A Empirical Application: Dataset and Code

From Febrero-Bande et al. [2022]: "a bitcoin address is a unique string of number and letters stored in the bitcoin's blockchain that can be the recipient or sender of bitcoins. A bitcoin transaction can have multiple addresses as inputs and outputs. One could think the address as the number of a bank account, but with some very distinct properties as public visibility of balance and all transactions, anonymity, cannot hold negative quantities of bitcoins (debt), and each user in this market usually has a large number of addresses. The user here, though, is not an actual individual, but a company/website/organization that typically holds multiple addresses; we refer to it as "entity". The balance of an address is the amount (possibly zero) of bitcoins in that address at a given time. The entity that a given address belongs to is usually unknown but some companies make some of its addresses public for various reasons. Based on this public information, one can identify more addresses as belonging to this same entity if they were inputs to the same transaction with one or more inputs from this entity."

The data used in our classification model is the cumulative credits of the first 3000 hours of the address. In Figure 2 below we show some example of balances over time of some addresses. The dataset and the R code used in this paper can be found here.

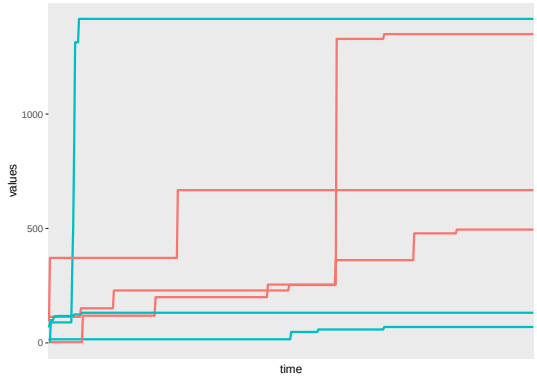


Figure 2: Example of cumulative credits for six different addresses across 501 data points. In red, addresses associated with criminal activity, in blue, addresses associated with noncriminal activities.

B Functional Gradient for the Deconvolution Problem

Remember that the operator A is given by

$$A[f](\mathbf{x}) = \int_{\mathbb{W}} k(\mathbf{x}-\mathbf{w}) f(\mathbf{w}) d\mu(\mathbf{w}). \tag{8}$$

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 we find, as in Eq. (5),

$$u_i(\mathbf{w}) = k(\mathbf{x}_i - \mathbf{w})\partial_2 \ell(\mathbf{y}_i, A[\hat{g}_{i-1}](\mathbf{x}_i)).$$

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.

C 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.

C.1 Functional Linear Regression

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

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

where the operator A is given by

$$A[f](\mathbf{x}) = \int_0^T f(s)\mathbf{x}(s)ds. \tag{9}$$

Recall the data generating process described in 5.1. We set $\mathbb{W} = [0, 1]$, $f^\circ(w) = \sin(4\pi w)$, and $\mathbf{X}$ simulated accordingly a Brownian motion in $[0, 1]$. We also consider a noise-signal ratio of 0.2. Next, we study also the case where f° oscillates between $1, -1$ in the points $\mathbf{w} = 0.25, 0.5, 0.75, 1$. 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 $O(1/\sqrt{N})$ (which satisfy the requirements discussed after 4.9). We simulate the data generating process 10 times in order to compute the metrics performance. We compare the methods in terms of Mean Square Error of the recovered function f°.

In Figure 3 we present the Mean Squared Error and with Error Bars representing 2 standard deviations. In this case, we can see that PFLR with different specifications out-perform our propposed algorithms, which achieves similar performance as Landweber iterations. It is important to note here, that while PFLR methods are tailored for this type of problems, ours, as well as Landweber iterations, are not. Nevertheless, we can see in Figure 1a that essentially all the algorithms are capable of recovering the true underlying function f°.

In Figure 4 we have a similar setup in a harder problem, where the underlying f° is not as smooth as before. In this case, the advantage of the PFLR reduces and the performance of all the methods are very similar. It is important to note that our approach makes use of only one sample at each iteration of our proposed algorithms. One can improve the stability and convergence of the estimated algorithms by simply using more samples at each time. In case one uses all the samples in each iteration (such as what is commonly done in Landweber iteration or boosting procedures in standard regression problems), Theorem 4.9 cannot be applied directly but empirically the methods perform well. We illustrate this approach in Figure 5, where we make use of all the samples in every iteration of our algorithms.

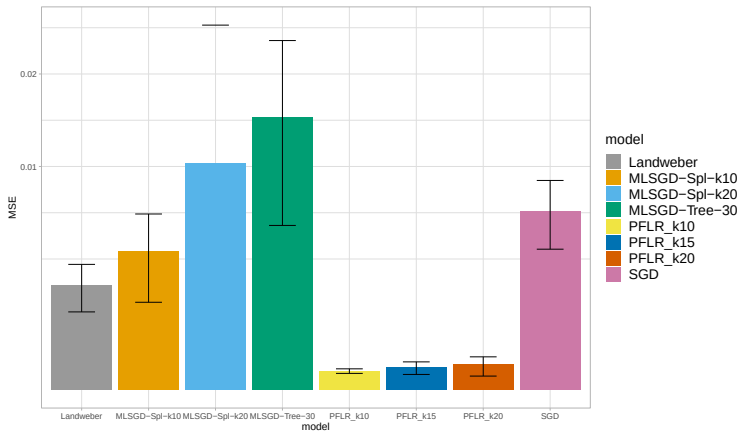


Figure 3: MSE with 2 standard deviations error bars for 10 simulations with f as the sine function. Y-axis in square-root scale.

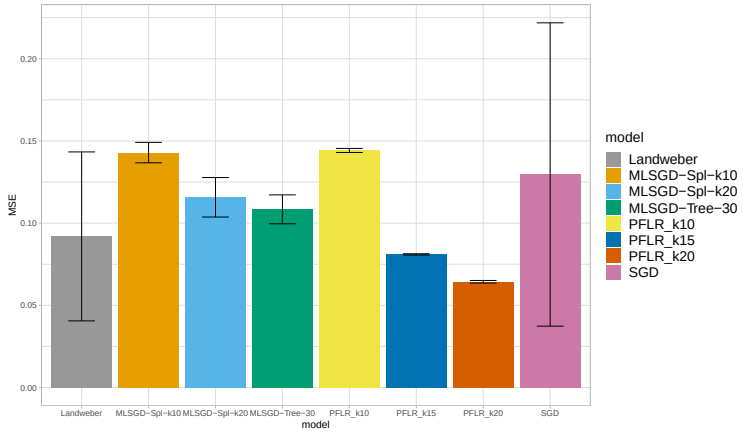


Figure 4: MSE with 2 standard deviations error bars for 10 simulations with f as step function.

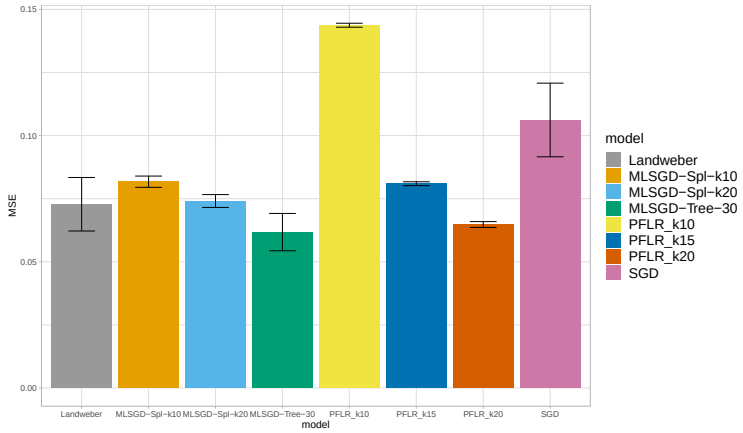


Figure 5: MSE with 2 standard deviations error bars for 10 simulations with f as step function and using all samples for the gradient computation.

C.2 Deconvolution

For the deconvolution problem we examine the following numerical exercise. We take two choices of functional parameters for Eq. (8), as a peak function:

$$f(w) = e^{-w^2}. \tag{10}$$

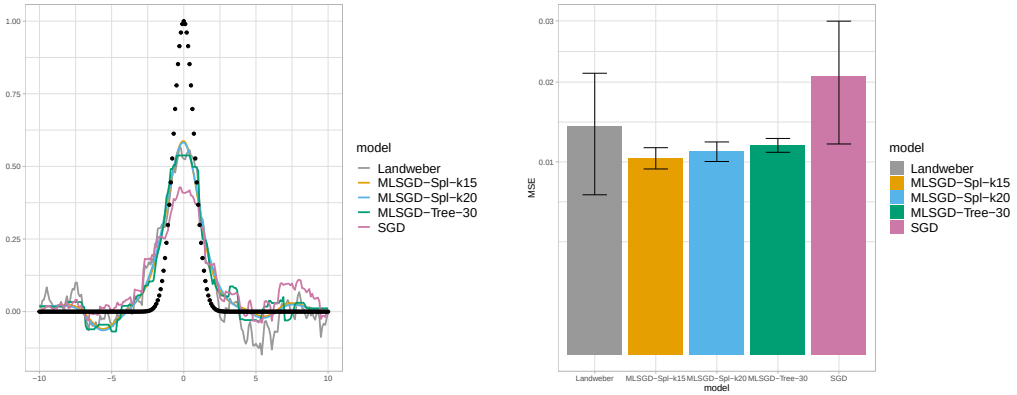
We consider the kernel to be given by

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

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 6a 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. Nevertheless, this apprently instability seems to

allow both the SGD-SIP and Landweber to better estimate the function in the peak, which compensate in the Mean Square Error estimator despite the increase in the volatility of the estimator. In Figure 6b we present the Mean Squared Errors and Error Bars with two standard deviations.



(a) (b)

Figure 6: Numerical results for the deconvolution problem. In (a) we have an example of the fitted functions for one simulation. In (b) we have the MSE with error bars representing two standard-deviations.

D Proof of Theorem 4.9

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 $\langle \cdot, \cdot \rangle$. 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

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

since $\hat{g}_i \in \mathcal{F}$ for all $i = 1, \dots, n$.
To bound the second term, notice that[4]

$$\begin{aligned}\|u_i\|^2 &= \|\Phi(\mathbf{x}_i,\cdot)\partial_2\ell(\mathbf{y}_i, A[\hat{g}_{i-1}](\mathbf{x}_i))\|^2 \leq \|\Phi(\mathbf{x}_i,\cdot)\|^2\|\partial_2\ell(\mathbf{y}_i, A[\hat{g}_{i-1}](\mathbf{x}_i))\|^2 \\ &\leq 2\tilde{C}\|\Phi(\mathbf{x}_i,\cdot)\|^2\cdot(\|\mathbf{y}_i\|^2+\|A[\hat{g}_{i-1}](\mathbf{x}_i)\|^2).\end{aligned}$$

Hence, if we take $C = \sup_{\mathbf{x}\in\mathbb{X}}\|\Phi(\mathbf{x},\cdot)\|^2 < +\infty$, we find[5]

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

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. (5)) give that

$$\begin{aligned}\mathbb{E}[\langle u_i-\nabla\mathcal{R}_A(\hat{g}_{i-1}),\hat{g}_{i-1}-f^\circ\rangle] &= \mathbb{E}[\mathbb{E}[\langle u_i-\nabla\mathcal{R}_A(\hat{g}_{i-1}),\hat{g}_{i-1}-f^\circ\rangle \mid \mathcal{D}_{i-1}]] \\ &= \mathbb{E}[\langle\mathbb{E}[u_i-\nabla\mathcal{R}_A(\hat{g}_{i-1}) \mid \mathcal{D}_{i-1}],\mathbb{E}[\hat{g}_{i-1}-f^\circ \mid \mathcal{D}_{i-1}]\rangle] \\ &= \mathbb{E}[\langle\mathbb{E}[u_i \mid \mathcal{D}_{i-1}]-\nabla\mathcal{R}_A(\hat{g}_{i-1}),\hat{g}_{i-1}-f^\circ\rangle] = 0.\end{aligned}$$

where $\mathcal{D}_{i-1}$ denotes the σ-algebra generated by the data $\{\mathbf{x}_k,\mathbf{y}_k\}_{k=1}^{i-1}$.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,

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

and the theorem is proved. □

[4]In the computations below we use the fact that the point-to-point loss function ordinarily has Lipschitz gradients which implies at most linear growth. The two examples analyzed in this paper trivially satisfies this bound.

[5]Abusing the notation and defining C as $C\,\tilde{C}$.