ECE 499/599 Data Compression/Information Theory
Spring 06

Dr. Thinh Nguyen

Homework 5
Due 05/30/06 at the beginning of the class

Problem 1: We have the following pixel values with the corresponding frequency of occurrence. (6pts)

Pixel values	1	2	3	4	5	6	7	8	9	10
Frequency	100	20	300	120	1000	600	900	100	400	123

a) Suppose you are using a codebook of size 3, and the initial codewords are C(0) = 1, C(1) = 2, and C(2) = 3. Show steps by steps the final codeword after running the Loyd-Max Algorithm. What is the distortion value?

Note that you may have different answer depending on how you decide the tie.

Let denote the codeword C=[C0,C1,C2] and the corresponding distortion Di=[D0,D1,D2]
Step 1: C=[1,2,3]
X(0)=[1]; X(1)=[2]; X(2)=[3,4,5,6,7, 8, 9,10]
Di'=[0, 0, 46847]
D'= 46847
C'=[1, 2, 6]

Step 2: C=[1, 2, 6]
X(0)=[1]; X(1)=[2, 3, 4]; X(2)=[5, 6, 7 8, 9,10]
Di'=[0, 780, 7868]
D'=8648
C'=[1, 3, 7]

Step 3: C=[1, 3, 7]
X(0)=[1, 2]; X(1)=[3, 4, 5]; X(2)=[6,7,8,9,10]
Di'=[20, 4120, 3407]
D'=7547
C'=[1, 4, 7]

Step 4: C'=[1, 4, 7]
X(0)=[1, 2]; X(1)=[3,4,5]; X(2)=[6,7, 8, 9, 10]
Di'=[20, 1300, 3407];
D=4727
C'=[1, 4, 7]

Step 5: C=[1 4 7]
X(0)=[1, 2]; X(1)=[3,4,5]; X(2)=[6,7, 8, 9, 10]

Di’=[20, 1300, 3407];
D’=4727
C’=[1, 4, 7]
|(D-D’)/D|=0

b)
Step 1**:** **C** =[8, 9, 10]
X(0)=[1, 2,3,4,5,6,7,8]; X(1)=[9]; X(2)=[10]
Di’= [27340, 0, 0]
D’= 27340
C’=[5, 9, 10]

Step 2: C=[5, 9, 10]
X(0)=[1, 2, 3, 4, 5, 6]; X(1)=[7,8,9]; X(2)=[10]
Di’= [3700, 3700, 0]
D’=7400
C’=[5, 8, 10]

Step 3: C=[5, 8, 10]
X(0)=[1, 2, 3, 4,5,6], X(1)=[7,8,9], X(2)=[10]
Di’=[3700, 3700, 0]
D’=7400
C’=[5 , 8, 10]

Step 4: C=[5, 8, 10]
X(0)=[1, 2, 3, 4,5,6]; X(1)=[7,8,9]; X(2)=[10]
Di’=[3700, 3700, 0]
D’=7400
C’=[5 , 8, 10]
|(D-D’)/D|=0

The distortion is D=7400
The final codeword C=[5, 8, 10]

c) Do you think the final codewords will always be the same?
No, see the results (a) and (b) above.

Problem 2: Show that distortion value in the Loyd-Max quantizer monotically decreases with the number of iterations. (4pts)

We know that a centroid codeword minimizes the distortion for all the data point x_i belongs to a set A.
In other words, let $x_i \in A$, and $c^* = \sum_i p_i x_i$, then $D^* = \sum_i p_i (x_i - c^*)^2$ is minimized over all possible value of c. The distortion in the first iteration of Lloyd-max algorithm is

$D = \sum_i p_i (x_i - c)^2$ where c is not yet a centroid of set A. Now in the next step,

We compute $c^* = \sum_i p_i x_i$ using all $x_i \in A$. Therefore, $D^* = \sum_i p_i (x_i - c^*)^2$. Clearly,

$D^* \le D$. However, D* is not the distortion in the second iteration! Remember that we have to find the "boundaries" for the codeword. In other words, we have to classify whether a data belongs to a codeword c1 or c2. This means that we have find the closest codeword to x_i, before we can compute the distortion. Now suppose x_i is now closer to codeword c' than c*, then by moving x_i from c* to c', the change in total distortion will be:
$p_i(x_i - c')^2 - p_i(x_i - c^*)^2 \le 0$ since $|x_i - c| \le |x_i - c^*|$. Thus distortion in the second distortion $D^{**} \le D^* \le D$. Q.E.D

Problem 3: In this problem, we will use MatLab to perform vector quantization on image. We will use the image Lena512.pgm (10pts).

The Matlab code is given below
Number of iteration to achieve error $<10^{-4}$ is 81
The reconstructed image is:

```
%The Generalized Lloyd Algorithm to encode Lena image
data = double(imread('lena512.pgm'));
e = 1000;
X = zeros(512,512);
D = zeros(16,1);
Dprime = zeros(16,1);
% generate random codewords
codebook = floor(rand(16,4) * 256);
%codebook = ran_array;
[X,D]=closest(data,codebook);
while error > 10^(-4)
    codebook=Ccalculation(data,X,codebook);
    % recompute X and D
    [X,D1]=closest(data,codebook);
    error = abs((sum(D)-sum(D1)))/sum(D);
    D = D1;
    error
    sum(D)
end

%begin to reconstruct image
i = 1;
j = 1;
while i <= 512
    while j <= 512
        data(i,j) = (codebook(X(i,j),1));
        data(i+1,j) = (codebook(X(i,j),2));
        data(i,j+1) = (codebook(X(i,j),3));
        data(i+1,j+1) = (codebook(X(i,j),4));
        j = j + 2;
    end
    i = i + 2;
    j = 1;
end
data = uint8(data);
imshow(data);
```

```
function [X,D]=closest(data,codeword)
% find closest blocks to the codewords
% return the closest blocks and the corresponding distortion
l=512;
i=1;
j=1;
D = zeros(16,1);
while i <= l
    while j <= l
        min = double(1);
        min_distance = double(2^32);
        for k = 1:16
            distance = double(0);
            distance = distance + (data(i,j)-codeword(k,1))^2+(data(i+1,j)-
codeword(k,2))^2+(data(i,j+1)-codeword(k,3))^2 +(data(i+1,j+1)-codeword(k,4))^2;
            if (distance < min_distance)
                min = double(k);
                min_distance = distance;
            end
        end
        X(i,j) = min;
        D(min) = D(min) + min_distance;
        j = j + 2;
    end
    i = i + 2;
    j = 1;
end
```

```
function c=Ccalculation(data,X,codebook)
% compute new codebook
    i = 1;
    j = 1;
    cbn = ones(16,4);
    cb1 = zeros(16,4);
    Dprime = zeros(16,1);
    while i <= 512
        while j <= 512
            cb1(X(i,j),1) = cb1(X(i,j),1) + data(i,j); cbn(X(i,j),1) = cbn(X(i,j),1) +
1;
            cb1(X(i,j),2) = cb1(X(i,j),2) + data(i+1,j); cbn(X(i,j),2) = cbn(X(i,j),2)
+ 1;
            cb1(X(i,j),3) = cb1(X(i,j),3) + data(i,j+1);  cbn(X(i,j),3) =
cbn(X(i,j),3) + 1;
            cb1(X(i,j),4) = cb1(X(i,j),4) + data(i+1,j+1); cbn(X(i,j),4) =
cbn(X(i,j),4) + 1;
            j = j + 2;
        end
        i = i + 2;
        j = 1;
    end

    for k=1:16
        codebook(k,1)=cb1(k,1)/cbn(k,1);
        codebook(k,2)=cb1(k,2)/cbn(k,2);
        codebook(k,3)=cb1(k,3)/cbn(k,3);
        codebook(k,4)=cb1(k,4)/cbn(k,4);
    end

c=codebook;
```