Data Processing and Representation – HW 1

Submitters:

ID	Email
315804831	Ido.amit@campus.technion.ac.il
315777433	hay.e@campus.technion.ac.il

Theory

Question 1

a.

$$\hat{f}(x) = \begin{cases} \varphi_i, & \exists i \in [N]: x \in I_i \\ 0, & else \end{cases}$$

P=1:

In that case, the optimal $\hat{f}(x)$ is the median of $f(x)$ in each interval I_i:

$$\varphi_i = median\{\varphi(t) | t \in [\Delta(i-1), \Delta i]\}$$

P=2:

In that case, the optimal $\hat{f}(x)$ is the mean of $f(x)$ in each interval I_i:

$$\varphi_i = \frac{1}{\Delta}\int_{(i-1)\Delta}^{i\Delta} \varphi(t)dt$$

b. For general ω, the optimal $\hat{f}_P$ when P=2 is:

$$\underset{\hat{f}}{\operatorname{argmin}} \int_0^1 \left|f(x) - \hat{f}(x)\right|^2 \omega(x)dx$$

In order to find the minimizer, we rewrite the integral:

Instead of calculating the integral over [0,1], we could separate it to N integrals – each over the I_i interval (guaranteed by integral linearity). In addition, for all x in I_i interval: $\hat{f}(x) = \varphi i$:

$$\int_0^1 \left|f(x) - \hat{f}(x)\right|^2 \omega(x)dx = \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|^2 \omega(x)dx$$

Next, we differentiate w.r.t φ_i:

Derivative of a sum equals the sum of the derivatives:

$$\frac{\partial}{\partial \varphi_i} \underset{\hat{f}}{\operatorname{argmin}}\, \varepsilon^2\left(f - \hat{f}\right) = \frac{\partial}{\partial \varphi_i} \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|^2 \omega(x)dx$$

$$= \sum_{i=1}^{N} \frac{\partial}{\partial \varphi_i} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|^2 \omega(x)dx$$

For all $j \in [N]$ *so that* $j \neq i$:

$$\frac{\partial}{\partial \varphi_i} \int_{\frac{j-1}{N}}^{\frac{j}{N}} \left|f(x) - \varphi_j\right|^2 \omega(x)dx = 0$$

Then:

$$= -2 \int_{\frac{i-1}{N}}^{\frac{i}{N}} (f(x) - \varphi_i)\omega(x)dx$$

$$\underbrace{\Longrightarrow}_{find\ minimizer} \quad -2 \int_{\frac{i-1}{N}}^{\frac{i}{N}} (f(x) - \varphi_i)\omega(x)dx = 0$$

$$\Longrightarrow \int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)\omega(x)dx = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \varphi_i \omega(x)dx$$

$$\Longrightarrow \boldsymbol{\varphi_i^{opt}} = \frac{\int_{\frac{i-1}{N}}^{\frac{i}{N}} \boldsymbol{f(x)\omega(x)dx}}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} \boldsymbol{\omega(x)dx}}$$

Then for general ω the optimal $\hat{f}_2$ is the **weighted mean** of $f(x)$ in each I_i interval:

$$\hat{f}_2(x) = \begin{cases} \varphi_i^{opt}, & \exists i \in [N]: x \in I_i \\ 0, & else \end{cases}$$

c. For general ω, the optimal $\hat{f}_P$ when P=1 is:

$$\underset{\hat{f}}{\operatorname{argmin}} \int_0^1 \left|f(x) - \hat{f}(x)\right|^1 \omega(x)dx$$

In order to find the minimizer, we rewrite the integral:

Instead of calculating the integral over [0,1], we could separate it to N integrals – each over the I_i interval (guaranteed by integral linearity). In addition, for all x in I_i interval: $\hat{f}(x) = \varphi i$:

$$\int_0^1 \left|f(x) - \hat{f}(x)\right|^1 \omega(x)dx = \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|\omega(x)dx$$

Next we differentiate w.r.t φ_i:

Derivative of a sum equals the sum of the derivatives:

$$\frac{\partial}{\partial \varphi_i} \underset{\hat{f}}{\operatorname{argmin}}\, \varepsilon^1\left(f - \hat{f}\right) = \frac{\partial}{\partial \varphi_i} \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|\omega(x)dx$$

$$= \sum_{i=1}^{N} \frac{\partial}{\partial \varphi_i} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|\omega(x)dx$$

For all $j \in [N]$ *so that* $j \neq i$:

$$\frac{\partial}{\partial \varphi_i} \int_{\frac{j-1}{N}}^{\frac{j}{N}} \left|f(x) - \varphi_j\right|\omega(x)dx = 0$$

Then:

$$= \frac{\partial}{\partial \varphi_i} \int_{\frac{i-1}{N}}^{\frac{i}{N}} |f(x) - \varphi_i|\omega(x)dx$$

We note that:

$$\frac{\partial}{\partial \varphi_i}|f(t) - \varphi_i| = \frac{\partial}{\partial \varphi_i}\sqrt{(f(t) - \varphi_i)^2} = \frac{-2(f(t)-\varphi_i)}{2\sqrt{(f(t)-\varphi_i)^2}} = -\frac{(f(t)-\varphi_i)}{|f(t)-\varphi_i|} = -sign(f(t) - \varphi_i)$$

Then:

$$= -\int_{\frac{i-1}{N}}^{\frac{i}{N}} sign(f(x) - \varphi_i)\omega(x)dx$$

$$= \int_{x:f(x)>\varphi_i} 1\, \omega(x)dx - \int_{x:f(x)<\varphi_i} (-1)\omega(x)dx - \int_{x:f(x)=\varphi_i} 0\, \omega(x)dx$$

$$\underbrace{=}_{\int 0 = 0} \int_{x:f(x)>\varphi_i} 1\, \omega(x)dx - \int_{x:f(x)<\varphi_i} (-1)\omega(x)dx$$

$$\underbrace{\Longrightarrow}_{find\ minimizer} \int_{x:f(x)>\varphi_i} 1\ \omega(x)dx - \int_{x:f(x)<\varphi_i} (-1)\omega(x)dx = 0$$

$$\Longrightarrow \int_{x:f(x)>\varphi_i} \boldsymbol{\omega(x)dx} = \int_{x:f(x)<\varphi_i} \boldsymbol{\omega(x)dx}$$

Then for general ω the optimal $\hat{f}_2$ is the **weighted median** of $f(x)$ in each I_i interval.

d. Let us define new functions per interval using indicators and the original functions:

$$f_i(x) = 1\{x \in I_i\} \cdot f(x)$$
$$\hat{f}_i(x) = 1\{x \in I_i\} \cdot \hat{f}(x)$$

And the new error per interval:

$$\varepsilon_i^P(f_i, \hat{f}_i) = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^P \omega(x)dx$$

Our motivation is to show that:

$$\sum_{i=1}^{N} \varepsilon_i^P(f_i, \hat{f}_i) = \varepsilon^P(f - \hat{f})$$

$$\sum_{i=1}^{N} \varepsilon_i^P(f_i, \hat{f}_i) = \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^P \omega(x)dx \underbrace{=}_{indicators=1\ in\ I_i}$$

$$= \sum_{i=1}^{N} \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f(x) - \hat{f}(x)\right|^P \omega(x)dx \underbrace{=}_{linearity}$$

$$= \int_0^1 \left|f(x) - \hat{f}(x)\right|^P \omega(x)dx \underbrace{=}_{by\ definition} \varepsilon^P(f - \hat{f})\ \blacksquare$$

e.

i)

Our motivation is to show that $\omega_{f_i,\hat{f}}(x) = \frac{|f_i(x)-\hat{f}_i(x)|^P}{|f_i(x)-\hat{f}_i(x)|^2}$ is defined and positive:

It is defined because we assume that $f_i(x) \neq \hat{f}_i(x)\ for\ all\ x \in I_i$

$$\Longrightarrow f_i(x) - \hat{f}_i(x) \neq 0$$

It is positive because $\left|f_i(x) - \hat{f}_i(x)\right|^P$ and $|f_i(x) - \hat{f}_i(x)|^2$ are positive as absolute values, powered.

Since $|f_i(x) - \hat{f}_i(x)|^2 = (f_i(x) - \hat{f}_i(x))^2$, we can choose the following function:

$$\boldsymbol{\omega_{f_i,\hat{f}}(x)} = \frac{\left|f_i(x) - \hat{f}_i(x)\right|^P}{\left(f_i(x) - \hat{f}_i(x)\right)^2} = \boldsymbol{|f_i(x) - \hat{f}_i(x)|^{P-2}}$$

That satisfies the equation:

$$\left(f_i(x) - \hat{f}_i(x)\right)^2 = \omega_{f_i,\hat{f}}(x)\left|f_i(x) - \hat{f}_i(x)\right|^P$$

ii)

$$\varepsilon_i^P(f_i, \hat{f}_i) = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^P \cdot 1 \cdot \omega(x)dx =$$

$$\int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^P \frac{\left|f_i(x) - \hat{f}_i(x)\right|^2}{|f_i(x) - \hat{f}_i(x)|^2} \omega(x) dx = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^2 \omega_{f_i,\hat{f}}(x)\omega(x)dx$$

Let us define a new function: $\omega'_{f_i,\hat{f}}(x) = \omega_{f_i,\hat{f}}(x)\omega(x)$

Then:

$$\varepsilon_i^P\left(f_i, \hat{f}_i\right) = \int_{\frac{i-1}{N}}^{\frac{i}{N}} \left|f_i(x) - \hat{f}_i(x)\right|^2 \omega'_{f_i,\hat{f}}(x)dx \blacksquare$$

iii)

It would be much simpler if the weight function was independent of $\hat{f}_i$ $(which\ we\ name\ \varphi_i)$ because we have found that:

$$\varphi_i^{opt} = \frac{\int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)\omega(x)dx}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} \omega(x)dx}$$

That relies on $\omega(x)$ independency of φ_i (and we can calculate it).

Given it would be independent of it, we will have to recalculate and not necessarily be able to calculate.

The weight function $\omega'_{f_i,\hat{f}}(x)$ <u>is dependent</u> of φ_i, and we can't calculate it.

iv)

recall:

$$\omega_{f_i,\hat{f}}(x) = |f_i(x) - \hat{f}_i(x)|^{P-2}$$

And the new weight function:

$$\widetilde{\omega}_{f_i,\hat{f}}(x) = \min\left\{\frac{1}{\varepsilon}, \left|f_i(x) - \hat{f}_i(x)\right|^{P-2}\right\}$$

When $\varepsilon > 0$ a small fixed number.

Our motivation is to show that in the worst case it's better, and in the general case it doesn't change much (statistically):

$P < 2$:

In this case, $\omega_{f_i,\hat{f}}(x) = \frac{1}{\left|f_i(x) - \hat{f}_i(x)\right|^C}$ for $C = -(P-2) > 0$

The function $\hat{f}_i(x)$ is an estimator of $f_i(x)$, therefore when it estimates well, we get that $\left|f_i(x) - \hat{f}_i(x)\right|^C \to 0$ and $\omega_{f_i,\hat{f}}(x) \to \infty$, enlarging the error when in fact our estimator estimates well.

in order to prevent this behavior, we can bound from top with a big but not infinitely big constant: $\frac{1}{\varepsilon}$.

$P \geq 2$:

In this case, $\omega_{f_i,\hat{f}}(x) = \left|f_i(x) - \hat{f}_i(x)\right|^C$ for $C = P - 2 > 0$

The function $\hat{f}_i(x)$ is an estimator of $f_i(x)$, therefore when it estimates well, we get that $\left|f_i(x) - \hat{f}_i(x)\right|^C \to 0$ and $\omega_{f_i,\hat{f}}(x) \to 0$, which is good because taking the minimum between $\omega_{f_i,\hat{f}}(x) \to 0$ and $\frac{1}{\varepsilon}$, we will most probably (for a large enough $\frac{1}{\varepsilon}$) get the original value of $\omega_{f_i,\hat{f}}(x)$.

To sum up, the new weight function $\widetilde{\omega}_{f_i,\hat{f}}(x)$ keeps stability and correctness.

v)

$\hat{f}_i(x) = 0$

$while(\left|\varepsilon_i^P(f_i, \hat{f}_i) - \varepsilon_i^P(f_i, \hat{f}_i^{next})\right| > \delta)$:

$$\omega'_{f_i, \hat{f}_i}(x) = \min\left\{\frac{1}{\varepsilon}, \left|f_i(x) - \hat{f}_i(x)\right|^{P-2}\omega(x)\right\}$$

$$\hat{f}_i^{next} = \frac{\int_{\frac{i-1}{N}}^{\frac{i}{N}} f(x)\omega'_{f_i, \hat{f}_i}(x)dx}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} \omega'_{f_i, \hat{f}_i}(x)dx}$$

$$\hat{f}_i(x) = \hat{f}_i^{next}(x)$$

$Return\ \hat{f}_i^{next}(x)$

f.

let us define the algorithm from the previous question as "$Prev - Algorithm$",

given the inputs: $f(x), \omega(x), P, \delta$:

output: $\hat{f}(x)$ – the approximate optimal solution.

Algorithm($f(x), \omega(x), P, \delta$):

For i in $\{1, \dots, N\}$:

$$\hat{f}_i(x) = Prev - Algorithm(f(x), \omega(x), P, \delta)$$

Define:

$$\hat{f}(x) = \begin{cases} \hat{f}_i(x), & x \in I_i \\ 0, & otherwise \end{cases}$$

Return $\hat{f}(x)$

Question 2

a.

$$\int_{t\in\Delta_i}(t-t_i)^k dt == \left[\frac{(t-t_i)^{k+1}}{k+1}\right]_{t=\frac{i-1}{N}}^{t=\frac{i}{N}} = \left[\frac{\left(t-\frac{\frac{i}{N}+\frac{i+1}{N}}{2}\right)^{k+1}}{k+1}\right]_{t=\frac{i-1}{N}}^{t=\frac{i}{N}} =$$

$$= \left(\frac{\left(\frac{2i}{2N}-\frac{2i-1}{2N}\right)^{k+1}}{k+1} - \frac{\left(\frac{2(i-1)}{2N}-\frac{2i-1}{2N}\right)^{k+1}}{k+1}\right) = \left(\frac{\left(\frac{1}{2N}\right)^{k+1}}{k+1} - \frac{\left(\frac{-1}{2N}\right)^{k+1}}{k+1}\right) =$$

$$= \frac{\left(\frac{1}{N}\right)^{k+1} - \left(\frac{-1}{N}\right)^{k+1}}{2^{k+1}(k+1)} = \frac{(\Delta_i)^{k+1}-(-\Delta_i)^{k+1}}{2^{k+1}(k+1)}$$

- If k is odd:
  $(\Delta_i)^{k+1}$ = $(-\Delta_i)^{k+1}$ because k+1 is even and both positive
  Then:
  $$\frac{(\Delta_i)^{k+1}-(-\Delta_i)^{k+1}}{2^{k+1}(k+1)} = \frac{0}{2^{k+1}(k+1)} = 0$$
- If k is even:
  $(\Delta_i)^{k+1}$ = $-(-\Delta_i)^{k+1}$ because k+1 is odd
  Then:
  $$\frac{(\Delta_i)^{k+1}-(-\Delta_i)^{k+1}}{2^{k+1}(k+1)} = \frac{2(\Delta_i)^{k+1}}{2^{k+1}(k+1)} = \frac{|\Delta_i|^{k+1}}{2^k(k+1)}$$

$$\Rightarrow \int_{t\in\Delta_i}(t-t_i)^k dt = \begin{cases} 0, & k \text{ is odd} \\ \frac{|\Delta_i|^{k+1}}{2^k(k+1)}, & k \text{ is even} \end{cases}$$

b.

In order to find the optimal coefficients a_i, c_i we will calculate the derivative of the MSE function – first w.r.t a_i and compare to 0, then do the same for c_i:

$$MSE = \frac{1}{1}\int_0^1 \left(\phi(t)-\hat{\phi}(t)\right)^2 dt$$

Separate the integral to each interval thanks to linearity:

$$= \sum_{i=1}^{N}\frac{1}{N}\int_{\Delta_i}\left(\phi(t)-\hat{\phi}(t)\right)^2 dt = \sum_{i=1}^{N}\frac{1}{N}\int_{\Delta_i}\left(\phi(t)-(a_i(t-t_i)+c_i)\right)^2 dt$$

Calculate the derivative w.r.t a:

$$\frac{\partial MSE}{\partial a_i} = \frac{\partial}{\partial a_i}\sum_{i=1}^{N}\frac{1}{N}\int_{\Delta_i}\left(\phi(t)-(a_i(t-t_i)+c_i)\right)^2 dt = 0$$

$$\Rightarrow -\frac{2}{N}\int_{\Delta_i}\left(\phi(t)-\left(a_i^{opt}(t-t_i)+c_i\right)\right)(t-t_i)dt=0$$

$$\Rightarrow \int_{\Delta_i}\phi(t)(t-t_i)dt-a_i^{opt}\int_{\Delta_i}(t-t_i)^2dt+c_i\int_{\Delta_i}(t-t_i)dt=0$$

According to the previous question, $\int_{\Delta_i}(t-t_i)^{k=1}dt=0$ since k=1 is odd.

In addition, according to the previous question, $\int_{\Delta_i}(t-t_i)^2dt=\frac{|\Delta_i|^3}{12}$ since k=2 is even.

$$\Rightarrow \int_{\Delta_i}\phi(t)(t-t_i)dt-a_i^{opt}\frac{|\Delta_i|^3}{12}=0$$

$$\Rightarrow a_i^{opt}=\frac{12}{|\Delta_i|^3}\int_{\Delta_i}\phi(t)(t-t_i)dt$$

Since $\Delta_i=\frac{1}{N}$ for all i, we get:

$$\Rightarrow a_i^{opt}=12N^3\int_{\Delta_i}\phi(t)(t-t_i)dt$$

Now we calculate derivative w.r.t c_i:

$$\frac{\partial MSE}{\partial c_i}=\frac{\partial}{\partial c_i}\sum_{i=1}^{N}\frac{1}{N}\int_{\Delta_i}\left(\phi(t)-(a_i(t-t_i)+c_i)\right)^2dt=0$$

$$\Rightarrow \frac{2}{N}\int_{\Delta_i}\phi(t)-\left(a_i^{opt}(t-t_i)+c_i^{opt}\right)dt=0$$

$$\Rightarrow \int_{\Delta_i}\phi(t)dt-a_i^{opt}\int_{\Delta_i}(t-t_i)dt-c_i^{opt}\int_{\Delta_i}dt=0$$

According to the previous question, $\int_{\Delta_i}(t-t_i)^{k=1}dt=0$ since k=1 is odd.

$$\Rightarrow \int_{\Delta_i}\phi(t)dt-c_i^{opt}|\Delta_i|=0$$

$$\Rightarrow c_i^{opt}=\frac{1}{|\Delta_i|}\int_{\Delta_i}\phi(t)dt=N\int_{\Delta_i}\phi(t)dt$$

c.

As we saw in the tutorial:

$$\boldsymbol{MSE^{opt}}=\frac{1}{1}\int_0^1\left(\phi(t)\right)^2dt-\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\phi}^{opt}(t)\right)^2=\sum_{j=1}^{N}\int_{\Delta_j}\left(\phi(t)\right)^2dt-\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\phi}^{opt}(t)\right)^2=$$

$$=\sum_{j=1}^{N}\int_{\Delta_j}\left(\phi(t)\right)^2dt-\frac{1}{N}\sum_{i=1}^{N}\left(a_i^{opt}(t-t_i)+c_i^{opt}\right)^2$$

$$=\sum_{j=1}^{N}\int_{\Delta_j}\left(\phi(t)\right)^2dt-\frac{1}{N}\sum_{i=1}^{N}\left(12N^3\int_{\Delta_i}\phi(t)(t-t_i)dt\,(t-t_i)+N\int_{\Delta_i}\phi(t)dt\right)^2$$

d.

We eish to compare between the minimal MSE for using piecewise-constant approximation and the minimal MSE for using piecewise-linear approximation:

The minimal MSE for using piecewise-constant:

$$MSE_{CONST}^{opt} = \int_0^1 (\phi(t))^2 dt - \sum_{i=1}^{N} \left(\theta_i^{opt}\right)^2$$

For θ_i^{opt} – the optimal piecewise-constant for each interval I_i, where:

$$\theta_i^{opt} = \frac{1}{N} \int_{\Delta_i} \phi_i(t) dt$$

We will compute the difference:

$$MSE_{CONST}^{opt} - MSE_{LIN}^{opt} =$$

$$= \left(\int_0^1 (\phi(t))^2 dt - \sum_{i=1}^{N} \left(\theta_i^{opt}\right)^2 \right) - \left(\sum_{j=1}^{N} \int_{\Delta_j} \left(\phi(t)\right)^2 dt - \frac{1}{N} \sum_{i=1}^{N} \left(a_i^{opt}(t - t_i) + c_i^{opt}\right)^2 \right)$$

$$= \sum_{i=1}^{N} \left(\int_{\Delta_i} \left(a_i^{opt}(t - t_i) + c_i^{opt}\right)^2 dt - \left(\theta_i^{opt}\right)^2 \right)$$

Then for each interval I_i we get:

$$\int_{\Delta i} \left(a_i^{opt}(t - t_i) + c_i^{opt}\right)^2 dt - (\theta_i^{opt})^2 = \left[\frac{\left(a_i^{opt}(t - t_i) + c_i^{opt}\right)^3}{3a_i^{opt}} \right]_{\frac{i-1}{N}}^{\frac{i}{N}} - (\theta_i^{opt})^2 =$$

$$\left[\frac{\left(a_i^{opt}\left(t - \frac{2i-1}{2N}\right) + c_i^{opt}\right)^3}{3a_i^{opt}} \right]_{\frac{i-1}{N}}^{\frac{i}{N}} - (\theta_i^{opt})^2 =$$

$$= \frac{\left(a_i^{opt}\left(\frac{2i}{2N} - \frac{2i-1}{2N}\right) + c_i^{opt}\right)^3}{3a_i^{opt}} - \frac{\left(a_i^{opt}\left(\frac{2i-2}{2N} - \frac{2i-1}{2N}\right) + c_i^{opt}\right)^3}{3a_i^{opt}} - (\theta_i^{opt})^2 =$$

$$= \frac{\left(\frac{a_i^{opt}}{2N} + c_i^{opt}\right)^3 - \left(-\frac{a_i^{opt}}{2N} + c_i^{opt}\right)^3}{3a_i^{opt}} - (\theta_i^{opt})^2 =$$

$$= \frac{\left(2Nc_i^{opt} + a_i^{opt}\right)^3 - \left(2Nc_i^{opt} - a_i^{opt}\right)^3}{24a_i^{opt}N^3} - (\theta_i^{opt})^2 = \frac{(A+B)^3 - (A-B)^3}{24a_i^{opt}N^3} - (\theta_i^{opt})^2$$

For A = $2Nc_i^{opt}$ and B = a_i^{opt}, we will use the following formula:

$$(A+B)^3 - (A-B)^3 =$$

$$A^3 + 3A^2B + 3B^2A + B^3 - (A^3 - 3A^2B + 3AB^2 - B^3) = 6A^2B + 2B^3$$

Then:

$$= \frac{24N^2\left(c_i^{opt}\right)^2 a_i^{opt} + 2{a_i^{opt}}^3}{24a_i^{opt}N^3} - {\theta_i^{opt}}^2 = \frac{12N^2(c_i^{opt})^2 + {a_i^{opt}}^2}{12N^3} - {\theta_i^{opt}}^2 =$$

$$= \frac{{c_i^{opt}}^2}{N} + \frac{{a_i^{opt}}^2}{12N^3} - {\theta_i^{opt}}^2$$

Let us plug in the values of a_i^{opt}, c_i^{opt} that we calculated before:

$$= \frac{\left(N\int_{\Delta i}\phi_i(t)\,dt\right)^2}{N} + \frac{\left(12N^3\int_{\Delta i}\phi_i(t)(t-t_i)\,dt\right)^2}{12N^3} - \left(\frac{1}{N}\int_{\Delta i}\phi_i(t)\,dt\right)^2 =$$

$$= \frac{(N^3-1)}{N^2}\left(\int_{\Delta i}\phi_i(t)\,dt\right)^2 + 12N^3\left(\int_{\Delta i}\phi_i(t)(t-t_i)\,dt\right)^2$$

We note the following:

- $\frac{(N^3-1)}{N^2} \geq 0 \ and \ 12N^3 \geq 0$ since N is a natural number > 0
- $\left(\int_{\Delta i}\phi_i(t)\,dt\right)^2 \geq 0$ and $\left(\int_{\Delta i}\phi_i(t)(t-t_i)\,dt\right)^2 \geq 0$ since both are squared

$$\Rightarrow \frac{(N^3-1)}{N^2}\left(\int_{\Delta i}\phi_i(t)\,dt\right)^2 + 12N^3\left(\int_{\Delta i}\phi_i(t)(t-t_i)\,dt\right)^2 \geq 0$$

Meaning:

$$MSE_{CONST}^{opt} - MSE_{LIN}^{opt} \geq 0$$

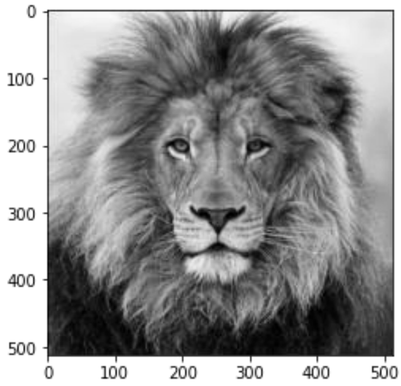
$$\Rightarrow MSE_{CONST}^{opt} \geq MSE_{LIN}^{opt}$$

We conclude that the linear-piecewise approximation MSE is lower than the constant version, as we predicted.

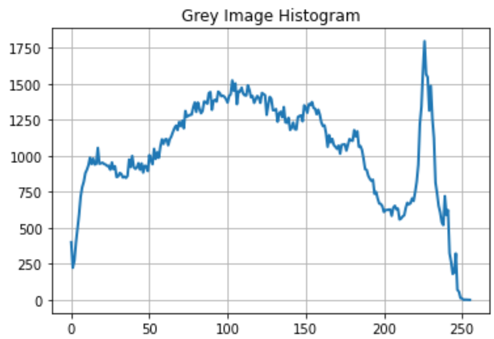
Implementation

Question 1

1. The photo we chose is as follows:

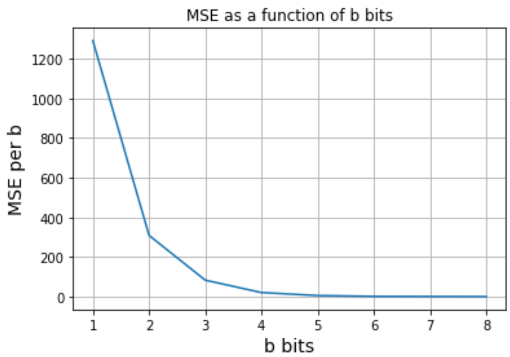


And the relevant histogram:

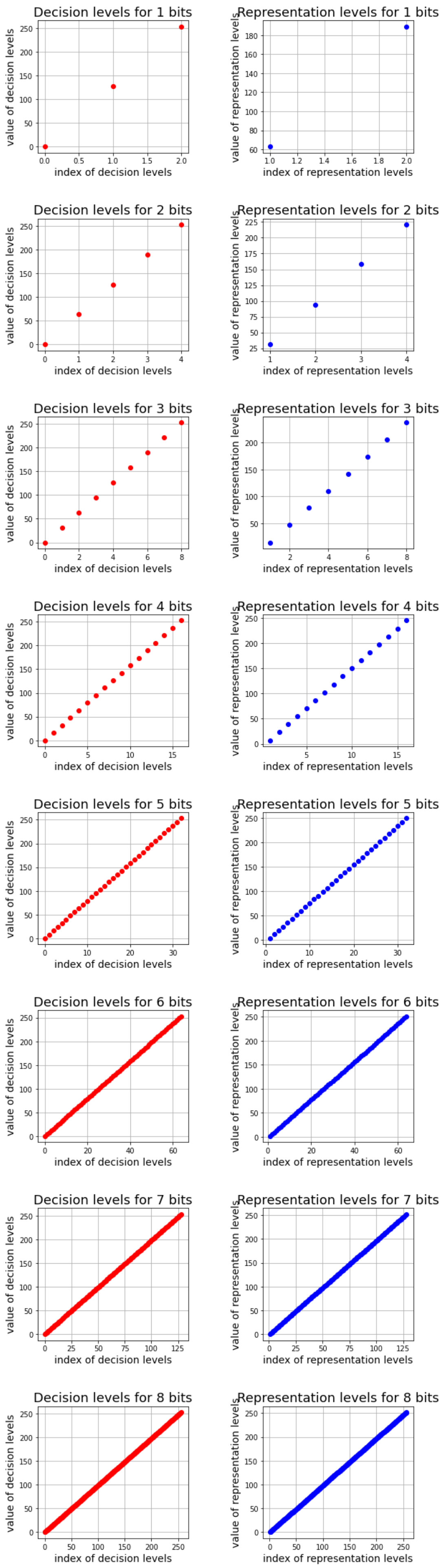


2.

a. By applying the uniform quantization, we get the following MSE as a function of b bit budgets:



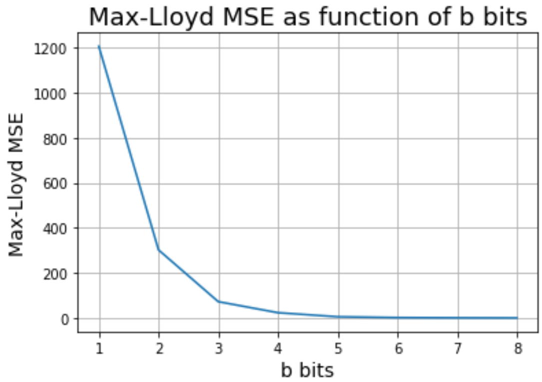
b. By applying the uniform quantization, we get the following decision levels and representation levels as a function of b bit budgets, as described in the following page:


Decision levels for 1 bits
value of decision levels
index of decision levels
Representation levels for 1 bits
value of representation levels
index of representation levels
Decision levels for 2 bits
value of decision levels
index of decision levels
Representation levels for 2 bits
value of representation levels
index of representation levels
Decision levels for 3 bits
value of decision levels
index of decision levels
Representation levels for 3 bits
value of representation levels
index of representation levels
Decision levels for 4 bits
value of decision levels
index of decision levels
Representation levels for 4 bits
value of representation levels
index of representation levels
Decision levels for 5 bits
value of decision levels
index of decision levels
Representation levels for 5 bits
value of representation levels
index of representation levels
Decision levels for 6 bits
value of decision levels
index of decision levels
Representation levels for 6 bits
value of representation levels
index of representation levels
Decision levels for 7 bits
value of decision levels
index of decision levels
Representation levels for 7 bits
value of representation levels
index of representation levels
Decision levels for 8 bits
value of decision levels
index of decision levels
Representation levels for 8 bits
value of representation levels
index of representation levels

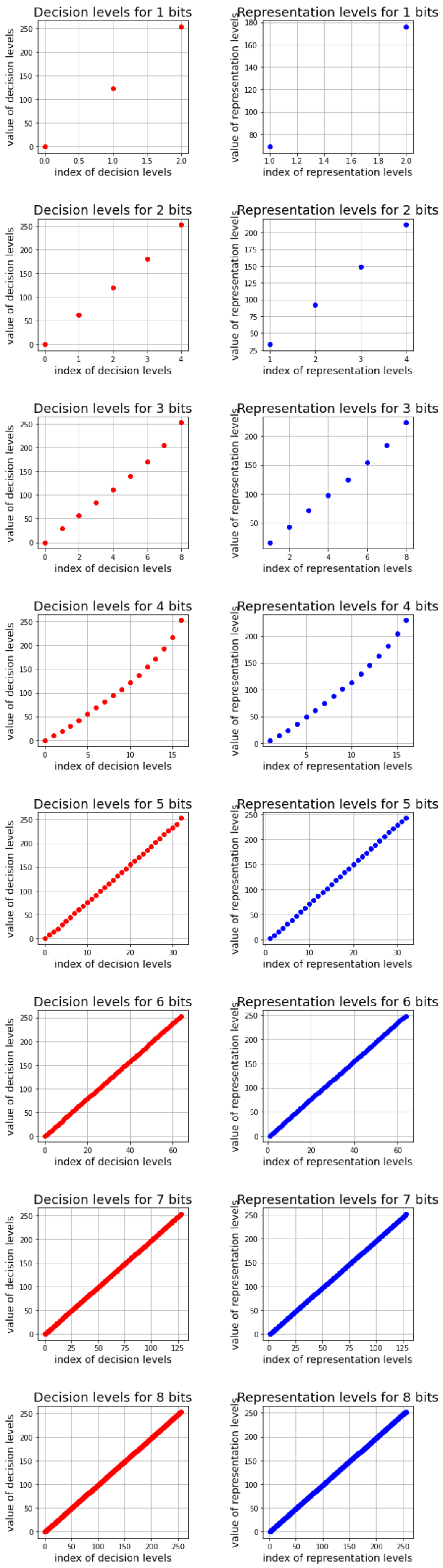
3. Implemented in the attached notebook

4. Aa

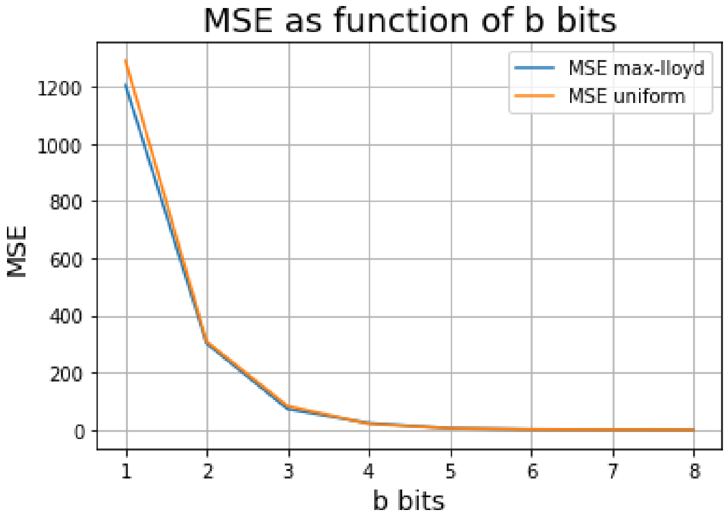
a. By applying Max-Lloyd algorithm, we get the following MSE as a function of b bit budgets:



b. By applying Max-Lloyd algorithm, we get the following decision levels and representation levels as a function of b bit budgets, as described in the following page:


Decision levels for 1 bits
value of decision levels
index of decision levels
Representation levels for 1 bits
value of representation levels
index of representation levels
Decision levels for 2 bits
value of decision levels
index of decision levels
Representation levels for 2 bits
value of representation levels
index of representation levels
Decision levels for 3 bits
value of decision levels
index of decision levels
Representation levels for 3 bits
value of representation levels
index of representation levels
Decision levels for 4 bits
value of decision levels
index of decision levels
Representation levels for 4 bits
value of representation levels
index of representation levels
Decision levels for 5 bits
value of decision levels
index of decision levels
Representation levels for 5 bits
value of representation levels
index of representation levels
Decision levels for 6 bits
value of decision levels
index of decision levels
Representation levels for 6 bits
value of representation levels
index of representation levels
Decision levels for 7 bits
value of decision levels
index of decision levels
Representation levels for 7 bits
value of representation levels
index of representation levels
Decision levels for 8 bits
value of decision levels
index of decision levels
Representation levels for 8 bits
value of representation levels
index of representation levels

c. The following graph compares between the MSE by uniform quantization and the MSE by Max-Lloyd algorithm:



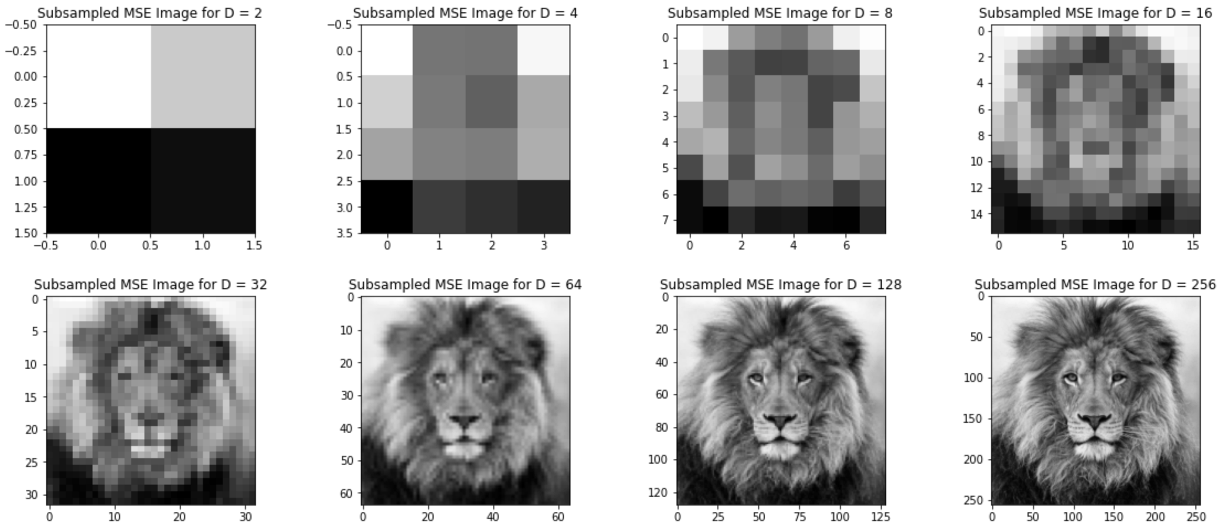
First we notice Max Lloyd algorithm achieves a lower MSE compared to the uniform quantization, as we predicted (since our histogram is far from being uniform).

As b variable get larger, the two cases MSEs get closer since the intervals become small enough so that the middle of each interval (which is chosen as representative by uniform quantizer) is very close to the mean in the interval (which is chosen as representative by Max Lloyd algorithm). Also, the decision levels calculated by Max Lloyd algorithm increases which makes the intervals smaller, thus the Max Lloyd decision levels become closer to the uniform quantizer decision levels. Overall, for a big enough b value – both cases behave very similar.

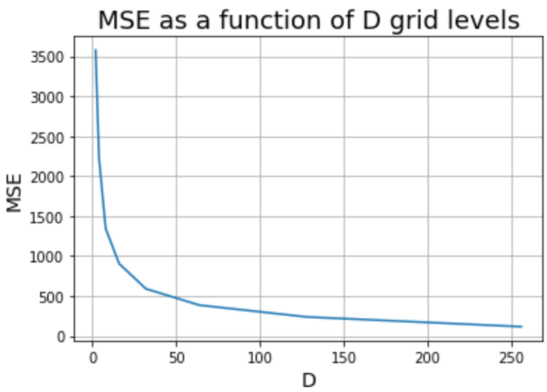
Question 2

1.

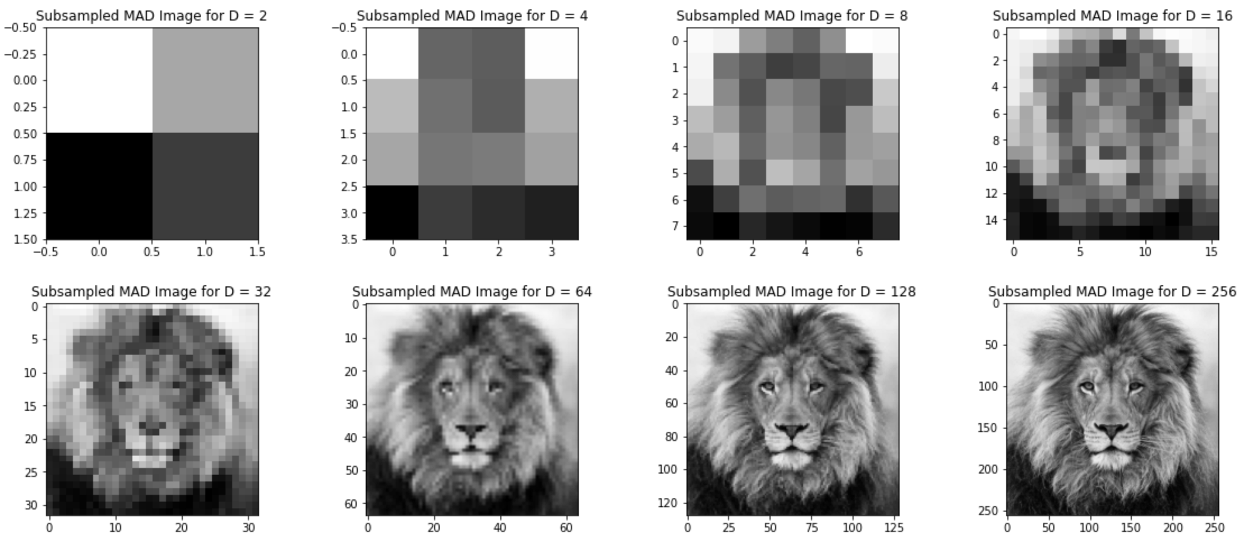
a. The sub-sampled images for $D \in \{2^1, .., 2^8\}$ in the MSE sense:



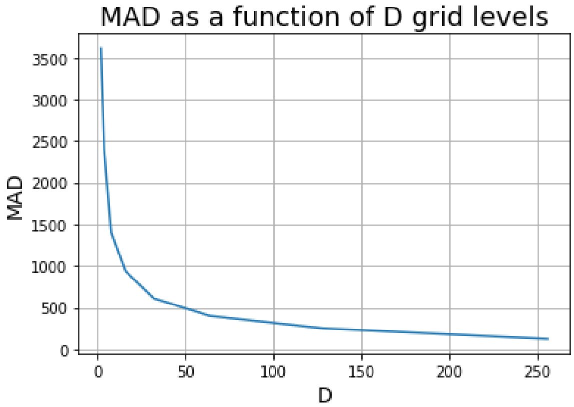
And the MSE as a function of the integer sub-sampling factor D:



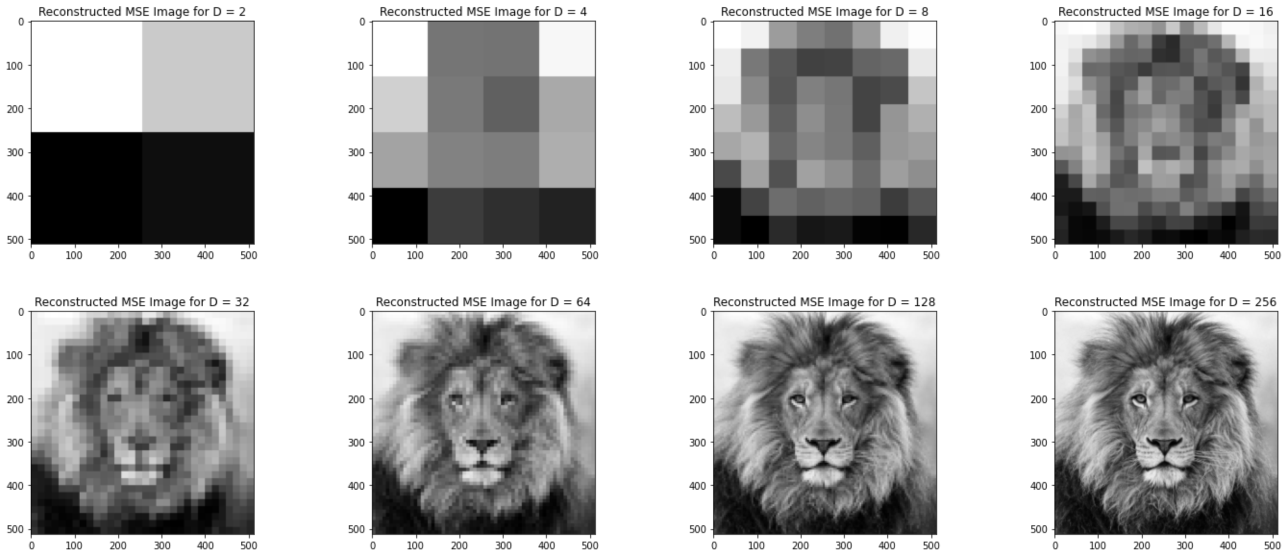
b. The sub-sampled images for $D \in \{2^1, .., 2^8\}$ in the MAD sense:



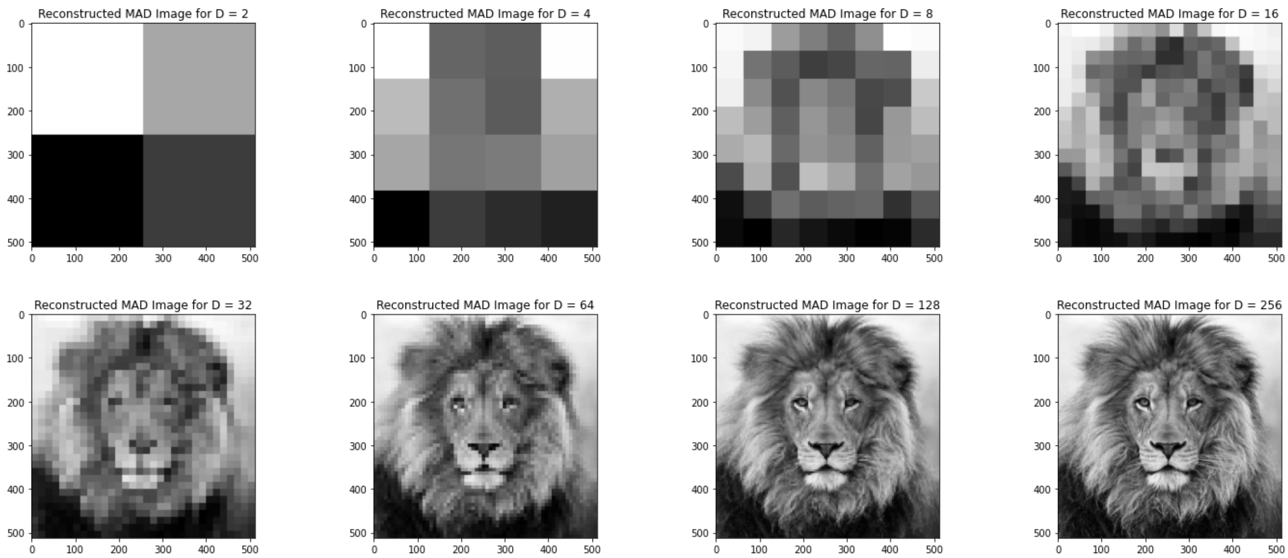
And the MAD as a function of the integer sub-sampling factor D:



2. The reconstructed images from the sub-sampled images, in **MSE** sense:



The reconstructed images from the sub-sampled images, in **MAD** sense:



3. As we predicted, as the factor D increases the reconstructed images get smoother. This happens because $\left(\frac{shape}{D}\right)^2$ values are represented by a single value – so for a larger D we represent smaller groups of pixels using a single value, in contrast to a small D where we represent a large group of pixels using only one value.
This applies both in MSE and MAD sense (meaning in large D cases (smaller squares), both mean and median are good representatives).

Question 3

1.

given the inputs: $f(x,y), \omega(x,y), P, \delta$:

output: $\hat{f}(x,y)$ – the approximate optimal solution.

Algorithm($f(x,y), \omega(x,y), P, \delta$):

For i in $\{1, \dots, N\}$:

For j in $\{1, \dots, N\}$:

Define:

$$f_{i,j}(x,y) = f(x,y) \ \ for\ (x,y) \in \left[\frac{i-1}{N}, \frac{i}{N}\right] x \left[\frac{j-1}{N}, \frac{j}{N}\right] \ else\ 0$$

Init:

$$\hat{f}_{i,j}(x,y) = 0$$

$$while(\left|\varepsilon_{i,j}^{P}(f_{i,j}, \hat{f}_{i,j}) - \varepsilon_{i,j}^{P}(f_{i,j}, \hat{f}_{i,j}^{next})\right| > \delta):$$

$$\omega'_{f_{i,j}, \hat{f}_{i,j}}(x,y) = \min\left\{\frac{1}{\varepsilon}, \left|f_{i,j}(x,y) - \hat{f}_{i,j}(x,y)\right|^{P-2} \omega(x,y)\right\}$$

$$\hat{f}_{i,j}^{next} = \frac{\int_{\frac{i-1}{N}}^{\frac{i}{N}} \int_{\frac{j-1}{N}}^{\frac{j}{N}} f(x,y)\omega'_{f_{i,j}, \hat{f}_{i,j}}(x,y)dxdy}{\int_{\frac{i-1}{N}}^{\frac{i}{N}} \int_{\frac{j-1}{N}}^{\frac{j}{N}} \omega'_{f_{i,j}, \hat{f}_{i,j}}(x,y)dxdy}$$

$$\hat{f}_{i,j}(x,y) = \hat{f}_{i,j}^{next}(x,y)$$

Define:

$$\hat{f}(x,y) = \begin{cases} \hat{f}_{i,j}^{next}(x,y), & (x,y) \in \left[\frac{i-1}{N}, \frac{i}{N}\right] x \left[\frac{j-1}{N}, \frac{j}{N}\right] \\ 0, & otherwise \end{cases}$$

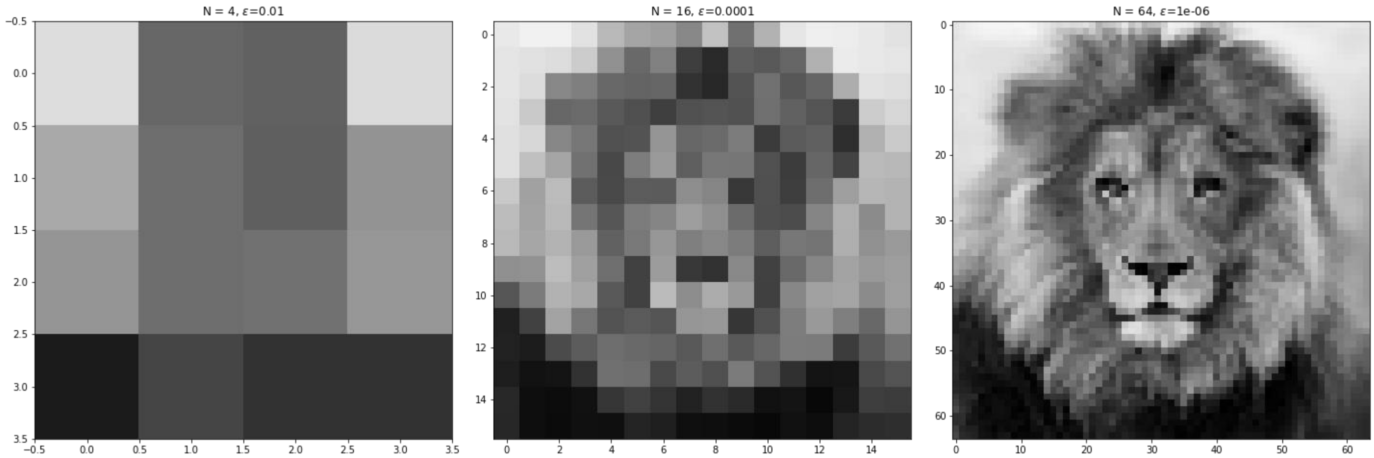
Return $\hat{f}(x,y)$

2. Implemented in the attached notebook

3. Implemented in the attached notebook

4. Here we will compare the results we obtain for the L_1 solution using the two algorithms that we implemented in 3.2 and 3.3 accordingly: the approximate algorithm and the exact algorithm:

First, below are 3 images produced using L_P solver algorithm with $P = 1$ and different values of N and ε:



In addition, we filled a table showing the relations between N and ε, as produced by L_P solver algorithm with $P = 1$ and $\delta = 0.001$:

N\\ε	**0.100000**	**0.010000**	**0.001000**	**0.000100**	**0.000010**	**0.000001**
2	0.202720	0.203921	0.204651	0.205581	0.204521	0.207229
4	0.144331	0.144890	0.145818	0.146045	0.146383	0.146254
8	0.105977	0.106240	0.107303	0.107204	0.107772	0.107317
16	0.084143	0.084748	0.085336	0.085796	0.085710	0.086179
32	0.066696	0.067195	0.067964	0.068538	0.068757	0.068436
64	0.052937	0.053394	0.054173	0.053796	0.053986	0.054194
128	0.041221	0.040975	0.041242	0.041377	0.042217	0.042237
256	0.028590	0.027881	0.027891	0.027917	0.027922	0.027922

According to the table above, as N increases the approximated error decreases (which makes sense because the resolution increases).
Each iteration of the algorithm, we estimate the weight by:

$$\omega'_{f_{i,j},\hat{f}_{i,j}}(x,y) = \min\left\{\frac{1}{\varepsilon}, \left|f_{i,j}(x,y) - \hat{f}_{i,j}(x,y)\right|^{P-2}\omega(x,y)\right\}$$

So for decreasing ε the value of $\frac{1}{\varepsilon}$ is increasing, and therefore ω' is less likely to get the value of $\frac{1}{\varepsilon}$, and would probably get the actual weight value which generally yields more accurate results.

Whereas for smaller N values, the approximated error increases as ε decreases, since the algorithm tries representing a larger square on the grid with one single value, which lowers the accuracy even more.
All together makes the approximation even worse than a random large number $\frac{1}{\varepsilon}$ which therefore affects the next iteration approximation (since it is chosen by as the min value).

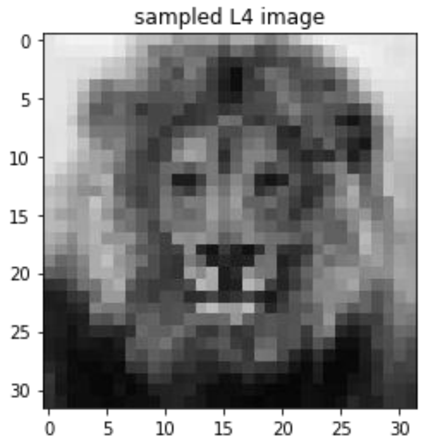
Below is a table showing the L_1 optimal error calculated by the formula given in the lecture and implemented by us, for different N values:

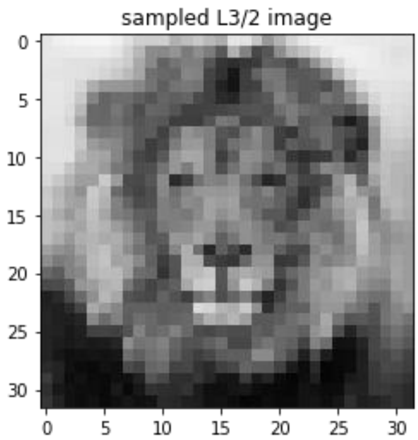
N=2	N=4	N=8	N=16	N=32	N=64	N=128	N=256
0.2024498	0.14362029	0.10527453	0.08375653	0.06620548	0.05247774	0.04043012	0.02781691

Using the table we will compare between the L_1 solver exact algorithm as implemented in Q3.3 and L_P solver algorithm with $P = 1$ as implemented in Q3.2.

As we saw in class, the exact algorithm gives the optimal approximation for L_1, so for every N and ε, we expect the L_P solver error to try to converge to the error of the exact algorithm, but not do better. We can see this behavior in the two tables we present above.

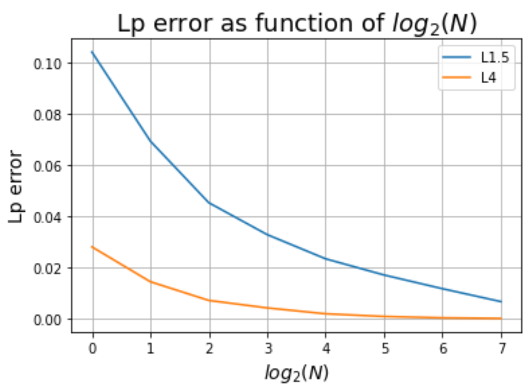
5. below are two sampled images, for the same values of $N, p, \varepsilon, \delta$.





For the left image, produced by L_4 solver, the approximation error is 0.001925373
For the right image, produced by $L_{\frac{3}{2}}$ solver, the approximation error is 0.0233917951

For ε =1e-5 we get the following graph, of both L_4 and $L_{\frac{3}{2}}$ as function of N:



Generally, there is not much to say about whether one p value is better than the other in terms of accuracy, since the scaling of the error is different.

For different values of P-Norm we take a different power (P) so in case of larger P we take a larger power of the error – which lays between 0 to 1, and we force it to be lower.

This behavior can be seen in the graph above:

L_4 error is constantly lower the $L_{\frac{3}{2}}$ error for every N value (and both decrease).