Homework sheet 1 2023-10-20

Due date: 2023-05-09 08:59

The goal of this homework, is to create your first iterative CT reconstruction. Here, we will focus on strictly convex smooth problems (such as least squares).

Delivery format

- Create a folder called “hw02” under the “homework” folder in your git repository.
- Answer all questions to the related to the homework in a file called “README.md” in the above created folder. **The readme must be a complete and standalone solution**. You are highly encouraged to provide instructions on how to recreate your results (e.g. run this script with this dataset, or run through this Jupyter notebook).
- Put all scripts and images you’ve created to reproduce and comprehend your answers in the same folder. Examples for this, are scripts to create plots, visualization, certain reconstructions to compare different strengths and weaknesses and similar things.
- The concrete implementations of the homework, must be put into the “aomip” folder. You are free to choose the structure, layout and form of your implementation.
- Once you’ve finished your homework, commit your changes and create a git tag with `git tag -a "hw02" -m "Tag for homework 2"` and push your changes. This needs to be tagged and pushed to your “main” branch before the deadline ends. Please use this exact command.
- Finally, once you push the changes with the tag to your repository. Create a release in GitLab. You find it in your project, under the menu ‘Deployments‘.

Notes

Generally, we expect and reward both explorativ work and good software engineering. Hence, we highly encourage you to look at the provided resources, do your own research and try different things and play around with the tools provided to you during the course.

Without explicitly permissions you are not allowed to pull any other dependencies. If you need dependencies to load or import a dataset, please get in touch with us and let us know the limitations. If you need to open `TIFF` files, use the already included dependency `tifffile`. Use it with `tifffile.imread`. See some examples at https://pypi.org/project/tifffile/#examples. This can also handle 3D TIFF-files, which makes this quite handy.

Homework 1: Preprocessing again

If you have looked into some of the datasets on the wiki page you will see that most of them are 3D datasets, which some are just huge and hence hard to compute. A nice way is to reconstruct a single slice of the volume. This can be achieved by slicing the projections and extract a lower dimensional sinogram. Basically, you extract a single row of each projection and stack them. Note for Cone-Beam geometry this is not always perfectly valid (without further processing). That depends on the rotation axis and such. But we ignore that for now.

Implement a last preprocessing step called *slicing* and present a sliced sinogram from one of the datasets listed in the Wiki.

Homework 2: Iterative Reconstruction

The first iterative reconstruction algorithm that we implement in this course is: *Gradient Descent* (I know woooow). Gradient Descent solves

$$\min_x f(x)$$

where $f : \mathbb{F}^n \mapsto \mathbb{F}$ is once differentiable. From, here we'll refer to functions which map from a space to the the scalar field *functionals*. Given an initial guess x_0, and a step length $\lambda \in \mathbb{R}_+$, the update rules is given by:

$$x_k = x_{k-1} - \lambda \nabla f(x_{k-1})$$

Implement a routine, which implements the gradient descent. Find a nice small 1D and 2D example you can visualize e.g. using contour plots or similar, that you are sure your algorithms works as expected.

Note: a couple of parameters are usually also chosen for such routines:

- A number of iterations for which you run
- if $x_k - x_{k-1} < \epsilon$ then stop the algorithm

Feel free to implement them.

Homework 3: Solving CT Problems

Now we have an algorithm to solve given some function. In this exercise, we will implement a couple of functionals that are relevant for X-ray CT.

i) Least Squares

The least squares problem is given by:

$$f(x) = \frac{1}{2}||Ax - b||_2^2$$

The derivative is given by

$$\dot{f}(x) = A'Ax - A'b = A'(Ax - b)$$

Implement it such that you can solve a simple reconstruction using the gradient descent algorithm from above and solve an X-ray problem with it.

A tip here, it can be quite beneficial to solve the problem with a known phantom (e.g. the Shepp-Logan phantom). Then you can create error graphs, and usually perform better analysis. However, in the end, the algorithm has to work on real data.

ii) ℓ^2-Norm squared

This is a special case of the least squares problem. We mostly need two different forms, with and without operator, i.e. $||Lx||_2^2$, where L could be the identity. Sometimes it makes sense to differentiate that, sometimes not.

The function is given by:

$$f(x) = \frac{1}{2}||Lx||_2^2$$

and its derivative:

$$\dot{f}(x) = L'Lx$$

The squared ℓ^2-Norm is given with the usual definition:

$$||x||_2^2 = \sum_{i=1}^{n} x_i^2$$

The problem formulation:

$$\frac{1}{2}||Ax - b||_2^2 + \frac{\beta}{2}||x||_2^2$$

is known as Tikhonov problem. With the least squares and ℓ^2 norm implemented, solve the Tikhonov problem where A is the X-ray projector. Compare it to solving without the ℓ^2 regularization. How does it change the reconstruction? What is the prior knowledge we enforce using ℓ^2 regularization?

iii) Huber Functional

The Huber functional is a decent approximation of the ℓ^1 norm, while still being differentiable. The Huber functional is given by

$$L_\delta(x) = \sum_{i=1}^{n} \begin{cases} \frac{1}{2}x_i^2 & \text{if } |x_i| < \delta \\ \frac{1}{2}\delta(|x_i| - \frac{1}{2}\delta) & \text{else} \end{cases}$$

for $\delta > 0$. The derivative is given as:

$$\dot{L}_\delta(x) = y, \quad \text{where } (y_i) = \begin{cases} x_i & \text{if } |x_i| < \delta \\ \delta sign(x_i) & \text{else} \end{cases}$$

Implement the Huber functional, and again apply the gradient descent method to X-ray CT using the Huber functional as regularization instead of the ℓ^2 norm. From looking at the definition, how would you try to choose δ and what are the consequences of bad choices of δ?

iv) Fair potential

As a final functional you should implement the fair potential. It is defined as

$$\psi_\delta(x) = \sum_{i=1}^{n} \delta^2 |\frac{x_i}{\delta}| - \log(1 + |\frac{x_i}{\delta}|)$$

and its derivative:

$$\dot{\psi}_\delta(x) = \frac{x}{1 + \frac{x}{\delta}}$$

And again, replace the ℓ^2 regularization in the Tikhonov problem, with the fair potential and solve the problem using gradient descent. How does it compare to the other two methods? How does the choice of δ influence the reconstruction?

Homework 4: Finite Differences

Operators are a crucial part of the reconstruction process. During the course, we might make use of multiple operators. Here, we will start implementing one important one for X-ray CT.

Finite Differences is the approximation of the derivative, and in the discrete space sometimes the best we can do. For a 1D signal, the finite difference operators is a band limited matrix of the form:

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The exact matrix is determined by some implementations details, such as choosing forward, central or backward differences (you find infos on that in the English Wikipedia article on Finite Differences)

For multi dimensional signals, the Finite Difference Operator δ can be seend as analogous to partial derivatives:

$$\Delta f(x,y) = \begin{bmatrix} \Delta_x f \\ \Delta_y f \end{bmatrix} = \begin{bmatrix} \frac{f(x+1,y)-f(x-1,y)}{2} \\ \frac{f(x,y+1)-f(x,y+1)}{2} \end{bmatrix}$$

Hence, if the signal $f \in \mathbb{F}^{m \times n}$ represents a 2 dimensional signal (it might not be stored like that!), the Finite Difference returns a 3 dimensional signal $\Delta f \in \mathbb{F}^{2 \times m \times n}$.

Implement Finite Differences as described above, use any implementation strategy. You can use zero boundary conditions. Feel free to implement other strategies and see how they impact final results.

The generalized Tikhonov problem is given as

$$\frac{1}{2}||Ax - b||_2^2 + \frac{1}{2}||Lx||_2^2$$

Using $L = \Delta$ what prior knowledge are we enforcing into the problem? Next, solve the problem using gradient descent. Again replace the ℓ^2 norm by the Huber functional and Fair potential. Can you see any difference? How does it differ? In what cases would you prefer which? How does the choice of δ influence the reconstruction now?

Homework 5: Iterative algorithm vs FBP

You have now implemented a couple of different problems using a very basic iterative algorithm. How does it compare to the previously implemented FBP?

Consider a couple of different cases, such as how do they deal with noise (gaussian or salt and pepper noise)? Or with how few poses can you still get away with reasonable reconstruction? How do the runtimes compare?

Finally, make a reconstruction with either the FBP or the iterative algorithm for the challenge dataset and upload your best solution to https://submission.ciip.in.tum.de/.