Multimedia Signal processing

Prof. Marco Tagliasacchi

March, 1 2012

1. **Question (8 pts)**:
Consider two sources, X and Y, that emit symbols according to the following joint p.m.f.

$$\Pr\{X = i, Y = j\} = w_{i,j}, \tag{1}$$

where the probability values $w_{i,j}$ are defined in the matrix below

$$\mathbf{W} = \frac{1}{8}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

(a) Compute the entropy of both sources, i.e., $H(X)$ and $H(Y)$.

(b) Compute the joint entropy $H(X,Y)$

(c) Quantize each source with 1 bit. Design the (scalar) Lloyd-Max quantizer. Initialize the intervals as $I_1 = (-\infty, a_2]$, $I_2 = (a_2, +\infty]$, $a_2 = 2$.

(d) Design the optimal vector quantizer to jointly quantize X and Y, allocating 1 bit per dimension. Initialize the centroids with the reconstruction levels found with Lloyd-Max. Execute only one iteration.

(e) In both cases, compute the quantization MSE and a lower bound on the minimum number of bits necessary to encode the output symbols of the quantizer.

Solution

(a) The joint p.m.f. describes a discrete source, since it is nonzero only at $(1,1), (1,3), (2,3), (3,2), (3,3), (4,4)$, where it is equal to a delta function having area $w_{i,j}$.

(b) The marginals are given by

$$\Pr\{X = 1\} = \frac{1}{4} \tag{3}$$

$$\Pr\{X = 2\} = \frac{1}{4} \tag{4}$$

$$\Pr\{X = 3\} = \frac{3}{8} \tag{5}$$

$$\Pr\{X = 4\} = \frac{1}{8} \tag{6}$$

$$\Pr\{Y=1\} = \frac{1}{8} \tag{7}$$
$$\Pr\{Y=2\} = \frac{1}{4} \tag{8}$$
$$\Pr\{Y=3\} = \frac{3}{2} \tag{9}$$
$$\Pr\{Y=4\} = \frac{1}{8} \tag{10}$$

The entropy values are equal to

$$H(X) = -\frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{3}{8}\log_2\frac{3}{8} - \frac{1}{8}\log_2\frac{1}{8} = 1.9056 \tag{11}$$
$$H(Y) = -\frac{1}{8}\log_2\frac{1}{8} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{8}\log_2\frac{1}{8} = 1.75 \tag{12}$$

(c) The joint entropy is

$$H(X,Y) = -4\frac{1}{8}\log_2\frac{1}{8} - 2\frac{1}{4}\log_2\frac{1}{4} = 2.50 \tag{13}$$

(d) Quantizer for source X. Initialize $I_1 = (-\infty, a_2]$, $I_2 = (a_2, +\infty]$, $a_2 = 2$.

1. Given the thresholds, update the reconstruction level

$$r_1 = E[x|x \in I_1] = \frac{1}{2}1 + \frac{1}{2}2 = \frac{3}{2} \tag{14}$$
$$r_2 = E[x|x \in I_2] = \frac{3}{4}3 + \frac{1}{4}4 = \frac{13}{4} \tag{15}$$
$$\tag{16}$$

2. Given the reconstruction level, update the thresholds. That is, $a_2 = (r_1+r_2)/2 = \frac{19}{8}$
3. Given the thresholds, update the reconstruction level

$$r_1 = E[x|x \in I_1] = \frac{1}{2}1 + \frac{1}{2}2 = \frac{3}{2} \tag{17}$$
$$r_2 = E[x|x \in I_2] = \frac{3}{4}3 + \frac{1}{4}4 = \frac{13}{4} \tag{18}$$
$$\tag{19}$$

4. Convergence.

Quantizer for source Y. Initialize $I_1 = (-\infty, a_2]$, $I_2 = (a_2, +\infty]$, $a_2 = 2$.

1. Given the thresholds, update the reconstruction level

$$r_1 = E[y|y \in I_1] = \frac{\frac{1}{8}}{\frac{1}{8}+\frac{1}{4}}1 + \frac{\frac{1}{4}}{\frac{1}{8}+\frac{1}{4}}2 = \frac{5}{3} \tag{20}$$
$$r_2 = E[y|y \in I_2] = \frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{8}}3 + \frac{\frac{1}{8}}{\frac{1}{2}+\frac{1}{8}}4 = \frac{16}{5} \tag{21}$$
$$\tag{22}$$

2. Given the reconstruction level, update the thresholds. That is, $a_2 = (r_1+r_2)/2 = 2.43$

3. Given the thresholds, update the reconstruction level

$$r_1 = E[y|x \in I_1] = \frac{\frac{1}{8}}{\frac{1}{8}+\frac{1}{4}}1 + \frac{\frac{1}{4}}{\frac{1}{8}+\frac{1}{4}}2 = \frac{5}{3} \tag{23}$$

$$\tag{24}$$

$$r_2 = E[y|x \in I_2] = \frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{8}}3 + \frac{\frac{1}{8}}{\frac{1}{2}+\frac{1}{8}}4 = \frac{16}{5} \tag{25}$$

$$\tag{26}$$

$$\tag{27}$$

4. Convergence.

(e) The initial centroids are given by the result of Lloyd-Max. That is,

$$\mathbf{x}_1 = [\frac{3}{2}, \frac{5}{3}]^T \tag{28}$$

$$\mathbf{x}_2 = [\frac{3}{2}, \frac{16}{5}]^T \tag{29}$$

$$\mathbf{x}_3 = [\frac{13}{4}, \frac{5}{3}]^T \tag{30}$$

$$\mathbf{x}_4 = [\frac{13}{4}, \frac{16}{5}]^T \tag{31}$$

$$\tag{32}$$

The four quantization regions I_i, $i = 1, \dots, 4$, are defined by the Voronoi diagram. In this simple case, the Voronoi diagram is defined by the lines $x = 2.43$ and $y = 2.375$. Note: you don't really need to compute the Voronoi diagram. The only thing you need to determine is the closest centroid to each input vector.

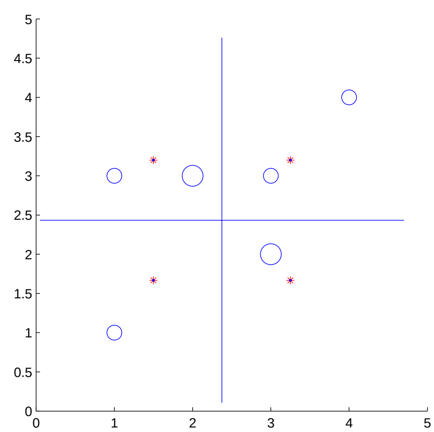

Figure 1: Voronoi diagram at the first iteration.

Therefore, we can recompute the codebook (i.e. the reconstruction vector) as follows

$$\mathbf{y}_1 = [1,1]^T \tag{33}$$

$$\mathbf{y}_2 = \frac{\frac{1}{8}[1,3]^T + \frac{2}{8}[2,3]^T}{\frac{1}{8}+\frac{2}{8}} = [5/3,3]^T \tag{34}$$

$$\mathbf{y}_3 = [3,2]^T \tag{35}$$

$$\mathbf{y}_4 = \frac{\frac{1}{8}[3,3]^T + \frac{1}{8}[4,4]^T}{\frac{1}{8}+\frac{1}{8}} = [7/2,7/2]^T \tag{36}$$

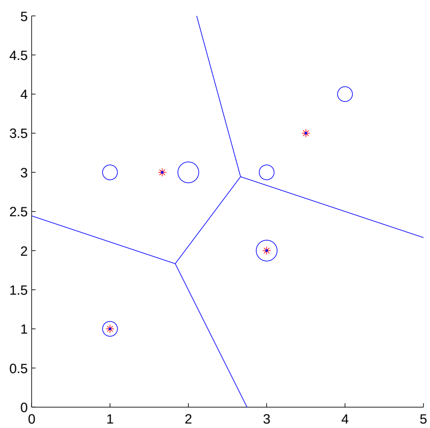

Figure 2: Voronoi diagram after the first iteration.

(f) The MSE in the case of the Lloyd-Max quantizer is

$$MSE_{LM} = \frac{1}{2}[\frac{1}{8}\|[1,1]^T - \mathbf{x}_1\|^2 + \tag{37}$$

$$\frac{1}{8}\|[1,3]^T - \mathbf{x}_2\|^2 + \tag{38}$$

$$\frac{2}{8}\|[2,3]^T - \mathbf{x}_2\|^2 + \tag{39}$$

$$\frac{2}{8}\|[3,2]^T - \mathbf{x}_3\|^2 + \tag{40}$$

$$\frac{1}{8}\|[3,3]^T - \mathbf{x}_4\|^2 + \tag{41}$$

$$\frac{1}{8}\|[4,4]^T - \mathbf{x}_4\|^2] \tag{42}$$

The MSE in the case of VQ is obtained as before, replacing $\mathbf{x}_i$ with $\mathbf{y}_i$.
A lower bound on the number of bits necessary to encode (disjointly) the output symbols of the two LM scalar quantizer is given by

$$R_{LM} \geq H(\tilde{X}) + H(\tilde{Y}) \tag{43}$$

where $\tilde{X}$ and $\tilde{Y}$ denote the sources that describe the output of the two scalar quantizers.
To compute the entropy of the output symbols, let

$$Pr\{x \in I_1\} = \frac{1}{8} + \frac{1}{8} + \frac{2}{8} \tag{44}$$

$$Pr\{x \in I_2\} = \frac{1}{8} + \frac{1}{8} + \frac{2}{8} \tag{45}$$

The symbols are equally likely, hence $H(\tilde{X}) = 1$.

$$Pr\{y \in I_1\} = \frac{1}{8} + \frac{2}{8} = \frac{3}{8} \tag{46}$$

$$Pr\{x \in I_2\} = \frac{1}{8} + \frac{1}{8} + \frac{2}{8} + \frac{1}{8} = \frac{5}{8} \tag{47}$$

Hence,

$$H(\tilde{Y}) = -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} = 0.9544 \tag{48}$$

$$R_{LM} \geq 1.9544 \tag{49}$$

In the case of VQ, we need to compute the probability of observing each output symbol $\mathbf{y}_i$. That is,

$$Pr\{\mathbf{y}_1\} = 1/8 \tag{50}$$

$$Pr\{\mathbf{y}_2\} = 3/8 \tag{51}$$

$$Pr\{\mathbf{y}_1\} = 2/8 \tag{52}$$

$$Pr\{\mathbf{y}_1\} = 2/8 \tag{53}$$

$$R_{VQ} \geq -\frac{1}{8}\log_2\frac{1}{8} - \frac{3}{8}\log_2\frac{3}{8} - \frac{2}{8}\log_2\frac{2}{8} - \frac{2}{8}\log_2\frac{2}{8} = 1.9056 \tag{54}$$

2. **Question (5 pts)**
   (a) Illustrate the block diagram of a perceptual audio codec (both encoder and decoder). Briefly describe the input and output of each block.
   (b) Describe why adaptive window switching is beneficial in terms of coding efficiency.

   Solution

   See slide set `MMSP_2.2_Audio coding`.

3. **Question (5 pts)**
   (a) Illustrate the block diagram of a video coding system.
   (b) Discuss the computational complexity of motion estimation.

   Solution

   See slide set `MMSP_2.1_Video_Coding`.

4. **Questions (5 pts - each answer can be either TRUE or FALSE)**

   (a) Let X denote a continuous source, and $D(R)$ the corresponding rate-distortion function, and σ_x^2 its variance.

   - $\boxed{\text{T}}\boxed{\text{F}}$ $H(X) \leq R(D)$
     FALSE. The entropy is undefined for continuous valued sources
   - $\boxed{\text{T}}\boxed{\text{F}}$ If $D(R)$ is achievable, then $D(R - \Delta R)$ is also achievable, when $\Delta R \geq 0$
     FALSE. In general, decreasing the rate at a given distortion might fall below the rate distortion curve.
   - $\boxed{\text{T}}\boxed{\text{F}}$ $R(D) \leq \frac{1}{2} \log_2 \frac{\sigma_x^2}{D}$
     TRUE. The rate-distortion function is upper bounded by the that of a Gaussian source having the same variance σ_x^2

   (b) Consider a speech coding system

   - $\boxed{\text{T}}\boxed{\text{F}}$ Encoding good quality speech requires at least 96 kbps
     FALSE. State-of-the-art encoders are able to encode good quality speech at 10-20kbps.
   - $\boxed{\text{T}}\boxed{\text{F}}$ In the case of voiced speech, the output of the short-term predictor is periodic.
     TRUE. The short term predictor takes into account the formant structure of the input speech, by whitening the envelope. The pitch (harmonic) structure is preserved
   - $\boxed{\text{T}}\boxed{\text{F}}$ The perceptual filter attenuates the error components at frequencies corresponding to the formants.
     TRUE. This is done to take into account frequency masking.

   (c) Let X denote a discrete source with memory.

   - $\boxed{\text{T}}\boxed{\text{F}}$ Shannon's coding is such that its average rate $R_S < H(X) + 1$
     TRUE.
   - $\boxed{\text{T}}\boxed{\text{F}}$ An instantaneous code is always uniquely decodable.
     TRUE.
   - $\boxed{\text{T}}\boxed{\text{F}}$ The shortest codeword in a Huffman code has always length equal to one.
     FALSE.

5. **Question (Matlab - 10pts)**

   Local image features are routinely used in several computer vision applications, including augmented reality (e.g Google Goggles). Their success is due to the ability of summarizing the input image in a way that is invariant to several transformations (scale, rotation, luminance, viewpoint etc.).

   (a) The file `feature_set.mat` contains a set of 64-dimensional SURF local features, extracted from a test image. You are asked to study the performance of DPCM coding on this feature set.

   - Load the file `feature_set`. Implement a closed-loop DPCM coding scheme on the feature set, i.e. quantizing the residuals between adjacent features with R bits per element ($1 \leq R \leq 8$). Plot the resulting rate distortion curve (R [bits per element] vs SNR [dB]).
     Hint: Consider each dimension d of the feature set as a different source. In other words, you have 64 separate DPCM encoders, one per dimension.

   (b) Recently, W. Gao et al. have proposed to sort the feature set in such a way that similar features appear one next the other in the set. This operation clearly improves the DPCM performance, since it reduces the variance of the residuals with respect to the case of the unordered set.

   - Load the file `feature_set_sorted.mat`, that contains the same set of features, this time sorted such that the distance between two consecutive features is minimized. Plot the ordered and the unordered features sets. (Hint: use the `imagesc` function). Compare the performance of the closed-loop DPCM scheme on the ordered set with respect to the unordered case, showing the new rate distortion curve.

Solution

```
clear all
close all
clc;

load feature_set;
N = size(features,2);


R = 1:8;

%PCM coding range (for the first feature)
ub_pcm = max(features(:));
lb_pcm = min(features(:));


%% DPCM CODING - NON SORTED
MSE_dpcm = zeros(size(R));
diff_feat = diff(features,1,2);
ub = max(diff_feat(:));
lb = min(diff_feat(:));
feat_out = NaN*ones(size(features));
ir = 1;
for r = R
    step_pcm = (ub_pcm-lb_pcm)/2^r;
    feat_out(:,1) = step_pcm*round(features(:,1)./step_pcm);

    step = (ub-lb)/2^r;
    for ifeat = 2:N
        e = features(:,ifeat) - feat_out(:,ifeat-1);
        eq = step*round(e./step);
        feat_out(:,ifeat) = feat_out(:,ifeat-1) + eq;
    end
    MSE_dpcm(ir) = var(features(:) - feat_out(:));
    ir = ir+1;
end

%% DPCM CODING - SORTED WITH GREEDY TSP
%sort_idx = greedy_sorting(features);
%features_sort = features(:,sort_idx);

load feature_set_sorted;
MSE_dpcm_sorted = zeros(size(R));
diff_feat = diff(features_sort,1,2);
ub = max(diff_feat(:));
lb = min(diff_feat(:));

feat_out = NaN*ones(size(features_sort));
ir = 1;
for r = R
    step_pcm = (ub_pcm-lb_pcm)/2^r;
    feat_out(:,1) = step_pcm*round(features_sort(:,1)./step_pcm);


    step = (ub-lb)/2^r;
    for ifeat = 2:N
        e = features_sort(:,ifeat) - feat_out(:,ifeat-1);
        eq = step*round(e./step);
        feat_out(:,ifeat) = feat_out(:,ifeat-1) + eq;
    end
    MSE_dpcm_sorted(ir) = var(features_sort(:) - feat_out(:));
    ir = ir+1;
end



%PLOT RESULTS
figure()
%colormap gray
subplot(121)
imagesc(features);
title('Non sorted feature set');

subplot(122)
imagesc(features_sort);
title('Sorted feature set');


figure()
hold on
plot(R,10*log10(var(features(:))./MSE_dpcm),'r');
plot(R,10*log10(var(features(:))./MSE_dpcm_sorted),'g');
xlabel('Rate [bits per element]');
ylabel('SNR [db]');
```

```
legend('DPCM - non sorted','DPCM - sorted');


% function sorted_desc = greedy_sorting(descriptors)
%
%     N = size(descriptors,2);
%     DIM = size(descriptors,1);
%
%     sorted_desc(1) = 1;
%     unproc_feat = 2:N;
%
%     ifeat = 2;
%     while ~isempty(unproc_feat)
%         %%compute distances
%         temp_feat = repmat(descriptors(:,sorted_desc(ifeat-1)),1,length(unproc_feat));
%         dist = sqrt(sum((temp_feat - descriptors(:,unproc_feat)).^2));
%
%         %find minimum distance
%         [v i] = min(dist);
%         sorted_desc(ifeat) = unproc_feat(i);
%         unproc_feat(i) = [];
%         ifeat = ifeat + 1;
%     end
% end
```