IOP Publishing Inverse Problems

Inverse Problems **35** (2019) 114008 (23pp) https://doi.org/10.1088/1361-6420/ab2787

Sampled Tikhonov regularization for large linear inverse problems

J Tanner Slagel[1], Julianne Chung[2], Matthias Chung[2,4], David Kozak[3] and Luis Tenorio[3]

[1] Department of Mathematics, Virginia Tech, Blacksburg, VA, United States of America
[2] Department of Mathematics, Computational Modeling and Data Analytics Division, Academy of Integrated Science, Virginia Tech, Blacksburg, VA, United States of America
[3] Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, United States of America

E-mail: slagelj@vt.edu, jmchung@vt.edu, mcchung@vt.edu, dkozak@mines.edu and ltenorio@mines.edu

Received 14 December 2018, revised 13 May 2019
Accepted for publication 6 June 2019
Published 4 October 2019

Abstract

In this paper we investigate iterative methods that are based on sampling of the data for computing Tikhonov-regularized solutions. We focus on very large inverse problems where access to the entire data set is not possible all-at-once (e.g. for problems with streaming or massive datasets). Row-access methods provide an ideal framework for solving such problems since they only require access to 'blocks' of the data at any given time. However, when using these iterative sampling methods to solve inverse problems, the main challenges include a proper choice of the regularization parameter, appropriate sampling strategies, and a convergence analysis. To address these challenges, we describe a family of sampled iterative methods that can incorporate data as they become available (e.g. randomly sampled). We consider two sampled iterative methods where the iterates can be characterized as solutions to a sequence of approximate Tikhonov problems. The first method requires the regularization parameter to be fixed *a priori* and converges asymptotically to an unregularized solution for randomly sampled data. This is undesirable for inverse problems. Thus, we focus on the second method where the main benefits are that the regularization parameter can be updated during the iterative process and the iterates converge asymptotically to a Tikhonov-regularized solution. We describe adaptive approaches to update the regularization parameter that are based on sampled residuals, and we provide a limited-memory variant for larger problems. Numerical examples, including a large-scale super-resolution imaging example, demonstrate the potential for these methods.

[4] Author to whom any correspondence should be addressed.

Keywords: imaging, sampling methods, recursive least squares, streaming data, Tikhonov regularization

(Some figures may appear in colour only in the online journal)

1. Introduction

With faster scan speeds on recently-developed imaging devices and new applications to computer vision and machine learning, datasets are becoming so large that the entire dataset can not be accessed 'all-at-once' [34]. Furthermore, in automated pipelines, data is being streamed and a major challenge is to obtain immediate feedback (e.g. a partial reconstruction) to inform the data acquisition process [43]. For these and other scenarios, existing methods that require all-at-once access to the data are not feasible. Instead, we consider randomized or sampling methods where only 'blocks' of the data are required at a given time.

In this paper, we focus on linear inverse problems of the form,

$$\mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon},$$

where $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$ contains the desired, unknown parameters, $\mathbf{A} \in \mathbb{R}^{m\times n}$ models the data acquisition process, $\mathbf{b} \in \mathbb{R}^m$ contains the observed data (which may be streaming), and $\boldsymbol{\epsilon} \in \mathbb{R}^m$ represents noise or errors in the data. We assume that the random vector $\boldsymbol{\epsilon}$ has mean zero and a finite second moment. In the generic setup, the goal of the inverse problem is to estimate $\mathbf{x}_{\text{true}}$, given a model $\mathbf{A}$ and observations $\mathbf{b}$. Typically, the matrix $\mathbf{A}$ represents a discrete, linear version of a given model stemming, for instance, from a discretized PDE network, integral equation, or regression model [26, 37]. For the problems of interest, m and n may be so large that accessing and/or storing all rows of $\mathbf{A}$ at once is infeasible.

In this work we consider *ill-posed* inverse problems where regularization is required to compute reasonable solutions. Here, we focus on solving the Tikhonov-regularized problem,

$$\min_{\mathbf{x}} f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{L}\mathbf{x}\|_2^2, \tag{1}$$

where $\lambda > 0$ is the regularization parameter, and for simplicity we assume that $\mathbf{L}$ has full column rank. When all of $\mathbf{b}$ and $\mathbf{A}$ are available or can be accessed at once (e.g. via matrix-vector multiplication with $\mathbf{A}$), the Tikhonov solution,

$$\mathbf{x}(\lambda) = (\mathbf{A}^\top\mathbf{A} + \lambda\mathbf{L}^\top\mathbf{L})^{-1}\mathbf{A}^\top\mathbf{b}, \tag{2}$$

can be computed using a plethora of existing iterative methods (e.g. Krylov or other optimization methods [25, 30]). Note that $\mathbf{x}(0)$ is the unregularized solution, which is defined if $\mathbf{A}$ has full column rank.

To solve (1) we consider sampled iterative methods of the form,

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k\mathbf{g}_k(\mathbf{x}_{k-1}), \qquad k \in \mathbb{N}, \tag{3}$$

where $\mathbf{x}_0$ is an initial iterate, $\mathbf{g}_k(\mathbf{x}_{k-1})$ is a vector (carrying gradient information of the least squares problem), and the matrix $\mathbf{B}_k \in \mathbb{R}^{n\times n}$ is updated at each iteration (carrying curvature information of the least squares problem). A learning rate or line search parameter is not required in this case and is set to its 'natural' value of 1 [10]. Specific choices for $\mathbf{B}_k$ and $\mathbf{g}_k$ will be described in section 2, with connections to other known stochastic approximation methods described in section 2.3.

In this work we focus on sampled methods that do not require all-at-once access to $\mathbf{A}$, but we note that if all of $\mathbf{A}$ is available, then iterated Tikhonov regularization methods for

linear problems [11, 20, 24] can be formulated as (3) where $\mathbf{B}_k^{-1} = \mathbf{A}^\top\mathbf{A} + \lambda_k\mathbf{L}^\top\mathbf{L}$ and $\mathbf{g}_k(\mathbf{x}_{k-1}) = \mathbf{A}^\top(\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b})$. Furthermore, we note that iterative methods of the form (3) typically stem from nonlinear optimization problems where $\mathbf{B}_k$ is an approximation to the inverse Hessian and $\mathbf{g}_k$ is the gradient at the current iterate $\mathbf{x}_{k-1}$ [39]. Such methods take only one step to converge for the linear problem (e.g. take $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{B}_1^{-1} = \mathbf{A}^\top\mathbf{A} + \lambda\mathbf{L}^\top\mathbf{L}$, and $\mathbf{g}_1 = \mathbf{A}^\top\mathbf{b}$). However, this is not possible if m and n are too large for the computer memory or if the data is not available all-at-once (e.g. in streaming problems). Furthermore, determining a suitable choice of λ can be computationally infeasible in such settings, and the information available, i.e. $\mathbf{B}_k$ and $\mathbf{g}_k$, may be subject to noise. Thus, we consider nonlinear methods of the form (3) for Tikhonov regularization with massive data, where the main benefits are that (i) the data is sampled (e.g. randomly) or streamed, (ii) the regularization parameter can be adapted, and (iii) the methods converge asymptotically and in one epoch to a Tikhonov-regularized solution. Sophisticated regularization parameter selection methods are well-established if the full system is available (for example, see [31, 45]); however, the ability to update the regularization parameter within iterative methods of the form (3) while also ensuring convergence of iterates to a regularized solution is, to the best of our knowledge, an unresolved problem.

1.1. Problem formulation

In the following, we describe a mathematical formulation of the problem that allows us to solve (1) in situations where samples of $\mathbf{A}$ and $\mathbf{b}$ become available over time. Such scenarios are common in medical imaging, e.g. in tomography where data is being processed as it is being collected [3], and in astronomy, e.g. in super-resolution imaging where a high-resolution image is constructed from low-resolution images that are being video streamed [28].

Formally, at the kth iteration, we assume that a set of rows of $\mathbf{A}$ and corresponding elements of $\mathbf{b}$ become available, which we denote by $\mathbf{W}_k^\top\mathbf{A}$ and $\mathbf{W}_k^\top\mathbf{b}$ respectively. Here the matrix $\mathbf{W}_k \in \mathbb{R}^{m\times\ell}$ is a *sampling* matrix, which selects rows of $\mathbf{A}$ and $\mathbf{b}$. For a fixed $M \in \mathbb{N}$ we assume that matrices $\{\mathbf{W}_i\}_{i=1}^M$ satisfy the following properties:

(i) for each $i \in \{1,\dots,M\}$, $\mathbf{W}_i \in \mathbb{R}^{m\times\ell}$, where $\ell = \frac{m}{M}$[5] and
(ii) the sum $\sum_{i=1}^M \mathbf{W}_i\mathbf{W}_i^\top = \mathbf{I}_m$.

The first assumption implies that the size of $\mathbf{W}_i^\top\mathbf{A}$ is smaller than the size of $\mathbf{A}$, and thus computationally manageable. The second assumption guarantees that all rows of $\mathbf{A}$ are given equal weight; however, importance sampling could be included and results in a weighted least squares problem.

Notice that if $\mathbf{W}_k$ is sparse with only a few non-zero elements in a subset of the m columns, $\mathbf{W}_k^\top\mathbf{A}$ extracts only rows of $\mathbf{A}$ where $\mathbf{W}_k$ has nonzero entries. These methods are commonly known as row-action methods [3, 21], where *Kaczmarz*-type methods are a prominent subclass of row-action methods [29, 48]. A notable difference of our methods compared to other approaches is that the matrices $\mathbf{B}_k$ accumulate information from the previous iterates. Randomized or sketching methods are also related in that a single realization of $\mathbf{W}_k$ is used to project a large system onto a lower dimensional subspace [17, 44]. However, these methods typically require access to all of the data at once.

[5] To avoid a notational distraction, we assume all matrices $\mathbf{W}_i$ are of the same dimension and $\ell M = m$; hence, $\ell \in \mathbb{N}$. However a generalization with different matrix sizes $\mathbf{W}_i \in \mathbb{R}^{m\times\ell_i}$ is straightforward.

1.2. Overview and outline

In this paper, we describe iterative sampling methods for solving Tikhonov-regularized problems, where the main distinction from existing methods such as hybrid Krylov methods and iterated Tikhonov methods is that we do not require all-at-once access to the forward model. The main contributions include the characterization of iterates as solutions to partial or full Tikhonov problems and asymptotic convergence results. In terms of methodology, we highlight the sampled Tikhonov method where the regularization parameter can be updated during the iterative process such that after each epoch of data, the iterates are Tikhonov-regularized solutions. Additionally, the sampled Tikhonov method converges asymptotically to a Tikhonov-regularized solution. Other developments include methods for updating the regularization parameter using sampled data and limited-memory variants for problems with many unknowns.

The paper is organized as follows. In section 2 we describe two iterative methods for Tikhonov regularization with sampling. Various theoretical results are provided, including asymptotic convergence results. In section 3 we describe sampled regularization parameter selection methods that can be used to update the regularization parameter. Numerical illustrations are provided throughout, and a limited-memory variant of these methods is described in section 4, along with results for a large-scale imaging problem. Conclusions and future work are discussed in section 5.

2. Iterative sampling methods for Tikhonov regularization

Iterative sampling methods for Tikhonov regularization can be used to solve massive linear inverse problems. We investigate two methods. Let $\mathbf{y}_0, \mathbf{x}_0 \in \mathbb{R}^n$ be initial iterates and let $\mathbf{W}_i \in \mathbb{R}^{m\times \ell}$, $i = 1, \ldots, k$ be arbitrary matrices. For notational convenience, we denote $\mathbf{A}_i = \mathbf{W}_i^\top \mathbf{A}$ and $\mathbf{b}_i = \mathbf{W}_i^\top \mathbf{b}$. Assuming a fixed regularization parameter λ, the first method that we consider is *regularized recursive least squares* (`rrls`)[6], which is defined as

$$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbf{B}_k \mathbf{A}_k^\top (\mathbf{A}_k \mathbf{y}_{k-1} - \mathbf{b}_k), \quad k \in \mathbb{N}, \tag{4}$$

where $\mathbf{B}_k = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$. If $\mathbf{W}_i$ is the ith column of the identity matrix, `rrls` is an extension of the recursive least squares algorithm [7] that includes a Tikhonov term. Since it may be difficult to know a good regularization parameter in advance, we propose a *sampled Tikhonov* (`sTik`) method, where the iterates are defined as

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k \left(\mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) + \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1}\right), \quad k \in \mathbb{N}, \tag{5}$$

where $\mathbf{B}_k = \left(\sum_{i=1}^k \Lambda_i \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$ and scalar $\sum_{i=1}^k \Lambda_i > 0$. Compared to `rrls`, the main advantages of the `sTik` method are that the regularization parameter can be updated during the iterative process and that in a sampled framework, the `sTik` iterates converge asymptotically to a Tikhonov solution whereas the `rrls` iterates converge asymptotically to an unregularized solution. Of course, selecting a good regularization parameter can be difficult, especially for problems with a small range of good values. In any case, for inverse problems it is desirable that the numerical method for solution computation converges to a regularized solution.

[6] This should not be confused with the residual-reducing LS (RRLS) algorithm referenced in [41].

In this section, we begin by showing that for arbitrary matrices $\mathbf{W}_i$, both `rrls` and `sTik` iterates can be recast as solutions to regularized least squares problems. See appendix A for proofs for all theorems from section 2.

Theorem 2.1. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{L} \in \mathbb{R}^{s\times n}$ have full column rank and $\mathbf{W}_i \in \mathbb{R}^{m\times \ell}$, $i = 1, \ldots, k$ be an arbitrary sequence of matrices.*

(i) For $\lambda > 0$ and an arbitrary initial guess $\mathbf{y}_0 \in \mathbb{R}^n$, the `rrls` iterate (4) with $\mathbf{B}_k = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$ is the solution of the least squares problem

$$\min_{\mathbf{x}} \left\| [\mathbf{W}_1, \ldots, \mathbf{W}_k]^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) \right\|_2^2 + \lambda \left\| \mathbf{L}(\mathbf{x} - \mathbf{y}_0) \right\|_2^2. \tag{6}$$

(ii) For $\lambda_k = \sum_{i=1}^k \Lambda_i > 0$ for any k and an arbitrary initial guess $\mathbf{x}_0 \in \mathbb{R}^n$, the `sTik` iterate (5) with $\mathbf{B}_k = \left(\sum_{i=1}^k \Lambda_i \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^k \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1}$ is the solution of the least squares problem

$$\min_{\mathbf{x}} \left\| [\mathbf{W}_1, \ldots, \mathbf{W}_k]^\top (\mathbf{A}\mathbf{x} - \mathbf{b}) \right\|_2^2 + \lambda_k \left\| \mathbf{L}\mathbf{x} \right\|_2^2. \tag{7}$$

The above results are true for any arbitrary sequence of matrices $\{\mathbf{W}_k\}$. Next, we consider a fixed set of matrices, as described in the introduction, and allow random sampling from this set. To be precise, define $\mathbf{W}_{\tau(k)}$ to be a random variable at the kth iteration, where $\tau(k)$ is a random variable that indicates a sampling strategy. For example, if we let $\tau(k)$ be a uniform random variable on the set $\{1, \ldots, M\}$, then we would be sampling with replacement. In section 2.1 we prove asymptotic convergence of `rrls` and `sTik` iterates using this sampling strategy. We then focus on random cyclic sampling, where for each $j \in \mathbb{N}$, $\{\tau(k)\}_{jM+1}^{(j+1)M}$ is a random permutation on the set $\{1, \ldots, M\}$. Note, cyclic sampling, where $\tau(k) = k \bmod M$, is a special case of random cyclic sampling. We note that, until all blocks have been sampled, random cyclic sampling is just sampling without replacement. For random cyclic sampling, we characterize iterates after each epoch and prove asymptotic convergence of `rrls` and `sTik` iterates in section 2.2. An illustrative example comparing the behavior of the solutions is provided in section 2.4. For notational simplicity we denote $\mathbf{A}_{\tau(k)} = \mathbf{W}_{\tau(k)}^\top \mathbf{A}$ and $\mathbf{b}_{\tau(k)} = \mathbf{W}_{\tau(k)}^\top \mathbf{b}$.

Notice that for both random sampling and random cyclic sampling, we have the following property,

$$\mathbb{E}\, \mathbf{W}_{\tau(k)} \mathbf{W}_{\tau(k)}^\top = \frac{1}{M} \mathbf{I}_m = \frac{\ell}{m} \mathbf{I}_m. \tag{8}$$

There are many choices for $\{\mathbf{W}_i\}$, see e.g. [14, 33, 35], but a simple choice is a block column partition of a permutation matrix. For the choice of $\{\mathbf{W}_i\}$ we will consider, $\mathbf{A}_{\tau(k)}$ is just a predefined block of rows of $\mathbf{A}$.

2.1. Random sampling

Next we investigate the asymptotic convergence of `rrls` and `sTik` iterates for the case of uniform random sampling. This is also referred to as sampling with replacement.

Theorem 2.2. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Let $\mathbf{L} \in \mathbb{R}^{s\times n}$ have full column rank and define $\mathbf{x}(\lambda)$ as in (2). Let $\{\mathbf{W}_i\}_{i=1}^M$ be a set of real valued $m \times \ell$ matrices with the property that*

$\sum_{i=1}^{M} \mathbf{W}_i \mathbf{W}_i^\top = \mathbf{I}_m$, *and let* $\tau(k)$ *be a uniform random variable on the set* $\{1, \dots M\}$.

(i) *Let* $\lambda > 0$, $\mathbf{y}_0 \in \mathbb{R}^n$ *be arbitrary, and define the sequence* $\{\mathbf{y}_k\}$ *as*

$$\mathbf{y}_k = \mathbf{y}_{k-1} - \mathbf{B}_k \mathbf{A}_{\tau(k)}^\top (\mathbf{A}_{\tau(k)} \mathbf{y}_{k-1} - \mathbf{b}_{\tau(k)}), \quad k \in \mathbb{N}, \tag{9}$$

where $\mathbf{B}_k = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1}$. *If* $\mathbf{A}$ *has full column rank, then* $\mathbf{y}_k \xrightarrow{\text{a.s.}} \mathbf{x}(0)$.

(ii) *Let* $\sum_{i=1}^{k} \Lambda_i > 0$ *for all* k, *and* $\lambda = \lim_{k\to\infty} \frac{M}{k} \sum_{i=1}^{k} \Lambda_i > 0$ *be finite. Let* $\mathbf{x}_0 \in \mathbb{R}^n$ *be arbitrary, and define the sequence* $\{\mathbf{x}_k\}$ *as*

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{B}_k \left(\mathbf{A}_{\tau(k)}^\top (\mathbf{A}_{\tau(k)} \mathbf{x}_{k-1} - \mathbf{b}_{\tau(k)}) + \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1}\right), \tag{10}$$

where $\mathbf{B}_k = \left(\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1}$. *Then* $\mathbf{x}_k \xrightarrow{\text{a.s.}} \mathbf{x}(\lambda)$.

The significance of theorem 2.2 is that the `rrls` iterates converge asymptotically to the *unregularized* least squares solution, $(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$, which is undesirable for ill-posed inverse problems. On the other hand, the `sTik` iterates converge asymptotically to a Tikhonov-regularized solution. Note that for a given λ, convergence to $\mathbf{x}(\lambda)$ is ensured by setting $\Lambda_k = \lambda/M$. A more realistic scenario would be to adapt Λ_k as data become available, since the desired regularization parameter is typically not known before the data is received. Hence, parameter selection strategies for selecting Λ_k are addressed in section 3.

2.2. Random cyclic sampling

Next we investigate `rrls` and `sTik` with random cyclic sampling. In addition to proving asymptotic convergence in this case, we can also describe the iterates as Tikhonov solutions after each epoch, where an epoch is defined as a sweep through all the data.

Theorem 2.3. *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{b} \in \mathbb{R}^m$. *Let* $\mathbf{L} \in \mathbb{R}^{s\times n}$ *have full column rank and* $\{\mathbf{W}_i\}_{i=1}^{M}$ *be a set of real valued* $m \times \ell$ *matrices with the property that* $\sum_{i=1}^{M} \mathbf{W}_i \mathbf{W}_i^\top = \mathbf{I}_m$, *and let* $\tau(k)$ *be a random variable such that for* $j \in \mathbb{N}$, $\{\tau(k)\}_{jM+1}^{(j+1)M}$ *is a random permutation on the set* $\{1, \dots, M\}$.

(i) If $\lambda > 0$, $\mathbf{y}_0 = \mathbf{0}$, *and the sequence* $\{\mathbf{y}_k\}$ *is defined as (9) with* $\mathbf{B}_k = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1}$, *then the iterate at the* j*th epoch is* $\mathbf{y}_{jM} = \mathbf{x}\left(\frac{1}{j}\lambda\right)$.

(ii) Let $\{\Lambda_k\}$ *be an infinite sequence with the property that* $\lambda_k = \sum_{i=1}^{k} \Lambda_i > 0$. *If* $\mathbf{x}_0$ *is arbitrary and the sequence* $\{\mathbf{x}_k\}$ *is defined as (10) with* $\mathbf{B}_k = \left(\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1}$, *then the iterate at the* j*th epoch is* $\mathbf{x}_{jM} = \mathbf{x}\left(\frac{1}{j}\lambda_{jM}\right)$.

Notice that at every epoch, the effective regularization parameter for `rrls`, i.e. λ/j, is reduced. Also, if $\mathbf{A}$ has full column rank, we have $\lim_{j\to\infty} \mathbf{y}_{jM} = \mathbf{x}(0)$. On the other hand, the `sTik` iterates converge to a Tikhonov-regularized solution, since at each epoch $j = k/M$ and we have $\mathbf{x}_{jM} = \mathbf{x}_k = \mathbf{x}\left(\frac{M}{k}\lambda_k\right)$ and $\frac{M}{k}\lambda_k > 0$. In section 2.4 we illustrate the convergence behavior of the `rrls` and `sTik` iterates, but first we make some connections to existing optimization methods.

2.3. Connections to stochastic approximation methods

There is a connection between the iterative methods with sampling presented in section 2 and stochastic approximation methods that becomes apparent if the Tikhonov problem (1) is recast as a stochastic optimization problem. For simplicity, consider random sampling (i.e. with replacement), where $\tau(k)$ is a uniform random variable on the set $\{1,\dots,M\}$. Then if we define $f_{\tau(k)}(\mathbf{x}) = \left\|\mathbf{W}_{\tau(k)}^{\top}\left(\mathbf{A}\mathbf{x}-\mathbf{b}\right)\right\|_2^2 + \frac{\lambda}{M}\|\mathbf{L}\mathbf{x}\|_2^2$, it is easy to show that

$$\mathbb{E} f_{\tau(k)} \propto f,$$

and therefore

$$\arg\min_{\mathbf{x}} \mathbb{E} f_{\tau(k)}(\mathbf{x}) = \arg\min_{\mathbf{x}} f(\mathbf{x}). \tag{11}$$

Stochastic approximation methods represent one class of methods that can be used to compute solutions to the expectation minimization problem on the left of (11) [47]. For the Tikhonov problem, a stochastic approximation method has the form,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k \nabla f_{\tau(k)}\left(\mathbf{x}_k\right), \tag{12}$$

where $\nabla f_{\tau(k)}\left(\mathbf{x}_k\right) = \mathbf{A}_{\tau(k)}^{\top}\left(\mathbf{A}_{\tau(k)}\mathbf{x}_k - \mathbf{b}_{\tau(k)}\right) + \frac{\lambda}{M}\mathbf{L}\mathbf{x}_k$ is the sample gradient for the Tikhonov problem. Different choices of $\mathbf{B}_k$ can be used in (12). If $\mathbf{B}_k = \left(\frac{k\lambda}{M}\mathbf{L}^{\top}\mathbf{L} + \sum_{i=1}^{k}\mathbf{A}_{\tau(i)}^{\top}\mathbf{A}_{\tau(i)}\right)^{-1}$, then all of the previously computed global curvature information is encoded in $\mathbf{B}_k$ and we recover the `sTik` method with $\Lambda_i = \frac{\lambda}{M}$. Theorem 2.2 (ii) shows that these iterates will converge asymptotically to the minimizer of (11), but storage can get costly. Another option is to take $\mathbf{B}_k = \mathbf{I}_n$, which corresponds to the stochastic gradient method [8]. For faster convergence closer to the minimizer, there are various methods in the stochastic optimization literature that can be used to approximate the global curvature information $\nabla^2 f\left(\mathbf{x}_k\right)$ [9, 32]. For example, a stochastic LBFGS method stores a small set of vectors, rather than matrix $\mathbf{B}_k$, and can perform multiplications in an efficient manner [12, 36].

We are most interested in the Tikhonov problem (1), but we note that there exists methods for the case where $\lambda = 0$ that have connections to stochastic optimization methods. Using the same reformulation as above, a stochastic approximation method would have the form (12). If we take $\mathbf{B}_k = \left(\nabla^2 f_{\tau(k)}\right)^{\dagger}$, then we get the randomized block Kaczmarz method [3, 38, 48]. Notice that the curvature information comes only from the current sample. On the other hand, if $\mathbf{B}_k$ is chosen to contain all previous curvature information, we get the `rrls` iterates,

$$\mathbf{y}_k = \mathbf{y}_{k-1} - \left(\lambda\mathbf{L}^{\top}\mathbf{L} + \sum_{i=1}^{k}\mathbf{A}_{\tau(i)}^{\top}\mathbf{A}_{\tau(i)}\right)^{-1}\mathbf{A}_{\tau(k)}^{\top}(\mathbf{A}_{\tau(k)}\mathbf{y}_{k-1} - \mathbf{b}_{\tau(k)}), \tag{13}$$

where $\lambda\mathbf{L}^{\top}\mathbf{L}$ is included to ensure invertibility and is often replaced with $\lambda\mathbf{I}_n$. The concern for inverse problems is that the iterates in (13) converge to the unregularized problem, see, theorem 2.2 (i). The connection between recursive least squares and stochastic approximation methods was noted in [32], and the approximation can be interpreted as a regularized stochastic approximation method that was considered, e.g. in [10, 14].

2.4. An illustration

In the following illustration, we use a small toy example to highlight the convergence behavior of `rrls` and `sTik` iterates. We investigate both random sampling and random cyclic

sampling, and we demonstrate convergence by plotting solutions after multiple epochs of the data. The example we use is a Tikhonov problem of the form (1), where

$$\mathbf{A} = \begin{bmatrix} \mathbf{1} & \boldsymbol{\delta}_{\mathbf{A}} \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{10\times 2}, \qquad \mathbf{b} = \mathbf{A}\mathbf{x}_{\text{true}} + \boldsymbol{\delta}_{\mathbf{b}}, \qquad \text{and} \quad \mathbf{x}_{\text{true}} = \mathbf{1}.$$

The vectors $\boldsymbol{\delta}_{\mathbf{A}}$ and $\boldsymbol{\delta}_{\mathbf{b}}$ are realizations from the normal distributions $\mathcal{N}(\mathbf{0}, 0.005\,\mathbf{I}_9)$ and $\mathcal{N}(\mathbf{0}, 0.1\,\mathbf{I}_{10})$ respectively, and $\mathbf{1}$ is the vector of ones of appropriate length. We further choose $\mathbf{L} = \mathbf{I}_2$ and fix $\lambda = 0.2$ for the `rrls` iterates $\mathbf{y}_k$. For `sTik` iterates $\mathbf{x}_k$, we choose the parameters Λ_k such that the regularization is constant at each epoch, i.e. $\frac{10}{k}\sum_{i=1}^{k}\Lambda_i = 0.2$. With this setup we have $\mathbf{x}(0) = [1.0869, -1.3799]^\top$ and $\mathbf{x}(\lambda) = [1.0698, -0.0271]^\top$. We let $\mathbf{W}_{\tau(i)}$ be the $\tau(i)$th column of the identity matrix, and set $\mathbf{x}_0 = \mathbf{y}_0 = \mathbf{0}$.

In figure 1, we provide two illustrations. In the left panel, we provide the true solution $\mathbf{x}_{\text{true}}$, the unregularized solution $\mathbf{x}(0)$, the Tikhonov solution $\mathbf{x}(\lambda)$, and the `rrls` iterates after each epoch. The `rrls` iterates with random sampling with replacement are denoted by $\mathbf{y}_k^{\text{r}}$, and the `rrls` iterates with random cyclic sampling are denoted by $\mathbf{y}_k^{\text{c}}$. Notice that by theorem 2.3, $\mathbf{y}_k^{\text{c}}$ at each epoch is a Tikhonov solution, i.e. after the jth epoch $\mathbf{y}_{jM}^{\text{c}} = \mathbf{x}\left(\frac{1}{j}\lambda\right)$. Thus, we get a set of Tikhonov solutions with vanishing regularization parameters, and these iterates asymptotically converge to the unregularized solution. For `rrls` with random sampling, we run 1000 simulations and provide one sample path, along with the mean (dotted line) and region of the 95th percentile shaded in grey. We note that the mean of $\{\mathbf{y}_k^{\text{r}}\}$ is almost identical to the random cyclic sequence $\{\mathbf{y}_k^{\text{c}}\}$ (red line) suggesting that the random sequence $\{\mathbf{y}_k^{\text{r}}\}$ is an unbiased estimator of the deterministic sequence $\{\mathbf{y}_k^{\text{c}}\}$ (at each epoch). In the right panel of figure 1, we provide the `sTik` iterates with random sampling, which are denoted by $\mathbf{x}_k^{\text{r}}$. Again, we run 1000 simulations and provide one simulation along with the shaded percentiles. It is evident that with more epochs, the iterates approach the desired Tikhonov solution. To aid with visual scaling, the axis for the right figure corresponds to the dotted rectangular box in the left figure. The `sTik` iterates with random cyclic sampling are omitted since $\mathbf{x}_{jM}^{\text{c}} = \mathbf{x}(\lambda)$ (i.e. we get the Tikhonov solution after each epoch).

We observe that for random sampling, both `rrls` and `sTik` iterates contain undesirable uncertainties in the estimates. Although `rrls` iterates provide approximations to the Tikhonov solution, the main disadvantages are that the regularization parameter cannot be updated during the process and the iterates converge asymptotically to the unregularized solution. Hence, we disregard the `rrls` method and focus on `sTik` with *random cyclic sampling*, where λ can be updated via Λ_k.

3. Sampled regularization parameter selection methods

The ability to update the regularization parameter without sacrificing favorable convergence properties makes the `sTik` method appealing for massive inverse problems. However, sampled regularization parameter selection methods must be developed to enable proper updates Λ_k. Adapting regularization parameters during an iterative processes is not a new concept; however, much of the previous work in this area utilize projected systems, see e.g. [31, 45], or are specialized to applications such as denoising [27]. Another common approach is to consider the unregularized problem and to terminate the iterative process before noise contaminates the solution. This phenomenon is called semiconvergence, and selecting a good stopping iteration can be very difficult. There have been investigations into semiconvergence behavior of iterative methods such as Kaczmarz, e.g. [19].

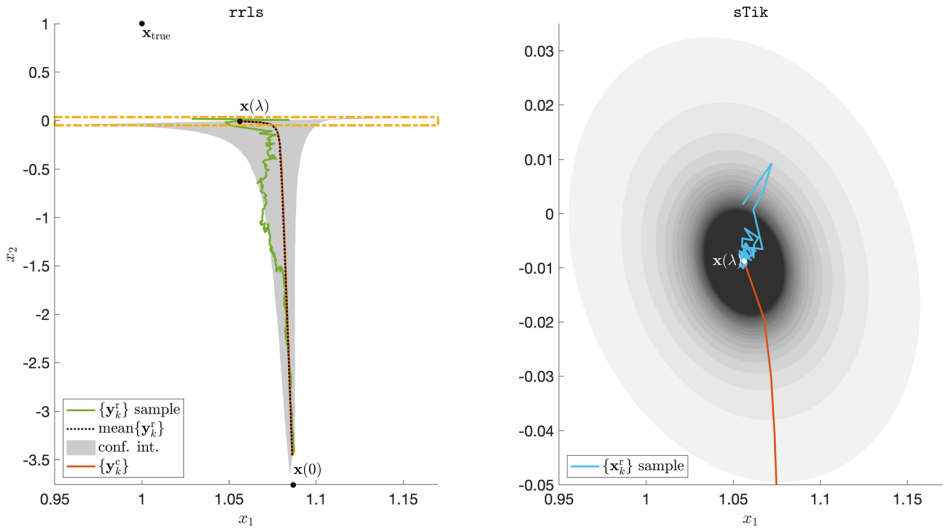

Figure 1. Illustration of convergence behaviors of `rrls` and `sTik` iterates. Shown in the left panel are the true solution $\mathbf{x}_{\text{true}}$, the unregularized solution $\mathbf{x}(0)$, the Tikhonov solution $\mathbf{x}(\lambda)$, and `rrls` iterates after multiple epochs. Both `rrls` with random sampling iterates $\{\mathbf{y}_k^{\text{r}}\}$ and `rrls` with random cyclic sampling iterates $\{\mathbf{y}_k^{\text{c}}\}$ converge asymptotically to the unregularized solution. In the right panel, we provide `sTik` with random sampling iterates $\{\mathbf{x}_k^{\text{r}}\}$ and confidence bounds. These iterates stay close to the Tikhonov solution. The axis for the right figure corresponds to the rectangular box in the left figure. The concentric gray circles represent the 95% confidence interval for these iterates after subsequent epochs.

Unfortunately, standard regularization parameter selection methods are not feasible in this setting because many of them require access to the full residual vector, $\mathbf{r}(\lambda) = \mathbf{A}\mathbf{x}(\lambda) - \mathbf{b}$, which is not available. In this section, we investigate variants of existing regularization parameter selection methods [4, 6, 49] that are based on the sample residual. In the following we assume that at the kth iteration, Λ_i for $i = 1, \ldots, k-1$ have been computed. Then the goal is to determine an appropriate update parameter Λ_k. From theorems 2.1 and 2.3, the kth `sTik` iterate can be represented as

$$\mathbf{x}_k(\lambda) = \mathbf{C}_k(\lambda)\mathbf{b}, \quad \text{where}$$
$$\mathbf{C}_k(\lambda) = \left(\left(\lambda + \sum_{i=1}^{k-1} \Lambda_i \right) \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A} \right)^{-1} \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top. \tag{14}$$

Similar to standard regularization parameter selection methods, we assume that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. For methods that require estimates of σ^2, there are various ways that one can obtain such an estimate, see e.g. [16, 49].

3.1. Sampled discrepancy principle

The basic idea of the *sampled discrepancy principle (sDP)* is that at the kth iteration, the goal is to select the parameter Λ_k so that the sum of squared residuals for the current sample

$\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k - \mathbf{b})\right\|_2^2$ is equal to $\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}\boldsymbol{\epsilon}\right\|_2^2$. Using properties of conditional expectation, we find

$$\begin{aligned}\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_{\text{true}} - \mathbf{b})\right\|_2^2 &= \mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}\boldsymbol{\epsilon}\right\|_2^2 \\ &= \mathbb{E}\,\mathbb{E}\left[\boldsymbol{\epsilon}^{\top}\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\boldsymbol{\epsilon}\,|\,\boldsymbol{\epsilon}\right] \\ &= \sigma^2\text{tr}\left(\mathbb{E}\,\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\right) \\ &= \sigma^2\ell,\end{aligned}$$

where $\text{tr}(\cdot)$ corresponds to the matrix trace function. Thus, at the kth iteration and for a given realization, we select λ such that

$$\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b})\right\|_2^2 \approx \gamma\sigma^2\ell,$$

where $\gamma > 1$ is a predetermined real number. For the sampled methods, we select λ_k that solves the optimization problem,

$$\min_{\lambda}\ \left(\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b})\|_2^2 - \gamma\sigma^2\ell\right)^2, \tag{15}$$

where $\gamma = 4$ as suggested in [26, 49] and σ^2 is the true noise variance.

3.2. Sampled unbiased predictive risk estimator

Next, we describe a method to select Λ_k based on a *sampled unbiased predictive risk estimator (sUPRE)*. The basic idea is to find Λ_k to minimize the sampled predictive risk,

$$\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{A}\mathbf{x}_{\text{true}})\right\|_2^2,$$

which is equivalent to

$$\mathbb{E}\left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b})\right\|_2^2 + 2\sigma^2\,\mathbb{E}\,\text{tr}\left(\mathbf{W}_{\tau(k)}\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\right) - \sigma^2\ell.$$

See appendix B.1 for details of the derivation. Then, similar to the approach used in the standard UPRE derivation, the parameter Λ_k is selected by finding a minimizer of the unbiased estimator for the sampled predictive risk,

$$U_k(\lambda) = \left\|\mathbf{W}_{\tau(k)}^{\top}(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b})\right\|_2^2 + 2\sigma^2\text{tr}\left(\mathbf{W}_{\tau(k)}^{\top}\mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\right) - \sigma^2\ell, \tag{16}$$

for a given realization.

3.3. Sampled generalized cross validation

Lastly, we describe the *sampled generalized cross validation (sGCV)* method for selecting Λ_k and point the interested reader to appendix B.2 for details of the derivation. The basic idea is to use a 'leave-one-out' cross validation approach to find a value of Λ_k, but the main differences compared to the standard GCV method are that at the kth iteration, we only have access to the

sample residual and the iterates only correspond to Tikhonov solutions with only partial data. The parameter λ_k is selected by finding a minimizer of the sGCV function,

$$G_k(\lambda) = \frac{\ell \left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}) \right\|_2^2}{\operatorname{tr}\left(\mathbf{I}_\ell - \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\right)^2} = \frac{\ell \left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}) \right\|_2^2}{\left(\ell - \operatorname{tr}\left(\mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\right)\right)^2}. \tag{17}$$

3.4. Example 2

In this example we investigate the behavior of the previously discussed sampled regularization parameter update strategies, i.e. sDP, sUPRE, and sGCV, for multiple ill-posed inverse problems from the Matlab matrix gallery and from Hansens' regularization tools toolbox [1, 2]. For simplicity, we set $m = n = 100$ and use the true solutions $\mathbf{x}_{\text{true}}$ that are provided by the toolbox. If no true solution is provided, we set $\mathbf{x}_{\text{true}} = \mathbf{1}$. We let $\mathbf{L} = \mathbf{I}_{100}$, and set $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, 0.01\,\mathbf{I}_{100})$. Sampling matrices $\mathbf{W}_j \in \mathbb{R}^{100\times 10}$ are given as $\mathbf{W}_j = [\mathbf{0}_{10(j-1)\times 10}; \mathbf{I}_{10}; \mathbf{0}_{10(10-j)\times 10}]$ for $j = 1, \ldots, 10$, such that $\mathbf{A}$ and $\mathbf{b}$ are sampled in 10 consecutive blocks. Here, we sample $\mathbf{W}$ in a random cyclic fashion and let σ^2 be the true noise variance for sDP and sUPRE.

We first consider the `prolate` example where $\mathbf{A}$ is an ill-conditioned Toeplitz matrix from Matlab's matrix gallery. In figure 2 we illustrate the asymptotic behavior of the sampled parameter selection strategies by plotting the number of epochs against the value of λ for sDP, sUPRE, and sGCV. For comparison, we provide the regularization parameter for the full problem corresponding to DP, UPRE, and GCV. DP and UPRE use the true noise variance, and γ is as above for DP. For comparison, we also provide the optimal parameter λ_{opt} for the full problem, which is the parameter that minimizes the two-norm of the error between the reconstruction and the true solution. This last approach is not possible in practice. We observe that with more iterations, the sampled regularization parameter selection methods tend to 'stabilize' in that after some point, they do not change much. The sDP regularization parameter stabilizes near the DP parameter for the full problem, but both sUPRE and sGCV stabilize closer to the optimal regularization parameter.

While we observe similar results for other test problems (results not shown), the sampled regularization parameters may not necessarily be close to the corresponding parameter for the full system. Nevertheless, the sampled regularization parameter selection methods often lead to appropriate reconstructions $\mathbf{x}_k(\lambda)$ after a moderate number of iterations. Next, we investigate the relative reconstruction error $\|\mathbf{x}_k(\lambda) - \mathbf{x}_{\text{true}}\|_2 / \|\mathbf{x}_{\text{true}}\|_2$ of sampled regularization methods after *one* epoch (corresponding to $k = 10$). Figure 3 illustrates results from four test problems (`prolate`, `baart`, `shaw`, and `gravity`). First note that by theorem 2.3, all solutions are Tikhonov solutions for a λ determined by the method, hence all relative reconstruction errors lie on a curve of relative errors for Tikhonov solutions. We note that the above regularization parameter selection methods (including the standard DP, UPRE, and GCV) can only provide empirical estimations. However, we observe that in terms of relative reconstruction errors, our sampled regularization parameter selection methods perform reasonably well on the test problems.

As we have shown, our sampled regularization parameter selection methods can be used to update the regularization parameter in the `sTik` method, where the main benefit is the favorable convergence property. In the next section, we turn our attention to problems where it may be infeasible to construct or work with the $n \times n$ matrix $\mathbf{B}_k$. Although reduced models

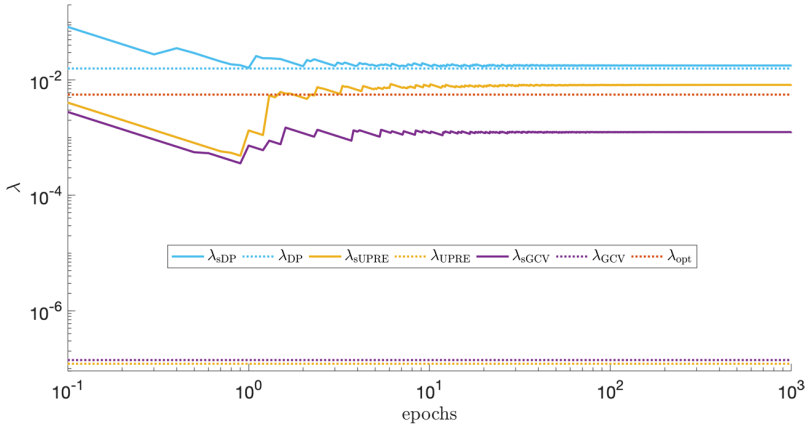

Figure 2. 'Asymptotic' behavior of the sampled regularization parameter selection methods for the `prolate` example. Corresponding regularization parameters computed using the full data are provided as horizontal lines for comparison.

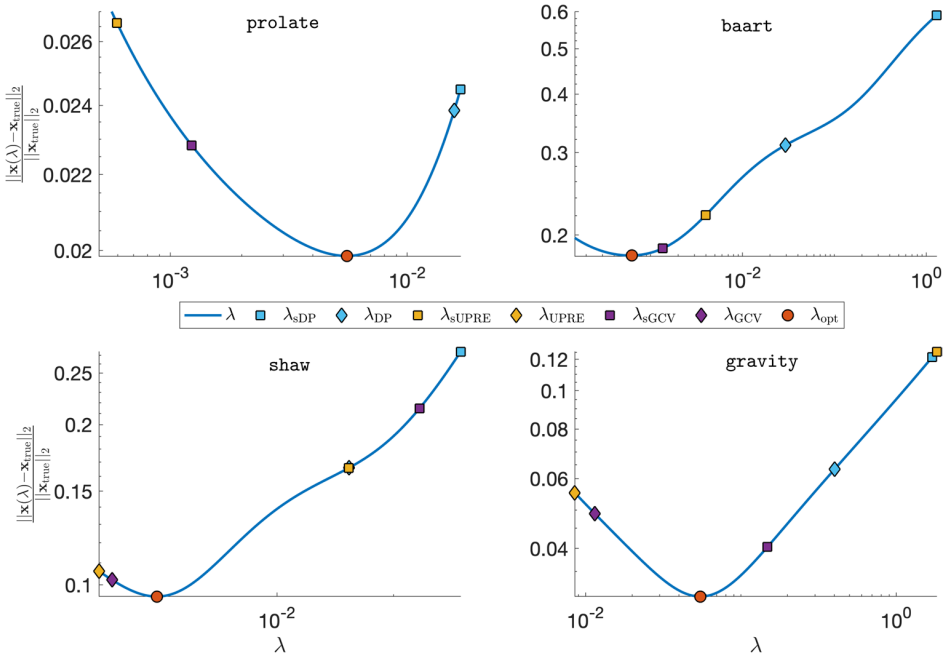

Figure 3. Relative reconstruction errors of the sampled and full regularization methods for four test problems `prolate`, `baart`, `shaw`, and `gravity`. All solutions lie on the solid line, which corresponds to relative errors for Tikhonov solutions. Note that the UPRE and GCV estimation in the `prolate` and `baart` test problem underperform significantly and are therefore omitted. The relative errors for λ_{sUPRE} and λ_{DP} coincide in the `shaw` example.

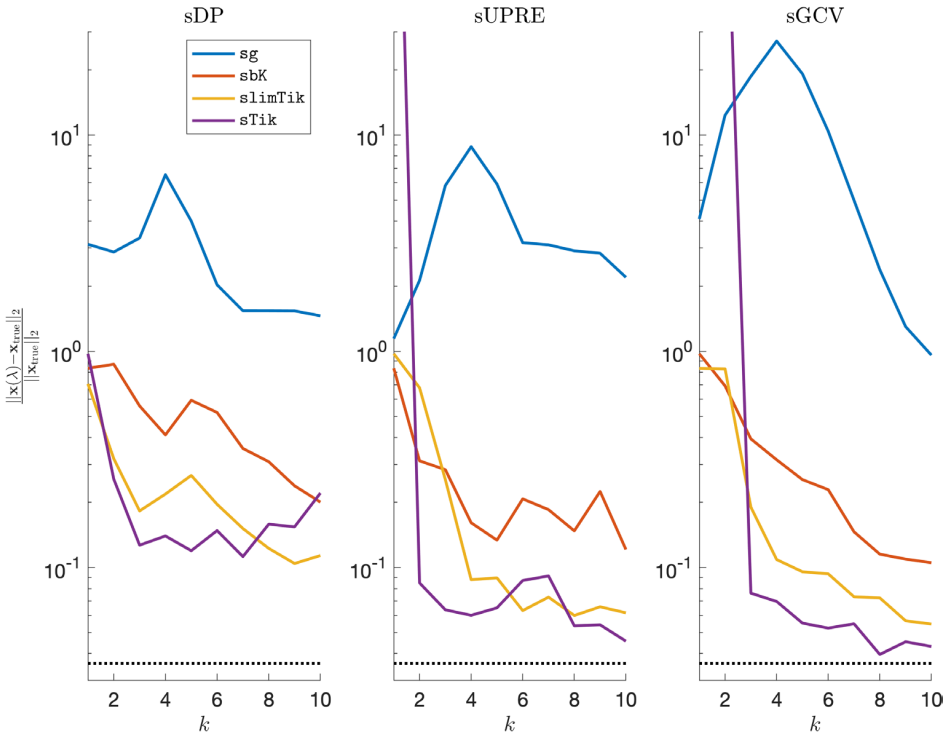

Figure 4. Comparison of relative reconstruction errors for `sg`, `sbK`, `slimTik`, and `sTik` iterates for `gravity` using various sampled regularization parameter selection methods for the first 10 iterations, i.e. one epoch. We compare sDP, sUPRE, and sGCV. The horizontal black line is the relative error corresponding to the optimal regularization parameter for the full problem, which is not feasible to obtain in practice.

or subspace projection methods may be used to reduce the number of unknowns, obtaining a realistic basis for the solution may be difficult.

4. Numerical results

In this section, we address some of the computational concerns and demonstrate our methods on a large imaging problem. First, we reformulate the updates as solutions to least squares problems so that iterative methods can be used to compute approximations efficiently. In addition to being computationally feasible, these methods can take advantage of the adaptive regularization parameter selection methods described in section 3.

These methods are based on the `sTik` method. In particular, we consider a sampled gradient (`sg`) method where the iterates are defined as (5) where $\mathbf{B}_k = \left(\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{I}_n\right)^{-1}$ and a sampled block Kaczmarz (`sbK`) method where the iterates are defined as (5) with $\mathbf{B}_k = \left(\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{A}_k^\top \mathbf{A}_k\right)^{-1}$. Notice that this corresponds to including only the current block $\mathbf{A}_k$. We also consider a *limited-memory* version of `sTik` called `slimTik`, which we describe below. First notice that the kth `sTik` iterate is given by $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{s}_k$ where

Figure 5. Super-resolution imaging example. On the left is the true high-resolution image, and on the right are three sample low-resolution images. The red-box corresponds to sub-images shown in figure 7.

$$\mathbf{s}_k = \arg\min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{k-1} \\ \mathbf{A}_k \\ \sqrt{\sum_{i=1}^{k} \Lambda_i}\mathbf{L} \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k \\ \frac{\Lambda_k}{\sqrt{\sum_{i=1}^{k} \Lambda_i}}\mathbf{L}\mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2 .$$

With this reformulation, we must solve a least squares problem with matrix $\begin{bmatrix}\mathbf{A}_1^\top & \cdots & \mathbf{A}_k^\top\end{bmatrix}^\top$, which grows with each iteration. Thus, we select a memory parameter $r \in \mathbb{N}_0$ and define $\mathbf{M}_k = \begin{bmatrix}\mathbf{A}_{k-r}^\top & \cdots & \mathbf{A}_{k-1}^\top\end{bmatrix}^\top \in \mathbb{R}^{r\ell \times n}$ and $\mathbf{A}_{k-r} = \mathbf{0}$ for non-positive integers $k - r$. Then `slimTik` iterates are given as $\mathbf{x}_k = \mathbf{x}_{k-1} - \tilde{\mathbf{s}}_k$ where

$$\tilde{\mathbf{s}}_k = \arg\min_{\mathbf{s}} \left\| \begin{bmatrix} \mathbf{M}_k \\ \mathbf{A}_k \\ \sqrt{\sum_{i=1}^{k} \Lambda_i}\mathbf{L} \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_k\mathbf{x}_{k-1} - \mathbf{b}_k \\ \frac{\Lambda_k}{\sqrt{\sum_{i=1}^{k} \Lambda_i}}\mathbf{L}\mathbf{x}_{k-1} \end{bmatrix} \right\|_2^2 . \tag{18}$$

In the case where $r = 0$, `slimTik` and `sbK` iterates are identical. First, we investigate the performance of `sg`, `sbK`, and `slimTik` while taking advantage of the regularization parameter update described in section 3. We use the `gravity` example from regularization tools,

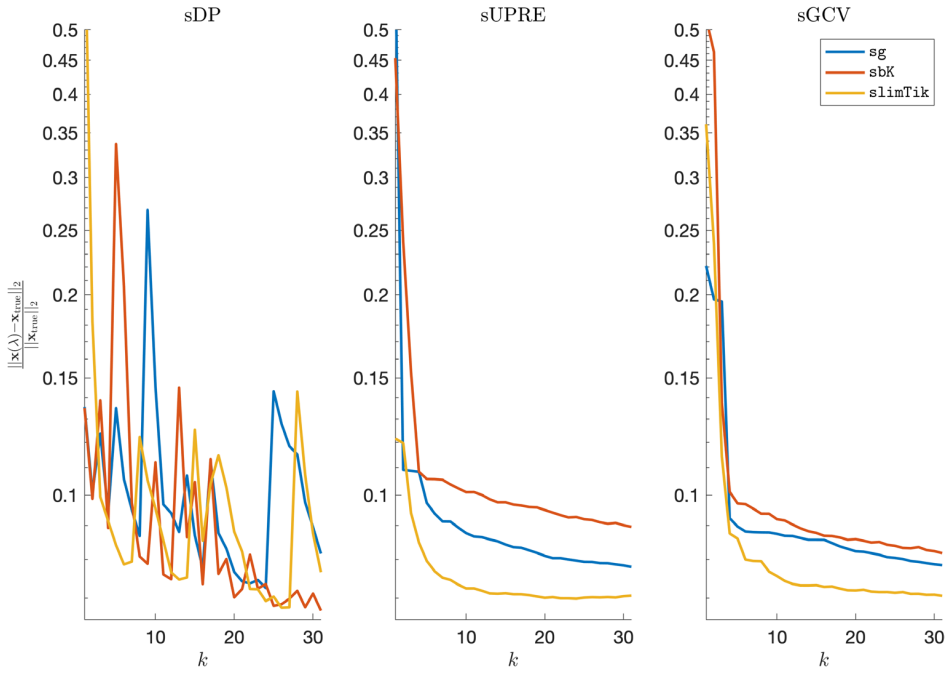

Figure 6. Relative reconstruction errors for the super-resolution imaging example for one epoch. We note that sUPRE and sGCV produce good reconstructions. Additionally, `slimTik` produces a smaller relative reconstruction error, since it is using more curvature information.

where $\mathbf{A} \in \mathbb{R}^{1000\times 1000}$, $\mathbf{L} = \mathbf{I}_{1000}$, and the noise level defined as $\|\boldsymbol{\epsilon}\|_2 \,/\, \|\mathbf{A}\mathbf{x}_{\text{true}}\|_2$ is 0.01. The samples consist of 10 blocks, each comprised of 100 consecutive rows of $\mathbf{A}$. The initial guess for the regularization parameter is chosen to be 0.1 (the optimal overall regularization parameter in this example is approximately 0.0196), and we iterate for one epoch.

In figure 4 we provide the relative reconstruction errors per iteration for `sg`, `sbK`, `slimTik`, and `sTik`. Overall, we notice a correspondence between the amount of curvature information used to approximate the Hessian and an improvement in the relative reconstruction error. Including more curvature results in greater computational costs and storage requirements, e.g. `sTik` may be infeasible for very large problems, but the number of row accesses is the same for each method. In terms of regularization parameter selection methods, sGCV performs better than sUPRE and sDP for this example. The relative reconstruction error corresponding to the best overall Tikhonov solution is provided as the horizontal line. Although the results are not shown here, we note that without regularization, the relative reconstruction errors will become very large for all of these methods due to semiconvergence.

Having demonstrated that regularization parameter update methods can be incorporated in a variety of stochastic optimization methods, we investigate the performance of these limited-memory methods for super-resolution image reconstruction. The basic goal of super-resolution imaging is to reconstruct an $n \times n$ high-resolution image represented by a vector $\mathbf{x}_{\text{true}} \in \mathbb{R}^{n^2}$ given M low-resolution images of size $\ell \times \ell$ represented by $\mathbf{b}_1 \cdots, \mathbf{b}_M$, where $\mathbf{b}_i \in \mathbb{R}^{\ell^2}$. The forward model for each low-resolution image is given as

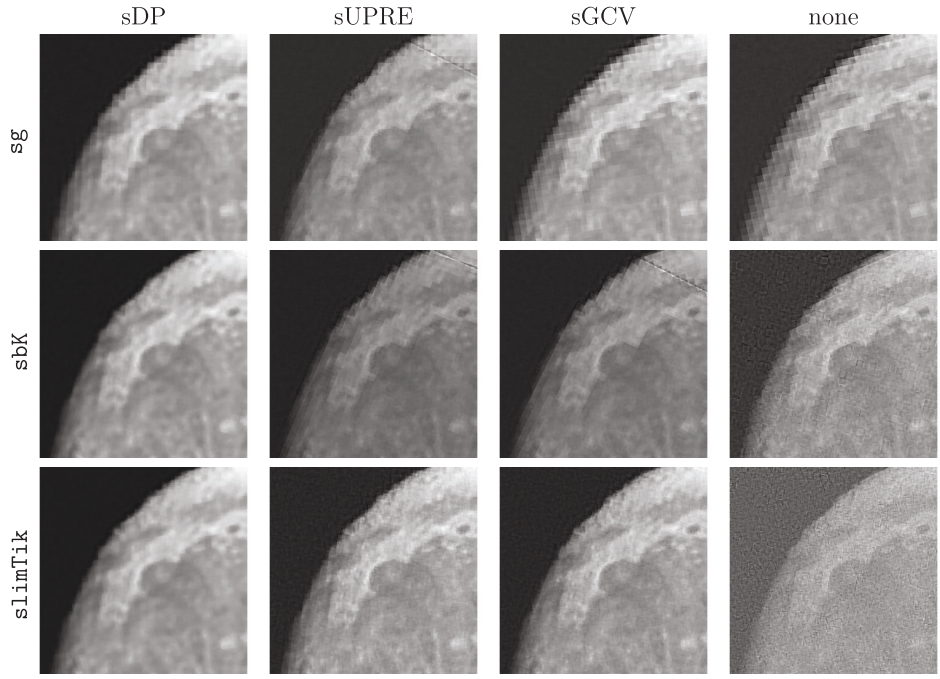

Figure 7. Sub-images of the reconstructed images for the super-resolution imaging example. Reconstructions correspond to `sg`, `sbK`, and `slimTik` with regularization parameter updates computed using sDP, sUPRE, and sGCV. For comparison, we provide reconstructions corresponding to no regularization, i.e. $\lambda = 0$.

$$\mathbf{b}_i = \mathbf{R}\mathbf{S}_i\mathbf{x}_{\text{true}} + \boldsymbol{\epsilon}_i,$$

where $\mathbf{R} \in \mathbb{R}^{\ell^2 \times n^2}$ is a restriction matrix, $\mathbf{S}_i \in \mathbb{R}^{n^2 \times n^2}$ represents an affine transformation that may account for shifts, rotations, and scalar multiplications, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}\left(\mathbf{0}_{\ell^2}, \sigma^2\mathbf{I}_{\ell^2}\right)$. To reconstruct a high-resolution image, we solve the Tikhonov problem,

$$\min_{\mathbf{x}} \left\| \begin{bmatrix} \mathbf{R}\mathbf{S}_1 \\ \vdots \\ \mathbf{R}\mathbf{S}_M \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_M \end{bmatrix} \right\|_2^2 + \lambda \|\mathbf{L}\mathbf{x}\|_2^2 .$$

For cases where the low-resolution images are being streamed or where the number of low-resolution images is very large, standard iterative methods may not be feasible. Furthermore, it can be very challenging to determine a good choice of λ prior to solution computation [15, 28, 42].

For our example, we have 30 images of size 128×128, and we wish to reconstruct a high-resolution image of size 2048×2048. In figure 5, we provide the true high-resolution image of the moon [13] and three of the low-resolution images. Here, $\mathbf{A}_i = \mathbf{R}\mathbf{S}_i \in \mathbb{R}^{128^2 \times 2048^2}$. Due to the inherent partitioning of the problem, we take $\mathbf{W}_i^\top \in \mathbb{R}^{128^2 \times 30 \cdot 128^2}$ to be a matrix such that $\mathbf{W}_i^\top \mathbf{A} = \mathbf{A}_i$; these $\mathbf{W}_i$ matrices are never computed. For the simulated low-resolution images, Gaussian white noise is added such that the noise level for each image is 0.01 and

take $\mathbf{L} = \mathbf{I}_{2048^2}$. Notice that the size of the matrix $\mathbf{A}$ is $491\,520 \times 4194\,304$, and holding $\mathbf{A}$ in computer memory is impractical despite its sparse structure.

We compare the performances of `sg`, `sbK`, and `slimTik`, including our sampled regularization parameter update methods sDP, sUPRE, and sGCV. The true noise variance is used for sDP and sUPRE, and the memory parameter for `slimTik` is $r = 2$. Each iteration of `sbK` and `slimTik` requires a linear solve, which can be handled efficiently by reformulating the problem as a least squares problem as in equation (18), and using standard techniques such as LSQR [40, 41]. These iterative methods can also be used to update the regularization parameter. Furthermore, we use the Hutchison trace estimator to efficiently evaluate the trace term in sGCV and sUPRE, see (16) and (17). More specifically, rather than compute 128^2 linear solves, we note that if $\mathbf{v}$ is a random variable such that $\mathbb{E}\,\mathbf{v}\mathbf{v}^\top = \mathbf{I}_{128^2}$, then

$$\mathrm{tr}\left(\mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\right) = \mathbb{E}\mathbf{v}^\top \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda)\mathbf{W}_{\tau(k)}\mathbf{v}.$$

Here we use the Rademacher distribution where the entries of $\mathbf{v}$ are $v_i = \pm 1$ with equal probability. We use a single realization of $\mathbf{v}$ to approximate the trace, hence resulting in just one linear solve [5, 23, 46].

Relative reconstruction errors are provided in figure 6, and sub-images of the reconstructions are provided in figure 7. We observe that, in general, sDP errors are more erratic than sUPRE and sGCV errors. Notice that for sUPRE and sGCV, `sbK` produces higher reconstruction errors compared to `sg`, which may be attributed to insufficient global curvature information. Furthermore, we observe that `slimTik` reconstructions contain more details than `sg` and `sbK` reconstructions.

5. Conclusions

In this work we describe iterative sampled Tikhonov methods for solving inverse problems for which it is not feasible to access the data all-at-once. Such methods are necessary when handling data sets that do not fit in memory and also can naturally handle streaming data problems.

We investigate two iterative methods, `rrls` and `sTik`, and show that under various sampling schemes, `rrls` iterates converge asymptotically to the unregularized solution while `sTik` iterates converge to a Tikhonov-regularized solution. Although the sampling mechanisms we discuss do not play a role in the asymptotic convergence, they do allow for interesting interpretations. In particular, for random cyclic sampling we can characterize the iterates as Tikhonov solutions after every epoch, providing insight into the path that the iterates take towards the solution. For iterative methods where the regularization parameter can be updated during the iterative process (e.g. `sTik`), we describe sampled variants of existing regularization parameter selection methods to update the parameter. Using a number of well-known data sets, we show empirically that sampled Tikhonov methods with automatic regularization parameter updates can be competitive. For very large inverse problems, we describe a limited-memory version of `sTik`, and we demonstrate the efficacy of the limited-memory approach on a standard benchmark dataset as well as on a streaming super-resolution image reconstruction problem.

Future directions of research include developing an asymptotic analysis of `slimTik` and a non-asymptotic analysis of the general sampling algorithms. This would involve bounding the mean square error at a fixed iteration k, which may help to explain the quick initial convergence seen in the numerical experiments. Another open question is how to accelerate

convergence by selecting $\mathbf{W}_\tau$ to sample important parts of the problem, e.g. using sketching matrices [17, 18]. Finally, extensions to nonlinear inverse problems would require more advanced convergence analyses and further algorithmic developments, e.g. incorporating adaptive regularization parameter selection within stochastic LBFGS [12, 36].

Acknowledgments

This work was partially supported by NSF DMS 1723005 (J Chung, M Chung, Tenorio) and NSF DMS 1654175 (J Chung).

Appendix A. Proofs for section 2

Proof of theorem 2.1. For (ii), note that the solution of the least squares problem (7) is given by

$$\mathbf{x}(\lambda_k) = \mathbf{B}_k \sum_{i=1}^{k} \mathbf{A}_i^\top \mathbf{b}_i.$$

Noticing the relationship $\mathbf{B}_k^{-1} = \mathbf{B}_{k-1}^{-1} + \mathbf{A}_k^\top \mathbf{A}_k + \Lambda_k \mathbf{L}^\top \mathbf{L}$, we get the following equivalencies for the `sTik` iterates

$$\begin{aligned}
\mathbf{x}_k &= \mathbf{x}_{k-1} - \mathbf{B}_k \left(\mathbf{A}_k^\top (\mathbf{A}_k \mathbf{x}_{k-1} - \mathbf{b}_k) + \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1} \right) \\
&= \mathbf{B}_k \left(\mathbf{B}_k^{-1} \mathbf{x}_{k-1} - \mathbf{A}_k^\top \mathbf{A}_k \mathbf{x}_{k-1} + \mathbf{A}_k^\top \mathbf{b}_k - \Lambda_k \mathbf{L}^\top \mathbf{L} \mathbf{x}_{k-1} \right) \\
&= \mathbf{B}_k \left(\mathbf{B}_{k-1}^{-1} \mathbf{x}_{k-1} + \mathbf{A}_k^\top \mathbf{b}_k \right) = \mathbf{B}_k \sum_{i=1}^{k} \mathbf{A}_i^\top \mathbf{b}_i = \mathbf{x}(\lambda_k).
\end{aligned}$$

A similar proof can be made for (i). □

Proof of theorem 2.2.

(i) From theorem 2.1 for any $k \in \mathbb{N}$ we have

$$\begin{aligned}
\mathbf{y}_k &= \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A} \right)^{-1} \left(\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{b} + \lambda \mathbf{L}^\top \mathbf{L} \mathbf{y}_0 \right) \\
&= \left(\frac{\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A}}{k} \right)^{-1} \left(\frac{\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{b} + \lambda \mathbf{L}^\top \mathbf{L} \mathbf{y}_0}{k} \right).
\end{aligned}$$

Using the fact that $\mathbb{E}\, \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top = \frac{\ell}{m} \mathbf{I}_m$ (see equation (8)), by the law of large numbers and Slutsky's theorem for a.s. convergence [49]

$$\frac{\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{b} + \lambda \mathbf{L}^\top \mathbf{L} \mathbf{y}_0}{k} \xrightarrow{\text{a.s.}} \frac{\ell}{m} \mathbf{A}^\top \mathbf{b},$$

and

$$\left(\frac{\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A}}{k}\right)^{-1} \xrightarrow{\text{a.s.}} \frac{m}{\ell} \left(\mathbf{A}^\top \mathbf{A}\right)^{-1}$$

and therefore

$$\mathbf{y}_k \xrightarrow{\text{a.s.}} \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{x}(0).$$

(ii) In a similar fashion, for any $k \in \mathbb{N}$ we have

$$\mathbf{x}_k = \left(\frac{\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A}}{k}\right)^{-1} \left(\frac{\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{b}}{k}\right).$$

Using the fact that $\mathbb{E}\, \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top = \frac{\ell}{m} \mathbf{I}_m$ and $\lim_{k\to\infty} \frac{\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L}}{k} = \frac{\ell}{m} \lambda \mathbf{L}^\top \mathbf{L}$, we have

$$\frac{\sum_{i=1}^{k} \mathbf{A}^\top \mathbf{W}_i \mathbf{W}_i^\top \mathbf{b}}{k} \xrightarrow{\text{a.s.}} \frac{\ell}{m} \mathbf{A}^\top \mathbf{b}$$

and

$$\left(\frac{\sum_{i=1}^{k} \Lambda_i \mathbf{L}^\top \mathbf{L} + \mathbf{A}^\top \mathbf{W}_i \mathbf{W}_i^\top \mathbf{A}}{k}\right)^{-1} \xrightarrow{\text{a.s.}} \frac{m}{\ell} \left(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{L}^\top \mathbf{L}\right)^{-1},$$

and thus we conclude that

$$\mathbf{x}_k \xrightarrow{\text{a.s.}} \left(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{L}^\top \mathbf{L}\right)^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{x}(\lambda).$$

□

Proof of theorem 2.3. Notice that for random cyclic sampling schemes and for any iteration jM, $\sum_{i=1}^{jM} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{A} = j \mathbf{A}^\top \mathbf{A}$ and $\sum_{i=1}^{jM} \mathbf{A}^\top \mathbf{W}_{\tau(i)} \mathbf{W}_{\tau(i)}^\top \mathbf{b} = j \mathbf{A}^\top \mathbf{b}$ are deterministic. Hence

$$\mathbf{y}_{jM} = j \left(\lambda \mathbf{L}^\top \mathbf{L} + j \mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} = \left(\frac{\lambda}{j} \mathbf{L}^\top \mathbf{L} + \mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{x}\left(\frac{1}{j}\lambda\right)$$

and

$$\mathbf{x}_{jM} = j \left(\lambda_{jM} \mathbf{L}^\top \mathbf{L} + j \mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} = \left(\frac{\lambda_{jM}}{j} \mathbf{L}^\top \mathbf{L} + \mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{x}\left(\frac{1}{j}\lambda_{jM}\right).$$

□

Appendix B. Derivations for sampled UPRE and sampled GCV

In this section, we provide derivations for (16) and (17). To estimate the overall regularization parameter λ at the kth iteration we are just required to update Λ_k since the estimate λ is uniquely determined by the preceding Λ_i's, $i = 1, \ldots, k-1$ and Λ_k. Hence, for ease of notation we will drop the iteration count on λ.

B.1. Derivation of the sampled UPRE

The basic idea is to find Λ_k by minimizing an estimate of the predictive error. Let the *sampled predictive error* be given by

$$P(\lambda) = \left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{A}\mathbf{x}_{\text{true}}) \right\|_2^2.$$

Using the notation from (14), the expected sampled predictive error, $\mathbb{E}\,P(\lambda)$, can be written as

$$\mathbb{E}\left\| \mathbf{W}_{\tau(k)}^\top \left(\mathbf{A}\mathbf{C}_k(\lambda) - \mathbf{I}_m\right) \mathbf{A}\mathbf{x}_{\text{true}} \right\|_2^2 + \sigma^2 \mathbb{E}\,\text{tr}\Big(\mathbf{C}_k(\lambda)^\top \mathbf{A}^\top \mathbf{W}_{\tau(k)} \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda) \Big), \tag{B.1}$$

where the mixed term vanishes due to independence of $\mathbf{W}_{\tau(1)}, \ldots \mathbf{W}_{\tau(k)}$ and $\boldsymbol{\epsilon}$ and since $\mathbb{E}\boldsymbol{\epsilon} = \mathbf{0}$. Similar to the derivation for standard UPRE, the predictive error is not computable in practice since $\mathbf{x}_{\text{true}}$ is not available. Thus, we perform a similar calculation for the expected sampled residual norm,

$$\begin{aligned} \mathbb{E}\,\left\| \mathbf{W}_{\tau(k)}^\top \left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right) \right\|_2^2 &= \mathbb{E}\left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{C}_k(\lambda) - \mathbf{I}_m)\mathbf{b} \right\|_2^2 \\ &= \mathbb{E}\left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{C}_k(\lambda) - \mathbf{I}_m)\mathbf{A}\mathbf{x}_{\text{true}} \right\|_2^2 + \mathbb{E}\left\| \mathbf{W}_{\tau(k)}^\top (\mathbf{A}\mathbf{C}_k(\lambda) - \mathbf{I}_m)\boldsymbol{\epsilon} \right\|_2^2. \end{aligned} \tag{B.2}$$

Next, notice that using the trace lemma for symmetric matrices [6], the second term in (B.2) can be written as

$$\sigma^2 \Big(\mathbb{E}\,\text{tr}\Big(\mathbf{C}_k(\lambda)^\top \mathbf{A}^\top \mathbf{W}_{\tau(k)} \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda) \Big) - 2\,\mathbb{E}\,\text{tr}\Big(\mathbf{W}_{\tau(k)} \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda) \Big) + \ell \Big). \tag{B.3}$$

Combining (B.1) with (B.2) and (B.3), we get

$$\mathbb{E}P(\lambda) = \mathbb{E}\left\| \mathbf{W}_{\tau(k)}^\top \left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right) \right\|_2^2 + 2\sigma^2\,\mathbb{E}\,\text{tr}\Big(\mathbf{W}_{\tau(k)} \mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda) \Big) - \sigma^2\ell.$$

Finally for a given realization, we get an estimator for the predictive risk

$$U_k(\lambda) = \left\| \mathbf{W}_{\tau(k)}^\top \left(\mathbf{A}\mathbf{x}_k(\lambda) - \mathbf{b}\right) \right\|_2^2 + 2\sigma^2 \text{tr}\Big(\mathbf{W}_{\tau(k)}^\top \mathbf{A}\mathbf{C}_k(\lambda) \mathbf{W}_{\tau(k)} \Big) - \sigma^2\ell,$$

which is equivalent to (16).

B.2. Derivation of the sampled GCV

We derive the sampled generalized cross validation function, following a similar derivation of the cross validation and generalized cross validation function found in [22]. For notational simplicity, we denote $\mathbf{A}_{\tau(i)} = \mathbf{W}_{\tau(i)}^\top \mathbf{A}$ and $\mathbf{b}_{\tau(i)} = \mathbf{W}_{\tau(i)}^\top \mathbf{b}$. Then, notice that the kth iterate of `sTik`, which is given by $\mathbf{x}_k(\lambda) = \mathbf{C}_k(\lambda)\mathbf{b}$ is the solution to the following problem,

$$\min_{\mathbf{x}} \left\| \mathbf{A}_{\tau(k)}\mathbf{x} - \mathbf{b}_{\tau(k)} \right\|_2^2 + \lambda \left\| \mathbf{L}\mathbf{x} \right\|_2^2 + \left\| \begin{bmatrix} \mathbf{A}_{\tau(1)} \\ \vdots \\ \mathbf{A}_{\tau(k-1)} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b}_{\tau(1)} \\ \vdots \\ \mathbf{b}_{\tau(k-1)} \end{bmatrix} \right\|_2^2.$$

To derive sampled GCV, at the kth iterate, define,

$$\mathbf{E}_j = \mathbf{I}_\ell - \mathbf{e}_j^\top \mathbf{e}_j.$$

Here $\mathbf{e}_j$ is the jth column of the identity matrix. Our goal is to find $\mathbf{x}_{[j]}(\lambda)$, which is the solution to

$$\min_{\mathbf{x}} \left\|\mathbf{E}_j\left(\mathbf{A}_{\tau(k)}\mathbf{x} - \mathbf{b}_{\tau(k)}\right)\right\|_2^2 + \lambda\left\|\mathbf{L}\mathbf{x}\right\|_2^2 + \left\| \begin{bmatrix} \mathbf{A}_{\tau(1)} \\ \vdots \\ \mathbf{A}_{\tau(k-1)} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{b}_{\tau(1)} \\ \vdots \\ \mathbf{b}_{\tau(k-1)} \end{bmatrix} \right\|_2^2.$$

Then, the sampled cross-validation estimate for λ minimizes the average error,

$$V_k(\lambda) = \frac{1}{\ell}\sum_{j=1}^{\ell}\left(\mathbf{e}_j^\top \mathbf{b}_{\tau(k)} - \mathbf{e}_j^\top \mathbf{A}_{\tau(k)}\mathbf{x}_{[j]}(\lambda)\right)^2.$$

Using the normal equations and the fact that $\mathbf{E}_j^\top \mathbf{E}_j = \mathbf{E}_j$, an explicit expression for $\mathbf{x}_{[j]}(\lambda)$ is given as

$$\begin{aligned}\mathbf{x}_{[j]}(\lambda) &= \left(\mathbf{A}_{\tau(k)}^\top \mathbf{E}_j^\top \mathbf{E}_j \mathbf{A}_{\tau(k)} + \lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k-1} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1} \left(\mathbf{A}_{\tau(k)}^\top \mathbf{E}_j^\top \mathbf{E}_j \mathbf{b}_{\tau(k)} + \sum_{i=1}^{k-1} \mathbf{A}_{\tau(i)}^\top \mathbf{b}_{\tau(i)}\right) \\ &= \left(\mathbf{B}_k(\lambda)^{-1} - \mathbf{A}_{\tau(i)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{A}_{\tau(i)}\right)^{-1} \left(\sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{b}_{\tau(i)} - \mathbf{A}_{\tau(k)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{b}_{\tau(k)}\right),\end{aligned}$$

where $\mathbf{B}_k(\lambda) = \left(\lambda \mathbf{L}^\top \mathbf{L} + \sum_{i=1}^{k} \mathbf{A}_{\tau(i)}^\top \mathbf{A}_{\tau(i)}\right)^{-1}$. Next defining $t_{jj} = \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{B}_k(\lambda) \mathbf{A}_{\tau(k)}^\top \mathbf{e}_j$ and using the Sherman–Morrison–Woodbury formula, we get

$$\left(\mathbf{B}_k(\lambda)^{-1} - \mathbf{A}_{\tau(i)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{A}_{\tau(i)}\right)^{-1} = \frac{1}{1 - t_{jj}}\left((1 - t_{jj})\mathbf{B}_k(\lambda) + \mathbf{B}_k(\lambda)\mathbf{A}_{\tau(k)}^\top \mathbf{e}_j \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{B}_k(\lambda)\right)$$

and after some algebraic manipulations, we arrive at

$$\mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{x}_{[j]}(\lambda) = \frac{1}{1 - t_{jj}}\left(\mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{C}_k(\lambda)\mathbf{b} - t_{jj}\mathbf{e}_j^\top \mathbf{b}_{\tau(k)}\right).$$

Thus,

$$\mathbf{e}_j^\top \mathbf{b}_{\tau(k)} - \mathbf{e}_j^\top \mathbf{A}_{\tau(k)} \mathbf{x}_{[j]}(\lambda) = \frac{1}{1 - t_{jj}} \mathbf{e}_j^\top \left(\mathbf{b}_{\tau(k)} - \mathbf{A}_{\tau(k)} \mathbf{x}_k(\lambda)\right),$$

and we can write the sampled cross-validation function as

$$V_k(\lambda) = \frac{1}{\ell}\left\|\mathbf{D}_k(\lambda)(\mathbf{b}_{\tau(k)} - \mathbf{A}_{\tau(k)}\mathbf{x}_k(\lambda))\right\|_2^2,$$

where $\mathbf{D}_k(\lambda) = \operatorname{diag}\left(\frac{1}{1-t_{11}}, \ldots, \frac{1}{1-t_{\ell\ell}}\right)$. The extension from the sampled cross-validation to the sampled generalized cross validation function is analogous to the generalization process from cross-validation to GCV provided in [22].

ORCID iDs

Julianne Chung https://orcid.org/0000-0002-6760-4736
Matthias Chung https://orcid.org/0000-0001-7822-4539
David Kozak https://orcid.org/0000-0002-3795-5834

References

[1] Matlab test matrices gallery www.mathworks.com/help/matlab/ref/gallery.html (Accessed: 16 November 2018)
[2] Regularization tools version 4.1 (for Matlab) www.imm.dtu.dk/∼pcha/Regutools/ (Accessed: 16 November 2018)
[3] Andersen M S and Hansen P C 2014 Generalized row-action methods for tomographic imaging *Numer. Algorithms* **67** 121–44
[4] Aster R C, Borchers B and Thurber C H 2018 *Parameter Estimation and Inverse Problems* (New York: Elsevier) (https://doi.org/10.1016/C2009-0-61134-X)
[5] Avron H and Toledo S 2011 Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix *J. ACM* **58** 8
[6] Bardsley J M 2018 *Computational Uncertainty Quantification for Inverse Problems* (Philadelphia, PA: SIAM)
[7] Björck A 1996 *Numerical Methods for Least Squares Problems* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9781611971484)
[8] Bottou L 1998 Online learning and stochastic approximations *Online Learning in Neural Networks* (Cambridge: Cambridge University Press) ch 2, pp 9–42
[9] Bottou L and Cun Y L 2004 Large scale online learning *Advances in Neural Information Processing Systems* pp 217–24
[10] Bottou L, Curtis F E and Nocedal J 2018 Optimization methods for large-scale machine learning *SIAM Rev.* **60** 223–311
[11] Buccini A, Donatelli M and Reichel L 2017 Iterated Tikhonov regularization with a general penalty term *Numer. Linear Algebr. Appl.* **24** e2089
[12] Byrd R H, Hansen S L, Nocedal J and Singer Y 2016 A stochastic quasi-Newton method for large-scale optimization *SIAM J. Optim.* **26** 1008–31
[13] NASA Goddard Space Flight Center 2014 Image from NASA's lunar reconnaissance orbitor https://lunar.gsfc.nasa.gov/imagesandmultimedia.html
[14] Chung J, Chung M, Slagel J T and Tenorio L 2017 Stochastic Newton and quasi-Newton methods for large linear least-squares problems (arXiv:1702.07367)
[15] Chung J, Haber E and Nagy J 2006 Numerical methods for coupled super-resolution *Inverse Problems* **22** 1261
[16] Donoho D L 1995 De-noising by soft-thresholding *IEEE Trans. Inf. Theory* **41** 613–27
[17] Drineas P, Magdon-Ismail M, Mahoney M W and Woodruff D P 2012 Fast approximation of matrix coherence and statistical leverage *J. Mach. Learn. Res.* **13** 3475–506
[18] Drineas P, Mahoney M W, Muthukrishnan S and Sarlós T 2011 Faster least squares approximation *Numer. Math.* **117** 219–49
[19] Elfving T, Hansen P C and Nikazad T 2014 Semi-convergence properties of Kaczmarz's method *Inverse Problems* **30** 055007
[20] Engl H W 1987 On the choice of the regularization parameter for iterated Tikhonov regularization of ill-posed problems *J. Approx. Theory* **49** 55–63
[21] Escalante R and Raydan M 2011 *Alternating Projection Methods* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/9781611971941)
[22] Golub G H, Heath M and Wahba G 1979 Generalized cross-validation as a method for choosing a good ridge parameter *Technometrics* **21** 215–23
[23] Haber E, Chung M and Herrmann F 2012 An effective method for parameter estimation with PDE constraints with multiple right-hand sides *SIAM J. Optim.* **22** 739–57
[24] Hanke M and Groetsch C W 1998 Nonstationary iterated tikhonov regularization *J. Optim. Theory Appl.* **98** 37–53

[25] Hansen P C 2010 *Discrete Inverse Problems: Insight and Algorithms* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9780898718836)
[26] Hansen P C, Nagy J G and O'Leary D P 2006 *Deblurring Images: Matrices, Spectra, and Filtering* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9780898718874)
[27] Hashemi S, Beheshti S, Cobbold R and Paul N 2015 Adaptive updating of regularization parameters *Signal Process.* **113** 228–33
[28] Huang B, Wang W, Bates M and Zhuang X 2008 Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy *Science* **319** 810–3
[29] Kaczmarz S 1937 Angenäherte Auflösung von systemen linearer Gleichungen *Bull. Int. l'Acad. Pol. Sci. Lett. Cl. Sci. Math. Nat.* A **35** 335–57
[30] Kaipio J and Somersalo E 2006 *Statistical and Computational Inverse Problems* vol 160 (New York: Springer) (https://doi.org/10.1007/b138659)
[31] Kilmer M E and O'Leary D P 2001 Choosing regularization parameters in iterative methods for ill-posed problems *SIAM J. Matrix Anal. Appl.* **22** 1204–21
[32] Kushner H J and Yin G G 1997 *Stochastic Approximation Algorithms and Applications* (New York: Springer) (https://doi.org/10.1007/978-1-4899-2696-8)
[33] Le E B, Myers A, Bui-Thanh T and Nguyen Q P 2017 A data-scalable randomized misfit approach for solving large-scale PDE-constrained inverse problems *Inverse Problems* **33** 065003
[34] Marchesini S, Krishnan H, Daurer B J, Shapiro D A, Perciano T, Sethian J A and Maia F 2016 SHARP: a distributed GPU-based ptychographic solver *J. Appl. Crystallogr.* **49** 1245–52
[35] Matoušek J 2008 On variants of the Johnson–Lindenstrauss lemma *Random Struct. Algorithms* **33** 142–56
[36] Mokhtari A and Ribeiro A 2015 Global convergence of online limited memory BFGS *J. Mach. Learn. Res.* **16** 3151–81
[37] Mueller J L and Siltanen S 2012 *Linear and Nonlinear Inverse Problems with Practical Applications* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9781611972344)
[38] Needell D and Tropp J A 2014 Paved with good intentions: analysis of a randomized block Kaczmarz method *Linear Algebr. Appl.* **441** 199–221
[39] Nocedal J and Wright S J 2006 *Numerical Optimization* 2nd edn (New York: Springer) (https://doi.org/10.1007/978-0-387-40065-5)
[40] Paige C C and Saunders M A 1982 Algorithm 583, LSQR: sparse linear equations and least-squares problems *ACM Trans. Math. Softw.* **8** 195–209
[41] Paige C C and Saunders M A 1982 LSQR: an algorithm for sparse linear equations and sparse least squares *ACM Trans. Math. Softw.* **8** 43–71
[42] Park S, Park M and Kang M 2003 Super-resolution image reconstruction: a technical overview *IEEE Signal Process. Mag.* **20** 21–36
[43] Parkinson D Y, Pelt D M, Perciano T, Ushizima D, Krishnan H, Barnard H S, MacDowell A A and Sethian J 2017 Machine learning for micro-tomography *Developments in X-Ray Tomography XI* vol 10391 (International Society for Optics and Photonics) p 103910J
[44] Pilanci M and Wainwright M J 2016 Iterative Hessian sketch: fast and accurate solution approximation for constrained least-squares *J. Mach. Learn. Res.* **17** 1–38
[45] Renaut R A, Vatankhah S and Ardestani V E 2017 Hybrid and iteratively reweighted regularization by unbiased predictive risk and weighted GCV for projected systems *SIAM J. Sci. Comput.* **39** B221–43
[46] Saibaba A K, Alexanderian Al and Ipsen I 2017 Randomized matrix-free trace and log-determinant estimators *Numer. Math.* **137** 353–95
[47] Shapiro A, Dentcheva D and Ruszczyński A 2009 *Lectures on Stochastic Programming: Modeling and Theory* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9780898718751)
[48] Strohmer T and Vershynin R 2009 A randomized Kaczmarz algorithm with exponential convergence *J. Fourier Anal. Appl.* **15** 262–78
[49] Tenorio L 2017 *An Introduction to Data Analysis and Uncertainty Quantification for Inverse Problems* (Philadelphia, PA: SIAM) (https://doi.org/10.1137/1.9781611974928)