ROYAL
STATISTICAL
SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Statistical Methodology
Series B

J. R. Statist. Soc. B (2019)
81, *Part* 2, *pp.* 431–456

Statistical inference for the population landscape via moment-adjusted stochastic gradients

Tengyuan Liang

University of Chicago, USA

and Weijie J. Su

University of Pennsylvania, Philadelphia, USA

[Received December 2017. Final revision January 2019]

Summary. Modern statistical inference tasks often require iterative optimization methods to compute the solution. Convergence analysis from an optimization viewpoint informs us only how well the solution is approximated numerically but overlooks the sampling nature of the data. In contrast, recognizing the randomness in the data, statisticians are keen to provide uncertainty quantification, or confidence, for the solution obtained by using iterative optimization methods. The paper makes progress along this direction by introducing moment-adjusted stochastic gradient descent: a new stochastic optimization method for statistical inference. We establish non-asymptotic theory that characterizes the statistical distribution for certain iterative methods with optimization guarantees. On the statistical front, the theory allows for model misspecification, with very mild conditions on the data. For optimization, the theory is flexible for both convex and non-convex cases. Remarkably, the moment adjusting idea motivated from 'error standardization' in statistics achieves a similar effect to acceleration in first-order optimization methods that are used to fit generalized linear models. We also demonstrate this acceleration effect in the non-convex setting through numerical experiments.

Keywords: Acceleration; Diffusion process; Discretized Langevin algorithm; Model misspecification; Non-asymptotic inference; Population landscape; Stochastic gradient methods

1. Introduction

Statisticians are interested in inferring properties about a population based on independently sampled data. In the parametric regime, the inference problem boils down to constructing point estimates and confidence intervals for a finite number of unknown parameters. When the data generation process is well specified by the parametric family, an elegant asymptotic theory—credited to Ronald Fisher in the 1920s—has been established for maximum likelihood estimation. This asymptotic theory is readily generalizable to the model misspecification setting, for a properly chosen risk function $l(\theta, z)$ and the corresponding empirical risk minimizer

$$\hat{\theta}_{\mathrm{ERM}} \triangleq \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} l(\theta, z_i)$$

(empirical risk minimizer) and

Address for correspondence: Tengyuan Liang, Booth School of Business, University of Chicago, 5807 South Woodlawn Avenue, Chicago, IL 60637, USA.
E-mail: tengyuan.liang@chicagobooth.edu

1369–7412/19/81431

$$\theta_* \triangleq \arg\min_\theta \mathbb{E}_{\mathbf{z}\sim P}[l(\theta,\mathbf{z})]$$

(population minimizer), with

$$\surd N(\hat{\theta}_{\mathrm{ERM}} - \theta_*) \xrightarrow{\mathcal{L}} \mathcal{N}\{0, \mathbf{H}(\theta_*)^{-1}\boldsymbol{\Sigma}(\theta_*)\mathbf{H}(\theta_*)^{-1}\}.$$

Here θ is the parameter of the model, z_is are independent and identically distributed (IID) draws from an unknown distribution P, Hessian $\mathbf{H}(\theta) =^{\triangle} \mathbb{E}[\nabla_\theta^2 l(\theta,\mathbf{z})]$ and $\boldsymbol{\Sigma}(\theta) =^{\triangle} \mathbb{E}[\nabla_\theta l(\theta,\mathbf{z}) \otimes \nabla_\theta l(\theta,\mathbf{z})]$. Define the *population landscape* $L(\theta)$ as

$$L(\theta) \triangleq \mathbb{E}_{\mathbf{z}\sim P}[l(\theta,\mathbf{z})]. \tag{1}$$

(It is also called a loss function in the statistical learning literature. In generalized methods of moments, $\mathbb{E}_{\mathbf{z}\sim P}[\nabla_\theta l(\theta,\mathbf{z})] = 0$ is also called a moment condition. The maximum likelihood estimator can be also viewed as a special case with $l(\theta,\mathbf{z}) = -\log\{p_\theta(\mathbf{z})\}$ and the data generation process being $P = P_{\theta_*}$.). Note that the elegant statistical theory for inference holds under rather mild regularity conditions, without requiring a convex $L(\theta)$. However, it overlooks one important aspect: the optimization difficulty of the landscape on θ.

Optimization techniques are required to solve for the above estimator $\hat{\theta}$, as they rarely take a closed form. Global convergence and computational complexity are only well understood when the sample analogue $(1/N)\Sigma_{i=1}^N l(\theta, z_i)$ is convex. The optimization is done iteratively:

$$\theta_{t+1} = \theta_t - \eta\,\mathbf{h}(\theta_t), \tag{2}$$

where the vector field $\mathbf{h}$ is based on the first- and/or second-order information and η is the step size. For the non-convex case, the convergence becomes less clear, but in practice people still employ these iterative methods. Nevertheless, in either case, the available convergence results fall short of the statistical goal: after a certain number of iterations, we are interested in knowing the sampling distribution of θ_t, for uncertainty quantification of the optimization algorithm.

The goal of the present work is to combine the strength of the two worlds in inference and optimization: to characterize the statistical distribution of the iterative methods, with good optimization guarantee. Specifically, we study particular stochastic optimization methods for the (possibly non-convex) population landscape $L(\theta)$ in the fixed dimension regime, and at the same time characterize the sampling distribution at each step, through establishing a non-asymptotic theory. We allow for model misspecification and require only mild moment conditions on the data-generating process.

1.1. Motivation

Observe the simple fact that what we actually wish to optimize is the population objective $L(\theta) = \mathbb{E}_{\mathbf{z}\sim P}[l(\theta,\mathbf{z})]$, not the sample version. Therefore, stochastic approximation pioneered by Robbins and Monro (1951) and Kiefer and Wolfowitz (1952) stands out as a natural optimization approach for the statistical inference problem. In modern practice, *stochastic gradient descent* (SGD) with small batches of size n is widely used:

$$\theta_{t+1} = \theta_t - \eta\hat{\mathbb{E}}_n[\nabla_\theta l(\theta_t,\mathbf{z})], \tag{3}$$

where $\hat{\mathbb{E}}_n$ is the empirical expectation over n independently sampled minibatch data.

Our first observation follows from the intuition that a Gaussian approximation holds for each step when n is not too small, which we shall make rigorous in a moment. Define

$$\mathbf{b}(\theta) = \mathbb{E}_{\mathbf{z}\sim P}[\nabla_\theta l(\theta,\mathbf{z})], \tag{4}$$

$$\mathbf{V}(\theta) = \mathrm{cov}\{\nabla_\theta l(\theta, \mathbf{z})\}^{1/2}, \tag{5}$$

then observe the following approximation (6) via the central limit theorem (CLT)

$$\begin{aligned}\theta_{t+1} &= \theta_t - \eta \hat{\mathbb{E}}_n[\nabla_\theta l(\theta_t, \mathbf{z})] \\ &= \theta_t - \eta \mathbb{E}[\nabla_\theta l(\theta_t, \mathbf{z})] + \eta(\mathbb{E}[\nabla_\theta l(\theta_t, \mathbf{z})] - \hat{\mathbb{E}}_n[\nabla_\theta l(\theta_t, \mathbf{z})]) \\ &\approx \theta_t - \eta \mathbf{b}(\theta_t) + \surd(2\beta^{-1}\eta)\mathbf{V}(\theta_t)\mathbf{g}_t, \qquad \beta \triangleq 2n/\eta,\end{aligned} \tag{6}$$

where $\mathbf{g}_t$, $t \geqslant 0$, are independent isotropic Gaussian vectors. (The CLT states that for X_i, $i \in [n]$ IID sampled, asymptotically the following convergence in distribution holds:

$$\sqrt{n}\left\{(1/n)\sum_{i=1}^{n} X_i - \mathbb{E}[X]\right\} \xrightarrow{\mathcal{L}} \mathcal{N}\{0, \mathrm{cov}(X)\}.$$

If we substitute $X_i = \mathbf{V}(\theta_t)^{-1}\nabla_\theta l(\theta_t, Z_i)$, conditional on θ_t, one can see where the isotropic Gaussian distribution emerges.) The combination of n and η provides a stronger approximation guarantee at each iteration for large n, in contrast with the asymptotic normal approximation for the average trajectory in Polyak and Juditsky (1992) as $t \to \infty$. β^{-1} quantifies the 'variance' that is injected in each step (due to sampled minibatches), or the 'temperature' parameter: the larger the β is, the closer the distribution is concentrated near the deterministic steepest gradient descent updates. The scaling of the step size η relates to Cauchy discretization of the Itô diffusion process (as $\eta \to 0$)

$$\mathrm{d}\theta_t = -\mathbf{b}(\theta_t)\mathrm{d}t + \surd(2\beta^{-1})\mathbf{V}(\theta_t)\mathrm{d}B_t.$$

Our second observation comes from a classic 'standardization' idea in statistics—we want to adjust the stochastic gradient vector at step t by $\mathbf{V}(\theta_t)$ so that the conditional noise (conditioned on θ_t) for each co-ordinate is independent and homogeneous:

$$\begin{aligned}\theta_{t+1} &= \theta_t - \eta \mathbf{V}(\theta_t)^{-1}\hat{\mathbb{E}}_n[\nabla_\theta l(\theta_t, \mathbf{z})] \\ &\approx \theta_t - \eta \mathbf{V}(\theta_t)^{-1}\mathbf{b}(\theta_t) + \surd(2\beta^{-1})\eta \mathbf{g}_t.\end{aligned} \tag{7}$$

Namely, noisier gradient information is weighted less. This standardization trick in statistics is similar to the Newton or quasi-Newton method in second-order optimization, though with notable difference. The similarity lies in the fact that the noisy gradient information is weighted according to some local version of 'curvature'. However, the former uses the root of the second-moment matrix, whereas the latter uses Hessians (second-order derivatives).

To answer the inference question about $L(\theta)$ by using the 'moment-adjusted' iterative method proposed in approximation (7), we need to know the sampling distribution of θ_t for a fixed t. One hopes to describe the distribution directly in a non-asymptotic fashion, instead of characterizing this distribution either through the asymptotic normal limit (Polyak and Juditsky, 1992) (passing over data one at a time) in the convex scenario, or through the invariant distribution which could in theory take exponential time to converge for general non-convex $L(\theta)$ (Bovier *et al.*, 2004; Raginsky *et al.*, 2017). One thing to note is that, at a fixed time t, the distribution is distinct from Gaussian, for general $\mathbf{b}$ and $\mathbf{V}$. From an optimization angle, we would like the iterative algorithm to converge (to a local optimum) quickly. This is also important for inference: given the distribution can be approximately characterized at each step, one hopes that the distribution will concentrate near a local minimum of the population landscape $L(\theta)$ within a reasonable time budget, before the error accumulates in the stochastic process and invalidates the approximation.

1.1.1. Notation

For a vector v, $\|v\| = \surd(v^{\mathrm{T}} v)$ denotes the l_2-norm, and $v \otimes v = vv^{\mathrm{T}}$ denotes the outer product. We use $\|M\|$ to denote the operator norm for a matrix M. For a positive semidefinite matrix M, $\langle v, w \rangle_M = v^{\mathrm{T}} M w$. We use $t \in [T]$ to denote indices $0 \leqslant t \leqslant T$, and '$\to^{\mathcal{L}}$' for convergence in distribution. For two matrices A and B, we use $A \otimes_K B$ to represent the Kronecker product. Moreover, O and o are Bachmann–Landau notation and $O_{\mathbf{p}}$ denotes stochastic boundedness. In the discussion, we use $O_{\epsilon,\delta}(\cdot)$ to denote the order of magnitude for parameters ϵ and δ only, treating others as constants. For two probability measures μ and ν, we use $D_{\mathrm{KL}}(\mu, \nu)$ and $D_{\mathrm{TV}}(\mu, \nu)$ to denote the Kullback–Leibler and total variation distance respectively. Throughout, we denote the population gradient $\mathbf{b} \in \mathbb{R}^p$ and moment matrix $\mathbf{V}, \mathbf{\Sigma} \in \mathbb{R}^{p \times p}$, using the bold typeface notation, with the hope of emphasizing their role in the paper.

1.2. Contributions and organization

We propose the moment-adjusted SGD method called 'MasGrad', which is an iterative optimization method that infers the stationary points of the population landscape $L(\theta)$, namely $\{\theta \in \mathbb{R}^p : \|\nabla L(\theta)\| = 0\}$. MasGrad is a simple variant of SGD that adjusts the descent direction by using $\mathbf{V}(\theta_t)^{-1}$ (defined in equation (5), the square root of the inverse covariance matrix) at the current location:

$$\theta_{t+1} = \theta_t - \eta \mathbf{V}(\theta_t)^{-1} \hat{\mathbb{E}}_n[\nabla_\theta l(\theta_t, \mathbf{z})].$$

We summarize our main contributions in two perspectives. Extensions including estimation and computation of the moment-adjusted gradients will be discussed later in Section 6.

1.2.1. Inference

The distribution of MasGrad updates $\theta_t \in \mathbb{R}^p$, with n independently sampled minibatch data at each step, can be characterized in a non-asymptotic fashion. Informally, for any data-generating distribution $\mathbf{z} \sim P$ under mild conditions, the distribution of θ_t—denoted as $\mu(\theta_t)$—satisfies

$$D_{\mathrm{TV}}\{\mu(\theta_t), \nu_{t,\eta}\} \leqslant O_{t,n}\left\{\sqrt{\left(\frac{t}{n}\right)}\right\} \Rightarrow \mu(\theta_t) \xrightarrow{\mathcal{L}} \nu_{t,\eta},$$

converging in distribution as $n \to \infty$. Here $\nu_{t,\eta}$ is the distribution of ξ_t that follows the update initialized with $\xi_0 = \theta_0$:

$$\xi_{t+1} = \xi_t - \eta \mathbf{V}(\xi_t)^{-1} \mathbf{b}(\xi_t) + \surd(2\beta^{-1}) \eta \mathbf{g}_t, \qquad \mathbf{g}_t \sim \mathcal{N}(0, I_p),\ \beta = 2n/\eta. \tag{8}$$

We remark that $\nu_{t,\eta}$ depends only on t and η, and the first and second moments $\mathbf{b}$ and $\mathbf{V}$ of $\nabla l(\theta, \mathbf{z})$, regardless of the specific data-generating distribution $\mathbf{z} \sim P$. The rigorous statement is deferred to theorem 1 in Section 3, and further extensions to the continuous time analogue are discussed in Appendix A.

1.2.2. Optimization

Interestingly, in the strongly convex case such as in generalized linear models (GLMs), the 'standardization' idea achieves Nesterov acceleration (Nesterov, 1983, 2013). Informally, the number of iterations for an ϵ-minimizer for gradient descent requires

$$T_{\mathrm{GD}} = O_{\epsilon,\kappa}\left\{\kappa \log\left(\frac{1}{\epsilon}\right)\right\}, \qquad \text{for some } \kappa > 1.$$

We show that, for GLMs under mild conditions, MasGrad reduces the number of iterations to

$$T_{\text{MasGrad}} = O_{\epsilon,\kappa}\left\{\surd\kappa \log\left(\frac{1}{\epsilon}\right)\right\},$$

which matches Nesterov's acceleration in the strongly convex case. The formal statement is deferred to Section 4, where extensions including proximal updates are discussed.

Combining the inference and optimization theory, we present informally the results for both the *convex* and the *non-convex* cases. Recall that $\theta \in \mathbb{R}^p$.

1.2.3. Convex

In the strongly convex case, MasGrad with a properly chosen step size and the following choice of parameters

$$T = O_\epsilon\left\{\log\left(\frac{1}{\epsilon}\right)\right\},$$
$$n = O_{\epsilon,p}\left(\frac{p}{\epsilon}\right)$$

satisfies

$$D_{\text{TV}}\{\mu(\theta_T), \mu(\xi_T)\} \leqslant O_\epsilon[\surd\{\epsilon \log(1/\epsilon)\}]$$

(inference) and

$$\mathbb{E}[L(\theta_T)] - \min_\theta L(\theta) \leqslant \epsilon,$$
$$\mathbb{E}[L(\xi_T)] - \min_\theta L(\theta) \leqslant \epsilon, \qquad \xi_T \sim \nu_{T,\eta},$$

(optimization) where the evolution of ξ_t is defined in expression (8). Here the total number of samples needed is $nT = O_\epsilon\{\epsilon^{-1}\log(1/\epsilon)\}$. The formal result is stated in theorem 2 in Section 4.

1.2.4. Non-convex

Under mild smoothness conditions, MasGrad with a proper step size and the following choice of parameters

$$T = O_{\epsilon,\delta,p}\left(\frac{1 \vee p\delta^2}{\epsilon^2}\right),$$
$$n = O_{\epsilon,\delta,p}\left(\frac{\delta^{-2} \vee p}{\epsilon^2}\right)$$

satisfies

$$D_{\text{TV}}\{\mu(\theta_t, t \in [T])\},$$
$$\mu\{\xi_t, t \in [T]\} \leqslant O_\delta(\delta)$$

(inference) and

$$\mathbb{E}[\min_{t \leqslant T} \|\nabla L(\theta_t)\|] \leqslant \epsilon,$$
$$\mathbb{E}[\min_{t \leqslant T} \|\nabla L(\xi_t)\|] \leqslant \epsilon, \qquad \xi_t \sim \nu_{t,\eta}, \text{ for } t \in [T]$$

(optimization). Here the total number of samples needed is $nT = O_{\epsilon,\delta}(\epsilon^{-4}\delta^{-2})$. The formal result is deferred to theorem 4 in Section 5.

2. Relationships to the literature

In the case of a differentiable convex $L(\theta)$, finding a minimum is equivalent to solving $\nabla L(\theta)=0$. This simple equivalence reveals that the vanilla SGD, which takes the form

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta l(\theta_t, z_t), \tag{9}$$

is an instance of stochastic first-order approximation methods. (Realize that $\nabla_\theta l(\theta_t, z_t)$ is an unbiased estimate of the population gradient as $\nabla_\theta L(\theta_t) = \mathbb{E}_{\mathbf{z}\sim P}[\nabla_\theta l(\theta_t, \mathbf{z})]$.) This class of methods is iterative algorithms that attempt to solve fixed point equations (e.g. $\nabla L(\theta)=0$) provided noisy observations (e.g. $\nabla_\theta l(\theta_t, z_t)$) (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952; Toulis and Airoldi, 2017; Chen *et al.*, 2016; Li *et al.*, 2017). Using slowly diminishing step sizes $\eta_t = O(1/t^\alpha)$ ($\alpha < 1$), Ruppert (1988) and Polyak (1990) showed that acceleration using the average over trajectories of this recursive stochastic approximation algorithm attains the optimal convergence rate for a strongly convex L (see Polyak and Juditsky (1992) for more details). Recently, the running times of stochastic first-order methods have been considerably improved by using combinations of variance reduction techniques (Roux *et al.*, 2012; Johnson and Zhang, 2013) and Nesterov's acceleration (Ghadimi and Lan, 2012, 2016; Cotter *et al.*, 2011; Jofré and Thompson, 2017; Arjevani and Shamir, 2016).

Despite the celebrated success of stochastic first-order methods in modern machine learning tasks, researchers have kept improving the per-iteration complexity of second-order methods such as Newton or quasi-Newton methods, because of their faster convergence. A fruitful line of research has focused on how to improve the asymptotic convergence rate as $t \to \infty$ through preconditioning: a technique that involves approximating the unknown Hessian $\mathbf{H}(\theta) = \nabla_\theta^2 L(\theta)$ (see, for instance, Bordes *et al.* (2009) and references therein). Utilizing the curvature information that is reflected by various efficient approximations of the Hessian matrix, stochastic quasi-Newton methods (Moritz *et al.*, 2016; Byrd *et al.*, 2016; Wang *et al.*, 2017; Schraudolph *et al.*, 2007; Mokhtari and Ribeiro, 2015; Becker and Fadili, 2012), Newton sketching or subsampled Newton methods (Pilanci and Wainwright, 2015; Xu *et al.*, 2016; Berahas *et al.*, 2017; Bollapragada *et al.*, 2016) and stochastic approximation of the inverse Hessian via Taylor series expansion (Agarwal *et al.*, 2017) have been proposed to strike a balance between convergence rate and per-iteration complexity.

In the information geometry literature, one closely related method is the natural gradient (Amari, 1998, 2012). When the parameter space enjoys a certain structure, it has been shown that the natural gradient method outperforms the classic gradient descent method both theoretically and empirically. To adapt the natural gradient to our setting, we relate the loss function to a generative model $l(\theta, z) = -\log\{p_\theta(z)\}$. The Riemannian structure of the parameter space (manifold) of the statistical model is defined by the Fisher information

$$\mathbf{I}(\theta) = \mathbb{E}_{\mathbf{z}\sim P}[\nabla_\theta l(\theta, \mathbf{z}) \otimes \nabla_\theta l(\theta, \mathbf{z})].$$

The natural gradient can be viewed as the steepest descent induced by the Riemannian metric

$$\begin{aligned}\theta_{t+1} &= \arg\min_\theta \left\{ L(\theta_t) + \langle \nabla_\theta L(\theta_t), \theta - \theta_t \rangle + \frac{1}{2\eta_t} \|\theta - \theta_t\|^2_{\mathbf{I}(\theta_t)} \right\} \\ &= \theta_t - \eta_t \mathbf{I}(\theta_t)^{-1} \nabla_\theta L(\theta_t).\end{aligned}$$

Note the intimate connection between natural gradient descent and the approximate second-order optimization method, as the Fisher information can be heuristically viewed as an approximation of the Hessian (Schraudolph, 2002; Martens, 2014).

Another popular and closely related example as such is 'AdaGrad' (Duchi *et al.*, 2011), which is a variant of SGD that adaptively determines learning rates for different co-ordinates by incorporating the geometric information of past iterates. In its simplest form, AdaGrad records previous gradient information through

$$G_t = \sum_{i=1}^{t} \nabla l(\theta_i, z_i) \otimes \nabla l(\theta_i, z_i),$$

and this procedure then updates iterates according to

$$\theta_{t+1} = \theta_t - \gamma G_t^{-1/2} \nabla l(\theta_t, z_t),$$

where $\gamma > 0$ is fixed. In large-scale learning tasks, evaluating $G_t^{-1/2}$ is computationally prohibitive and thus it is often suggested to use $\text{diag}(G_t)^{-1/2}$ instead. It should be noted, however, that the theoretical derivation of the regret bound for AdaGrad considers $G_t^{-1/2}$. AdaGrad is a flexible improvement on SGD and can easily extend to non-smooth optimization and non-Euclidean optimization such as mirror descent. With the geometric structure G_t learned from past gradients, AdaGrad assigns different learning rates to different components of the parameter, allowing infrequent features to take relatively larger learning rates. This adjustment is shown to speed up convergence dramatically in a wide range of empirical problems (Pennington *et al.*, 2014).

Stochastic gradient Langevin dynamics have been an active research field in sampling and optimization in recent years (Welling and Teh, 2011; Dalalyan, 2017; Bubeck *et al.*, 2015; Raginsky *et al.*, 2017; Mandt *et al.*, 2017; Brosse *et al.*, 2017; Tzen *et al.*, 2018; Durmus *et al.*, 2018). Stochastic gradient Langevin dynamics inject additional $\surd(2\beta^{-1}\eta)$-level isotropic Gaussian noise to each step of SGD with step size η, where β is the inverse temperature parameter. Besides similar optimization benefits to those of SGD such as convergence and chances of escaping stationary points, the injected randomness of stochastic gradient Langevin dynamics provides an efficient way of sampling from the targeted invariant distribution of the continuous time diffusion process, which has been shown to be useful statistically in Bayesian sampling (Welling and Teh, 2011; Mandt *et al.*, 2017; Durmus *et al.*, 2018).

In the current paper, we take a distinct approach: we motivate and analyse a variant of SGD through the lens of Langevin dynamics, from a frequentist point of view, and then present the optimization benefits as a by-product of the statistical motivation. The approximation in equation (6) relates the density evolution of θ_s to a discretized version of an Itô diffusion process (as $\eta \to 0$)

$$\mathrm{d}\theta_s = -\mathbf{b}(\theta_s)\mathrm{d}s + \surd(2\beta^{-1})\mathbf{V}(\theta_s)\mathrm{d}B_s.$$

The invariant distribution $\pi(\theta)$ satisfies the Fokker–Planck equation

$$\beta^{-1} \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}(\pi \mathbf{a}_{ij}) + \sum_i \frac{\partial}{\partial x_i}(\pi \mathbf{b}_i) = 0$$

where $\mathbf{a}_{ij}(x) = (\mathbf{V}(x)\mathbf{V}(x)')_{ij}$. In general, the stationary distribution is difficult to characterize unless both $\mathbf{V}$ and $\mathbf{b}$ take special simple forms. For example, when $\mathbf{b}(x)$ is linear and $\mathbf{V}(x)$ is independent of x as in Mandt *et al.* (2017), the diffusion process reduces to an Ornstein–Uhlenbeck process with a multivariate Gaussian distribution as the invariant distribution. Another simple case is, when $\mathbf{V}(x)=\mathbf{I}$, the diffusion process is also referred to as Langevin dynamics, with the Gibbs measure $\pi(\theta)\propto\exp\{-\beta L(\theta)\}$ as the unique invariant distribution (Welling and Teh, 2011; Dalalyan, 2017; Raginsky *et al.*, 2017).

3. Statistical inference via Langevin diffusion

In this section we shall explain why MasGrad produces recursive updates whose statistical distribution can be characterized. We mention that MasGrad at the same time achieves significant acceleration in optimization in the strongly convex case (detailed in Section 4). For the general non-convex case, we provide non-asymptotic theory for inference and optimization in Section 5. We first present the simplest version of the algorithm, assuming that $\mathbf{V}(\theta)^{-1}$ can be evaluated at any given θ. Statistical estimation and efficient direct computation of $\mathbf{V}(\theta)^{-1}$ will be discussed in Section 6.

Recall the moment-adjusted SGD that we introduced, which adjusts the gradient direction by using the root of the inverse covariance matrix at the current location:

$$\theta_{t+1}=\theta_t-\eta\mathbf{V}(\theta_t)^{-1}\hat{\mathbb{E}}_n[\nabla_\theta l(\theta_t,\mathbf{z})]. \tag{10}$$

As we have heuristically outlined in equation (6), MasGrad can be approximated by the following discretized Langevin diffusion:

$$\xi_{t+1}=\xi_t-\eta\mathbf{V}(\xi_t)^{-1}\mathbf{b}(\xi_t)+\surd(2\beta^{-1}\eta)\mathbf{g}_t. \tag{11}$$

In this section, we establish non-asymptotic bounds on the distance between the distribution of the MasGrad process $\mathcal{L}(\theta_t,t\in[T])$ and discretized diffusion process $\mathcal{L}(\xi_t,t\in[T])$.

The proof is based on the entropic CLT (Barron, 1986; Bobkov *et al.*, 2013, 2014). The classic CLT based on convergence in distribution is too weak for our purpose: we need to translate the non-asymptotic bounds at each step to the whole stochastic process. It turns out that the entropic CLT couples naturally with the chain rule property of relative entropy, which together provide non-asymptotic characterization on closeness of the distributions for the stochastic processes.

We first state the standard assumptions for entropic CLT. These assumptions can be found in Bobkov *et al.* (2013). We remark that we are focusing on the fixed dimension setting.

Assumption 1 (absolute continuity to Gaussian). Assume that random vector $X\in\mathbb{R}^p$ has bounded entropic distance to the Gaussian distribution, for some constant D_1:

$$D_{\mathrm{KL}}\{\mu(X)||\mu(\mathbf{g})\}<D_1, \qquad \mathbf{g}\sim\mathcal{N}(0,I_p).$$

Assumption 2 (finite $(4+\delta)$th moments). Assume that there is a constant D_2,

$$\mathbb{E}||X||^{4+\delta}<D_2, \qquad \text{for some small } \delta>0.$$

Define, $\forall\, i$, the stochastic component of the adjusted gradient direction:

$$X_i(\theta)=\mathbf{V}(\theta)^{-1}\,(\nabla_\theta l(\theta,z_i)-\mathbb{E}_{\mathbf{z}\sim P}[\nabla_\theta l(\theta,\mathbf{z})]). \tag{12}$$

It is clear that X_is are IID with $\mathbb{E}[X_i(\theta)]=0$ and $\operatorname{cov}\{X_i(\theta)\}=I_p$. Here $X_i(\theta)$ is defined on the same σ-field as z_i drawn from P.

Theorem 1 (non-asymptotic bound for inference). Let $\mu(\theta_t,t\in[T])$ denote $\mathcal{L}(\theta_t,t\in[T])$, the joint distribution of the MasGrad process, and $\mu(\xi_t,t\in[T])$ be the joint distribution of the discretized diffusion process (11). Consider the same initialization $\theta_0=\xi_0$.

Assume that, uniformly for any θ, $X(\theta)$ defined in equation (12) satisfies assumptions 1 and 2 with constants D_1 and D_2 that depend only on p. Then the following bound holds:

$$D_{\mathrm{TV}}\{\mu(\theta_t, t\in[T]), \mu(\xi_t, t\in[T])\} \leqslant C\sqrt{\left[\frac{T}{n} + o\left\{\frac{T\log(n)^{\{p-(4+\delta)\}/2}}{n^{1+\delta/2}}\right\}\right]}, \tag{13}$$

where C is some constant that depends on D_1 and D_2 only.

Remark 1. Theorem 1 characterizes the sampling distribution of MasGrad—θ_t, using a measure that depends only on the first and second moments of $\nabla l(\theta, \mathbf{z})$, namely $\mathbf{V}(\theta)^{-1}\mathbf{b}(\theta)$, regardless of the specific data-generating distribution $\mathbf{z} \sim P$. Observe that the distribution closeness is established in a strong total variation distance sense, for the two stochastic processes $\{\theta_t, t\in[T]\}$ and $\{\xi_t, t\in[T]\}$. If we dig into the proof, we can easily obtain the following marginal result:

$$D_{\mathrm{TV}}\{\mu(\theta_T), \mu(\xi_T)\} \leqslant \surd[2D_{\mathrm{KL}}\{\mu(\theta_T)||\mu(\xi_T)\}] \leqslant \surd[2D_{\mathrm{KL}}\{\mu(\theta_t, t\in[T])||\mu(\xi_t, t\in[T])\}],$$

where the last inequality follows from the chain rule of relative entropy. Therefore, we can as well prove for the last step distribution

$$D_{\mathrm{TV}}\{\mu(\theta_T), \mu(\xi_T)\} \leqslant C\sqrt{\left(\frac{T}{n}\right)}.$$

Remark 2. One important fact about theorem 1 is that it holds for any step size η, which provides us with the additional freedom of choosing the optimal step size for the optimization purpose. Theorem 1 is stated in the fixed dimensional setting when p does not change with n. We remark in addition that the Gaussian approximation at each step still holds with high probability, in the moderate dimensional setting when

$$p = o\left[\frac{\log(n)}{\log\{\log(n)\}}\right],$$

as shown in the non-asymptotic bound in theorem 1. We emphasize that the current paper considers only the fixed dimension setting, while considering the minibatch sample size n and running time T varying. Assumptions 1 and 2 are standard assumptions in the entropic CLT: assumption 1 states that the distribution for each stochastic gradient is non-lattice with bounded relative entropy to Gaussian; assumption 2 is the standard weak moment condition. Note here that the constants D_1 and D_2 depend on the dimension implicitly.

For statistical inference, we can always approximately characterize the distribution of MasGrad by using theorem 1. As an additional benefit, the result naturally provides us with an algorithmic way of sampling this target universal distribution $\mu(\xi_t)$. For some particular tasks, it remains of theoretical interest to characterize the distribution of MasGrad analytically by using the continuous time Langevin diffusion and its invariant distribution. We defer the analysis of the discrepancy between the discretized diffusion to the continuous time analogue to the on-line appendix A.

4. Convexity and acceleration

In this section, we shall demonstrate that the 'moment adjusting' idea motivated from standardizing the error from an inference perspective achieves a similar effect to those of acceleration in convex optimization. We shall investigate GLMs as the main example. Later, we shall also discuss the case with non-smooth regularization. Using first-order information to achieve acceleration was first established in the seminal work by Nesterov (1983, 2013) based on the ingenious notion of an estimating sequence. Before diving into the technical analysis, we point out that in MasGrad the moment adjusting matrix $\mathbf{V}(\theta)$ can be estimated by using only first-order information; however, as we shall see, MasGrad achieves acceleration for GLMs in a way that resembles the approximate second-order method such as the quasi-Newton method.

4.1. Inference and optimization for optima

Now we are ready to state the theory for inference and optimization using MasGrad in the strongly convex case. Let $L(w):\mathbb{R}^p \to \mathbb{R}$ be a smooth convex function. Recall that $\mathbf{b}(w)=\nabla L(w)$, $\mathbf{H}(w)=\nabla^2 L(w)$ and $\mathbf{V}(w)\in\mathbb{R}^{p\times p}$ are positive definite matrices. Define

$$\begin{aligned}\alpha &\triangleq \min_{v,w} \lambda_{\min}\{\mathbf{V}(w)^{-1/2}\mathbf{H}(v)\mathbf{V}(w)^{-1/2}\}>0,\\ \gamma &\triangleq \max_{v,w} \lambda_{\max}\{\mathbf{V}(w)^{-1/2}\mathbf{H}(v)\mathbf{V}(w)^{-1/2}\}>0.\end{aligned} \tag{14}$$

Theorem 2 (MasGrad: strongly convex). Let α and γ be defined as in expression (14). Consider the MasGrad updates θ_t in equation (10) with step size $\eta=1/\gamma$, and the corresponding discretized diffusion ξ_t,

$$\xi_{t+1}=\xi_t-\eta\mathbf{V}(\xi_t)^{-1}\mathbf{b}(\xi_t)+\sqrt{(2\beta^{-1}\eta)}\mathbf{g}_t, \qquad \beta=2n/\eta.$$

Then, for any precision $\epsilon>0$, we can choose

$$\begin{aligned}T&=\frac{\gamma}{\alpha}\log\left[\frac{2\{L(\theta_0)-\min_\theta L(\theta)\}}{\epsilon}\right],\\ n&=\frac{4p\max_\theta\|\mathbf{V}(\theta)\|}{\alpha\epsilon},\end{aligned} \tag{15}$$

such that

(a) $D_{\mathrm{TV}}\{\mu(\theta_t,t\in[T]),\mu(\xi_t,t\in[T])\}\leqslant O_\epsilon[\sqrt{\{\epsilon\log(1/\epsilon)\}}]$ and
(b) $\mathbb{E}[L(\theta_t)]-\min_\theta L(\theta)\leqslant\epsilon$ and $\mathbb{E}[L(\xi_t)]-\min_\theta L(\theta)\leqslant\epsilon$,

with in total $O_\epsilon\{\epsilon^{-1}\log(1/\epsilon)\}$ independent data samples.

Remark 3. In plain language, the discretized diffusion process ξ_t, $t\in[T]$, whose distribution depends on only the adjusted moments $\mathbf{V}^{-1}\mathbf{b}$, approximates the sampling distribution of MasGrad $\theta_t, t\in[T]$, in a strong sense, i.e. the distributions of paths are close in total variation distance. In addition, as a stochastic optimization method, MasGrad's optimization guarantee depends on the 'modified' condition number defined in expression (14). We sketch the proof. Using lemma B.1 in the on-line appendix B, for all $t>0$, one can prove that

$$\mathbb{E}[L(\xi_t)]-\min_\theta L(\theta)\leqslant\left(1-\frac{\alpha}{\gamma}\right)^t\{L(\theta_0)-\min_\theta L(\theta)\}+\max_\theta\|\mathbf{V}(\theta)\|\frac{\gamma}{\alpha}\beta^{-1}p.$$

Therefore we can define the condition number of MasGrad as

$$\begin{aligned}\kappa_{\mathrm{MasGrad}}&=\frac{\max_{w,v}\lambda_{\max}\{\mathbf{V}(w)^{-1/2}\mathbf{H}(v)\mathbf{V}(w)^{-1/2}\}}{\min_{w,v}\lambda_{\min}\left\{\mathbf{V}(w)^{-1/2}\mathbf{H}(v)\mathbf{V}(w)^{-1/2}\right\}},\\ \kappa_{\mathrm{GD}}&=\frac{\max_v\lambda_{\max}\{\mathbf{H}(v)\}}{\min_v\lambda_{\min}\{\mathbf{H}(v)\}},\end{aligned} \tag{16}$$

in contrast with the condition number in gradient descent.

If $\beta = 2n/\eta$ and T and n are chosen as in expression (15), we know that $\mathbb{E}[L(\xi_T)] - L(\theta_*) \leqslant \epsilon$. Recall the result that we established in theorem 1; the total variation distance between MasGrad and the discretized diffusion in this case is bounded by $\surd(T/n) = O_\epsilon[\surd\{\epsilon \log(1/\epsilon)\}]$, and the total number of samples used is of the order $nT = O_{\epsilon,p}\{p/\epsilon \log(1/\epsilon)\}$. This result can be contrasted with the classical asymptotic normality for maximum likelihood estimation or the empirical risk minimizer to achieve an ϵ-minimizer,

$$\epsilon \geqslant L(\hat{\theta}_N) - L(\theta_*) \asymp \|\hat{\theta}_N - \theta_*\|^2 \asymp \frac{p}{N} \Leftrightarrow N = O_{\epsilon,p}(p/\epsilon),$$

the asymptotic sample complexity scales $O_{\epsilon,p}(p/\epsilon)$. Similar calculations also hold with the Ruppert–Polyak average on stochastic approximation with a carefully chosen decreasing step size. As we can see, our result holds non-asymptotically, and it achieves both the optimization and the inference goal, with an additional logarithmic factor.

4.2. *Acceleration for generalized linear models*

Now we take GLMs as an example to articulate the effect of acceleration. We shall first use an illustrating toy example to show the intuition in an informal way, and then we present the rigorous acceleration result for GLMs.

4.2.1. *Toy example (informal)*

Consider $y_i = \langle x_i, \theta_* \rangle + \epsilon_i$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ IID for $i \in [N]$. We focus on the fixed design case (where the expectation is over $\mathbf{y}$ only); the loss $l\{\theta, (x, y)\} = \frac{1}{2}(\langle x, \theta \rangle - y)^2$. Denote $X \in \mathbb{R}^{N \times p}$; then we have

$$\mathbf{b}(\theta) = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}(x_i^{\mathrm{T}}\theta - y_i)x_i\right] = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^{\mathrm{T}}(\theta - \theta_*) = \frac{1}{N}X^{\mathrm{T}}X(\theta - \theta_*),$$

$$\mathbf{V}(w) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i^{\mathrm{T}}\sigma^2\right)^{1/2} = \sigma\left(\frac{1}{N}X^{\mathrm{T}}X\right)^{1/2},$$

and the Hessian is $\mathbf{H}(w) = X^{\mathrm{T}}X/N$. Therefore, in this case, we have

$$\kappa_{\text{MasGrad}} = \surd\kappa_{\text{GD}}.$$

By applying lemma B.1 in the on-line appendix B, we achieve the same effect as Nesterov's acceleration in the strongly convex case (Nesterov, 2013). We remark that the above analysis is to demonstrate the intuition and is not rigorous—as MasGrad is designed for the random design.

4.2.2. *Generalized linear models, random design and misspecified model*

Now we provide a rigorous and unified treatment for GLMs. Consider the GLM (McCullagh, 1984) where the response random variable $\mathbf{y}$ follows from the exponential family parameterized by (θ, ϕ):

$$f(y; \theta, \phi) = b(y, \phi)\exp\left\{\frac{y\theta - c(\theta)}{d(\phi)}\right\}$$

where $\mu = \mathbb{E}[\mathbf{y}|\mathbf{x}=x] = c'(\theta)$, $c''(\theta) > 0$ and the natural parameter satisfies the linear relationship $\theta = \theta(\mu) = x^{\mathrm{T}}w$. In this case, we choose the loss function according to the negative log-likelihood:

$$l\{w, (x, y)\} = -y_i x_i^{\mathrm{T}} w + c(x_i^{\mathrm{T}} w).$$

Special cases include

(a) the Bernoulli model (logistic regression), $c(\theta) = \log\{1 + \exp(\theta)\}$, where $x_i^{\mathrm{T}} w = \theta = \log\{\mu/(1-\mu)\}$,
(b) the Poisson model (Poisson regression), $c(\theta) = \exp(\theta)$, where $x_i^{\mathrm{T}} w = \theta = \log(\mu)$, and
(c) the Gaussian model (linear regression), $c(\theta) = \frac{1}{2}\theta^2$, where $x_i^{\mathrm{T}} w = \theta = \mu$.

We are interested in inference even when the model can be *misspecified*. Consider the statistical learning setting where $z_i = (x_i, y_i) \sim P = P_{\mathbf{x}} \times P_{\mathbf{y}|\mathbf{x}}$, $i \in [N]$ IID, from some unknown joint distribution P. We are trying to infer the parameters w by fitting the data by using a parametric exponential family; however, we allow the flexibility that the exponential family model for $P(\mathbf{y}|\mathbf{x}=x)$ can be misspecified. Specifically, the true regression function $m_*(x) = \mathbb{E}[\mathbf{y}|\mathbf{x}=x]$ may not be $c'(x^{\mathrm{T}}w)$ for all w, namely, it may not be realized by any model in the exponential family model class. We have the population landscape

$$L(w) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P}[-\mathbf{y}\mathbf{x}^{\mathrm{T}}w + c(\mathbf{x}^{\mathrm{T}}w)]. \tag{17}$$

Define the conditional variance $\xi(x) = \mathrm{var}(\mathbf{y}|\mathbf{x}=x) \in \mathbb{R}$ and the bias

$$\beta(\mathbf{x}, w) \triangleq c'(\mathbf{x}^{\mathrm{T}}w) - m_*(\mathbf{x}) \in \mathbb{R};$$

we have the following acceleration result for GLMs.

Theorem 3 (acceleration). Consider the condition number defined in expression (16) for MasGrad and gradient descent and assume that there is a constant $C > 1$ such that, for any x, w and v,

$$0 < \max\left\{ \frac{\xi(x)^2 + \beta(x, w)^2}{c''(x^{\mathrm{T}}v)}, \frac{c''(x^{\mathrm{T}}v)}{\xi(x)^2} \right\} < C^{1/3}.$$

Then, for the optimization problem that is associated with GLMs defined in equation (17), the following inequality holds:

$$\kappa_{\mathrm{MasGrad}} < C\sqrt{\kappa_{\mathrm{GD}}}.$$

Remark 4. Theorem 3 together with lemma B.1 in the on-line appendix B states that, in the noiseless setting, the time complexity for MasGrad is $O\{\sqrt{\kappa_{\mathrm{GD}}}\log(1/\epsilon)\}$ in contrast with the complexity of gradient descent—$O\{\kappa_{\mathrm{GD}}\log(1/\epsilon)\}$, which is crucial when the condition number is large. The proof is based on matrix inequalities and the analytic expressions

$$\mathbf{b}(w) = \mathbb{E}[-\mathbf{y}\mathbf{x} + c'(\mathbf{x}^{\mathrm{T}}w)\mathbf{x}] = \mathbb{E}[\{c'(\mathbf{x}^{\mathrm{T}}w) - m_*(\mathbf{x})\}\mathbf{x}],$$
$$\mathbf{V}(w) = \{\mathbb{E}[\xi(\mathbf{x})^2\mathbf{x}\mathbf{x}^{\mathrm{T}}] + \mathrm{cov}[\beta(\mathbf{x}, w)\mathbf{x}]\}^{1/2},$$
$$\mathbf{H}(w) = \mathbb{E}[c''(\mathbf{x}^{\mathrm{T}}w)\mathbf{x}\mathbf{x}^{\mathrm{T}}].$$

4.3. Non-smooth regularization

In this section, we extend the acceleration result to problems with non-smooth regularization. The main results are based on a simple modification called *moment-adjusted proximal gradient descent*, ‘MadProx’.

Consider the population loss function that can be decomposed into

$$L(w)=g(w)+h(w) \tag{18}$$

where $g(w)$ is a smooth and convex function in w, and $h(w)$ is a non-smooth regularizer that is convex. Special cases include

(a) sparse regression with $l\{w,(x_i,y_i)\}=\frac{1}{2}(x_i^{\mathrm{T}}w-y_i)^2+\lambda\|w\|_1$ and

$$L(w)=\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P}[\tfrac{1}{2}(\mathbf{x}^{\mathrm{T}}w-\mathbf{y})^2]+\lambda\|w\|_1:=g(w)+h(w)$$

and

(b) low rank matrix trace regression with $l\{W,(X_i,y_i)\}=\frac{1}{2}(\langle X_i,W\rangle-y_i)^2+\lambda\|W\|_*$

$$L(W)=\mathbb{E}_{(\mathbf{X},\mathbf{y})\sim P}[\tfrac{1}{2}(\langle\mathbf{X},W\rangle-\mathbf{y})^2]+\lambda\|W\|_*:=g(W)+h(W).$$

Now we shall show the role of moment matrix $\mathbf{V}$ in 'speeding up' the convergence of proximal gradient descent in the following proposition. Here we focus on an easier case when $\mathbf{V}(w)$ does not depend on w (as in the linear regression fixed design case, where $\mathbf{V}(w)=\mathbb{E}[\xi(\mathbf{x})^2\mathbf{x}\mathbf{x}^{\mathrm{T}}]^{1/2}$ does not depend on w).

Define the moment-adjusted proximal function and MadProx:

$$\operatorname{prox}_{\eta,\mathbf{V}}(w)=\arg\min_u\left\{\frac{1}{2\eta}\|u-w\|_{\mathbf{V}}^2+h(u)\right\}, \tag{19}$$

$$w_{t+1}=\operatorname{prox}_{\eta,\mathbf{V}}\{w_t-\eta\mathbf{V}^{-1}\nabla g(w_t)\}. \tag{20}$$

(MadProx).

Proposition 2 (moment-adjusted proximal). Consider $L(w)=g(w)+h(w)$ as in equation (18). Denote $\mathbf{H}$ as the Hessian of g, and define

$$\alpha\triangleq\min_v\lambda_{\min}\{\mathbf{V}^{-1/2}\mathbf{H}(v)\mathbf{V}^{-1/2}\}>0,$$
$$\gamma\triangleq\max_v\lambda_{\max}\{\mathbf{V}^{-1/2}\mathbf{H}(v)\mathbf{V}^{-1/2}\}>0.$$

Consider the MadProx updates that are defined in equation (20) with step size $\eta=1/\gamma$ and adjusting matrix $\mathbf{V}$. If

$$T\geqslant\frac{\gamma}{\alpha}\log\left(\frac{\alpha}{2\epsilon}\|w_0-w_*\|_{\mathbf{V}}^2+1\right),$$

we have $L(w_T)-\min_w L(w)\leqslant\epsilon$.

Remark 5. We can see that MadProx implements a moment-adjusted gradient (using implicit updates) because w_{t+1} satisfies the implicit equation

$$w_{t+1}=w_t-\eta\mathbf{V}^{-1}\{\nabla g(w_t)+\partial h(w_{t+1})\},$$

in comparison with the subgradient step (explicit updates)

$$w_{t+1}=w_t-\eta\mathbf{V}^{-1}\{\nabla g(w_t)+\partial h(w_t)\}.$$

We remark that, as in the GLMs case, the moment adjustment idea speeds up the computation as the number of proximal steps scales with the adjusted condition number $\kappa_{\mathrm{MadProx}}\approx\sqrt{\kappa_{\mathrm{GD}}}$. However, to be fair, it can be computationally difficult to implement each proximal step for a non-diagonal $\mathbf{V}$. Motivated from the diagonalizing idea in AdaGrad (Duchi *et al.*, 2011), we can substitute $\mathbf{V}$ by diag($\mathbf{V}$) to save the per-iteration computation.

5. Non-convex inference

In this section, we study non-asymptotic inference and optimization for stationary points of a smooth non-convex population landscape $L(\theta)$, via our proposed MasGrad method.

5.1. Inference and optimization for stationary points

First we state a theorem that quantifies how well our proposed MasGrad method achieves both the inference and the optimization goal.

Theorem 4 (MasGrad: non-convex). Let $L(w): \mathbb{R}^p \to \mathbb{R}$ be a smooth function. Recall that $\mathbf{b}(w) = \nabla L(w)$, and $\mathbf{H}(w)$ is the Hessian matrix of L. $\mathbf{V}(w) \in \mathbb{R}^{p \times p}$ is a positive definite matrix. Assume that

$$\gamma \triangleq \max_{v,w} \lambda_{\max}\{\mathbf{V}(w)^{-1/2}\mathbf{H}(v)\mathbf{V}(w)^{-1/2}\} > 0.$$

Consider the MasGrad updates θ_t in equation (10) with step size $\eta = 1/\gamma$, and the corresponding discretized diffusion ξ_t,

$$\xi_{t+1} = \xi_t - \eta \mathbf{V}(\xi_t)^{-1}\mathbf{b}(\xi_t) + \surd(2\beta^{-1}\eta)\mathbf{g}_t, \qquad \beta = 2n/\eta.$$

Then for any precision $\epsilon, \delta > 0$, we can choose

$$\begin{aligned} T &= \frac{2\gamma\{L(\theta_0) - \min_\theta L(\theta)\} + p\delta^2}{\epsilon^2}(\max_\theta \|\mathbf{V}(\theta)\| \vee 1), \\ n &= \frac{T}{\delta^2}, \end{aligned} \tag{21}$$

such that

(a) $D_{\mathrm{TV}}\{\mu(\theta_t, t \in [T]), \mu(\xi_t, t \in [T])\} \leqslant O_\delta(\delta)$ and
(b) $\mathbb{E}[\min_{t \leqslant T} \|\nabla L(\theta_t)\|] \leqslant \epsilon$ and $\mathbb{E}[\min_{t \leqslant T} \|\nabla L(\xi_t)\|] \leqslant \epsilon$,

with in total $O_{\epsilon,\delta}(\epsilon^{-4}\delta^{-2})$ independent data samples.

Remark 6. We would like to contrast the optimization part of theorem 4 with the sample complexity result of classic SGD. To obtain an ϵ-stationary point w such that in expectation $\|\nabla L(w)\| \leqslant \epsilon$, SGD needs $O_\epsilon(\epsilon^{-4})$ iterations for non-convex smooth functions (with step size $\eta_t = \min\{1/\gamma, 1/\surd t\}$). Here we show that, we can achieve this accuracy with the same dependence on ϵ with MasGrad, while being able to make statistical inference at the same time. And the additional price that we pay for δ-closeness in distribution for statistical inference is a factor of δ^{-2}.

The result can also be compared with theorem 2 (the strongly convex case). In both cases, statistically, we have shown that the discretized diffusion ξ_t tracks the non-asymptotic distribution of MasGrad θ_t, as long as the data-generating process satisfies conditions like weak moment and bounded entropic distance to Gaussian. The distribution of ξ_t is universal regardless of the specific data-generating distribution and depends only on the moments $\mathbf{V}(\theta)^{-1}\mathbf{b}(\theta)$. In terms of optimization, to obtain an ϵ-minimizer, the discretized diffusion approximation to MasGrad—with the proper step size η, and inverse temperature $\beta = 2n/\eta$—achieves the acceleration in the strongly convex case and enjoys the same dependence on ϵ as SGD in the non-convex case in terms of sample complexity.

5.2. Why local inference

For a general non-convex landscape, we shall discuss why we focus on inference about local optima, or more precisely stationary points. Our theorem 4 can be read as, within a reasonable number of steps, MasGrad converges to a population stationary point, and the distribution is well described by the discretized Langevin diffusion. One can argue that the random perturbation that is introduced by the isotropic Gaussian noise in Langevin diffusion makes the process difficult to converge to a typical saddle point. Therefore, intuitively, MasGrad will converge to a distribution that is well concentrated near a certain local optimum (depending on the initialization) as the temperature parameter $\beta^{-1}=\eta/2n$ is small. In this asymptotic low temperature regime, the Eyring–Kramer law states that the transition time from one local optimum to another local optimum, or the exit time from a certain local optimum, is very long—roughly $\exp(\beta h)$ where h is the depth of the basin of the local optimum (Bovier *et al.*, 2004; Tzen *et al.*, 2018). Therefore, a reasonable and tangible goal is to establish statistical inference for population local optima, for a particular initialization.

6. Estimation and computation of MasGrad direction

We address in this section how to estimate and approximate efficiently the MasGrad direction $\mathbf{V}(\theta)^{-1}\mathbf{b}(\theta)$ at a current parameter location θ. The estimation part involves a plug-in approach relying on the theory of self-normalized processes (Peña *et al.*, 2008). For efficient computation of the preconditioning matrix, we devise a fast iterative algorithm to approximate directly the root of the inverse covariance matrix, which in a way resembles the advantage of quasi-Newton methods (Wright and Nocedal, 1999), however, with noticeable differences. The quasi-Newton methods approximate a Hessian with first-order information, whereas MasGrad uses stochastic gradient information to approximate the root of the inverse covariance matrix as preconditioning. In this section we deliberately state all propositions working with general sample covariance matrix $\hat{\Sigma}$ with dimension d, to emphasize that the results extend beyond the discussions for MasGrad.

6.1. Statistical estimation and self-normalized processes

Recall that $\mathbf{V}(\theta)$ is the matrix root of the covariance. We estimate the moment-adjusted gradient direction $\mathbf{V}(\theta)^{-1}\mathbf{b}(\theta)$ at current location θ, based on a minibatch of size n. This section concerns this estimation part, borrowing tools from self-normalized processes. Define the sample estimates based on IID data z_i as

$$\hat{\mathbf{b}}(\theta)\triangleq\frac{1}{n}\sum_{i=1}^{n}\nabla_{\theta}l(\theta,z_i),$$

$$\hat{\mathbf{\Sigma}}(\theta)\triangleq\frac{1}{n-1}\sum_{i=1}^{n}[\{\nabla_{\theta}l(\theta,z_i)-\hat{\mathbf{b}}(\theta)\}\otimes\{\nabla_{\theta}l(\theta,z_i)-\hat{\mathbf{b}}(\theta)\}]$$

and $\hat{\mathbf{V}}(\theta)$ satisfies $\hat{\mathbf{V}}(\theta)\hat{\mathbf{V}}(\theta)^{\mathrm{T}}=\hat{\mathbf{\Sigma}}(\theta)$; we shall show that the plug-in approach $\hat{\mathbf{V}}(\theta)^{-1}\hat{\mathbf{b}}(\theta)$ estimates the population moment-adjusted gradient direction $\mathbf{V}(\theta)^{-1}\mathbf{b}(\theta)$ consistently at a parametric rate, in the fixed dimension setting.

Proposition 3 (connection to self-normalized processes). Consider $\{x_i\in\mathbb{R}^d, 1\leqslant i\leqslant n\}$ IID with mean μ, $\bar{x}$ and $\hat{\Sigma}$ the sample mean vector and sample covariance. Consider $d\ll n$ and $\hat{\Sigma}$ invertible. Denote the centred moments

$$S_n\triangleq\sum_{i=1}^{n}(x_i-\mu),$$

$$V_n^2\triangleq\sum_{i=1}^{n}\{(x_i-\mu)\otimes(x_i-\mu)\}$$

and the multivariate self-normalized process

$$M_n \triangleq V_n^{-1} S_n \in \mathbb{R}^d.$$

Then there exists $\hat{V}$, which satisfies $\hat{V}\hat{V}^{\mathrm{T}} = \hat{\Sigma}$ such that

$$\surd n \hat{V}^{-1}(\bar{x}-\mu) = M_n \sqrt{\left(\frac{n-1}{n-\|M_n\|^2}\right)}.$$

Remark 7. In the case $d=1$, proposition 3 reduces to a standard result in Peña *et al.* (2008). In our matrix version, the proof relies on the Sherman–Morrison–Woodbury matrix identity, together with a rank 1 update formula for matrix roots that we derive in lemma B.4 in the on-line appendix B. Recall the law of the iterated logarithm on the norm of a self-normalized process $\|M_n\|^2 \sim \log\{\log(n)\}$ (theorem 14.11 in Peña *et al.* (2008)), in the case when the dimension is fixed; a direct application of the above formula implies that

$$\hat{\mathbf{V}}(\theta)^{-1}\{\hat{\mathbf{b}}(\theta)-\mathbf{b}(\theta)\} = \frac{1}{\surd n} M_n \sqrt{\left(\frac{n-1}{n-\|M_n\|^2}\right)} = \frac{1+O_{\mathbf{p}}[\log\{\log(n)\}/n]}{\surd n} M_n,$$

where M_n is a self-normalized process with asymptotic distribution $\mathcal{N}(0, I_p)$. By lemma B.2 in appendix B, when $p \ll n$, the following approximation holds

$$\hat{\mathbf{V}}(\theta)^{-1}\hat{\mathbf{b}}(\theta) - \mathbf{V}(\theta)^{-1}\mathbf{b}(\theta) = \overbrace{\hat{\mathbf{V}}(\theta)^{-1}\{\hat{\mathbf{b}}(\theta)-\mathbf{b}(\theta)\}}^{\text{self-normalized processes}} + O_{\mathbf{p}}\left[\sqrt{\left\{\frac{p\log(n)}{n}\right\}}\right],$$

where the approximation is with respect to l_2-norm. Altogether, this implies that we can estimate $\mathbf{V}(\theta)^{-1}\mathbf{b}(\theta)$ consistently in the fixed dimension p and large n setting.

6.2. Efficient computation via direct rank 1 updates

In this section we devise a fast iterative formula for calculating $\hat{\mathbf{V}}(\theta)^{-1}$ directly via rank 1 updates.

Recall the brute force approach of calculating $\hat{\mathbf{\Sigma}}(\theta)$ first and then solving for the inverse root $\hat{\mathbf{V}}(\theta)^{-1}$ involves $O(np^2+p^3)$ complexity in the computation. Instead, we shall provide an algorithm that approximates $\hat{\mathbf{V}}(\theta)^{-1}$ directly through iterative rank 1 updates that is only $O(np^2)$ in complexity, utilizing the fact that the sample covariance is a finite sum of rank 1 matrices. To the best of our knowledge, this direct approach of calculating the root of the inverse covariance matrix is new.

Proposition 4 (iterative rank 1 updates of matrix inverse root). Initialize $H_0 = I_d$, and define the recursive rank 1 updates for the matrix inverse root, for $v_i \in \mathbb{R}^d$:

$$H_{i+1} = H_i - \frac{1}{\alpha_i} H_i v_{i+1} v_{i+1}^{\mathrm{T}} H_i^{\mathrm{T}} H_i \tag{22}$$

with

$$\alpha_i \triangleq \{1+\surd(1+v_{i+1}^{\mathrm{T}} H_i^{\mathrm{T}} H_i v_{i+1})\}\surd(1+v_{i+1}^{\mathrm{T}} H_i^{\mathrm{T}} H_i v_{i+1}) \in \mathbb{R}.$$

Then, for all n, H_n is the matrix inverse root of $I_d + \Sigma_{i=1}^n v_i \otimes v_i$. In other words, define $V_n =^{\triangle} H_n^{-1}$; then $V_n V_n^{\mathrm{T}} = I_d + \Sigma_{i=1}^n v_i \otimes v_i$.

Remark 8. One can directly apply the above result to evaluate $\hat{\mathbf{V}}(\theta)^{-1}$ efficiently. Define $v_i = \nabla_\theta l(\theta, \mathbf{z}_i) - \hat{\mathbf{b}}(\theta)$; we can use equation (22) in proposition 4 for fast iterative calculations, and

$$\{\surd(n-1)H_n\}^{-1}\{\surd(n-1)H_n^{\mathrm{T}}\}^{-1} = \frac{1}{n-1} I_d + \hat{\mathbf{\Sigma}} \approx \hat{\mathbf{\Sigma}}.$$

Therefore $\hat{\mathbf{V}}(\theta)^{-1}$ is approximated by $\surd(n-1)H_n$. We remark that the quality of the approximation depends on the spectral decay of the true covariance Σ.

For each iteration, the computational complexity for equation (22) is $4d^2$, with some careful design in calculation: it takes d^2 operations to calculate $H_i v_{i+1} \in \mathbb{R}^d$, then an additional d^2 to calculate $(H_i v_{i+1})^{\mathrm{T}} H_i \in \mathbb{R}^d$, another d^2 operations for multiplication of rank 1 vectors $H_i v_{i+1} \times (H_i v_{i+1})^{\mathrm{T}} H_i$ and finally d^2 operations for matrix addition. Hence, the total complexity is $O(nd^2)$ (for MasGrad, simply substitute $d=p$).

6.3. Optimal updates for on-line least squares

In the case of a least squares loss $l(\theta, z) = \frac{1}{2}(y - x^{\mathrm{T}}\theta)^2$, we offer a simple and efficient on-line rule for estimating $\mathbf{V}(\theta)$ without any loss of accuracy compared with offline counterparts. This is based on the fact that the data points z_i and the parameter θ can be 'decoupled' in least squares. To show this, first write the covariance as

$$\begin{aligned}\mathbf{V}(\theta)^2 &= \mathrm{cov}\{(\mathbf{y} - \mathbf{x}^{\mathrm{T}}\theta)\mathbf{x}\}\\ &= \mathrm{cov}(\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta) + \mathrm{cov}(\mathbf{y}\mathbf{x}) - 2\,\mathrm{cov}(\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta, \mathbf{x}\mathbf{y}).\end{aligned}$$

To estimate $\mathrm{cov}(\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta)$ efficiently in an on-line fashion, we observe that

$$\mathrm{cov}(\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta) = \mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta\theta^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}] - \mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta](\mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta])^{\mathrm{T}}. \tag{23}$$

Recalling that '$\otimes_{\mathrm{K}}$' denotes the Kronecker product and letting $\mathrm{vec}(X)$ be the vector that is formed by stacking the columns of X into a single column, we express $\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta\theta^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}$ as

$$\mathrm{vec}(\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta\theta^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}) = \{(\mathbf{x}\mathbf{x}^{\mathrm{T}}) \otimes_{\mathrm{K}} (\mathbf{x}\mathbf{x}^{\mathrm{T}})\}(\theta \otimes_{\mathrm{K}} \theta).$$

This expression shows that

$$\mathrm{vec}(\mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta\theta^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}]) = \mathbb{E}_{\mathbf{z}\sim P}[(\mathbf{x}\mathbf{x}^{\mathrm{T}}) \otimes_{\mathrm{K}} (\mathbf{x}\mathbf{x}^{\mathrm{T}})](\theta \otimes_{\mathrm{K}} \theta).$$

Accordingly, we can simply keep track of $\Sigma_{i=1}^{t}\{(\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}) \otimes_{\mathrm{K}} (\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}})\}$ in the on-line setting and estimate $\mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta\theta^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ through mapping the vector

$$\frac{\sum_{i=1}^{t}\{(\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}) \otimes_{\mathrm{K}} (\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}})\}}{t}(\theta \otimes_{\mathrm{K}} \theta)$$

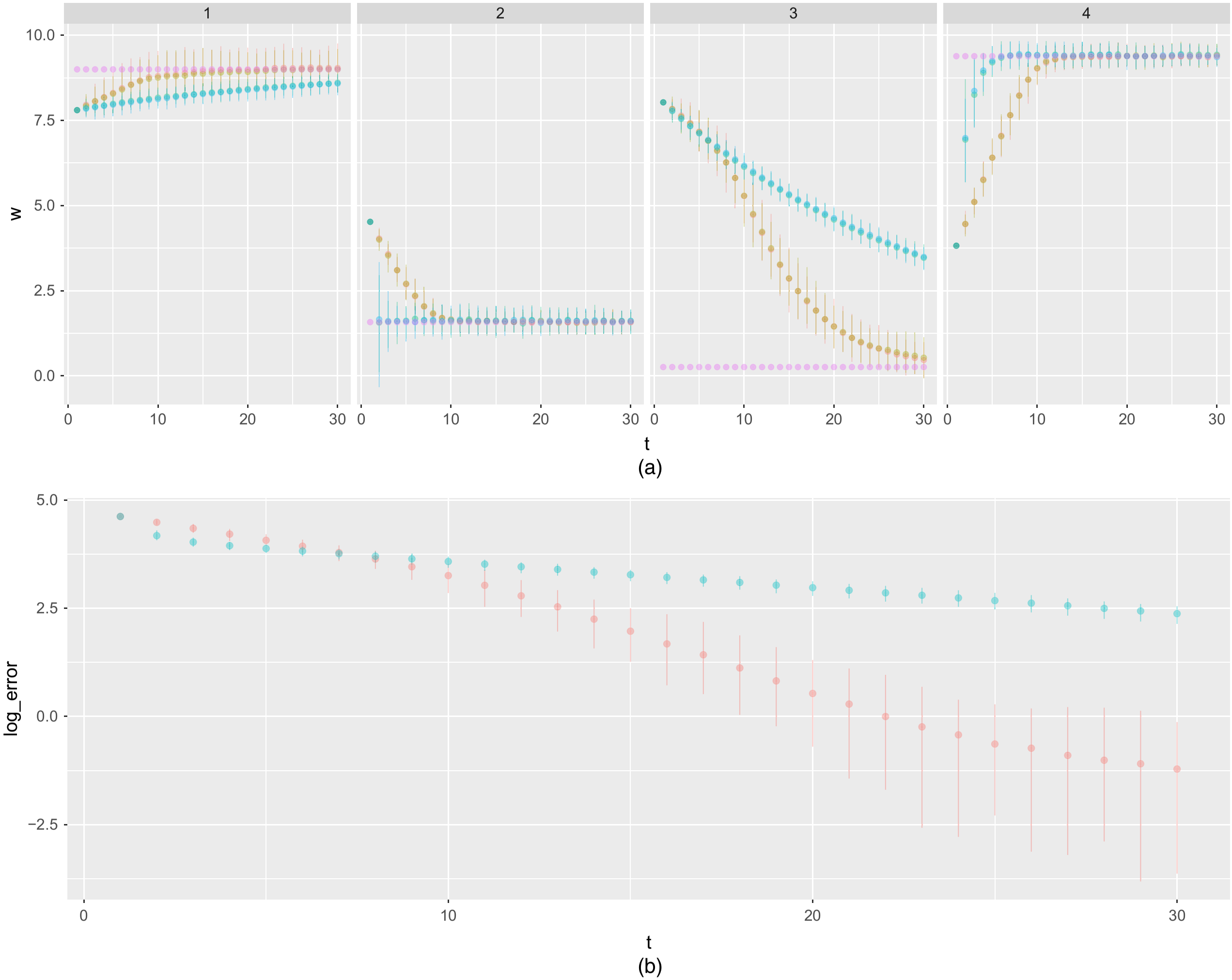
to its associated matrix. It remains to estimate $\mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta]$ in equation (23). Recognizing that $\mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta] = \mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}]\theta$, this can be done by simply recording the sum $\mathbf{x}_1\mathbf{x}_1^{\mathrm{T}} + \ldots + \mathbf{x}_t\mathbf{x}_t^{\mathrm{T}}$ in an on-line manner and replacing $\mathbb{E}_{\mathbf{z}\sim P}[\mathbf{x}\mathbf{x}^{\mathrm{T}}]$ by the average $(\mathbf{x}_1\mathbf{x}_1^{\mathrm{T}} + \ldots + \mathbf{x}_t\mathbf{x}_t^{\mathrm{T}})/t$. Likewise, $\mathrm{cov}(\mathbf{y}\mathbf{x})$ and $\mathrm{cov}(\mathbf{x}\mathbf{x}^{\mathrm{T}}\theta, \mathbf{x}\mathbf{y})$ can be estimated in the on-line setting regardless of a varying θ. We omit this part for brevity.

7. Numerical experiments

In this section we present results for numerical experiments. Full details of the experiments are deferred to the on-line appendix C.

7.1. Linear models

The first numerical example is simple linear regression, as in Fig. 1. Here we generate two plots as a proof of concept. Fig. 1(a) summarizes the trajectory of several methods for inference—



(a)

(b)

Fig. 1. Linear regression: ●, MASG; ●, diff_MASG ((a)); ●, diff_MASG ((b)) ●, SGD; ●, diff_SGD; ●, truth

our proposed *MasGrad*, the discretized diffusion approximation *diff_MasGrad*, as well as the classical *SGD*, and the diffusion approximation *diff_SGD*—with the confidence intervals (95% coverage) at each time step t. In this convex setting, we can solve for the global optimum, which is labelled as the *truth*. Here the minibatch size is $n=50$. We run 100 independent chains to calculate the confidence intervals at each step. We look at the low dimensional case $p=4$, and the four subfigures (in Fig. 1(a)) each correspond to one co-ordinate of the parameter w_i, $i\in[p]$. The x-axis is t, the evolution time, and the y-axis is the value of the parameter w. We remark that MasGrad and diff_MasGrad are pathwise close in terms of distribution, which verifies our statistical theory in theorem 1. This also holds for *GD* and *diff_GD*. We remark that, in this simulation, the condition number of the empirical Gram matrix is 30.98, and the first and third co-ordinates have very small population eigenvalues, which explains why in those co-ordinates MasGrad has significant acceleration compared with SGD as shown in Fig. 1. To be fair, at each time step, both MasGrad and SGD sample the same amount of data, and the step size is chosen as in theorem 3. All four chains start with the same random initialization.

To examine the optimization side of the story, we plot the logarithm of the l_2-error according to time t, for diff_MasGrad and diff_SGD, in Fig. 1(b). We remark that the error bar quantifies the confidence interval for the log-error. In theory, we should expect that the slope of MasGrad is twice that of the slope of SGD. In simulation, it seems that the acceleration is slightly better than what the theory predicts. We remark that compared with gradient descent, in which different co-ordinates make uneven progress (fast progress in the second and fourth co-ordinates, but slow in the others), MasGrad adaptively adjusts the relative step size on each co-ordinate for synchronized progress. This effect has also been observed in AdaGrad and natural gradient descent.

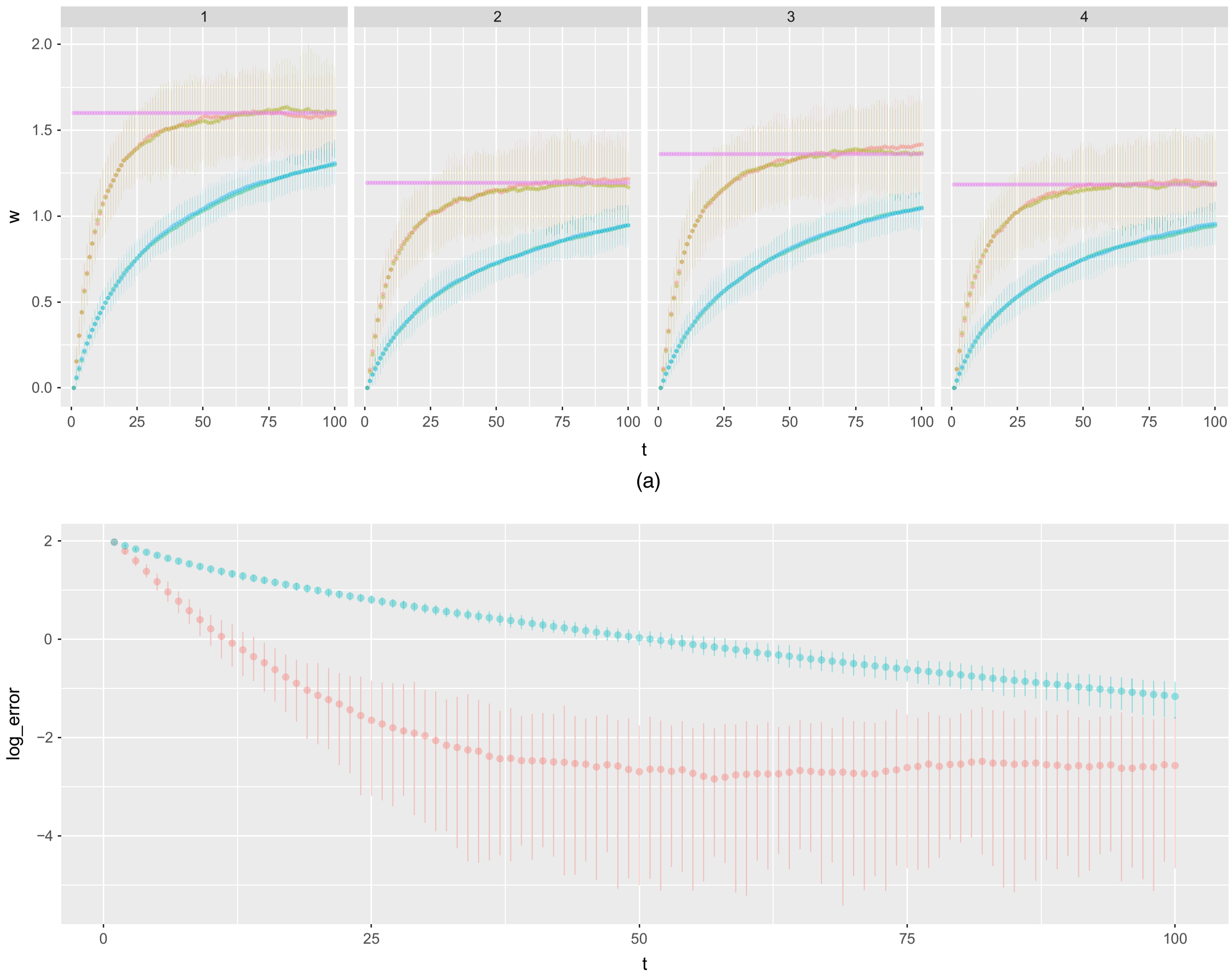
7.2. Logistic model

Fig. 2 illustrates the acceleration for inference in logistic regression. Fig. 2 should be read the same way as in the linear case. In this case, we sample a much larger number of samples ($N=500$) and then use the GLMs package in R (R Development Core Team, 2012) to fit the global optimum. For MasGrad and SGD, we generate bootstrap subsamples ($n=25$) to evaluate stochastic descents at each iteration. Again, we run 100 independent chains to calculate the confidence interval at each step. In this case, there is no theoretically optimal way of choosing the step size, so we choose the same step size ($\eta=0.2$) for both MasGrad and SGD.

Statistically, MasGrad and diff_MasGrad are close in distribution when $t<100$, and they both reach a stationary distribution after around 50 steps, simultaneously for all $p=4$ co-ordinates. Then the distribution fluctuates around stationarity. However, GD and diff_GD make much slower progress, and they fail to reach the global optimum in 100 steps. For optimization, empirically the acceleration in the log-error plot seems to be better than what the theoretical results predict. We remark that the confidence intervals are on the scale of log-error; therefore, it is negative skewed.

7.3. Gaussian mixture

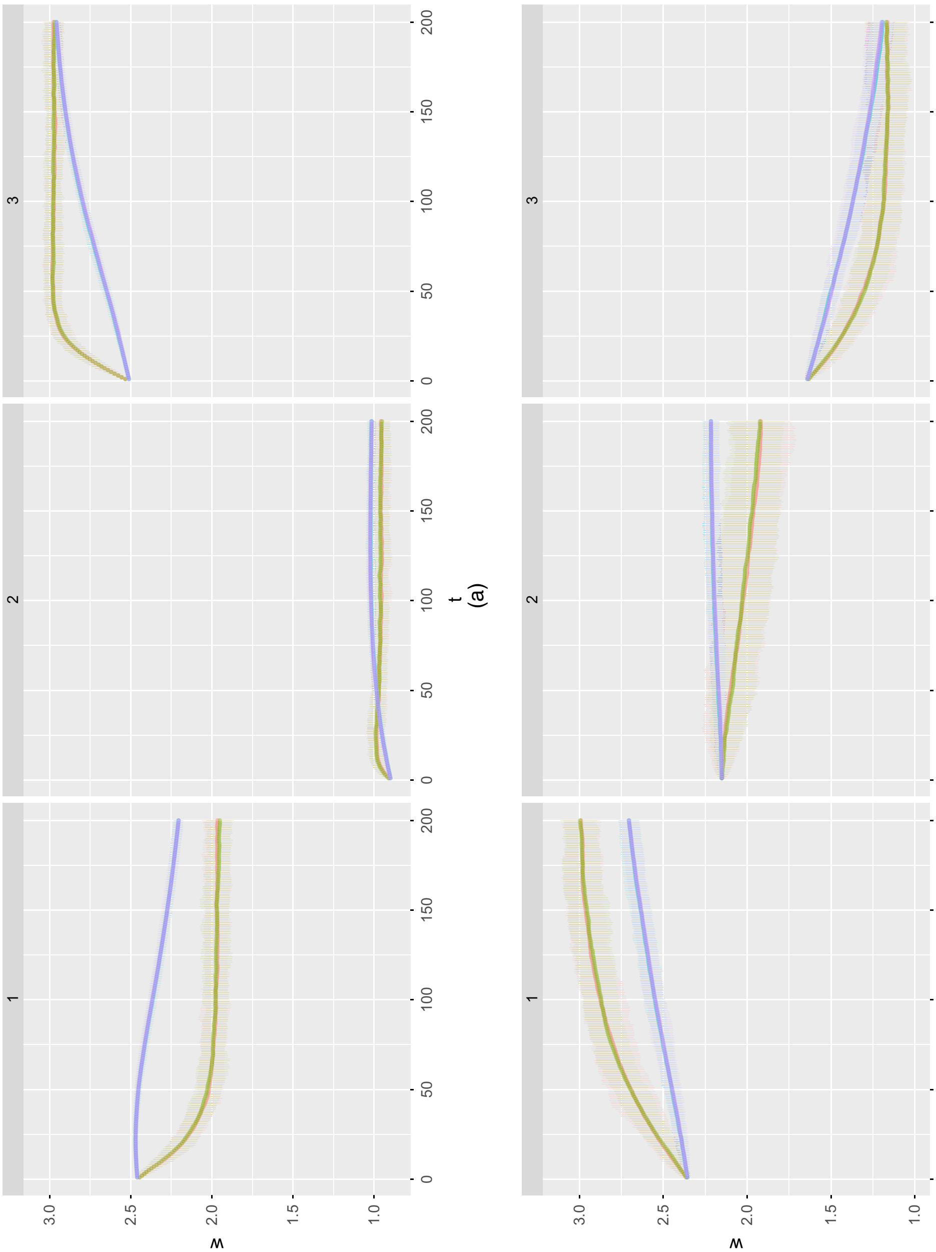
Here we showcase inference via MasGrad for the non-convex case, using the Gaussian mixture model. We shall consider a simple setting: the data $z_i\in\mathbb{R}^n$, $1\leqslant i\leqslant[N]$, generated from a mixture of p Gaussian distributions, with mean $(\theta_1,\theta_2,\ldots,\theta_p)=^{\triangle}\theta$ and variance σ^2. The goal is to infer the unknown mean vector $\theta\in\mathbb{R}^p$. The problem is non-convex because of the mixture nature: the maximum likelihood is multimodal, as we can shuffle the co-ordinates of θ to obtain the equivalent class of local optima.



(a)

(b)

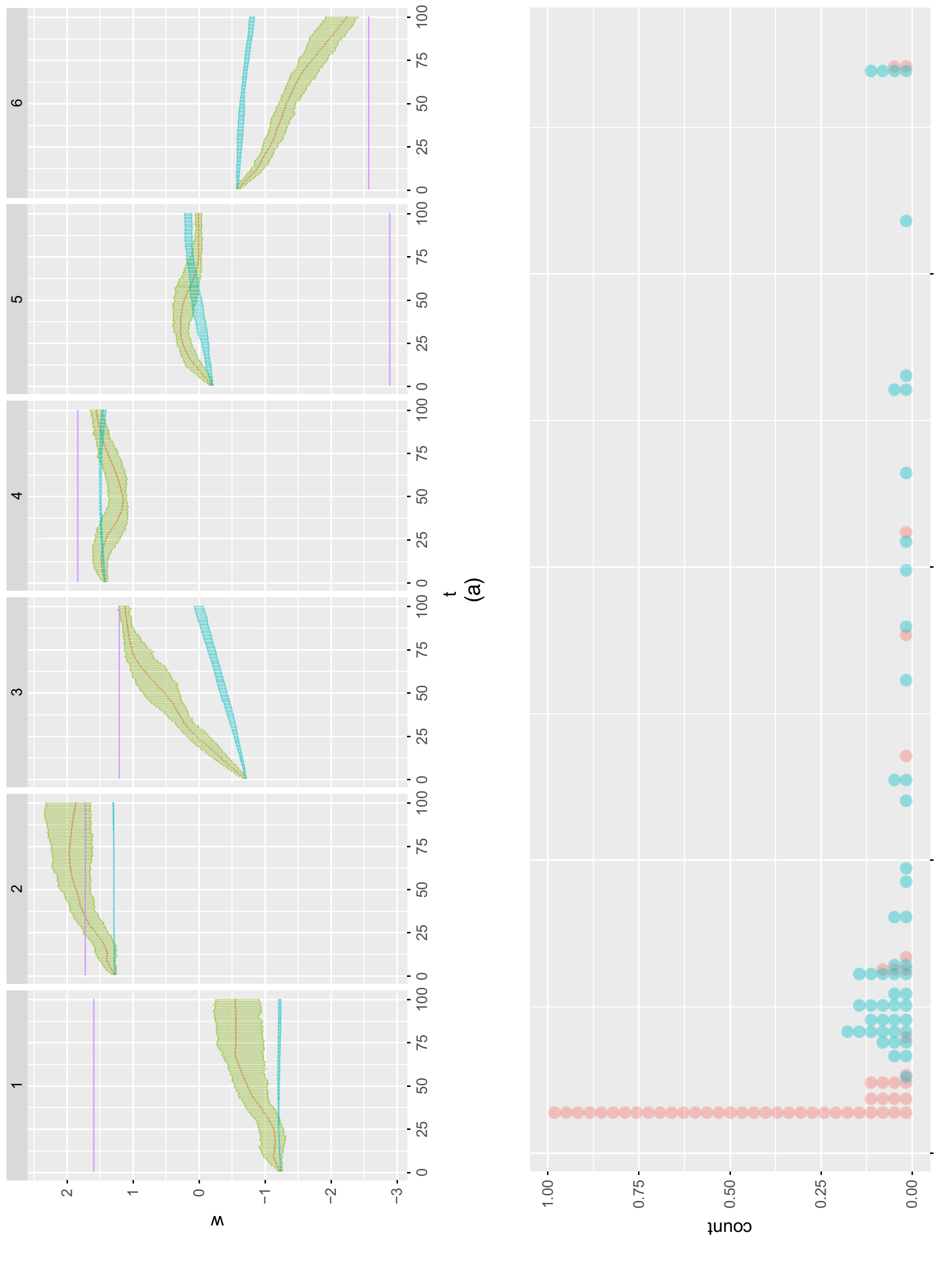
Fig. 2. Logistic regression: ●, MASG; ●, diff_MASG ((a)); ●, diff_MASG ((b)); ●, SGD; ●, diff_SGD; ●, truth



(a)

(b)

Fig. 3. Gaussian mixture: ●, MASG; ●, diff_MASG; ●, SGD; ●, diff_SGD



(a)

(b)

Fig. 4. Shallow neural nets: —, GD_MA ((a)); —, diff_MASG ((a)); —, SGD ((a)); —, param ((a)); ●, diff_MASG ((b)); ●, SGD ((b))

Fig. 3 illustrates the acceleration for inference in the Gaussian mixture model. Here we run two simulations, according to the difficulty (or separability) of the problem defined as the signal-to-noise ratio

$$\mathrm{SNR} \triangleq \min_{i \neq j} |\theta_i - \theta_j|/\sigma.$$

Fig. 3(a) is for the easy case with $\mathrm{SNR} = 3.3$ and Fig. 3(b) is for the difficult case with $\mathrm{SNR} = 1$. In both simulations, $\theta = (1, 2, 3) \in \mathbb{R}^3$, and we choose a random initial point to start the chains. The plot is presented as before. At each iteration, we subsample $n = 20$ data points to calculate the direction of descent, and the step size is fixed to be $\eta = 0.05$. We remark that there are many population local optima (at least $3! = 6$), and both MasGrad and diff_MasGrad seem to be able to find a good local optimum relatively quickly (which concentrates near a permutation of 1, 2, 3 for each co-ordinate), compared with SGD and diff_SGD. The acceleration effect in both cases is apparent. Again, we want to emphasize that the convergence for each co-ordinate in MasGrad seems to happen around the same number of iterations, which is not true for SGD.

7.4. Shallow neural networks

We also run MasGrad on a two-layer rectified linear unit neural network, as a proof of concept for non-convex models. Define the rectified linear unit activation $\sigma(x) = \max(x, 0)$; a two-layer neural network (with k hidden units) represents a function

$$f_w(x) = \sigma\{W_2\, \sigma(W_1 x)\}, \qquad x \in \mathbb{R}^d, \quad w = \{W_1 \in \mathbb{R}^{k \times d}, W_2 \in \mathbb{R}^{1 \times k}\}.$$

In our experiment, we work with the square loss $l\{w, (x, y)\} = \frac{1}{2}\{y - f_w(x)\}^2$. The gradients can be calculated through back-propagation. In this case, it is more difficult to calculate the global optimum; instead, to compare diff_MasGrad and SGD, we run 50 experiments with random initializations to explore the population landscape.

For each experiment (as illustrated in Fig. 4(a)), we randomly initialize the weights by using standard Gaussian distributions. As usual, we run 100 independent chains with the same initial points for diff_MasGrad and SGD to calculate the confidence interval. As expected, the distribution is quite non-Gaussian (for instance, in co-ordinate 2 and 6). We run the chain for 100 steps and then evaluate the population loss function for the two methods. Out of the 50 experiments, $45/50 = 90\%$ of the time the population loss that is returned by diff_MasGrad is much smaller than that of SGD. Fig. 4(b) plots the histogram (a dot plot using `ggplot2` (Wickham, 2009)) of the population error (test accuracy). Empirically, diff_MasGrad seems to converge to 'better' local optima most of the time. There could be several explanations: first, MasGrad uses better local geometry (similar to a natural gradient) so it induces better implicit regularization; second, MasGrad as an optimization method accelerates the chain so that it mixes to a local optimum faster, compared with SGD which may not yet converge within a certain time budget.

8. Further discussions

We discuss more about $\mathbf{V}(\theta_t)$. In the fixed dimension setting, we can estimate the covariance matrix of the gradient $\nabla l(\theta, \mathbf{z})$ by using the empirical version with N independent samples, when N is large. Let us be more precise in this statement.

(a) When the population landscape is convex, then the global optima of $\hat{L}_N(\theta)$ and $L(\theta)$ are within $1/\sqrt{N}$. We can always treat $\hat{L}_N(\theta)$ as the population version and at each step we bootstrap subsamples of size n to evaluate the stochastic gradients, adjusted by using the empirical covariance $\hat{\mathbf{V}}_N$ calculated by using N data points. Intuitively, when $\eta < O(n/N)$ (so that $\beta > N$), we know that MasGrad will concentrate near the optimum of $\hat{L}_N(\theta)$ with better accuracy than $1/\sqrt{N}$.

(b) In the non-convex case, things become unclear. However, under stronger conditions such as strongly Morse (Mei *et al.*, 2016), i.e. when there is nice one-to-one correspondence between the stationary points of $\hat{L}_N(\theta)$ and $L(\theta)$, we may still use the bootstrap idea above with $\hat{\mathbf{V}}_N$.

(c) Computation of $\hat{\mathbf{V}}_N$ and its inverse could be burdensome; thus we may want to use the efficient rank 1 updates designed in Section 6.3, or to calculate a diagonalized version of $\hat{\mathbf{V}}_N$ as done in AdaGrad (Duchi *et al.*, 2011).

(d) To have fully rigorous non-asymptotic theory in the case where $\mathbf{V}$ is known, we may require involved tools from self-normalized processes (Peña *et al.*, 2008) to establish a similar version of entropic CLT for multivariate self-normalized processes, where we standardize $\hat{\mathbb{E}}_n[\nabla l(\theta, \mathbf{z})]$ by the empirical covariance matrix $\hat{\mathbf{V}}_n$ calculated on the basis of the same samples. To the best of our knowledge, this is an ambitious and challenging goal that is beyond the scope and focus of the current paper.

We conclude this section by discussing the connections between preconditioning methods and our moment adjusting method. Preconditioning considers performing a linear transformation $\xi = A^{-1}\theta$ on the original parameter space on θ. In other words, consider $\tilde{L}(\xi) =^{\triangle} L(A\xi)$, and perform the updates on ξ:

$$\xi_{t+1} = \xi_t - \eta\nabla_\xi \tilde{L}(\xi) = \xi_t - \eta A\mathbf{b}(A\xi_t) \Rightarrow \theta_{t+1} = \theta_t - \eta A^2\mathbf{b}(\theta_t).$$

Therefore, in the noiseless case, the moment adjusting method is equivalent to preconditioning when the moment matrix $\mathbf{V}(\theta)$ is a constant matrix with respect to θ. However, in Langevin diffusion when the isotropic Gaussian noise is presented, the connection becomes more subtle—as $\mathbf{V}^{-1}(\theta)\mathbf{b}(\theta)$ may not be the gradient vector field for any function. The moment adjusting idea motivated from standardizing noise in statistics is different from the preconditioning idea in optimization. We would also like to point out that a nice idea using Hessian information to speed up the Langevin diffusion for sampling from log-concave distributions has been considered in Dalalyan (2017). We remark that we use the moment matrix at the current point θ_t (time varying) instead of the optimal point θ_* (which is unknown). We also use the matrix root instead of the covariance matrix itself. In the case when the model is well specified and the loss function is chosen to be the negative log-likelihood, $V(\theta_*)$ is the root of the Fisher information matrix.

9. Supplemental materials

For brevity we have relegated further discussion of Langevin diffusion to the on-line appendix A, detailed proofs to appendix B and remaining details about experiments to appendix C.

Acknowledgements

The authors thank the Associate Editor and the referees for the constructive feedback that significantly improved the content and presentation of the paper.

Liang gratefully acknowledges support from the George C. Tiao Faculty Fellowship. Su gratefully acknowledges support from the National Science Foundation via grant CCF-1763314.

References

Agarwal, N., Bullins, B. and Hazan, E. (2017) Second-order stochastic optimization for machine learning in linear time. *J. Mach. Learn. Res.*, **18**, 4148–4187.

Amari, S.-I. (1998) Natural gradient works efficiently in learning. *Neurl Computn*, **10**, 251–276.

Amari, S.-I. (2012) *Differential-geometrical Methods in Statistics*, vol. 28. New York: Springer Science and Business Media.

Arjevani, Y. and Shamir, O. (2016) Oracle complexity of second-order methods for finite-sum problems. *Preprint arXiv:1611.04982*.

Barron, A. R. (1986) Entropy and the central limit theorem. *Ann. Probab.*, **14**, 336–342.

Becker, S. and Fadili, J. (2012) A quasi-Newton proximal splitting method. In *Advances in Neural Information Processing Systems*, pp. 2618–2626.

Berahas, A. S., Bollapragada, R. and Nocedal, J. (2017) An investigation of Newton-sketch and subsampled Newton methods. *Preprint arXiv:1705.06211*.

Bobkov, S. G., Chistyakov, G. P. and Götze, F. (2013) Rate of convergence and Edgeworth-type expansion in the entropic central limit theorem. *Ann. Probab.*, **41**, 2479–2512.

Bobkov, S. G., Chistyakov, G. P. and Götze, F. (2014) Berry–Esseen bounds in the entropic central limit theorem. *Probab. Theory Reltd Flds*, **159**, 435–478.

Bollapragada, R., Byrd, R. and Nocedal, J. (2016) Exact and inexact subsampled Newton methods for optimization. *Preprint arXiv:1609.08502*.

Bordes, A., Bottou, L. and Gallinari, P. (2009) Sgd-qn: careful quasi-Newton stochastic gradient descent. *J. Mach. Learn. Res.*, **10**, 1737–1754.

Bovier, A., Eckhoff, M., Gayrard, V. and Klein, M. (2004) Metastability in reversible diffusion processes i: Sharp asymptotics for capacities and exit times. *J. Eur. Math. Soc.*, **6**, 399–424.

Brosse, N., Durmus, A., Moulines, É. and Pereyra, M. (2017) Sampling from a log-concave distribution with compact support with proximal Langevin Monte Carlo. *Preprint arXiv:1705.08964*.

Bubeck, S., Eldan, R. and Lehec, J. (2015) Sampling from a log-concave distribution with projected Langevin Monte Carlo. *Preprint arXiv:1507.02564*.

Byrd, R. H., Hansen, S. L., Nocedal, J. and Singer, Y. (2016) A stochastic quasi-Newton method for large-scale optimization. *SIAM J. Optimizn*, **26**, 1008–1031.

Chen, X., Lee, J. D., Tong, X. T. and Zhang, Y. (2016) Statistical inference for model parameters in stochastic gradient descent. *Preprint arXiv:1610.08637*.

Cotter, A., Shamir, O., Srebro, N. and Sridharan, K. (2011) Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems*, pp. 1647–1655.

Dalalyan, A. S. (2017) Theoretical guarantees for approximate sampling from smooth and log-concave densities. *J. R. Statist. Soc.* B, **79**, 651–676.

Duchi, J., Hazan, E. and Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, **12**, 2121–2159.

Durmus, A., Moulines, E. and Pereyra, M. (2018) Efficient Bayesian computation by proximal Markov chain Monte Carlo: when Langevin meets Moreau. *SIAM J. Imgng Sci.*, **11**, 473–506.

Ghadimi, S. and Lan, G. (2012) Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: a generic algorithmic framework. *SIAM J. Optimizn*, **22**, 1469–1492.

Ghadimi, S. and Lan, G. (2016) Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Math. Program.*, **156**, 59–99.

Jofré A. and Thompson, P. (2017) On variance reduction for stochastic smooth convex optimization with multiplicative noise. *Preprint arXiv:1705.02969*.

Johnson, R. and Zhang, T. (2013) Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323.

Kiefer, J. and Wolfowitz, J. (1952) Stochastic estimation of the maximum of a regression function. *Ann. Math. Statist.*, **23**, 462–466

Li, T., Liu, L., Kyrillidis, A. and Caramanis, C. (2017) Statistical inference using SGD. *Preprint arXiv:1705.07477*.

Mandt, S., Hoffman, M. D. and Blei, D. M. (2017) Stochastic gradient descent as approximate bayesian inference. *Preprint arXiv:1704.04289*.

Martens, J. (2014) New insights and perspectives on the natural gradient method. *Preprint arXiv:1412.1193*.

McCullagh, P. (1984) Generalized linear models. *Eur. J. Oper. Res.*, **16**, 285–292.

Mei, S., Bai, Y. and Montanari, A. (2016) The landscape of empirical risk for non-convex losses. *Preprint arXiv:1607.06534*.

Mokhtari, A. and Ribeiro, A. (2015) Global convergence of online limited memory bfgs. *J. Mach. Learn. Res.*, **16**, 3151–3181.

Moritz, P., Nishihara, R. and Jordan, M. (2016) A linearly-convergent stochastic L-BFGS algorithm. In *Proc.19th Int. Conf. Artificial Intelligence and Statistics*, pp. 249–258.

Nesterov, Y. (1983) A method of solving a convex programming problem with convergence rate o (1/k2). *Sov. Math. Dokl.*, **27**, 372–376.

Nesterov, Y. (2013) *Introductory Lectures on Convex Optimization: a Basic Course*. New York: Springer Science and Business Media.

Peña, V. H., Lai, T. L. and Shao, Q.-M. (2008) *Self-normalized Processes: Limit Theory and Statistical Applications*. New York: Springer Science and Business Media.

Pennington, J., Socher, R. and Manning, C. (2014) Glove: global vectors for word representation. In *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 1532–1543.

Pilanci, M. and Wainwright, M. J. (2015) Newton sketch: a linear-time optimization algorithm with linear-quadratic convergence. *Preprint arXiv:1505.02250*.

Polyak, B. T. (1990) New stochastic approximation type procedures. *Autom. Telemkh*, **7**, 98–107.

Polyak, B. T. and Juditsky, A. B. (1992) Acceleration of stochastic approximation by averaging. *SIAM J. Control Optimizn*, **30**, 838–855.

Raginsky, M., Rakhlin, A. and Telgarsky, M. (2017) Non-convex learning via stochastic gradient Langevin dynamics: a nonasymptotic analysis. *Preprint arXiv:1702.03849*.

R Development Core Team (2012) *R: a Language and Environment for Statistical Computing*. Vienna: R Foundation for statistical Computing.

Robbins, H. and Monro, S. (1951) A stochastic approximation method. *Ann. Math. Statist.*, **22**, 400–407.

Roux, N. L., Schmidt, M. and Bach, F. R. (2012) A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pp. 2663–2671.

Ruppert, D. (1988) Efficient estimations from a slowly convergent Robbins–Monro process. *Technical Report*. Department of Operations Research and Industrial Engineering, Cornell University, Ithaca.

Schraudolph, N. N. (2002) Fast curvature matrix-vector products for second-order gradient descent. *Neurl Computn*, **14**, 1723–1738.

Schraudolph, N. N., Yu, J. and Günter, S. (2007) A stochastic quasi-Newton method for online convex optimization. In *Proc. 12th Int. Conf. Artificial Intelligence and Statistics*, pp. 436–443.

Toulis, P. and Airoldi, E. M. (2017) Asymptotic and finite-sample properties of estimators based on stochastic gradients. *Ann. Statist.*, **45**, 1694–1727.

Tzen, B., Liang, T. and Raginsky, M. (2018) Local optimality and generalization guarantees for the Langevin algorithm via empirical metastability. *Proc. Mach. Learn. Res.*, **75**, 857–875.

Wang, X., Ma, S., Goldfarb, D. and Liu, W. (2017) Stochastic quasi-Newton methods for non-convex stochastic optimization. *SIAM J. Optimizn*, **27**, 927–956.

Welling, M. and Teh, Y. W. (2011) Bayesian learning via stochastic gradient Langevin dynamics. In *Proc. 28th Int. Conf. Machine Learnng*, pp. 681–688.

Wickham, H. (1999) *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer.

Wright, S. and Nocedal, J. (1999) *Numerical Optimization*, pp. 67–68. New York: Springer.

Xu, P., Yang, J., Roosta-Khorasani, F., Ré, C. and Mahoney, M. W. (2016) Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pp. 3000–3008.

Supporting information

Additional ‘supporting information’ may be found in the on-line version of this article:

‘Supplement to “Statistical inference for the population landscape via moment-adjusted stochastic gradients”’.