LSD TP2 Report : Diffusion Ordered SpectroscopY

Imen AYADI

November 2020

Abstract

In this lab, we are interested in reconstructing a diffusion signal x given y its discrete Laplace transformation, denoted y, considered as the noisy signal to denoise..The technique is called **DOSY** (Diffusion Ordered SpectroscopY) and it aims to derive the decomposition of a given molecule. This approach can be reformulated as an optimization problem where different strategies of penalization are studied.

1 Generation of synthetic data

For the code, see `LSD-TP2-Ayadi.ipynb`.
We assume that we know [1]the diffusion signal $\overline{x} \in \mathbb{R}^N$ and we construct a synthetic noisy signal $y \in \mathbb{R}^M$ given as

$$y = K\overline{x} + w$$

where $K \in \mathbb{R}^{M\times N}$ is the discretization matrix and $w\ \mathcal{N}(0, \sigma^2)$ is the noise.
The following plots depict the evolution of the original diffusion signal as function of sampled grid T and the evolution of the noisy signal as a function of the acquisition time t.
In the Figure 1, the diffusion signal $\overline{x}$ has 2 pics (the distribution of $\chi(T)$ is a mixture of 2 Gaussians). Each pic corresponds to a different component of the studied molecule. At fist glance, the observed signal y does not convey directly the idea of "separation". It decreases to 0 during the acquisition. The challenge is then to denoise it to get a signal close to the original one $\overline{x}$ i.e. we hope that the estimated signal contains distinguished pics.

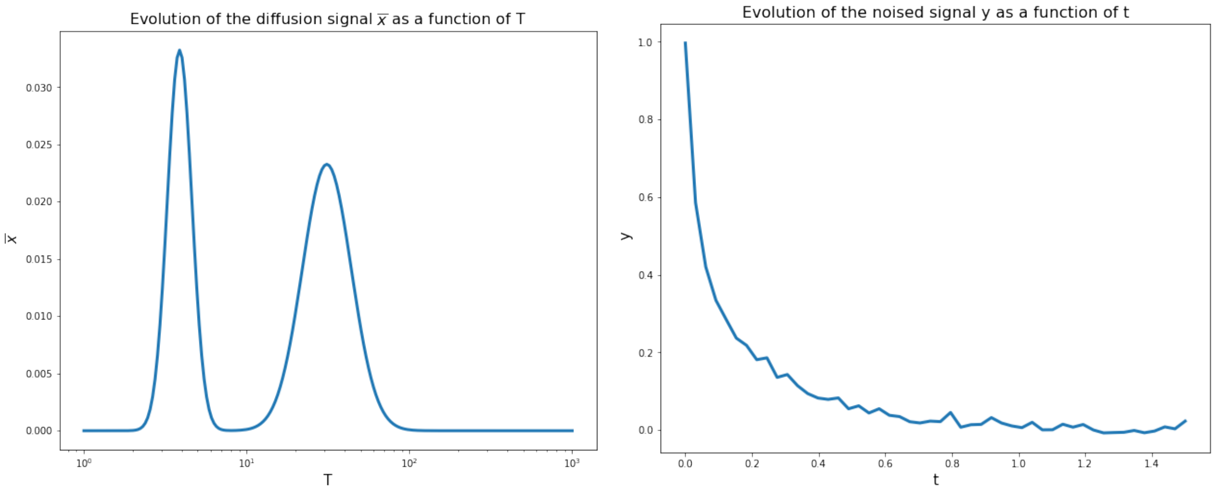

Figure 1: **left:** The real diffusion signal $\overline{x}$ - **right :**The nosiy synthetic signal y

2 Comparison of regularization strategies

The denoising procedure takes a form of an optimization problem, in which the loss function can be penalized in different ways.

$$\boxed{min_{x\in\mathbb{R}^N}\frac{1}{2}||Kx-y||_2^2+\beta g(x)} \tag{1}$$

where $\beta \geq 0$ is the penalty parameter and $g : \mathbb{R}^N \to \mathbb{R}_+ \in \Gamma_0(\mathbb{R}^N)$ is the regularization function.
In this section, we study 3 strategies of penalization and we compare their performance in term of reconstruction quality.

[1] In reality, we do not have access to $\overline{x}$, but, our goal in the lab is to assess the performance of the some penalization strategies. Thus, we need to compare the denoised signal with the real one

2.1 Smoothness prior

$$\forall x \in R^N, \quad g(x) = \frac{1}{2}||Dx||_2^2$$

where $D \in \mathbb{R}^{N\times N}$ is the matrix of discrete gradient operator.

1. Existence and uniqueness of a solution to Problem 1:
   Let $h : \mathbb{R}^N \to \mathbb{R}$ be the function defined as: $h(x) = \frac{1}{2}||Kx - y||_2^2 + \frac{\beta}{2}||Dx||_2^2$. h is differentiable. Its gradient is $\nabla h(x) = K^T(Kx - y) + \beta D^T Dx = (K^T K + \beta D^T D)x - K^T y$.
   We notice that $(K^T K + \beta D^T D)$is definite positive (in particular non-singular). In fact, for $v \in \mathbb{R}^N$,

$$\begin{aligned} v^T(K^T K + \beta D^T D)v = 0 &\iff ||Kv||_2^2 + \beta||Dv||_2^2 = 0 \\ &\iff ||Kv|| = 0 \text{ and } ||Dv|| = 0 \\ &\iff Kv = 0 \text{ and } Dv = 0 \\ &\iff Kv = 0 \text{ and } \exists\alpha \in \mathbb{R} \text{ such that } v = \alpha\mathbf{1_N} \quad (\text{since } \forall 1 \leq n \leq N, [Dv]_n = v_n - v_{n-1}) \\ &\iff \exists\alpha \in \mathbb{R} \text{ such that } v = \alpha\mathbf{1}_N \text{ and } \alpha K\mathbf{1}_N = 0 \\ &\iff v = 0 \quad (\text{because } \forall 1 \leq m \leq M, [K\mathbf{1}_N]_m = \textstyle\sum_{n=1}^N K_{mn} > 0) \end{aligned}$$

   Thus, the unique saddle point of h is $\hat{x} = (K^T K + \beta D^T D)^{-1}K^T y$. Moreover, the hessian of h for each point x is $\nabla^2 h = K^T K + \beta D^T D$, which is definite positive. so, h is strictly convex and therefore, its saddle point is its unique (global) minimizer. To sum up, in case of the smoothness prior, the problem 1 has a unique solution given by:

$$\hat{x} = (K^T K + \beta D^T D)^{-1}K^T y \tag{2}$$

2. Approach to solve the problem 1:
   As we get a closed-form solution with calculus and since the dimension of the problem is not so high, evidently, the simplest way is to write a one-lined code that yields the Equation (2).No specific algorithm is needed. One detail related to Python is that is better to use `numpy.linalg.solve(K.T@K + beta D.T@D,K.T@y )`[ii] rather than to compute the inverse with `numpy.linalg.inv(K.T@K + beta D.T@D)` for a question of computational time.

3. Normalized quadratic error:
   The only parameter to tune is β. I derived $\beta_{optimal}$ (among a range of values from 2^{-10} to 4) that yields the minimal normalized quadratic error. I found (for one execution) $\beta_{optimal} = 2$ and $E(\hat{x}, \overline{x}) = 0.46$. It would be better to run multiple executions (i.e. create several random noise w) and then compute an average of the normalized quadratic error aver all the executions to be able to extrapolate.

2.2 Constrained Smoothness prior

we add a constraint to the former problem :

$$\forall x \in R^N, \quad g(x) = \frac{1}{2}||Dx||_2^2 + \iota_{[x_{min},x_{max}]^N}(x)$$

1. Existence and uniqueness of a solution to Problem 1:
   Since $\mathcal{C} := [x_{min}, x_{max}]^N$ is a closed and bounded subset of $\mathbb{R}^N$, then, it is a compact. The function $h : x \mapsto \frac{1}{2}||Kx - y||_2^2 + \frac{\beta}{2}||Dx||_2^2$ is continuous, then, $h(\mathcal{C})$ is a compact subset of $\mathbb{R}_+$. In particular, there exists at least one minimizer of h over $\mathcal{C}$.
   On the other hand, $\mathcal{C}$ is a non-empty convex subset of $\mathbb{R}^N$ and $h_{|\mathcal{C}}$ is a strictly convex function . Hence, the minimizer of h over $\mathcal{C}$ is unique.

2. Approach to solve the problem 1:
   The difference of this minimization problem with report to the previous unconstrained one is the unsmothness of $\iota_{\mathcal{C}}$. However,the projection onto $\mathcal{C}$, denoted $P_{\mathcal{C}}$, is very easy to compute and to implement. That is why we choose the Projected Gradient algorithm as implementation strategy. First, let us check if the required hypothesis to apply the algorithm hold:
   (a) The function $h : x \mapsto \frac{1}{2}||Kx - y||_2^2 + \frac{\beta}{2}||Dx||_2^2$ is proper, continuous and convex over $\mathbb{R}^N$,then, $h \in \Gamma_0(\mathbb{R}^N)$.

[ii] In Python, `numpy.linalg.solve(A,B)` solves the linear system AX = B and it is computationally more efficient than computing the inverse of the matrix A

(b) h is differentiable and $\forall x_1, x_2 \in \mathbb{R}^N$, we have :

$$||\nabla h(x_1) - \nabla h(x_2)||_2 = ||(K^TK + \beta D^TD)(x_1 - x_2)||_2 \leq |||(K^TK + \beta D^TD)||| \;\; ||x_1 - x_2||_2 \tag{3}$$

where $|||.|||$ is the operator norm or called also spectral norm on $\mathbb{R}^{N\times N}$ defined as: $|||A||| = sup_{||x||\leq 1}||Ax||_2$ for $A \in \mathbb{R}^{N\times N}$.
Since $K^TK + \beta D^TD \in \mathcal{S}_n^+$, $|||(K^TK + \beta D^TD)|||$ is equal to the largest eigenvalue of $K^TK + \beta D^TD$. Let us call it $\nu > 0$. Consequently, ∇h is a ν-Lipschitz function.

- Proof: Let $\lambda_1 \geq ... \geq_N \geq 0$ be the eigenvalues of $K^TK + \beta D^TD$ and let $(e_1, .., e_N)$ the orthonormal basis formed of the eigenvectors of the mentioned eigenvalues. Let $u = (u_1, .., u_N) \in \mathbb{R}^N$. Then,

$$||(K^TK + \beta D^TD)u||_2^2 = ||\sum_{i=1}^N \lambda_i u_i e_i||_2^2 = \sum_{i=1}^N \lambda_i^2 u_i^2 \leq \lambda_1^2 ||u||_2^2$$

  On the other hand, $||(K^TK + \beta D^TD)e_1||_2 = ||\lambda_1 e_1||_2 = \lambda_1$. Thus, $|||K^TK + \beta D^TD||| = \lambda_1$

- Remark: Since K^TK and D^TD commute (verified with `Python`), then, there exists a common orthogonal matrix U that diagonalizes simultaneously K^TK and D^TD and thus, each eigenvalue of $K^TK+\beta D^TD$ is the sum of an eigenvalue of K^TK and an eigenvalue of D^TD multiplied by β. Moreover, each eigenvalue ofK^TK (respectively D^TD) is the square of a singular value of K (respectively D). Thus, instead of computing `max(np.linalg.eigvals(K.T@K+ beta * D.T@D ))` to calculate ν, we can compute it by taking `np.linalg.svds(K)[1][-1]**2+ beta* np.linalg.svds(D)[1][-1]**2` to avoid matrical multiplication.

(c) $\mathcal{C}$ is a non-empty closed convex subset of $\mathbb{R}^N$

(d) $\text{argmin}_{x\in\mathcal{C}}(h(x)) \neq \emptyset$

Thus, we can use the **projected gradient algorithm** with a constant step-size to compute the minimizer of the penalized loss.

```
Algorithm 1: Projected Gradient Algorithm
  Result: $x_n$
  initialization : $x_0 \in \mathbb{R}^N$ , $\gamma \in ]0, \frac{2}{\nu}[$ , $\lambda \in ]0, 2 - \frac{\nu\gamma}{2}[$
  while not stopping criterion do
      $y_n = x_n - \gamma(K^TKx_n - K^Ty + \beta D^TDx_n)$
      $x_{n+1} = x_n + \lambda(P_{\mathcal{C}}(y_n) - x_n)$
```

The projection onto $\mathcal{C} = [x_{min}, x_{max}]^N$ is given by:

$$P_{\mathcal{C}}(x) = \begin{cases} x_{min} \text{ if } x_i < x_{min} \\ x_{max} \text{ if } x_i > x_{max} \\ x_i \text{ otherwise} \end{cases}$$

The convergence rate is $O\left(\frac{1}{n}\right)$ (if we stop at iteration n).
For the choice of the constants, we can choose $\gamma = \frac{1.99}{\nu}$ and $\lambda = 0.99\left(2 - \frac{\gamma\lambda}{2}\right)$. Since in our case we have access to $\overline{x}$, we can "cheat" on the choice of x_0 by taking $\frac{\overline{x}_{min}+\overline{x}_{max}}{2}\mathbb{1}_N$.In practice ($\overline{x}$ is unknown), we choose a random initialization.

- Remark: Apart from having a ν-Lipchitzian gradient, the function h is strongly convex. In fact, $\forall x_1, x_2 \in \mathbb{R}^N$, we have:

$$\langle \nabla_h(x_1) - \nabla_h(x_2), x_1 - x_2 \rangle = \langle (K^TK + \beta D^TD)(x_1 - x_2), (x_1 - x_2) \rangle \geq \alpha ||x_1 - x_2||_2^2$$

  where α is the smallest eigenvalue of $K^TK + \beta D^TD$. Therefore, h is α-strongly convex. Thus, the convergence rate is $O\left(\frac{1}{\sqrt{n}}\right)$.

3. Normalized quadratic error:
As in the unconstrained smoothness prior, we apply a grid search on a large range of β. I found $\beta_{optimal} = 8$, for which the construction error is $E(\hat{x}(\beta_{optimal}), \overline{x}) = 0.76$.

2.3 Sparsity prior

$$\forall x \in R^N, \quad g(x) = ||x||_1$$

1. Existence and uniqueness of a solution to Problem 1:
   Since the norms are equivalent in finite dimension, there exists $c > 0$ such that $\forall x \in \mathbb{R}^N$, $||x||_1 \geq c||x||_2$.
   Thus, $h(x) = \frac{1}{2}||Kx - y||_2^2 + \beta||x||_1 \geq c\beta||x||_2 \xrightarrow[||x||_2 \longrightarrow +\infty]{} +\infty$. Consequently, h is coercive and then, it admits at least one minimizer.
   However, we cannot conclude about the uniqueness of the minimizer.

2. Approach to solve the problem 1:
   We can use the **forward-backward algorithm**. Let us first check the required hypothesis:

   (a) The function $h_1 : x \mapsto \beta||x||_1$ is proper, continuous and convex (because every norm is a convex function). Thus, $h_1 \in \Gamma_0(\mathbb{R}^N)$

   (b) The function $h_2 : x \mapsto \frac{1}{2}||Kx - y||_2^2$ is differentiable such that ∇h_2 is a ν'-Lipschitz function where ν' is the largest eigenvalue of $K^T K$.

   (c) $\text{argmin}(h_1 + h_2) \neq \emptyset$

   Furthermore, we notice that the proximity operator of L^1 norm is easy to compute. In fact,since $||x||_1 = \sum_{i=1}^N |x_i|$, then, for $x \in \mathbb{R}^N$:

$$\begin{aligned} prox_{\gamma\beta||.||_1}(x) &= (prox_{\gamma\beta|.|}(x_i))_{1\leq i\leq N} \\ &= \left(argmin_{t\in\mathbb{R}}|t| + \frac{1}{2\gamma\beta}(t - x_i)^2\right)_{1\leq i\leq N} \end{aligned} \tag{4}$$

   If $t \leq 0$,$|t| + \frac{1}{2\gamma\beta}(t - x_i)^2 = \frac{1}{2\gamma\beta}[t - (x_i + \gamma\beta)]^2 - \frac{\gamma\beta+2x}{2}$: decreases until $t = \gamma\beta + x_i$ then keeps increasing. Since we restrict the domain to negative reals, the minimum of $|t| + \frac{1}{2\gamma\beta}(t - x_i)^2$ over $\mathbb{R}_-$ is reached at $min\{0, x_i + \gamma\beta\}$.
   Similarly, for $t \geq 0$, $|t| + \frac{1}{2\gamma\beta}(t - x_i)^2 = \frac{1}{2\gamma\beta}[t - (x_i - \gamma\beta)]^2 + \frac{2x-\gamma\beta}{2}$: decreases until $t = x_i - \gamma\beta$ then keeps increasing. Then, the minimum of $|t| + \frac{1}{2\gamma\beta}(t - x_i)^2$ over $\mathbb{R}_+$ is reached at $max\{0, x_i - \gamma\beta\}$.
   Thus,$argmin_{t\in\mathbb{R}}|t| + \frac{1}{2\gamma\beta}(t - x_i)^2 = sgn(x_i)max\{0, |x_i| - \gamma\beta\}$.
   Finally, the forward-backward algorithm can be formulated as following:

```
Algorithm 2: Forward-Backward Algorithm
  Result: $x_n$
  initialization : $x_0 \in \mathbb{R}^N$ , $\gamma \in ]0, \frac{2}{\nu}[$ , $\lambda \in ]0, 2 - \frac{\nu\gamma}{2}[$
  while not stopping criterion do
      $y_n = x_n - \gamma K^T(Kx_n - y)$
      $x_{n+1} = x_n + \lambda(prox_{\gamma\beta||.||_1}(y_n) - x_n)$
```

3. Normalized quadratic error:
   Using a grid search, I found $\beta_{optimal} = 0.125$, for which the construction error is $E(\hat{x}(\beta_{optimal}), \overline{x}) = 0.8$.

2.4 Comparison of the penalization terms

To compare the different studied penalisation strategies, we plot all the estimated signals (with the optimal choice $\beta_{optimal}$ in the same Figure 2 with the original signal $\overline{x}$ The shapes of the signals derived from the sparsity prior and the constrained smoothness prior are very similar, but very different from the original signal. Thus, these penalisation strategies are not accurate. The best strategy is the unconstrained smoothness prior since the normalized error is the smallest and the shape of the estimated signal shows two pics of the same altitude.

3 Entropy minimisation

In the context of DOSY NMR data processing, a standard strategy for restoring the target signal is to define it as the solution of the so-called maximum entropy problem :

$$\forall x \in \mathbb{R}^N, \quad g(x) = ent(x)$$

where

$$ent(x) = \sum_{i=1}^N \phi(x_i)$$

with

$$\forall u \in \mathbb{R}, \ \phi(u) = \begin{cases} u\log(u) & \text{if} \quad u > 0 \\ 0 & \text{if} \quad u = 0 \\ +\infty & \text{otherwise} \end{cases}$$

We expect that such regularization scheme favors low-density separation.

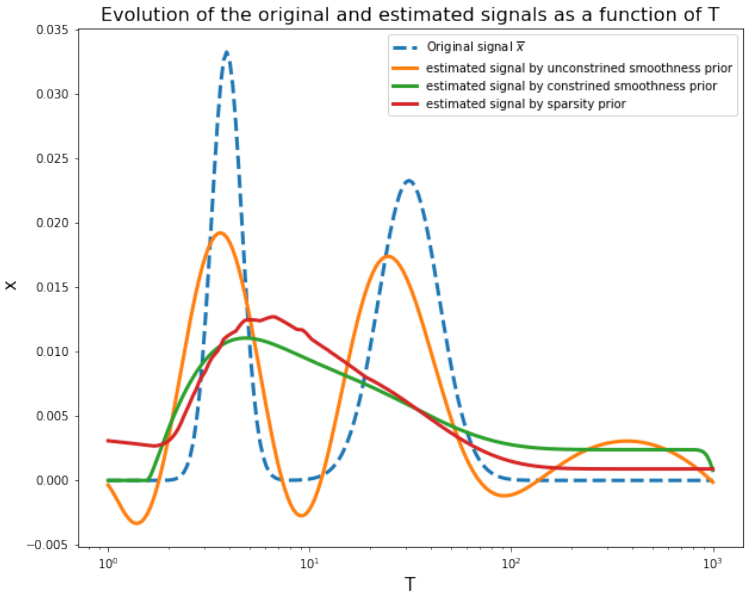

Figure 2: Evolution of the different estimated signals

1. Existence and uniqueness of a solution to Problem 1:
   - The function ϕ is proper, then, ent is proper
   - $\forall u > 0, \phi''(u) = \frac{1}{u} > 0$. Then, ϕ is strictly convex over $\mathbb{R}_+^*$.
   * $\forall t \in]0,1[, \forall u > 0,\ \ \phi(tu + (1-t) \times 0) = \phi(tu) = tu \log(tu) < t \log(u) = t\phi(u) = t\phi(u) + (1-t)\phi(0)$.
   * $\forall t \in [0,1], \forall u \geq 0, \forall v < 0$:

     - if $tu + (1-t)v < 0 :\ \phi(tu + (1-t)v) = +\infty = t\phi(u) + (1-t)\phi(v)$ because $\phi(v) = +\infty$.
     - if $tu + (1-t)v \geq 0 :\ \phi(tu + (1-t)v) < +\infty = t\phi(u) + (1-t)\phi(v)$ because $\phi(v) = +\infty$.

   Therefore, ϕ is strictly convex. Hence, ent is strictly convex as a sum of strictly convex functions.
   - ϕ is continuous over $\mathbb{R}_+^*$. Since, $lim_{u \to 0^+} u \log(u) = 0$. Thus, ϕ is continuous over $\mathbb{R}_+$. Using the fact that $\phi = +\infty$ over $\mathbb{R}_-^*$, it follows that ϕ is s.c.i. Therefore, ent is s.c.i.
   Finally, we conclude that $ent \in \Gamma_0(\mathbb{R}^N)$.
   - $lim_{u \to +\infty} u \log(u) = +\infty$.
   Thus, $lim_{||x||_2 \to +\infty} ent(x) = +\infty$.
   Thus, $h := \frac{1}{2}||K. - y||_2^2 + \beta ent \geq \beta ent$ is coercive.
   Consequently, h has a unique minimizer i.e. there exists a unique solution of 1.

2. Approach to solve the problem 1:

   (a) **Forward-backward algorithm**:
   First, we need to derive the proximity operator of the entropy function.

$$prox_{\gamma ent}(x) = (prox_{\gamma\phi}(x_i))_{1 \leq i \leq N}$$

   where

$$prox_{\gamma\phi}(x_i) = argmin_{t \in \mathbb{R}}\ \frac{1}{2}(t - x_i)^2 + \gamma\phi(t)$$

   Let $\rho : t \in \mathbb{R} \mapsto \frac{1}{2}(t - x_i)^2 + \gamma\phi(t)$.
   ρ is convex, coercive and differentiable over $\mathbb{R}_+^*$.
   * For $t > 0$, $\rho'(t) = t - x_i + \gamma(1 + \log(t)) = (t + \gamma\log(t)) - (x_i - \gamma)$
   Then,

$$\begin{aligned}
\rho'(t) = 0 &\iff t + \gamma\log(t) = x_i - \gamma \\
&\iff \log(t) + \frac{t}{\gamma} = \frac{x_i}{\gamma} - 1 \\
&\iff \frac{t}{\gamma}\exp\left(\frac{t}{\gamma}\right) = \frac{1}{\gamma}\exp\left(\frac{x_i}{\gamma} - 1\right) \\
&\iff t = \gamma\mathcal{W}\left(\frac{1}{\gamma}\exp\left(\frac{x_i}{\gamma} - 1\right)\right)
\end{aligned} \tag{5}$$

where $\mathcal{W}$ is the Lambert function, the inverse of $t \mapsto te^t$.

* For $t = 0$, $\rho(0) = \frac{x_i^2}{2}$

*If t_0 denotes $\gamma \mathcal{W}\left(\frac{1}{\gamma}\exp\left(\frac{x_i}{\gamma} - 1\right)\right)$, then, $x_i = t_0 + \gamma \log(t_0) + \gamma$.

So, $\rho(t_0) = \frac{1}{2}\gamma^2(1+log(t_0))^2 + \gamma t_0 \log(t_0) = \frac{1}{2}\left(\gamma(1+log(t_0))+t\right)^2 - \gamma t_0 + \frac{t_0^2}{2} = \frac{x_i^2}{2} - \gamma t_0 + \frac{t_0^2}{2} = \rho(0) - \gamma t_0 + \frac{t_0^2}{2}$
Therefore, if $-\gamma t_0 + \frac{t^2}{2} < 0$, the minimum is reached at t_0, otherwise it is attained at 0.
Hence, the proximal is not difficult to derive. Now, let us check the requirements of the forward-backward algorithm :

- The function $h_1 : x \mapsto \beta ent(x) \in \Gamma_0(\mathbb{R}^N)$
- The function $h_2 : x \mapsto \frac{1}{2}||Kx - y||_2^2$ is differentiable such that ∇h_2 is a ν'-Lipschitz function where ν' is the largest eigenvalue of $K^T K$.
- $\text{argmin}(h_1 + h_2) \neq \emptyset$

Algorithm 3: Forward-Backward Algorithm

```
Result: $x_n$
initialization : $x_0 \in \mathbb{R}^N$ , $\gamma \in ]0, \frac{2}{\nu}[$ , $\lambda \in ]0, 2 - \frac{\nu\gamma}{2}[$
while not stopping criterion do
    $y_n = x_n - \gamma K^T(Kx_n - y)$
    $x_{n+1} = x_n + \lambda(prox_{\gamma\beta ent}(y_n) - x_n)$
```

(b) **Douglas-Rachford algorithm**:
There are no needed requirements on the smoothness of h_1 and h_2 to apply the Douglas-Rachford algorithm. On the other hand, the proximity operators of both functions are easy to derive. We have computed the proximal of $h_1 = \beta ent$. Moreover, the proximal of $h_2 = \frac{1}{2}||K. - y||_2^2$ is given by:

$$prox_{\frac{\gamma}{2}||K.-y||_2^2}(x) = argmin_{w \in \mathbb{R}^N} \frac{1}{2}||w - x||_2^2 + \frac{\gamma}{2}||Kw - y||_2^2 \tag{6}$$

Let $\rho : w \mapsto \frac{1}{2}||w - x||_2^2 + \frac{\gamma}{2}||Kw - y||_2^2$. This function is strictly convex, coercive and differentiable.
$\rho'(w) = 2(I_N + \gamma K^T K)w - 2(x + K^T y)$.
$I_N + \gamma K^T K$ is invertible. In fact, if $(I_N + \gamma K^T K)u = 0$,then, $u^T(I_N + \gamma K^T K)u = 0$ and it follows that $||u||_2^2 + \gamma||Ku||_2^2 = 0$.Thus, $||u||_2^2 = 0$. Then, $u = 0$.
Therefore, $\rho'(w) = 0 \Longleftrightarrow w = (I_N + \gamma K^T K)^{-1}(x + K^T y)$. Finally, we get:

$$prox_{\frac{\gamma}{2}||K.-y||_2^2}(x) = (I_N + \gamma K^T K)^{-1}(x + K^T y) \tag{7}$$

We can formulate the algorithm as following:

Algorithm 4: Douglas-Rachford Algorithm

```
Result: $z_n$
initialization : $x_0 \in \mathbb{R}^N$ , $\gamma > 0$ , $\lambda \in ]0, 2[$
while not stopping criterion do
    $y_n = x_n - prox_{\frac{\gamma}{2}||K.-y||_2^2}(x_n)$
    $z_n = prox_{\gamma ent}(2y_n - x_n)$
    $x_{n+1} = x_n + \lambda(z_n - y_n)$
```

We notice that this algorithm is not very sensitive to the initialisation, which is a major advantage. However, one of its drawbacks is its non-monotonicity (i.e we have no guarantee that $f(z_n)$ decreases along the iterations). This results in fluctuations on $z_{n+1} - z_n$ and hence, we need a larger tolerance on the stopping criterion in order to make the algorithm faster.

For a first comparison between the two algorithms, we implemented the forward-backward algorithm and the Douglas-Rachford algorithm for the same value $\beta = 10^{-2}$. The results are shown in the following table :

Algorithm	Normalized Error	Entropy
Forward-Backward	0.762	$+\infty$
Douglas-Rachford	8.343	-16.696

Obviously, the F-B yields the smallest reconstruction error but the the largest entropy.
For a more rigorous comparison, we search for each algorithm the optimal value of β then we compare the two estimated signals. These signals are plotted in Figure 4: The following table sums up the normalized errors and the entropies of the two estimated signals for optimal choice of β :

Algorithm	Normalized Error	Entropy
Forward-Backward	0.730	-5.541
Douglas-Rachford	0.459	-3.251

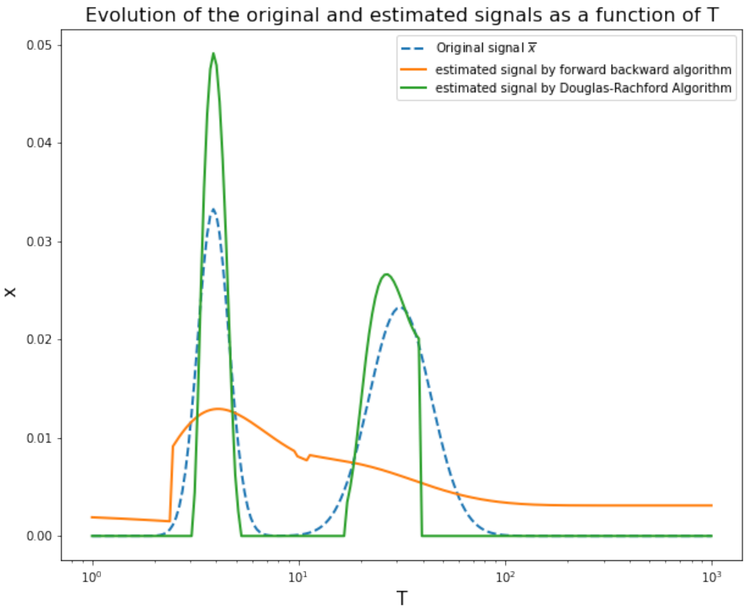

Figure 3: Evolution of the estimated signals by FB and DR algorithms for $\beta_{optimal}$

The shape of the Douglas-Rachford estimated signal is the most similar to the shape of the original signal : it manage to separate the two pics corresponding to different particles.

New optimization problem:
In practice, adjusting the parameter β may be difficult, while one has often information about the level of noise corrupting the data. A more practical formulation may therefore be obtained by solving the following optimization problem :

$$\boxed{\text{minimize } ent(x) \quad \text{subject to } \ ||Kx-y||_2^2 \leq \eta M \sigma^2} \tag{8}$$

where $\eta > 0$.
A good candidate to get a numerical solution of the new problem is Parallel form of Douglas-Rachford. In fact, we want to minimize $x \in \mathbb{R}^N \mapsto ent(x) + \iota_{\overline{\mathcal{B}}(y,\sqrt{\eta M}\sigma)}(Kx)$.
We use the notations of the slide 37/38 of the fixed point course, where $h(x) = h_1(L_1 x) + h_2(L_2 x)$ with $h_1 = ent \in \Gamma_0(\mathbb{R}^N)$, $L_1 = I_N$, $h_2 = \iota_{\overline{\mathcal{B}}(y,\sqrt{\eta M}\sigma)} \in \Gamma_0(\mathbb{R}^N)$ and $L_2 = K$. We have that $ran\left(\begin{bmatrix} I_N & 0 \\ 0 & K \end{bmatrix}\right) = \mathbb{R}^{2N}$ is closed and $v = (v_1, v_2) \in \mathbb{R}^N \times \mathbb{R}^N \mapsto v_1 + K^T K v_2$ is an isomorphism since K is invertible.
Moreover, $argmin(h) \neq \emptyset$. In fact, $\mathcal{C} := \{x \in \mathbb{R}^N | \ ||Kx-y||_2^2 \leq \eta M \sigma^2\}$ is a closed and bounded subset of $\mathbb{R}^N$ as the image of a closed bounded subset (which is $\overline{\mathcal{B}}(y, \sqrt{\eta M}\sigma)$) by a linear bounded function $x \mapsto Kx$, then, $\mathcal{C}$ is a compact. The function $h_1 : x \mapsto ent(x)$ is continuous, then, $h_1(\overline{\mathcal{B}}(y, \sqrt{\eta M}\sigma))$ is a compact subset of $\mathbb{R}_+$. In particular, there exists at least one minimizer of h_1 over $\overline{\mathcal{B}}(y, \sqrt{\eta M}\sigma)$.
We have already derived the proximal of ent and the proximal of $\iota_{\overline{\mathcal{B}}(w,r)}$ is the orthogonal projection onto $\overline{\mathcal{B}}(w,r)$:

$$P_{\overline{\mathcal{B}}(w,r)}(x) = argmin_{z \in \overline{\mathcal{B}}(w,r)} ||z-x||_2 \tag{9}$$

If $x \in \overline{\mathcal{B}}(w,r)$, then, $P_{\overline{\mathcal{B}}(w,r)}(x) = x$.
Now, let us assume that $x \notin \overline{\mathcal{B}}(w,r)$ i.e $||x-w||_2 > r$.
Let $v \in \mathbb{R}^N$ such that $(v-w)^T(z-w) = 0$ and $||v-w||_2 = 1$.
Let $z \in \overline{\mathcal{B}}(w,r)$. There exists a, b such that $z - w = a\frac{x-w}{||x-w||_2} + b(v-w)$. Since $||z-w||_2 \leq r$, then, $a^2 + b^2 \leq r^2$

$$\begin{aligned} ||z-x||_2^2 &= ||(z-w)+(w-x)||_2^2 \\ &= ||(a - ||x-w||_2)\frac{x-w}{||x-w||_2} + b(v-w)||_2 \\ &= (||x-w||_2 - a)^2 + b^2 \qquad \text{(because } x-w \text{ and } v-w \text{ are orthogonal)} \\ &\geq (||x-w||_2 - a)^2 \\ &\geq (||x-w||_2 - r)^2 \qquad \text{(since } ||x-w||_2 > r \geq |a|) \end{aligned} \tag{10}$$

We notice that $t := w + \frac{r}{||x-w||_2}(x-w) \in \overline{\mathcal{B}}(w,r)$ and $||t-x||_2 = ||\left(1 - \frac{r}{||x-w||_2}\right)(w-x)||_2 = ||x-w||_2 - r$.
Therefore, $argmin_z ||z-x||_2 = t$.

Consequently, we have:

$$P_{\overline{\mathcal{B}}(w,r)}(x) = \begin{cases} x & \text{if } \ ||x-w||_2 \leq r \\ w + \frac{r}{||x-w||_2}(x-w) & \text{otherwise} \end{cases}$$

By combining the two expression, we can write $P_{\overline{\mathcal{B}}(w,r)}(x) = w + \frac{r}{max\{r, ||x-w||_2\}}(x-w)$.

Thus, the PPXA+ could be written as :

```
Algorithm 5: PPXA+
  Result: $v_n$
  initialization : $x_0^{(1)} \in \mathbb{R}^N$ ,$x_0^{(2)} \in \mathbb{R}^N$, $\gamma > 0$ , $\lambda \in ]0, 2[$
  $v_0 = (I_N + K^T K)^{-1}(x_0^{(1)} + K^T x_0^{(2)})$
  while not stopping criterion do
      $y_n^{(1)} = prox_{\gamma ent}(x_n^{(1)})$
      $y_n^{(2)} = P_{\overline{\mathcal{B}}(y, \sqrt{\eta M}\sigma)}(x_n^{(2)})$
      $c_n = (I_N + K^T K)^{-1}(y_n^{(1)} + K^T y_n^{(2)})$
      $x_{n+1}^{(1)} = x_n^{(1)} + \lambda(2c_n - v_n - y_n^{(1)})$
      $x_{n+1}^{(2)} = x_n^{(2)} + \lambda(K(2c_n - v_n) - y_n^{(2)})$
      $v_{n+1} = v_n + \lambda(c_n - v_n)$
```

I tried different admissible values for λ and γ and I found that for $\gamma = 1$ and$\lambda = 1.8$, we manage to get an estimated signal that illustrates the separation. Then, by a grid search, I searched the optimal value of η that minimizes the reconstruction error. I plotted the corresponded signal as a function of T in Figure 4:

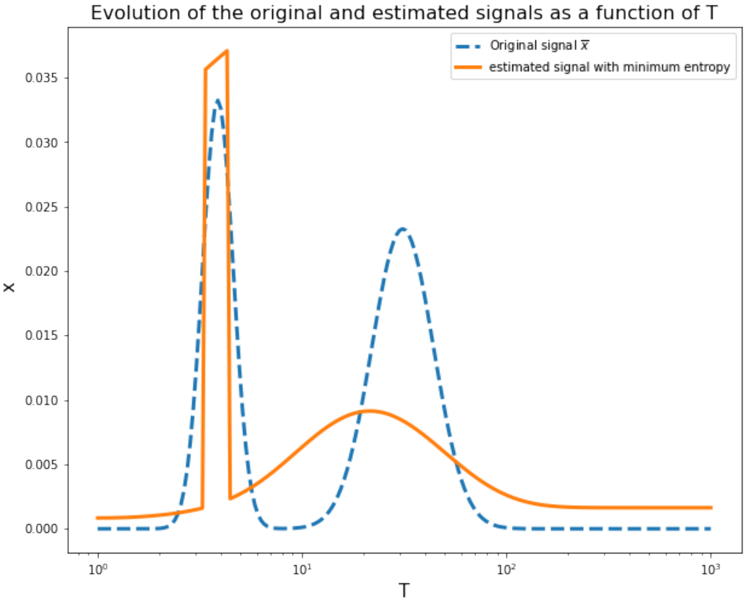

Figure 4: Evolution of the estimated signal by the PPXA+

4 Conclusion

In the first part of this lab, we studied different regularization strategies: unconstrained smoothness prior, constrained smoothness prior, sparsity prior and entropy prior. Experiments showed that the estimated signal by the Douglas Rachford algorithm in the case of the entropy prior is the closest to the original signal. Then, we studied another optimisation problem based on minimizing only the entropy without taking into account the L^2 error. In fact, entropy minimization has been widely used in unsupervised domain adaptation (UDA) and seems to be very suited with separating two pics.
However, the experiments lack sometimes some tests on the constants of the different algorithm to tune them correctly.