Homework 10 (2023) — SOLUTIONS

JAIST — School of Information Science — I232 Information Theory

1. *Rate-Distortion with Erasures* Find the rate-distortion function $R(D)$ for the following problem. Consider a binary source X with $p_{\mathsf{X}}(1) = p$ from $\mathcal{X} = \{0,1\}$ This is to be coded to a ternary code with $\widehat{\mathcal{X}} = \{0, ?, 1\}$ using the distortion metric:

$$d(x,\widehat{x}) = \begin{cases} 0 & \text{if } \widehat{x} = x \\ 1 & \text{if } \widehat{x} = ? \\ \infty & \text{otherwise} \end{cases} .$$

Solution: As preparation, write the distortion matrix in matrix form:

$$d(x,\widehat{x}) = \begin{bmatrix} 0 & 1 & \infty \\ \infty & 1 & 0 \end{bmatrix},$$

where the rows are $\mathcal{X} = \{0,1\}$ and the columns are $\widehat{\mathcal{X}} = \{0,?,1\}$.

Next write a parameterized test channel. Since $d(0,1) = d(1,0) = \infty$, we have $p_{\mathsf{X}|\widehat{\mathsf{X}}}(0|1) = p_{\mathsf{X}|\widehat{\mathsf{X}}}(1|0) = 0$. Define parameter $q = p_{\mathsf{X}|\widehat{\mathsf{X}}}(1|?)$, so that $p_{\mathsf{X}|\widehat{\mathsf{X}}}(0|?) = 1-q$. By symmetry of the distortion metric, $p_{\mathsf{X}|\widehat{\mathsf{X}}}(?|1) = q$ and $p_{\mathsf{X}|\widehat{\mathsf{X}}}(1|1) = 1-q$. Then the parameterized test channel is:

$$p_{\mathsf{X}|\widehat{\mathsf{X}}}(x|\widehat{x}) = \begin{bmatrix} 1 & 0 \\ q & 1-q \\ 0 & 1 \end{bmatrix}$$

Define $r_0, r_?, r_1$ as the input distribution parameters, $p_{\widehat{\mathsf{X}}}(0) = r_0, p_{\widehat{\mathsf{X}}}(?) = r_?, p_{\widehat{\mathsf{X}}}(1) = r_1$. Clearly the joint distribution is:

$$p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x}) = \begin{bmatrix} r_0 & 0 \\ qr_? & (1-q)r_? \\ 0 & r_1 \end{bmatrix}$$

To satisfy the distortion constraint with equality:

$$\sum_{x\in\mathcal{X}} \sum_{\widehat{x}\in\widehat{\mathcal{X}}} p_{\mathsf{X},\widehat{\mathsf{X}}}(x,\widehat{x}) d(x,\widehat{x}) = D$$

$$qr_? + (1-q)r_? = r_? = D$$

Next, find $\min I(\mathsf{X};\widehat{\mathsf{X}}) = H(\mathsf{X}) - H(\mathsf{X}|\widehat{\mathsf{X}})$ by computing $\max H(\mathsf{X}|\widehat{\mathsf{X}})$:

$$\begin{aligned} H(\mathsf{X}|\widehat{\mathsf{X}}) &= \sum_{\widehat{x}\in\widehat{\mathcal{X}}} p_{\widehat{\mathsf{X}}}(x) H(\mathsf{X}|\widehat{\mathsf{X}} = \widehat{x}) \\ &= r_? h(q) = Dh(q) \\ \max H(\mathsf{X}|\widehat{\mathsf{X}}) &= \max Dh(q) = D, \end{aligned}$$

achieved by $q = \frac{1}{2}$. So the rate distortion function is:

$$\begin{aligned} R(D) &= \min I(\mathsf{X};\widehat{\mathsf{X}}) \\ &= H(\mathsf{X}) - \max H(\mathsf{X}|\widehat{\mathsf{X}}) \\ &= h(p) - D \end{aligned}$$

While not required, it is possible to find the input distribution $p_{\widehat{\mathsf{X}}}(x)$, that satisfies the source distribution $p_{\mathsf{X}}(x)$:

$$\frac{1}{2}D + r_0 = 1-p \text{ and } \frac{1}{2}D + r_1 = p$$

$$r_0 = 1 - p - \frac{D}{2} \text{ and } r_1 = p - \frac{D}{2}$$

2. *K-means for Gaussian Quantization* Perform quantization of a Gaussian source, by implementing the K-means algorithm in your favorite programming language. Generate M random samples from a zero-mean, unit variance Gaussian source, $x_1, x_2, \ldots, x_M$ (use a large value such as $M = 1000$ or $M = 10000$). Apply the K-means algorithm to obtain K reconstruction points, $m_1, m_2, \ldots, m_K$ (This is a one-dimensional K-means algorithm). The distortion function is the average mean-squared error, computed as:

$$MSE = \frac{1}{M} \sum_{k=1}^{K} \sum_{x' \in \widehat{\mathcal{X}}_i} (x' - m_i)^2$$

For a fixed data set, you may repeat the K-means algorithm several times and take the best MSE value. You should write the source code yourself and not use a library.

(a) Plot the rate-distortion function for this source.

(b) On the same plot, show the theoretical R-D pair for $K = 2$.

(c) On the same plot, show the R-D pairs obtained using your K-means algorithm, for $K = 2, 4$ and 8.

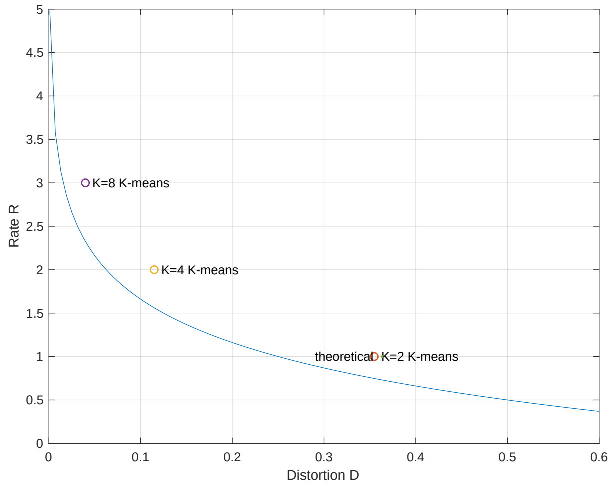
(d) Submit the source code you wrote.

Solution: Source code is provided for Matlab. The script below runs the K-Means algorithm and plots,

```
var = 1;
M   = 10000;
X   = sqrt(var) * randn(1,M);
D2  = kmeansMatlab(X,2);
D4  = kmeansMatlab(X,4);
D8  = kmeansMatlab(X,8);

cla;
D = linspace(1E-3,0.6);
plot(D,0.5*log2(var ./ D));
grid on
hold on;
plot(D2,1,'o'); text(D2,1,'  K=2 K-means')
plot(D4,2,'o'); text(D4,2,'  K=4 K-means')
plot(D8,3,'o'); text(D8,3,'  K=8 K-means')
plot((1- 2/pi) * var, 1,'.');
ha = text((1- 2/pi) * var, 1,'theoretical  ');
ha.HorizontalAlignment = 'right'
```

Resulting in this figure:



The Matlab implementation of the K-means algorithm is below:

```
function [D,m] = kmeansMatlab(X,K)

M = size(X,2);

%initialization
pi = randperm(M);
m  = X(:,pi(1:K));  %instead of xhat, use m to denote the means

for ii = 1:7

    m_prev = m;         %remember old means as stopping condition

    %precompute distance matrix
    dist = zeros(K,M);
    for kk = 1:K
        for mm = 1:M
            dist(kk,mm) = sum((m(:,kk) - X(:,mm) ) .^2);
        end
    end

    %assignment step
    [~,clusters] = min(dist);  %min of each column
    %see "help min" for details
    for kk = 1:K
        Xhat{kk} = find(clusters == kk);
    end

    %assignment (centroid) step
    for kk = 1:K
        m(:,kk) = mean(X(:, Xhat{kk})')';
    end

    if all(m - m_prev == 0)
        break;
    end
end

D = 0;
```

```
for kk = 1:K
  D = D + sum((X(:,Xhat{kk}) - m(kk)).^2);
end
D = D / M;

return

cla
plot(X,0,'b.','markersize',12)
hold on
plot(m,0,'r.','markersize',24)
```