Multimedia Signal processing

Prof. Marco Tagliasacchi

September, 10 2013

1. **Question (8 pts)**:
   Consider a sequence of symbols *abbbbaaacccacbccccbcacbbbbbbc* emitted by a source X.
   (a) Learn the p.m.f. of the source from the observed data and compute the entropy of the source (assuming a memoryless source).
   (b) To exploit source memory, group symbols in blocks of size $N = 2$ and compute $H_2(X)$.
   (c) Compute the number of bits needed to encode the sequence *bbbcaa* using Huffman coding in two cases: i) without exploiting source memory; ii) exploiting source memory.

   Solution
   (a) The p.m.f. can be estimated as follows

$$Pr(X = a) = 6/30 \tag{1}$$
$$Pr(X = b) = 13/30 \tag{2}$$
$$Pr(X = c) = 11/30 \tag{3}$$

   The entropy of the source is

$$H(X) = -6/30 \log_2 6/30 - 13/30 \log_2 13/30 - 11/30 \log_2 11/30 = 1.52\text{bits / symbol} \tag{4}$$

   (b) By grouping in blocks of $N = 2$ symbols, we obtain $3^2 = 9$ supersymbols, whose probabilities are indicated below

$$Pr(X_1 = a, X_2 = a) = 1/15 \tag{5}$$
$$Pr(X_1 = a, X_2 = b) = 1/15 \tag{6}$$
$$Pr(X_1 = a, X_2 = c) = 1/15 \tag{7}$$
$$Pr(X_1 = b, X_2 = a) = 1/15 \tag{8}$$
$$Pr(X_1 = b, X_2 = b) = 4/15 \tag{9}$$
$$Pr(X_1 = b, X_2 = c) = 2/15 \tag{10}$$
$$Pr(X_1 = c, X_2 = a) = 1/15 \tag{11}$$
$$Pr(X_1 = c, X_2 = b) = 1/15 \tag{12}$$
$$Pr(X_1 = c, X_2 = c) = 3/15 \tag{13}$$

$$H_2(X) = \frac{H(x_1, x_2)}{2} = \frac{1}{2}\left(-6/15 \log_2 1/15 - 2/15 \log_2 2/15 - 3/15 \log_2 3/15 - 4/15 \log_2 4/15\right) = 1.46\text{bits / symbol} \tag{14}$$

(c) In the memoryless case, a possible Huffman code is

symbol	codeword	length
a	00	2
b	1	1
c	01	2

Hence, the sequence *bbbcaa* is encoded as 111010000, using 9 bits.

Instead, when exploiting memory, a possible Huffman code is

symbol	codeword	length
aa	0000	4
ab	0001	4
ac	0010	4
ba	0011	4
bb	01	2
bc	101	3
ca	1000	4
cb	1001	4
cc	11	2

Hence, the sequence *bbbcaa* is encoded as 011010000, using 9 bits.

2. **Question (5 pts)** Transform coding is a key component in most audio-visual coding standard.
   (a) Illustrate the basic principles of transform coding, providing a geometric intrepretation.
   (b) Indicate how transform coding is used in current audio-visual coding standards.

   Solution

   See slide set `MMSP_1.4_Transform coding`.

3. **Question (5 pts)**

   Unlike general audio signals, speech has some unique properties in the time-frequency domain
   (a) Illustrate the properties of speech signals.
   (b) Describe how state-of-the-art speech coders exploit such properties.

   Solution

   See slide set `MMSP_2.1_Speech_Coding`.

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

   (a) Let X and Y denote two discrete sources and $R(D)$ the rate-distortion function.

   - $\boxed{\text{T}}\boxed{\text{F}}$ If X and Y are both memoryless and they have the same p.m.f., $R_X(D) = R_Y(D)$.
     TRUE. A memoryless source is completely specified by its p.m.f. If two sources have the same p.m.f., they will also have the same rate-distortion function.
   - $\boxed{\text{T}}\boxed{\text{F}}$ If the number of symbols of X is greater than the number of symbols of Y, then $R_X(0) > R_Y(0)$
     FALSE. Note that $R_X(0) = H(X)$ and $R_Y(0) = H(Y)$. Although $H(X) \leq \log_2 M_Y$ and $H(Y) \leq \log_2 M_Y$, the fact that $M_X > M_Y$ does not imply that $H(X) > H(Y)$.
   - $\boxed{\text{T}}\boxed{\text{F}}$ $R_X(\sigma_X^2) = 0$.
     TRUE.
   - $\boxed{\text{T}}\boxed{\text{F}}$ Any practical coding scheme is bound to achieve an operational rate-distortion function $R_X^o(D)$ such that $R_X^o(D) \geq R_X^S(D)$, where $R_X^S(D)$ is Shannon's bound for the source X.
     TRUE. Shannon's bound is a lower bound on the number of bits needed to encode a source, given a target distortion.

   (b) Consider a JPEG codec

   - $\boxed{\text{T}}\boxed{\text{F}}$ If a JPEG compressed image I_1 is re-compressed with JPEG with the same quantization matrix to obtain I_2, then $I_1 = I_2$.
     FALSE. Although the two images are nearly identical, there are some differences due to the process of rounding and truncation that are applied when JPEG decoding I_1 in the pixel domain.
   - $\boxed{\text{T}}\boxed{\text{F}}$ The DC coefficient is encoded by means of DPCM.
     TRUE.
   - $\boxed{\text{T}}\boxed{\text{F}}$ The size of the JPEG bitstream is fixed, once the spatial resolution of the image is given.
     FALSE. The number of bits needed to encode a JPEG image is content-dependent.

   (c) Motion estimation is a key component in any video coding architecture.

   - $\boxed{\text{T}}\boxed{\text{F}}$ Motion estimation is typically performed in the DCT domain.
     FALSE. In the pixel domain.
   - $\boxed{\text{T}}\boxed{\text{F}}$ The cost of motion estimation is always quadratic in W, where W is the size of the $W \times W$ search window.
     FALSE. It is quadratic in the case of full search. However, when using alternative ME algorithms, the cost is typically sub-quadratic in W.
   - $\boxed{\text{T}}\boxed{\text{F}}$ Motion estimation is performed both at the encoder and at the decoder.
     FALSE. Motion estimation is performed only at the encoder. Instead, motion compensation is performed at both sides.

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

   (a) Generate $N = 1000$ samples of a signal obtained by filtering an i.i.d. Gaussian noise signal ($\mu_z = 0$, $\sigma_z^2 = 1$) with the following filter.

$$H(z) = \frac{1}{1 - 0.9z^{-1}} \tag{15}$$

   (b) Implement a DPCM encoder, which receives as input the following parameters:

   - $\Delta \in \{0.025, 0.05, 0.1, 0.2, 0.4\}$, which is the quantization step size

- $\alpha \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$, which is a parameter used to tune the predictor $\hat{x}(n)$, i.e., $\hat{x}(n) = \alpha\tilde{x}(n-1)$, where $\tilde{x}(n)$ represents the DPCM decoded sequence.

(c) For each value of α, plot the rate-distortion curve (Hint: approximate rate as the entropy of the quantization symbols).

(d) Compare the rate-distortion curves and comment the result.

Solution

```
clear
close all
style = {'b', 'm', 'r', 'g', 'c'};

N = 1000;
z = randn(N, 1);

x = filter(1,[1, -0.9],z);

Q = [0.025 0.05 0.1, 0.2, 0.4];

alpha = [0.8, 0.9, 1, 1.1, 1.2];

SNR = NaN*ones(length(alpha), length(Q));
R = NaN*ones(length(alpha), length(Q));

for a = 1:length(alpha)
    for q = 1:length(Q)

        xq = NaN*ones(N,1);
        eq = NaN*ones(N,1);
        es = NaN*ones(N,1);


        xq(1) = Q(q)*round(x(1)/Q(q));

        for i = 2:N

            e = x(i) - alpha(a)*xq(i-1);

            es(i) = round(e/Q(q));

            eq(i) = Q(q)*es(i);

            xq(i) = alpha(a)*xq(i-1) + eq(i);

        end

        MSE = mean((xq - x).^2);
        SNR(a,q) = 10*log10(var(x)./MSE);

        h = hist(es, min(es):max(es));
        p = h(h ~= 0);
        p = p./sum(p);
        R(a,q) = -sum(p.*log2(p));


    end
    plot(R(a,:), SNR(a,:), char(style(a)))
    hold on
end

% The best RD curve is obtained when using alpha = 0.9. Indeed, in this
% case the predictor is the same that would be obtained by means of LPC
% analysis.
```