SKETCHED NEWTON-RAPHSON[*]

RUI YUAN[†], ALESSANDRO LAZARIC[‡], AND ROBERT M. GOWER[§]

Abstract. We propose a new globally convergent stochastic second order method. Our starting point is the development of a new Sketched Newton-Raphson (`SNR`) method for solving large scale nonlinear equations of the form $F(x) = 0$ with $F : \mathbb{R}^p \to \mathbb{R}^m$. We then show how to design several stochastic second order optimization methods by re-writing the optimization problem of interest as a system of nonlinear equations and applying `SNR`. For instance, by applying `SNR` to find a stationary point of a generalized linear model (GLM), we derive completely new and scalable stochastic second order methods. We show that the resulting method is very competitive as compared to state-of-the-art variance reduced methods. Furthermore, using a variable splitting trick, we also show that the *Stochastic Newton method* (`SNM`) is a special case of `SNR`, and use this connection to establish the first global convergence theory of `SNM`.

We establish the global convergence of `SNR` by showing that it is a variant of the online stochastic gradient descent (`SGD`) method, and then leveraging proof techniques of `SGD`. As a special case, our theory also provides a new global convergence theory for the original Newton-Raphson method under strictly weaker assumptions as compared to the classic monotone convergence theory.

Key words. Nonlinear systems, stochastic methods, iterative methods, stochastic Newton method, randomized Kaczmarz, randomized Newton, randomized Gauss-Newton, randomized fixed point, randomized subspace-Newton.

MSC codes. 58C15, 90C06, 90C53, 62L20, 46N10, 46N40, 49M15, 68W20, 68W40, 65Y20

ACM subject classifications. G.1.6 Optimization

1. Introduction. One of the fundamental problems in numerical computing is to find roots of systems of nonlinear equations such as

$$(1.1) \qquad\qquad F(x) = 0,$$

where $F : \mathbb{R}^p \to \mathbb{R}^m$. We assume throughout that $F : \mathbb{R}^p \to \mathbb{R}^m$ is continuously differentiable and that there exists a solution to (1.1), that is

ASSUMPTION 1.1. $\exists x^* \in \mathbb{R}^p$ *such that* $F(x^*) = 0$.

This includes a wide range of applications from solving the phase retrieval problems [10], systems of polynomial equations related to cryptographic primitives [6], discretized integral and differential equations [47], the optimal power flow problem [57] and, our main interest here, solving nonlinear minimization problems in machine learning. Most convex optimization problems such as those arising from training a Generalized Linear Model (GLM), can be re-written as a system of nonlinear equations (1.1) either by manipulating the stationarity conditions or as the Karush-Kuhn-Tucker equations[1].

When dealing with non-convex optimization problems, such as training a Deep Neural Network (DNN), finding the global minimum is often infeasible (or not needed

[†]Meta AI; LTCI, Télécom Paris; Institut Polytechnique de Paris (ruiyuan@fb.com).

[‡]Meta AI (lazaric@fb.com).

[§]CCM, Flatiron Institute; Part of this work was done when author was affiliated with Télécom Paris and Meta AI (gowerrobert@gmail.com).

[1]Under suitable constraint qualifications [46].

[31]). Instead, the objective is to find a good stationary point x such that $\nabla f(x) = 0$, where f is the total loss we want to minimize.

In particular, the task of training an overparametrized DNN (as they often are) can be cast as solving a special nonlinear system. That is, when the DNN is sufficiently overparametrized, the DNN can interpolate the data. As a consequence, if $f_i(x)$ is the loss function over the ith data point, then there is a solution to the system of nonlinear equations $\|\nabla f_i(x)\|^2 = 0, \forall i$.

The building block of many iterative methods for solving nonlinear equations is the Newton-Raphson (`NR`) method given by

$$(1.2) \qquad x^{k+1} = x^k - \gamma \left(DF(x^k)^\top\right)^\dagger F(x^k)$$

at kth iteration, where $DF(x) \stackrel{\text{def}}{=} [\nabla F_1(x) \ \cdots \ \nabla F_m(x)] \in \mathbb{R}^{p \times m}$ is the transpose of the Jacobian matrix of F at x, $\left(DF(x^k)^\top\right)^\dagger$ is the Moore-Penrose pseudoinverse of $DF(x^k)^\top$ and $\gamma > 0$ is the stepsize.

The `NR` method is at the heart of many commercial solvers for nonlinear equations [47]. The success of `NR` can be partially explained by its invariance to affine coordinate transformations, which in turn means that the user does not need to tune any parameters (standard `NR` sets $\gamma = 1$). The downside of `NR` is that we need to solve a linear least squares problem given in (1.2) which costs $\mathcal{O}(\min\{pm^2, mp^2\})$ when using a direct solver. When both p and m are large, this cost per iteration is prohibitive. Here we develop a randomized `NR` method based on the sketch-and-project technique [24] which can be applied in large scale, as we show in our experiments.

1.1. The sketched Newton-Raphson method. Our method relies on using *sketching matrices* to reduce the dimension of the Newton system.

DEFINITION 1.2. *The sketching matrix* $\mathbf{S} \in \mathbb{R}^{m \times \tau}$ *is a random matrix sampled from a distribution* $\mathcal{D}$, *where* $\tau \in \mathbb{N}$ *is the sketch size. We use* $\mathbf{S}_k \in \mathbb{R}^{m \times \tau}$ *to denote a sketching matrix sampled from a distribution* $\mathcal{D}_{x^k}$ *that can depend on the iterate* x^k.

By sampling a sketching matrix $\mathbf{S}_k \sim \mathcal{D}_{x^k}$ at kth iteration, we *sketch* (row compress) `NR` update and compute an approximate *Sketched Newton-Raphson* (`SNR`) step, see (1.3) in Alg. 1. We use $\mathcal{D}_x$ to denote a distribution that depends on x, and allow the distribution of the sketching matrix to change from one iteration to the next.

Algorithm 1 `SNR`: Sketched Newton-Raphson

1: **parameters:** $\mathcal{D}$ = distribution of sketching matrix; stepsize parameter $\gamma > 0$
2: **initialization:** Choose $x^0 \in \mathbb{R}^p$
3: **for** $k = 0, 1, \cdots$ **do**
4:      Sample a fresh sketching matrix: $\mathbf{S}_k \sim \mathcal{D}_{x^k}$

$$(1.3) \qquad x^{k+1} = x^k - \gamma DF(x^k)\mathbf{S}_k \left(\mathbf{S}_k^\top DF(x^k)^\top DF(x^k)\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top F(x^k)$$

5: **return:** last iterate x^k

Because the sketching matrix $\mathbf{S}_k$ has τ columns, the dominating costs of computing the `SNR` step (1.3) are linear in p and m. In particular, $DF(x^k)\mathbf{S}_k \in \mathbb{R}^{p \times \tau}$ can be computed by using τ directional derivatives of $F(x^k)$, one for each column of $\mathbf{S}_k$. Using automatic differentiation [13], these directional derivatives cost τ evaluations of the function $F(x)$. Furthermore, it costs $\mathcal{O}(p\tau^2)$ to form the linear system in (1.3) of

Alg. 1 by using the computed matrix $DF(x^k)\mathbf{S}_k$ and $\mathcal{O}(\tau^3)$ to solve it , respectively. Finally the matrix vector product $\mathbf{S}_k^\top F(x^k)$ costs $\mathcal{O}(m\tau)$. Thus, without making any further assumptions to the structure of F or the sketching matrix, the total cost in terms of operations of the update (1.3) is given by

$$(1.4) \qquad \text{Cost(update } (1.3)) = \mathcal{O}\left((\text{eval}(F) + m) \times \tau + p\tau^2 + \tau^3\right).$$

Thus Alg. 1 can be applied when both p and m are large and τ is relatively small.

The rest of the paper is organized as follows. In the next section, we provide some background and contrast it with our contributions. After introducing some notations in Section 1.3 and presenting alternative sketching techniques in Section 1.4, we show that (1.3) can be viewed as a sketch-and-project type method in Section 2. This is the viewpoint that first motivated the development of this method. After which, we provide another crucial equivalent viewpoint of (1.3) in Section 3, where we show that SNR can be seen as *Stochastic Gradient Descent* (SGD) applied to an equivalent reformulation of (1.1). We then provide a global convergence theory by leveraging this insight in Section 4. As a special case, our theory also provides a new global convergence theory for the original NR method (1.2) under strictly weaker assumptions as compared to the monotone convergence theory in Section 5, albeit for different step sizes. For the other extreme where the sketching matrix samples a single row, we present the new nonlinear Kaczmarz method as a variant of SNR and its global convergence theory in Section 6. We then show how to design several stochastic second order optimization methods by re-writing the optimization problem of interest as a system of nonlinear equations and applying SNR. For instance, using a variable splitting trick, we show that the *Stochastic Newton method* (SNM) [51, 35] is a special case of SNR, and use this connection to establish the first global convergence theory of SNM in Section 7. In Section 8, by applying SNR to find a stationary point of a GLM, we derive completely new and scalable stochastic second order methods. We show that the resulting method is very competitive as compared to state-of-the-art variance reduced methods.

1.2. Background and contributions.

a) Stochastic second-order methods. There is now a concerted effort to develop efficient second-order methods for solving high dimensional and stochastic optimization problems in machine learning. Most recently developed Newton methods fall into one of two categories: *subsampling* and *dimension reduction*. The subsampling methods [19, 52, 34, 8, 64] and [1, 48][2] use mini-batches to compute an approximate Newton direction. Though these methods can handle a large number of *data points* (n), they do not scale well in the number of *features* (d). On the other hand, second-order methods based on dimension reduction techniques such as [21] apply Newton's method over a subspace of the features, and as such, do not scale well in the number of data points. Sketching has also been used to develop second-order methods in the online learning setting [26, 39, 9] and quasi-Newton methods [22].

Contributions. We propose a new family of stochastic second-order method called SNR. Each choice of the sketching distribution and nonlinear equations used to describe the stationarity conditions, leads to a particular algorithm. For instance, we show that a nonlinear variant of the Kaczmarz method is a special case of SNR. We also show that

---

[2]Newton sketch [48] and LiSSa [1] use subsampling to build an estimate of the Hessian but require a full gradient evaluation. As such, these methods are not efficient for very large n.

the subsampling based SNM [51, 35] is a special case of SNR. By using a different norm in the sketch-and-project viewpoint, we show that the dimension reduced method *Randomized Subspace Newton* (RSN) [21] is also a special case of SNR. We provide a concise global convergence theory, that when specialized to SNM gives its first global convergence result. Furthermore, the convergence theory of SNR allows for any sketch size, which translates to any mini-batch size for the nonlinear Kaczmarz and SNM. In contrast, excluding SNM, the subsampled based Newton methods [19, 52, 34, 8, 64, 1, 48] rely on high probability bounds that in turn require large mini-batch sizes [3]. We detail the nonlinear Kaczmarz method in Sec. 6, the connection with SNM in Sec. 7 and RSN in App. E.

b) New method for GLMs. There exist several specialized methods for solving GLMs, including variance reduced gradient methods such as SAG/SAGA [53, 15] and SVRG [28], and methods based on dual coordinate ascent like SDCA [55], dual free SDCA (dfSDCA) [54] and Quartz [49].

Contributions. We develop a specialized variant of SNR for GLMs in Sec. 8. Our resulting method scales linearly in the number of dimensions d and the number of data points n, has the same cost as SGD per iteration in average. We show in experiments that our method is very competitive as compared to state-of-the-art variance reduced methods for GLMs.

c) Viewpoints of (Sketched) Newton-Raphson. We show in Sec. 3 that SNR can be seen as SGD applied to an equivalent reformulation of our original problem. We will show that this reformulation is *always* a smooth and interpolated function [40, 58]. These gratuitous properties allow us to establish a simple global convergence theory by only assuming that the reformulation is a *star-convex* function: a class of nonconvex functions that include convexity as a special case [45, 36, 67, 27]. The details of the SGD interpretation can be found in Sec. 3. In addition, we also show in App. A that SNR can be seen as a type of stochastic Gauss-Newton method or as a type of stochastic fixed point method.

d) Classic convergence theory of Newton-Raphson. The better known convergence theorems for NR (the Newton-Kantorovich-Mysovskikh Theorems) only guarantee local or semi-local convergence [30, 47]. To guarantee global convergence of NR, we often need an additional globalization strategy, such as damping sequences or adaptive trust-region methods [14, 38, 17, 32], continuation schemes such as interior point methods [44, 62], and more recently cubic regularization [35, 45, 11]. Globalization strategies are used in conjunction with other second-order methods, such as inexact Newton backtracking type methods [5, 3], Gauss-Newton or Levenberg-Marquardt type methods [66, 65, 63] and quasi-Newton methods [63][4]. The only global convergence theory that does not rely on such a globalization strategy, requires strong assumptions on $F(x)$, such as in the monotone convergence theory (MCT) [17].

Contributions. We show in Sec. 5.3 that our main theorem specialized to the standard NR method guarantees a global convergence under *strictly* less assumptions as compared to the MCT, albeit under a different stepsize. Indeed, MCT holds for step size equal to one ($\gamma = 1$) and our theory holds for step sizes less than one ($\gamma < 1$).

Furthermore, we give an explicit sublinear $O(1/k)$ convergence rate, as opposed to only an asymptotic convergence in MCT. This appears to not be known before since,

---

[3]The batch sizes in these methods scale proportional to a condition number [1] or ϵ^{-1} where ϵ is the desired tolerance.

[4]A recent paper [20] shows that quasi-Newton converges globally for self-concordant functions without globalization strategy.

as stated by [17] w.r.t. the `NR` method "*Not even an a-priori estimation for the number of iterations needed to achieve a prescribed accuracy* **may** *be possible*". We show that it is possible by monitoring which iterate achieves the best loss (suboptimality).

e) Sketch-and-project. The sketch-and-project method was originally introduced for solving linear systems in [24, 25], where it was also proven to converge linearly and globally. In [50], the authors then go on to show that the sketch-and-project method is in fact `SGD` applied to a particular reformulation of the linear system.

Contributions. It is this `SGD` viewpoint in the linear setting [50] that we extend to the nonlinear setting. Thus the `SNR` algorithm and our theory are generalizations of the original sketch-and-project method for solving linear equations to solving nonlinear equations, thus greatly expanding the scope of applications of these techniques.

1.3. Notations. In calculating an update of `SNR` (1.3) and analyzing `SNR`, the following random matrix is key

$$(1.5) \qquad \mathbf{H}_{\mathbf{S}}(x) \overset{\text{def}}{=} \mathbf{S} \left(\mathbf{S}^\top DF(x)^\top DF(x) \mathbf{S} \right)^\dagger \mathbf{S}^\top .$$

The sketching matrix $\mathbf{S}$ in (1.5) is sampled from a distribution $\mathcal{D}_x$ and $\mathbf{H}_{\mathbf{S}}(x) \in \mathbb{R}^{m \times m}$ is a random matrix that depends on x. We use $\mathbf{I}_p \in \mathbb{R}^{p \times p}$ to denote the identity matrix of dimension p and use $\|x\|_{\mathbf{M}} \overset{\text{def}}{=} \sqrt{x^\top \mathbf{M} x}$ to denote the seminorm of $x \in \mathbb{R}^p$ induced by a symmetric positive semi-definite matrix $\mathbf{M} \in \mathbb{R}^{p \times p}$. Notice that $\|x\|_{\mathbf{M}}$ is not necessarily a norm as $\mathbf{M}$ is allowed to be non invertible. We handle this with care in our forthcoming analysis. We also define the following sets: $F(U) = \{F(x) \mid x \in U\}$ for a given set $U \subset \mathbb{R}^p$; $W^\perp = \{v \mid \langle u, v \rangle = 0, \ \forall u \in W\}$ to denote the orthogonal complement of a subspace W; $\mathbf{Im}(\mathbf{M}) = \{y \in \mathbb{R}^m \mid \exists x \in \mathbb{R}^p \text{ s.t. } \mathbf{M}x = y\}$ to denote the image space and $\mathbf{Ker}(\mathbf{M}) = \{x \in \mathbb{R}^p \mid \mathbf{M}x = 0\}$ to denote the null space of a matrix $\mathbf{M} \in \mathbb{R}^{m \times p}$. If $\mathbf{M}$ is a random matrix sampled from a certain distribution $\mathcal{D}$, we use $\mathbb{E}_{\mathbf{S} \sim \mathcal{D}}[\mathbf{M}] = \int_{\mathbf{M}} \mathbf{M} d\mathbb{P}_{\mathcal{D}}(\mathbf{M})$ to denote the expectation of the random matrix. We omit the notation of the distribution $\mathcal{D}$, i.e. $\mathbb{E}[\mathbf{M}]$, when the random source is clear. In particular, when $\mathbf{M}$ is sampled from a discrete distribution with $r \in \mathbb{N}$ s.t. $\mathbb{P}[\mathbf{M} = \mathbf{M}_i] = p_i > 0$, for $i = 1, \cdots, r$ and $\sum_{i=1}^r p_i = 1$, then $\mathbb{E}[\mathbf{M}] = \sum_{i=1}^r p_i \mathbf{M}_i$.

1.4. Sketching matrices. Here we provide examples of sketching matrices that can be used in conjunction with `SNR`. We point the reader to [61] for a detailed exposure and introduction. The most straightforward sketch is given by the Gaussian sketch where every coordinate $\mathbf{S}_{ij}$ of the sketch $\mathbf{S} \in \mathbb{R}^{m \times \tau}$ is sampled i.i.d according to a Gaussian distribution with $\mathbf{S}_{ij} \sim \mathcal{N}(0, \frac{1}{\tau})$ for $i = 1, \cdots, m$ and $j = 1, \cdots, \tau$. The sketch we mostly use here is the uniform subsampling sketch, whereby

$$(1.6) \qquad \mathbb{P}[\mathbf{S} = \mathbf{I}_C] = \frac{1}{\binom{m}{\tau}} , \quad \text{for all set } C \subset \{1, \cdots, m\} \text{ s.t. } |C| = \tau ,$$

where $\mathbf{I}_C \in \mathbb{R}^{m \times \tau}$ denotes the concatenation of the columns of the identity matrix $\mathbf{I}_m$ indexed in the set C. More sophisticated sketches that are able to make use of fast Fourier type routines include the random orthogonal sketches (ROS) [48, 2]. We will not cover ROS sketches here since these sketches are fast when applied only once to a fixed matrix $\mathbf{M}$, as opposed to being re-sampled at every iteration .

2. The sketch-and-project viewpoint. The viewpoint that motivated the development of Alg. 1 was the following iterative *sketch-and-project* method applied to the Newton system. For this viewpoint, we assume that

ASSUMPTION 2.1. $F(x) \in \mathbf{Im}\left(DF(x)^\top\right) \forall x \in \mathbb{R}^p$.

This assumption guarantees that there exists a solution to the Newton system in (1.2). Indeed, we can now re-write the NR method (1.2) as a projection of the previous iterate x^k onto the solution space of a Newton system

$$(2.1) \qquad x^{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^p} \|x - x^k\|^2 \quad \text{s. t.} \quad DF(x^k)^\top (x - x^k) = -\gamma F(x^k).$$

Since this is costly to solve when $DF(x^k)$ has many rows and columns, we *sketch* the Newton system. That is, we apply a random row compression to the Newton system using the sketching matrix $\mathbf{S}_k^\top \in \mathbb{R}^{\tau \times m}$ and then project the previous iterates x^k onto this *sketched* system as follows

$$(2.2) \quad x^{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^p} \|x - x^k\|^2 \quad \text{s. t.} \quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\gamma \mathbf{S}_k^\top F(x^k).$$

That is, x^{k+1} is the projection of x^k onto the solution space of the sketched Newton system. This viewpoint was our motivation for developing the SNR method. Next we establish our core theory. The theory does not rely on the assumption $F(x) \in \mathbf{Im}\left(DF(x)^\top\right)$, though this assumption will appear again in several specialized corollaries. Without this assumption, we can still interpret the Newton step (1.2) as the least squares solution of the linear system (2.1), as we show next.

3. Reformulation as stochastic gradient descent. Our insight into interpreting and analyzing the SNR in Alg. 1 is through its connection to the SGD. Next, we show how SNR can be seen as SGD applied to a sequence of equivalent reformulations of (1.1). Each reformulation is given by a vector $y \in \mathbb{R}^p$ and the following minimization problem

$$(3.1) \qquad\qquad \min_{x \in \mathbb{R}^p} \mathbb{E}_{\mathbf{S} \sim \mathcal{D}_y} \left[\tfrac{1}{2} \|F(x)\|_{\mathbf{H}_\mathbf{S}(y)}^2 \right],$$

where $\mathbf{H}_\mathbf{S}(y)$ is defined in (1.5). To abbreviate notations, let

$$(3.2) \qquad f_{\mathbf{S},y}(x) \stackrel{\text{def}}{=} \frac{1}{2} \|F(x)\|_{\mathbf{H}_\mathbf{S}(y)}^2 \quad \text{and} \quad f_y(x) \stackrel{\text{def}}{=} \mathbb{E}\left[f_{\mathbf{S},y}(x)\right] = \frac{1}{2} \|F(x)\|_{\mathbb{E}[\mathbf{H}_\mathbf{S}(y)]}^2.$$

Every solution $x^* \in \mathbb{R}^p$ to (1.1) is a solution to (3.1), since $f_y(x)$ is non-negative for every $x \in \mathbb{R}^p$ and $f_y(x^*) = 0$ is thus a global minima. With an extra assumption, we can show that every solution to (3.1) is also a solution to (1.1) in the following lemma.

LEMMA 3.1. *If Asm 1.1 holds and the following* reformulation *assumption*

$$(3.3) \qquad\qquad F(\mathbb{R}^p) \cap \mathbf{Ker}\left(\mathbb{E}_{\mathbf{S} \sim \mathcal{D}_y}\left[\mathbf{H}_\mathbf{S}(y)\right]\right) = \{0\}, \quad \forall y \in \mathbb{R}^p$$

holds, then $\operatorname{argmin}_{x \in \mathbb{R}^p} f_y(x) = \{x \mid F(x) = 0\}$ *for every* $y \in \mathbb{R}^p$.

Proof. Let $y \in \mathbb{R}^p$. Previously, we show that $\{x \mid F(x) = 0\} \subset \operatorname{argmin}_{x \in \mathbb{R}^p} f_y(x)$. Now let $x^* \in \operatorname{argmin}_{x \in \mathbb{R}^p} f_y(x)$. By Asm. 1.1, we know that any global minimizer x^* of $f_y(x)$ must be s.t. $f_y(x^*) = 0$. This implies that $F(x^*) \in \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(y)\right]\right)$ since $\mathbb{E}\left[\mathbf{H}_\mathbf{S}(y)\right]$ is symmetric. However $F(x^*) \in F(\mathbb{R}^p)$ and thus from (3.3), we have that $F(x^*) \in F(\mathbb{R}^p) \cap \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(y)\right]\right) = \{0\}$, which implies $F(x^*) = 0$. Thus, we have $\operatorname{argmin}_{x \in \mathbb{R}^p} f_y(x) \subset \{x \mid F(x) = 0\}$ which concludes the proof. $\qquad\square$

Thus with the extra reformulation assumption in (3.3), we can now use any viable optimization method to solve (3.1) for any fixed $y \in \mathbb{R}^p$ and arrive at a solution to (1.1). In Lemma B.1, we give sufficient conditions on the sketching matrix and on

the function $F(x)$ that guarantee (3.3) hold. We also show how (3.3) holds for our forthcoming examples in Sec. B as a direct consequence of Lemma B.1. However, (3.3) imposes for all $y \in \mathbb{R}^p$ which can be sometimes restrictive. In fact, we do *not need* for (3.1) to be equivalent to solving (1.1) for *every* $y \in \mathbb{R}^p$. Indeed, by carefully and iteratively updating y, we can solve (3.1) and obtain a solution to (1.1) *without* relying on (3.3). The trick here is to use an *online `SGD`* method for solving (3.1).

Since (3.1) is a stochastic optimization problem, `SGD` is a natural choice for solving (3.1). Let $\nabla f_{\mathbf{S},y}(x)$ denote the gradient of the function $f_{\mathbf{S},y}(\cdot)$ which is

$$(3.4) \qquad \nabla f_{\mathbf{S},y}(x) = DF(x)\mathbf{H}_{\mathbf{S}}(y)F(x).$$

Since we are free to choose y, we allow y to *change* from one iteration to the next by setting $y = x^k$ at the start of the kth iteration. We can now take a `SGD` step by sampling $\mathbf{S}_k \sim \mathcal{D}_{x^k}$ at kth iteration and updating

$$(3.5) \qquad x^{k+1} = x^k - \gamma \nabla f_{\mathbf{S}_k,x^k}(x^k).$$

It is straightforward to verify that the `SGD` update (3.5) is exactly the same as the `SNR` update in (1.3).

The objective function $f_{\mathbf{S},y}(x)$ has many properties that makes it very favourable for optimization including the interpolation condition and a gratuitous smoothness property. Indeed, for any $x^* \in \mathbb{R}^p$ s.t. $F(x^*) = 0$, we have that the stochastic gradient is zero, i.e. $\nabla f_{\mathbf{S},y}(x^*) = 0$. This is known as the *interpolation condition*. When it occurs together with strong convexity, it is possible to shows that `SGD` converges linearly [58, 40]. We will also give a linear convergence result in Sec. 4 by assuming that $f_y(x)$ is quasi-strongly convex. We detail the smoothness property next.

However, we need to be careful, since (3.5) is not a classic `SGD` method. In fact, from the kth iteration to the $(k+1)$th iteration, we change our objective function from $f_{x^k}(x)$ to $f_{x^{k+1}}(x)$ and the distribution from $\mathcal{D}_{x^k}$ to $\mathcal{D}_{x^{k+1}}$. Thus it is an online `SGD`. We handle this with care in our forthcoming convergence proofs.

4. Convergence theory. Using the viewpoint of `SNR` in Sec. 3, we adapt proof techniques of `SGD` to establish the global convergence of `SNR`.

4.1. Smoothness property. In our upcoming proof, we rely on the following type of smoothness property thanks to our `SGD` reformulation (3.1).

LEMMA 4.1. *For every $x \in \mathbb{R}^p$ and any realization $\mathbf{S} \sim \mathcal{D}_x$ associated with any distribution $\mathcal{D}_x$,*

$$(4.1) \qquad \frac{1}{2}\|\nabla f_{\mathbf{S},x}(x)\|^2 = f_{\mathbf{S},x}(x).$$

Proof. Turning to the definition of $f_{\mathbf{S},x}$ in (3.2), we have that

$$\|\nabla f_{\mathbf{S},x}(x)\|^2 \overset{(3.4)}{=} \|DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x)\|^2 = F(x)^\top \mathbf{H}_{\mathbf{S}}(x)^\top DF(x)^\top DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x)$$
$$= F(x)^\top \mathbf{H}_{\mathbf{S}}(x)F(x) = 2f_{\mathbf{S},x}(x),$$

where we used the property $\mathbf{M}^\dagger \mathbf{M} \mathbf{M}^\dagger = \mathbf{M}^\dagger$ with $\mathbf{M} = \mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}$ to establish that $\mathbf{H}_{\mathbf{S}}(x)^\top DF(x)^\top DF(x)\mathbf{H}_{\mathbf{S}}(x) \overset{(1.5)}{=} \mathbf{H}_{\mathbf{S}}(x)$. $\qquad \square$

This is not a standard smoothness property. Indeed, since $\nabla f_{\mathbf{S},x}(x^*) = 0$ and $f_x(x^*) = 0$, we have that (4.1) implies that

$$\|\nabla f_{\mathbf{S},x}(x) - \nabla f_{\mathbf{S},x}(x^*)\|^2 \le 2(f_{\mathbf{S},x}(x) - f_{\mathbf{S},x}(x^*)),$$

which is usually a consequence of assuming that $f_{\mathbf{S},x}(x)$ is convex and 1–smooth (see Theorem 2.1.5 and Equation 2.1.7 in [43]). Yet in our case, eq. (4.1) is a direct consequence of the definition of $f_{\mathbf{S},x}$ as opposed to being an extra assumption. This gratuitous property will be key in establishing a global convergence result.

4.2. Convergence for star-convex. We use the shorthand $f_k(x) \stackrel{\text{def}}{=} f_{x^k}(x)$, $f_{\mathbf{S},k} \stackrel{\text{def}}{=} f_{\mathbf{S}_k,x^k}$ and $\mathbb{E}_k[\cdot] \stackrel{\text{def}}{=} \mathbb{E}[\cdot \mid x^k]$. Here we establish the global convergence of SNR by supposing that f_k is *star-convex* which is a large class of nonconvex functions that includes convexity as a special case [45, 36, 67, 27].

ASSUMPTION 4.2 (Star-Convexity). *Let x^* satisfy Asm. 1.1, i.e. let x^* be a solution to* (1.1). *For every x^k given by Alg. 1 with $k \in \mathbb{N}$, we have that*

$$(4.2) \qquad f_k(x^*) \geq f_k(x^k) + \langle \nabla f_k(x^k), x^* - x^k \rangle .$$

We now state our main theorem.

THEOREM 4.3. *Let x^* satisfy Asm. 4.2. If $0 < \gamma < 1$, then*

$$(4.3) \qquad \mathbb{E}\left[\min_{t=0,\ldots,k-1} f_t(x^t)\right] \leq \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[f_t(x^t)\right] \leq \frac{1}{k}\frac{\|x^0 - x^*\|^2}{2\gamma\,(1-\gamma)}.$$

Written in terms of F and for $\gamma = 1/2$ the above gives

$$\mathbb{E}\left[\min_{t=0,\ldots,k-1}\|F(x^t)\|^2_{\mathbb{E}[\mathbf{H}_\mathbf{S}(x^t)]}\right] \quad \leq \quad \frac{4\|x^0 - x^*\|^2}{k}.$$

Besides, if the stochastic function $f_{\mathbf{S},x}(x)$ is star-convex along the iterates x^k, i.e.

$$(4.4) \qquad f_{\mathbf{S}_k,x^k}(x^*) \geq f_{\mathbf{S}_k,x^k}(x^k) + \langle \nabla f_{\mathbf{S}_k,x^k}(x^k), x^* - x^k \rangle$$

for all $\mathbf{S}_k \sim \mathcal{D}_{x^k}$, then the iterates x^k of SNR (1.3) *are bounded with*

$$(4.5) \qquad \|x^k - x^*\| \leq \|x^0 - x^*\|.$$

Proof. Let $t \in \{0,\ldots,k-1\}$ and $\delta_t \stackrel{\text{def}}{=} x^t - x^*$. We have that

$$\begin{aligned}
\mathbb{E}_t\left[\|\delta_{t+1}\|^2\right] &\stackrel{(3.5)}{=}& \mathbb{E}_t\left[\|x^t - \gamma\nabla f_{\mathbf{S}_t,t}(x^t) - x^*\|^2\right] \\
&=& \|\delta_t\|^2 - 2\gamma\left\langle\delta_t, \nabla f_t(x^t)\right\rangle + \gamma^2\mathbb{E}_t\left[\|\nabla f_{\mathbf{S}_t,t}(x^t)\|^2\right] \\
&\stackrel{(4.2)}{\leq}& \|\delta_t\|^2 - 2\gamma(f_t(x^t) - f_t(x^*)) + \gamma^2\mathbb{E}_t\left[\|\nabla f_{\mathbf{S}_t,t}(x^t)\|^2\right] \\
&\stackrel{(4.1)}{=}& \|\delta_t\|^2 - 2\gamma\,(1-\gamma)\,(f_t(x^t) - f_t(x^*)) \\
(4.6) \qquad &\stackrel{f_t(x^*)=0}{=}& \|\delta_t\|^2 - 2\gamma\,(1-\gamma)\,f_t(x^t).
\end{aligned}$$

Taking total expectation for all $t \in \{0,\ldots,k-1\}$, we have that

$$(4.7) \qquad \mathbb{E}\left[\|\delta_{t+1}\|^2\right] \leq \mathbb{E}\left[\|\delta_t\|^2\right] - 2\gamma\,(1-\gamma)\,\mathbb{E}\left[f_t(x^t)\right].$$

Summing both sides of (4.7) from 0 to $k-1$ gives

$$\mathbb{E}\left[\|x^k - x^*\|^2\right] + 2\gamma\,(1-\gamma)\sum_{t=0}^{k-1}\mathbb{E}\left[f_t(x^t)\right] \leq \|x^0 - x^*\|^2.$$

Dividing through by $2\gamma(1-\gamma) > 0$ and by k, we have that

$$\mathbb{E}\left[\min_{t=0,\ldots,k-1} f_t(x^t)\right] \leq \min_{t=0,\ldots,k-1} \mathbb{E}\left[f_t(x^t)\right] \leq \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[f_t(x^t)\right] \leq \frac{1}{k}\frac{\|x^0 - x^*\|^2}{2\gamma(1-\gamma)},$$

where in the most left inequality we used Jensen's inequality.

Finally, if (4.4) holds, then we can repeat the steps leading up to (4.6) without the conditional expectation, so that

$$\|\delta_{t+1}\|^2 \overset{(3.5)+(4.4)+(4.1)}{\leq} \|\delta_t\|^2 - 2\gamma(1-\gamma)f_{\mathbf{S}_t,t}(x^t).$$

Since $f_{S_t,t}(x^t) \geq 0$, we have $\|\delta_{t+1}\|^2 \leq \|\delta_t\|^2$, i.e. (4.5) holds. □

Thm. 4.3 is an unusual result for `SGD` methods. Currently, to get a $\mathcal{O}(1/k)$ convergence rate for `SGD`, one has to assume smoothness and strong convexity [23] or convexity, smoothness and interpolation [58]. Here we get a $\mathcal{O}(1/k)$ rate by *only* assuming star-convexity. This is because we have smoothness and interpolation properties as a by-product due to our reformulation (3.1). However, the star-convexity assumption of $f_k(\cdot)$ for all $k \in \mathbb{N}$ is hard to interpret in terms of assumptions on F in general. But, we are able to interpret it in many important extremes. That is, for the full `NR` method, we show that it suffices for the Newton direction to be 2–co-coercive (see (5.8) in Sec. 5). For the other extreme where the sketching matrix samples a single row, then the star-convexity assumption is even easier to check, and is guaranteed to hold so long as $F_i(x)^2$ is convex for all $i = 1, \cdots, m$ (see Sec. 6).

Next, we will show the convergence of $F(x^k)$ instead of $f_k(x^k)$ via Thm. 4.3.

4.2.1. Sublinear convergence of the Euclidean norm $\|F\|$. If $\mathbb{E}[\mathbf{H_S}(x)]$ is invertible for all $x \in \mathbb{R}^p$, we can use Thm. 4.3 with the bound (4.5) to guarantee that $\|F\|$ converges sublinearly. Indeed, when $\mathbb{E}[\mathbf{H_S}(x)]$ is invertible, $\mathbb{E}[\mathbf{H_S}(x)]$ is symmetric positive definite. Thus there exists $\lambda > 0$ that bounds the smallest eigenvalue away from zero in any closed bounded set (e.g. $\{x \in \mathbb{R}^p \mid \|x - x^*\| \leq \|x^0 - x^*\|\}$ [5]):

$$(4.8) \qquad \min_{x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}} \lambda_{\min}\left(\mathbb{E}[\mathbf{H_S}(x)]\right) = \lambda > 0,$$

where $\lambda_{\min}(\cdot)$ is the smallest eigenvalue operator. Consequently, under the assumption of Thm. 4.3 with the condition (4.4), from (4.5) and (4.8), we have

$$(4.9) \quad \lambda\mathbb{E}\left[\min_{t=0,\ldots,k-1}\|F(x^t)\|^2\right] \leq \mathbb{E}\left[\min_{t=0,\ldots,k-1}\|F(x^t)\|^2_{\mathbb{E}[\mathbf{H_S}(x^t)]}\right] \overset{(4.3)}{\leq} \frac{1}{k}\frac{\|x^0 - x^*\|^2}{\gamma(1-\gamma)}.$$

It turns out that using the smallest eigenvalue of $\mathbb{E}[\mathbf{H_S}(x)]$ in the above bound is overly pessimistic. To improve it, first note that we do *not need* that $\mathbb{E}[\mathbf{H_S}(x)]$ is invertible. Instead, we only need that $F(x) \in \mathbf{Im}(DF(x)^\top) \subset \mathbf{Im}(\mathbb{E}[\mathbf{H_S}(x)])$, as we show in Cor. 4.5. But first, we need the following lemma.

LEMMA 4.4 (Lemma 10 in [21]). *For any matrix $\mathbf{W}$ and symmetric positive semi-definite matrix $\mathbf{G}$ s.t. $\mathbf{Ker}(\mathbf{G}) \subset \mathbf{Ker}(\mathbf{W})$, we have $\mathbf{Ker}(\mathbf{W}^\top) = \mathbf{Ker}(\mathbf{WGW}^\top)$.*

Note $L \overset{\text{def}}{=} \sup_{x \in \{x \mid \|x - x^*\| \leq \|x^0 - x^*\|\}} \|DF(x)\| > 0$. Such L exists because x is in a closed bounded convex set and because we have assumed that $DF(\cdot)$ is continuous. A continuous mapping over a closed bounded convex set is bounded. Now we can state the sublinear convergence results for $\|F\|$.

---

[5] We can re-write the set as the closure of the ball $\{x \in \mathbb{R}^p \mid x \in \overline{\mathcal{B}(x^*, \|x^0 - x^*\|)}\}$.

Corollary 4.5. *Let*

$$(4.10) \qquad \rho(x) \overset{def}{=} \min_{v \in \mathbf{Im}(DF(x))/\{0\}} \frac{v^\top DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right]DF(x)^\top v}{\|v\|^2},$$

$$(4.11) \qquad \rho \overset{def}{=} \min_{x \in \{x|\|x-x^*\| \le \|x^0-x^*\|\}} \rho(x).$$

It follows that $0 \le \rho(x) \le 1$. If

$$(4.12) \qquad F(x) \in \mathbf{Im}(DF(x)^\top) \subset \mathbf{Im}(\mathbb{E}\left[\mathbf{H_S}(x)\right]) \quad \textit{for all } x \in \mathbb{R}^p,$$

then $\quad \rho(x) = \lambda_{\min}^+\left(DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right]DF(x)^\top\right) > 0 \quad \forall x \in \mathbb{R}^p, \quad \textit{and} \quad \rho > 0,$ *where $\lambda_{\min}^+$ is the smallest non-zero eigenvalue. Furthermore, if the star-convexity for each sketching matrix* (4.4) *holds , then*

$$(4.13) \qquad \mathbb{E}\left[\min_{t=0,\dots,k-1}\|F(x^t)\|^2\right] \le \frac{1}{k} \cdot \frac{L^2\|x^0-x^*\|^2}{\rho\gamma\left(1-\gamma\right)}.$$

Proof. First recall that

$$DF(x)\mathbf{H_S}(x)DF(x)^\top DF(x)\mathbf{H_S}(x)DF(x)^\top = DF(x)\mathbf{H_S}(x)DF(x)^\top \quad \textit{for all } x \in \mathbb{R}^p,$$

which is shown in the proof of Lemma 4.1. Thus $DF(x)\mathbf{H_S}(x)DF(x)^\top$ is a projection. By Jensen's inequality, the eigenvalues of an expected projection are between 0 and 1. Thus by the definition of $\rho(x)$, we have $0 \le \rho(x) \le 1$. Next, by (4.12), we have $\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H_S}(x)\right]\right) \subset \mathbf{Ker}\left(DF(x)\right)$. Thus, by Lemma 4.4 we have that

$$(4.14) \quad \mathbf{Im}\left(DF(x)\right) = \left(\mathbf{Ker}\left(DF(x)^\top\right)\right)^\perp = \left(\mathbf{Ker}\left(DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right]DF(x)^\top\right)\right)^\perp,$$

where the second equality is obtained by Lemma 4.4. Now from the definition of $\rho(x)$ in (4.10), we have

$$\rho(x) \overset{(4.14)}{=} \min_{v \in (\mathbf{Ker}(DF(x)\mathbb{E}[\mathbf{H_S}(x)]DF(x)^\top))^\perp/\{0\}} \frac{v^\top DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right]DF(x)^\top v}{\|v\|^2}$$
$$= \lambda_{\min}^+\left(DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right]DF(x)^\top\right) > 0.$$

It now follows that $\rho > 0$, since the definition of ρ in (4.11) is given by minimizing $\rho(x)$ over the closed bounded set $\{x \mid \|x-x^*\| \le \|x^0-x^*\|\}$. Next, given $x \in \{x \mid \|x-x^*\| \le \|x^0-x^*\|\}$, since $F(x) \in \mathbf{Im}(DF(x)^\top)$ by (4.12) and notice that $\mathbf{Im}(DF(x)^\top) = \mathbf{Im}(DF(x)^\top DF(x))$, there exists $v \in \mathbb{R}^m$ s.t. $F(x) = DF(x)^\top DF(x)v$.

If $F(x) \ne 0$, then $DF(x)v \in \mathbf{Im}\left(DF(x)\right)/\{0\}$, we have

$$\|F(x)\|^2_{\mathbb{E}[\mathbf{H_S}(x)]} = v^\top DF(x)^\top DF(x)\mathbb{E}\left[\mathbf{H_S}(x)\right]DF(x)^\top DF(x)v$$
$$(4.15) \qquad\qquad \overset{(4.10)}{\ge} \rho(x)v^\top DF(x)^\top DF(x)v.$$

Since $F(x) = DF(x)^\top DF(x)v$ and $\mathbf{Im}(DF(x)^\top) \oplus \mathbf{Ker}(DF(x)) = \mathbb{R}^m,$[6] we have that

$$\exists! \; y \in \mathbf{Ker}(DF(x)) \subset \mathbb{R}^m \text{ s.t. } v = (DF(x)^\top DF(x))^\dagger F(x) + y.$$

---

[6]The operator $\oplus$ denotes the direct sum of two vector spaces.

Thus
$$DF(x)v = DF(x)(DF(x)^\top DF(x))^\dagger F(x) = (DF(x)^\top)^\dagger F(x).$$

Substituting this in (4.15), we have that

$$(4.16) \qquad \|F(x)\|^2_{\mathbb{E}[\mathbf{H_S}(x)]} \geq \rho(x)\|F(x)\|^2_{(DF(x)^\top DF(x))^\dagger} \geq \frac{\rho}{L^2}\|F(x)\|^2,$$

where on the last inequality, we use that $\sup_{x \in \{x | \|x-x^*\| \leq \|x^0-x^*\|\}} \|DF(x)\| \leq L$ and $\rho(x) \geq \rho$ by the definition of ρ in (4.11) .

If $F(x) = 0$, (4.16) still holds. Thus, for all $x \in \{x \mid \|x-x^*\| \leq \|x^0-x^*\|\}$, (4.16) holds. Consequently by Thm. 4.3 and (4.5) under the star-convexity condition (4.4) with $\|x^t - x^*\| \leq \|x^0 - x^*\|$ for all $t \in \{0, \cdots, k-1\}$, we have that

$$\frac{\rho}{L^2}\mathbb{E}\left[\min_{t=0,\ldots,k-1}\|F(x^t)\|^2\right] \overset{(4.16)}{\leq} \mathbb{E}\left[\min_{t=0,\ldots,k-1}\|F(x^t)\|^2_{\mathbb{E}[\mathbf{H_S}(x^t)]}\right] \overset{(4.3)}{\leq} \frac{1}{k}\frac{\|x^0-x^*\|^2}{\gamma(1-\gamma)},$$

which after multiplying through by $L^2/\rho > 0$ concludes the proof. $\qquad\square$

Thus with Cor. 4.5, we show that $F(x^t)$ converges to zero. This lemma relies on the inclusion (4.12), which in turn imposes some restrictions on the sketching matrix and $F(x)$. In our forthcoming examples in Sec. 5 and 6, we can directly verify the inclusion of (4.12). For other examples in Sec. 7 and 8, we provide the following Lemma 4.6 where we give sufficient conditions for (4.12) to hold.

LEMMA 4.6. *Let* $F(x) \in \mathbf{Im}(DF(x)^\top)$. *Furthermore, we suppose that* $\mathbf{S} \sim \mathcal{D}_x$ *is adapted to* $DF(x)$ *by which we mean*

$$(4.17) \qquad \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{SS}^\top\right]\right) \subset \mathbf{Ker}(DF(x)) \subset \mathbf{Ker}\left(\mathbf{S}^\top\right), \quad \text{for all } \mathbf{S} \sim \mathcal{D}_x.$$

Then it follows that (4.12) *holds for all* $x \in \mathbb{R}^p$.

Proof. Since $\mathbf{Ker}(DF(x)^\top DF(x)) = \mathbf{Ker}(DF(x)) \overset{(4.17)}{\subset} \mathbf{Ker}(\mathbf{S}^\top)$, we have

$$(4.18) \qquad \mathbf{Ker}\left(\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger\right) = \mathbf{Ker}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right) = \mathbf{Ker}(\mathbf{S}),$$

where the last equality is obtained by Lemma 4.4 with $\mathbf{Ker}(DF(x)^\top DF(x)) \subset \mathbf{Ker}(\mathbf{S}^\top)$. Thus, using Lemma 4.4 again with $\mathbf{G} = \left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger$, $\mathbf{W} = \mathbf{S}$ and $\mathbf{Ker}(\mathbf{G}) \subset \mathbf{Ker}(\mathbf{W})$ given by (4.18), we have that
(4.19)
$$\mathbf{Ker}(\mathbf{H_S}(x)) \overset{(1.5)}{=} \mathbf{Ker}(\mathbf{S}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger \mathbf{S}^\top) = \mathbf{Ker}(\mathbf{S}^\top) = \mathbf{Ker}(\mathbf{SS}^\top).$$

As $\mathbf{H_S}(x)$ is symmetric positive semi-definite $\forall\, \mathbf{S} \sim \mathcal{D}_x$, we have that

$$v \in \mathbf{Ker}\left(\mathbb{E}\left[\mathbf{H_S}(x)\right]\right) \iff \mathbb{E}\left[\mathbf{H_S}(x)\right]v = 0 \iff \|v\|^2_{\mathbb{E}[\mathbf{H_S}(x)]} = 0 \quad (\text{as } \mathbb{E}\left[\mathbf{H_S}(x)\right] \succeq 0)$$

$$\iff \mathbb{E}\left[\|v\|^2_{\mathbf{H_S}(x)}\right] = 0 \iff \int_{\mathbf{S}}\|v\|^2_{\mathbf{H_S}(x)}d\mathbb{P}_{\mathcal{D}_x}(\mathbf{S}) = 0$$

$$\iff \|v\|^2_{\mathbf{H_S}(x)} = 0 \,\forall\, \mathbf{S} \sim \mathcal{D}_x \quad \left(\text{as } \|v\|^2_{\mathbf{H_S}(x)} \geq 0 \,\forall\, \mathbf{S}\right)$$

$$\overset{\mathbf{H_S}(x) \succeq 0}{\iff} v \in \mathbf{Ker}\left(\mathbf{H_S}(x)\right) \,\forall\, \mathbf{S} \sim \mathcal{D}_x \iff v \in \bigcap_{\mathbf{S} \sim \mathcal{D}_x}\mathbf{Ker}\left(\mathbf{H_S}(x)\right),$$

where we use $\bigcap_{\mathbf{S} \sim \mathcal{D}_x} \mathbf{Ker}\left(\mathbf{H_S}(x)\right)$ to note the intersection of the random subsets $\mathbf{Ker}\left(\mathbf{H_S}(x)\right)$ for all $\mathbf{S} \sim \mathcal{D}_x$. Similarly, we have $\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{SS}^\top\right]\right) = \bigcap_{\mathbf{S} \sim \mathcal{D}_x} \mathbf{Ker}\left(\mathbf{SS}^\top\right)$ because $\mathbf{SS}^\top$ is also symmetric, positive semi-definite for all $\mathbf{S} \sim \mathcal{D}_x$. Thus we have

$$\begin{aligned}
\mathbf{Ker}(\mathbb{E}\left[\mathbf{H_S}(x)\right]) &= \bigcap_{\mathbf{S} \sim \mathcal{D}_x} \mathbf{Ker}(\mathbf{H_S}(x)) \\
&\overset{(4.19)}{=} \bigcap_{\mathbf{S} \sim \mathcal{D}_x} \mathbf{Ker}(\mathbf{SS}^\top) = \mathbf{Ker}(\mathbb{E}\left[\mathbf{SS}^\top\right]) \overset{(4.17)}{\subset} \mathbf{Ker}(DF(x)).
\end{aligned}$$

Consequently, by considering the complement of the above, we arrive at (4.12).   □

We refer to a sketching matrix $\mathbf{S} \sim \mathcal{D}_x$ that satisfies (4.17) as a sketch that is *adapted to* $DF(x)$. One easy way to design such adapted sketches is the following.

LEMMA 4.7. *Let* $\hat{\mathbf{S}} \in \mathbb{R}^{p \times \tau}$ *s.t.* $\hat{\mathbf{S}} \sim \mathcal{D}$ *a fixed distribution independent to* x *and* $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) \subset \mathbf{Ker}(DF(x)^\top)$. *Thus,* $\mathbf{S} = DF(x)^\top \hat{\mathbf{S}} \in \mathbb{R}^{m \times \tau}$ *is adapted to* $DF(x)$.

Proof. First, $\mathbf{Ker}(DF(x)) \subset \mathbf{Ker}(\hat{\mathbf{S}}^\top DF(x)) = \mathbf{Ker}(\mathbf{S}^\top)$. Furthermore, from Lemma 4.4 with $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) \subset \mathbf{Ker}(DF(x)^\top)$, we conclude the proof with

$$\mathbf{Ker}\left(\mathbb{E}\left[\mathbf{SS}^\top\right]\right) = \mathbf{Ker}(DF(x)^\top \mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]DF(x)) \subset \mathbf{Ker}(DF(x)).$$ □

The condition $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) \subset \mathbf{Ker}(DF(x)^\top)$ in Lemma 4.7 holds for many standard sketches including Gaussian and subsampling sketches presented as follows.

LEMMA 4.8. *For Gaussian and uniform subsampling sketches defined in Sec. 1.4, we have that* $\mathbb{E}\left[\mathbf{SS}^\top\right] = c\mathbf{I}_m$ *with* $c > 0$ *a fixed constant depending on the sketch.*

Proof. For Gaussian sketches with $\mathbf{S}_{ij} \sim \mathcal{N}(0, \frac{1}{\tau})$, we have that $c = 1$. Indeed, since the mean is zero, off-diagonal elements of $\mathbb{E}\left[\mathbf{SS}^\top\right]$ are all zero. We note $\mathbf{S}_{i:}$ the ith row of $\mathbf{S}$, then the ith diagonal element of the matrix $\mathbb{E}\left[\mathbf{SS}^\top\right]$ is given by

$$\mathbb{E}\left[\mathbf{S}_{i:}\mathbf{S}_{i:}^\top\right] = \sum_{j=1}^{\tau} \mathbb{E}\left[\mathbf{S}_{ij}^2\right] = \sum_{j=1}^{\tau} \frac{1}{\tau} = 1.$$

For the uniform subsampling sketch (1.6), we have again that off-diagonal elements are zero since the rows of $\mathbf{S}$ are orthogonal. The diagonal elements are constant with

$$\mathbb{E}\left[\mathbf{S}_{i:}\mathbf{S}_{i:}^\top\right] = \frac{1}{\binom{m}{\tau}} \sum_{C \subset \{1,\ldots,m\}, |C|=\tau, i \in C} 1 = \frac{\binom{m-1}{\tau-1}}{\binom{m}{\tau}} = \frac{\tau}{m}, \quad \text{for all } i = 1, \cdots, m.$$ □

From Lemma 4.8, we know that $\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top] = c\mathbf{I}_p$ invertible with $c > 0$. Thus $\mathbf{Ker}(\mathbb{E}[\hat{\mathbf{S}}\hat{\mathbf{S}}^\top]) = \{0\} \subset \mathbf{Ker}(DF(x)^\top)$ holds for *any* sketch size τ.

4.3. Convergence for strongly convex. Here we establish a global linear convergence of SNR when assuming that f_y is strongly quasi-convex.

ASSUMPTION 4.9 (μ-Strongly Quasi-Convexity). *Let* x^* *satisfy Asm. 1.1 and*

$$(4.20) \quad \exists\, \mu > 0 \text{ s.t. } f_y(x^*) \geq f_y(x) + \langle \nabla f_y(x), x^* - x \rangle + \frac{\mu}{2}\|x^* - x\|^2 \;\; \forall\, x, y \in \mathbb{R}^p.$$

This Asm. 4.9 is strong, so much so, we have the following lemma.

LEMMA 4.10. *Asm. 4.9 implies* (3.3) *and that the solution to* (1.1) *is unique.*

Proof. Let $y \in \mathbb{R}^p$ and let $u \in F(\mathbb{R}^p) \cap \mathbf{Ker}\,(\mathbb{E}\,[\mathbf{H_S}(y)])$. $u \in F(\mathbb{R}^p)$ implies that $\exists x \in \mathbb{R}^p$ s.t. $F(x) = u$. Besides, $u \in \mathbf{Ker}\,(\mathbb{E}\,[\mathbf{H_S}(y)])$ implies that $\mathbb{E}\,[\mathbf{H_S}(y)]\,F(x) = 0$. Now we apply (4.20) at point x knowing that $f_y(x^*) = 0$:

$$0 \geq f_y(x) + \langle \nabla f_y(x), x^* - x \rangle + \frac{\mu}{2}\|x^* - x\|^2$$

$$\implies 0 \geq 0 + \langle 0, x^* - x \rangle + \frac{\mu}{2}\|x^* - x\|^2 \quad (\text{as } \mathbb{E}\,[\mathbf{H_S}(y)]\,F(x) = 0) \iff x = x^*.$$

Thus $F(x) = u = 0$. We conclude $F(\mathbb{R}^p) \cap \mathbf{Ker}\,(\mathbb{E}\,[\mathbf{H_S}(y)]) = \{0\}$, i.e. (3.3) holds.

Besides, let x' be a global minimizer of $f_y(\cdot)$. Then $f_y(x') = f_y(x^*) = 0$ and $\nabla f_y(x') = 0$. Similarly, by applying (4.20) at point x', we obtain $x' = x^*$. Consequently, x^* is the unique minimizer of $f_y(\cdot)$ for all y, thus the unique solution to (1.1), according to (3.3) and Lemma 3.1. $\qquad\square$

Under Asm. 4.9 , choosing $\gamma = 1$ guarantees a fast global linear convergence.

THEOREM 4.11. *If x^* satisfies Asm. 4.9 and $\gamma \leq 1$, then* SNR *converges linearly:*

$$(4.21) \qquad \mathbb{E}\left[\|x^{k+1} - x^*\|^2\right] \leq (1 - \gamma\mu)^{k+1}\|x^0 - x^*\|^2 \qquad with \;\; \mu \leq 1.$$

Proof. Let $\delta_k \overset{\text{def}}{=} x^k - x^*$. By expanding the squares, similarly we have that

$$\begin{aligned}
\mathbb{E}_k\left[\|\delta_{k+1}\|^2\right] &= \|\delta_k\|^2 - 2\gamma\langle \delta_k, \nabla f_k(x^k) \rangle + \gamma^2\mathbb{E}_k\left[\|\nabla f_{\mathbf{S}_k,k}(x^k)\|^2\right] \\
&\overset{(4.20)}{\leq} (1 - \gamma\mu)\|\delta_k\|^2 - 2\gamma(f_k(x^k) - f_k(x^*)) + \gamma^2\mathbb{E}_k\left[\|\nabla f_{\mathbf{S}_k,k}(x^k)\|^2\right] \\
&\overset{(4.1)}{\leq} (1 - \gamma\mu)\|\delta_k\|^2 - 2\gamma(1 - \gamma)(f_k(x^k) - f_k(x^*)) \\
&\leq (1 - \gamma\mu)\|\delta_k\|^2. \qquad (\text{since } \gamma(1-\gamma)(f_k(x^k) - f_k(x^*)) \geq 0)
\end{aligned}$$

Now by taking total expectation, we have that

$$\mathbb{E}\left[\|x^{k+1} - x^*\|^2\right] \;\leq\; (1 - \gamma\mu)\mathbb{E}\left[\|x^k - x^*\|^2\right] \;\leq\; (1 - \gamma\mu)^{k+1}\|x^0 - x^*\|^2.$$

Next, we show that $\mu \leq 1$. In fact, when we imply (4.20) at the point x^k, it shows

$$(4.20) \quad \overset{(4.1)}{\implies} \quad f_k(x^*) \geq \frac{1}{2}\mathbb{E}_k\left[\|\nabla f_{\mathbf{S}_k,k}(x^k)\|^2\right] + \langle x^* - x^k, \nabla f_k(x^k) \rangle + \frac{\mu}{2}\|x^* - x^k\|^2$$

$$\iff \quad f_k(x^*) \geq \frac{1}{2}\mathbb{E}_k\left[\|x^* - (x^k - \nabla f_{\mathbf{S}_k,k}(x^k))\|^2\right] - \frac{1 - \mu}{2}\|x^* - x^k\|^2$$

$$\overset{f_k(x^*)=0}{\implies} (1 - \mu)\|x^* - x^k\|^2 \geq \mathbb{E}_k\left[\|x^* - (x^k - \nabla f_{\mathbf{S}_k,k}(x^k))\|^2\right] \geq 0.$$

Thus $\mu \leq 1$. $\qquad\square$

5. New global convergence theory of the NR **method.** As a direct consequence of our general convergence theorems, in this section we develop a new global convergence theory for the original NR method. We first provide the results in 1-dimension in Sec. 5.1, then a general result in higher dimensions in the subsequent Sec. 5.2 and compare this result to the classic monotone convergence theory in Sec. 5.3.

5.1. A single nonlinear equation. Consider the case where $F(x) = \phi(x) \in \mathbb{R}$ is a one dimensional function and $x \in \mathbb{R}$. This includes common applications of the NR method such as calculating square roots of their reciprocal[7] and finding roots of

---

[7]Used in particular to compute angles of incidence and reflection in games such as quake https://en.wikipedia.org/wiki/Fast_inverse_square_root

polynomials. Even in this simple one dimension case, we find that our assumptions of global convergence given in Cor. 4.5 are *strictly weaker* than the standard assumptions used to guarantee NR convergence, as we explain next.

The NR method in one dimension at every iteration k is given by

$$x^{k+1} \ = \ x^k - \frac{\phi(x^k)}{\phi'(x^k)} \ \stackrel{\text{def}}{=} \ g(x^k).$$

To guarantee that this is well defined, we assume that $\phi'(x^k) \neq 0$ for all k. A sufficient condition for this procedure to converge locally is that $|g'(x)| < 1$ with $x \in I$ where I is a given interval containing the solution x^*. See for example Section 1.1 in [17] or Chapter 12 in [47]. We can extend this to a global convergence by requiring that $|g'(x)| < 1$ globally. In the case of NR, since $g'(x) \ = \ 1 - \frac{\phi'(x)^2 - \phi(x)\phi''(x)}{\phi'(x)^2} \ = \ \frac{\phi(x)\phi''(x)}{\phi'(x)^2}$, this condition amounts to requiring

$$(5.1) \qquad \qquad \frac{|\phi(x)\phi''(x)|}{\phi'(x)^2} < 1.$$

Curiously, this condition (5.1) has an interesting connection to convexity. In fact, condition (5.1) implies that $\phi^2(x)$ is convex and twice continuously differentiable. To see this, note that $\frac{d^2}{dx^2}\phi^2(x) \geq 0$ is equivalent to

$$(5.2) \qquad \frac{d^2}{dx^2}\phi^2(x) \ = \ 2\frac{d}{dx}\phi'(x)\phi(x) \ = \ 2\left(\phi(x)\phi''(x) + \phi'(x)^2\right) \ \geq \ 0.$$

Now it is easy to see that (5.1) implies (5.2). Finally (5.2) also implies that $\phi^2(x)$ is *star-convex*, which is exactly what is required by our convergence theory in Cor. 4.5.

Indeed, in this one dimensional setting, Asm. 4.2 is equivalent to (4.4) and our reformulation in (3.1) boils down to minimizing $f_y(x) \ = \ (\phi(x)/\phi'(y))^2$. Thus by Cor. 4.5, the NR method converges globally if $f_{x^k}(x)$, or simply if $\phi(x)^2$ is star-convex and $\phi'(x^k) \neq 0$ for all iterates of NR, which shows that our condition is strictly weaker than the other conditions, because there exists functions that are star-convex but not convex, e.g. $\phi(x)^2 = |x|(1 - \exp(-|x|))$ from [45, 36].

For future reference and convenience, we can re-write the star-convexity of each $\phi(x)^2$ as

$$0 = \phi(x^*)^2 \ \geq \ \phi^2(x) + 2\phi(x)\phi'(x)(x^* - x),$$

where x^* is the global minimum of $\phi(x)^2$, i.e. $\phi(x^*) = 0$. This can be re-written as

$$(5.3) \qquad \qquad 0 \geq \phi(x)\left(\phi(x) + 2\phi'(x)(x^* - x)\right).$$

By verifying (5.3) and that $\phi'(x^k) \neq 0$ on the iterates of NR, we can guarantee that the method converges globally.

5.2. The full NR. Now let $F(x) \in \mathbb{R}^m$ and consider the full NR method (1.2). Similarly, since $\mathbf{S} = \mathbf{I}_m$, Asm. 4.2 is equivalent to (4.4). Cor. 4.5 sheds some new light on the convergence of NR. In this case, our reformulation (3.1) is given by

$$(5.4) \qquad f_y(x) \ = \ \tfrac{1}{2}F(x)^\top(DF(y)^\top DF(y))^\dagger F(x) \ = \ \tfrac{1}{2}\|\left(DF(y)^\top\right)^\dagger F(x)\|^2$$

and Cor. 4.5 states that `NR` converges if $f_{x^k}(x)$ is star-convex for all the iterates $x^k \in \mathbb{R}^p$. This has a curious re-interpretation in this setting. Indeed, let

$$(5.5) \qquad n(x) \stackrel{\text{def}}{=} -(DF(x)^\top)^\dagger F(x)$$

be the Newton direction. From (5.4) and (5.5), we have that

$$(5.6) \qquad f_x(x) = \frac{1}{2}\|n(x)\|^2.$$

Using (5.6), Cor. 4.5 can be stated in this special case as the following corollary.

COROLLARY 5.1. *Consider x^k given by the `NR` (1.2) with $\gamma < 1$. If we have*

$$(5.7) \qquad F(x) \in \mathbf{Im}(DF(x)^\top),$$

$$(5.8) \qquad \frac{1}{2}\|n(x)\|^2 \le \langle n(x), x^* - x \rangle$$

hold for every $x = x^k$ with solution x^*, then it exists $L > 0$ s.t. $\|DF(x^k)\| \le L$ and

$$(5.9) \qquad \min_{t=0,\ldots,k-1} \|F(x^t)\|^2 \le \frac{1}{k} \cdot \frac{L^2 \|x^0 - x^*\|^2}{\gamma\,(1-\gamma)}.$$

Proof. From (3.4), we have that

$$(5.10) \quad \nabla f_x(x) = DF(x)(DF(x)^\top DF(x))^\dagger F(x) = (DF(x)^\top)^\dagger F(x) = -n(x).$$

Substituting (5.6) and (5.10) in (4.4) yields (5.8). Next, for $\mathbf{S} = \mathbf{I}_m$, we have that

$$\mathbf{Im}(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right]) = \mathbf{Im}((DF(x)^\top DF(x))^\dagger) = \mathbf{Im}(DF(x)^\top DF(x)) = \mathbf{Im}(DF(x)^\top).$$

Thus, we have that $F(x) \in \mathbf{Im}(DF(x)^\top) \subset \mathbf{Im}(\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x)\right])$, i.e. (4.12) holds. So all the conditions in Cor. 4.5 are verified. Since $\mathbf{S} = \mathbf{I}_m$, we have that $\rho(x) = 1$ for all x, so $\rho = 1$. Furthermore, because we assume that $DF(\cdot)$ is continuous and the iterates x^k are in a closed bounded convex set (4.5) which is implied by (4.4) from Thm. 4.3, there exists $L > 0$ s.t. $\|DF(x^k)\| \le L$ for all the iterates. Finally, by Cor. 4.5, the iterates converge sublinearly according to (4.13) which in this case is given by (5.9). $\square$

Condition (5.8) can be seen as a co-coercivity property of the Newton direction. This co-coercivity establishes a curious link with the modern proofs of convergence of gradient descent which rely on the co-coercivity of the gradient direction. That is, if $f(x)$ is convex and L–smooth, then we have that the gradient is L–co-coercive with

$$\frac{1}{L}\|\nabla f(x)\|^2 \le \langle \nabla f(x), x - x^* \rangle.$$

This is the key property for proving convergence of gradient descent, see e.g. Section 5.2.4 in [4]. To the best of our knowledge, this is the first time that the co-coercivity of the Newton direction has been identified as a key property for proving convergence of the Newton's method. In particular, global convergence results for the `NR` method such as the monotone convergence theories (MCT) only hold for functions $F : \mathbb{R}^p \to \mathbb{R}^m$ with $p = m$ and rely on a stepsize $\gamma = 1$, see [47, 17]. Cor. 5.1 accommodates "non-square" functions $F : \mathbb{R}^p \to \mathbb{R}^m$. Excluding the difference in stepsizes and focusing on "square" functions $F : \mathbb{R}^p \to \mathbb{R}^m$ with $p = m$, next we show in Thm. 5.2 that our assumptions are strictly weaker than those used for establishing the global convergence of `NR` with constant stepsizes through the MCT.

5.3. Comparing to the classic monotone convergence theory of `NR`. Consider $m = p$. Here we show that our Asm. 1.1, (5.7) and (5.8) are strictly weaker than the classic assumptions used for establishing the global convergence of `NR` with constant stepsize. To show this, we take the assumptions used in the MCT in Section 13.3.4 in [47] and compare with our assumptions in the following theorem.

THEOREM 5.2. *Let $F : \mathbb{R}^p \to \mathbb{R}^p$ and let x^k be the iterates of the `NR` method with stepsize $\gamma = 1$, that is*

$$(5.11) \qquad\qquad x^{k+1} = x^k - \left(DF(x^k)^\top\right)^\dagger F(x^k).$$

Consider the following two sets of assumptions

(I) $F(x)$ is component wise convex, $(DF(x)^\top)^{-1}$ exists and is element-wise positive $\forall x \in \mathbb{R}^p$. There exist x and y s.t. $F(x) \le 0 \le F(y)$ element-wise.

(II) There exists a unique $x^* \in \mathbb{R}^p$ s.t. $F(x^*) = 0$, (5.7) and (5.8) hold for $k \ge 1$.

If (I) holds, then (II) always holds. Furthermore, there exist problems for which (II) holds and (I) does not hold.

Proof. First, we prove (I) $\implies$ (II). Assume that (I) holds. Since $DF(x)$ is invertible, (5.7) holds trivially. By 13.3.4 in [47], we know that there exists a unique $x^* \in \mathbb{R}^p$ s.t. $F(x^*) = 0$. It remains to verify if (5.8) holds for $k \ge 1$. First, note that the invertibility of $DF(x^k)$ gives

$$(5.12) \quad f_k(x^k) \;=\; \frac{1}{2}\|F(x^k)\|^2_{\left(DF_k^\top DF_k\right)^\dagger} \;=\; \frac{1}{2}\|(DF_k^\top)^{-1}F_k\|^2 \;\overset{(5.11)}{=}\; \frac{1}{2}\|x^{k+1} - x^k\|^2,$$

with abbreviations $f_k(x^k) \equiv f_{x^k}(x^k)$, $F_k \equiv F(x^k)$ and $DF_k \equiv DF(x^k)$. Furthermore,

$$(5.13) \quad \nabla f_k(x^k) \;=\; DF_k(DF_k^\top DF_k)^{-1}F(x^k) \;=\; (DF_k^\top)^{-1}F(x^k) \;\overset{(5.11)}{=}\; x^k - x^{k+1}.$$

Thus we can re-write the right hand side of the star-convexity assumption (4.2) as

$$f_k(x^k) + \langle \nabla f_k(x^k), x^* - x^k \rangle \overset{(5.12)+(5.13)}{=} \frac{1}{2}\|x^{k+1} - x^k\|^2 + \langle x^k - x^{k+1}, x^* - x^k \rangle$$

$$= \frac{1}{2}\|x^{k+1} - x^k\|^2 + \langle x^k - x^{k+1}, x^{k+1} - x^k + x^* - x^{k+1} \rangle$$

$$= -\frac{1}{2}\|x^{k+1} - x^k\|^2 + \langle x^k - x^{k+1}, x^* - x^{k+1} \rangle.$$

From (I), we induce by Lemma 3.1 in [17] that `NR` is component wise monotone with $x^* \le x^{k+1} \le x^k$ for $k \ge 1$. Thus $x^k - x^{k+1} \ge 0$ and $x^* - x^{k+1} \le 0$ component wise and consequently, $\langle x^k - x^{k+1}, x^* - x^{k+1} \rangle \le 0$. Thus it follows that

$$f_k(x^k) + \langle \nabla f_k(x^k), x^* - x^k \rangle \;\le\; 0 \;=\; f_k(x^*).$$

Thus (5.8) holds for $k \ge 1$ and this concludes that (I) $\implies$ (II).

We now prove that (II) does **not** imply (I). Consider the example $F(x) = Ax - b$, where $A \in \mathbb{R}^{p \times p}$ is invertible and $b \in \mathbb{R}^p$. Thus, $DF(x) = A^\top$ is invertible and (5.7) holds. As for (5.8), let x^* be the solution, i.e. $Ax^* = b$, we have that

$$f_k(x) = \frac{1}{2}\|F(x)\|^2_{(DF(x_k)^\top DF(x_k))^{-1}} = \frac{1}{2}\|A(x - x^*)\|^2_{(AA^\top)^{-1}} = \frac{1}{2}\|x - x^*\|^2,$$

which is a convex function and so (5.8) holds and thus (II) holds. However, (I) does not necessarily hold. Indeed, if $A = -\mathbf{I}_p$, then $DF(x)$ is not element-wise positive. $\square$

We observe that our assumptions are also strictly weaker than the affine covariates formulations of convex functions given in Lemma 3.1 in [17]. The proof is verbatim to the above.

Thm. 5.2 only considers the case that the stepsize $\gamma = 1$. We also investigate the case where the stepsize $\gamma < 1$ in particular in 1-dimension and show that MCT does not hold in this case in App. C. Thus we claim that our assumptions are strictly weaker than the assumptions used in MCT [47, 17] for establishing the global convergence of NR, albeit for different step sizes.

6. Single row sampling: the nonlinear Kaczmarz method. The SNR enjoys many interesting instantiations. Among which, we have chosen three to present in the main text: the nonlinear Kaczmarz method in this section, the Stochastic Newton method [51, 35] in Sec. 7 and a new specialized variant for solving GLMs in Sec. 8.

Here we present the new nonlinear Kaczmarz method as a variant of SNR. Consider the original problem (1.1). We use a single row importance weighted subsampling sketch to sample rows of $F(x) = 0$. That is, let $\mathbb{P}[\mathbf{S} = e_i] = p_i$ with the ith unit coordinate vector $e_i \in \mathbb{R}^m$ for $i = 1, \cdots, m$. Then the SNR update (1.3) is given by

$$(6.1) \qquad x^{k+1} = x^k - \gamma \frac{F_i(x^k)}{\|\nabla F_i(x^k)\|^2} \nabla F_i(x^k).$$

We dub (6.1) the *nonlinear Kaczmarz method*, as it can be seen as an extension of the randomized Kaczmarz method [29, 56] for solving linear systems to the nonlinear case[8]. By (3.2), this nonlinear Kaczmarz method is simply SGD applied to minimizing

$$f_{x^k}(x) = \sum_{i=1}^{m} \mathbb{P}[\mathbf{S} = e_i] f_{e_i, x^k}(x) \overset{(3.2)+(1.5)}{=} \frac{1}{2} \sum_{i=1}^{m} p_i \frac{F_i(x)^2}{\|\nabla F_i(x^k)\|^2}.$$

A sufficient condition for (3.3) to hold is that the diagonal matrix

$$(6.2) \qquad \mathbb{E}_{e_i}\left[\mathbf{H}_{e_i}(x^k)\right] \overset{(1.5)}{=} \sum_{i=1}^{m} p_i \frac{e_i e_i^\top}{\|\nabla F_i(x^k)\|^2} = \mathbf{Diag}\left(\frac{p_i}{\|\nabla F_i(x^k)\|^2}\right)$$

is invertible. Thus $\mathbb{E}\left[\mathbf{H}_\mathbf{S}(x^k)\right]$ is invertible if $\nabla F_i(x^k) \neq 0$ for all $i \in \{1, \cdots, m\}$ and $x^k \in \mathbb{R}^p$. In which case $\mathbf{Ker}\left(\mathbb{E}[\mathbf{H}_\mathbf{S}(y)]\right) = \{0\}$ for all $y \in \mathbb{R}^p$ and (3.3) holds.

Finally, to guarantee that (6.1) converges through Thm. 4.3, we need $f_{x^k}(x)$ to be star-convex on x^k at every iteration. In this case, it suffices for each $F_i(x)^2$ to be star-convex, since any conic combination of star-convex functions is star-convex [36]. This is a straightforward abstraction of the one dimension case, in that, if (5.3) holds for every F_i in the place of ϕ, we can guarantee the convergence of (6.1). This is also equivalent to assuming the star-convexity for each sketching matrix (4.4). Furthermore, if we have $F(x) \in \mathbf{Im}(DF(x)^\top)$ hold for all x, then (4.12) holds, as $\mathbf{Ker}\left(\mathbb{E}[\mathbf{H}_\mathbf{S}(y)]\right) = \{0\}$. We can guarantee the convergence of (6.1) through Cor. 4.5.

7. The Stochastic Newton method. We now show that the Stochastic Newton method (SNM) [51, 35] is a special case of SNR. This connection combined with the global convergence theory of SNR, gives us the first global convergence theory of SNM, which we detail in Sec. 7.2.

---

[8]We note that there exists a nonlinear variant of the Kaczmarz method which is referred to as the Landweber–Kaczmarz method [37]. Though the Landweber–Kaczmarz is very similar to Kaczmarz, it is not truly an extension since it does not adaptively re-weight the stepsize by $\|\nabla F_i(x^k)\|^2$.

SNM [35] is a stochastic second order method that takes a Newton-type step at each iteration to solve optimization problems with a finite-sum structure

$$\text{(7.1)} \qquad \min_{w \in \mathbb{R}^d} \left[P(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \phi_i(w) \right],$$

where each $\phi_i : \mathbb{R}^d \to \mathbb{R}$ is twice differentiable and strictly convex. In brevity, the updates in SNM at the kth iteration are given by

$$\text{(7.2)} \qquad w^{k+1} \;=\; \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i^k) \right),$$

$$\text{(7.3)} \qquad \alpha_i^{k+1} \;=\; \begin{cases} w^{k+1} & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases},$$

where $\alpha_1^k, \cdots, \alpha_n^k$ are auxiliary variables, initialized in SNM, and $B_n \subset \{1,\ldots,n\}$ is a subset of size τ chosen uniformly on average from all subsets of size τ.

7.1. Rewrite SNM as a special case of SNR. Since $P(w)$ is strictly convex, every minimizer of P satisfies $\nabla P(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(w) = 0$. Our main insight to deducing SNM is that we can re-write this stationarity condition using a *variable splitting trick*. That is, by introducing a new variable $\alpha_i \in \mathbb{R}^d$ for each gradient $\nabla \phi_i$, and let $p := (n+1)d$ and $x = \left[w \,; \alpha_1 \,; \cdots \,; \alpha_n \right] \in \mathbb{R}^p$ be the stacking[9] of the w and α_i variables, we have that solving $\nabla P(w) = 0$ is equivalent to finding the *roots* of the following nonlinear equations

$$\text{(7.4)} \quad F(x) \;=\; F(w \,; \alpha_1 \,; \cdots \,; \alpha_n) \;\stackrel{\text{def}}{=}\; \left[\frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i) \,; \; w - \alpha_1 \,; \; \cdots \,; \; w - \alpha_n \right],$$

where $F : \mathbb{R}^{(n+1)d} \to \mathbb{R}^{(n+1)d}$. Our objective now becomes solving $F(x) = 0$ with $p = m = (n+1)d$. To apply SNR to (7.4), we are going to use a structured sketching matrix. But first, we need some additional notations.

Divide $\mathbf{I}_{nd} \in \mathbb{R}^{nd \times nd}$ into n contiguous blocks of size $nd \times d$ as follows

$$\mathbf{I}_{nd} \stackrel{\text{def}}{=} [\; \mathbf{I}_{nd,1} \; \mathbf{I}_{nd,2} \cdots \mathbf{I}_{nd,n} \;]$$

where $\mathbf{I}_{nd,i}$ is the ith block of $\mathbf{I}_{nd}$. Let $B_n \subset \{1,\ldots,n\}$ with $|B_n| = \tau$ chosen uniformly at average. Let $\mathbf{I}_{B_n} \in \mathbb{R}^{nd \times \tau d}$ denote the concatenation of the blocks $\mathbf{I}_{nd,i}$ such that the indices $i \in B_n$.

At the kth iteration of SNR, denote $x^k = [w^k; \alpha_1^k; \cdots; \alpha_n^k]$, we define our sketching matrix $\mathbf{S} \sim \mathcal{D}_{x^k}$ as

$$\text{(7.5)} \qquad \mathbf{S} = \begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{1}{n} \nabla^2 \phi_1(\alpha_1^k) & \\ \vdots & \mathbf{I}_{B_n} \\ \frac{1}{n} \nabla^2 \phi_n(\alpha_n^k) & \end{bmatrix} \in \mathbb{R}^{(n+1)d \times (\tau+1)d}.$$

Here the distribution $\mathcal{D}_{x^k}$ depends on the iterates x^k. The sketch size of $\mathbf{S}$ is $(\tau+1)d$ with *any* $\tau \in \{1, \cdots, n\}$. Now we can state the following lemma.

---

[9] In this paper, vectors are columns by default, and given $x_1, \ldots, x_n \in \mathbb{R}^q$, we note $[x_1; \ldots; x_n] \in \mathbb{R}^{qn}$ the (column) vector stacking the x_i's on top of each other with $q \in \mathbb{N}$.

LEMMA 7.1. *Let ϕ_i be strictly convex for $i = 1, \ldots, n$. At each iteration k, the updates of* SNR *(1.3) with F defined in (7.4), the sketching matrix $\mathbf{S}_k$ defined in (7.5) , and stepsize $\gamma = 1$, are equal to the updates (7.2) and (7.3) of* SNM.

In our upcoming proof of Lemma 7.1, we still need the following lemma.

LEMMA 7.2. *Let ϕ_i be twice differentiable and strictly convex for $i = 1, \ldots, n$. The Jacobian $DF(x)^\top$ of $F(x)$ defined in (7.4) is invertible for all $x \in \mathbb{R}^{(n+1)d}$.*

Proof. Let $x \in \mathbb{R}^{(n+1)d}$. Let $y \stackrel{\text{def}}{=} (u; v_1; \cdots; v_n) \in \mathbb{R}^{(n+1)d}$ with $u, v_1, \cdots, v_n \in \mathbb{R}^d$ such that $DF(x)y = 0$. The transpose of the Jacobian of $F(x)$ is given by

$$(7.6) \qquad DF(x) = \begin{bmatrix} 0 & \mathbf{I}_d & \cdots & \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & -\mathbf{I}_{nd} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix}.$$

From $DF(x)y = 0$ and (7.6), we obtain

$$\sum_{i=1}^n v_i = 0, \quad \text{and} \quad \frac{1}{n}\nabla^2\phi_i(\alpha_i)u = v_i \quad \text{for all } i = 1, \cdots, n.$$

Plugging the second equation in the first one gives $\left(\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i)\right) u = 0$. Since every ϕ_i is twice differentiable and strictly convex, we have $\nabla^2\phi_i(\alpha_i) > 0$. This implies $\frac{1}{n}\sum_{i=1}^n \nabla^2\phi_i(\alpha_i) > 0$, and is thus invertible. Consequently $u = 0$ and $v_i = 0$ from which we conclude that the Jacobian $DF(x)^\top$ is invertible. □

Now we can give the proof of Lemma 7.1.

Proof. Consider an update of SNR (1.3) with F defined in (7.4), the sketching matrix $\mathbf{S}_k$ defined in (7.5) , and stepsize $\gamma = 1$ at the kth iteration. By Lemma 7.2, we have that $DF(x)$ is invertible and thus Asm. 2.1 holds. By (2.2), the SNR update (1.3) can be re-written as

(7.7)
$$x^{k+1} = \operatorname{argmin} \|w - w^k\|^2 + \sum_{i=1}^n \|\alpha_i - \alpha_i^k\|^2 \text{ s.t. } \mathbf{S}_k^\top DF(x^k)^\top(x - x^k) = -\mathbf{S}_k^\top F(x^k).$$

Plugging (7.4), (7.5) and (7.6) into the constraint in (7.7) gives

$$\begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1^k) & \\ \vdots & \mathbf{I}_{B_n} \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n^k) & \end{bmatrix}^\top \begin{bmatrix} 0 & \mathbf{I}_d & \cdots & \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & -\mathbf{I}_{nd} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix}^\top \begin{bmatrix} w - w^k \\ \alpha_1 - \alpha_1^k \\ \vdots \\ \alpha_n - \alpha_n^k \end{bmatrix}$$

$$= -\begin{bmatrix} \mathbf{I}_d & 0 \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1^k) & \\ \vdots & \mathbf{I}_{B_n} \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n^k) & \end{bmatrix}^\top \begin{bmatrix} \frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\alpha_i^k) \\ w^k - \alpha_1^k \\ \vdots \\ w^k - \alpha_n^k \end{bmatrix}.$$

After simplifying the above matrix multiplications, we have that (7.7) is given by

$$x^{k+1} = [w^{k+1}; \alpha_1^{k+1}; \cdots; \alpha_n^{k+1}] = \text{argmin} \, \|w - w^k\|^2 + \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\text{s. t.} \quad \frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k)(w - \alpha_i^k) = -\frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i^k),$$

(7.8) $$w = \alpha_j, \quad \text{for } j \in B_n.$$

To solve (7.8), first note that $\alpha_i^{k+1} = \alpha_i^k$ for $i \notin B_n$, since there is no constraint on the variable α_i in this case. Furthermore, by the invertibility of $\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k)$, we have that (7.8) has a unique solution s.t. $\alpha_j = w$ for all $j \in B_n$ and

$$w = \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i^k) \right).$$

Concluding, we have that the SNR update (7.8) is given by

$$w^{k+1} = \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i^k) \right),$$

$$\alpha_i^{k+1} = \begin{cases} w^{k+1} & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases},$$

which is exactly the Stochastic Newton method's updates (7.2) and (7.3) in [35]. □

Thus we conclude that SNM is a special case of SNR. However, in practice for solving GLMs, instead of sampling $\mathbf{S} \sim \mathcal{D}_{x^k}$ provided in (7.5), we only sample B_n and we execute the efficient updates as suggested in [35].

7.2. Global convergence theory of SNM. Let $x' \stackrel{\text{def}}{=} (w'; \alpha_1'; \cdots; \alpha_n') \in \mathbb{R}^{(n+1)d}$ and $\mathbf{S} \sim \mathcal{D}_x$ defined in (7.5). By applying the global convergence theory of SNR, we can now provide the first global convergence theory for SNM.

COROLLARY 7.3. *Let w^* be a solution to $\nabla P(w) = 0$. Consider the iterate $x^k = (w^k; \alpha_1^k; \cdots; \alpha_n^k)$ given by SNM (7.2) and (7.3) and note $x^* \stackrel{\text{def}}{=} (w^*; w^*; \cdots; w^*) \in \mathbb{R}^{(n+1)d}$. If there exists $\mu > 0$ such that for all $x, x' \in \mathbb{R}^{(n+1)d}$,*

(7.9) $$f_{x'}(x^*) \geq f_{x'}(x) + \langle \nabla f_{x'}(x), x^* - x \rangle + \frac{\mu}{2} \|x^* - x\|^2$$

$$= f_{x'}(x) + \langle \nabla f_{x'}(x), x^* - x \rangle + \frac{\mu}{2} \left(\|w^* - w\|^2 + \sum_{i=1}^{n} \|w^* - \alpha_i\|^2 \right),$$

then the iterates $\{x^k\}$ of SNM converge linearly according to

(7.10) $$\mathbb{E} \left[\|x^{k+1} - x^*\|^2 \right] \leq (1 - \mu)^{k+1} \|x^0 - x^*\|^2.$$

Proof. As $\nabla P(w^*) = 0$, this implies immediately that x^* is a solution of F. Besides, (7.9) satisfies Asm. 4.9. Thus by Thm. 4.11, we get (7.10). □

Even though (7.9) is a strong assumption, this is the first global convergence theory of SNM, since only local convergence results of SNM are addressed in [35].

As a by-product, we find that the function $F(x)$ in (7.4) and the sketch $\mathbf{S}$ defined in (7.5) actually satisfy (4.12) through Lemma 4.6, namely as the following lemma.

LEMMA 7.4. *Consider the function F defined in* (7.4) *and the sketching matrix* **S** *defined in* (7.5), *then we have the condition* (4.12) *hold.*

Proof. First, we show that $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is invertible $\forall x \in \mathbb{R}^{(n+1)d}$. By the definition of **S** in (7.5),

$$
\mathbf{S}\mathbf{S}^\top = \begin{bmatrix} \mathbf{I}_d & \frac{1}{n}\nabla^2\phi_1(\alpha_1) & \cdots & \frac{1}{n}\nabla^2\phi_n(\alpha_n) \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & \mathbf{I}_{B_n}\mathbf{I}_{B_n}^\top + \mathbf{M} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix},
$$

where $\mathbf{M} = \{\mathbf{M}_{ij}\}_{1\leq i\leq n, 1\leq j\leq n}$ is divided into $n \times n$ contiguous blocks of size $d \times d$ with each block $\mathbf{M}_{ij}$ defined as the following

$$
\mathbf{M}_{ij} \stackrel{\text{def}}{=} \frac{1}{n}\nabla^2\phi_i(\alpha_i) \cdot \frac{1}{n}\nabla^2\phi_j(\alpha_j) \in \mathbb{R}^{d\times d} \qquad \text{and} \qquad \mathbf{M} \in \mathbb{R}^{nd\times nd}.
$$

Taking the expectation over **S** w.r.t. the distribution $\mathcal{D}_{(w,\alpha_1,\cdots,\alpha_n)}$ gives

$$
\begin{aligned}
\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right] &= \begin{bmatrix} \mathbf{I}_d & \frac{1}{n}\nabla^2\phi_1(\alpha_1) & \cdots & \frac{1}{n}\nabla^2\phi_n(\alpha_n) \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) & & & \\ \vdots & & \frac{\tau}{n}\mathbf{I}_{nd} + \mathbf{M} & \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) & & & \end{bmatrix} \\
(7.11) \qquad &= \begin{bmatrix} \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) \\ \vdots \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) \end{bmatrix} \begin{bmatrix} \mathbf{I}_d \\ \frac{1}{n}\nabla^2\phi_1(\alpha_1) \\ \vdots \\ \frac{1}{n}\nabla^2\phi_n(\alpha_n) \end{bmatrix}^\top + \frac{\tau}{n}\begin{bmatrix} 0 & 0 \\ 0 & \mathbf{I}_{nd.} \end{bmatrix}
\end{aligned}
$$

$\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is symmetric, positive semi-definite. Let $(u; v_1; \cdots; v_n) \in \mathbb{R}^{(n+1)d}$ s.t.

$$
(u; v_1; \cdots; v_n)^\top \mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right](u; v_1; \cdots; v_n) = 0.
$$

From (7.11), we obtain

$$
\left\| \left[\mathbf{I}_d; \frac{1}{n}\nabla^2\phi_1(\alpha_1); \cdots; \frac{1}{n}\nabla^2\phi_n(\alpha_n)\right]^\top [u; v_1; \cdots; v_n] \right\|^2 + \frac{\tau}{n}\sum_{i=1}^n \|v_i\|^2 = 0.
$$

Since both terms are non negative, we obtain $\sum_{i=1}^n \|v_i\|^2 = 0 \implies \forall\, i, v_i = 0$, and then $u = 0$. This confirms that $\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]$ is positive-definite, thus invertible and $\mathbf{Ker}(\mathbb{E}\left[\mathbf{S}\mathbf{S}^\top\right]) = \{0\}$. Besides, from Lemma 7.2, we get $DF(x)$ invertible. Thus $F(x) \in \mathbf{Im}(DF(x)^\top)$ and $\mathbf{Ker}(DF(x)) = \{0\}$. We have (4.17) hold. By Lemma 4.6, we have that (4.12) holds for all $x \in \mathbb{R}^{(n+1)d}$. □

From Lemma 7.4, we know that for *any* size of the subset sampling $|B_n| = \tau \in \{1, \cdots, n\}$, the condition (4.12) holds. The corresponding sketch size of **S** is $(\tau+1)d$.

Furthermore, using Lemma 7.1, we can also provide the global convergence of SNM with stepsizes $\gamma < 1$ (SNM with relaxation) by using the weaker star-convexity assumption and Thm. 4.3. We expand on this comment in App. D in the appendix.

8. Applications to GLMs – *tossing-coin-sketch* method. Consider the problem of training a generalized linear model

$$(8.1) \qquad w^* = \arg\min_{w\in\mathbb{R}^d} P(w) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \phi_i(a_i^\top w) + \frac{\lambda}{2}\|w\|^2,$$

where $\phi_i : \mathbb{R} \to \mathbb{R}^+$ is a convex and continuously twice differentiable loss function, $a_i \in \mathbb{R}^d$ are data samples and $w \in \mathbb{R}^d$ is the parameter to optimize. As the objective function is strongly convex, the unique minimizer satisfies $\nabla P(w) = 0$, that is

$$(8.2) \qquad \nabla P(w) = \frac{1}{n}\sum_{i=1}^{n} \phi_i'(a_i^\top w)a_i + \lambda w = 0.$$

Let $\Phi(w) \overset{\text{def}}{=} \begin{bmatrix}\phi_1'(a_1^\top w) & \cdots & \phi_n'(a_n^\top w)\end{bmatrix}^\top \in \mathbb{R}^n$ and $\mathbf{A} \overset{\text{def}}{=} \begin{bmatrix}a_1 & \cdots & a_n\end{bmatrix} \in \mathbb{R}^{d\times n}$. By introducing auxiliary variables $\alpha_i \in \mathbb{R}$ s.t. $\alpha_i \overset{\text{def}}{=} -\phi_i'(a_i^\top w)$, we can re-write (8.2) as

$$(8.3) \qquad w = \frac{1}{\lambda n}\mathbf{A}\alpha, \qquad \text{and} \qquad \alpha = -\Phi(w).$$

Note $x = [\alpha; w] \in \mathbb{R}^{n+d}$. The objective of finding the minimum of (8.1) is now equivalent to finding *zeros* for the function

$$(8.4) \qquad F(x) = F(\alpha; w) \overset{\text{def}}{=} \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}\alpha - w \\ \alpha + \Phi(w) \end{bmatrix},$$

where $F : \mathbb{R}^{n+d} \to \mathbb{R}^{n+d}$. Our objective now becomes solving $F(x) = 0$ with $p = m = n + d$. For this, we will use a variant of the SNR. The advantage in representing (8.2) as the nonlinear system (8.4) is that we now have one row per data point (see the second equation in (8.3)). This allows us to use sketching to *subsample* the data.

Since the function F has a block structure, we will use a structured sketching matrix which we refer to as a *Tossing-coin-sketch*. But first, we need the following definition of a block sketch.

DEFINITION 8.1 ((n, τ)–block sketch). *Let $B_n \subset \{1, \ldots, n\}$ be a subset of size τ uniformly sampling at random. We say that $\mathbf{S} \in \mathbb{R}^{n\times\tau}$ is a (n, τ)–block sketch if $\mathbf{S} = \mathbf{I}_{B_n}$ where $\mathbf{I}_{B_n}$ denotes the column concatenation of the columns of the identity matrix $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ whose indices are in B_n.*

Our Tossing-coin-sketch is a sketch that alternates between two blocks depending on the result of a coin toss.

DEFINITION 8.2 (Tossing-coin-sketch). *Let $\mathbf{S}_d \in \mathbb{R}^{d\times\tau_d}$ and $\mathbf{S}_n \in \mathbb{R}^{n\times\tau_n}$ be a (d, τ_d)–block sketch and a (n, τ_n)–block sketch, respectively. Let $b \in (0; 1)$. Now each time we sample $\mathbf{S}$, we "toss a coin" to determine the structure of $\mathbf{S} \in \mathbb{R}^{(d+n)\times(\tau_d+\tau_n)}$. That is, $\mathbf{S} = \begin{bmatrix}\mathbf{S}_d & 0 \\ 0 & 0\end{bmatrix}$ with probability $1 - b$ and $\mathbf{S} = \begin{bmatrix}0 & 0 \\ 0 & \mathbf{S}_n\end{bmatrix}$ with probability b.*

By applying the SNR method with a tossing-coin-sketch for solving (8.4), we arrive at an efficient method for solving (8.1) that we call the *TCS* method. By using a tossing-coin-sketch, we can alternate between solving a linear system based on the first d rows of (8.4) and a nonlinear system based on the last n rows of (8.4).

TCS is inspired by the first-order stochastic dual ascent methods [55, 54, 49]. Indeed, eq. (8.3) can be seen as primal-dual systems with primal variables w and

dual variables α. Stochastic dual ascent methods are efficient to solve (8.3). At each iteration, they update alternatively the primal and the dual variables w and α with the first-order informations. Thus, by sketching alternatively the primal and the dual systems and updating accordingly with the Newton-type steps, TCS's updates can be seen as the second-order stochastic dual ascent methods.

We show in the next section that the TCS method verifies (4.12). Using sketch sizes s.t. $\tau_n \ll n$, the TCS method has the same cost as `SGD` in the case $d \ll n$. The low computational cost per iteration is thus another advantage of the TCS method. See App. H the cost per iteration analysis. For a detailed derivation of the TCS method, see App. F and a detailed implementation in Algorithm 4 in the appendix.

8.1. The condition (4.12) **in the case of TCS method.** In this section, we show that the TCS method verifies (4.12) through Lemma 4.6 in the following.

LEMMA 8.3. *Consider the function F defined in* (8.4) *and the tossing-coin-sketch* **S** *defined in Definition 8.2, then* (4.12) *holds.*

Proof. First, we show that $\mathbb{E}\left[\mathbf{SS}^\top\right]$ is invertible. By Definition 8.2, it is straightforward to verify that

$$\mathbb{E}\left[\mathbf{SS}^\top\right] = \begin{bmatrix} \frac{(1-b)\tau_d}{n}\mathbf{I}_d & 0 \\ 0 & \frac{b\tau_n}{n}\mathbf{I}_n \end{bmatrix}$$

is invertible and $\mathbf{Ker}(\mathbb{E}\left[\mathbf{SS}^\top\right]) = \{0\}$. Now we show the Jacobian $DF^\top(x)$ invertible. Let $x = [\alpha; w] \in \mathbb{R}^{n+d}$ with $\alpha \in \mathbb{R}^n$ and $w \in \mathbb{R}^d$. Then $DF(x)$ is written as

$$(8.5) \qquad\qquad DF(x)^\top = \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A} & -\mathbf{I}_d \\ \mathbf{I}_n & \nabla\Phi(w)^\top \end{bmatrix},$$

where $\nabla\Phi(w)^\top = \mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_n''(a_n^\top w)\right)\mathbf{A}^\top \in \mathbb{R}^{n\times d}$. Denote the diagonal matrix $D(w) \stackrel{\text{def}}{=} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_n''(a_n^\top w)\right)$. Since ϕ_i is continuously twice differentiable and convex, $\phi_i''(a_i^\top w) \geq 0$ for all i. Thus, $D(w) \geq 0$.

Let $(u; v) \in \mathbb{R}^{n+d}$ with $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^d$ such that $DF(x)^\top[u; v] = 0$. We have

$$(8.6)$$
$$DF(x)^\top \begin{bmatrix} u \\ v \end{bmatrix} = 0 \stackrel{(8.5)}{\Longleftrightarrow} \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A} & -\mathbf{I}_d \\ \mathbf{I}_n & \nabla\Phi(w)^\top \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = 0 \Longrightarrow \left(\mathbf{I}_n + \frac{1}{\lambda n}D(w)\mathbf{A}^\top\mathbf{A}\right)u = 0.$$

If $D(w)$ is invertible, (8.6) becomes

$$(8.7) \quad D(w)\left(D(w)^{-1} + \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A}\right)u = 0 \Longleftrightarrow \left(D(w)^{-1} + \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A}\right)u = 0.$$

Since $D(w)$ is invertible, i.e. $D(w) > 0$, we obtain $D(w)^{-1} > 0$. As $\frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A} \geq 0$, we get $D(w)^{-1} + \frac{1}{\lambda n}\mathbf{A}^\top\mathbf{A} > 0$, thus invertible. From (8.7), we get $u = 0$.

Otherwise, $D(w)$ is not invertible. Without losing generality, we assume that $\phi_1''(a_1^\top w) \geq \phi_2''(a_2^\top w) \geq \cdots \geq \phi_n''(a_n^\top w) = 0$. Let j be the largest index for which $\phi_j''(a_j^\top w) > 0$. If j does not exist, then $D(w) = 0$. From (8.6), we get $u = 0$ directly. If j exists, we have $1 \leq j < n$ and

$$D(w)\mathbf{A}^\top\mathbf{A} = \mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_j''(a_j^\top w), 0, \ldots, 0\right)\mathbf{A}^\top\mathbf{A}$$
$$(8.8) \qquad\qquad = \begin{bmatrix} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_j''(a_j^\top w)\right)\mathbf{A}_{1:j}^\top\mathbf{A}_{1:j} & 0 \\ 0 & 0 \end{bmatrix},$$

where $\mathbf{A}_{1:j} \overset{\text{def}}{=} [a_1 \; \cdots \; a_j] \in \mathbb{R}^{d \times j}$. Note $u = [u_1; \cdots; u_n] \in \mathbb{R}^n$. Plugging (8.8) into (8.6), we get

$$\left(\mathbf{I}_n + \frac{1}{\lambda n} \begin{bmatrix} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_j''(a_j^\top w) \right) \mathbf{A}_{1:j}^\top \mathbf{A}_{1:j} & 0 \\ 0 & 0 \end{bmatrix} \right) u = 0$$

$$(8.9) \quad \Longleftrightarrow \begin{cases} \left(\mathbf{I}_j + \frac{1}{\lambda n} \mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_j''(a_j^\top w) \right) \mathbf{A}_{1:j}^\top \mathbf{A}_{1:j} \right) u_{1:j} = 0 \\ u_{(j+1):n} = 0 \end{cases},$$

where $u_{1:j} \overset{\text{def}}{=} [u_1; \cdots; u_j] \in \mathbb{R}^j$ and $u_{(j+1):n} \overset{\text{def}}{=} [u_{j+1}; \cdots; u_n] \in \mathbb{R}^{n-j}$. From (8.9), $u_{(j+1):n} = 0$. Now $\mathbf{Diag}\left(\phi_1''(a_1^\top w), \ldots, \phi_j''(a_j^\top w) \right)$ is invertible in the subspace $\mathbb{R}^j$ as every coordinate in the diagonal $\phi_i''(a_i^\top w)$ is strictly positive for all $1 \le i \le j$. Similarly, we obtain $u_{1:j} = 0$ from the first equation of (8.9). Overall we get $u = 0$.

Thus, in all cases, $u = 0$, then $v = \frac{1}{\lambda n} Au = 0$. We can thus induce that $DF(\alpha; w)^\top$ is invertible for all α and w. Similar to Lemma 7.4, we have (4.17) hold, and by Lemma 4.6, we have that (4.12) holds. □

From Lemma 8.3, we know that for *any* size of the block sketch $\tau_d \in \{1, \cdots, d\}$ and $\tau_n \in \{1, \cdots, n\}$, (4.12) holds. The corresponding sketch size of $\mathbf{S}$ is $\tau_d + \tau_n$.

8.2. Experiments for TCS method applied for GLM. We consider the logistic regression problem with 8 datasets[10] taken from LibSVM [12], except for one artificial dataset. Table 1 provides the details of these datasets, including the *condition number* (C.N.) of the model and the smoothness constant L of the model. C.N. of the logistic regression problem is given by

$$\text{C.N.} \overset{\text{def}}{=} \frac{\lambda_{\max}(\mathbf{A}\mathbf{A}^\top)}{4n\lambda} + 1,$$

where $\lambda_{\max}(\cdot)$ is the largest eigenvalue operator. The smoothness constant L is given by

$$L \overset{\text{def}}{=} \frac{\lambda_{\max}(\mathbf{A}\mathbf{A}^\top)}{4n} + \lambda.$$

As for the logistic regression problem, we consider the loss function ϕ_i in (8.1) in the form

$$\phi_i(t) = \ln(1 + e^{-y_i t})$$

where y_i are the target values for $i = 1, \cdots, n$.

The artificial dataset. The artificial dataset $\mathbf{A}^\top \in \mathbb{R}^{n \times d}$ in Table 1 is of size 10000×50 and generated by a Gaussian distribution whose mean is zero and covariance is a Toeplitz matrix. Toeplitz matrices are completely determined by their diagonal. We set the diagonal of our Toeplitz matrix as

$$[c^0; c^1; \cdots; c^{d-1}] \in \mathbb{R}^d$$

where $c \in \mathbb{R}^+$ is a parameter. We choose $c = 0.9$ (closed to 1) which results in $\mathbf{A}$ having highly correlated columns, which in turn makes $\mathbf{A}$ an ill-conditioned data set. We set the ground truth coefficients of the model

$$\mathbf{w} = [(-1)^0 \cdot e^{-\frac{0}{10}}; \cdots; (-1)^{d-1} \cdot e^{-\frac{d-1}{10}}] \in \mathbb{R}^d$$

---

[10]All datasets except for the artificial dataset can be found downloaded on https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/. Some of the datasets can be found originally in [33, 7, 41, 60, 18].

TABLE 1
Details of the data sets for binary classification

dataset	dimension (d)	samples (n)	C.N. of the model	L
covetype	54	581012	7.45×10^{12}	1.28×10^{7}
a9a	123	32561	5.12×10^{4}	1.57
fourclass	2	862	4.86×10^{6}	5.66×10^{3}
artificial	50	10000	3.91×10^{4}	3.91
ijcnn1	22	49990	2.88×10^{3}	5.77×10^{-2}
webspam	254	350000	7.47×10^{4}	2.13×10^{-1}
epsilon	2000	400000	3.51×10^{4}	8.76×10^{-2}
phishing	68	11055	1.04×10^{3}	9.40×10^{-2}

and the target values of the dataset

$$\mathbf{y} = \mathrm{sgn}\left(\mathbf{A}^{\top}\mathbf{w} + \mathbf{r}\right) \in \mathbb{R}^{n}$$

where $\mathbf{r} \in \mathbb{R}^{n}$ is the noise generated from a standard normal distribution.

We compare the TCS method with SAG [53], SVRG [28], dfSDCA [54] and Quartz [49]. All experiments were initialized at $w^0 = 0 \in \mathbb{R}^d$ (and/or $\alpha^0 = 0 \in \mathbb{R}^n$ for TCS/dfSDCA methods) and were performed in Python 3.7.3 on a laptop with an Intel Core i9-9980HK CPU and 32 Gigabyte of DDR4 RAM running OSX 10.14.5. For all methods, we used the stepsize that was shown to work well in practice. For instance, the common rule of thumb for SAG and SVRG is to use a stepsize $\frac{1}{L}$, where L is the smoothness constant. This rule of thumb stepsize is not supported by theory. Indeed for SAG, the theoretical stepsize is $\frac{1}{16L}$ and it should be even smaller for SVRG depending on the condition number. For dfSDCA and Quartz's, we used the stepsize suggested in the experiments in [54] and [49] respectively. For TCS, we used two types of stepsize, related to the C.N. of the model. If the condition number is big (Fig. 1 top row), we used $\gamma = 1$ except for a9a with $\gamma = 1.5$. If the condition number is small (Fig. 1 bottom row), we used $\gamma = 1.8$. We also set the Bernoulli parameter b (probability of the coin toss) depending on the size of the dataset (see Table 2 in Sec. H), and $\tau_d = d$. We tested three different sketch sizes $\tau_n = 50, 150, 300$. More details of the parameter settings are presented in Sec. H.

We used $\lambda = \frac{1}{n}$ regularization parameter, evaluated each method 10 times and stopped once the gradient norm[11] was below 10^{-5} or some maximum time had been reached. In Fig. 1, we plotted the central tendency as a solid line and all other executions as a shaded region for the wall-clock time vs gradient norm.

From Fig. 1, TCS outperforms all other methods on ill-conditioned problems (Fig. 1 top row), but not always the case on well-conditioned problems (Fig. 1 bottom row). This is because in ill-conditioned problems, the curvature of the optimization function is not uniform over directions and varies in the input space. Second-order methods effectively exploit information of the local curvature to converge faster than first-order methods. To further illustrate the performance of TCS on ill-conditioned problems, we compared the performance of TCS on the artificial dataset in the top right of Fig. 1. Note as well that for reaching an approximate solution at early stage (i.e. $tol = 10^{-3}, 10^{-4}$), TCS is very competitive on all problems. TCS also has the

---

[11]We evaluated the true gradient norm every 1000 iterations. We also paused the timing when computing the performance evaluation of the gradient norm.
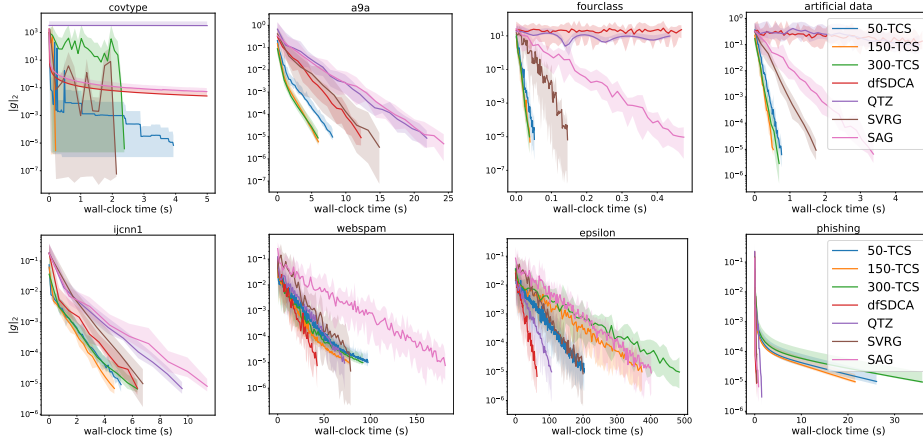
FIG. 1. *Experiments for TCS method applied for generalized linear model.*

smallest variance compared to the first-order methods based on eye-balling the shaded error bars in Fig. 1, especially compared to SVRG. Among the three tested sketch sizes, 150 performed the best except on the *epsilon* dataset.

9. Conclusion and future work. We introduced the `SNR` method, for which we provided strong convergence guarantees. We also developed several promising applications of `SNR` to show that `SNR` is very flexible and tested one of these specialized variants for training GLMs. `SNR` is flexible by the fact that its primitive goal is to solve efficiently nonlinear equations. Since there are many ways to re-write an optimization problem as nonlinear equations, each re-write leads to a distinct method, thus leads to a specific implementation in practice (e.g. `SNM`, TCS methods) when using `SNR`. Besides, the convergence theories presented in Sec. 4 guarantee a large variety of choices for the sketch. This flexibility allows us to discover many applications of `SNR` and their induced consequences, especially providing new global convergence theories. As such, we believe that `SNR` and its global convergence theory will open the way to designing and analyzing a host of new stochastic second order methods. Further venues of investigation include exploring the use of adaptive norms for projections and leveraging efficient sketches (e.g. the fast Johnson-Lindenstrauss sketch [48], sketches with determinantal sampling [42]) to design even faster variants of `SNR` or cover other stochastic second-order methods. Since `SNR` can be seen as `SGD`, it might be possible to design and develop efficient accelerated `SNR` or `SNR` with momentum methods. On the experimental side, it would be interesting to apply our method to the training of deep neural networks.

Appendix A. Other viewpoints of `SNR`.
Beside the connection between `SNR` and `SGD`, in the next section we reformulate `SNR` as a stochastic Gauss-Newton (`GN`) method and a stochastic fixed point method in the subsequent Appendix A.2.

A.1. Stochastic Gauss-Newton method. The `GN` method is a method for solving nonlinear least-squares problems such as

$$(A.1) \qquad \min_{x \in \mathbb{R}^p} \|F(x)\|_{\mathbf{G}}^2,$$

where $\mathbf{G}$ is a symmetric positive-definite matrix. Like the Newton-Raphson method, at each step of the GN method, the function $F(x)$ is replaced by its linearization in (A.1) and then solved to give the next iterate. That is

(A.2) $$x^{k+1} \in \operatorname{argmin}_{x \in \mathbb{R}^p} \|DF(x^k)^\top (x - x^k) + \gamma F(x^k)\|_{\mathbf{G}}^2,$$

where x^{k+1} is the least-norm solution to the above.

Now consider the GN method where the matrix that defines the norm in (A.2) changes at each iteration as is given by $\mathbf{G} \equiv \mathbf{G}^k \stackrel{\text{def}}{=} \mathbb{E}\left[\mathbf{H_S}(x^k)\right]$ and let $d \stackrel{\text{def}}{=} x - x^k$. Since $\mathbf{G}^k$ is an expected matrix, we can write

$$\|DF(x^k)^\top d + \gamma F(x^k)\|_{\mathbb{E}[\mathbf{H_S}(x^k)]}^2 = \mathbb{E}\left[\|DF(x^k)^\top d + \gamma F(x^k)\|_{\mathbf{H_S}(x^k)}^2\right].$$

This suggests a stochastic variant of the GN where we use the unbiased estimate $\mathbf{H_S}(x^k)$ instead of $\mathbf{G}^k$. This stochastic GN method is in fact equivalent to SNR, as we show next.

LEMMA A.1. *Let $x^0 \in \mathbb{R}^p$ and consider the following* Stochastic Gauss-Newton *method*

$$d^k \in \operatorname*{argmin}_{d \in \mathbb{R}^p} \|DF(x^k)^\top d + \gamma F(x^k)\|_{\mathbf{H}_{\mathbf{S}_k}(x^k)}^2$$

(A.3) $$x^{k+1} = x^k + d^k$$

where $\mathbf{S}_k$ is sampled from $\mathcal{D}_{x^k}$ at kth iteration and d^k is the least-norm solution. If Assumption 2.1 holds, then the iterates (A.3) are equal to the iterates of SNR (1.3).

Proof. Differentiating (A.3) in d, we find that d^k is a solution to

$$DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)DF(x^k)^\top d^k = -\gamma DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)F(x^k).$$

Let $\mathbf{A} \stackrel{\text{def}}{=} DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)DF(x^k)^\top$. Taking the least-norm solution to the above gives

$$\begin{aligned}
d^k &= -\gamma \mathbf{A}^\dagger DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)F(x^k) = -\gamma \mathbf{A}^\dagger \mathbf{A} v \\
&= -\gamma \mathbf{A}^\dagger \mathbf{A}\mathbf{A} v = -\gamma \mathbf{A} v \\
&= -\gamma DF(x^k)\mathbf{H}_{\mathbf{S}_k}(x^k)F(x^k),
\end{aligned}$$

where on the first line, we used that Assumption 2.1 shows there exists $v \in \mathbb{R}^p$ such that $F(x^k) = DF(x^k)^\top v$. On the second line, we used that $\mathbf{A} = \mathbf{A}\mathbf{A}$ which is shown in the proof of Corollary 4.5 . Then we used $\mathbf{A}^\dagger \mathbf{A}\mathbf{A} = \mathbf{A}$ which is a property of the pseudoinverse operator that holds for all symmetric matrices. Consequently $x^{k+1} = x^k + d^k$ which is exactly the update given in (1.3). □

Thus our sketched Newton-Raphson method can also be seen as a stochastic Gauss-Newton method. Furthermore, if $\mathbf{S} = \mathbf{I}$, then (A.3) is no longer stochastic and is given by

$$d^k \in \operatorname*{argmin}_{d \in \mathbb{R}^p} \|DF(x^k)^\top d + \gamma F(x^k)\|_{(DF(x^k)^\top DF(x^k))^\dagger}^2$$

(A.4) $$x^{k+1} = x^k + d^k.$$

Thus as a consequence of Lemma A.1, we have that this variant (A.4) of GN is in fact the Newton-Raphson method.

A.2. Stochastic fixed point method. In this section, we reformulate SNR as a stochastic fixed point method. Such interpretation is inspired from [50]'s stochastic fixed point viewpoint. We extend their results from the linear case to the nonlinear case.

Assume that Assumption 2.1 holds and re-consider the sketch-and-project viewpoint (2.2) in Section 2. Note the zeros of the function F

$$\mathcal{L} \overset{\text{def}}{=} \{x \mid F(x) = 0\}$$

and the sketched Newton system based on y

$$\mathcal{L}_{\mathbf{S},y} \overset{\text{def}}{=} \left\{x \in \mathbb{R}^p \mid \mathbf{S}^\top DF(y)^\top (x - y) = -\mathbf{S}^\top F(y)\right\}$$

with $y \in \mathbb{R}^p$ and $\mathbf{S} \sim \mathcal{D}_y$. For a closed convex set $\mathcal{Y} \subseteq \mathbb{R}^d$, let $\Pi_{\mathcal{Y}}$ denote the projection operator onto $\mathcal{Y}$. That is

$$(\text{A.5}) \qquad \Pi_{\mathcal{Y}}(x) \overset{\text{def}}{=} \text{argmin}_{y \in \mathbb{R}^p} \{\|y - x\| : y \in \mathcal{Y}\}.$$

Then, from (2.2) by plugging $\mathcal{Y} = \mathcal{L}_{\mathbf{S},y}$ and $y = x$ into (A.5), we have

$$(\text{A.6}) \qquad \Pi_{\mathcal{L}_{\mathbf{S},x}}(x) = x - DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x).$$

Now we can introduce the fixed point equation as follows

$$(\text{A.7}) \qquad \chi \overset{\text{def}}{=} \left\{x \mid x = \mathbb{E}_{\mathbf{S} \sim \mathcal{D}_x}\left[\Pi_{\mathcal{L}_{\mathbf{S},x}}(x)\right]\right\}.$$

Assumption 2.1 guarantees that finding fixed points of (A.7) is equivalent to the reformulated optimization problem (3.1) with $y = x$, as we show next.

LEMMA A.2. *If Assumption 2.1 holds, then*

$$(\text{A.8}) \qquad \chi = \text{argmin}_{x \in \mathbb{R}^p} \frac{1}{2}\|F(x)\|^2_{\mathbb{E}_{\mathbf{S} \sim \mathcal{D}_x}[\mathbf{H}_{\mathbf{S}}(x)]}.$$

Proof. Let $\chi_{\mathbf{S}} \overset{\text{def}}{=} \left\{x \mid x = \Pi_{\mathcal{L}_{\mathbf{S},x}}(x)\right\}$ with $\mathbf{S} \sim \mathcal{D}_x$. First, we show that

$$(\text{A.9}) \qquad \chi_{\mathbf{S}} = \text{argmin}_{x \in \mathbb{R}^d} \frac{1}{2}\|F(x)\|^2_{\mathbf{H}_{\mathbf{S}}(x)}.$$

In fact,

$$x \in \chi_{\mathbf{S}} \overset{(\text{A.6})}{\iff} DF(x)\mathbf{H}_{\mathbf{S}}(x)F(x) = 0$$

$$\overset{\text{Assumption } 2.1}{\iff} \exists v \in \mathbb{R}^d \text{ s.t. } F(x) = DF(x)^\top v \text{ and } DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top v = 0$$

$$\iff \mathbf{H}_{\mathbf{S}}(x)F(x) = 0 \qquad (\text{as } DF(x)\mathbf{H}_{\mathbf{S}}(x)DF(x)^\top \succeq 0)$$

$$\iff \frac{1}{2}\|F(x)\|^2_{\mathbf{H}_{\mathbf{S}}(x)} = 0.$$

So we induce (A.9). Finally (A.8) follows by taking expectations with respect to $\mathbf{S}$ in (A.9). $\qquad \square$

To solve the fixed point equation (A.7), the natural choice of method is the stochastic fixed point method with relaxation. That is, we pick a relaxation parameter $\gamma > 0$, and consider the following equivalent fixed point problem

$$x = \mathbb{E}_{\mathbf{S} \sim \mathcal{D}_x}\left[\gamma\Pi_{\mathcal{L}_{\mathbf{S},x}}(x) + (1 - \gamma)x\right].$$

Using relaxation is to improve the contraction properties of the map. Then at kth iteration,

$$(A.10) \qquad x^{k+1} = \gamma \Pi_{\mathcal{L}_{\mathbf{S}, x^k}}(x^k) + (1 - \gamma)x^k,$$

where $\mathbf{S} \sim \mathcal{D}_{x^k}$. Consequently, it is straight forward to verify that (A.10) is exactly the update given in (1.3).

Appendix B. Sufficient conditions for reformulation assumption (3.3)**.**

To give sufficient conditions for (3.3) to hold, we need to study the spectra of $\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x)]$. The expected matrix $\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x)]$ has made an appearance in several references [21, 42, 16] in different contexts and with different sketches. We build upon some of these past results and adapt them to our setting.

First note that (3.3) holds if $\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x)]$ is invertible. The invertibility of $\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x)]$ was already studied in detail in the linear setting in Thm. 3 in [25] when $\mathbf{S}$ is sampled from a discrete distribution. Here we can state a sufficient condition of (3.3) for sketching matrices that have a continuous distribution.

LEMMA B.1. *For every $x \in \mathbb{R}^p$, if $\mathbb{E}_{\mathbf{S} \sim \mathcal{D}_x}[\mathbf{S}\mathbf{S}^\top]$ and $DF(x)^\top DF(x)$ are invertible, then $\mathbb{E}_{\mathbf{S} \sim \mathcal{D}_x}[\mathbf{H}_{\mathbf{S}}(x)]$ is invertible.*

Proof. Let $x \in \mathbb{R}^p$ and $\mathbf{S} \sim \mathcal{D}_x$. Let $\mathbf{G} = DF(x)^\top DF(x)$ which is thus symmetric positive definite and $\mathbf{W} = \mathbf{S}^\top$. In this case, since $\mathbf{G}$ is invertible we have that $\mathbf{Ker}(\mathbf{G}) = \{0\} \subset \mathbf{Ker}(\mathbf{W})$ verified, by Lemma 4.4, we have that

$$(B.1) \qquad \mathbf{Ker}\left(\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger\right) = \mathbf{Ker}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right) = \mathbf{Ker}(\mathbf{S}),$$

Consequently, using Lemma 4.4 again with $\mathbf{G} = \left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger$, $\mathbf{W} = \mathbf{S}$ and $\mathbf{Ker}(\mathbf{G}) \subset \mathbf{Ker}(\mathbf{W})$ given by (B.1), we have that
(B.2)
$$\mathbf{Ker}(\mathbf{H}_{\mathbf{S}}(x)) = \mathbf{Ker}\left(\mathbf{S}\left(\mathbf{S}^\top DF(x)^\top DF(x)\mathbf{S}\right)^\dagger \mathbf{S}^\top\right) = \mathbf{Ker}(\mathbf{S}^\top) = \mathbf{Ker}(\mathbf{S}\mathbf{S}^\top).$$

Following the same steps in the proof of Lemma 4.6 right after (4.19), we obtain

$$\mathbf{Ker}(\mathbb{E}[\mathbf{H}_{\mathbf{S}}(x)]) = \bigcap_{\mathbf{S} \sim \mathcal{D}_x} \mathbf{Ker}(\mathbf{H}_{\mathbf{S}}(x)) \overset{(B.2)}{=} \bigcap_{\mathbf{S} \sim \mathcal{D}_x} \mathbf{Ker}(\mathbf{S}\mathbf{S}^\top) = \mathbf{Ker}(\mathbb{E}[\mathbf{S}\mathbf{S}^\top]) = \{0\},$$

where the last equality follows as $\mathbb{E}[\mathbf{S}\mathbf{S}^\top]$ is invertible, which concludes the proof. $\square$

The invertibility of $\mathbb{E}[\mathbf{S}\mathbf{S}^\top]$ states that the sketching matrices need to "span every dimension of the space" in expectation. This is the case for Gaussian and subsampling sketches which are shown in Lemma 4.8. This is also the case for our applications SNM and TCS which are shown in the proofs of Lemma 7.4 and Lemma 8.3, respectively.

As for the invertibility of $DF(x)^\top DF(x) \in \mathbb{R}^{m \times m}$, this imposes that $DF(x)$ has full-column rank for all $x \in \mathbb{R}^p$, thus $m \leq p$. This excludes the regime of solving $F(x) = 0$ with $m > p$. However, our applications SNM and TCS also satisfy this condition which are again shown in the proofs of Lemma 7.2 and Lemma 8.3, respectively.

Consequently, by Lemma B.1, we have that SNM and TCS satisfy (3.3).

Appendix C. The monotone convergence theory of NR with stepsize $\gamma < 1$.

The MCT of the NR in both [47] and [17] need to have the stepsize $\gamma = 1$. If $\gamma < 1$ which is the case in our convergence Theorem 4.3 and Corollary 4.5, the iterates $\{x^k\}_{k \geq 1}$ under the set of assumptions (I) proposed in Theorem 5.2 are no longer guaranteed to be component wise monotonically decreasing. Here we investigate alternatives. In particular, we consider the case in 1-dimension for function $F = \phi : \mathbb{R} \to \mathbb{R}$.

LEMMA C.1. *Let x^k be the iterate of the NR method with stepsize $\gamma < 1$ for solving $\phi(x) = 0$, that is*

(C.1)
$$x^{k+1} = x^k - \gamma \frac{\phi(x^k)}{\phi'(x^k)}.$$

If ϕ satisfies the set of assumptions (I) proposed in Theorem 5.2, then
  (a) *The iterates of the ordinary NR method (5.11) are necessarily monotonically decreasing.*
  (b) *The iterates of the NR method (C.1) with $\gamma < 1$ are not necessarily monotonically decreasing.*
  (c) *Assumption (5.7) holds; for $\frac{1}{2} \leq \gamma < 1$, there exists $k' \geq 0$ such that for all $k \neq k'$, there exists a unique x^* that satisfies Assumption 1.1 and the iterates x^k and the optimum x^* satisfy (5.8).*
  (d) *The iterates x^k following the NR method (C.1) with $\frac{1}{2} \leq \gamma < 1$ converge sublinearly to a zero of ϕ.*

Remark of (a). Even though this result is known and generalized in d-dimension in [47] and [17], we stress it here to highlight the impact of the stepsize γ in the NR method and leverage the analysis of (a) in the special 1-dimensional case to prove (b).

Proof. If ϕ satisfies (I), then ϕ is convex and $\phi'^{-1} > 0$, which implies $\phi'' \geq 0$ and $\phi' > 0$. From $\phi' > 0$, we obtain that ϕ is strictly increasing. Besides, from (I), $\exists \, x, y \in \mathbb{R}$ such that $\phi(x) \leq 0 \leq \phi(y)$. This with the strictly increase of ϕ induces that $\exists! \, x^*$ such that $\phi(x^*) = 0$, i.e. x^* satisfies Assumption 1.1. So $\forall x < x^*$, $\phi(x) < 0$ and $\forall x > x^*$, $\phi(x) > 0$. Now consider the following two functions

$$u(x) \overset{\text{def}}{=} x - \frac{\phi(x)}{\phi'(x)} \qquad \text{and} \qquad U(x) \overset{\text{def}}{=} x - \gamma \frac{\phi(x)}{\phi'(x)}$$

which are exactly the updates of the ordinary NR (5.11) and the NR (C.1) with a stepsize $\gamma \in (0, 1)$, respectively. We first analyze the behaviour of the function u and show (a), which can be formulated as $x^* \leq x^{k+1} \leq x^k$ for all $k \geq 1$. The derivative of u is

$$u'(x) = \frac{\phi(x)\phi''(x)}{\phi'(x)^2}.$$

By the sign of functions ϕ, ϕ' and ϕ'', we know that if $x > x^*$, then $u'(x) \geq 0$ and if $x < x^*$, then $u'(x) \leq 0$. This implies that the function u is increasing in $[x^*, +\infty[$ and decreasing in $] -\infty, x^*]$. Overall, we have

(C.2a) $\min_{x \in \mathbb{R}} u(x) = u(x^*) \overset{\phi(x^*)=0}{=} x^*,$

(C.2b) $u(x) < x$ and u increasing,   when $x > x^*$,

(C.2c) $u(x) > x$ and u decreasing,   when $x < x^*$.

Consequently, $x^* \leq x^{k+1}$ is obtained by $x^* \overset{(C.2a)}{=} \min u(x) \leq u(x^k) = x^{k+1}$. As for the inequality $x^{k+1} \leq x^k$, $x^* \overset{(C.2a)}{=} \min u(x) \leq u(x^{k-1}) = x^k$ for $k \geq 1$ and $x^{k+1} = u(x^k) \overset{(C.2b)}{\leq} x^k$ as $x^k \geq x^*$.

To show (b), we analyze the behavior of the function U. Consider its derivative

$$U'(x) = (1 - \gamma) + \gamma \frac{\phi(x)\phi''(x)}{\phi'(x)^2}.$$

By the sign of functions ϕ, ϕ' and ϕ'', if $x > x^*$, $U'(x) > 0$. However, $U'(x^*) \overset{\phi(x^*)=0}{=} 1 - \gamma > 0$, which implies $\min U(x) < x^*$. Here we include the case where $\min U(x) = -\infty$. Also by the sign of functions ϕ and ϕ' and $\gamma < 1$, when $x > x^*$, we have $u(x) < U(x)$ and $u(x) > U(x)$ for $x < x^*$. In summary, we have

(C.3a)

(C.3b)

(C.3c)
$$\begin{cases} \min_{x \in \mathbb{R}} U(x) < U(x^*) \overset{\phi(x^*)=0}{=} u(x^*) \overset{(C.2a)}{=} x^*, \\ u(x) < U(x) < x \text{ and } U \text{ increasing}, \quad \text{when } x > x^*, \\ u(x) > U(x) > x \text{ when } x < x^*. \end{cases}$$

In NR with stepsize $\gamma < 1$, consider $x^0 \in \mathbb{R}$. We discuss different cases based on the comparison between x^0 and x^*.

If $x^0 \geq x^*$ named as case **(i)**, by induction, we get $x^* \overset{(C.3a)}{=} U(x^*) \overset{(C.3b)}{\leq} = U(x^k) = x^{k+1} \overset{(C.3b)}{\leq} x^k$ for all $k \geq 0$. In this case, the iterates decrease monotonically.

If $x^0 < x^*$, there are two cases, named as case **(ii)**, for all $k \in \mathbb{N}$, $U(x^k) \leq x^*$, and case **(iii)**, $\exists k' \in \mathbb{N}$, $U(x^{k'}) > x^*$.

If **(ii)** holds, we have that the iterates increase monotonically. Indeed, by **(ii)** and by induction, we get $x^k \overset{(C.3c)}{\leq} U(x^k) = x^{k+1} \overset{\textbf{(ii)}}{\leq} x^*$ for all $k \in \mathbb{N}$.

Otherwise, we are in case **(iii)**. Let k' be the smallest index that $U(x^{k'}) > x^*$. Then we conclude that the iterates $\{x^k\}_{k \geq 0}$ increase monotonically when $k \leq k'$ and $\{x^k\}_{k \geq k'+1}$ decrease monotonically. In fact, by the definition of k', we know that for $k \in [\![0, k' - 1]\!]$, $U(k) \leq x^*$. Then by induction as in case **(ii)** but for $k \leq k'$, we get $\{x^k\}_{k \geq 0}$ increase monotonically when $k \leq k'$. When $k \geq k' + 1$, by induction as in case **(i)** but for $U(x^{k'}) = x^{k'+1} > x^*$, we get $\{x^k\}_{k \geq k'+1}$ decrease monotonically. We thus observe (b).

Statement (c) follows from the proof of Theorem 5.2 in 1-dimension in taking account the stepsize $\gamma < 1$. Then (5.7) holds and (5.8) becomes

(C.4)
$$f_k(x^k) + \langle \nabla f_k(x^k), x^* - x^k \rangle$$
$$= \underbrace{\frac{1 - 2\gamma}{2\gamma^2}(x^{k+1} - x^k)^2}_{\leq 0 \text{ as } \frac{1}{2} \leq \gamma \text{ in (c)}} + \frac{1}{\gamma} \underbrace{(x^k - x^{k+1})(x^* - x^{k+1})}_{\overset{\text{def}}{=} (*)}$$

in considering

$$f_k(x^k) = \frac{1}{2} \left(\frac{\phi(x^k)}{\phi'(x^k)} \right)^2 \overset{(C.1)}{=} \frac{1}{2\gamma^2}(x^{k+1} - x^k)^2,$$

$$\nabla f_k(x^k) = \frac{\phi(x^k)}{\phi'(x^k)} \overset{(C.1)}{=} \frac{1}{\gamma} \left(x^k - x^{k+1} \right),$$

with $\phi = F$ and $\phi' = DF$. To get (5.8) hold, from (C.4), it suffices to prove $(*) \leq 0$.

By the analysis of (b), we know: in case **(i)**, $(*) \leq 0$ for all $k \geq 0$ as $x^* \leq x^{k+1} \leq x^k$; in case **(ii)**, $(*) \leq 0$ for all $k \geq 0$ as $x^* \geq x^{k+1} \geq x^k$; finally in case **(iii)**, for $k \neq k'$, $(*) \leq 0$ as $x^* \geq x^{k+1} \geq x^k$ for $k \leq k' - 1$ and $x^* \leq x^{k+1} \leq x^k$ for $k \geq k' + 1$. So in all cases, $(*) \leq 0$ for all k or for $k \neq k'$. We thus obtain (c).

It remains to show (d), which is simply obtained by (c) and Corollary 4.5, as (5.8) holds for all iterates x^k except for just one iterate $x^{k'}$ potentially.   □

The monotone convergence theory is based on assumptions (I) with stepsize $\gamma = 1$. Under the same assumptions with $\gamma < 1$, such theory may not hold. Indeed, following the analysis in Lemma C.1 in 1-dimension case, by (C.3c) we do not have the monotone property for the function U when $x < x^*$. That is the reason why (b) happens but not (a) in Lemma C.1. In d-dimension case, without such monotone property for the function U, $\{x^k\}$ is not guaranteed to be monotone, which is the main clue in their theory's proof. However, with stepsize $\gamma < 1$, assumptions (I) can still imply our Assumptions 1.1, (5.7) and (5.8) under constraint $\frac{1}{2} \leq \gamma < 1$ in 1-dimension case. In addition, though our theory does not either require any constraint for stepsize $\gamma < 1$ or guarantee that the NR method is monotonic in terms of the iterates component wisely, we still guarantee the sublinear global convergence. We thus conclude that Assumptions 1.1, (5.7) and (5.8) are strictly weaker than the assumptions used in the monotone convergence theory in [47] and [17], albeit for different step sizes.

Appendix D. Stochastic Newton method with relaxation.

Consider the function $P(\cdot)$ defined in (7.1). By the analysis of Lemma 7.1, we can even develop a variant of SNM in the case stepsize $\gamma < 1$ and we call the method *Stochastic Newton method with relaxation*. The updates are the following

(D.1)
$$w^{k+1} = \gamma \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \right)^{-1} \left(\frac{1}{n} \sum_{i=1}^{n} \nabla^2 \phi_i(\alpha_i^k) \alpha_i^k - \frac{1}{n} \sum_{i=1}^{n} \nabla \phi_i(\alpha_i^k) \right) + (1 - \gamma) w^k,$$

(D.2)
$$\alpha_i^{k+1} = \begin{cases} w^{k+1} - (1-\gamma)(w^k - \alpha_i^k) & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases},$$

In the rest of Section D, we use the shorthand $x \stackrel{\text{def}}{=} (w; \alpha_1; \cdots; \alpha_n) \in \mathbb{R}^{(n+1)d}$ and $x^k \stackrel{\text{def}}{=} (w^k; \alpha_1^k; \cdots; \alpha_n^k)$ the iterates of SNR in Lemma 7.1 with stepsize $\gamma < 1$ at the kth iteration.

LEMMA D.1. *At each iteration k, the updates of SNR x^k are equal to the updates* (D.1), (D.2) *of SNM with relaxation.*

Proof. Following the proof of Lemma 7.1 and taking account the stepsize γ, by (7.7) and (2.2), the updates of SNR x^{k+1} at $(k+1)$th iteration are given by

(D.3)
$$x^{k+1} = \arg\min \|w - w^k\|^2 + \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2 \text{ s.t. } \mathbf{S}^\top DF(x^k)^\top (x - x^k) = -\gamma \mathbf{S}^\top F(x^k),$$

where the sketching matrix $\mathbf{S} \sim \mathcal{D}_{x^k}$ is defined in (7.5). Similar to (7.8), (D.3) can

be re-written as

(D.4)

$$x^{k+1} = \operatorname*{argmin} \|w - w^k\|^2 + \sum_{i=1}^{n} \|\alpha_i - \alpha_i^k\|^2$$

$$\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \nabla^2\phi_i(\alpha_i^k)(w - w^k) = -\gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k) + \frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)(w^k - \alpha_i^k)\right),$$

$$w - \alpha_i = (1-\gamma)(w^k - \alpha_i^k), \quad \text{for } i \in B_n.$$

Similarly, note that if $i \notin B_n$, then $\alpha_i^{k+1} = \alpha_i^k$, since there is no constraint on the variable α_i in this case. Then by the invertibility of $\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)$, we have the unique solution of (D.4), which is

$$w^{k+1} = \gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k)\right) + (1-\gamma)w^k,$$

$$\alpha_i^{k+1} = w^{k+1} - (1-\gamma)(w^k - \alpha_i^k), \quad \text{for } i \in B_n.$$

Overall, we have

$$w^{k+1} = \gamma\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\right)^{-1}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla^2\phi_i(\alpha_i^k)\alpha_i^k - \frac{1}{n}\sum_{i=1}^{n}\nabla\phi_i(\alpha_i^k)\right) + (1-\gamma)w^k,$$

$$\alpha_i^{k+1} = \begin{cases} w^{k+1} - (1-\gamma)(w^k - \alpha_i^k) & \text{if } i \in B_n \\ \alpha_i^k & \text{if } i \notin B_n \end{cases},$$

which is exactly the updates (D.1) and (D.2) in SNM with relaxation. $\square$

Notice that both the original SNM and SNM with relaxation have the same complexity. Consequently, Theorem 4.3 allows us to develop the following global convergence theory of SNM with relaxation γ.

COROLLARY D.2. *Consider the iterate* $x^k = \left(w^k; \alpha_1^k; \cdots; \alpha_n^k\right)$ *given by* (D.1) *and* (D.2). *Note* $x^* \stackrel{def}{=} (w^*; w^*; \cdots; w^*) \in \mathbb{R}^{(n+1)d}$ *where* w^* *is the stationary point of* $\nabla P(w)$ *that satisfies*

(D.5) $$f_{x^k}(x^*) \geq f_{x^k}(x^k) + \left\langle \nabla f_{x^k}(x^k), x^* - x^k \right\rangle, \quad \text{for all } k \in \mathbb{N},$$

then

$$\min_{t=0,\ldots,k-1} \mathbb{E}\left[f_{x^t}(x^t)\right] \quad \leq \quad \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[f_{x^t}(x^t)\right] \quad \leq \quad \frac{1}{k}\frac{\|x^0 - x^*\|^2}{2\gamma(1-\gamma)}.$$

Proof. Equation (D.5) with the assumption that w^* is the stationary point of $\nabla P(w)$ implies that Assumption 1.1 and 4.2 hold. Then we conclude the proof by Theorem 4.3. $\square$

Appendix E. Extension of SNR and Randomized Subspace Newton.

In the SNR method in (1.3), we only consider a projection under the standard Euclidean norm. If we allow SNR and (1.3) for a changing norm that depends on the iterates, we find that the Randomized Subspace Newton [21] (RSN) method is in fact a special case of SNR under this extension.

The changing norm projection of SNR is that, at kth iteration of SNR, instead of applying (1.3), we can apply the following update

(E.1) $x^{k+1} = x^k - \gamma \mathbf{W}_k^{-1} DF(x^k) \mathbf{S}_k \left(\mathbf{S}_k^\top DF(x^k)^\top \mathbf{W}_k^{-1} DF(x^k) \mathbf{S}_k \right)^\dagger \mathbf{S}_k^\top F(x^k)$

where $\mathbf{W}_k \equiv \mathbf{W}(x^k)$ with $\mathbf{W}(x^k)$ a certain symmetric positive-definite matrix associated with the kth iterate $x^k \in \mathbb{R}^p$.

The interpretation of using the matrix $\mathbf{W}(x^k)$ is that, assuming Assumption 2.1 holds, then instead of considering (2.2), we apply the following updates

$$x^{k+1} = \mathrm{argmin}_{x \in \mathbb{R}^p} \|x - x^k\|_{\mathbf{W}_k}^2$$
(E.2) $$\text{s. t.}\quad \mathbf{S}_k^\top DF(x^k)^\top (x - x^k) = -\gamma \mathbf{S}_k^\top F(x^k),$$

using the projection $\|\cdot\|_{\mathbf{W}_k}$ which changes at each iteration. One can verify easily that (E.2) is equivalent to (E.1) under Assumption 2.1, even though this assumption is not necessary and the update (E.1) is still available.

Now we can show that RSN is a special case of SNR with a changing norm projection. The RSN method [21] is a stochastic second order method that takes a Newton-type step at each iteration to solve the minimization problem

$$\min_{x \in \mathbb{R}^p} P(x)$$

where $P : \mathbb{R}^p \to \mathbb{R}$ is a twice differentiable and convex function. In brevity, the updates in RSN at the kth iteration are given by

(E.3) $$x^{k+1} = x^k - \frac{1}{\hat{L}} \mathbf{S}_k \left(\mathbf{S}_k^\top \nabla^2 P(x^k) \mathbf{S}_k \right)^\dagger \mathbf{S}_k^\top \nabla P(x^k)$$

where $\mathbf{S}_k$ is sampled i.i.d from a fixed distribution $\mathcal{D}$ and $\hat{L} > 0$ is the *relative smoothness* constant [21].

Since $P(x)$ is convex, it suffices to find a stationary point x such that $\nabla P(x) = 0$. We can recover the exact same update (E.3) by applying SNR to solve $\nabla P(x) = 0$ with an adaptive changing norm. That is, let $F(x) = \nabla P(x)$ and $DF(x) = \nabla^2 P(x)$. At the kth iteration, let $\mathbf{W}_k = \nabla^2 P(x^k)$. Then (E.1) is exactly the RSN update (E.3) with $\gamma = \frac{1}{\hat{L}}$.

Appendix F. Explicit formulation of the TCS method.

Here we provide details about how TCS method presented in Section 8 is obtained from the general SNR method 1.

Consider the SNR method (1.3) applied for the nonlinear equations $F(\alpha; w) = 0$ with F defined in (8.4) and the Jacobian of $F(\alpha; w)$ in (8.5).

At kth iteration $(\alpha^k, w^k) \in \mathbb{R}^n \times \mathbb{R}^d$, let

$$\begin{bmatrix} \alpha^{k+1} \\ w^{k+1} \end{bmatrix} \overset{\text{def}}{=} \begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} + \gamma \cdot \begin{bmatrix} \Delta \alpha^k \\ \Delta w^k \end{bmatrix}$$

and $\mathbf{S}_k \in \mathbb{R}^{(d+n) \times (\tau_d + \tau_n)}$ the random sketching matrix. By (1.3), we obtain the closed form update

$$\begin{bmatrix} \Delta \alpha^k \\ \Delta w^k \end{bmatrix} = -DF(\alpha^k; w^k) \mathbf{S}_k \left(\mathbf{S}_k^\top DF(\alpha^k; w^k)^\top DF(\alpha^k; w^k) \mathbf{S}_k \right)^\dagger \mathbf{S}_k^\top$$
(F.1) $$\begin{bmatrix} \frac{1}{\lambda n} \mathbf{A} \alpha^k - w^k \\ \alpha^k + \Phi(w^k) \end{bmatrix}.$$

As for the tossing-coin-sketch, consider a Bernoulli parameter b with $b \in (0, 1)$. There is a probability $1 - b$ that the random sketching matrix has the type $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d \in \mathbb{R}^{d \times \tau_d}$, a (d, τ_d)–block sketch, and a probability b that the random sketching matrix has the type $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n \in \mathbb{R}^{n \times \tau_n}$, a (n, τ_n)–block sketch. So $\mathbf{S}_d = \mathbf{I}_{B_d}$ and $\mathbf{S}_n = \mathbf{I}_{B_n}$.

Let $\mathbf{A}_{B_d,:} \equiv \mathbf{I}_{B_d}^\top \mathbf{A} \in \mathbb{R}^{\tau_d \times n}$ denote a row subsampling of $\mathbf{A}$ and $\mathbf{A}_{:,B_n} \equiv \mathbf{A}\mathbf{I}_{B_n} \in \mathbb{R}^{d \times \tau_n}$ denote a column subsampling of $\mathbf{A}$. Let $\nabla\Phi_{B_n}^k \equiv \nabla\Phi^k \mathbf{I}_{B_n} \in \mathbb{R}^{d \times \tau_n}$ denote a column subsampling of $\nabla\Phi^k$ with $\nabla\Phi^k \equiv \nabla\Phi(w^k)$ and $\Phi^k \equiv \Phi(w^k)$. We also use the shorthands $v_{B_n} \equiv \mathbf{I}_{B_n}^\top v \in \mathbb{R}^{\tau_n}$ with $v \in \mathbb{R}^n$ and $v_{B_d} \equiv \mathbf{I}_{B_d}^\top v \in \mathbb{R}^{\tau_d}$ with $v \in \mathbb{R}^d$.

If $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$, the update (F.1) applied for the function (8.4) and its Jacobian (8.5) becomes

$$
\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = - \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}^\top \mathbf{S}_d \\ -\mathbf{S}_d \end{bmatrix} \left(\mathbf{S}_d^\top \left(\frac{1}{\lambda^2 n^2}\mathbf{A}\mathbf{A}^\top + \mathbf{I}_d \right) \mathbf{S}_d \right)^\dagger \mathbf{S}_d^\top \left(\frac{1}{\lambda n}\mathbf{A}\alpha^k - w^k \right)
$$

$$
\text{(F.2)} \qquad = - \begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}_{B_d,:}^\top \\ -\mathbf{I}_{B_d} \end{bmatrix} \left(\frac{\mathbf{A}_{B_d,:}\mathbf{A}_{B_d,:}^\top}{\lambda^2 n^2} + \mathbf{I}_{\tau_d} \right)^\dagger \left(\frac{\mathbf{A}_{B_d,:}\alpha^k}{\lambda n} - w_{B_d}^k \right).
$$

Similarly, if $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$, the update (F.1) becomes

$$
\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = - \begin{bmatrix} \mathbf{S}_n \\ \nabla\Phi^k \mathbf{S}_n \end{bmatrix} \left(\mathbf{S}_n^\top \left([\nabla\Phi^k]^\top \nabla\Phi^k + \mathbf{I}_n \right) \mathbf{S}_n \right)^\dagger \mathbf{S}_n^\top \left(\alpha^k + \Phi^k \right)
$$

$$
\text{(F.3)} \qquad = - \begin{bmatrix} \mathbf{I}_{B_n} \\ \nabla\Phi_{B_n}^k \end{bmatrix} \left([\nabla\Phi_{B_n}^k]^\top \nabla\Phi_{B_n}^k + \mathbf{I}_{\tau_n} \right)^\dagger \left(\alpha_{B_n}^k + \Phi_{B_n}^k \right).
$$

Then we update $\begin{bmatrix} \alpha^{k+1} \\ w^{k+1} \end{bmatrix} = \begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} + \gamma \cdot \begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix}$.

See Algorithm 2 the pseudocode for the updates (F.2) and (F.3).

Appendix G. Pseudo code and implementation details for GLMs.
We also provide a more efficient and detailed implementation of Algorithm 2 in Algorithm 4 in this section.

It is beneficial to first understand Algorithm 2 in the simple setting where $\tau_d = \tau_n = 1$. We refer to this setting as the Kaczmarz–TCS method.

G.1. Kaczmarz–TCS. Let $f_j \in \mathbb{R}^d$ ($e_i \in \mathbb{R}^n$) be the jth (the ith) unit coordinate vector in $\mathbb{R}^d$ (in $\mathbb{R}^n$, respectively). For $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d = f_j$, from (F.2) we get

$$
\text{(G.1)} \qquad \begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = - \frac{\frac{1}{\lambda n}\sum_{l=1}^n a_{lj}\alpha_l^k - w_j^k}{\frac{1}{\lambda^2 n^2}\sum_{l=1}^n a_{lj}^2 + 1} \begin{bmatrix} \frac{1}{\lambda n} \begin{pmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{pmatrix} \\ -f_j \end{bmatrix}.
$$

Algorithm 2 τ–TCS

1: Choose $(\alpha^0; w^0) \in \mathbb{R}^{n+d}$, $\gamma > 0$, $\tau_d, \tau_n \in \mathbb{N}$ and $b \in (0,1)$.
2: Let $v \sim B(b)$ be a Bernoulli random variable (the coin toss)
3: **for** $k = 0, 1, \cdots$ **do**
4:     Sample $v \in \{0, 1\}$
5:     **if** $v = 0$ **then**
6:         Sample $B_d \subset \{1, \ldots, d\}$ with $|B_d| = \tau_d$ uniformly.
7:         Compute $y_d \in \mathbb{R}^{\tau_d}$ the least norm solution to
8:         $\left(\frac{\mathbf{A}_{B_d,:} \mathbf{A}_{B_d,:}^\top}{\lambda^2 n^2} + \mathbf{I}_{\tau_d} \right) y_d = \frac{\mathbf{A}_{B_d,:} \alpha^k}{\lambda n} - w_{B_d}^k$
9:         Compute the updates
10:        $\begin{bmatrix} \Delta \alpha^k \\ \Delta w^k \end{bmatrix} = - \begin{bmatrix} \frac{1}{\lambda n} \mathbf{A}_{B_d,:}^\top \\ -\mathbf{I}_{B_d} \end{bmatrix} y_d$
11:    **else**
12:        Sample $B_n \subset \{1, \ldots, n\}$ with $|B_n| = \tau_n$ uniformly.
13:        Compute $y_n \in \mathbb{R}^{\tau_n}$ the least norm solution to
14:        $\left([\nabla \Phi_{B_n}^k]^\top \nabla \Phi_{B_n}^k + \mathbf{I}_{\tau_n} \right) y_n = \alpha_{B_n}^k + \Phi_{B_n}^k$
15:        Compute the updates
16:        $\begin{bmatrix} \Delta \alpha^k \\ \Delta w^k \end{bmatrix} = - \begin{bmatrix} \mathbf{I}_{B_n} \\ \nabla \Phi_{B_n}^k \end{bmatrix} y_n$
17:    $w^{k+1} = w^k + \gamma \Delta w^k$
18:    $\alpha^{k+1} = \alpha^k + \gamma \Delta \alpha^k$
19: **return:** last iterate α^k, w^k

For $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = e_i$, from (F.3) we get

$$(G.2) \qquad \begin{bmatrix} \Delta \alpha^k \\ \Delta w^k \end{bmatrix} = - \frac{\alpha_i^k + \phi_i'(a_i^\top w^k)}{\|a_i\|_2^2 \phi_i''(a_i^\top w^k)^2 + 1} \begin{bmatrix} e_i \\ \phi_i''(a_i^\top w^k) a_i \end{bmatrix}.$$

See Algorithm 3 an efficient implementation of Algorithm 2 in a single row sampling case. Notice that we introduce an auxiliary variable $\overline{\alpha}^k$ to update the term $\frac{1}{\lambda n} \sum_{l=1}^n a_{lj} \alpha_l^k$ for $j = 1, \cdots, d$ in (G.1) and we store a $d \times d$ matrix `cov` which can be seen as the covariance matrix of the dataset $\mathbf{A}$ to update the term $\frac{1}{\lambda^2 n^2} \sum_{l=1}^n a_{lj}^2$ for $j = 1, \cdots, d$ in (G.1) (see Algorithm 3 Line 12). We also store a vector `sample` $\in \mathbb{R}^n$ to update the term $\|a_i\|_2^2$ for $i = 1, \cdots, n$ in (G.2) (see Algorithm 3 Line 21).

Cost per iteration analysis of Algorithm 3. From Algorithm 3, the cost of computing (G.1) is $\mathcal{O}(n)$ with n coordinates' updates of the auxiliary variable α (see Algorithm 3 Line 15). This is affordable as the cost of each coordinate's update is 1. Besides $\mathbf{A}$ is often sparse. The update in this case can be much cheaper than n. Besides, the cost of computing (G.2) is $\mathcal{O}(d)$. If we choose the Bernoulli parameter $b = n/(n+d)$ which selects one row of F uniformly, the total cost of the updates TCS in expectation with respect to the Bernoulli distribution will be

Cost(update TCS) $= \mathcal{O}(n) * (1 - b) + \mathcal{O}(d) * b = \mathcal{O}(nd/(n + d)) = \mathcal{O}(\min(n, d))$.

So the TCS method can have the same cost per iteration as the stochastic first-order methods in the case $d < n$, such as SVRG [28], SAG [53], dfSDCA [54] and Quartz [49].

G.2. τ–Block TCS. Here we provide Algorithm 4 which is a detailed implementation of Algorithm 2 in a more efficient way. Similar to Algorithm 3 but with

---

Algorithm 3 Kaczmarz-TCS

1: **parameters:** $\mathcal{D}$ = distribution over random matrices
2: **store in memory:**
3:     `sample:` $(\|a_i\|_2^2)_{1 \le i \le n} \in \mathbb{R}^n$
4:     `cov:` $\frac{1}{\lambda^2 n^2} \mathbf{A}\mathbf{A}^\top \in \mathbb{R}^{d \times d}$
5: **initialization:**
6:     Choose $(\alpha^0, w^0) \in \mathbb{R}^n \times \mathbb{R}^d$ and a step size $\gamma \in \mathbb{R}^{++}$
7:     Set $\overline{\alpha}^0 = \frac{1}{\lambda n} \mathbf{A}\alpha^0$
8: **for** $k = 0, 1, \cdots$ **do**
9:     sample a fresh tossing-coin sketching matrix: $\mathbf{S}_k \sim \mathcal{D}$
10:     **if** $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d = f_j$ **then**
11:         **update** (G.1):   ▷ Sketch a linear system based on the first d rows of the Jacobian
12: $$\Delta w_j^k = \frac{\overline{\alpha}_j^k - w_j^k}{\text{cov}[j,\ j]+1}$$
13: $$\Delta\alpha^k = -\Delta w_j^k \cdot \frac{1}{\lambda n} \begin{bmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{bmatrix}$$
14:         $w_j^{k+1} = w_j^k + \gamma \cdot \Delta w_j^k$ ▷ jth coordinate's update of the variable w^k
15:         $\alpha^{k+1} = \alpha^k + \gamma \cdot \Delta\alpha^k$ ▷ full vector's update of the auxiliary variable α^k
16:         $\overline{\alpha}^{k+1} = \overline{\alpha}^k - \gamma \cdot \Delta w_j^k \cdot \text{cov}[:,\ j]$
17:     **else**
18:         $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = e_i$
19:         **update** (G.2): ▷ Sketch a system based on the last n rows of the Jacobian
20:         `temp` $= a_i^\top w^k$ ▷ temporal scalar
21:         $\Delta\alpha_i^k = -\frac{\alpha_i^k + \phi_i'(\text{temp})}{\text{sample}[i] \cdot \phi_i''(\text{temp})^2 + 1}$
22:         $\Delta w^k = \Delta\alpha_i^k \cdot \phi_i''(\text{temp}) \cdot a_i$
23:         $w^{k+1} = w^k + \gamma \cdot \Delta w^k$ ▷ full vector's update of the variable w^k
24:         $\alpha_i^{k+1} = \alpha_i^k + \gamma \cdot \Delta\alpha_i^k$ ▷ ith coordinate's update of the auxiliary variable α^k
25:         $\overline{\alpha}^{k+1} = \overline{\alpha}^k + \gamma \cdot \Delta\alpha_i^k \cdot \frac{1}{\lambda n} a_i$
26: **return:** last iterate α^k, w^k

---

sketch sizes τ_d and τ_n, we also store a $d \times d$ matrix `cov`, but not a vector `sample`. We refer to Algorithm 4 as the τ-block TCS method.

From Algorithm 4, the cost of solving the $\tau_d \times \tau_d$ system (see Algorithm 4 Line 12) is $\mathcal{O}(\tau_d^3)$ for a direct solver and the cost of updating α and $\overline{\alpha}$ (see Algorithm 4 Line 15 and Line 16) are $\mathcal{O}(\tau_d n)$ and $\mathcal{O}(\tau_d d)$ respectively. Overall, this implies that the cost of executing the sketching of the first d rows is

$$(G.3) \qquad\qquad c_d \stackrel{\text{def}}{=} \mathcal{O}(\max(\tau_d^3, \tau_d n, \tau_d d)).$$

Similarly, the dominant cost of executing the last n rows sketch comes from forming the $\tau_n \times \tau_n$ linear system or solving such system (see Line 25), which gives

$$(G.4) \qquad\qquad c_n \stackrel{\text{def}}{=} \mathcal{O}(\max(\tau_n^3, \tau_n^2 d)).$$

Algorithm 4 τ–Block TCS

1: **parameters:** $\mathcal{D}$ = distribution over random matrices
2: **store in memory:**
3:     $\texttt{cov}$: $\frac{1}{\lambda^2 n^2}\mathbf{A}\mathbf{A}^\top \in \mathbb{R}^{d\times d}$
4: **initialization:**
5:     Choose $(\alpha^0, w^0) \in \mathbb{R}^n \times \mathbb{R}^d$ and a step size $\gamma \in \mathbb{R}^{++}$
6:     Set $\overline{\alpha}^0 = \frac{1}{\lambda n}\mathbf{A}\alpha^0$
7: **for** $k = 0, 1, \cdots$ **do**
8:     sample a fresh tossing-coin sketching matrix: $\mathbf{S}_k \sim \mathcal{D}$
9:     **if** $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d = \mathbf{I}_{B_d}$ and $|B_d| = \tau_d$ **then**
10:         **Update** (F.2):   ▷ Sketch a linear system based on the first d rows of the Jacobian
11:         Compute $y_d \in \mathbb{R}^{\tau_d}$ the least norm solution to the $\tau_d \times \tau_d$ linear system
12:         $(\texttt{cov}[B_d,\ B_d] + \mathbf{I}_{\tau_d})\, y_d = -\left(\overline{\alpha}^k_{B_d} - w^k_{B_d}\right)$
13:         Compute the updates
14:         $w^{k+1}_{B_d} = w^k_{B_d} - \gamma \cdot y_d$ ▷ τ_d coordinates' update of the variable w^k
15:         $\alpha^{k+1} = \alpha^k + \gamma \cdot \frac{1}{\lambda n}\mathbf{A}^\top_{B_d,:}y_d$ ▷ full vector's update of the auxiliary variable α^k
16:         $\overline{\alpha}^{k+1} = \overline{\alpha}^k + \gamma \cdot \texttt{cov}[:,\ B_d]y_d$
17:     **else**
18:         $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = \mathbf{I}_{B_n}$ and $|B_n| = \tau_n$
19:         **Update** (F.3):   ▷ Sketch a system based on the last n rows of the Jacobian
20:         $\texttt{temp} = \mathbf{A}^\top_{:,B_n}w^k \in \mathbb{R}^{\tau_n}$ ▷ Temporal vector
21:         $\mathbf{D}^k_{B_n} = \mathbf{Diag}\left(\phi''_{B_n}(\texttt{temp})\right) \in \mathbb{R}^{\tau_n \times \tau_n}$ ▷ Compute $\phi''_i(a_i^\top w^k)$ element-wise $\forall i \in B_n$
22:         $\nabla\Phi^k_{B_n} = \mathbf{A}_{:,B_n}\mathbf{D}^k_{B_n} \in \mathbb{R}^{d\times\tau_n}$
23:         $\Phi^k_{B_n} = \phi'_{B_n}(\texttt{temp})$ ▷ Compute $\phi'_i(a_i^\top w^k)$ element-wise $\forall i \in B_n$
24:         Compute $y_n \in \mathbb{R}^{\tau_n}$ the least norm solution to the $\tau_n \times \tau_n$ linear system
25:         $\left([\nabla\Phi^k_{B_n}]^\top\nabla\Phi^k_{B_n} + \mathbf{I}_{\tau_n}\right) y_n = -\left(\alpha^k_{B_n} + \Phi^k_{B_n}\right)$
26:         Compute the updates
27:         $w^{k+1} = w^k + \gamma \cdot \nabla\Phi^k_{B_n} y_n$ ▷ full vector's update of the variable w^k
28:         $\alpha^{k+1}_{B_n} = \alpha^k_{B_n} + \gamma \cdot y_n$ ▷ τ_n coordinates' update of the auxiliary variable α^k
29:         $\overline{\alpha}^{k+1} = \overline{\alpha}^k + \gamma \cdot \frac{1}{\lambda n}\mathbf{A}_{:,B_n}y_n$
30: **return:** last iterate α^k, w^k

In average, which means taking the Bernoulli parameter b into account, the total cost per iteration of the TCS updates in expectation is

$$(\text{G.5}) \quad \begin{aligned} c_{avg} &\overset{\text{def}}{=} c_d \times (1-b) + c_n \times b \\ &= \mathcal{O}(\max(\tau_d^3, \tau_d n, \tau_d d)) \times (1-b) + \mathcal{O}(\max(\tau_n^3, \tau_n^2 d)) \times b. \end{aligned}$$

Depending on the sketch sizes (τ_d, τ_n) and the Bernoulli parameter b, the nature of c_{avg} can be different from $\mathcal{O}(d)$ (see Kaczmarz-TCS in Algorithm 3) to $\mathcal{O}(d^2)$ (see the cost per iteration analysis paragraph in the next section). We discuss the total cost

TABLE 2
Details of the parameters' choices (γ and b) for 50-TCS, 150-TCS and 300-TCS

dataset	stepsize	50-TCS Bernoulli	150-TCS Bernoulli	300-TCS Bernoulli
covetype	1.0	$\frac{n}{n+\tau_n*3}$	$\frac{n}{n+\tau_n*3}$	$\frac{n}{n+\tau_n*3}$
a9a	1.5	$\frac{n}{n+\tau_n}-0.03$	$\frac{n}{n+\tau_n}-0.03$	$\frac{n}{n+\tau_n}-0.11$
fourclass	1.0	$\frac{n}{n+\tau_n}-0.11$	$\frac{n}{n+\tau_n}-0.11$	$\frac{n}{n+\tau_n}-0.11$
artificial	1.0	$\frac{n}{n+\tau_n}-0.03$	$\frac{n}{n+\tau_n}-0.11$	$\frac{n}{n+\tau_n}-0.11$
ijcnn1	1.8	$\frac{n}{n+\tau_n}-0.03$	$\frac{n}{n+\tau_n}-0.11$	$\frac{n}{n+\tau_n}-0.11$
webspam	1.8	$\frac{n}{n+\tau_n*3}$	$\frac{n}{n+\tau_n*3}$	$\frac{n}{n+\tau_n*3}$
epsilon	1.8	$\frac{n}{n+\tau_n*3}$	$\frac{n}{n+\tau_n*3}$	$\frac{n}{n+\tau_n*3}$
phishing	1.8	$\frac{n}{n+\tau_n}-0.03$	$\frac{n}{n+\tau_n}-0.11$	$\frac{n}{n+\tau_n}-0.11$

per iteration of the TCS method in practice in different cases in the next section.

Appendix H. Additional experimental details.

All the sampling of the methods was pre-computed before starting counting the wall-clock time for each method and each dataset. We also paused the timing when the algorithms were under process of the performance evaluation of the gradient norm or of the logistic regression loss that were necessary to generate the plots.

In the following, from the experimental results for GLM in Section 8.2, we discuss the parameters' choices for TCS in practice, including the sketching sizes (τ_d, τ_n), the Bernoulli parameter b, the stepsize γ and the analysis of total cost per iteration. See Table 2 for the parameters we chose for TCS in the experiments in Figure 1. Such choices are due to TCS's cost per iteration.

Choice of the sketch size τ_d. For all of our experiments, $\tau_d = d$ performs always the best in time and in number of iterations. That means, when $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ at kth iteration, we choose $\mathbf{S}_d = \mathbf{I}_d$. Note also that the first d rows (8.4) are linear, thus using $\mathbf{S}_d = \mathbf{I}_d$ gives an exact solution to these first d equations. We found that such an exact solution from the linear part induces a fast convergence when $d < n$. We did not test datasets for which $d > n$ with d very large.

Choice of the Bernoulli parameter b for uniform sampling. First, we calculate the probability of sampling one row of the function F (8.4). Since there exists two types of sketching for TCS method depending on the *coin toss*, we address both of them. The probability of sampling one specific row of the first block (fist d rows of (8.4)) is

$$p_d = \frac{\tau_d}{d} \times (1 - b)$$

and the one of the second block is

$$p_n = \frac{\tau_n}{n} \times b.$$

It is natural to choose b such that the uniform sampling of the whole system, i.e. $p_d = p_n$, is guaranteed. This implies to set

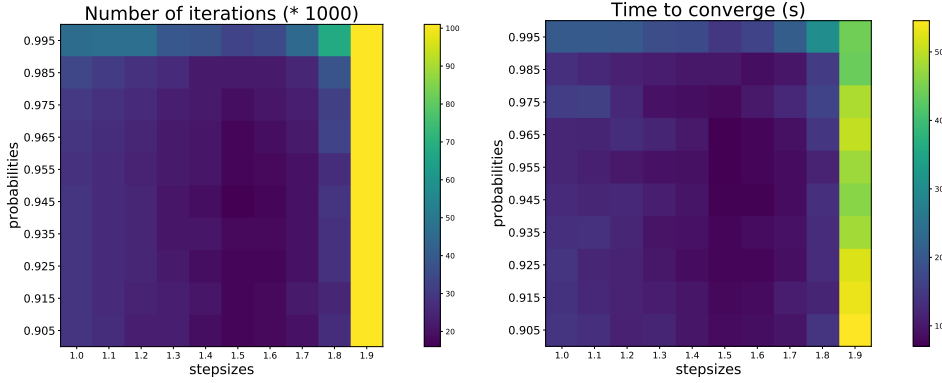$$p_{uniform} \stackrel{\text{def}}{=} \frac{\tau_d n}{\tau_d n + \tau_n d}.$$

FIG. 2. *a9a dataset: Grid search of the Bernoulli parameter b and the stepsize γ with 150-TCS method. The darker colors correspond to a resulting small gradient norm and thus a better solution.*

As we choose $\tau_d = d$, this implies

$$p_{uniform} = \frac{n}{n + \tau_n}.$$

However, we found through multiple experiments that when setting b slightly smaller than $p_{uniform}$ (e.g. -1%), this reduces significantly the number of iterations to get convergence. See in Figure 2 for a grid search of the Bernoulli parameter b and the stepsize γ for a9a dataset with $b = p_{uniform} = 0.995$ in the first line of the figure. Before giving details about how to choose b in practice, we first provide the cost per iteration analysis of the TCS method in detail.

Total cost per iteration in expectation analysis for TCS in different cases. Recall two types of costs per iteration c_d (G.3) and c_n (G.4). In our cases, consider $b = \frac{n}{n + \tau_n}$ and $d = \tau_d < n$. To summarize, the cost per iteration in expectation can be one of the three following cases followed with their bounds:

1. If $\tau_n < \sqrt{n} < d < n$ such as *epsilon* dataset, then $c_d = \mathcal{O}(d^3) > c_n = \mathcal{O}(\tau_n^2 d)$, and

$$c_{avg1} = \mathcal{O}(d^3) \times \left(1 - \frac{n}{n + \tau_n}\right) + \mathcal{O}(\tau_n^2 d) \times \frac{n}{n + \tau_n}$$

$$= \mathcal{O}(\frac{\tau_n d}{n + \tau_n}(d^2 + \tau_n n))$$

(H.1) $\implies \mathcal{O}(\tau_n^2 d) \leq c_{avg1} \leq \mathcal{O}(\tau_n d^2);$

2. if $\tau_n < d < \sqrt{n}$ such as *webspam* dataset with 50-TCS and 150-TCS, *a9a*, *phishing* and *covtype* datasets with 50-TCS method, then $c_d = \mathcal{O}(dn) > c_n = \mathcal{O}(\tau_n^2 d)$, and

$$c_{avg2} = \mathcal{O}(dn) \times \left(1 - \frac{n}{n + \tau_n}\right) + \mathcal{O}(\tau_n^2 d) \times \frac{n}{n + \tau_n}$$

(H.2) $= \mathcal{O}(\frac{\tau_n n}{n + \tau_n}(d + \tau_n d)) = \mathcal{O}(\tau_n^2 d) \quad \text{as } \frac{1}{2} \leq \frac{n}{n + \tau_n} < 1;$

3. if $d < \sqrt{n}$ and $d < \tau_n$ such as all the other experiments for TCS methods in

TABLE 3
Cost per iteration for different datasets and different sketch sizes.

dataset	50-TCS	150-TCS	300-TCS
covetype	$c_d > c_n$	$c_d > c_n$	$c_d > c_n$
a9a	$c_d > c_n$	$c_d > c_n$	$c_d < c_n$
fourclass	$c_d < c_n$	$c_d < c_n$	$c_d < c_n$
artificial	$c_d > c_n$	$c_d < c_n$	$c_d < c_n$
ijcnn1	$c_d > c_n$	$c_d < c_n$	$c_d < c_n$
webspam	$c_d > c_n$	$c_d > c_n$	$c_d > c_n$
epsilon	$c_d > c_n$	$c_d > c_n$	$c_d > c_n$
phishing	$c_d > c_n$	$c_d < c_n$	$c_d < c_n$

Figure 1, then $c_d = \mathcal{O}(dn)$, $c_n = \mathcal{O}(\tau_n^3)$, and

$$
\begin{aligned}
c_{avg3} &= \mathcal{O}(dn) \times \left(1 - \frac{n}{n + \tau_n}\right) + \mathcal{O}(\tau_n^3) \times \frac{n}{n + \tau_n} \\
&= \mathcal{O}(\frac{\tau_n n}{n + \tau_n}(d + \tau_n^2)) \\
&= \mathcal{O}(\tau_n^3) > \mathcal{O}(d^3) \quad \text{as } \frac{1}{2} \le \frac{n}{n + \tau_n} < 1.
\end{aligned}
$$

(H.3)

Notice that $c_{avg1}, c_{avg2}, c_{avg3} \ll \mathcal{O}(dn)$ in general for large scale datasets with large n. For example, $c_{avg1} < \mathcal{O}(dn)$ when $\tau_n d < n$. This justifies that TCS method is cheaper than the first-order method which requires evaluating the full gradient and thus has a cost per iteration of at least $\mathcal{O}(dn)$. From c_{avg1} (H.1), we know that TCS method can have the same cost per iteration as the stochastic first-order methods which is $\mathcal{O}(d)$ in practice, such as SVRG [28], SAG [53], dfSDCA [54] and Quartz [49].

Furthermore, from the above analysis of computational cost, we can easily obtain the comparisons between c_d and c_n for different datasets and different sketch sizes in Table 3. These comparisons helped us to choose b as we detail in the following.

Choice of the Bernoulli parameter b in practice. From the above discussion about $p_{uniform}$, heuristically, we decrease b from $p_{uniform}$ to achieve faster convergence. For a large range of choices b, TCS converges. However, b affects directly the computational cost per iteration. From (G.5), we know that if $c_d > c_n$, decreasing b will increase the average cost of the method. In this case, there is a trade-off between the number of iterations and the average cost to achieve the fastest convergence in time (see Figure 2). For a large dataset with n large such as *epsilon, webspam and covtype*, we decrease b slightly, as for a small dataset, we make a relatively big decrease for b. If $c_d < c_n$, decreasing b will also decrease the average cost. In this case, we tend to decrease b even further. See Table 2 the choices of b.

Choice of the sketch size τ_n. As for τ_n, we observe that with bigger sketch size τ_n, the method requires less number of iterations to get convergence. From Figure 3, this is true for all the datasets except for *covtype* dataset. However, choosing bigger sketch size τ_n will also increase the cost per iteration. Consequently, there exists an optimal sketch size such that the method converges the fastest in time taking account the balance between the number of iterations and the cost per iteration. From the experiments in Figure 1, we show that $\tau_n = 150$ is in general a very good choice for any scale of n.
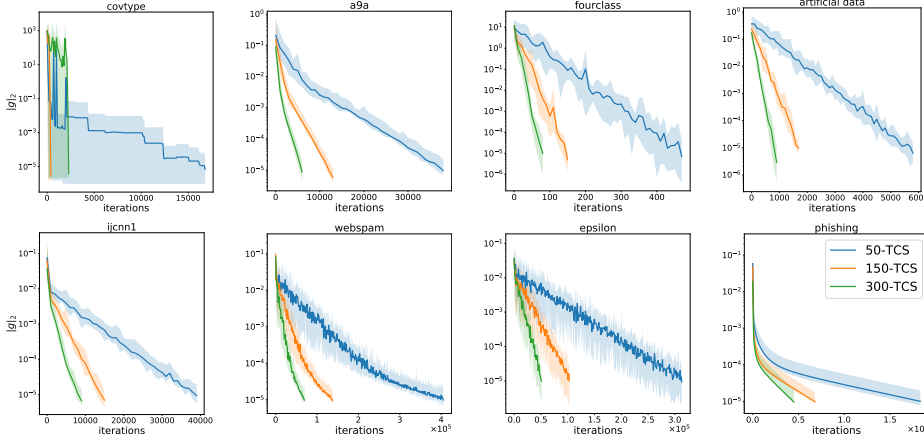
FIG. 3. *Comparisons of different sketch sizes for TCS method in terms of the number of iterations.*

Choice of the stepsizes. Different to our global convergence theories, in practice, choosing constant stepsize $\gamma > 1$ may converge faster (see Figure 2) for certain datasets. Here we need to be careful that the stepsize we mentioned is the stepsize used for the sketch $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$. As for the sketch $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$, we always choose stepsize $\gamma = 1$. Because stepsize $\gamma = 1$ solves exactly the linear system. Henceforth, we use γ to designate the stepsize used for the sketch of the last n rows of F (8.4). In our experiments, we find that the choice of the stepsize is related to the condition number (C.N.) of the model. If the dataset is ill-conditioned with a big C.N. of the model, $\gamma = 1$ is a good choice (see Figure 1 top row and Table 2 first four lines except for a9a); if the dataset is well-conditioned with a small C.N. of the model, all $\gamma \in (1, 1.8]$ gets convergence (see Figure 2). In practice, $\gamma = 1.8$ is a good choice for well-conditioned datasets (see Figure 1). However, from the grid search of stepsizes for a9a (see Figure 2), we know that the optimal stepsize for a9a is $\gamma = 1.5$. To avoid tuning the stepsizes, i.e. a grid search procedure, we will apply a stochastic line search process [59] in the next Section I.

Furthermore, we observe that the stepsize γ is highly related to the smoothness constant L. If L is big, then we choose γ close to 1, if L is small, we increase γ until $\gamma = 1.8$ (see Table 2). Such observation remains conjecture.

Finally, to summarize in practice for the TCS method with $d < n$, we choose $\tau_d = d$ and $\tau_n = 150$, we choose b following the guideline introduced above; we always choose stepsize $\gamma = 1$ for the sketch of first d rows (8.4); as for the sketch of last n rows (8.4), we choose stepsize $\gamma = 1$ if the dataset is ill-conditioned and we can choose stepsize $\gamma = 1.8$ if the dataset is well-conditioned.

Appendix I. Stochastic line-search for TCS methods applied in GLM.

In order to avoid tuning the stepsizes, we can modify Algorithm 1 by applying a stochastic line-search introduced by [59]. This is because again SNR can be interpreted as a SGD method. It is a *stochastic* line-search because on the kth iteration we sample a *stochastic* sketching matrix $\mathbf{S}_k$, and search for a stepsize γ_k satisfying the following

condition:

$$f_{\mathbf{S}_k,w^k}\left(w^k - \gamma_k \nabla f_{\mathbf{S}_k,w^k}(w^k)\right) \ \leq\ f_{\mathbf{S}_k,w^k}(w^k) - c \cdot \gamma_k \|\nabla f_{\mathbf{S}_k,w^k}(w^k)\|^2$$

$$(\text{I.1}) \qquad\qquad \overset{(4.1)}{=} (1 - 2c\gamma_k)\, f_{\mathbf{S}_k,w^k}(w^k).$$

Here, $c > 0$ is a hyper-parameter, usually a value close to 0 is chosen in practice.

I.1. Stochastic line-search for TCS method. Now we focus on GLMs, which means we develop the stochastic line-search based on (I.1) for TCS method. At kth iteration, if $\mathbf{S}_k = \begin{bmatrix} \mathbf{S}_d & 0 \\ 0 & 0 \end{bmatrix}$ with $\mathbf{S}_d = \mathbf{I}_{B_d}$, we sketch a linear system based on the first d rows of the Jacobian (8.5). Because of this linearity, the function $f_{\mathbf{S}_k,(\alpha^k,w^k)}(\alpha^k; w^k)$ is quadratic. Thus (I.1) can be re-written as

$$f_{\mathbf{S}_k,k}\left(\begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} - \gamma_k \nabla f_{\mathbf{S}_k,k}(\alpha^k; w^k)\right) = (1 - \gamma_k)^2 f_{\mathbf{S}_k,k}(\alpha^k; w^k)$$

$$(\text{I.2}) \qquad\qquad\qquad \leq (1 - 2c\gamma_k) f_{\mathbf{S}_k,k}(\alpha^k; w^k),$$

where we use the shorthand $f_{\mathbf{S}_k,k}(\alpha^k; w^k) \equiv f_{\mathbf{S}_k,(\alpha^k;w^k)}(\alpha^k; w^k)$. To achieve the Armijo line-search condition (I.2), it suffices to take $\gamma = 1$ and $0 < c \leq \frac{1}{2}$ which is a common choice. Consequently, we do not need extra function evaluations. It is also well known that stepsize equal to 1 is optimal as for Newton's method applied in quadratic problems.

If $\mathbf{S}_k = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{S}_n \end{bmatrix}$ with $\mathbf{S}_n = \mathbf{I}_{B_n}$, we have

$$(\text{I.3}) \quad f_{\mathbf{S}_k,k}(\alpha^k; w^k) = \frac{1}{2}\left(\alpha_{B_n}^k + \Phi_{B_n}^k\right)^\top \left([\nabla \Phi_{B_n}^k]^\top \nabla \Phi_{B_n}^k + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha_{B_n}^k + \Phi_{B_n}^k\right).$$

and

$$f_{\mathbf{S}_k,k}\left(\begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} - \gamma_k \nabla f_{\mathbf{S}_k,k}(\alpha^k; w^k)\right)$$

$$= \frac{1}{2}F(\alpha^k + \gamma_k \Delta\alpha^k; w^k + \gamma_k \Delta w^k)^\top \mathbf{H}_{\mathbf{S}_k}(\alpha^k; w^k) F(\alpha^k + \gamma_k \Delta\alpha^k; w^k + \gamma_k \Delta w^k)$$

$$= \frac{1}{2}F(\alpha^k + \gamma_k \Delta\alpha^k; w^k + \gamma_k \Delta w^k)^\top \begin{bmatrix} 0 \\ \mathbf{I}_{B_n} \end{bmatrix} \left([\nabla \Phi_{B_n}^k]^\top \nabla \Phi_{B_n}^k + \mathbf{I}_{\tau_n}\right)^\dagger$$

$$(\text{I.4}) \qquad \begin{bmatrix} 0 \\ \mathbf{I}_{B_n} \end{bmatrix}^\top F(\alpha^k + \gamma_k \Delta\alpha^k; w^k + \gamma_k \Delta w^k)$$

with

$$\begin{bmatrix} 0 \\ \mathbf{I}_{B_n} \end{bmatrix}^\top F(\alpha^k + \gamma_k \Delta\alpha^k; w^k + \gamma_k \Delta w^k)$$

$$(\text{I.5}) \qquad = \alpha_{B_n}^k + \gamma_k \mathbf{I}_{B_n}^\top \Delta\alpha^k + \phi'_{B_n}\left(\mathbf{A}_{:,B_n}^\top w^k + \gamma_k \mathbf{A}_{:,B_n}^\top \Delta w^k\right).$$

By (F.3), we recall that

$$(\text{I.6}) \qquad \Delta\alpha^k = -\mathbf{I}_{B_n}\left([\nabla \Phi_{B_n}^k]^\top \nabla \Phi_{B_n}^k + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha_{B_n}^k + \Phi_{B_n}^k\right),$$

$$(\text{I.7}) \qquad \Delta w^k = -\nabla \Phi_{B_n}^k \left([\nabla \Phi_{B_n}^k]^\top \nabla \Phi_{B_n}^k + \mathbf{I}_{\tau_n}\right)^\dagger \left(\alpha_{B_n}^k + \Phi_{B_n}^k\right).$$

Note that the cost for evaluating (I.3) and (I.4) are $\mathcal{O}(\tau_n)$ and $\mathcal{O}\left(\max\left(\tau_n^3, \tau_n d\right)\right)$ respectively, which are not expensive. Because one part of them are essentially a by-product from the computation of y_n, $\Delta\alpha^k$ and Δw^k in Algorithm 2. See Algorithm 5 the implementation of TCS combined with the stochastic Armijo line-search. $\beta \in (0,1)$ is a discount factor.

Algorithm 5 τ`--TCS+Armijo`

1: Choose $(\alpha^0; w^0) \in \mathbb{R}^{n+d}$, $c, \beta, \gamma > 0$, $\tau_d, \tau_n \in \mathbb{N}$ and $b \in (0,1)$.
2: Let $v \sim B(b)$ be a Bernoulli random variable (the coin toss)
3: **for** $k = 0, 1, \cdots$ **do**
4:      Sample $v \in \{0, 1\}$
5:      **if** $v = 0$ **then**
6:          Sample $B_d \subset \{1, \ldots, d\}$ with $|B_d| = \tau_d$ uniformly.
7:          Compute $y_d \in \mathbb{R}^{\tau_d}$ the least norm solution to
8:          $\left(\frac{\mathbf{A}_{B_d,:}\mathbf{A}_{B_d,:}^\top}{\lambda^2 n^2} + \mathbf{I}_{\tau_d}\right) y_d = \frac{\mathbf{A}_{B_d,:}\alpha^k}{\lambda n} - w_{B_d}^k$
9:          Compute the updates
10:          $\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = -\begin{bmatrix} \frac{1}{\lambda n}\mathbf{A}_{B_d,:}^\top \\ -\mathbf{I}_{B_d} \end{bmatrix} y_d$
11:          $w^{k+1} = w^k + \Delta w^k$
12:          $\alpha^{k+1} = \alpha^k + \Delta\alpha^k$
13:      **else**
14:          Reset γ to the initial stepsize.
15:          Sample $B_n \subset \{1, \ldots, n\}$ with $|B_n| = \tau_n$ uniformly.
16:          Compute $y_n \in \mathbb{R}^{\tau_n}$ the least norm solution to
17:          $\left([\nabla\Phi_{B_n}^k]^\top \nabla\Phi_{B_n}^k + \mathbf{I}_{\tau_n}\right) y_n = \alpha_{B_n}^k + \Phi_{B_n}^k$
18:          Compute the updates
19:          $\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix} = -\begin{bmatrix} \mathbf{I}_{B_n} \\ \nabla\Phi_{B_n}^k \end{bmatrix} y_n$
20:          **while** $f_{\mathbf{S}_k,k}\left(\begin{bmatrix} \alpha^k \\ w^k \end{bmatrix} + \gamma\begin{bmatrix} \Delta\alpha^k \\ \Delta w^k \end{bmatrix}\right) > (1 - 2c\gamma)f_{\mathbf{S}_k,k}(\alpha^k, w^k)$ **do**
21:             $\gamma \leftarrow \beta \cdot \gamma$
22:          $w^{k+1} = w^k + \gamma\Delta w^k$
23:          $\alpha^{k+1} = \alpha^k + \gamma\Delta\alpha^k$
24: **return:** last iterate α^k, w^k

I.2. Experimental results for stochastic line search. For all experiments, we set the initial stepsize $\gamma = 2$ with γ the stepsize for the last n rows' sketch and reduce the stepsize by a factor $\beta = 0.9$ when the line-search (I.1) is not satisfied. We choose the stepsize $\gamma = 1$ with γ the stepsize for the first d rows' sketch and $c = 0.09$.

From Figure 4, we observe that stochastic line search guarantees the convergence of the algorithm and does not tune any parameters. However, it slows down the convergence speed compared to the original algorithm with its rule of thumb parameters' choice. This is expected, as it does extra function evaluations at each step for the stochastic line search procedure.

<div align="center">REFERENCES</div>

[1] N. AGARWAL, B. BULLINS, AND E. HAZAN, *Second-order stochastic optimization for machine learning in linear time*, Journal of Machine Learning Research, 18 (2017), pp. 1–40.
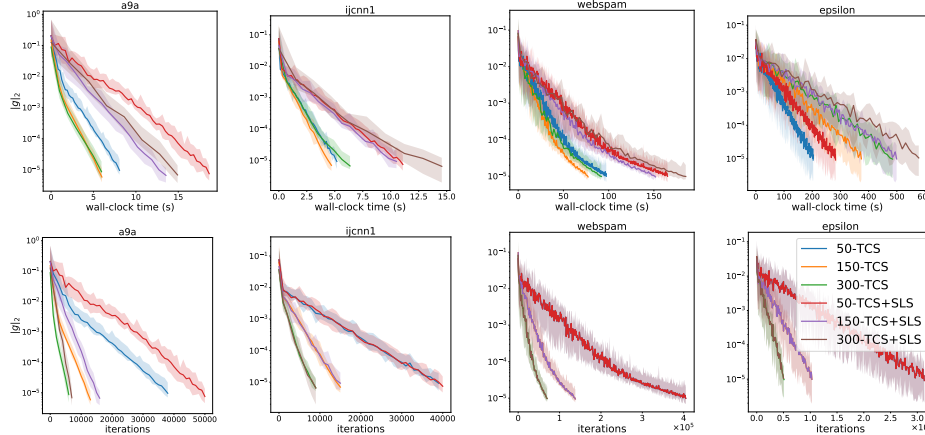
FIG. 4. *Experiments for TCS method combined with the stochastic line-search.*

[2] N. AILON AND B. CHAZELLE, *The fast Johnson-Lindenstrauss transform and approximate nearest neighbors*, SIAM J. Comput., 39 (2009), pp. 302–322.

[3] H. AN AND Z. BAI, *A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations*, Applied Numerical Mathematics, 57 (2007), pp. 235–252.

[4] F. BACH, *Learning Theory from First Principles*, The MIT Press, DRAFT, 2021.

[5] S. BELLAVIA AND B. MORINI, *A globally convergent Newton-GMRES subspace method for systems of nonlinear equations*, SIAM J. Sci. Comput., 23 (2001), pp. 940–960.

[6] A. BJÖRKLUND, P. KASKI, AND R. WILLIAMS, *Solving systems of polynomial equations over GF(2) by a parity-counting self-reduction*, in 46th International Colloquium on Automata, Languages, and Programming, vol. 132 of LIPIcs, 2019, pp. 26:1–26:13.

[7] J. BLACKARD AND D. DEAN, *Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables*, Computers and Electronics in Agriculture, (1999).

[8] R. BOLLAPRAGADA, R. H. BYRD, AND J. NOCEDAL, *Exact and inexact subsampled Newton methods for optimization*, IMA Journal of Numerical Analysis, 39 (2018), pp. 545–578.

[9] D. CALANDRIELLO, A. LAZARIC, AND M. VALKO, *Efficient second-order online kernel learning with adaptive embedding*, in Advances in Neural Information Processing Systems 30, 2017, pp. 6140–6150.

[10] E. J. CANDÈS, X. LI, AND M. SOLTANOLKOTABI, *Phase retrieval via wirtinger flow: Theory and algorithms*, IEEE Transactions on Information Theory, 61 (2015), pp. 1985–2007.

[11] C. CARTIS, N. I. M. GOULD, AND P. L. TOINT, *Adaptive cubic regularisation methods for unconstrained optimization . part i : motivation , convergence and numerical results*, Mathematical Programming, 127 (2009), pp. 1–38.

[12] C.-C. CHANG AND C.-J. LIN, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2 (2011), pp. 27:1–27:27.

[13] B. CHRISTIANSON, *Automatic Hessians by reverse accumulation*, IMA Journal of Numerical Analysis, (1992).

[14] A. R. CONN, N. I. M. GOULD, AND P. L. TOINT, *Trust-region Methods*, Society for Industrial and Applied Mathematics, 2000.

[15] A. DEFAZIO, F. BACH, AND S. LACOSTE-JULIEN, *SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives*, in Advances in Neural Information Processing Systems 27, 2014.

[16] M. DEREZINSKI, F. T. LIANG, Z. LIAO, AND M. W. MAHONEY, *Precise expressions for random projections: Low-rank approximation and randomized Newton*, in Advances in Neural Information Processing Systems, 2020.

[17] P. DEUFLHARD, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, 2011.

[18] D. DUA AND C. GRAFF, *UCI machine learning repository*, 2017.

[19] M. A. ERDOGDU AND A. MONTANARI, *Convergence rates of sub-sampled Newton methods*, in Advances in Neural Information Processing Systems 28, 2015, pp. 3052–3060.

[20] W. GAO AND D. GOLDFARB, *Quasi-Newton methods: superlinear convergence without line*

searches for self-concordant functions, Optimization Methods and Software, 34 (2019), pp. 194–217.

[21] R. Gower, D. Koralev, F. Lieder, and P. Richtarik, *RSN: Randomized subspace Newton*, in Advances in Neural Information Processing Systems 32, 2019, pp. 614–623.

[22] R. M. Gower, D. Goldfarb, and P. Richtárik, *Stochastic block BFGS: Squeezing more curvature out of data*, Proceedings of the 33rd International Conference on Machine Learning, (2016).

[23] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik, *SGD: General analysis and improved rates*, in Proceedings of the 36th International Conference on Machine Learning, vol. 97, 09–15 Jun 2019, pp. 5200–5209.

[24] R. M. Gower and P. Richtárik, *Randomized iterative methods for linear systems*, SIAM Journal on Matrix Analysis and Applications, 36 (2015), pp. 1660–1690.

[25] R. M. Gower and P. Richtárik, *Stochastic dual ascent for solving linear systems*, (2015).

[26] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo, *A globally convergent incremental Newton method*, Mathematical Programming, 151 (2015), p. 283–313.

[27] O. Hinder, A. Sidford, and N. Sohoni, *Near-optimal methods for minimizing star-convex functions and beyond*, in Proceedings of Thirty Third Conference on Learning Theory, 2020, pp. 1894–1938.

[28] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems 26, 2013, pp. 315–323.

[29] M. S. Kaczmarz, *Angenäherte auflösung von systemen linearer gleichungen*, Bulletin International de l'Académie Polonaise des Sciences et des Lettres. Série A, Sciences Mathématiques, 35 (1937), pp. 355–357.

[30] L. Kantorovitch, *The method of successive approximation for functional equations*, Acta Math., 71 (1939), pp. 63–97.

[31] K. Kawaguchi, *Deep learning without poor local minima*, in Advances in Neural Information Processing Systems 29, 2016, pp. 586–594.

[32] C. T. Kelley, *Numerical methods for nonlinear equations*, Acta Numerica, 27 (2018), p. 207–287.

[33] R. Kohavi, *Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid.*, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996.

[34] J. M. Kohler and A. Lucchi, *Sub-sampled cubic regularization for non-convex optimization*, in Proceedings of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 1895–1904.

[35] D. Kovalev, K. Mishchenko, and P. Richtarik, *Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates*, (2019).

[36] J. C. H. Lee and P. Valiant, *Optimizing star-convex functions*, in IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS, I. Dinur, ed., 2016, pp. 603–614.

[37] A. Leitão and B. F. Svaiter, *On projective Landweber–Kaczmarz methods for solving systems of nonlinear ill-posed equations*, Inverse Problems, 32 (2016), p. 025004.

[38] S. Lu, Z. Wei, and L. Li, *A trust region algorithm with adaptive cubic regularization methods for nonsmooth convex minimization*, Computational Optimization and Applications, 51 (2010), pp. 551–573.

[39] H. Luo, A. Agarwal, N. Cesa-Bianchi, and J. Langford, *Efficient second order online learning by sketching*, in Advances in Neural Information Processing Systems 29, 2016, pp. 902–910.

[40] S. Ma, R. Bassily, and M. Belkin, *The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning*, in Proceedings of the 35th International Conference on Machine Learning, 2018.

[41] R. Mohammad, F. Thabtah, and T. Mccluskey, *An assessment of features related to phishing websites using an automated technique*, in International Conference for Internet Technology and Secured Transactions, 2012.

[42] M. Mutny, M. Dereziński, and A. Krause, *Convergence analysis of block coordinate algorithms with determinantal sampling*, in International Conference on Artificial Intelligence and Statistics, 2020.

[43] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, 2 ed., 2014.

[44] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, Studies in Applied Mathematics, Society for Industrial and Applied Mathematics, 1994.

[45] Y. E. Nesterov and B. T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program., 108 (2006), pp. 177–205.

[46] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, vol. 43 of Springer Series in Operations Research, Springer, 1999.

[47] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Society for Industrial and Applied Mathematics, 2000.

[48] M. PILANCI AND M. J. WAINWRIGHT, *Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence*, SIAM Journal on Optimization, 27 (2017), pp. 205–245.

[49] Z. QU, P. RICHTÁRIK, AND T. ZHANG, *Quartz: Randomized dual coordinate ascent with arbitrary sampling*, in Advances in Neural Information Processing Systems 28, 2015.

[50] P. RICHTÁRIK AND M. TAKÁČ, *Stochastic reformulations of linear systems: Algorithms and convergence theory*, 2017.

[51] A. RODOMANOV AND D. KROPOTOV, *A superlinearly-convergent proximal newton-type method for the optimization of finite sums*, in Proceedings of The 33rd International Conference on Machine Learning, vol. 48 of Proceedings of Machine Learning Research, PMLR, 20–22 Jun 2016, pp. 2597–2605.

[52] F. ROOSTA-KHORASANI AND M. W. MAHONEY, *Sub-sampled Newton methods I: globally convergent algorithms*, (2016).

[53] M. SCHMIDT, N. LE ROUX, AND F. BACH, *Minimizing finite sums with the stochastic average gradient*, Mathematical Programming, 162 (2017), pp. 83–112.

[54] S. SHALEV-SHWARTZ, *SDCA without duality, regularization, and individual convexity*, Proceedings of The 33rd International Conference on Machine Learning, 48 (2016), pp. 747–754.

[55] S. SHALEV-SHWARTZ AND T. ZHANG, *Stochastic dual coordinate ascent methods for regularized loss*, Journal of Machine Learning Research, 14 (2013), pp. 567–599.

[56] T. STROHMER AND R. VERSHYNIN, *A randomized Kaczmarz algorithm with exponential convergence*, Journal of Fourier Analysis and Applications, 15 (2009), pp. 262–278.

[57] G. L. TORRES AND V. H. QUINTANA, *Optimal power flow by a nonlinear complementary method*, IEEE Transactions on Power Systems, 15 (2000), pp. 1028–1033.

[58] S. VASWANI, F. BACH, AND M. W. SCHMIDT, *Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron*, in AISTATS 2019, 2019, pp. 1195–1204.

[59] S. VASWANI, A. MISHKIN, I. LARADJI, M. SCHMIDT, G. GIDEL, AND S. LACOSTE-JULIEN, *Painless stochastic gradient: Interpolation, line-search, and convergence rates*, in Advances in Neural Information Processing Systems 32, 2019, pp. 3732–3745.

[60] D. WANG, D. IRANI, AND C. PU, *Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006*, in Proc. of 8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2012), October 2012.

[61] D. P. WOODRUFF, *Sketching as a tool for numerical linear algebra*, arXiv preprint arXiv:1411.4357, (2014).

[62] S. WRIGHT AND J. NOCEDAL, *Interior-point methods for nonlinear programming*, (2006), pp. 563–597.

[63] Y.-X. YUAN, *Recent advances in numerical methods for nonlinear equations and nonlinear least squares*, Numerical Algebra, Control & Optimization, 1 (2011), p. 15.

[64] D. ZHOU, P. XU, AND Q. GU, *Stochastic variance-reduced cubic regularized Newton methods*, in Proceedings of the 35th International Conference on Machine Learning, vol. 80, 10–15 Jul 2018.

[65] W. ZHOU, *On the convergence of the modified Levenberg–Marquardt method with a nonmonotone second order armijo type line search*, Journal of Computational and Applied Mathematics, (2013).

[66] W. ZHOU AND X. CHEN, *Global convergence of a new hybrid Gauss–Newton structured BFGS method for nonlinear least squares problems*, SIAM Journal on Optimization, 20 (2010), pp. 2422—-2441.

[67] Y. ZHOU, J. YANG, H. ZHANG, Y. LIANG, AND V. TAROKH, *SGD converges to global minimum in deep learning via star-convex path*, in International Conference on Learning Representations, 2019.