LIÈGE université

FACULTÉ DES SCIENCES APPLIQUÉES
Ecole d'ingénieurs et d'informaticiens

MATH0461-2
Introduction to Numerical Optimization

Project: Compressed Sensing

François Lievens

Master 1 Data science
Faculty of Applied Sciences

Academic years 2020-2021

Preliminary note: All files and results used in this project are available at the following GitHub repository: https://github.com/francoislievens/MATH0461-2-Numerical-Optimization-Project-2021

1 Modelling

In this report, we have decided to keep the same nomenclature as provided by the project statement. So, we have:

1. The reference signal $r \in \mathbb{R}^N$ that we wish to reconstruct by the approximation $\hat{r} \in \mathbb{R}^N$.
2. The measurement matrix $\Phi \in \mathbb{R}^{M \times N}$
3. The measurement signal $m \in \mathbb{R}^M$ with $M < N$. So, this signal represent a projection of the reference signal in the measurement matrix Φ in order to obtain a signal of size M where each values are the product of the associated row of Φ with r. So: $m = \Phi r$
4. The basis matrix $\Psi \in \mathbb{R}^{N \times N}$ who form a basis in which r is assumed to have a sparse representation.
5. And $x \in \mathbb{R}^N$ a sparse representation of r such that $r = \Psi x$

Our problem will therefore consist in finding a signal $\hat{r}$, such that its projection into the basis Ψ, $(\hat{x})$ is as sparse as possible, but also such that the sparse signal that we optimize, projected into the measurement matrix Φ, gives the vector m.

So, the objective function that we will optimize have catch the sparsity of the vector $\hat{X}$. This can be done by minimizing the norm of this vector.

The constraints of our program will catch the fact that the signal that we are optimizing $(\hat{x})$ in non-sparse version $(\hat{r} = \Psi x)$ projected into the measurement matrix, must be equal to the signal m.

The optimization problem can so be formulated as follow:

$$\begin{aligned} \min \quad & ||\hat{x}|| \\ \text{s. t.} \quad & m_i - (\Phi\Psi\hat{x})_i = 0 \quad \forall i \in [0, M] \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

At the end of the day, we can so recover the original image by reshaping the vector $\hat{r} = \Phi\Psi\hat{x}$ in 78x78 values.

The choice of the type of norm used as the objective function is important to keep the problem solvable, and also to obtain a good reconstruction of the initial signal. This choice will be discussed in the following sections of this report.

1.1 l_0-Norm

This standard may seem to be the most appropriate since it counts the number of non-zero entries in our vector x. The problem can be written as follow:

$$\begin{aligned} \min \quad & ||\hat{x}||_0 \\ \text{s. t.} \quad & m_i - (\Phi\Psi\hat{x})_i = 0 \quad \forall i \in [0, M] \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

But in fact, this l_0 norm is not really a norm, since it doesn't respect the triangle inequality property $(f(x + y) \leq f(x) + f(y))$. Moreover, l_0-norm is not convex since a function $f : \mathbb{R}^n -> \mathbb{R}$ is convex if, for all x, $y \in \mathbb{R}^n$ and $\lambda \in [0, 1]$, we have:

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

But if we take a value of λ of 0.5 and the two points $x = (1, 0)$ and $y = (0, 1)$, we obtain:

$$f(0.5(1, 0) + 0.5(0, 1)) \leq 0.5f(1, 0) + 0.5f(0, 1)$$

$$f(0.5, 0.5) \leq 0.5f(1,0) + 0.5f(0,1)$$
$$2 \nleq 1$$

Since algorithms that we want to use need a convex objective function to effectively converge towards a global solution, the l_0-norm can't be used to solve our problem.

1.2 l_1-Norm

The l_1-norm is expressed as the sum of absolute values of the given vector. So, we can express our problem as:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} |\hat{x}_i| \\ \text{s. t.} \quad & m_j - (\Phi\Psi\hat{x})_j = 0 \quad \forall j \in [0, M] \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

Despite the fact that this program is not linear, we have seen during the lectures that we can easily rewrite the problem of minimizing the nonnegative sum of absolute values into a equivalent linear program. So, our program becomes:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} t_i \\ \text{s. t.} \quad & m_j - (\Phi\Psi\hat{x})_j = 0 \qquad\qquad \forall j \in [0, M] \\ & -t_i \leq \hat{x}_i \leq t_i \forall i \in [1, N] \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

And this program is linear.

1.3 l_2-Norm

The l_2-norm can be interpreated as the euclidean distance between our vector and the point at the origin. So, we can express our problem as follow:

$$\begin{aligned} \min \quad & \sqrt{\sum_{i=1}^{N} x_i^2} \\ \text{s. t.} \quad & m_j - (\Phi\Psi\hat{x})_j = 0 \quad \forall j \in [0, M] \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

Once again, we have an objective function who is not linear. But by using the epigraph trick, we can reformulate the problem in the following form:

$$\begin{aligned} \min \quad & t \\ \text{s. t.} \quad & m_j - (\Phi\Psi\hat{x})_j = 0 \quad \forall j \in [0, M] \\ & t \geq \sqrt{\sum_{i=1}^{N} \hat{x}_i^2} \qquad\qquad \hat{x} \in \mathbb{R}^N \end{aligned}$$

Since a second order cone is define by:

$$\mathbb{L}^n = \{x \in \mathbb{R}^{n+1} | x_1^2 + ... + x_n^2 \leq x_0^2, x_0 \geq 0\}$$

We can so view our second constraint as the following second order cone:

$$\mathbb{Q}^N = \{(x_1, ..., x_N, t) \in \mathbb{R}^{N+1} | x_1^2 + ... + x_N^2 \leq t^2, t \geq 0\}$$

And we can write the program as:

$$\begin{aligned} \min \quad & t \\ \text{s. t.} \quad & m_j - (\Phi\Psi\hat{x})_j = 0 \qquad \forall j \in [0, M] \\ & (t, x_1, ..., x_N) \in \mathbb{Q}^N \qquad\qquad\qquad \hat{x} \in \mathbb{R}^N \end{aligned}$$

1.4 Closed-form solution for l_2-Norm problem

To provide this expression, we can inject constraints in the objective function by using the Lagrangian method. This consist to penalize the value of the initial objective function by adding a linear combination of the linear constraints, obtained by using the vector of weights λ. The Lagrangian of our problem is so:

$$\mathcal{L}(\hat{x}, \lambda) = ||\hat{x}||_2^2 + \lambda^T(\Phi\Psi\hat{x} - m)$$

Since the square function is monotone increasing, we have, in the previous expression replaced the l_2-norm par his square, who don't change the solution of our convex optimization problem.

To obtain the solution of this problem, we have to identify the values of the vector $\hat{x}$ for which, the partial derivatives of the Lagrangian with respect to values of $\hat{x}$ are equal to zero. We have so:

$$\frac{\partial(||\hat{x}||_2^2 + \lambda^T(\Phi\Psi\hat{x} - m))}{\partial\hat{x}} = 0$$

Who give:

$$2||x||_2 \frac{\hat{\hat{x}}^T}{||\hat{x}||_2} + \lambda^T\Phi\Psi = 0$$

$$\hat{x} = \frac{1}{2}(\Phi\Psi)^T\lambda$$

Since we know that $\Phi\Psi\hat{x} = m$, we can get the value of λ by the following way:

$$\Phi\Psi\hat{x} = -\frac{1}{2}(\Phi\Psi)(\Phi\Psi)^T\lambda = m$$

And so:

$$\lambda = -2((\Phi\Psi)(\Phi\Psi)^T)^{-1}m$$

1.5 First Robust variant of l_1-Norm

The first variant is very intuitive: instead of keeping strict equality equations in our constraints, we want to replace it by an inequality and a parameter ϵ. In other words, we are not longer looking for a vector $\hat{x}$, who projected in our basis and our measurement matrix give exactly the vector m, but we keep a certain liberty ϵ to our optimizer. This liberty let us to deal with the uncertainty of our data. We can so written our problem as following:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} t_j \\ \text{s. t.} \quad & -\epsilon_i \leq |\frac{m_i - (\Phi\Psi\hat{x})_i}{m_i}| \leq \epsilon_i \quad \forall i \in [0, M] \\ & -t_j \leq \hat{x}_j \leq t_j \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

Who is a linear program who let the optimizer finding an approximation $\hat{x}$ who provide a signal $\hat{m}$, close to the vector m according to a freedom. This freedom is represented by a certain proportion of the values of m given by the parameter ϵ.

1.6 Second Robust variant of l_1-Norm

This time, we allow the optimizer to find an approximation $\hat{x}$ who provide a signal $\hat{m}$ who have an Euclidean distance distance from the vector m less or equal to the parameter ϵ.

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} t_j \\ \text{s. t.} \quad & ||m - \Phi\Psi\hat{x}|| \leq \epsilon \\ & -t_j \leq \hat{x}_j \leq t_j \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

So, we have already see in the part about the l_2-Norm that this expression can be view as a second order cone, and we can written:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} t_j \\ \text{s. t.} \quad & (\epsilon, (\Psi\Phi\hat{x})) \in \mathbb{Q}^M \\ & -t_j \leq \hat{x}_j \leq t_j \\ & \hat{x} \in \mathbb{R}^N \end{aligned}$$

2 Numerical Experiments

2.1 Code

Our implementation of the l_1-Norm and L_2-Norm using Julia JuMP can be find in the files *l1.jl* and *l2.jl* respectively. For the two robust implementation of l_1-Norm, we have respectively the files *l1_rob_A.jl* and *l1_rob_B.jl* and the versions *l1_rob_A_loop.jl* and *l1_rob_B_loop.jl* who perform the experiments for both files.
During each optimization process, the optimized sparse vector $\hat{x}$ is exported as a *.csv* that can be used by the python script of the file *histo.py* in order to generate the associated histograms of values contained in the vector.

2.2 l_1 and l_2-Norm problems for uncorrupted measurements

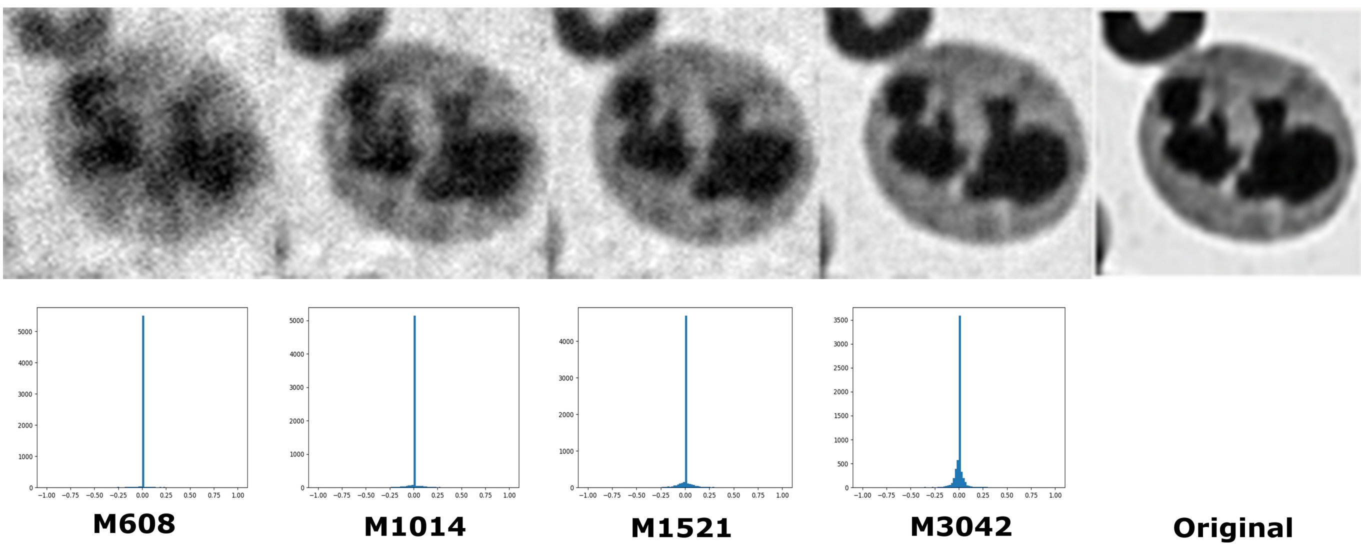

FIGURE 1: Reconstructed images using l_1-norm

In the figure 1, we can observe reconstructed images obtain by using the l_1-norm as objective function. As we could expect, the quality of the reconstruction improve with the size of the input signal m, and so with the quantity of information contained in the measurement vector m.

Globally, we can see that solving the problem by using this norm makes good performances and we obtain a with $M3042$ a cell image relatively close to the original image. The optimization process converge in average in 15 iterations, but the execution time increase a lot with the size of m to reach an average of 300s on our reference computer
In the bottom part of the figure 1 are show the histogram of values of $\hat{x}$. We can see that the strict majority of the values we get are close to zero, which reflects the high sparsity of the vector. With the increasing of the size of m, we progressively loss a part of this sparsity. This looks logical to us since the vector contain more information and provide a clearer image with more details.

For the l_2-norm optimization problem, results are show in the figure 2.

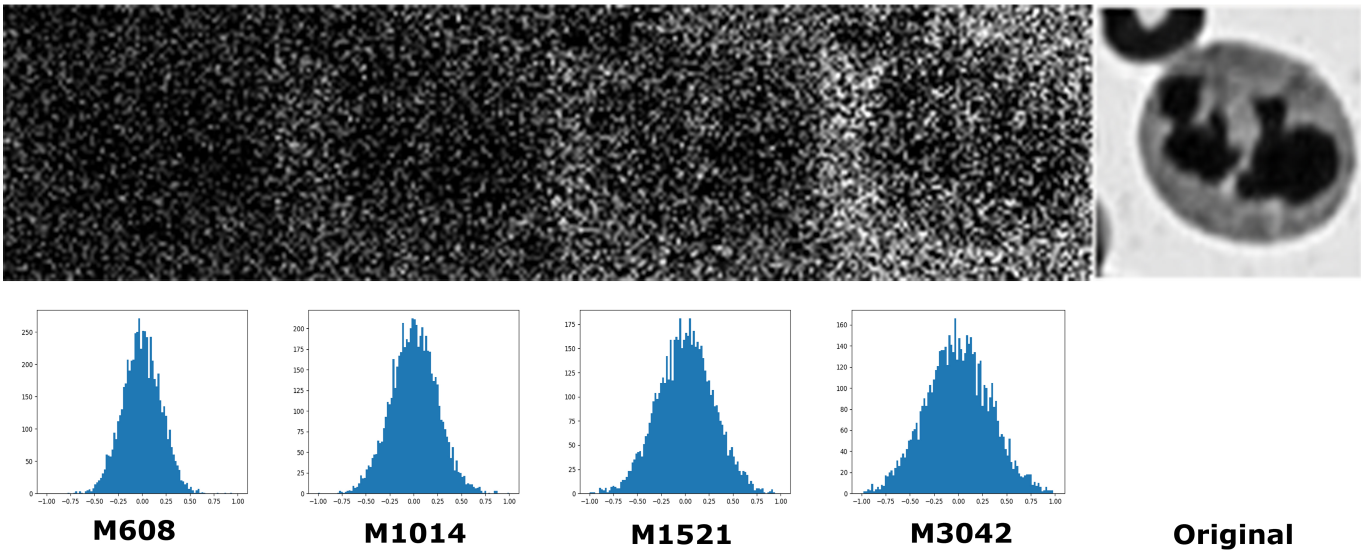

FIGURE 2: Reconstructed images using l_2-norm

Despite an optimization process who converge in only 5 iterations in average and who is much faster, we obtain really poor results, even for $M3042$. Associated histograms show that the optimizer don't find a so sparse representation of $\hat{x}$, with values who are more normally distributed over the whole spectre.
This process provide us really noisy images in which we can only vaguely distinguish the general shapes of the cell.

More generally, we can conclude that the l_1-norm is really more adapted to optimize the sparsity of a vector. We can probably associate the poor results of the l_2-norm optimization problem with the properties of the l_2-norm who give a too large attention to entries with large values and less for entries close to zero by penalizing quadratically values in the vector. This process provide a vector with a lot of value close to zero, but not really at zero, and so not really sparse.

2.3 Robust l_1-norm robust formulations over noisy measurements

Cell's images obtained by using our first robust implementation of l_1-norm, and corresponding histograms of $\hat{x}$ values are show in figures 3 and 4 respectively.
As we ca, see, the increasing of the value of ϵ progressively reduce the noise and give and an image that tends more and more towards the white. This increasing of ϵ have as effect a decreasing of the noise in the reconstructed image, but this denoising also loss some details in the image, especially for clear pixels.
If we are looking at corresponding histograms of $\hat{x}$, who are show using a logarithmic scale in the figure 4, we can see that the increasing of ϵ have almost no effect on it.

Concerning the second robust variant, reconstructed images and histograms are respectively show in figures 5 and 6.

This time, the relationship between the value of ϵ and the sparsity of $\hat{x}$ seems to be more evident. The approximed vector begin more and more sparse with the increasing of ϵ. This look us to be logical in the sense that increasing epsilon let more freedom to the optimizer to find a more sparse representation. So, the increasing of ϵ tends toward a vector who only contain zeros, and so a fully black pixels reconstructed image. It's interesting to notice the fact that this effect is more and more pronounced with the increasing of the size of the measurement vector.

In the point of view of the reconstructed image, the increasing of the ϵ value gives more and more blurry images, which can reflect the loss of information due to too sparse vectors $\hat{x}$.

2.4 Interpretation of dual variables for equality constraints for the l_1-norm formulation

Since the dual objective function represent a linear combination of values of the measurement vector m, weighted by a multiplication factor p_i for each lines of $(\Phi\Psi\hat{x})$, we can view the vector p as the weights given to each approximated values of m. So, this vector can be consider as the importance of each constraints, and so each approximated value of m over the value of the objective function (and so on the norm of the vector $\hat{x}$. This can be used to evaluate the impact of small changes in the measurement signal can affect the reconstructed image, and so the impact of the noise in the measurement signal.

Appendices

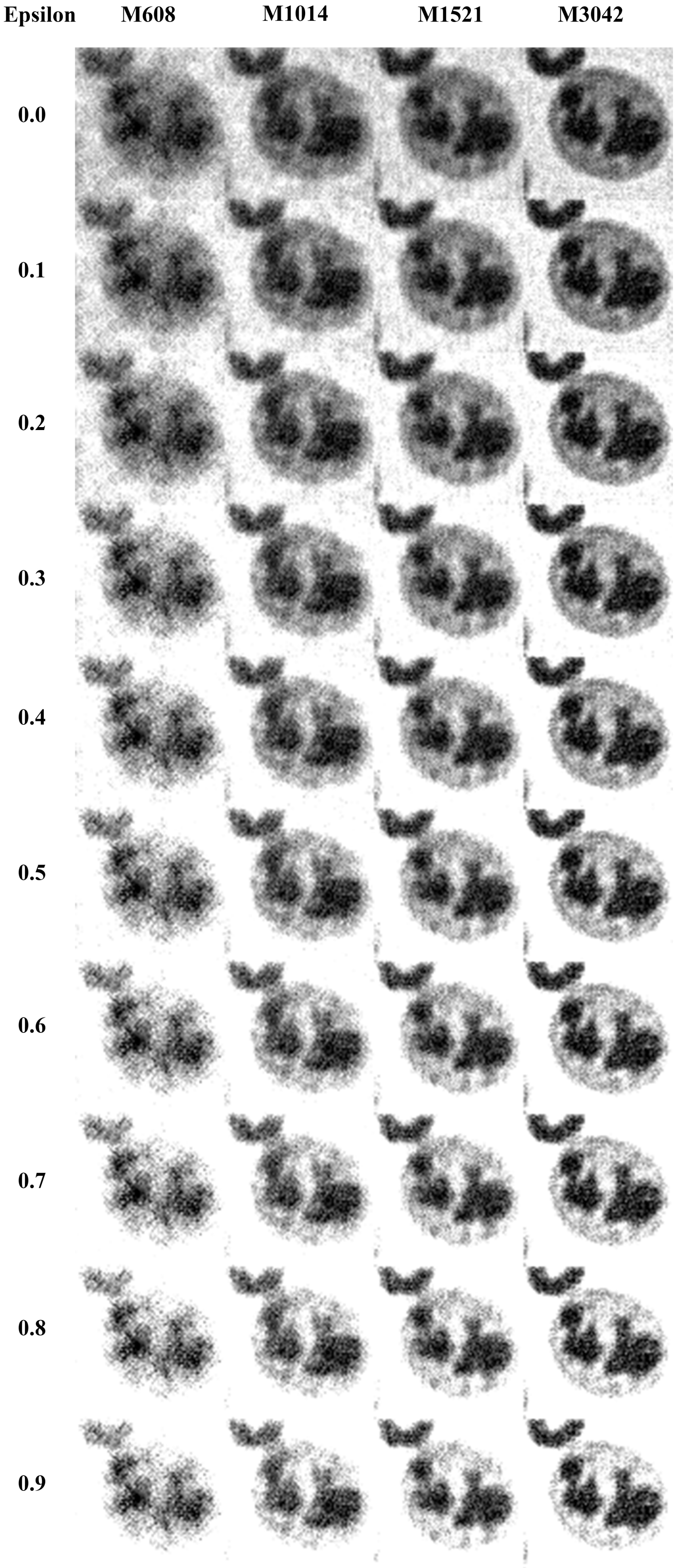

FIGURE 3: l_1-norm reconstitution, using the first robust program over noisy measurements

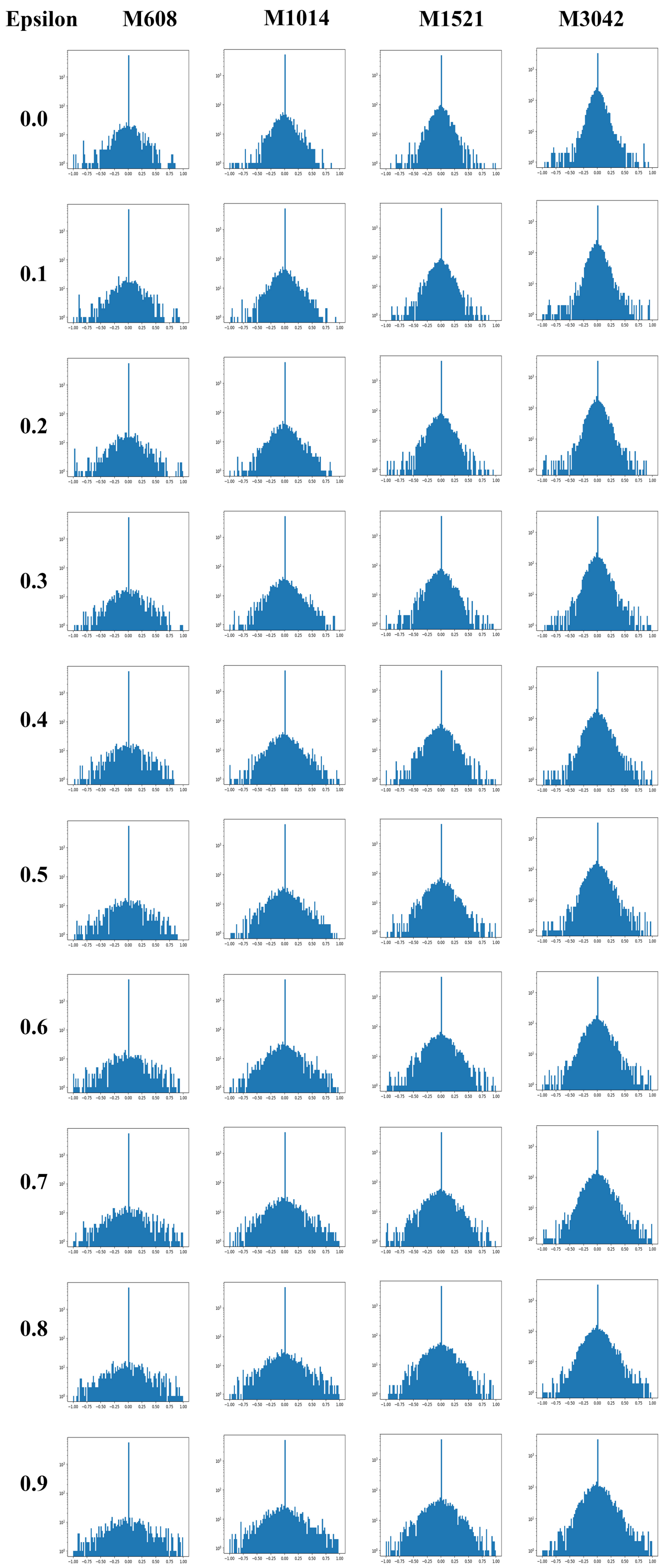

FIGURE 4: l_1-norm reconstitution, using the first robust program over noisy measurements. Histograms of $\hat{x}$ values in logarithmic scale

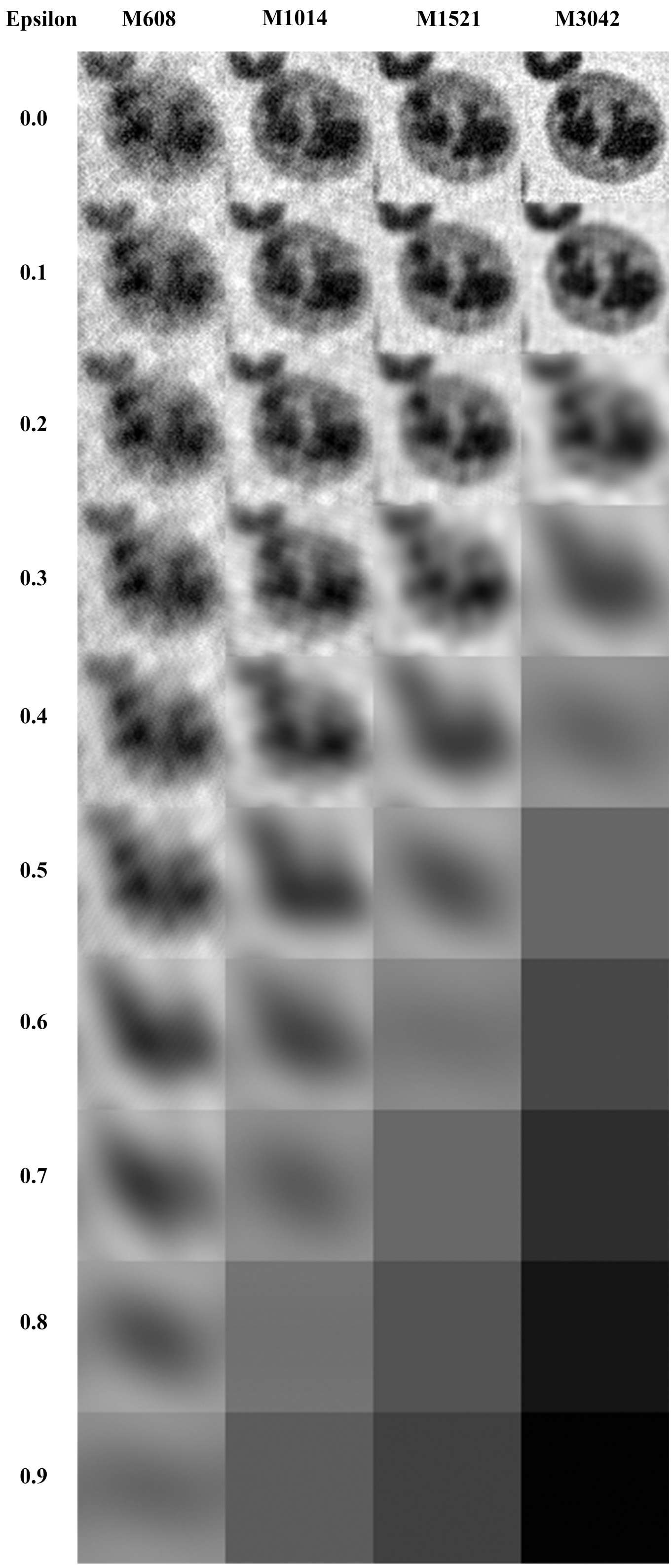

FIGURE 5: l_1-norm reconstitution, using the second robust program over noisy measurements

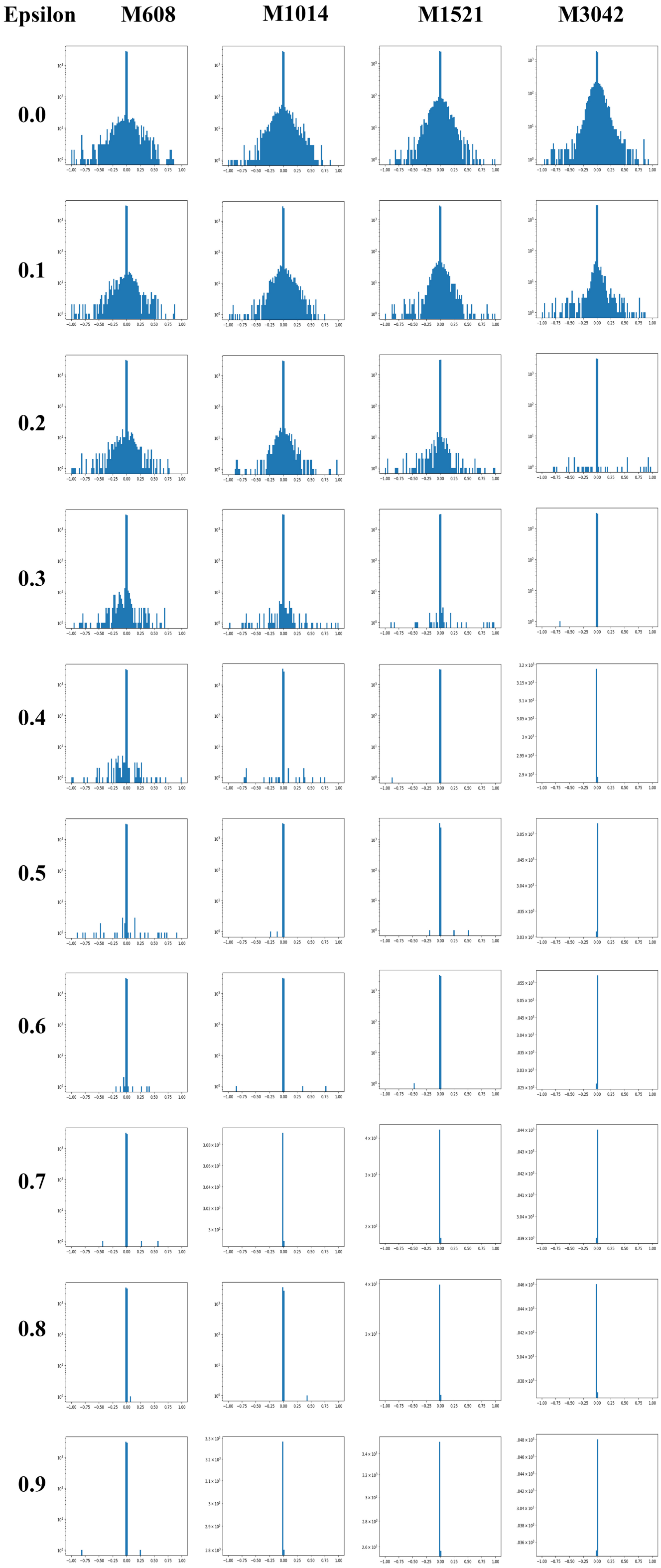

FIGURE 6: l_1-norm reconstitution, using the second robust program over noisy measurements. Histograms of $\hat{x}$ values in logarithmic scale