Statistical Learning and Inverse Problems: A Stochastic Gradient Approach

Anonymous Author(s)
Affiliation
Address
email

Abstract

Inverse problems are paramount in Science and Engineering. In this paper, we consider the setup of Statistical Inverse Problem (SIP) and demonstrate how Stochastic Gradient Descent (SGD) algorithms can be used in the linear SIP setting. We provide consistency and finite sample bounds for the excess risk. We also propose a modification for the SGD algorithm where we leverage machine learning methods to smooth the stochastic gradients and improve empirical performance. We exemplify the algorithm in a setting of great interest nowadays: the Functional Linear Regression model. In this case we consider a synthetic data example and examples with a real data classification problem.

1 Introduction

Inverse Problems (IP) might be described as the search of an unknown parameter (that could be a function) that satisfies a given, known equation. Considering the notation:

$$y = A[f] + \text{ noise},$$

where f and y are elements of given Hilbert spaces, we would like to compute (or estimate) f given the data y for some level of noise. Typically, IPs are ill-posed in the sense that the solution does not depend continuously on the data. There are several very important and impressive examples of IPs in our daily lives. Medical imaging has been using IPs for decades and it has shaped the area, as for instance, Computerized Tomography (CT) and Magnetic Resonance Imaging (MRI). For an introductory text, see Vogel [2002].

A vast literature of IPs is devoted to deterministic problems where the noise term is also a element of a Hilbert space and usually assumed small in norm, which is not usually verified in practice. In this work, we will take a different avenue, known as Statistical Inverse Problems (SIP). This approach is a formalization of IPs within a probabilistic setting, where the uncertainty of all measurements are properly considered. Our focus in this work is to propose a direct and practical method for solving SIP problems and, at the same time, provide theoretical guarantees for the excess risk performance of the algorithm we develop. Our algorithm is based on a gradient descent framework, where stochastic gradients (or base learner that approximates the stochastic gradients) are used to estimate general functional parameters.

The paper is organized as follows. We finish this section contextualizing our paper in the broad literature and stating our main contributions. In Section 2, we formally introduce the learning problem that we analyze. In Section 3, we provide examples of practical problems that fits within our formulation. In Section 4 we provide our main results and algorithms. Finally, in Section 5 we provide numerical examples and a real data application for a Functional Linear Regression problem (FLR). Due to space constraints, some of the figures, proofs and experiments were moved to the supplementary material.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

1.1 Contribution

We provide a novel numerical method to estimate functional parameters in SIP problems using stochastic gradients. More precisely, we extend the properties and flexibility of SGD and boosting algorithms to a broader class of problems by bridging the gap between the IP and optimization communities.

Whereas most of the IPs methods focus on regularization strategies to "invert" the operator A, we propose a gradient descent type of algorithm to estimate the functional parameter directly. Our algorithm works in the same spirit as Stochastic Gradient Descent algorithms with sample averaging. While results of SGD are well understood in the context of regression problems in finite and infinite dimensions and SGD is well understood in deterministic IPs, SGD have not yet been considered under the SIP formulation.

We show that our procedure also ensures risk consistency in expectation and high probability under the statistical setting. Furthermore, we propose a modification in our algorithm to substitute the stochastic gradients by base learners similarly to boosting algorithms, Mason et al. [1999], Friedman [2001]. This modification improve a common challenge faced by SIP problems: the discretization procedures of the operator A that arises in SIP.

1.2 Literature Review

Historically, SIP was first introduced in Sudakovand and Khalfin [1964] where IPs from Mathematical Physics were recast into a statistical framework. For a more structured introduction, we forward the reader to Kaipio and Somersalo [2004]. Several advances were made in the parametric approach to SIP, where the unknown function is assumed to be completely described by an unknown parameter living in a finite dimensional space, see for instance Evans and Stark [2002]. In our paper, however, we will consider the nonparametric framework as described in Cavalier [2008]. In this setting, we see the IP as a search of an element of an infinite dimensional space.

When considering IPs (and SIP, in particular), there are several ways to regularize the problem in order to deal with its ill-posedness. For instance, one could consider roughness penalty or a functional basis as in Tenorio [2001]. Additionally, one could examine Tikhonov and spectral cut-off regularizations as in Bissantz et al. [2007]. For many of those standard approaches, consistency under the SIP setting and rates of convergences were established. See for instance Bissantz et al. [2004], Bissantz and Holzmann [2008]. A thoroughly discussion of stochastic gradient algorithms is outside the scope of this work and we refer the reader to Zinkevich [2003], Nesterov et al. [2018] and references therein.

Applications of Machine Learning (or Deep Learning, in particular) to solve IPs have been numerous in recent years, but, in our opinion, akin of the capabilities that these new techniques could bring to this area of research. Some attention have been given to imaging problems as in Jin et al. [2017] and Ongie et al. [2020]. Under deterministic IP, the paper Li et al. [2020] studies the regularization and convergence rates of penalized neural networks when solving regression problems. See also Adler and Öktem [2017] and Bai et al. [2020]. Other important references regarding SGD for deterministic IP are Jin et al. [2021], Tang et al. [2019], Jin et al. [2020].

The main examples we bring in our paper is the class of Functional Linear Regression (FLR). This problem has drawn the attention of the statistical, econometric and computer science communities in the past decade, see Cai and Hall [2006], Yao et al. [2005], Hall and Horowitz [2007]. The usual methodology applied to this problem is the well-known FDA. For example, one could consider a prespecified basis functions to regularize the regression problem Goldsmith et al. [2011] or one could use the Functional Principal Component (FPC) basis, Morris [2015]. More recently, methods inspired in machine learning for standard linear regression problems were also extended to the FLR setting, see for instance James et al. [2009], Fan et al. [2015] for methods that are suitable for high dimensional covariates or interpretable in the LASSO sense. In this work we show how our modification to the SGD algorithm can be seen as an averaging of boosting estimators and can also be used to estimate FLR models in the high-dimensional setting.

2 Problem Formulation

We start by fixing a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ and a vector space $\mathbb{X}$ of inputs. We denote the random input by $\mathbf{X} \in L^2(\Omega, \mathcal{A}, \mathbb{P})$ taking values in $\mathbb{X}$ and consider the space $L^2(\mathbb{X})$ of functions $g : \mathbb{X} \longrightarrow \mathbb{R}^d$ with inner product $\langle g_1, g_2 \rangle_{L^2(\mathbb{X})} = \mathbb{E}[\langle g_1(\mathbf{X}), g_2(\mathbf{X}) \rangle]$ and norm $\|g\|^2_{L^2(\mathbb{X})} = \mathbb{E}[\|g(\mathbf{X})\|^2] < +\infty$, where $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ are the inner product and norm of $\mathbb{R}^d$.

Similarly, we define the space $L^2(\mathbb{W})$ of functions from $\mathbb{W}$ taking values in $\mathbb{R}^k$ square-integrable with respect to the measure space $(\mathbb{W}, \mathcal{B}, \mu)$. Finally, we consider an operator $A : L^2(\mathbb{W}) \longrightarrow L^2(\mathbb{X})$. This operator defines a direct problem and we assume that it is known. Given $f \in L^2(\mathbb{W})$, we use the notation $A[f] \in L^2(\mathbb{X})$, i.e. $A[f]$ is a square-integrable function $A[f] : \mathbb{X} \longrightarrow \mathbb{R}^d$.

We are interested in solving the statistical inverse problem related to A: jointly to observing samples of $\mathbf{X}$ taking values in $\mathbb{X}$, we observe noisy samples of $A[f^\circ](\mathbf{X})$, for some fixed, unknown $f^\circ \in L^2(\mathbb{W})$, which we denote by $\mathbf{Y}$:

$$\mathbf{Y} = A[f^\circ](\mathbf{X}) + \epsilon, \tag{1}$$

where ϵ is a zero-mean random noise. The problem we will pore over in this paper is the estimation of f° based on this given sample.

Let $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ be a point-to-point loss function as, for example, the squared loss $\ell(\mathbf{y}, \mathbf{y}') = \frac{1}{2}\|\mathbf{y} - \mathbf{y}'\|^2$. We define the populational risk as:

$$\mathcal{R}_A(f) \triangleq \mathbb{E}[\ell(\mathbf{Y}, A[f](\mathbf{X}))],$$

and we would like to solve:

$$\inf_{f \in \mathcal{F}} \mathcal{R}_A(f), \tag{2}$$

where $\mathcal{F} \subset L^2(\mathbb{W})$ is a linear subspace with $f^\circ \in \mathcal{F}$. We will denote by ∂_2 the partial derivative with respect to the second argument.

Given a sample, we will study how to control the excess risk of a functional estimator $\hat{f}$ of f°:

$$\mathcal{R}_A(\hat{f}) - \inf_{f \in \mathcal{F}} \mathcal{R}_A(f). \tag{3}$$

Instead of taking the standard route of solving the Empirical Risk Minimization problem and later establishing results for (3), in 4 we show how our Algorithms allows us to control directly to tackle (3) directly by constructing stochastic gradients directly for the populational risk.

3 Examples: motivation

Before we formalize our results, we first motivate the study of Eq. (1) with a few of applications. Each of those problems have a myriad of solutions on their own. For more information on those IPs, see, for instance, Vogel [2002].

Deconvolution. This type of inverse problems relate the values of $\mathbf{Y}$ and $\mathbf{X}$ through the following convolution equation:

$$\mathbf{Y} = \int_{\mathbb{W}} k(\mathbf{X} - \mathbf{w}) f(\mathbf{w}) d\mu(\mathbf{w}) + \epsilon,$$

where $\mathbb{X} = \mathbb{W} = \mathbb{R}^d$ and the kernel k is known. In this case, we define the operator A as:

$$A[f](\mathbf{x}) = \int_{\mathbb{W}} k(\mathbf{x} - \mathbf{w}) f(\mathbf{w}) d\mu(\mathbf{w}).$$

Functional Linear Regression. Consider the scalar, multivariate response model of functional linear regression: let $\mathbb{X} = L^2([0, T])$ taking values in $\mathbb{R}^d$, $\mathbb{W} = [0, T]$, μ is the Lebesgue measure in $[0, T]$ and $\mathbf{Y}$ is giving by the following model

$$\mathbf{Y} = \int_0^T f(s) \mathbf{X}(s) ds + \epsilon,$$

where $f \in L^2(\mathbb{W})$ is taking values in $\mathbb{R}$. Here we changed the notation from $\mathbf{w}$ to s in order to keep the classical notation from FLR. In this case,

$$A[f](\mathbf{x}) = \int_0^T f(s)\mathbf{x}(s)ds.$$

The model can be easily extended to deal with $\mathbf{Y}$ taking label values such as in a classification problem as we will show in the numerical studies.

Due to space constraints, we provide examples in the FLR setting. In the supplementary material, we demonstrate how the Algorithms presented in Section 4.2 also in deconvolution problems.

4 Theoretical Results and Algorithms

In this paper, we consider the following set of assumptions.

Assumption 4.1.

1. $A : L^2(\mathbb{W}) \longrightarrow L^2(\mathbb{X})$ is a linear, bounded operator;
2. ℓ is a convex and C^2 function in its second argument;
3. There exists $\theta_0 > 0$ such that, for all $f, g \in L^2(\mathbb{W})$,
$$\sup_{|\theta| \leq \theta_0} \mathbb{E}[\partial_{22}\ell(Y, A[f](\mathbf{X}) + \theta A[g](\mathbf{X}))(A[g](\mathbf{X}))^2] < \infty;$$
4. $f^\circ \in \arg\min_{f \in \mathcal{F}} \mathcal{R}_A(f)$ and $\mathcal{R}_A(f^\circ) > -\infty$;
5. $\sup_{f, f' \in \mathcal{F}} \|f - f'\|_{L^2(\mathbb{W})} = D < \infty$.

Assumption 1 is our strongest one, since it imposes that our operator is linear and bounded. Nevertheless, linear SIPs encompass a wide class of problems of practical and theoretical interest for engineering, statistics and computer science communities among others, a few of them presented in Section 3. Moreover, the nonlinear case could be similarly studied with more cumbersome notation and assumptions. Assumption 2 is standard for gradient based algorithms and is commonly assumed in many learning problems. Assumption 3 is a mild integrability condition of the loss function commonly satisfied in many practical situations. For instance, in the squared loss case, this assumption becomes $\mathbb{E}[(A[g](\mathbf{X}))^2] < \infty$, which is automatically satisfied since $A[g] \in L^2(\mathbb{X})$. Assumption 4 is needed so the problem we analyze indeed have a solution. Assumption 5 is stating that the diameter of the set $\mathcal{F}$ is finite.

One should notice that our set of assumptions does include the class of ill-posed (linear) inverse problems since we do not need to assume that A is bijective. If that were the case, it is known that then A would have a bounded inverse, and then, the IP would not be ill-posed.

In the next sections we provide our theoretical results. Instead of following the common approach of minimizing the Empirical Risk Minimization problem, we show how to compute stochastic gradients in order to control directly for the excess risk (3) both in expectation and in probability.

4.1 Preliminaries

Our first result allows us to compute the gradient of the populational risk at a given functional parameter f. Before we present it, note that, by linearity, $A : L^2(\mathbb{W}) \to L^2(\mathbb{X})$ is differentiable and, for every $f, g \in L^2(\mathbb{W})$, we have that the directional derivative of $A[f]$ in the direction g is given by

$$DA[f](g) = \lim_{\delta \to 0} \frac{1}{\delta} (A[f + \delta g] - A[f]) = A[g].$$

Note that the directional derivative does not dependent on the point f that we are evaluating the gradient. Let A^* denote the adjoint operator of A defined as the linear and bounded operator $A^* : L^2(\mathbb{X}) \longrightarrow L^2(\mathbb{W})$ such that[1]

$$\langle A[f], g \rangle_{L^2(\mathbb{X})} = \langle f, A^*[g] \rangle_{L^2(\mathbb{W})}.$$

The following lemma holds true:

[1]The adjoint of a linear, bounded operator always exists.

Lemma 4.2. *Under 1, 2 and 3 of Assumption 4.1 we have that*

$$\nabla \mathcal{R}_A(f) = A^*[\phi] \ \in L^2(\mathbb{W}),$$

where $\phi(\mathbf{x}) = \mathbb{E}[\partial_2 \ell(\mathbf{Y}, A[f](\mathbf{x})) \mid \mathbf{X} = \mathbf{x}]$.

Proof. Firstly, define, for fixed $(\mathbf{X}, \mathbf{Y})$,

$$\psi(\delta) = \ell(\mathbf{Y}, A[f + \delta g](\mathbf{X})) = \ell(\mathbf{Y}, A[f](\mathbf{X}) + \delta A[g](\mathbf{X})).$$

Then, we get the directional derivative of the risk function in direction g by applying the Taylor formula for ψ as a function of δ around $\delta = 0$:

$$\begin{aligned} D\mathcal{R}_A(f)(g) &= \lim_{\delta \to 0} \frac{1}{\delta} \left(\mathcal{R}_A(f + \delta g) - \mathcal{R}_A(f) \right) \\ &= \lim_{\delta \to 0} \frac{1}{\delta} \mathbb{E}[\psi(\delta) - \psi(0)] \\ &= \lim_{\delta \to 0} \mathbb{E}\Bigg[\frac{1}{\delta} \Bigg(\delta \partial_2 \ell(\mathbf{Y}, A[f](\mathbf{X})) A[g](\mathbf{X}) \\ &\quad + \tfrac{1}{2} \delta^2 \partial_{22} \ell(\mathbf{Y}, A[f](\mathbf{X}) + \theta A[g](\mathbf{X})) (A[g](\mathbf{X}))^2 \Bigg) \Bigg], \end{aligned}$$

where θ comes from the Taylor formula and it is between $-\theta_0$ and θ_0, for some fixed $\theta_0 > 0$. Hence, by Assumption 3, we find

$$D\mathcal{R}_A(f)(g) = \mathbb{E}[\partial_2 \ell(\mathbf{Y}, A[f](\mathbf{X})) A[g](\mathbf{X})].$$

By the definition of ϕ and by conditioning in $\mathbf{X}$, we find

$$D\mathcal{R}_A(f)(g) = \mathbb{E}[\phi(\mathbf{X}) A[g](\mathbf{X})] = \langle \phi, A[g] \rangle_{L^2(\mathbb{X})} = \langle A^*[\phi], g \rangle_{L^2(\mathbb{W})}.$$

Finally, we get that the descent direction $\nabla \mathcal{R}_A(f)$ is given by $A^*[\phi] \ \in L^2(\mathbb{W})$. □

By the linearity and boundedness of A^*, we find the following very useful representation for the gradient of $\mathcal{R}_A$:

Lemma 4.3. *Consider the case* $d = 1$ *for simplicity of notation. For each* $\mathbf{w} \in \mathbb{W}$ *fixed, there exists a kernel* $\Phi(\cdot; \mathbf{w}) : \mathbb{X} \longrightarrow \mathbb{R}$ *such that*

$$A^*[g](\mathbf{w}) = \mathbb{E}[\Phi(\mathbf{X}; \mathbf{w}) g(\mathbf{X})].$$

Proof. First, notice that $\varphi_{\mathbf{w}}(g) = A^*[g](\mathbf{w})$ is an element of the dual of $L^2(\mathbb{X})$. Hence, by the Riesz Representation Theorem, there exists a kernel $\Phi(\cdot; \mathbf{w}) : \mathbb{X} \longrightarrow \mathbb{R}$ such that, for all $g \in L^2(\mathbb{X})$,

$$A^*[g](\mathbf{w}) = \varphi_{\mathbf{w}}(g) = \langle \Phi(\cdot; \mathbf{w}), g \rangle_{L^2(\mathbb{X})} = \mathbb{E}[\Phi(\mathbf{X}; \mathbf{w}) g(\mathbf{X})],$$

as desired. □

Corollary 4.4. *If* ℓ *is the squared loss function* $\ell(\mathbf{y}, \mathbf{y}') = \frac{1}{2} \|\mathbf{y} - \mathbf{y}'\|^2$*, then the gradient of the risk function with respect to* f *is given by*

$$\nabla \mathcal{R}_A(f) = -A^*[\overline{\mathbf{Y}} - A[f]] = \mathbb{E}[\Phi(\mathbf{X}; \mathbf{w})(A[f](\mathbf{X})) - \overline{\mathbf{Y}}(\mathbf{X})],$$

where $\overline{\mathbf{Y}}(\mathbf{x}) = \mathbb{E}[\mathbf{Y} \mid \mathbf{X} = \mathbf{x}]$.

Because of the results above, it is possible to construct an unbiased estimator for the gradient for the risk function for any f. In fact, considering the case $d = 1$, for a given sample $(\mathbf{x}, \mathbf{y})$ and a fixed function f, we define, for any $\mathbf{w} \in \mathbb{W}$,

$$u_f(\mathbf{w}; \mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}; \mathbf{w})(A[f](\mathbf{x}) - \mathbf{y}). \tag{4}$$

Therefore, conditioning in $\mathbf{X}$, we find

$$\begin{aligned} \mathbb{E}[u_f(\mathbf{w}; \mathbf{X}, \mathbf{Y})] &= \mathbb{E}[\Phi(\mathbf{X}; \mathbf{w})(A[f](\mathbf{X}) - \mathbf{Y})] \\ &= \mathbb{E}[\mathbb{E}[\Phi(\mathbf{X}; \mathbf{w})(A[f](\mathbf{X}) - \mathbf{Y}) \mid \mathbf{X}]] \\ &= \mathbb{E}[\Phi(\mathbf{X}; \mathbf{w})(A[f](\mathbf{X}) - \mathbb{E}[\mathbf{Y} \mid \mathbf{X}])] \\ &= \mathbb{E}[\Phi(\mathbf{X}; \mathbf{w})(A[f](\mathbf{X}) - \overline{\mathbf{Y}}(\mathbf{X}))] \\ &= \nabla \mathcal{R}_A(f)(\mathbf{w}). \end{aligned}$$

The main benefit is that with only one observation we are able to compute an unbiased estimator for the gradient of the risk function under the true distribution.

4.2 Proposed Algorithms

Inspired by Lemma 4.2, we propose the following SGD algorithm for SIP problems that we called SGD-SIP: given an initial guess f_0, for each step i, we compute, following Eq. (4), an unbiased estimator u_i for the gradient of the loss function. Next, we update an accumulated functional parameter by taking a stochastic gradient step in the direction of u_i with step size α_i. In the last step, we average all the accumulated gradient steps in the same spirit as Polyak and Juditsky [1992]. The choice of the step size needs to satisfy two criteria: $\sum_{i=1}^{n} \alpha_i$ sublinear in n, and $n\alpha_n \to \infty$ as $n \to +\infty$. We formally justify those desired properties in Theorem 4.5.

Algorithm 1: SGD-SIP

input : sample $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$, operator A, initial guess f_0

output : $\hat{f}_n$

$\hat{g}_0 = f_0$;

for $1 \leq i \leq n$ **do**

  Compute $u_i(\mathbf{w}) = \Phi(\mathbf{x}_i; \mathbf{w})(A[\hat{g}_{i-1}](\mathbf{x}_i) - \mathbf{y}_i)$;

  $\hat{g}_i = \hat{g}_{i-1} - \alpha_i u_i$;

end

Set $\hat{f}_n = \frac{1}{n}\sum_{i=1}^{n} \hat{g}_i$;

Algorithm 1 uses only one sample at a time in order to estimate the gradient of the true risk function. In order to preserve this property, we make the number of iterations equal to the sample size; it cannot be larger. This connects with the stopping rules in iterative algorithms in IP.

Algorithm 1 has a limitation common to many approaches to Inverse Problems: one cannot hope to compute u_i for every possible $\mathbf{w}_i$ and some discretization of the operator A is needed, see Kaipio and Somersalo [2007]. Since the SGD-SIP algorithm only computes the stochastic gradient in the points of discretization, it risks overfitting the data and provides non-smooth estimators. Next, we motivate Algorithm 2 in order to overcome the discretization problem by leveraging machine learning methods.

Consider that the space $\mathbb{W}$ was discretized in a grid of size n_w. In order to fully estimate the function $\hat{f}_n(\mathbf{w})$ for every $\mathbf{w} \in \mathbb{W}$, we consider a hypothesis class $\mathcal{H}$ and, in each step, we fit a function $\hat{h}_i^\star$ on the stochastic gradient u_i in the discretized grid of $\mathbb{W}$. Note that in this case, $\mathcal{F}$ will be given by the linear span of the class $\mathcal{H}$. Each of these functions $h_i^\star$ can be seen as a base-learner in the same spirit of Boosting estimators, widely used in standard regression problem in the context of SIP Mason et al. [1999], Friedman [2001]. Next we present our algorithm ML-SGD.

Algorithm 2: ML-SGD

input : sample $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{n}$, discretization $\{\mathbf{w}_j\}_{j=1}^{n_z}$ of $\mathbb{W}$, operator A, initial guess f_0

output : $\hat{f}_n$

$\hat{g}_0 = f_0$;

for $1 \leq i \leq n$ **do**

  for $1 \leq j \leq n_w$ **do**

    Compute $u_i(\mathbf{w}_j) = \Phi(\mathbf{x}_i; \mathbf{w}_j)(A[\hat{g}_{i-1}](\mathbf{x}_i) - \mathbf{y}_i)$;

  end

  $h_i^\star \in \arg\min_{h \in \mathcal{H}} \sum_{j=1}^{n_z} (u_i(\mathbf{w}_j) - h(\mathbf{w}_j))^2$;

  $\hat{g}_i = \hat{g}_{i-1} - \alpha_i h_i^\star$;

end

Set $\hat{f}_n = \frac{1}{n}\sum_{i=1}^{n} \hat{g}_i$;

The goal of ML-SGD is twofold. First, it allows us to interpolate the function $h_j^\star$ to points $\mathbf{w}$ not used in the discretization grid. Second, the ML procedure smooths the noise in each gradient step calculation leading to smoother approximations that helps avoiding over-fitting. We show in Section 5 and in the supplementary material the benefits of such an approximation when estimating the functional parameter f° in both simulated and empirical examples.

4.3 Main Result

Our main result is a finite sample bound for the expected excess risk of Algorithm 1. The result also extends to Algorithm 2 in the case where the base learner are also unbiased estimators.

Theorem 4.5. *Under Assumption 4.1 and if the kernel Φ satisfies $C = \sup_{\mathbf{x}\in\mathbb{X}} \|\Phi(\mathbf{x},\cdot)\|^2 < +\infty$[2], we have the following performance guarantee for Algorithm 1:*

$$\mathbb{E}\left[\mathcal{R}_A(\hat{f}_n) - \inf_{f\in\mathcal{F}} \mathcal{R}_A(f)\right] \leq \frac{D^2}{2n\alpha_n} + \frac{M(A,\mathcal{F})}{n}\sum_{i=1}^{n}\alpha_i,$$

where $M(A,\mathcal{F}) = C(\mathbb{E}[|\mathbf{Y}|^2] + \|A\|^2 D^2) < \infty$.

The proof of the theorem is provided in the supplementary material. Theorem 4.5 implies that if we pick the decreasing sequence $\{\alpha_i\}_{i=1}^n$ so that $n\alpha_n \to \infty$ (α_n cannot decrease too fast) but fast enough so that $\frac{1}{n}\sum_{i=1}^n \alpha_i \to 0$, then we get the convergence result. For instance, one could take $\alpha_i = \eta/\sqrt{i}$ for some fixed number η normally taken to be in $(0,1)$. In this case, the excess risk decreases in expectation with rate $O(1/\sqrt{n})$.

Theorem 4.5 also implies that the excess risk converges to zero in probability. For $\alpha_i = \eta/\sqrt{i}$ it is straightforward to check that

$$\limsup_{n\to+\infty} \mathbb{P}\left(\mathcal{R}_A(\hat{f}_n) - \inf_{f\in\mathcal{F}} \mathcal{R}_A(f) > 0\right) = 0.$$

Finite sample bounds with high probability can also be provided under stronger assumptions about the stochastic gradients. See for instance Nemirovski et al. [2009].

5 Functional Linear Regression: numerical results

In this section, we provide two applications of the Functional Linear Regression problem. We first demonstrate the performance of both algorithms in simulated data and next we provide an example for generalized linear models, when predicting the presence or not of Multiple Sclerosis (MS) after receiving as input a tract profile of corpus callosum (CCA) obtained by Diffusion Tensor Imaging.

As we have seen in Section 3, the operator in the FLR case is given by

$$A[f](\mathbf{x}) = \int_0^T f(s)\mathbf{x}(s)ds. \tag{5}$$

Remember that in this example we are denoting $\mathbf{w}$ by s. Hence

$$\begin{aligned}\langle A[f], g\rangle_{L^2(\mathbb{X})} &= \mathbb{E}[A[f](\mathbf{X})g(\mathbf{X})] \\ &= \mathbb{E}\left[\int_0^T f(s)\mathbf{X}(s)ds\, g(\mathbf{X})\right] \\ &= \int_0^T f(s)\mathbb{E}\left[\mathbf{X}(s)\, g(\mathbf{X})\right] ds = \langle f, A^*[g]\rangle_{L^2(\mathbb{W})}\end{aligned}$$

where $A^* : L^2(\mathbb{X}) \longrightarrow L^2(\mathbb{W})$ is given by

$$A^*[g](s) = \mathbb{E}\left[\mathbf{X}(s)\, g(\mathbf{X})\right].$$

Therefore, we have $\Phi(\mathbf{x}; s) = \mathbf{x}(s)$, and using the squared loss, we find, as in Eq. (4),

$$u_i(s) = \mathbf{x}_i(s)(A[\hat{g}_{i-1}](\mathbf{x}_i) - \mathbf{y}_i).$$

[2]This assumption is satisfied for all the examples analyzed in this paper.

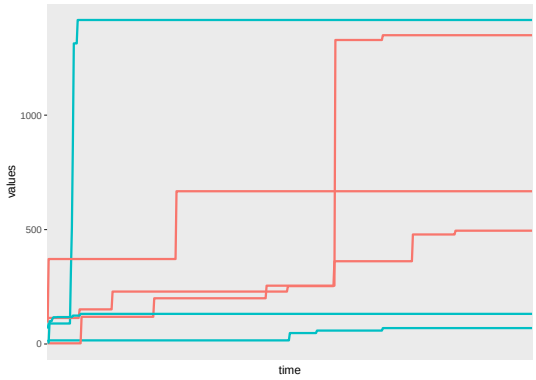

Figure 1: Example of cumulative credits for six different addresses across 501 data points. In red, addresses associated with criminal activity, in blue, addresses associated with noncriminal activities.

5.1 Synthetic Data

We will consider the simulation study presented in González-Manteiga and Martínez-Calvo [2011]. Specifically, we set $\mathbb{W} = [0, 1]$, $f^{\circ}(z) = \sin(4\pi z)$, and $\mathbf{X}$ simulated accordingly a Brownian motion in $[0, 1]$. We also consider a noise-signal ratio of 0.2. We generate 100 samples of $\mathbf{X}$ and $\mathbf{Y}$ with the integral defining the operator A approximated by a finite sum of 1000 points in $[0, 1]$. For the observed data used in the algorithm procedure, we consider a coarser grid where and each functional sample is observed at only 100 equally-spaced times. For the ML-SGD algorithm, we used smoothing splines as base learners. We compared the results with a Functional Penalized Linear Regression (FPLR) with B-splines and cross-validation to select the number of basis expansion. In order to fit the PFLR model, we used the package *refund* Goldsmith et al. [2021] available in R. In the supplementary material we can see the estimated observation $A[\hat{f}_n](\mathbf{x}_i)$ against the true values $A[f^{\circ}](\mathbf{x}_i)$ using both the SGD-SIP and the ML-SGD algorithm essentially recovers the true function, with a noisier fit for the SGD-SIP. The benchmark algorithms also recovers the true function. We refer the reader to the supplementary material for more details about the simulation.

5.2 Real Data Application

Next we consider a classification problem in the FLR setting. The data set contains 3000 examples of bitcoin wallets (address) and the respective cumulative credit in each wallet. Each address contain 501 equally spaced observations from April 2011 and April 2017. For each address, we also have a category describing the category of the address. In Table 5.2 we present a summary of the data. We refer the reader to the supplementary material for more information about the data set used that we make available online. Here we use the cumulative credit curve at each point in time as the explanatory variables $\mathbf{X} \in \mathbb{X} = L^2([0, 1])$ and $Y \in \{-1, 1\}$ as the predicted outcome for the indicator variable that the category is *darknet* (addresses associated with ilegal activities). We propose the following model with a the log-likelihood of the negative binomial as the loss function:

$$\log \frac{P(Y = 1|\mathbf{X})}{P(Y = -1|\mathbf{X})} = \int_0^T f(s)\mathbf{X}(s)ds.$$

	category	obs	mean_credit_begin	mean_credit_end
1	Darknet Marketplace	1512	234.41	1264.86
2	Exchanges	379	673.19	14026.14
3	Gambling	390	86.83	2369.12
4	Pools	374	1211.11	15334.20
5	Services/others	345	242.39	4094.89

Table 1: Summary information for the bitcoin wallet observations.

In Figure 1 we have the cumulative credit for 3 legal and 3 illegal accounts. We compare the SGD-SIP and ML-SGD algorithm with PFLR methods. For the ML-SGD algorithm, we use two types of base

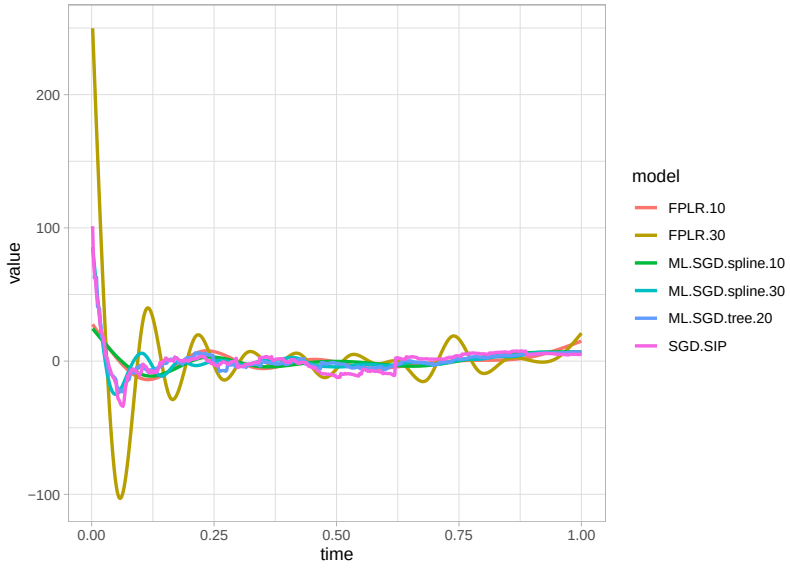

Figure 2: Fitted curves for each of the methods described in Section 5.2.

learner, regression trees and cubic splines. The step sizes are taken to be equal of the form $O(1/\sqrt{i})$, where $i = 1, \cdots, n$ is the current step of the algorithm and n is the total number of steps/sample. For the PFLR algorithm, we use cubic splines with different degrees of freedom and quadratic penalty term. We highlight that those choices of splines and penalty term are widely used in the literature. In Table 5.2 we provide 3-fold cross validation for the accuracy and kappa metrics. The SGD-SIP Algorithm achieved the best average performance in terms of accuracy. The same performance is achieved by the Functional PLR with cubic splines with number of knots equal to 30 and penalization for the derivative of the estimate. The ML-SGD algorithm with smooth splines with 30 degrees of freedom also achieved similar performance with a smoother estimator. In Figure 2 we show the final function fit for each of the methods listed in Table 5.2. We refer the reader to the supplementary material for results under different choices of step size, number of knots and base functions for the PFLR model, other metrics and confusion matrices. In order to fit the PFLR mode, we used the package *refund* Goldsmith et al. [2021] available in R.

	fold_1	fold_2	fold_3	avg_accuracy
ML-SGD-spline(k = 20)	0.78	0.79	0.78	0.79
ML-SGD-spline(k = 10)	0.74	0.74	0.70	0.73
ML-SGD-tree(depth = 20)	0.79	0.78	0.77	0.78
SGD-SIP	0.80	0.80	0.80	0.80
FPLR(k = 10)	0.75	0.74	0.72	0.74
FPLR(k = 20)	0.82	0.79	0.80	0.80

Table 2: Results for three fold cross-validation.

6 Conclusion

In this work, we provided a novel numerical method to solve SIP based on stochastic gradients with theoretical guarantees for the excess risk. Moreover, we have shown how one can improve algorithmic performance by estimating base-learners for each stochastic gradient in the same spirit as boosting algorithms. Our "framework" can be applied in a variety of settings ranging from deconvolution problems to FLR problems and others. We demonstrate the performance of our method with numerical studies and also with a real world application data and comparing with widely used techniques in the FLR setting.

References

Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002.

Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent in function space. In *Proc. NIPS*, volume 12, pages 512–518, 1999.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

V.N. Sudakovand and L.A. Khalfin. Statistical approach to ill-posed problems in mathematical geophysics. *Sov. Math.—Dokl.*, 157, 1964.

J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*. Springer, 2004.

Steven N Evans and Philip B Stark. Inverse problems as statistics. *Inverse problems*, 18(4):R55, 2002.

Laurent Cavalier. Nonparametric statistical inverse problems. *Inverse Problems*, 24(3):034004, 2008.

Luis Tenorio. Statistical regularization of inverse problems. *SIAM review*, 43(2):347–366, 2001.

Nicolai Bissantz, Thorsten Hohage, Axel Munk, and Frits Ruymgaart. Convergence rates of general regularization methods for statistical inverse problems and applications. *SIAM Journal on Numerical Analysis*, 45(6):2610–2636, 2007.

Nicolai Bissantz, Thorsten Hohage, and Axel Munk. Consistency and rates of convergence of nonlinear tikhonov regularization with random noise. *Inverse Problems*, 20(6):1773, 2004.

Nicolai Bissantz and Hajo Holzmann. Statistical inference for inverse problems. *Inverse Problems*, 24(3): 034009, 2008.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. NETT: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, 2020.

Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

Yanna Bai, Wei Chen, Jie Chen, and Weisi Guo. Deep learning methods for solving linear inverse problems: Research directions and paradigms. *Signal Processing*, page 107729, 2020.

Bangti Jin, Zehui Zhou, and Jun Zou. On the saturation phenomenon of stochastic gradient descent for linear inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 9(4):1553–1588, 2021.

Junqi Tang, Karen Egiazarian, and Mike Davies. The limitation and practical acceleration of stochastic gradient algorithms in inverse problems. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7680–7684. IEEE, 2019.

Bangti Jin, Zehui Zhou, and Jun Zou. On the convergence of stochastic gradient descent for nonlinear ill-posed problems. *SIAM Journal on Optimization*, 30(2):1421–1450, 2020.

T Tony Cai and Peter Hall. Prediction in functional linear regression. *The Annals of Statistics*, 34(5):2159–2179, 2006.

Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional linear regression analysis for longitudinal data. *The Annals of Statistics*, pages 2873–2903, 2005.

Peter Hall and Joel L Horowitz. Methodology and convergence rates for functional linear regression. *The Annals of Statistics*, 35(1):70–91, 2007.

Jeff Goldsmith, Jennifer Bobb, Ciprian M Crainiceanu, Brian Caffo, and Daniel Reich. Penalized functional regression. *Journal of computational and graphical statistics*, 20(4):830–851, 2011.

Jeffrey S Morris. Functional regression. *Annual Review of Statistics and Its Application*, 2:321–359, 2015.

Gareth M James, Jing Wang, and Ji Zhu. Functional linear regression that's interpretable. *The Annals of Statistics*, 37(5A):2083–2108, 2009.

Yingying Fan, Gareth M James, and Peter Radchenko. Functional additive regression. *The Annals of Statistics*, 43(5):2296–2325, 2015.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Jari Kaipio and Erkki Somersalo. Statistical inverse problems: discretization, model reduction and inverse crimes. *Journal of computational and applied mathematics*, 198(2):493–504, 2007.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Wenceslao González-Manteiga and Adela Martínez-Calvo. Bootstrap in functional linear regression. *Journal of Statistical Planning and Inference*, 141(1):453–461, 2011.

Jeff Goldsmith, Fabian Scheipl, Lei Huang, Julia Wrobel, Chongzhi Di, Jonathan Gellar, Jaroslaw Harezlak, Mathew W. McLean, Bruce Swihart, Luo Xiao, Ciprian Crainiceanu, and Philip T. Reiss. *refund: Regression with Functional Data*, 2021. URL `https://CRAN.R-project.org/package=refund`. R package version 0.1-24.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We did both in the problem formulation and in the numerical studies.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not believe that the methods proposed here could cause a negative impact in the society of any kind and this discussion was not necessary.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Please refer to 4
   (b) Did you include complete proofs of all theoretical results? [Yes] Proofs are in the main text and in the supplementary material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The codes and data used are available online and in the supplementary material. Please, refer to the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to Section 5 and the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the main text we reported results with 3folds cross-validation. In the online supplementary material we report results with synthetic data with error bars for the seed and additional validation.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] The algorithms are simple and all the methods run in standard computers. Cloud computer and GPUs were not used.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Please, see Section 5
   (b) Did you mention the license of the assets? [Yes] Please, see Section 5
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] The existing assets are already available online.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The assets used are already available for public use.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data is not identifiable or offensive. Please, refer to the description provided in Section 5

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Functional Gradient for the Deconvolution Problem

Remember that the operator A is given by

$$A[f](\mathbf{x}) = \int_{\mathbb{W}} k(\mathbf{x} - \mathbf{w}) f(\mathbf{w}) d\mu(\mathbf{w}). \tag{6}$$

Hence,

$$\begin{aligned}
\langle A[f], g \rangle_{L^2(\mathbb{X})} &= \mathbb{E}[A[f](\mathbf{X}) g(\mathbf{X})] \\
&= \mathbb{E}\left[\left(\int_{\mathbb{W}} k(\mathbf{X} - \mathbf{w}) f(\mathbf{w}) d\mu(\mathbf{w}) \right) g(\mathbf{X}) \right] \\
&= \int_{\mathbb{W}} \mathbb{E}[k(\mathbf{X} - \mathbf{w}) g(\mathbf{X})] f(\mathbf{w}) d\mu(\mathbf{w}) \\
&= \langle f, A^*[g] \rangle_{L^2(\mathbb{W})},
\end{aligned}$$

where

$$A^*[g](\mathbf{w}) = \mathbb{E}[k(\mathbf{X} - \mathbf{w}) g(\mathbf{X})].$$

Therefore, we have $\Phi(\mathbf{x}; \mathbf{w}) = k(\mathbf{x} - \mathbf{w})$, and using the squared loss, we find, as in Eq. (4),

$$u_i(\mathbf{w}) = k(\mathbf{x}_i - \mathbf{w})(A[\hat{g}_{i-1}](\mathbf{x}_i) - \mathbf{y}_i).$$

We highlight here the need to use each observation only once in order to compute the stochastic gradient so we can have precisely n steps for the SGD-SIP/ML-SGD algorithm. In this case, the samples can be used to provide unbiased estimators for the gradient of the risk function under the populational distribution.

B Numerical Studies: Synthetic Data

In this section we present the numerical studies of our proposed algorithms with standard benchmarks from the literature. We studied both the Functional Linear Regression problem and the Deconvolution problem. We remind the reader that the same framework can also be used to solve different types of inverse problems under a statistical framework, such as ODEs and PDEs.

B.1 Functional Linear Regression

Recall 5 where for the FLR problem our goal is to recover f° when we have access to observations of the form

$$Y = A[f^\circ](X) + \epsilon,$$

where the operator A is given by

$$A[f](\mathbf{x}) = \int_0^T f(s)\mathbf{x}(s) ds. \tag{7}$$

Recall the data generating process described in 5.1. We set $\mathbb{W} = [0, 1]$, $f^\circ(z) = \sin(4\pi z)$, and $\mathbf{X}$ simulated accordingly a Brownian motion in $[0, 1]$. We also consider a noise-signal ratio of 0.2. We generate 3000 samples of $\mathbf{X}$ and $\mathbf{Y}$ with the integral defining the operator A approximated by a finite sum of 1000 points in $[0, 1]$. For the observed data used in the algorithm procedure, we consider a coarser grid where and each functional sample is observed at only 100 equally-spaced times. For the ML-SGD algorithm, we used smoothing splines as base learners. We compare our algorithm with the Landweber method, which is a Gradient Descent version for deterministic Inverse Problems and Functional Penalized Linear Regression (FPLR). For the ML-SGD,SGD and Landweber method, the step sizes were taken fixed to be $(100/\sqrt{(}n))$ (which satisfy the requirements discussed after 4.5).

Table 3: MSE results for three fold cross-validation.

	fold_1	fold_2	fold_3
ML-SGD-spline(k = 20, eta = 1.5)	1.8E-05	1.6E-05	1.5E-05
ML-SGD-spline(k = 20, eta = 1)	2.8E-05	3.0E-05	2.3E-05
ML-SGD-tree(depth = 10)	1.5E-03	1.7E-03	1.3E-03
SGD	1.6E-03	1.6E-03	1.1E-03
Landweber	1.6E-03	1.6E-03	1.1E-03
FPLR(k = 5)	9.3E-03	9.5E-03	9.5E-03
FPLR(k = 10)	9.3E-03	9.5E-03	9.5E-03
FPLR(k = 15)	9.3E-03	9.5E-03	9.5E-03

In **??** and **??** we present the Mean Absolute Error and Mean Square Error for out-of-sample predictions of the outcome Y using three-folds cross valudation. In **??** we have an example of a plot for the ML-SGD method with smoothing splines with 15 degrees of freedom. We also added an estimator for the SGD algorithm and FPLR with 15 knots. All the procedures essentially recovers the true f° with a slightly worse (and noiser) estimator for the SGD.

Table 4: MAE results for three fold cross-validation.

	fold_1	fold_2	fold_3
ML-SGD-spline(k = 20, eta = 1.5)	3.4E-03	3.2E-03	3.1E-03
ML-SGD-spline(k = 20, eta = 1)	4.2E-03	4.4E-03	3.9E-03
ML-SGD-tree(depth = 10)	3.1E-02	3.3E-02	2.8E-02
SGD	3.2E-02	3.2E-02	2.6E-02
Landweber	3.2E-02	3.2E-02	2.6E-02
FPLR(k = 5)	7.7E-02	7.8E-02	7.8E-02
FPLR(k = 10)	7.7E-02	7.8E-02	7.8E-02
FPLR(k = 15)	7.7E-02	7.8E-02	7.8E-02

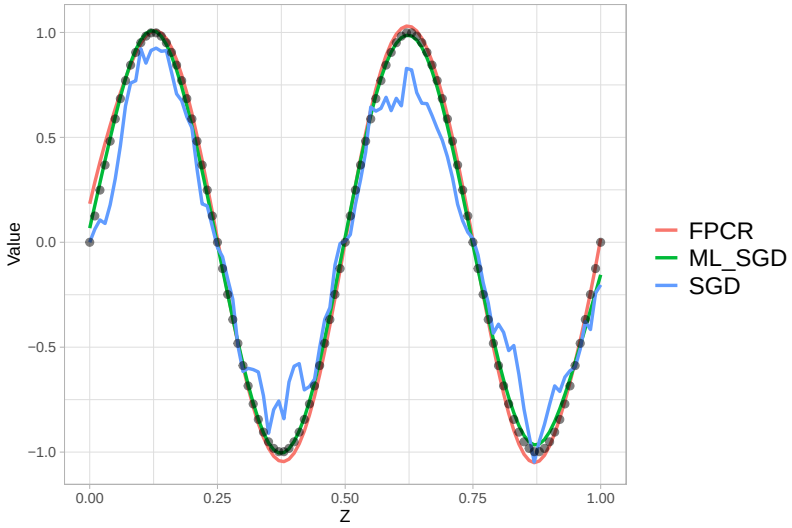

Figure 3: Functional parameter estimated with SGD-SIP, ML-SGD, FPLR and the respective true function.

B.2 Deconvolution

For the deconvolution problem we examine the following numerical exercise. We take two choices of functional parameters for Eq. (6), as a peak function:

$$f(z) = e^{-w^2}. \tag{8}$$

We consider the kernel to be given by

$$k(z) = 1_{\{z \geq 0\}}$$

and the following parameters for the data generating process. First we discretize the space $\mathbb{W} = [-10, 10]$ with increments $h = 0.01$. We use the same for the space $\mathbb{X} = [-10, 10]$. Next, we use the discretized space to generate the true values $A[f]$ where we approximate the integral by a finite sum. The second step is to generate the random observations. For that, we consider a coarser grid for $\mathbb{X}$, with grid $h_{obs} = 0.1$, i.e. 10 times less information than the simulation used to generate the true observations. This reproduces the fact that in practice one cannot hope to observe the functional data over all points. Moreover, when computing the operator A in our algorithm, we again consider a coarser grid for $\mathbb{W}$, with grid $h_{obs} = 0.1$. We then add iid noise terms $N(0, 2)$ to the observations $A[f]$ collected from the coarse grid. For the ML-SGD algorithm (Algorithm 2), we used smooth splines with 5 degrees of freedom as $\mathcal{H}$ in order to estimate the stochastic gradients. We compare our algorithms with the well-known landweber iteration, which resambles the standard Gradient Descent algorithm when ignoring noise and using all the samples available in all the iterations. We start with $f_0(z) = 0$ in all the algorithms.

In Figure 4 we can see that ML-SGD outputs a smooth estimator for the functional parameter f° while the other two methods tends to overfit the data. The situation is worse for the Landweber method, which requires a better understanding in when stopping the procedure due to the reuse of all the data available in every iteration.

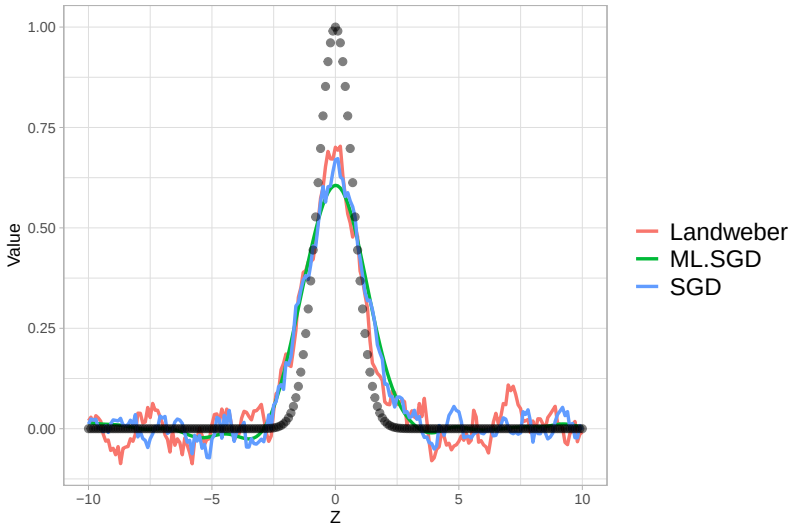

Figure 4: Deconvolution: COmparison with Landweber. True values are shown as the black points.

C Proof of Theorem 4.5

Proof. First, it is straightforward to check that $\mathcal{R}_A$ is convex in $\mathcal{F}$: if $f, g \in \mathcal{F}$ and $\lambda \in [0,1]$, then

$$\begin{aligned}\mathcal{R}_A(\lambda f + (1-\lambda)g) &= \mathbb{E}[\ell(\mathbf{Y}, A[\lambda f + (1-\lambda)g](\mathbf{X}))] \\ &= \mathbb{E}[\ell(\mathbf{Y}, \lambda A[f](\mathbf{X}) + (1-\lambda)A[g](\mathbf{X}))] \\ &\leq \mathbb{E}[\lambda \ell(\mathbf{Y}, A[f](\mathbf{X}))] + \mathbb{E}[(1-\lambda)\ell(\mathbf{Y}, A[g](\mathbf{X}))] \\ &= \lambda \mathcal{R}_A(f) + (1-\lambda)\mathcal{R}_A(g).\end{aligned}$$

For simplicity of notation we will denote the norm and inner product in $L^2(\mathbb{W})$ by $\|\cdot\|$ and $\langle \cdot, \cdot \rangle$. Moreover, we assume $d = 1$. The multivariate case follows similarly.

By the Algorithm 1 procedure, we have that

$$\begin{aligned}&\frac{1}{2}\|\hat{g}_i - f^\circ\|^2 = \frac{1}{2}\|\hat{g}_{i-1} - \alpha_i u_i - f^\circ\|^2 \\ &= \frac{1}{2}\|\hat{g}_{i-1} - f^\circ\|^2 - \alpha_i \langle u_i, \hat{g}_{i-1} - f^\circ \rangle + \frac{\alpha_i^2}{2}\|u_i\|^2 \\ &= \frac{1}{2}\|\hat{g}_{i-1} - f^\circ\|^2 - \alpha_i \langle u_i - \nabla\mathcal{R}_A(\hat{g}_{i-1}), \hat{g}_{i-1} - f^\circ \rangle + \frac{\alpha_i^2}{2}\|u_i\|^2 - \alpha_i \langle \nabla\mathcal{R}_A(\hat{g}_{i-1}), \hat{g}_{i-1} - f^\circ \rangle \\ &\leq \frac{1}{2}\|\hat{g}_{i-1} - f^\circ\|^2 - \alpha_i \langle u_i - \nabla\mathcal{R}_A(\hat{g}_{i-1}), \hat{g}_{i-1} - f^\circ \rangle + \frac{\alpha_i^2}{2}\|u_i\|^2 - \alpha_i (\mathcal{R}_A(\hat{g}_{i-1}) - \mathcal{R}_A(f^\circ)),\end{aligned}$$

where the last inequality follows from convexity of the loss function (Assumption 2). Rearranging terms we get

$$\mathcal{R}_A(\hat{g}_{i-1}) - \mathcal{R}_A(f^\circ) \leq \frac{1}{2\alpha_i}\left(\|\hat{g}_{i-1} - f^\circ\|^2 - \|\hat{g}_i - f^\circ\|\right) + \frac{\alpha_i}{2}\|u_i\|^2 - \langle u_i - \nabla\mathcal{R}_A(\hat{g}_{i-1}), \hat{g}_{i-1} - f^\circ \rangle.$$

Summing over i leads to

$$\begin{aligned}\sum_{i=1}^n \mathcal{R}_A(\hat{g}_{i-1}) - \mathcal{R}_A(f^\circ) \leq &\sum_{i=1}^n \frac{1}{2\alpha_i}\left(\|\hat{g}_{i-1} - f^\circ\|^2 - \|\hat{g}_i - f^\circ\|^2\right) \\ &+ \sum_{i=1}^n \frac{\alpha_i}{2}\|u_i\|^2 \\ &- \sum_{i=1}^n \langle u_i - \nabla\mathcal{R}_A(\hat{g}_{i-1}), \hat{g}_{i-1} - f^\circ \rangle.\end{aligned}$$

For the first term, by Assumption 5, we find

$$\sum_{i=1}^{n} \frac{1}{2\alpha_i} \left(\|\hat{g}_{i-1} - f^\circ\|^2 - \|\hat{g}_i - f^\circ\|^2 \right) = \sum_{i=2}^{n} \left(\frac{1}{2\alpha_i} - \frac{1}{2\alpha_{i-1}} \right) \|\hat{g}_{i-1} - f^\circ\|^2 + \frac{1}{2\alpha_1} \|\hat{g}_0 - f^\circ\|^2 - \frac{1}{2\alpha_n} \|\hat{g}_n - f^\circ\|^2 \leq \sum_{i=2}^{n} \left(\frac{1}{2\alpha_i} - \frac{1}{2\alpha_{i-1}} \right) D^2 + \frac{1}{2\alpha_1} D^2 = \frac{D^2}{2\alpha_n},$$

since $\hat{g}_i \in \mathcal{F}$ for all $i = 1, \ldots, n$.

To bound the second term, notice that

$$\|u_i\|^2 = \|\Phi(\mathbf{x}_i, \cdot)(\mathbf{y}_i - A[\hat{g}_{i-1}](\mathbf{x}_i))\|^2 = \|\Phi(\mathbf{x}_i, \cdot)\|^2 \cdot |\mathbf{y}_i - A[\hat{g}_{i-1}](\mathbf{x}_i)|^2 \leq 2\|\Phi(\mathbf{x}_i, \cdot)\|^2 \cdot (|\mathbf{y}_i|^2 + |A[\hat{g}_{i-1}](\mathbf{x}_i)|^2).$$

Hence, if we take $C = \sup_{\mathbf{x} \in \mathbb{X}} \|\Phi(\mathbf{x}, \cdot)\|^2 < +\infty$, we find

$$\mathbb{E}[\|u_i\|^2] \leq 2C\mathbb{E}[(|\mathbf{Y}|^2 + |A[\hat{g}_{i-1}](\mathbf{X})|^2)] = 2C(\mathbb{E}[|\mathbf{Y}|^2] + \|A[\hat{g}_{i-1}\|^2_{L^2(\mathbb{X})}) \leq 2C(\mathbb{E}[|\mathbf{Y}|^2] + \|A\|^2 \|\hat{g}_{i-1}\|^2_{L^2(\mathbb{X})}) \leq 2C(\mathbb{E}[|\mathbf{Y}|^2] + \|A\|^2 D^2).$$

Finally, for the third term, note that, after taking expectation, the tower property and the fact that u_i is an unbiased estimator of the gradient of $\mathcal{R}_A$ (see Eq. (4)) give that

$$\begin{aligned}
&\mathbb{E}[\langle u_i - \nabla \mathcal{R}_A(\hat{g}_{i-1}), \hat{g}_{i-1} - f^\circ \rangle] = \int_{\mathbb{W}} \mathbb{E}[(u_i(\mathbf{w}) - \nabla \mathcal{R}_A(\hat{g}_{i-1})(\mathbf{w}))(\hat{g}_{i-1}(\mathbf{w}) - f^\circ(\mathbf{w}))] d\mu(\mathbf{w}) \\
&= \int_{\mathbb{W}} \mathbb{E}[\mathbb{E}[(u_i(\mathbf{w}) - \nabla \mathcal{R}_A(\hat{g}_{i-1})(\mathbf{w}))(\hat{g}_{i-1}(\mathbf{w}) - f^\circ(\mathbf{w})) \mid \mathcal{D}_{i-1}]] d\mu(\mathbf{w}) \\
&= \int_{\mathbb{W}} \mathbb{E}[(\mathbb{E}[u_i(\mathbf{w}) \mid \mathcal{D}_{i-1}] - \nabla \mathcal{R}_A(\hat{g}_{i-1})(\mathbf{w}))(\hat{g}_{i-1}(\mathbf{w}) - f^\circ(\mathbf{w}))] d\mu(\mathbf{w}) \\
&= \int_{\mathbb{W}} \mathbb{E}[(\mathbb{E}_{(\mathbf{X},\mathbf{Y})}[\Phi(\mathbf{X}, \mathbf{w})(A[\hat{g}_{i-1}](\mathbf{X}) - \mathbf{Y})] - \nabla \mathcal{R}_A(\hat{g}_{i-1})(\mathbf{w}))(\hat{g}_{i-1}(\mathbf{w}) - f^\circ(\mathbf{w}))] d\mu(\mathbf{w}) \\
&= \int_{\mathbb{W}} \mathbb{E}[(\nabla \mathcal{R}_A(\hat{g}_{i-1})(\mathbf{w}) - \nabla \mathcal{R}_A(\hat{g}_{i-1})(\mathbf{w}))(\hat{g}_{i-1}(\mathbf{w}) - f^\circ(\mathbf{w}))] d\mu(\mathbf{w}) = 0,
\end{aligned}$$

where $\mathcal{D}_{i-1}$ denotes the σ-algebra generated by the data $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{i-1}$ and $\mathbb{E}_{(\mathbf{X},\mathbf{Y})}$ is the expectation only with respect to $(\mathbf{X}, \mathbf{Y})$.

Again, by convexity of the risk function, $\mathcal{R}_A(\hat{f}_n) \leq \frac{1}{n} \sum_{i=1}^{n} \mathcal{R}_A(\hat{g}_i)$. Therefore,

$$\mathbb{E}\left[\mathcal{R}_A(\hat{f}_n) - \mathcal{R}_A(f^\circ)\right] \leq \frac{D^2}{2n\alpha_n} + \frac{1}{2n} \sum_{i=1}^{n} \alpha_i \mathbb{E}[\|u_i\|^2] \leq \frac{D^2}{2n\alpha_n} + \frac{C(\mathbb{E}[|\mathbf{Y}|^2] + \|A\|^2 D^2)}{n} \sum_{i=1}^{n} \alpha_i,$$

and the theorem is proved. □

D Dataset

A link for a github webpage with the data will be provided. The link is omitted at the moment to not reveal the authors identity.