

Soft Decision Combined Source-Channel Coding

Design Chapter

Tamra Baxter

Ken Edwards

Tim Evans

February 2, 2005

Supervised by:

Dr. Fady Alajaji

Dr. Firouz Behnamfar

2 Design

2.1 Mathematical and Software Models

2.1.1 Binary Symmetric Channel

The binary symmetric channel is one in which each bit in the stream has an equal probability to flip. If the probability of one bit flipping 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 Vector Quantization

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

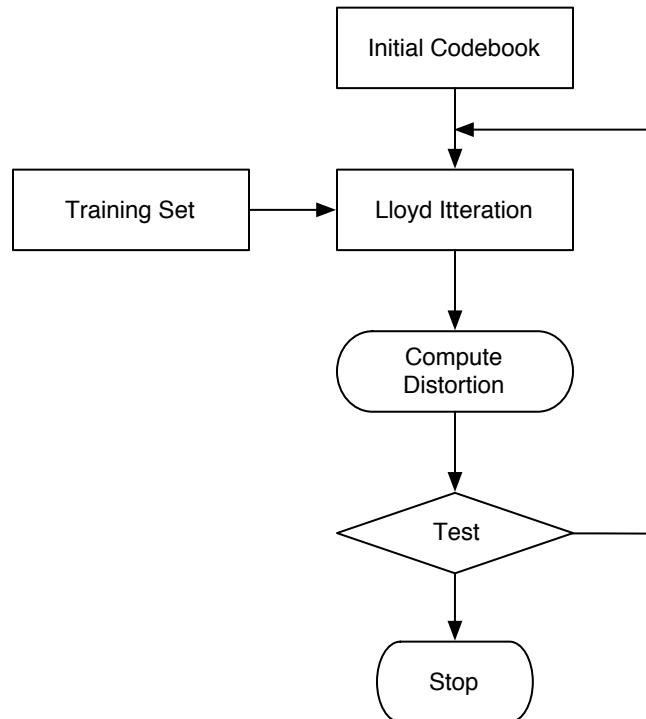


Figure 1: Lloyd Algorithm Flow Chart

2.2.1 Nearest Neighbour Condition

$$R_i = \{x : \|x - c_i\| \leq \|x - c_j\|, j = 1, \dots, N\} \forall i = 1, \dots, N$$

Where we have that R_i is the i^{th} decision region, x is an input vector, c_i is the i^{th} codevector, and N is the number of codevectors.

2.2.2 Centroid Condition

$$c_i = \frac{\sum_{m: x_m \in R_i} x_m}{\sum_{m: x_m \in R_i} 1}$$

After the threshold has been reached we call the codebook $\mathcal{C} = c_{i=1}^N$

2.3 Channel Optimized Vector Quantization

2.3.1 Nearest Neighbour Condition

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

2.3.2 Centroid Condition

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

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

2.4 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 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 in fact works quite well. The

sacrifices come in the fact that the program has not been coded in a manner that follows standard 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. One 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 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. To begin the simplest channel, the binary symmetric channel, was implemented. 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 that is pre-computed 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 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 implementation of the 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

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

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 qualitatively, by examining the resulting transmissions. Standard images such as Lena and the baboon will be used, see Figure 2.



Figure 2: Standard sample images in communications theory

2.5 Division of Labour

After setting up several separate 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.

Here are some guidelines for the Design chapter.

- * Describe the engineering design component of the project. Include detailed descriptions of the relevant technical issues, diagrams, etc.

- *[DONE] Describe your design methodology: how you went about designing your project.

What approach did you follow ? How is the adopted approach mathematically justified ? (If it is not, explain why and the difficulties involved). What were the alternative methods ? How did you arrive at the final design ? Give justifications.

* [DONE] Describe the division of labour and scheduling among the group members, and the technical cooperation with the TA's and the supervising faculty.

* [Our original proposal was crap] Discuss your compliance to your original proposal.

References

- [1] T.M. Cover and J. Thomas. Elements of Information Theory. Toronto, John Wiley & Sons Inc., 1991.
- [2] A. Gersho and R. M. Gray . Vector Quantization and Signal Compression. Kluwer, 1992.
- [3] Julian Cheng. Channel Optimized Quantization of Images over Binary Channels with Memory. M.Sc. Thesis (Department of Mathematics & Statistics), September 1997.
- [4] N. Phamdo and F. Alajaji, “Soft-Decision Demodulation Design for COVQ over White, Colored, and ISI Gaussian Channels,” IEEE Transactions on Communications, Vol. 48, No. 9, pp. 1499-1506, September 2000.
- [5] F. Alajaji and N. Phamdo, “Soft-Decision COVQ for Rayleigh Fading Channels,” IEEE Communications Letters, Vol. 2, No. 6, pp. 162-164, June 1998.