# Homework 10

28 May 2023 12:31

10.1

- $p_X(x) = [1-p \quad p]$ ; $d(x,\hat{x}) = \begin{cases} 0 & x = \hat{x} \\ 1 & \hat{x} = ? \\ \infty & \text{otherwise} \end{cases}$

- $p_{X|\hat{X}}(x|\hat{x}) = \begin{bmatrix} 1 & 0 \\ r & 1-r \\ 0 & 1 \end{bmatrix}$

- $p_{\hat{X}}(\hat{x}) = [s \quad t \quad 1-s-t]$

- $p_{X,\hat{X}}(x,\hat{x}) = \begin{bmatrix} s & tr & 0 \\ 0 & t(1-r) & 1-s-t \end{bmatrix}$

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

$= tr + t(1-r)$

$= t$

- $\sum_{\hat{x}} p_{X,\hat{X}}(x,\hat{x}) = p_X(x)$

$\Leftrightarrow \begin{cases} s + tr = 1-p \\ t(1-r) + 1 - s - t = p \end{cases}$

$\Leftrightarrow \begin{cases} s + Dr = 1-p \\ D(1-r) + 1 - s - D = p \end{cases}$

$\Leftrightarrow s = 1 - p - rD$

- $H(X|\hat{X}) = \sum_{\hat{x}} p_{\hat{X}}(\hat{x})\, H(X|\hat{X}=\hat{x})$

$= p_{\hat{X}}(0) \cdot 0 + p_{\hat{X}}(1) \cdot 0 + p_{\hat{X}}(?)\, h(r)$

$= D \cdot h(r)$

$$= p_{\hat{X}}(0) \cdot 0 + p_{\hat{X}}(1) \cdot 0 + p_{\hat{X}}(?)\, h(r)$$
$$= D \cdot h(r)$$

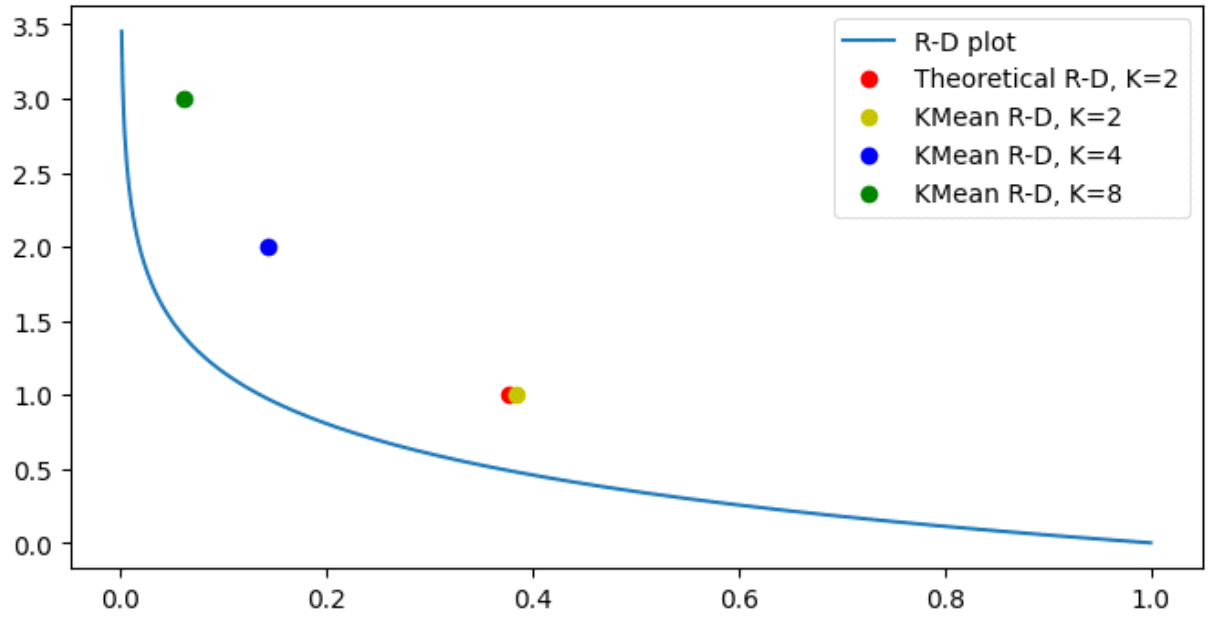
* $R(D) = \min_{p_{X|\hat{X}}} I(X;\hat{X})$
$= \min_{p_{X|\hat{X}}} H(X) - H(X|\hat{X})$
$= \min_{r} h(p) - D\,h(r)$
$\geq h(p) - D$

with equality when $r = \frac{1}{2}$

If $0 \leq D \leq h(p)$, we can achieve $R(D) = h(p) - D$

If $h(p) < D$, we can achieve $R(D) = 0$ by letting $D = h(p)$

10.3




```
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams["figure.figsize"] = (8, 4)

def rate_distortion_func(var, D):
    return 0.5 * np.log(var / D)
```

```
def k_mean_mse_func(X, C):
    R = {i: [] for i in range(len(C))}
    mse = 0
    for i, x in enumerate(X):
        x_ind = np.argmin([(x - c) ** 2 for c in C])
        R[x_ind].append(x)

    for i in range(len(C)):
        if len(R[i]) == 0:
            R[i] = np.random.choice(X, size=1)
        C[i] = np.mean(R[i])
        mse += np.sum(np.power(np.array(R[i]) - C[i], 2))

        R[i] = []
    mse /= M
    return C, mse

M = 1000
X = np.random.normal(size=M)

var = 1
D = np.linspace(0, 1, 1000)
R = [rate_distortion_func(var, d) for d in D]

plt.plot(D, R, label="R-D plot")

C_K2 = [-np.sqrt(2 / np.pi), np.sqrt(2 / np.pi)]
_, D_k2 = k_mean_mse_func(X, C_K2)
R_k2 = 1
plt.plot(D_k2, R_k2, "ro", label="Theoretical R-D, K=2")

colors = ["yo", "bo", "go"]
for i, K in enumerate([2, 4, 8]):
    mse = None
    C = np.random.choice(X, size=K, replace=False)
    while True:
        C, cur_mse = k_mean_mse_func(X, C)
        if mse is not None and np.abs(cur_mse - mse) <= 1e-2:
            break
        mse = cur_mse
    R_ki = np.log2(K)
    plt.plot(mse, R_ki, colors[i], label=f"KMean R-D, K={K}")

plt.legend(loc='upper right')
plt.show()
```