

EE368B – Image and Video Compression

Homework Set #2

due Friday, October 20, 2000, 9 a.m.

Introduction

The Lloyd-Max quantizer is a scalar quantizer which can be seen as a special case of a vector quantizer (VQ) designed with the LBG algorithm. For a given distortion criterion, encoding rule, and training set, given an initial codebook C_1 , the LBG algorithm, a.k.a. the Generalized Lloyd Algorithm, iteratively optimizes the decoder while fixing the encoder and then the encoder while fixing the decoder, until the average distortion converges. The iterative procedure guarantees nonincreasing distortion and thus convergence. The resulting codebook is then indexed with a fixed-length code. In practice and in this homework, we will use mean squared error (MSE) as the distortion criterion, and a minimum distortion or nearest neighbor mapping as the encoding rule, which is optimal for encoding with a given decoder.

Similarly, the entropy-constrained Lloyd-Max quantizer is a special case of entropy-constrained VQ (ECVQ), designed with a modified LBG algorithm. Rather than indexing the representative vectors with a fixed-length code and minimizing average distortion D , this quantizer uses a variable-length code (VLC) to index the representative vectors and minimizes a cost function $J = D + \lambda R$, where R is the average rate of codeword indices over the distribution of the training set.

Before we describe the LBG algorithm - with or without the entropy constraint - in detail, let's first define some notations:

codebook

$C = \{y_i\}$, where each y_i is a representative vector, also referred to as a reproduction codeword, and i is the index codeword

encoder (lossy)

maps from an input vector to an index codeword, denoted as $A(x) = i$

entropy coder (lossless) maps from an index codeword to a variable-length channel codeword, denoted as $G(i)$

entropy decoder

maps from a variable-length channel codeword to an index codeword, i.e. $G^{-1}(G(i)) = i$

decoder

maps from an index codeword to a reproduction codeword, denoted as $B(i) = y$

distortion between vectors x and y

$d(x, y) = \|x - y\|^2$, i.e. squared error or squared Euclidean distance

cost of mapping vector x to a reproduction codeword y , which is associated with a channel codeword with length l , with the Lagrangian multiplier λ

$j(x, y, l) = d(x, y) + \lambda l$

training set

$T = \{x_i\}$

partition of T using C

$P = \{R_i\}$, where

- in the unconstrained case $R_i = \{x : d(x, y_i) \leq d(x, y_k); \forall k \neq i\}$.
- in the entropy-constrained case $R_i = \{x : j(x, y_i) \leq j(x, y_k); \forall k \neq i\}$.

training set quantized with codebook C using partition P

$T' = Q(T, C, P) = \{x'_i\}$

average distortion between T and T'

$D(T, T') = E \{(d(x, y))\}$

average rate of codebook $C = \{y_i\}$, associated with channel codewords of length l_i , over a distribution $p(i)$

$R = E \{l\} = \sum_i p(i) l_i$

average cost of C over T , with partition P and associated channel codewords of length $\{l_i\}$

$J = D(T, T') + \lambda R$, where $T' = Q(T, C, P)$, and $p(i)$ is the distribution in T'

Given a training set T , the LBG algorithm works as follows:

1. Initialization

Begin with an initial codebook C_1 . Set $m = 1$ and $D_m = D(T, T')$, where T' results from quantizing T using C_1 .

2. Optimize the encoder

Given the codebook, $C_m = \{y_i\}$, find the optimal partition into quantization cells, i.e. use the nearest neighbor condition to form the nearest neighbor cells: $P = \{R_i\}$, where $R_i = \{x : d(x, y_i) \leq d(x, y_k); \forall k \neq i\}$. Note: in the scalar case, quantization cells are simply parts of the real line.

3. Optimize the decoder

Find the centroid in each quantization cell, $y_i = E\{x|x \in R_i\}$, which form C_{m+1} , the optimal reproduction alphabet (codebook) for the cells.

4. Update the average distortion

Let $T' = Q(T, C_{m+1}, P)$. Compute $D_{m+1} = D(T, T')$. If it has changed by a small enough amount, i.e.

$$\frac{D_m - D_{m+1}}{D_m} < \epsilon,$$

stop. Otherwise set $m + 1 \rightarrow m$ and go to step 2.

Given a training set T and a value for λ , the modified LBG algorithm for ECVQ works as follows:

1. Initialization

Begin with an initial codebook C_1 associated with a set of initial channel codeword lengths $\{l_i\}$. Set $m = 1$ and $J_m = D(T, T') + \lambda R$, where T' results from quantizing T using C_1 , and R is calculated using distribution of T' and channel codeword lengths $\{l_i\}$.

2. Optimize the encoder

Given the codebook, $C_m = \{y_i\}$, find the optimal partition into quantization cells: $P = \{R_i\}$, where $R_i = \{x : j(x, y_i) \leq j(x, y_k); \forall k \neq i\}$.

3. Optimize the entropy coder

Find distribution $p(i)$ of index codewords from partition P . Update the set of channel codeword lengths using $l_i = -\log_2(p(i))$.

4. Find the centroid in each quantization cell, $y_i = E\{x|x \in R_i\}$, which form C_{m+1} , the optimal reproduction alphabet (codebook) for the cells.

5. Let $T' = Q(T, C_{m+1}, P)$. Compute $J_{m+1} = D(T, T') + \lambda R$, where R is calculated using distribution of T' and the updated channel codeword lengths $\{l_i\}$. If the average cost has changed by a small enough amount, i.e.

$$\frac{J_m - J_{m+1}}{J_m} < \epsilon,$$

stop. Otherwise set $m + 1 \rightarrow m$ and go to step 2.

During the design process, if an empty quantization cell is encountered, we can either remove the corresponding codeword and thus end up with a smaller codebook, or replace this codeword by an alternative codeword. One way to choose this alternative codeword is to identify the codeword associated with the most populated quantization cell, add a small offset to it and use that as a new codeword.

If you are interested in more details in Lloyd-Max algorithm and LBG algorithm, please refer to "Vector Quantization and Signal Compression" by Allen Gersho and Robert M. Gray. For a detailed analysis of Entropy-Constrained Vector Quantization, please refer to "Entropy-Constrained Vector Quantization" by Philip A. Chou, Tom Lookabaugh, and Robert M. Gray, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 1, Jan. 1989.

1 Lloyd-Max Quantizer

Design a 3-bit and 4-bit Lloyd-Max quantizer for the image set *boats*, *harbour*, and *peppers*. To reduce the computational complexity, subsample the images and take every fourth row and column. Note that there are different ways of choosing the training set. The way the training set is chosen will have an impact on the performance of the resulting quantizer.

Choosing the initial representative vectors is also important in the quantizer design. We recommend that you start with a uniform quantizer. During the iterative optimization process, one or more codeword(s) might end up having no samples mapped to it, forming empty quantization cell(s). To waste no code word for the fixed-length code, it is recommended to replace the codeword in an empty cell with a new codeword, generated by splitting the most populated cell.

You should select $\epsilon = 0.001$ to obtain good quantizers. Hence, it will take several iterations (up to 15) to find a local minimum. We recommend a efficient implementation of the full search with the Matlab function 'min'. To minimize your time spent on this problem, you also might want to write a few general functions and then each quantizer will only take a few lines of code.

Plot the probabilities for the 3-bit and 4-bit Lloyd-Max quantizer output and plot the two obtained rate-distortion pairs in a rate-PSNR plane. The *Peak Signal to Noise Ratio* (PSNR) is a widely used logarithmic distortion measure for images. It is defined for 8-bit images as follows:

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{D} \right)$$

Plot the rate of the fixed length code as well as the smallest average code word length obtained by a theoretically optimal entropy code. Determine the slope of the rate-distortion curves in dB/bit. Report also the number of steps the algorithm takes to convergence.

2 Entropy-Constrained Scalar Quantization

Extend your Lloyd-Max quantizer design algorithm to enable entropy-constrained scalar quantization. Take the same training set as used in Problem 1. For initialization, start with a 4-bit uniform quantizer.

You only need to produce a quantized image, not a coded image. Therefore, you can assume a theoretically optimal entropy code without having to actually implement it.

During the iterative optimization process, one or more codeword(s) might end up having no samples mapped to it, forming empty quantization cell(s). For entropy-constrained algorithms, removing a codeword does not cause any waste in bits, whereas replacing a codeword by splitting another cell might actually increase entropy. So we recommend to simply remove the codeword in an empty cell.

You should also select $\epsilon = 0.001$ to obtain good entropy-constrained quantizers.

Design 7 quantizers for $\lambda = 0, 5, 10, 30, 62, 70,$ and 90 . Plot the probabilities for the entropy-constrained scalar quantizer output for $\lambda = 0$ and $\lambda = 70$ and add the obtained rate-distortion pairs to your rate-PSNR plane. Determine the slope of the rate-distortion curve in dB/bit. Report also the number of steps the algorithm takes to convergence.