Collegio Universitario Luciano Fonda

Università degli Studi di Trieste

Anno accademico 2022-2023

Elaborato finale

(Machine) Learning Inverse Problems

Ispirato alla conferenza

Seeing the invisible with inverse problems: from mathematics to artificial intelligence

RELATORE:

Prof. Giovanni Alberti

STUDENTI:

Michele Alessi
Massimo Lupo Giulio Di Cosimo
Andrea Lavarone
Antonio Santaniello

Contents

Abstract

This project aims to briefly present the mathematical setting of Inverse Problems and provide different contexts in which they can be applied and understood concretely.
After introducing the general idea of an Inverse Problem, we explain what a *Regularization* of the latter is, focusing on a linear algebraic setting to keep the arguments simple. This background facilitates the translation of the abstractly sketched ideas into terms of imaging techniques.
Lastly, we mention how Machine Learning can serve as an ally to tackle the issues related to Inverse Problems. In the last section, we present some examples to demonstrate how classic algorithms and artificial intelligence methods perform in a few problems related to images.

1 Inverse Problems

Inverse problems can be intuitively interpreted as the task of reconstructing a cause from its observable effects. In general, an inverse problem is the recovery of unknown data that has undergone a certain transformation, based on the observations of the outcome. This represents the critical issue of inverse problems: the only available information is the result of the transformation of the unknown data. Introducing some symbolism, inverse problems can be stated as follows. Given the observable effect y and the cause x, the two are related through an equation of the form:

$$y = A(x) + \varepsilon, \tag{1}$$

where A is a map, called *forward operator*, that is not necessarily linear (however, in this project the linear case will be analyzed in depth) and ε represents noise. It is also possible, more generally, to model the relation between x and y with the following equation:

$$y = \mathcal{N}(A(x)),$$

where $\mathcal{N}$ is a probability distribution. An inverse problem consists of finding x given y.
The first observation we can make about the problem expressed in this way is that the problem is ill-posed. For a problem to be well-posed, it has to satisfy the three Hadamard conditions:

1. the problem has a solution;
2. the solution is unique;
3. the solution changes continuously with the parameters of the problem.

The problem cannot be defined as well-posed since it is not possible, in general, to recover a unique solution from y without making any assumption about x, that is without making any assumption about the data.

1.1 Regularization techniques

In this section, we will see how the features explained above show up in concrete examples, and we will discuss some procedures that allow us to tackle these problems.
As a typical transformation to be inverted, we first consider a linear operator acting between vector spaces $A : \mathbb{R}^n \to \mathbb{R}^m$, naturally equipped with Euclidean metric. Let $m = n$ and the map A be invertible and positive with eigenvalues $0 < \lambda_1 < ... < \lambda_n$. It is possible to show [1] that given an error of order δ (supposed to be a small quantity) within image vectors, it translates into an error of order at most $\kappa\delta$ between vectors in the domain of A, where $\kappa = \frac{\lambda_n}{\lambda_1}$ is called *condition number*. This implies that large values of κ tend to amplify noise over the matrix inversion, and therefore make it a delicate task. Moreover, in general, the map A that we are interested in reversing may have a nontrivial kernel which prevents us from finding an inverse, instead the values of x that satisfy (1)[1] span an affine space in the domain. These two obstructions generally make the inverse problems ill-posed. This abstract setup, yet being far from the most general one, serves in many cases involving the manipulation of images. Any image can be thought of as a vector in $\mathbb{R}^n$ where n is of the order of the number of pixels, the linear map A represents the process undergone by the images that we want to reverse and the noise ε is a random perturbation that always occurs to some extent in image acquisition. The problems outlined before in the linear algebraic setting show up in daily Imaging tasks, some of the inverse problems we may want to solve in this context are displayed in the following table [2].

[1] Now to be intended as a vector equation.

Technique	Description	Forward Operator
Denoising	Remove noise from an image	Identity
Deblurring	Increase detail	$A_{ij} = g_{i-j+1}$

The techniques that aim to solve the issues of an ill-posed problem try to integrate some extra, a priori knowledge on the desired inverted data x_{rec}, and go under the name of Regularization. In the following we focus on *analytic techniques*, meaning that the a priori knowledge is implemented deterministically, as opposed to the case where statistical properties of the desired inversion are known. Between all the possible ways to integrate extra knowledge to constrain the inversion to a well-posed problem, we present the so-called *Variational Approaches* that are best suited to be implemented using Machine Learning algorithms [3]. Let's first rephrase the inversion problem to a minimization task, namely: (1) $\iff$ $||y - Ax - \varepsilon|| = 0$. Being the norm positive we can find the x_{rec} satisfying the above equation by minimizing with respect to x. Now, the above issues related to the inversion of A translate to classical issues of minimizing functions, in fact, a nontrivial kernel translates into flat directions of the function, and noise sensitivity also implies a slowly varying function in some direction. The new formulation allows us to modify the equation by introducing a Regularizer $\mathcal{R}_\theta(x)$, which is a function of x depending on a set of parameters θ:

$$||y - Ax - \varepsilon||^2 + \mathcal{R}_\theta(x). \tag{2}$$

The explicit form of the regularizer depends on which plausible vectors satisfying (1) we want to promote to x_{rec} based on our a priori information, the values of the parameters are to be tuned to get the best estimate from the minimization. Some standard examples are [3]:

- Tychonov: $\mathcal{R}_\theta(x) = \theta||x||^2$.
  The regularizer consists of the norm of the vector, therefore vectors' smaller norms are prioritized by the minimization of the regularized equation. Let's visualize geometrically what is going on with a low-dimensional example. Set $A : \mathbb{R}^2 \to \mathbb{R},\ A = (1\ 0)$, then using GeoGebra we can plot the function to be minimized at different values of the parameter θ.

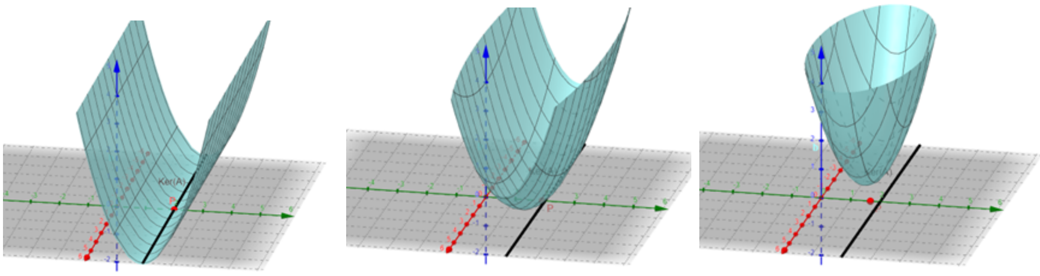

Figure 1: From left to right: $\theta = 0$, $\theta = 0.05$, $\theta = 0.2$

  It is clear that the larger the parameter value the easier the function is to minimize but the further the minimum lays from the original (unregularized) locus, therefore a compromise is to be reached. We can also notice that among all possible values of the kernel of A, our choice of the explicit form of $\mathcal{R}(x)$ selects points closer to the origin.

- Total Variation: $\mathcal{R}_\theta(x) = \theta||\nabla x||_1 = \theta \sum_{<i,j>} |x_i - x_j|^2$.
  This regularizer was historically used to deal with inverse problems concerning Imaging, as the one listed before. The function implemented for the regularization penalizes vectors with large variations between neighboring pixels and therefore promotes smoothness.

[2]where $< i, j >$ denotes a sum over first neighbourghing pixels

1.2 Machine-learning approaches

Machine learning offers a versatile and data-driven approach to solving inverse problems, making it a valuable tool in fields where extracting meaningful information from data is essential. It can be used to automatize the choice of the regularization and the tuning of its parameters.
In the following section, we will present some of the most common training techniques in machine learning in the context of inverse problems.
Training in machine learning refers to the process of teaching a machine learning model to make predictions or decisions based on data. There are two fundamental types of training: supervised and unsupervised. The first one uses a labeled dataset where each input is already associated with an output. In contrast, the second one deals with unlabeled data, where the model's objective is to discover patterns or structures in the data without explicit guidance in the form of labels.
In the end, the best way to train a model in an inverse problem comes from the level of knowledge of the measurement operator A and from the possibility of accessing the (x, y) pairs. We will now explore all the possible cases concerning these two factors.

1. A fully known.
   In this case, the fact of having the (x, y) pairs already coupled, decoupled, or having just the ground truth x doesn't change the training methods since we can apply the A operator and treat the problem just as we know (x, y). A simple way to incorporate the knowledge of A in the training process is to apply to y an approximate inverse $\hat{A}^{-1}$ and then use a supervised learning approach to train the network to remove the artifacts coming from the approximation. Other techniques can include a recurrent block architecture to force the reconstructive network to learn how to behave as a proximal operator [4]. If there is access only to the measurements y themself, more advanced methods relying on some statistical assumptions have been developed [5].

2. A partially known.
   In this case, it is necessary to decompose the analysis into four different cases based on where we want to train the network:

   - Matched (x, y) pairs: In this case, the idea is always to use supervised learning to instruct the network to remove the artifacts that come from the partial knowledge of A
   - Uncoupled x's and y's: In this case, a powerful strategy comes from an algorithm called CycleGAN. It aims to train two generative models $F : X \to Y$ ang $G : Y \leftarrow X$ (where X is the domain of the x's and Y and is the domain of the y's) that are approximately one the inverse of the other so that $F(G(y)) \approx y$ and vice versa. In this way, by minimizing the operator $\mathbb{E}[||x - G(F(x))||_1] + \mathbb{E}[||y - F(G(y))||_1]$, where $\mathbb{E}$ denote the expected value, one can train F and G in order to map togheter the (x, y) pairs.
   - From x's only: A very successful method in this case involves an architecture called autoencoder. The general idea is to take the ground truth data, apply some kind of transformation to simulate a possible outcome of a measurement, and then train the autoencoder to reconstruct the original data by minimizing a loss function according to a chosen metric (that is a way to quantify how close two objects are). Figure 2 sketches how the procedure works. Another example of this method is GANs. They consist of two neural networks: a generator and a discriminator. The generator tries to generate data that is indistinguishable from real data, while the discriminator tries to distinguish between real and generated data. Through a competitive process, GANs implicitly capture the underlying data distribution without explicitly defining a generative prior. The generator learns to generate samples that match the real data distribution as closely as possible. To apply this method in the case of inverse problems, it is necessary to at least know the distribution of A [6].

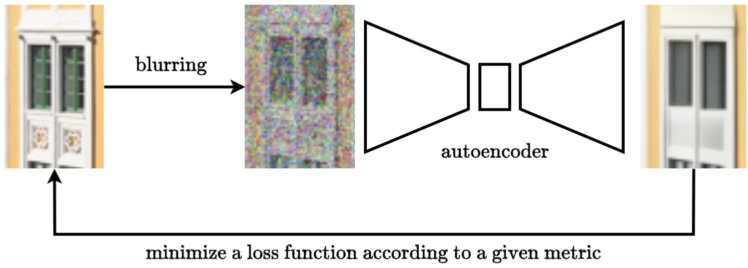

Figure 2: A scheme of how Autoencoders can be used in inverse problems

- From y's only: If it is possible to have access to a noisy version $\hat{x}$ of x so that $\mathbb{E}[\hat{x}(y)] = x$, then the learned neural network is $f_{\hat{\theta}}$, where $\hat{\theta} = argmin_{\theta}[\mathbb{E}[||f_{\theta}(y) - \hat{x}||^2]]$. If it is not possible to have access to the ground truth alone, more sophisticated methods have been recently developed such as AmbientGAN [7].

3. *A* Completely unknown.
In this case, if the pairs (x, y) are known, the simplest way to tackle the problem is to treat the reconstruction map $y \rightarrow x$ as a "black box" that can be well approximated by conventional neural networks trained with backpropagation. If it is not possible to access the coupled pairs (x, y), then there are few options. Recently, a preprint has been published proposing a model that attempts to address this challenge by incorporating information from multiple incomplete sensing operators $A_1, ..., A_n$ or assuming that the signal model is invariant under a certain group action to train a model from the y's measurements only. The general principle is that each operator can provide additional information about the signal model if it has a different null space [8].

2 Examples

In the present section, we will present our results concerning what was exposed in the previous two chapters.
Two problems are faced, namely denoising and deblurring an image. Both have been addressed by applying standard regularization techniques and enhancing the power of contemporary neural networks.

2.1 Deblurring problem

To solve the deblurring problem, we have initially applied a Gaussian filter to the original image, thus obtaining a blurred version of the image, as shown in Figure 3. The blurred image now

Figure 3: On the left the original image, on the right the blurred image.

plays the role of the measurement, and our task is to recover the original image as much as we can. On one hand, the image is deblurred using only regularization techniques, specifically an iterative algorithm based on Total Variation (TV) Minimization. On the other hand, a DiffPIR [9] diffusion algorithm is deemed suitable for the problem and thus applied to our image, leading to the results of Figure 4.

Figure 4: From left to right: TV and DiffPIR solutions for the deblurring problem.

2.2 Denoising problem

To address the denoising problem, we employ a similar methodology. Initially, random noise is introduced into the original image. In Figure 5, two different levels of noise are observable. Subsequently, as previously discussed, we explore two approaches to tackle the inverse problem.

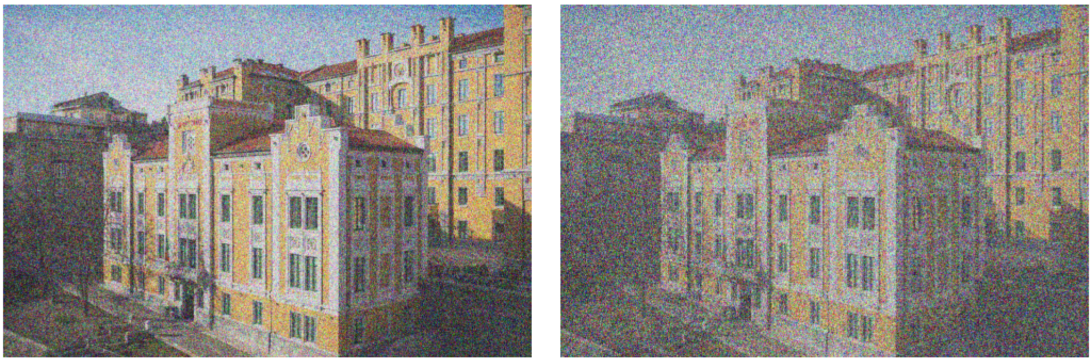

Figure 5: From left to right: noise level set to 70% and 95%

Regarding regularization techniques, we initially assess Total Variation. However, as demonstrated in Figure 6, its performance falls short. Consequently, we incorporate another regularization technique called Wavelet denoising [10], which enhances the former. We employ the

Figure 6: From left to right: TV, Wavelet, and DDRM solutions for noise level 70%.

DDRM [11] diffusion algorithm to offer a solution relying on neural networks. In Figure 7, illustrating the inverse problem solutions with a noise level set at 95%, it becomes evident that in this context, neural networks outperform conventional regularization methods.

Figure 7: From left to right: Wavelet and DDRM solutions for noise level 95%.

3 Conclusions

In conclusion, we analyzed inverse problems from the formal idea to real-world applications. Our study aimed to grasp the fundamental blocks of this discipline and to be able to see them in action through both regularization iterative algorithmic and AI perspectives. For these reasons, we selected a few enlightening examples that we hope serve as instructive visualizations. The reader may keep in mind that these techniques are employed in various fields, ranging from Medical Diagnosis to Observational Astrophysics. We also emphasized the geometric side of the subject, heavily relying on linear algebra. We understand that this may cut out a fraction of the readers, but we considered its insights too powerful to be excluded. Moreover, this was the easiest abstract setup to introduce the subject; more powerful and general arguments involve Functional Analysis in Hilbert or Banach Spaces but are beyond this project's scope.

References

[1] Martin Burge. "Lecture Notes - Inverse Problems". In: (2007).

[2] Gregory Ongie et al. *Deep Learning Techniques for Inverse Problems in Imaging*. 2020. arXiv: `2005.06001 [eess.IV]`.

[3] Simon Arridge et al. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174. DOI: `10.1017/S0962492919000059`.

[4] Y. Chen and T. Pock. "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2018), pp. 1256–1272.

[5] Y. C. Eldar. "Generalized sure for exponential families: Applications to regularization". In: *IEEE Transactions on Signal Processing* 57 (2009), pp. 471–481.

[6] D. Mishkin O. Kupyn V. Budzan. M. Mykhailych and J. Matas. "Deblurgan: Blind motion deblurring using conditional adversarial networks". In: *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 8183–8192.

[7] E. Price A. Bora and A. G. Dimakis. "Ambientgan: Generative models from lossy measurements." In: *ICLR* 2 (2018), p. 5.

[8] D. Chen J. Tachella and M. Davies. "Unsupervised Learning From Incomplete Measurements for Inverse Problems". In: (2022). URL: `https://arxiv.org/abs/2201.12151`.

[9] Yuanzhi Zhu et al. "Denoising Diffusion Models for Plug-and-Play Image Restoration". In: (2023).

[10] Guomin Luo and Daming Zhang. "Wavelet Denoising". In: Apr. 2012. ISBN: 978-953-51-0494-0. DOI: `10.5772/37424`.

[11] Bahjat Kawar et al. "Denoising Diffusion Restoration Models". In: (2022). arXiv: `2201.11793 [eess.IV]`.