Exploring instabilities of inverse problem solvers with low-dimensional manifolds

Nathanaël Munier, Emmanuel Soubies, Pierre Weiss

Chapter 1
Exploring instabilities of inverse problem solvers with low-dimensional manifolds

Nathanaël Munier, Emmanuel Soubies & Pierre Weiss

Abstract Inverse problem solvers are mappings $S : \mathscr{Y} \to \mathscr{X}$, where $\mathscr{Y}$ is the space of measurements and $\mathscr{X}$ the space of signals we wish to recover. We propose a simple algorithm to visualize the main instability of a solver implemented within an automatic differentiation framework. We justify it through simple considerations and illustrate its behavior on a deconvolution problem solved with a neural network based reconstruction method. The proposed algorithm can be used to provide additional insights on the properties of inverse problem solvers, and can be viewed as a simple uncertainty quantification technique.

1.1 Introduction

Consider a forward model of the form $y = \mathscr{P}(A(x))$, where $\mathscr{X}$ and $\mathscr{Y}$ are two vector spaces with $\dim(\mathscr{X}) = N$, $\dim(\mathscr{Y}) = M$, the map $A : \mathscr{X} \to \mathscr{Y}$ is a sensing mapping and $\mathscr{P} : \mathscr{Y} \to \mathscr{Y}$ is a deterministic or random perturbation (e.g., quantization, additive noise). An inverse problem solver $S : \mathscr{Y} \to \mathscr{X}$ takes the measurements y as an input and returns an estimate $S(y)$ of the underlying vector x. We design a numerical algorithm to characterize the main directions of instability of the solver. It leverages the power of automatic differentiation and enables visualizing the instabilities as a low-dimensional manifold. This is a follow-up work of a recent method called Jackpot [9], that we are developing in parallel of this paper in a more intricate setting where the solver is not available explicitly.

Nathanaël Munier
IMT, 118 Rte de Narbonne, 31400 Toulouse
e-mail: nathanael.munier@ens-rennes.fr

Emmanuel Soubies
IRIT, Cr Rose Dieng-Kuntz, 31400 Toulouse

Pierre Weiss
IRIT, Cr Rose Dieng-Kuntz, 31400 Toulouse

The problem

Solving ill-posed inverse problems requires introducing explicit or implicit regularization terms. These terms improve the stability of the inversion to noise on the data. However, strong regularizers may also hinder the recovery of accurate solutions. Hence, there is a balance to be found between stability and accuracy [4]. The solvers' accuracy has improved considerably with the advent of deep learning, and we can expect less stability. While the estimated signals/images typically look sharp with details we would not have dreamt of recovering a few years ago, to what extent can we trust them? Two popular approaches are often used to answer this critical question.

Sampling methods

To simplify the discussion, we assume that the perturbation $\mathscr{P}$ is an additive white Gaussian noise, that is $y = A(x) + b$ with $b \sim \mathscr{N}(0, \sigma^2 \mathrm{Id}_M)$. Hence, if the recovery mapping S is unstable, the noise b may result in unfaithful reconstructions. Sampling methods [8, 5, 1] consist in constructing a set of measurements $y_k = y + b_k$, for $k = 1, \dots, K$, where $b_k \sim \mathscr{N}(0, \varepsilon^2 \mathrm{Id}_M)$ and recover the corresponding estimates $x_k = S(y_k)$. The samples $(x_1, \dots, x_K)$ provide a snapshot of how the solver S reacts to perturbations and can be analyzed through statistical techniques such as Principal Component Analyses (PCA) or bootstrapping methods [2]. For instance, by setting $\varepsilon = \sigma$ and letting $K \to \infty$, this approach provides a complete description of the pushforward distribution $S(\mathscr{N}(y, 2\sigma^2 \mathrm{Id}_M))$. It yields a good overview of the average uncertainty lying in the estimation, however it can suffer from low convergence rates due to the curse of dimensionality and fails to identify the worst case attacks.

Adversarial attacks

An alternative strategy consists in searching for worst case perturbations [6]. They can be obtained by solving optimization problems of the form:

$$b^\star = \underset{b \in \mathscr{Y}, \|b\|_2 \leq \varepsilon}{\operatorname{argmax}} \|S(y+b) - S(y)\|_2^2, \tag{1.1}$$

that is, we look for the perturbation of amplitude ε that most alters the reconstruction. This problem can be solved with projected gradient descent for instance. This type of approaches is very handy to illustrate that some solvers can create dangerous hallucinations in medical applications [4, 3]. However, it is highly nonconvex so that solvers can easily get trapped in shallow local minimizers. In addition, it requires adding some extra noise to the already noisy data, therefore yielding pessimistic stability results.

Our contribution

In this paper, we propose an alternative to these popular techniques, which lies in between worst case and average attacks. We are looking for a D-dimensional submanifold in $\mathscr{X}$ that best captures the uncertainty/unstability set $\mathscr{U}_y = S(y+\mathscr{B}(0,\varepsilon))$. Here D is a user defined parameter, and $\mathscr{B}(0,\varepsilon)$ denotes a ball centered at 0 of radius ε. This algorithm is rather straightforward to implement, given that the solver is implemented in an automatic differentiation framework such as PyTorch, Tensorflow or Jax. The article will be organized as follows: in Section 1.2 we introduce our proposed algorithm and some approximation properties about the computed adversarial manifold. In Section 1.3, we illustrate its behavior on physics informed deep learning solvers.

1.2 Main idea

The main idea is rather simple. If the perturbation b is of small amplitude ε and the mapping S is of class C^1 we can linearize it:

$$S(y+b) = S(y) + J_S(y)\cdot b + o(v\|b\|_2), \tag{1.2}$$

where $v \in \mathscr{X}$ is a unit vector and $J_S(y) \in \mathbb{R}^{N\times M}$ denotes the Jacobian matrix of the solver S evaluated at point y.

For additive white Gaussian noise $b \sim \mathscr{N}(0,\varepsilon^2\mathrm{Id}_M)$ and small ε, the linearization (1.2) informs us that the uncertainty around the estimate $S(y)$ is approximately distributed as $\mathscr{N}\left(0,\varepsilon^2 C_y\right)$, where $C_y \overset{\text{def}}{=} J_S(y)^T J_S(y)$. Hence, the main directions of uncertainty are given by the eigenvectors associated to the dominant eigenvalues of this symmetric positive semi-definite matrix.

Given an integer $D \in \mathbb{N}$, we can define the subspace $\mathscr{T}_y = \mathrm{span}(v_1,\dots,v_D)$ generated by the eigenvectors of C_y associated to the D largest eigenvectors $\lambda_1 \geq \cdots \geq \lambda_D$. Then we design two approximations of the uncertainty set $\mathscr{U}_y = S(y+\mathscr{B}(0,\varepsilon))$:

- $\mathscr{L}_y = S(y) + J_S(y)\cdot\mathscr{T}_y$: the linearized D-dimensional space,
- $\mathscr{M}_y = S(y+\mathscr{T}_y \cap \mathscr{B}(0,\varepsilon))$: the D-dimensional manifold.

Let $d(A|B) \overset{\text{def}}{=} \max_{x\in A} d(x,B)$ denote the maximum distance from A to B.

Proposition 1 *The sets $\mathscr{L}_y$ and $\mathscr{M}_y$ admit the following properties.*

- *If S is a quadratic function, the space $\mathscr{L}_y$ minimizes $d(\mathscr{U}_y|L)$ among affine spaces L. It is the optimal D-dimensional affine space to approximate the uncertainty set $\mathscr{U}_y$.*
- *If S is of class C^1, the linearized space $\mathscr{L}_y$ is asymptotically the best D-dimensional affine approximation space of the uncertainty set $\mathscr{U}_y$ as it minimizes $L \mapsto \lim_{\varepsilon\to 0} \frac{d(\mathscr{U}_y|L)}{\varepsilon}$. Moreover $\lim_{\varepsilon\to 0} \frac{d(\mathscr{U}_y|\mathscr{L}_y)}{\varepsilon} = \sqrt{\lambda_{D+1}}$ where λ_{D+1} is the $(D+1)^{th}$ eigenvalue of C_y.*

- *If $D \leq \operatorname{rank}(J_S(y))$, the set $\mathscr{M}_y$ is locally a submanifold of $\mathscr{U}_y$ near $S(y)$. It has the same property as $\mathscr{L}_y$:* $\lim_{\varepsilon\to 0} \frac{d(\mathscr{U}_y|\mathscr{M}_y)}{\varepsilon} = \sqrt{\lambda_{D+1}}$.

Numerical computation

Algorithm 1 shows how to compute a discretized subset of the manifold $\mathscr{M}_y$. Computing matrix-vector products with the Jacobian can be achieved through automatic differentiation. Depending on the size of $\mathscr{Y}$, one can either store the matrix C_y or use iterative methods derived from the power method. In what follows, we compute the dominant eigenpairs using the LOBPCG algorithm [7]. It can be seen as a Stiefel manifold gradient descent with a random initialization.

Algorithm 1 Our proposed method

Require: $y \in \mathscr{Y}$, $\varepsilon > 0$, $D \leq M$ and $\mathscr{G}$ a set of points in the ball $\mathscr{B}_D(0,\varepsilon)$
Compute $V = [v_1, \dots, v_D]$ the D dominant eigenvectors of C_y
Evaluate $S(y+Vg)$ for each $g \in \mathscr{G}$
return $\mathscr{M}_y(\mathscr{G}) = S(y+V\mathscr{G})$ a discretized version of $\mathscr{M}_y$.

Discussion

While sampling methods provide a picture of the average sensitivity, adversarial attacks give the worst case $S(y+b^\star)$ of sensitivity as defined in (1.1). In comparison, our approach can be seen as an intermediate between these two classes of methods. For $D = 1$, it provides an approximation $S(y \pm \varepsilon v_1)$ of the worst case perturbation, with an error of order $o(\|\varepsilon\|_2)$. Moreover, it provides information beyond this worst case analysis by taking $D > 1$. In this case, the main directions of uncertainty are identified and a discretization of the sub-manifold $\mathscr{M}_y$ is computed.

Compared to conventional sampling scheme, our method captures more extreme cases. Indeed, the mean sampling distance to the center $S(y)$ for the sets $\mathscr{U}_y$ and $\mathscr{M}_y$ can be approximated by (1.3) and (1.4) below:

$$\mathbb{E}_{b\sim\mathscr{N}(0,\varepsilon^2\mathrm{Id}_M)}\left[\frac{\|S(y+b)-S(y)\|_2^2}{\varepsilon^2}\right] \underset{\varepsilon\to 0}{\longrightarrow} \frac{1}{M}\sum_{m=1}^{M}\lambda_i \tag{1.3}$$

$$\mathbb{E}_{g\sim\mathscr{N}(0,\varepsilon^2\mathrm{Id}_D)}\left[\frac{\|S(y+Vg)-S(y)\|_2^2}{\varepsilon^2}\right] \underset{\varepsilon\to 0}{\longrightarrow} \frac{1}{D}\sum_{d=1}^{D}\lambda_d. \tag{1.4}$$

Assume that D is selected in such a way that the sums of eigenvalues are close $\sum_{m=1}^{M}\lambda_m \approx \sum_{d=1}^{D}\lambda_d$. Then notice that our method captures points that are in average

$\sqrt{M/D}$ times larger than with conventional sampling methods. This is a dramatic difference for fast decaying spectra in large dimensions.

1.3 Experiments

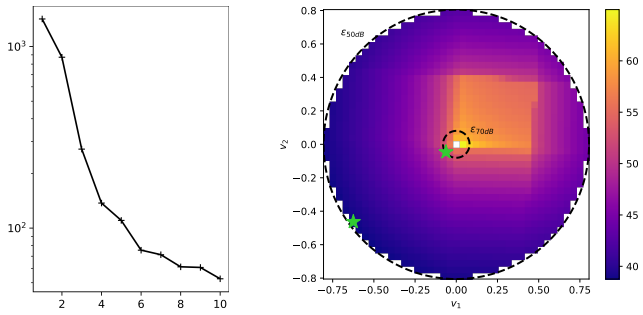
To illustrate the method, we consider an image deblurring problem. We set a convolution operator A with a motion Point Spread Function (PSF) and $\mathscr{P}$ an additive white Gaussian noise. The clean image, the motion point spread function, and the degraded image are depicted in Figure 1.1 (a)–(c). Plug-and-play (PnP) methods with deep-learning based denoisers yield accurate reconstruction results for such problems. For this experiment, we used a half-quadratic splitting solver S where the proximal step is replaced by a DRUNet denoiser [10]. A reconstruction result is presented in Figure 1.1 (d).

(a) Clean image (b) Degraded image (c) Restored image

Fig. 1.1: Deconvolution problem. From left to right: clean image with point spread function on top left corner, degraded (blurry and noisy) image, restored image using the PnP method (see text for details). At the right side of each image, two zoomed-in sections are plotted.

Algorithm 1 results

We use Algorithm 1 to analyze the stability of the considered PnP method. The outputs of Algorithm 1 are presented in Figure 1.2. We observe that C_y admits two dominant eigenvalues. As such, we set the dimension of the computed manifold $\mathscr{M}_y$ to $D=2$. In Figure 1.2 (b) we present the output SNR on $\mathscr{M}_y$, that is $\mathrm{SNR}(S(\tilde{y}),S(y))$ for points $\tilde{y}\in\mathscr{M}_y$. We observe that the decrease of output SNR is similar in both directions which is in agreement with the fact that the first two eigenvalues are of the same order of magnitude. The two stars represent the worst output SNR for the inputs perturbation levels $\varepsilon_{50\mathrm{dB}}$ and $\varepsilon_{70\mathrm{dB}}$; that is perturbations $y+b$ of y such that $\mathrm{SNR}(y+b,y)$ equals 50dB and 70dB. The corresponding restored images $S(y+b)$ are displayed in Figures 1.3 (b) and 1.4 (b). We observe that some structures of the initial image are hallucinated when restoring a slightly perturbed input y. This shows that the considered PnP algorithm presents some instabilities which are identified by our method.



(a) Eigenvalues of C_y (b) Output SNR in dB

Fig. 1.2: Stability analysis using Algorithm 1. The output SNR in (b) corresponds to $\mathrm{SNR}(S(y+Vg),S(y))$ where V contains the eigenvectors associated to the two largest eigenvalues ($D=2$) and $\mathscr{G}$ is a Cartesian grid. The two dashed circles represent the input perturbation levels (on y) of 50dB and 70dB. The green stars denote the worst output SNR for the input perturbation levels 50dB and 70dB.

Comparison with sampling methods and adversarial attacks

For both levels $\varepsilon \in \{\varepsilon_{50\mathrm{dB}}, \varepsilon_{70\mathrm{dB}}\}$, we applied the sampling and adversarial attack methods described in the introduction. For the sampling method, we considered $K = 200$ points $y_k = y + b_k$, $k \in \{1,\dots,K\}$, where the b_k are uniformly sampled on $\mathscr{S}_M(0,\varepsilon)$, the M-dimensional centered sphere of radius ε. Note that we restricted the generation to the sphere rather than the ball as we are interested in finding the worst perturbation. Regarding the adversarial attack, we solved Problem (1.1) using a projected gradient ascent from 20 random initial points in $\mathscr{S}_M(0,\varepsilon)$, and kept the best local maximum.

We report on Figures 1.3 and 1.4 the deconvolved images with the worst case perturbation (i.e., the one leading to the lower output SNR) found by each method. We observe that the sampling method is unable to detect the instabilities of the solver, as opposed to the proposed method and the adversarial attack. Moreover, as expected, the proposed method reaches the performance of the adversarial attack asymptotically when $\varepsilon \to 0$ (Figure 1.4). For larger levels, we observe that the adversarial attack lead to hallucinations not captured by Algorithm 1 with $D = 2$. Yet, the proposed method is not restricted to the computation of a single attack and provides a whole manifold of adversarial perturbations. Overall, it yields a compromise between sampling methods and adversarial attacks.

Acknowledgements The authors acknowledge a support from the ANR Micro-Blind ANR-21-CE48-0008 and from the HPC resources from GENCI-IDRIS (Grant 2021-AD011012210R3).

(a) Sampling (57.2 dB) (b) Our method (38.8 dB) (c) Adversarial (29.8 dB)

Fig. 1.3: Comparison of computed worst perturbations for $\varepsilon = \varepsilon_{50\text{dB}}$.

(a) Sampling (77.6 dB) (b) Our method (51.4 dB) (c) Adversarial (48.6 dB)

Fig. 1.4: Comparison of computed worst perturbations for $\varepsilon = \varepsilon_{70\text{dB}}$

References

1. Johnathan M Bardsley, Antti Solonen, Heikki Haario, and Marko Laine. Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 2014.
2. Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.
3. Martin Genzel, Jan Macdonald, and Maximilian März. Solving inverse problems with deep neural networks–robustness included? *IEEE transactions on pattern analysis and machine intelligence*, 2022.
4. Nina M Gottschling, Vegard Antun, Anders C Hansen, and Ben Adcock. The troublesome kernel–on hallucinations, no free lunches and the accuracy-stability trade-off in inverse problems. 2020.
5. Jon C Helton, Jay Dean Johnson, Cedric J Sallaberry, and Curt B Storlie. Survey of sampling-based methods for uncertainty and sensitivity analysis. *Reliability Engineering & System Safety*, 2006.
6. Yixing Huang, Tobias Würfl, Katharina Breininger, Ling Liu, Günter Lauritsch, and Andreas Maier. Some investigations on robustness of deep learning in limited angle tomography. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018*. Springer, 2018.
7. Andrew V Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM journal on scientific computing*, 2001.
8. Klaus Mosegaard and Albert Tarantola. Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth*, 1995.
9. Nathanaël Munier, Emmanuel Soubies, and Pierre Weiss. Jackpot: Approximating uncertainty domains with adversarial manifolds. *HAL preprint*, 2024.
10. Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.