Numerical Optimization
Assignment 4

patrick.gaisberger@student.tugraz.at

January 2023

1 Signal Denoising

1. Analytically determine the optimal solution to find p_k for the FW algorithm given the unit simplex constraint.

$$p_k = \arg\min_p \nabla f(x_k)^T p \qquad s.t \qquad p \in C$$

Lets calculate $\nabla f(x)$

$$f(x) = \frac{1}{2}||Ax-b||_2^2 = \frac{1}{2}\sqrt{\left(\sum_i (Ax-b)_i^2\right)}^2$$

$$f(x) = \frac{1}{2}\sum_i \left(\left(\sum_j a_{i,j}x_j\right) - b_i\right)^2$$

$$\frac{\mathrm{d}f(x)}{\mathrm{d}x_k} = \frac{1}{2}\sum_i \left(\left(\sum_j a_{i,j}x_j\right) - b_i\right)^2 \frac{\mathrm{d}}{\mathrm{d}x} = \frac{1}{2}\sum_i \frac{\mathrm{d}}{\mathrm{d}x}\left(\left(\sum_j a_{i,j}x_j\right) - b_i\right)^2$$

$$= \frac{1}{2}\sum_i 2\left(\left(\sum_j a_{i,j}x_j\right) - b_i\right) a_{i,k} = \sum_i \left(\left(\sum_j a_{i,j}x_j\right) - b_i\right) a_{i,k}$$

$$= \langle A_k, Ax-b\rangle$$

$$\nabla f(x) = \langle A, Ax-b\rangle$$

Where $\langle\cdot,\cdot\rangle$ defines the dot product.

$$p_k = \arg\min_p \nabla f(x_k)^T ps.tp \in C$$

Since $C = \{x \in \mathbb{R}^n : x_i \geq 0, \sum_i = 1^n x_i = 1\}$ = Unit simplex. From the slides

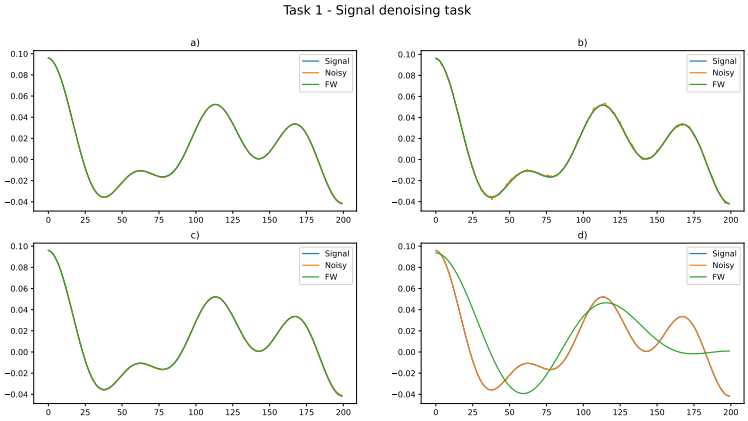
$$p_k = e_{i^*}, i^* \in \arg\min_i (\nabla f(x))_i$$

e_{i^*} is a vector with 1 at position i^* and the rest is 0.

If there are multiple minimal values take an arbitrary minimizing index.

2. Implementation in the python script.
In general I let the FW algorithm run for K=200 iterations, since after that $\tau_k < 0.01$.

3. Discuss each experiment and state how the chosen parameter for d and σ^2 affect the outcome



As we can see in the graphic a) $\sigma^2 = 0.01$ and b) $\sigma^2 = 0.03$ with both $d = 15$, σ^2 does not affect the outcome (a lot). But nonetheless, the higher the noisy (random) part, the worse the outcome. As we can see in c) $d = 100$ an d) $d = 5$ with both $\sigma^2 = 0.01$, that a low d under-fits the signal. This implies also that the higher d is, the higher the frequencies can be which this algorithm can capture. This could lead to over-fitting the noisy signal.

2 Image Compression

1. Determine the number of non-overlapping blocks B when using $n_B = 8$. We have 256 columns and 256 rows, which we chunk into $n_B \times n_B$ blocks. This means we have $\frac{256}{8} = 32$ blocks in a row and colum. This results in $B = 32^2 = 1024$ blocks in total.

2. Analytically determine the optimal solution to find p_k for the FW algorithm given the scaled l_1 constraint.

$$p_k = \arg\min_p \nabla f(x_k)^T ps.tp \in C$$

For $\nabla f(x)$ we can take the solution from TASK.1

$$\nabla f(x) = \langle A, Ax - b \rangle$$

Since C is the scaled l_1 norm. Solution in the lecture slides.

$$p_k \in -t * sgn((\nabla f(x))_{i^*}) * e_{i^*}, i^* \in \arg\max |(\nabla f(x))_i|$$

3. Derive the closed form solution for the problem.

$$\min_x \sum_{s=1}^{B} \frac{1}{2}||Ax_s - b_s||_2^2 + \lambda||x_s||_1$$

First we notice that $(Ax_s)_i$ has to have the same sign as $(b_s)_i$ in order to minimize the formula and since $A^TA = I \rightarrow \text{sgn}(x_s) = \text{sgn}(A^Tb_s)$. Second thing, both terms are convex, hence it is sufficient to set the first derivative to 0 and solve for x_s.

$$\min_x = \sum_{s=1}^{B} \frac{1}{2}||Ax_s - b_s||_2^2 + \lambda||x_s||_1$$

In order to minimize the sum we need to minimize each $s \in 1 \dots B$

First for $x_s \geq 0$

$$\frac{\partial f(x_s)}{\partial x_s} = \frac{1}{2}||Ax_s - b_s||_2^2 + \lambda||x_s||_1 \frac{\partial}{\partial x_s} = \langle A, Ax_s - b_s \rangle + \lambda\text{sgn}(x_s) =$$
$$= A^TAx_s - A^Tb_s + \lambda\text{sgn}(x_s) = x_s - A^Tb_s + \lambda\text{sgn}(x_s)$$
$$0 = x_s - A^Tb_s + \lambda\text{sgn}(x_s) = x_s - A^Tb_s + \lambda\text{sgn}(A^Tb_s)$$
$$x_s = \max(0, |A^Tb_s| - \lambda)\text{sgn}(A^Tb_s)$$

Next for $x_s \leq 0$

$$\frac{\partial f(x_s)}{\partial x_s} = \frac{1}{2}||Ax_s - b_s||_2^2 + \lambda||x_s||_1 \frac{\partial}{\partial x_s} = x_s - A^Tb_s + \lambda\text{sgn}(x_s)$$
$$0 = x_s - A^Tb_s + \lambda\text{sgn}(x_s) = x_s - A^Tb_s + \lambda\text{sgn}(A^Tb_s)$$
$$x_s = \max(0, |A^Tb_s| - \lambda)\text{sgn}(A^Tb_s)$$

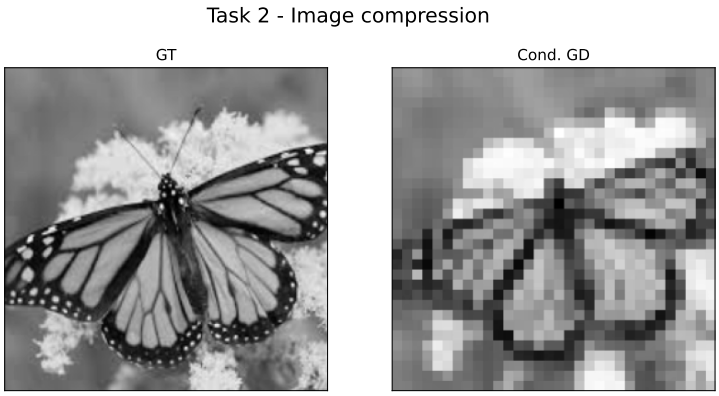
Both settings combined:

$$x_s = \max(0, |A^Tb_s| - \lambda)\text{sgn}(A^Tb_s)$$

The reason for $\max(0, |A^Tb_s| - \lambda)$ is that in the case $\lambda > |A^Tb_s|$ this formula would violate the respective setting ($x_s \geq 0$ and $x_s \leq 0$). Another explanation is that the l_1 norm dominates given $\lambda > |A^Tb_s|$. In this case 0 is a subgradient at point $x_s = 0$.

4. Implementation in the python script.

5. Result of FW for image compression.



In general I let the FW algorithm run for 200 iterations, since after that $\tau_k < 0.01$. Together with the constraint of 0.01 results in a difference per iteration of < 0.0001

6. Result for LASSO