the nearest neighbour and centroid conditions must be changed accordingly. The difference between the nearest neighbour and centroid conditions in a COVQ and a VQ is that the transition probabilities of the channel must be taken into account. These probabilities incorporate the fact that noise might take a transmitted signal from one codevector to another.

The nearest neighbour condition becomes:

$$R_i^* = \{x : \sum_{j=0}^{M-1} P_{J|I}(j|i) \|\mathbf{x} - \mathbf{c}_j\|^2 \leq \sum_{j=0}^{M-1} P_{J|I}(j|k) \|\mathbf{x} - \mathbf{c}_j\|^2 \forall k = 0, \dots, M-1\}$$

while the centroid condition changes to:

$$\mathbf{c}_i^* = \frac{\sum_{j=0}^{M-1} P_{J|I}(j|i) \sum_{m:\mathbf{x}_m \in R_j} \mathbf{x}_m}{\sum_{j=0}^{M-1} P_{J|I}(j|i) \sum_{m:\mathbf{x}_m \in R_j} 1} \text{ for } j = 0, \dots, M-1$$

We now get an optimal codebook to get $\mathcal{C}^* = \{\mathbf{c}_i^*\}, i = 0, \dots, M-1$.

2.5 Software Design and Analysis

The old saying in software engineering goes: “Fast, cheap, or good. Pick any two.” In this adage *fast* refers to time to market, *cheap* to the cost of the project, and *good* to how well the product is implemented. Our software must be developed quickly. We have a hard deadline of the end of March, 2005. By that time our software must be stable and be producing correct results. Since we are but undergraduate students, our cost comes in terms of an opportunity cost. We have to make tradeoffs both in terms of different parts of this project and in terms of time division between this project and our other coursework. It is in the *good* of the project that we had to make some sacrifices. This is not to say that the code produced does not work, it does. The sacrifices come in the fact that the program has not been developed in a manner that follows what are considered good software design patterns. The tradeoffs with price and development time are not the only reason for this sacrifice. This project is one in which the source code goes through many incremental changes. Once the team understands a new mathematical concept, we go back to the code and make the necessary changes.

The code base started with a program provided by Dr. Behnamfar which simulated vector quantization of a Gaussian source. The team’s first step was to make the connection between our mathematical understanding of Gaussian sources, the nearest neighbour condition, and the centroid condition, and the source code provided. Dr. Behnamfar stressed that a strong understanding of the original source code and the underlying mathematics would be critical to the success of the project. Once the connection was made, the first step was to simulate communication across a channel. The first channel implemented was the binary symmetric channel, the simplest channel to understand. Once this step was complete, the team modified the code to adapt the vector quantization to the characteristics of the channel, in effect, the team implemented channel optimized vector quantization (COVQ).

The channel is represented by an n by n^1 transition probability matrix. We pre-computed the matrix so that when channel errors are simulated, only a table lookup need be preformed. The next step in the project is to change the channel to one that models additive white Gaussian

¹Where n is the number of vectors in our code book

noise (AWGN). Once a more realistic channel is in place, the soft decision decoding portion of the project will be introduced. This will involve a more significant amount of programming than the previous portions of the project. If time remains after soft decision decoding is implemented two other possible avenues of exploration include implementing a Rayleigh fading channel or to move on to a higher dimension quantization method, for example moving from BPSK to QAM.

The project does not end with the simulation of different channels and decoders. Once complete, a numerous amount of simulations must be run and analysis preformed. One of the team members has some experience in automated testing and report generation, so these skills will be exploited in a hope that extensive automated testing will lead to a greater amount of data, which will in turn lead to a more precise analysis.

The analysis will include comparing VQ to COVQ and soft decision COVQ to hard decision COVQ. The analysis will also include details on how a ‘good’ codebook is chosen. Good can be described both quantitatively, by examining various signal to noise ratios (SDR, SNR), and quantitatively, by examining the resulting transmissions. Standard images such as Lena and the baboon [Figure 3] will be used.



Figure 3: Standard sample images in communications theory

2.6 Division of Labour

After setting up several meetings with our advisor Dr. Behnamfar, the team decided that it would be in our best interests to make the meeting at least a weekly occurrence. The team members are fortunate in that they have very similar academic schedules, which makes group work very easy to schedule. Eighty to ninety percent of the work done so far has been done as a group. Any work that has been done individually has been reviewed together as a group. As mentioned in a previous section, Dr. Behnamfar has been extremely helpful by providing guidance both with the mathematical concepts and the computer implementation.