*The Annals of Statistics*
2017, Vol. 45, No. 4, 1694–1727
DOI: 10.1214/16-AOS1506

# ASYMPTOTIC AND FINITE-SAMPLE PROPERTIES OF ESTIMATORS BASED ON STOCHASTIC GRADIENTS[1]

BY PANOS TOULIS AND EDOARDO M. AIROLDI

*University of Chicago and Harvard University*

Stochastic gradient descent procedures have gained popularity for parameter estimation from large data sets. However, their statistical properties are not well understood, in theory. And in practice, avoiding numerical instability requires careful tuning of key parameters. Here, we introduce implicit stochastic gradient descent procedures, which involve parameter updates that are implicitly defined. Intuitively, implicit updates shrink standard stochastic gradient descent updates. The amount of shrinkage depends on the observed Fisher information matrix, which does not need to be explicitly computed; thus, implicit procedures increase stability without increasing the computational burden. Our theoretical analysis provides the first full characterization of the asymptotic behavior of both standard and implicit stochastic gradient descent-based estimators, including finite-sample error bounds. Importantly, analytical expressions for the variances of these stochastic gradient-based estimators reveal their exact loss of efficiency. We also develop new algorithms to compute implicit stochastic gradient descent-based estimators for generalized linear models, Cox proportional hazards, M-estimators, in practice, and perform extensive experiments. Our results suggest that implicit stochastic gradient descent procedures are poised to become a workhorse for approximate inference from large data sets.

**1. Introduction.** Parameter estimation by optimization of an objective function is a fundamental idea in statistics and machine learning [Fisher (1922), Hastie, Tibshirani and Friedman (2009), Lehmann and Casella (1998)]. However, classical procedures, such as Fisher scoring, the EM algorithm or iteratively reweighted least squares [Dempster, Laird and Rubin (1977), Fisher (1925), Green (1984)], do not scale to modern data sets with millions of data points and hundreds or thousands of parameters [National Research Council (2013)].

In particular, suppose we want to estimate the true parameter $\theta_\star \in \mathbb{R}^p$ of a distribution $f$ from $N$ i.i.d. data points $(X_i, Y_i)$ such that conditional on covariate $X_i \in \mathbb{R}^p$ outcome $Y_i \in \mathbb{R}^d$ is distributed according to $f(Y_i; X_i, \theta_\star)$. Such estimation problems often reduce to optimization problems. For instance, the maximum

Received September 2015; revised August 2016.

[1] Supported in part by NSF CAREER award IIS-1149662, ARO MURI award W911NF-11-1-0036, ONR YIP award N00014-14-1-0485, and the 2012 Google US/Canada Ph.D. Fellowship in Statistics.

*MSC2010 subject classifications.* 62L20, 62F10, 62L12, 62F12, 62F35.

*Key words and phrases.* Stochastic approximation, implicit updates, asymptotic variance, generalized linear models, Cox proportional hazards, M-estimation, maximum likelihood, exponential family, statistical efficiency, numerical stability.

likelihood estimator (MLE) is obtained by solving $\theta_N^{\mathrm{mle}} = \arg\max_\theta \sum_{i=1}^N \log f(Y_i; X_i, \theta)$. Classical optimization procedures, such as Newton–Raphson or Fisher scoring, have a runtime complexity that ranges between $\mathrm{O}(Np^{1+\varepsilon})$ and $\mathrm{O}(Np^{2+\varepsilon})$, in the best case and worst case, respectively [Lange (2010)]. Quasi-Newton (QN) procedures are the only viable alternative in practice because they have $\mathrm{O}(Np^2)$ complexity per iteration, or $\mathrm{O}(Np^{1+\varepsilon})$ in certain favorable cases [Hennig and Kiefel (2013)]. However, estimation from large data sets requires an even better runtime complexity that is roughly $\mathrm{O}(Np^{1-\varepsilon})$, that is, linear in data size $N$ but sublinear in parameter dimension $p$. The first requirement on $N$ is generally unavoidable because all data points carry information from the i.i.d. assumption. Sublinearity in $p$ is therefore critical.

Such requirements have recently generated interest in stochastic optimization procedures, especially those only relying on first-order information, that is, gradients. Perhaps the most widely popular procedure in this family is *stochastic gradient descent* (SGD), defined for $n = 1, 2, \ldots,$ as

$$\theta_n^{\mathrm{sgd}} = \theta_{n-1}^{\mathrm{sgd}} + \gamma_n C_n \nabla \log f\big(Y_n; X_n, \theta_{n-1}^{\mathrm{sgd}}\big), \tag{1}$$

where $\gamma_n > 0$ is the learning rate sequence, typically defined as $\gamma_n = \gamma_1 n^{-\gamma}$, $\gamma_1 > 0$ is the learning rate parameter, $\gamma \in (0.5, 1]$, and $C_n$ are $p \times p$ positive-definite matrices, also known as condition matrices.

Stochastic optimization procedures of this kind are special cases of stochastic approximation [Robbins and Monro (1951)], where the estimation problem is not formulated as an optimization problem but more generally as a characteristic equation. Early research considered a streaming data setting—akin to a superpopulation setting—where the characteristic equation is

$$\mathbb{E}\big(\nabla \log f(Y; X, \theta_\star) \mid X\big) = 0, \tag{2}$$

with the expectation being over the true conditional distribution of outcome $Y$ given covariate $X$. More recent research, largely in computer science and optimization, considers a finite N setting with characteristic equation

$$\mathbb{E}\big(\nabla \log f\big(Y; X, \theta_N^{\mathrm{mle}}\big)\big) = 0, \tag{3}$$

where the expectation is over the empirical distribution of $(X, Y)$ in the finite data set.[2] In both settings, SGD of equation (1) is well defined: in the finite population setting of equation (3) the data point $(X_n, Y_n)$ is a random sample with replacement from the finite data set; in the infinite population setting of equation (2) the data point $(X_n, Y_n)$ is simply the $n$th data point in the stream.

From a computational perspective, SGD in equation (1) is appealing because it avoids expensive matrix inversions, as in Newton–Raphson, and the log-likelihood

[2] If regularization is used, then equation (3) could approximate the maximum a posteriori estimate (MAP) instead of the MLE.

is evaluated at a single data point $(X_n, Y_n)$ and not on the entire data set. From a theoretical perspective, SGD in equation (1) converges, under suitable conditions, to $\theta_\infty^{\text{sgd}}$ where $\mathbb{E}(\log f(Y; X, \theta_\infty^{\text{sgd}}) \mid X) = 0$ [Benveniste, Métivier and Priouret (1990), Ljung, Pflug and Walk (1992), Borkar (2008)]. This condition can satisfy both equation (2) and equation (3), implying that SGD can be used on both finite and infinite population settings. For the rest of this paper, we assume an infinite population setting, as it is the most natural setting for stochastic approximations. The main difference between the data setting studied in the computer science and optimization literature and the infinite population setting we consider here is that we do not condition on the observed ordering of data points, but we condition on a random ordering instead. Moreover, most of the theoretical results presented in this paper for the infinite population setting can be applied to the finite population setting, where instead of estimating $\theta_\star$ we estimate, say, the MLE, or the MAP estimate if there is regularization.

In this paper, we introduce *implicit* stochastic gradient descent procedures—*implicit SGD* for short—defined as

$$\boldsymbol{\theta}_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n C_n \nabla \log f(Y_n; X_n, \boldsymbol{\theta}_n^{\text{im}}), \tag{4}$$

where $\gamma_n, C_n$ are defined as in standard SGD equation (1). Furthermore, we provide a theoretical analysis of estimators based on stochastic gradients, for both implicit and standard procedures. To distinguish the two procedures, we will refer to standard SGD in equation (1) as SGD with *explicit updates*, or *explicit SGD* for short, because the next iterate $\theta_n^{\text{sgd}}$ can be immediately computed given $\theta_{n-1}^{\text{sgd}}$ and the data point $(X_n, Y_n)$. In contrast, the update in equation (4) is *implicit* because the next iterate $\theta_n^{\text{im}}$ appears on both sides of the equation, where the iterate was typed in boldface to emphasize the fact.

1.1. *Illustrative example.* Here, we motivate the main results of this paper on the comparison between implicit and explicit SGD. Let $\theta_\star \in \mathbb{R}$ be the true parameter of a normal model with i.i.d. observations $Y_i | X_i \sim \mathcal{N}(X_i\theta_\star, \sigma^2)$, where the variance $\sigma^2$ is assumed known for simplicity. The log-likelihood is $\log f(Y_i; X_i, \theta) = -\frac{1}{2\sigma^2}(Y_i - X_i\theta)^2$, and the score function (i.e., gradient of log-likelihood) is given by $\nabla \log f(Y_i; X_i, \theta) = \frac{1}{\sigma^2}(Y_i - X_i\theta)X_i$. Let $X_i$ be distributed according to some unknown distribution with bounded second moment. Assume $\gamma_n = \gamma_1/n$, for some $\gamma_1 > 0$ as the learning rate. Then the explicit SGD procedure in equation (1) is

$$\begin{aligned} \theta_n^{\text{sgd}} &= \theta_{n-1}^{\text{sgd}} + \gamma_n(Y_n - \theta_{n-1}^{\text{sgd}} X_n)X_n \\ &= (1 - \gamma_n X_n^2)\theta_{n-1}^{\text{sgd}} + \gamma_n Y_n X_n. \end{aligned} \tag{5}$$

Procedure (5) is the least mean squares filter (LMS) in signal processing, also known as the Widrow–Hoff algorithm [Widrow and Hoff (1960)]. The implicit

SGD procedure can be derived in closed form in this problem using update in equation (4) as

$$\begin{aligned}\theta_n^{\text{im}} &= \theta_{n-1}^{\text{im}} + \gamma_n(Y_n - X_n\theta_n^{\text{im}})X_n \\ &= \frac{1}{1+\gamma_n X_n^2}\theta_{n-1}^{\text{im}} + \frac{\gamma_n}{1+\gamma_n X_n^2} Y_n X_n.\end{aligned} \tag{6}$$

The procedure defined by equation (6) is known as the normalized least mean squares filter (NLMS) in signal processing [Nagumo and Noda (1967)].

From equation (5), we see that it is crucial for explicit SGD to have a well-specified learning rate parameter $\gamma_1$. For instance, if $\gamma_1 X_1^2 \gg 1$ then $\theta_n^{\text{sgd}}$ will diverge to a value at the order of $2^{\gamma_1}/\sqrt{\gamma_1}$, before converging to the true value $\theta_\star$ (see Section 2.5, Lemma 2.1). In contrast, implicit SGD is more stable to misspecification of the learning rate parameter $\gamma_1$. For example, a very large $\gamma_1$ will not cause divergence as in explicit SGD, but it will simply put more weight on the $n$th observation $Y_n X_n$ than the previous iterate $\theta_{n-1}^{\text{im}}$. Assuming for simplicity $\theta_{n-1}^{\text{sgd}} = \theta_{n-1}^{\text{im}} = 0$, it also holds $\theta_n^{\text{im}} = \frac{1}{1+\gamma_n X_n^2}\theta_n^{\text{sgd}}$, showing that implicit SGD iterates are shrinked versions of explicit ones (see also Section 5).

Let $v^2 = \mathbb{E}(X^2)$, then by Theorem 2.2 the asymptotic variance of $\theta_n^{\text{im}}$ (and of $\theta_n^{\text{sgd}}$) satisfies $n\,\text{Var}(\theta_n^{\text{im}}) \to \gamma_1^2\sigma^2 v^2/(2\gamma_1 v^2 - 1)$ if $2\gamma_1 v^2 - 1 > 0$. Since $\gamma_1^2/(2\gamma_1 v^2 - 1) \geq 1/v^2$, it is best to set $\gamma_1 = 1/v^2$. In this case $n\,\text{Var}(\theta_n^{\text{im}}) \to \sigma^2/v^2$. Implicit SGD can thus be optimal by setting $\gamma_n = (\sum_{i=1}^n X_i^2)^{-1}$ in which case $\theta_n^{\text{sgd}}$ is exactly the OLS estimator, and $\theta_n^{\text{im}}$ is an approximate but more stable version of the OLS estimator. Thus, the implicit SGD estimator $\theta_n^{\text{im}}$ in equation (6) inherits the efficiency properties of $\theta_n^{\text{sgd}}$, with the added benefit of being stable over a wide range of learning rates. Overall, implicit SGD is a superior form of SGD.

1.2. *Related work.* Historically, the duo of explicit–implicit updates originate from the numerical methods introduced by Euler (ca. 1770) for solving ordinary differential equations [Hoffman and Frankel (2001)]. The explicit SGD procedure was first proposed by Sakrison (1965) as a recursive statistical estimation method and it is theoretically based on the stochastic approximation method of Robbins and Monro (1951). Statistical estimation with explicit SGD is a straightforward generalization of Sakrison's method and has recently attracted attention in the machine learning community as a fast learning method for large-scale problems [Zhang (2004), Bottou (2010), Toulis and Airoldi (2015a)]. Applications of explicit SGD procedures in massive data problems can be found in many diverse areas such as large-scale machine learning [Zhang (2004)], online EM algorithm [Cappé and Moulines (2009)], image analysis and deep learning [Dean et al. (2012)] and MCMC sampling [Welling and Teh (2011)].

The implicit SGD procedure is less known and not well understood. In optimization, implicit methods have recently attracted attention under the guise of proximal methods, such as mirror-descent [Nemirovsky and Yudin (1983)]. In fact, the implicit SGD update in equation (4) can be expressed as a proximal update:

$$\theta_n^{\mathrm{im}} = \arg\max_{\theta} \left\{ -\frac{1}{2\gamma_n} \|\theta - \theta_{n-1}^{\mathrm{im}}\|^2 + \log f(Y_n; X_n, \theta) \right\}. \tag{7}$$

From a Bayesian perspective, $\theta_n^{\mathrm{im}}$ is the posterior mode of a model with the standard multivariate normal $\mathcal{N}(\theta_{n-1}^{\mathrm{im}}, \gamma_n I)$ as the prior, and $\log f(Y_n; X_n, \theta)$ as the log-likelihood of $\theta$ for observation $(X_n, Y_n)$. Arguably, the normalized least mean squares (NLMS) filter [Nagumo and Noda (1967)], introduced in equation (6), was the first statistical model that used an implicit update as in equation (4), and was shown to be consistent and robust under excessive input noise [Slock (1993)]. From an optimization perspective, the update in equation (7) corresponds to a stochastic version of the proximal point algorithm by Rockafellar (1976), which has been generalized through the idea of splitting algorithms [Lions and Mercier (1979), Beck and Teboulle (2009), Singer and Duchi (2009)]; see also the comprehensive review of proximal methods in optimization by Parikh and Boyd (2013). Additional intuition of implicit methods has been provided by Krzysztof et al. (2007) and Nemirovski et al. (2008), who have argued that proximal methods can fit better in the geometry of the parameter space. Bertsekas (2011) derived the convergence rate of an implicit procedure similar to equation (4) on a fixed data set, and compared the rates between procedures that randomly sampled data $(X_n, Y_n)$ or simply cycled through them. Toulis, Airoldi and Rennie (2014) derived the asymptotic variance of $\theta_n^{\mathrm{im}}$ as estimator of $\theta_\star$ in the family of generalized linear models, and provided an algorithm to efficiently compute the implicit update of equation (4) in such models and in the simplified setting where $C_n = I$.

1.3. *Contributions.* Prior work on procedures similar to implicit SGD has considered mostly an optimization setting, in which the focus is on speed of convergence [for example, Bertsekas (2011)]. Instead, we focus on statistical efficiency, that is, the sampling variability of the estimator implied by implicit and explicit SGD procedures—the relevant analysis and the results of Theorem 2.1 and Theorem 2.2 are novel. Furthermore, our procedure, which we generalized in Toulis and Airoldi (2015b), is different than typical stochastic proximal gradient procedures [see, e.g., Duchi and Singer (2009), Rosasco, Villa and Vũ (2014)]. In such procedures, the parameter updates are obtained by combining a stochastic explicit update and a deterministic implicit update. In implicit SGD, there is a single stochastic implicit update, which prevents numerical instability.

With regard to theoretical contributions, the asymptotic statistical efficiency of SGD procedures (both explicit and implicit) derived in Theorem 2.2 is a key contribution of our work. Our analysis is in fact general enough that allowed us to derive the asymptotic efficiency of other popular stochastic optimization procedures, notably of AdaGrad [Duchi, Hazan and Singer (2011)] in equation (13) of our paper.

The asymptotic normality of implicit SGD in Theorem 2.4 is new and enables a novel comparison of explicit SGD and implicit SGD in terms of the normality of their iterates, which is also a clear point of departure from the typical optimization literature. The results in Section 2.5 are also new, and formalize the advantages of implicit SGD over explicit SGD in terms of numerical stability.

With regard to practical contributions, Algorithm 1 and its variants presented in the paper are a significant extension of our earlier work beyond first-order GLMs [Toulis, Airoldi and Rennie (2014), Algorithm 1]. The key contribution here is that these new algorithms make implicit SGD as simple to implement as standard explicit SGD, whenever the fixed-point computation of the implicit update is feasible. We provide extensive applications in Section 3 and experiments in Section 4 of implicit SGD compared to explicit SGD. Importantly, we developed a concrete implementation of implicit SGD through the `R` package `sgd` [Tran, Toulis and Airoldi (2015)] available at https://cran.r-project.org/web/packages/sgd/index.html to compare implicit SGD with state-of-art procedures, including R's `glm()` function (Fisher scoring), `biglm` package, the elastic net [Friedman, Hastie and Tibshirani (2010), `glmnet`], AdaGrad [Duchi, Hazan and Singer (2011)], Prox-SVRG [Xiao and Zhang (2014)], and Prox-SAG [Schmidt, Le Roux and Bach (2013)].

## 2. Theory.

The norm $\|\cdot\|$ denotes the $L_2$ norm. If a positive scalar sequence $a_n$ is nonincreasing and $a_n \to 0$, we write $a_n \downarrow 0$. For two positive scalar sequences $a_n, b_n$, equation $b_n = \mathrm{O}(a_n)$ denotes that $b_n$ is bounded above by $a_n$, that is, there exists a fixed $c > 0$ such that $b_n \leq c a_n$, for all $n$. Furthermore, $b_n = \mathrm{o}(a_n)$ denotes that $b_n / a_n \to 0$. Similarly, for a sequence of vectors (or matrices) $X_n$, we write $X_n = \mathrm{O}(a_n)$ to denote $\|X_n\| = \mathrm{O}(a_n)$, and write $X_n = \mathrm{o}(a_n)$ to denote $\|X_n\| = \mathrm{o}(a_n)$. For two positive definite matrices $A, B$, we write $A \prec B$ to express that $B - A$ is positive definite. The set of eigenvalues of a matrix $A$ is denoted by $\mathrm{eig}(A)$; thus, $A \succ 0$ if and only if $\lambda > 0$ for every $\lambda \in \mathrm{eig}(A)$.

ASSUMPTION 2.1. The explicit SGD procedure in equation (1) and the implicit SGD procedure in equation (4) operate under a combination of the following assumptions.

(a) The learning rate sequence $\{\gamma_n\}$ is defined as $\gamma_n = \gamma_1 n^{-\gamma}$, where $\gamma_1 > 0$ is the learning parameter, and $\gamma \in (0.5, 1]$.

(b) For the log-likelihood $\log f(Y; X, \theta)$ there exists function $\ell$ such that $\log f(Y; X, \theta) \equiv \ell(X^\intercal \theta; Y)$, which depends on $\theta$ only through the natural parameter $X^\intercal \theta$.

(c) Function $\ell$ is concave, twice differentiable almost surely w.r.t. natural parameter $X^\intercal \theta$ and Lipschitz with constant $L_0$ w.r.t. $\theta$.

(d) The observed Fisher information matrix $\hat{\mathcal{I}}_n(\theta) = -\nabla^2 \ell(X_n^\intercal \theta; Y_n)$ has nonvanishing trace, that is, there exists constant $b > 0$ such that $\mathrm{trace}(\hat{\mathcal{I}}_n(\theta)) \geq b$

almost surely, for all $\theta$. The Fisher information matrix, $\mathcal{I}(\theta_\star) = \mathbb{E}(\hat{\mathcal{I}}_n(\theta_\star))$, has minimum eigenvalue $\underline{\lambda_f} > 0$ and maximum eigenvalue $\overline{\lambda_f} < \infty$. Typical regularity conditions hold [Lehmann and Casella (1998), Theorem 5.1, page 463].

(e) Every condition matrix $C_n$ is a fixed positive-definite matrix, such that $C_n = C + \mathrm{O}(\gamma_n)$, where $||C|| = 1$, $C \succ 0$ and symmetric, and $C$ commutes with $\mathcal{I}(\theta_\star)$. For every $C_n$, $\min \operatorname{eig}(C_n) = \underline{\lambda_c} > 0$, and $\max \operatorname{eig}(C_n) = \overline{\lambda_c} < \infty$.

(f) Let $\Xi_n = \mathbb{E}(\nabla \log f(Y_n; X_n, \theta_\star) \nabla \log f(Y_n; X_n, \theta_\star)^\intercal \mid \mathcal{F}_{n-1})$, then $\|\Xi_n - \Xi\| = \mathrm{O}(1)$ for all $n$, and $\|\Xi_n - \Xi\| \to 0$, for a symmetric positive-definite $\Xi$. Let $\sigma_{n,s}^2 = \mathbb{E}(\mathbb{I}_{\|\xi_n(\theta_\star)\|^2 \geq s/\gamma_n} \|\xi_n(\theta_\star)\|^2)$, then for all $s > 0$, $\sum_{i=1}^n \sigma_{i,s}^2 = \mathrm{o}(n)$ if $\gamma = 1$, and $\sigma_{n,s}^2 = \mathrm{o}(1)$ otherwise.

REMARKS. Assumption 2.1(a) is typical in stochastic approximation as it implies that $\sum_i \gamma_i = \infty$ and $\sum_i \gamma_i^2 < \infty$, as posited by Robbins and Monro (1951). Assumption 2.1(b) narrows our focus to models for which the likelihood depends on parameters $\theta$ through the linear combination $X^\intercal \theta$. This family of models is large and includes generalized linear models, Cox proportional hazards models, and M-estimation. Furthermore, in Section 5 we discuss a significant relaxation of Assumption 2.1(b). Assumption 2.1(c) puts a Lipschitz condition on the log-likelihood but it is used only for deriving finite-sample error bounds in Theorem 2.1—it is possible that this condition can be relaxed. Assumption 2.1(d) is equivalent to assuming strong convexity for the negative log-likelihood, which is typical for proving convergence in probability. The assumption on the observed Fisher information is less standard and, intuitively, it posits that a minimum of statistical information is received from any data point, at least for certain model parameters. Making this assumption allows us to forgo boundedness assumptions on the errors of stochastic gradients that were originally used by Robbins and Monro (1951), and have since been standard. Finally, Assumption 2.1(f) posits the typical Lindeberg conditions that are necessary to invoke the central limit theorem and prove asymptotic normality; this assumption follows the conditions defined by Fabian (1968) for the normality of explicit SGD procedures.

### 2.1. *Finite-sample error bounds.*

Here, we derive bounds for the errors $\mathbb{E}(\|\theta_n^{\mathrm{im}} - \theta_\star\|^2)$ on a finite sample of fixed size $n$.

THEOREM 2.1. *Let* $\delta_n = \mathbb{E}(\|\theta_n^{\mathrm{im}} - \theta_\star\|^2)$. *Suppose that Assumptions* 2.1(a), (b), (c), (d) *and* (e) *hold. Then there exist constants* $n_0 > 0$ *and* $\kappa = 1 + 2\gamma_1 \mu \underline{\lambda_c} \underline{\lambda_f}$ *for some* $\mu \in (0, 1]$ *such that*

$$\delta_n \leq \frac{4L_0^2 \overline{\lambda_c}^2 \gamma_1 \kappa}{\mu \underline{\lambda_f} \underline{\lambda_c}} n^{-\gamma} + \exp(-\log\kappa \cdot \phi_\gamma(n))[\delta_0 + \kappa^{n_0} \Gamma^2],$$

*where* $\Gamma^2 = 4L_0^2 \overline{\lambda_c}^2 \sum_i \gamma_i^2 < \infty$, *and* $\phi_\gamma(n) = n^{1-\gamma}$ *if* $\gamma < 1$, *and* $\phi_\gamma(n) = \log n$ *if* $\gamma = 1$.

Not surprisingly, implicit SGD in equation (4) matches the asymptotic rate of explicit SGD in equation (1). In particular, the iterates $\theta_n^{\mathrm{im}}$ have squared error with rate $\mathrm{O}(n^{-\gamma})$, as seen in Theorem 2.1, which is identical to the rate of error for the explicit iterates $\theta_n^{\mathrm{sgd}}$ [Benveniste, Métivier and Priouret (1990), Theorem 22, page 244]. One way to explain intuitively this similarity in convergence rates is to assume that both explicit and implicit SGD are at the same estimate $\theta_0$. Then, using definitions in equation (1) and in equation (4), a Taylor approximation of the gradient $\nabla \log f(Y_n; X_n, \theta_n^{\mathrm{im}})$ yields

$$\Delta\theta_n^{\mathrm{im}} \approx \bigl[I + \gamma_n \hat{\mathcal{I}}_n(\theta_0)\bigr]^{-1} \Delta\theta_n^{\mathrm{sgd}}, \tag{8}$$

where $\Delta\theta_n^{\mathrm{im}} = \theta_n^{\mathrm{im}} - \theta_0$ and $\Delta\theta_n^{\mathrm{sgd}} = \theta_n^{\mathrm{sgd}} - \theta_0$. Therefore, as $n \to \infty$, we have $\Delta\theta_n^{\mathrm{im}} \approx \Delta\theta_n^{\mathrm{sgd}}$, and the two procedures coincide.

Despite the similarity in convergence rates, the critical advantage of implicit SGD—more generally of implicit procedures—is their robustness to initial conditions and excess noise. This can be seen in Theorem 2.1 where the implicit procedure discounts the initial conditions $\mathbb{E}(\|\theta_0^{\mathrm{im}} - \theta_\star\|^2)$ at an exponential rate through the term $\exp(-\log\kappa \cdot \phi_\gamma(n))$. Importantly, the discounting of initial conditions happens regardless of the specification of the learning rate. In fact, large values of $\gamma_1$ can lead to faster discounting, and thus possibly to faster convergence, however, at the expense of increased variance as implied by Theorem 2.2, which is presented in the following section. The implicit iterates are therefore *unconditionally stable*, that is, virtually any specification of the learning rate will lead to a stable discounting of the initial conditions.

In contrast, explicit SGD is known to be very sensitive to the learning rate, and can numerically diverge if the rate is misspecified. For example, Moulines and Bach (2011), Theorem 1, showed that there exists a term $\exp(L^2\gamma_1^2 n^{1-2\gamma})$, where $L$ is a Lipschitz constant for the gradient of the log-likelihood, amplifying the initial conditions $\mathbb{E}(\|\theta_0^{\mathrm{sgd}} - \theta_\star\|^2)$ of explicit SGD, which can be catastrophic if the learning rate parameter $\gamma_1$ is misspecified.[3] Thus, although implicit and explicit SGD have identical asymptotic performance, they are crucially different in their stability properties. This is investigated further in Section 2.5 and in the experiments of Section 4.

2.2. *Asymptotic variance and optimal learning rates.* In the previous section, we showed that $\theta_n^{\mathrm{im}} \to \theta_\star$ in quadratic mean, that is, the implicit SGD iterates converge to the true model parameters $\theta_\star$, similar to classical results for the explicit SGD iterates $\theta_n^{\mathrm{sgd}}$. Thus, $\theta_n^{\mathrm{im}}$ and $\theta_n^{\mathrm{sgd}}$ are consistent estimators of $\theta_\star$. In the

[3] The Lipschitz conditions are different in the two works; however, this does not affect our conclusions. Our result remains effectively unchanged if we assume Lipschitz continuity of the gradient $\nabla\ell$ instead of the log-likelihood $\ell$, similar to Moulines and Bach (2011); see comment after the proof of Theorem 2.1.

following theorem, we show that both SGD estimators have the same asymptotic variance.

THEOREM 2.2. *Consider SGD procedures in equation* (1) *and in equation* (4), *and suppose that Assumptions* 2.1(a), (c), (d), (e) *hold, where* $\gamma = 1$, *and that* $2\gamma_1 C\mathcal{I}(\theta_\star) \succ I$. *The asymptotic variance of the explicit SGD estimator in equation* (1) *satisfies*

$$n \operatorname{Var}(\theta_n^{\mathrm{sgd}}) \to \gamma_1^2 \bigl(2\gamma_1 C\mathcal{I}(\theta_\star) - I\bigr)^{-1} C\mathcal{I}(\theta_\star)C.$$

*The asymptotic variance of the implicit SGD estimator in equation* (4) *satisfies*

$$n \operatorname{Var}(\theta_n^{\mathrm{im}}) \to \gamma_1^2 \bigl(2\gamma_1 C\mathcal{I}(\theta_\star) - I\bigr)^{-1} C\mathcal{I}(\theta_\star)C.$$

REMARKS. Although the implicit SGD estimator $\theta_n^{\mathrm{im}}$ is significantly more stable than the explicit estimator $\theta_n^{\mathrm{sgd}}$ (Theorem 2.1), both estimators have the same asymptotic efficiency in the limit according to Theorem 2.2. This implies that implicit SGD is a superior form of SGD, and should be preferred when the calculation of implicit updates in equation (4) is computationally feasible. In Section 3, we show that this is possible in a large family of statistical models, and illustrate with several numerical experiments in Section 4.1.

Asymptotic variance results in stochastic approximation similar to Theorem 2.2 were first obtained by Chung (1954), Sacks (1958), and followed by Fabian (1968), Polyak and Tsypkin (1979), and several other authors [see also Ljung, Pflug and Walk (1992), Parts I, II]. We contribute to this literature in two important ways. First, our asymptotic variance result includes implicit SGD, which is a stochastic approximation procedure with implicitly defined updates, whereas other works consider only explicit updates. Second, in our setting we estimate recursively the true parameters $\theta_\star$ of a statistical model, and thus we can exploit the typical regularity conditions of Assumption 2.1(d) to derive the asymptotic variance of $\theta_n^{\mathrm{im}}$ (and $\theta_n^{\mathrm{sgd}}$) in a simplified closed-form. We illustrate the asymptotic variance results of Theorem 2.2 in Section 4.1.1.

2.2.1. *Optimal learning rates.* Crucially, the asymptotic variance formula of Theorem 2.2 depends on the limit of the sequence $C_n$ used in the SGD procedures of equation (1) and equation (4). We distinguish two classes of procedures, one where $C_n = I$, known as *first-order procedures*, and a second class where $C_n$ is not trivial, known as *second-order procedures*.

In first-order procedures, only gradients are used in the SGD procedures. Inevitably, no matter how we set the learning rate parameter $\gamma_1$, first-order SGD procedures will lose statistical efficiency. We can immediately verify this by comparing the asymptotic variance in Theorem 2.2 with the asymptotic variance of the maximum likelihood estimator (MLE), denoted by $\theta_N^{\mathrm{mle}}$, on a data set with

$N$ data points $\{(X_n, Y_n)\}$, $n = 1, 2, \dots, N$. Under the regularity conditions of Assumption 2.1(d), the MLE is the asymptotically optimal unbiased estimator and $N \operatorname{Var}(\theta_N^{\text{mle}} - \theta_\star) \to \mathcal{I}(\theta_\star)^{-1}$. By Theorem 2.2 and convergence of implicit SGD, it holds $N \operatorname{Var}(\theta_N^{\text{im}} - \theta_\star) \to \gamma_1^2(2\gamma_1\mathcal{I}(\theta_\star) - I)^{-1}\mathcal{I}(\theta_\star)$, which also holds for $\theta_N^{\text{sgd}}$. For any $\gamma_1 > 0$, we have as an identity that

$$\gamma_1^2\bigl(2\gamma_1\mathcal{I}(\theta_\star) - I\bigr)^{-1}\mathcal{I}(\theta_\star) \succeq \mathcal{I}(\theta_\star)^{-1}. \tag{9}$$

The proof is rather quick if we consider $\lambda_i \in \operatorname{eig}(\mathcal{I}(\theta_\star))$ and note that $\gamma_1^2\lambda_i/(2\gamma_1\lambda_i - 1)$ is the corresponding eigenvalue of the left-hand matrix in inequality (9) and $1/\lambda_i$ is the eigenvalue of $\mathcal{I}(\theta_\star)^{-1}$, and that $(2\gamma_1\underline{\lambda_f} - 1) > 0$ implies that

$$\gamma_1^2\lambda_i/(2\gamma_1\lambda_i - 1) \geq 1/\lambda_i,$$

for every $\lambda_i \in \operatorname{eig}(\mathcal{I}(\theta_\star))$. Therefore, both SGD estimators lose information and this loss can be quantified exactly by inequality (9). This inequality can also be used to find the optimal choice for $\gamma_1$ given an appropriate objective. As demonstrated in the experiments in Section 4, this often suffices to achieve estimates that are comparable with MLE in statistical efficiency but with substantial computational gains. One reasonable objective is to minimize the trace of the asymptotic variance matrix, that is, to set $\gamma_1$ equal to

$$\gamma_1^\star = \arg\min_{x > 1/2\underline{\lambda_f}} \sum_i x^2\lambda_i/(2x\lambda_i - 1). \tag{10}$$

Equation (10) is defined under the constraint $x > 1/(2\underline{\lambda_f})$ because Theorem 2.2 requires $2\gamma_1\mathcal{I}(\theta_\star) - I$ to be positive definite.

Of course, the eigenvalues $\lambda_i$ are unknown in practice and need to be estimated from the data. This problem has received significant attention recently and several methods exist [see El Karoui (2008), and references within]. We will use equation (10) extensively in our experiments (Section 4) in order to tune the SGD procedures. However, we note that in first-order SGD procedures, knowing the eigenvalues $\lambda_i$ of $\mathcal{I}(\theta_\star)$ does not necessarily achieve statistical efficiency because of the spectral gap of $\mathcal{I}(\theta_\star)$, that is, the ratio between its maximum eigenvalue $\overline{\lambda_f}$ and minimum eigenvalue $\underline{\lambda_f}$; for instance, if $\underline{\lambda_f} = \overline{\lambda_f}$, then the choice of learning rate parameter according to equation (10) leads to statistically efficient first-order SGD procedures. However, this case is not typical in practice, especially in many dimensions.

In second-order procedures, we assume nontrivial condition matrices $C_n$. Such procedures are called second-order because they usually leverage curvature information from the Fisher information matrix (or the Hessian of the log-likelihood). They are also known as *adaptive* procedures because they adapt their hyperparameters, that is, learning rates $\gamma_n$ or condition matrices $C_n$, according to observed data. For instance, let $C_n \equiv \mathcal{I}(\theta_\star)^{-1}$ and $\gamma_1 = 1$. Plugging in $C_n = \mathcal{I}(\theta_\star)^{-1}$ in Theorem 2.2, the normalized asymptotic variance of the SGD estimators is

$$\gamma_1^2(2\gamma_1\mathcal{I}(\theta_\star)^{-1}\mathcal{I}(\theta_\star) - I)^{-1}\mathcal{I}(\theta_\star)^{-1}\mathcal{I}(\theta_\star)\mathcal{I}(\theta_\star)^{-1} = \mathcal{I}(\theta_\star)^{-1},$$

which is the theoretically optimal asymptotic variance of the MLE, that is, the Cramér–Rao lower bound.

Therefore, to achieve asymptotic efficiency, second-order procedures need to estimate the Fisher information matrix at $\theta_\star$. Because $\theta_\star$ is unknown one can simply use $C_n = \mathcal{I}(\theta_n^{\text{im}})^{-1}$ [or $C_n = \mathcal{I}(\theta_{n-1}^{\text{sgd}})^{-1}$] as an iterative estimate of $\mathcal{I}(\theta_\star)$, and the same optimality result holds. This approach in second-order explicit SGD was first studied by Sakrison (1965), and later by Nevelson and Khasminskiĭ (1973), Chapter 8, Theorem 5.4. It was later extended by Fabian (1978) and several other authors. Notably, Amari (1998) refers to the direction $\mathcal{I}(\theta_{n-1}^{\text{sgd}})^{-1}\nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{sgd}})$ as the "natural gradient" and uses information geometry arguments to prove statistical optimality.

An alternative way to implement second-order procedures is to use stochastic approximation to estimate $\mathcal{I}(\theta_\star)$, in addition to the approximation procedure estimating $\theta_\star$. For example, Amari, Park and Fukumizu (2000) proposed the following second-order procedure:

$$\begin{aligned} C_n^{-1} &= (1 - a_n)C_{n-1}^{-1} + a_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{am}}) \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{am}})^\intercal, \\ \theta_n^{\text{am}} &= \theta_{n-1}^{\text{am}} + \gamma_n C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{am}}), \end{aligned} \tag{11}$$

where $a_n = a_1/n$ is a learning rate sequence, separate from $\gamma_n$. By standard stochastic approximation, $C_n^{-1}$ converges to $\mathcal{I}(\theta_\star)$, and thus the procedure in equation (11) is asymptotically optimal. However, there are two important problems with this procedure. First, it is computationally costly because of matrix inversions. A faster way is to apply quasi-Newton ideas. SGD-QN developed by Bordes, Bottou and Gallinari (2009) is such a procedure where the first expensive matrix computations are substituted by the secant condition. Second, the stochastic approximation of $\mathcal{I}(\theta_\star)$ is usually very noisy in high-dimensional problems and this affects the main approximation for $\theta_\star$. Recently, more robust variants of SGD-QN have been proposed [Byrd et al. (2016)].

Another notable adaptive procedure is AdaGrad [Duchi, Hazan and Singer (2011)], which is defined as

$$\begin{aligned} C_n^{-1} &= C_{n-1}^{-1} + \text{diag}(\nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{ada}}) \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{ada}})^\intercal), \\ \theta_n^{\text{ada}} &= \theta_{n-1}^{\text{ada}} + \gamma_1 C_n^{1/2} \nabla \log f(Y_n; X_n, \theta_{n-1}^{\text{ada}}), \end{aligned} \tag{12}$$

where diag(·) takes the diagonal matrix of its matrix argument, and the learning rate is set constant to $\gamma_n \equiv \gamma_1$. AdaGrad can be considered a second-order procedure because it tries to approximate the Fisher information matrix; however, it only uses gradient information so technically it is first-order. Under appropriate conditions, $C_n^{-1} \to \operatorname{diag}(\mathcal{I}(\theta_\star))$ and a simple modification in the proof of Theorem 2.2 can show that the asymptotic variance of the AdaGrad estimate is given by

$$\sqrt{n}\,\mathrm{Var}\big(\theta_n^{\mathtt{ada}}\big) \to \frac{\gamma_1}{2}\operatorname{diag}\big(\mathcal{I}(\theta_\star)\big)^{-1/2}. \tag{13}$$

This result reveals an interesting trade-off achieved by AdaGrad and a subtle contrast to first-order SGD procedures. The asymptotic variance of AdaGrad is $\mathrm{O}(1/\sqrt{n})$, which indicates significant loss of information. However, this rate is attained *regardless* of the specification of the learning rate parameter $\gamma_1$.[4] In contrast, as shown in Theorem 2.2, first-order SGD procedures require $2\gamma_1\mathcal{I}(\theta_\star) - I \succ 0$ in order to achieve the $\mathrm{O}(1/n)$ rate, and the rate is significantly worse if this condition is not met. For instance, Nemirovski et al. (2008) give an example of misspecification of $\gamma_1$ where the rate of first-order explicit SGD is $\mathrm{O}(n^{-\varepsilon})$, and $\varepsilon$ can be arbitrarily small. The variance result in equation (13) is illustrated in the numerical experiments of Section 4.1.1.

2.3. *Optimality with averaging.* As shown in Section 2.2.1, Theorem 2.2 implies that first-order SGD procedures can be statistically inefficient, especially in many dimensions. One surprisingly simple idea to achieve statistical efficiency is to combine larger learning rates with averaging of the iterates. In particular, we consider the procedure

$$\begin{aligned} \theta_n^{\mathrm{im}} &= \theta_{n-1}^{\mathrm{im}} + \gamma_n \nabla \log f\big(Y_n; X_n, \theta_n^{\mathrm{im}}\big), \\ \overline{\theta_n^{\mathrm{im}}} &= \frac{1}{n}\sum_{i=1}^{n} \theta_i^{\mathrm{im}}, \end{aligned} \tag{14}$$

where $\theta_n^{\mathrm{im}}$ are the typical implicit SGD iterates in equation (4), and $\gamma_n = \gamma_1 n^{-\gamma}$, $\gamma \in [0.5, 1)$. Under suitable conditions, the iterates $\overline{\theta_n^{\mathrm{im}}}$ are asymptotically efficient. This is formalized in the following theorem.

THEOREM 2.3. *Consider the SGD procedure defined in equation* (14) *and suppose Assumptions* 2.1(a), (c), (d) *and* (e) *hold*, *where* $\gamma \in [0.5, 1)$. *Then* $\overline{\theta_n^{\mathrm{im}}}$ *converges to* $\theta_\star$ *in probability and is asymptotically efficient*, *that is*,

$$n\,\mathrm{Var}\big(\overline{\theta_n^{\mathrm{im}}}\big) \to \mathcal{I}(\theta_\star)^{-1}.$$

[4] This follows from a property of recursions [Toulis and Airoldi (2017), Lemma 2.4]. On a high-level, the term $\gamma_{n-1}/\gamma_n$ is important for the variance rates of AdaGrad and SGD. When $\gamma_n \propto 1/n$, as in Theorem 2.2, it holds that $\gamma_{n-1}/\gamma_n = 1 + \gamma_n/\gamma_1 + \mathrm{O}(\gamma_n^2)$, which explains the quantity $2\mathcal{I}(\theta_\star) - I/\gamma_1$ in first-order SGD. The rate $\mathrm{O}(1/n)$ is attained only if $2\mathcal{I}(\theta_\star) - I/\gamma_1 \succ 0$. When $\gamma_n \propto 1/\sqrt{n}$, as in AdaGrad, it holds that $\gamma_{n-1}/\gamma_n = 1 + \mathrm{o}(\gamma_n)$ and the rate $\mathrm{O}(1/\sqrt{n})$ is attained without any additional requirements.

REMARKS. In the context of explicit stochastic approximations, averaging was first proposed and analyzed by Ruppert (1988) and Bather (1989). Ruppert (1988) argued that larger learning rates in stochastic approximation uncorrelates the iterates allowing averaging to improve efficiency. Polyak and Juditsky (1992) expanded the scope of averaging by proving asymptotic optimality in more general explicit stochastic approximations that operate under suitable conditions similar to Theorem 2.3. Polyak and Juditsky (1992) thus proved that slowly converging stochastic approximations can be improved by using larger learning rates and averaging of the iterates. Recent work has analyzed explicit updates with averaging [Zhang (2004), Xu (2011), Bach and Moulines (2013), Shamir and Zhang (2012)], and has shown their superiority in numerous learning tasks. More recently, Toulis, Tran and Airoldi (2016) derived the finite-sample error bounds of the averaged implicit SGD estimator.

2.4. *Asymptotic normality.* Asymptotic distributions, or more generally invariance principles, are well studied in classical stochastic approximation [Ljung, Pflug and Walk (1992), Chapter II.8]. In this section, we leverage Fabian's theorem [Fabian (1968)] to show that iterates from implicit SGD are asymptotically normal.

THEOREM 2.4. *Suppose that Assumptions* 2.1(a), (c), (d), (e), (f) *hold. Then the iterate* $\theta_n^{\text{im}}$ *of implicit SGD in equation* (4) *is asymptotically normal*, *such that*

$$n^{\gamma/2}(\theta_n^{\text{im}} - \theta_\star) \to \mathcal{N}_p(0, \Sigma),$$

*where* $\Sigma = \gamma_1^2(2\gamma_1 C\mathcal{I}(\theta_\star) - I)^{-1} C\mathcal{I}(\theta_\star)C$.

REMARKS. The combined results of Theorems 2.1, 2.2 and 2.4 indicate that implicit SGD is numerically stable and has known asymptotic variance and distribution. Therefore, contrary to explicit SGD that has severe stability issues, implicit SGD emerges as a stable estimation procedure with known standard errors, which enables typical statistical tasks, such as confidence intervals, hypothesis testing and model checking. We show empirical evidence supporting this claim in Section 4.1.2.

2.5. *Stability.* To illustrate the stability, or lack thereof, of both SGD estimators in small-to-moderate samples, we simplify the SGD procedures and inspect the size of the biases $\mathbb{E}(\theta_n^{\text{sgd}} - \theta_\star)$ and $\mathbb{E}(\theta_n^{\text{im}} - \theta_\star)$. In particular, based on Theorem 2.1, we simply assume the Taylor expansion $\nabla \log f(Y_n; X_n, \theta_n) = -\mathcal{I}(\theta_\star)(\theta_n - \theta_\star) + \mathrm{O}(\gamma_n)$; to simplify further we ignore the remainder term $\mathrm{O}(\gamma_n)$.

Under this simplification, the SGD procedures in equation (1) and in equation (4) can be written as follows:

$$\mathbb{E}(\theta_n^{\text{sgd}} - \theta_\star) = (I - \gamma_n \mathcal{I}(\theta_\star))\mathbb{E}(\theta_{n-1}^{\text{sgd}} - \theta_\star) = P_1^n b_0, \tag{15}$$

$$\mathbb{E}(\theta_n^{\text{im}} - \theta_\star) = (I + \gamma_n \mathcal{I}(\theta_\star))^{-1}\mathbb{E}(\theta_{n-1}^{\text{im}} - \theta_\star) = Q_1^n b_0, \tag{16}$$

where $P_1^n = \prod_{i=1}^n (I - \gamma_i \mathcal{I}(\theta_\star))$, $Q_1^n = \prod_{i=1}^n (I + \gamma_i \mathcal{I}(\theta_\star))^{-1}$, and $b_0$ denotes the initial bias of the two procedures from a common starting point $\theta_0$. Thus, the matrices $P_1^n$ and $Q_1^n$ describe how fast the initial bias decays for the explicit and implicit SGD, respectively. In the limit, $P_1^n \to 0$ and $Q_1^n \to 0$ [Toulis and Airoldi (2017), proof of Lemma 2.4], and thus both methods are *asymptotically stable*.

However, the explicit procedure has significant stability issues in small-to-moderate samples. By inspection of equation (15), the magnitude of $P_1^n$ is dominated by $\overline{\lambda_f}$, the maximum eigenvalue of $\mathcal{I}(\theta_\star)$. Furthermore, the rate of convergence is dominated by $\underline{\lambda_f}$, the minimum eigenvalue of $\mathcal{I}(\theta_\star)$.[5] For stability, it is desirable $|1 - \gamma_1 \lambda_i| < 1$, for all eigenvalues $\lambda_i \in \text{eig}(\mathcal{I}(\theta_\star))$. This implies the requirement $\gamma_1 < 2/\overline{\lambda_f}$ for stability. Furthermore, Theorem 2.2 implies the requirement $\gamma_1 > 1/2\underline{\lambda_f}$ for fast convergence. This is problematic in high-dimensional settings because $\overline{\lambda_f}$ is typically orders of magnitude larger than $\underline{\lambda_f}$. Thus, the requirements for stability and speed of convergence are in conflict in explicit procedures: to ensure stability we need a small learning rate parameter $\gamma_1$, thus paying a high price in convergence which will be at the order of $\mathrm{O}(n^{-\gamma_1 \underline{\lambda_f}})$, and vice versa.

In contrast, the implicit procedure is *unconditionally stable*. The eigenvalues of $Q_1^n$ are $\lambda_i' = \prod_{j=1}^n 1/(1 + \gamma_1 \lambda_i / j) = \mathrm{O}(n^{-\gamma_1 \lambda_i})$. Critically, it is no longer required to have a small $\gamma_1$ for stability because the eigenvalues of $Q_1^n$ are always less than one. We summarize these findings in the following lemma.

LEMMA 2.1. *Let* $\overline{\lambda_f} = \max \text{eig}(\mathcal{I}(\theta_\star))$, *and suppose* $\gamma_n = \gamma_1/n$ *and* $\gamma_1 \overline{\lambda_f} > 1$. *Then the maximum eigenvalue of* $P_1^n$ *satisfies*

$$\max_{n>0} \max \text{eig}(P_1^n) = \Theta\big(2^{\gamma_1 \overline{\lambda_f}} / \sqrt{\gamma_1 \overline{\lambda_f}}\big).$$

*For the implicit method*,

$$\max_{n>0} \max \text{eig}(Q_1^n) = \mathrm{O}(1).$$

REMARKS. Lemma 2.1 shows that in the explicit SGD procedure the effect from the initial bias can be amplified in an arbitrarily large way before fading out, if the learning rate is misspecified (i.e., if $\gamma_1 \gg 1/\overline{\lambda_f}$). This sensitivity of explicit SGD is well known and requires problem-specific considerations to be avoided in practice, for example, preprocessing, small-sample tests, projections, truncation [Chen, Lei and Gao (1988)]. In fact, there exists voluminous work, which is still ongoing, in designing learning rates to stabilize explicit SGD; see, for example, a review by George and Powell (2006). Implicit procedures render such ad-hoc designs obsolete because they remain stable regardless of learning rate design, and still maintain the asymptotic convergence and efficiency properties of explicit SGD procedures.

[5] To see this, note that the eigenvalues of $P_1^n$ are $\lambda_i' = \prod_j (1 - \gamma_1 \lambda_i / j) = \mathrm{O}(n^{-\gamma_1 \lambda_i})$ if $0 < \gamma_1 \lambda_i < 1$. See also the proof of Lemma 2.1.

**3. Applications.** Here, we show how to apply implicit SGD in equation (4) for estimation in generalized linear models, Cox proportional hazards, and more general M-estimation problems. We start by developing an algorithm that efficiently computes the implicit update in equation (4), and is applicable to all aforementioned models.

3.1. *Efficient computation of implicit updates.* The main difficulty in applying implicit SGD is the solution of the multidimensional fixed-point equation (4). In a large family of models where the likelihood depends on the parameter $\theta_\star$ only through the natural parameter $X_n^\intercal\theta_\star$, the solution of the fixed-point equation is feasible and computationally efficient. We prove the general result in Theorem 3.1.

For the rest of this section, we will treat $\ell(X^\intercal\theta; Y)$ as a function of the natural parameter $X^\intercal\theta$ for a fixed outcome $Y$. Thus, $\ell'(X^\intercal\theta; Y)$ will refer to the first derivative of $\ell$ with respect to $X^\intercal\theta$ with fixed $Y$.

THEOREM 3.1. *Suppose Assumption* 2.1(b) *holds, then the gradient of the log-likelihood is a scaled version of covariate $X$, that is, for every $\theta \in \mathbb{R}^p$ there is a scalar $\lambda \in \mathbb{R}$ such that*

$$\nabla \log f(Y; X, \theta) = \lambda X.$$

*Thus, the gradient in the implicit update in equation* (4) *is a scaled version of the gradient calculated at the previous iterate, that is,*

$$\nabla \log f\big(Y_n; X_n, \theta_n^{\mathrm{im}}\big) = \lambda_n \nabla \log f\big(Y_n; X_n, \theta_{n-1}^{\mathrm{im}}\big), \tag{17}$$

*where the scalar $\lambda_n$ satisfies*

$$\lambda_n \ell'\big(X_n^\intercal\theta_{n-1}^{\mathrm{im}}; Y_n\big) = \ell'\big(X_n^\intercal\theta_{n-1}^{\mathrm{im}} + \gamma_n\lambda_n\ell'\big(X_n^\intercal\theta_{n-1}^{\mathrm{im}}; Y_n\big)X_n^\intercal C_n X_n; Y_n\big). \tag{18}$$

REMARKS. Theorem 3.1 implies that computing the implicit update in equation (4) reduces to numerically solving the one-dimensional fixed-point equation for $\lambda_n$—this idea is implemented in Algorithm 1. As shown in the proof of Theorem 3.1, this implementation is fast because $\lambda_n$ lies on an interval $B_n$ of size $\mathrm{O}(\gamma_n)$. We also note that Theorem 3.1 can be readily extended to cases with linearly separable regularizers, for instance, regularizers using the $L_1$ norm $\|\theta\| = \sum_i |\theta_i|$. In such cases, there are additional fixed-point equations as in Step 9 of Algorithm 1 that involve the components of the regularizer. More generally, for families of models that do not satisfy Assumption 2.1(b) there are methods to *approximately* perform the implicit update—we discuss one such method in Section 3.3.

3.2. *Generalized linear models.* In this section, we apply implicit SGD to estimate generalized linear models (GLMs). In such models, $Y_n$ follows an exponential distribution conditional on $X_n$, and $\mathbb{E}(Y_n \mid X_n) = h(X_n^\intercal\theta_\star)$, where $h$ is the

**Algorithm 1:** Efficient implementation of implicit SGD in equation (4)

```
for all $n \in \{1, 2, \ldots\}$ do
    # compute search bounds $B_n$
    $r_n \leftarrow \gamma_n \ell'(X_n^\intercal \theta_{n-1}^{\mathrm{im}}; Y_n)$
    $B_n \leftarrow [0, r_n]$
    if $r_n \le 0$ then
        $B_n \leftarrow [r_n, 0]$
    end if
    # solve fixed-point equation by a root-finding method
    $\xi = \gamma_n \ell'(X_n^\intercal \theta_{n-1}^{\mathrm{im}} + \xi X_n^\intercal C_n X_n; Y_n)$, $\xi \in B_n$
    $\lambda_n \leftarrow \xi / r_n$
    # following update is equivalent to update in equation (4)
    $\theta_n^{\mathrm{im}} \leftarrow \theta_{n-1}^{\mathrm{im}} + \gamma_n \lambda_n C_n \nabla \log f(Y_n; X_n, \theta_{n-1}^{\mathrm{im}})$
end for
```

*transfer function* of the GLM model [Nelder and Wedderburn (1972)]. Furthermore, the gradient of the GLM log-likelihood for parameter value $\theta$ at data point $(X_n, Y_n)$ is given by

$$\nabla \log f(Y_n; X_n, \theta) = \bigl[Y_n - h\bigl(X_n^\intercal \theta\bigr)\bigr] X_n. \tag{19}$$

The conditional variance of $Y_n$ is $\mathrm{Var}(Y_n \mid X_n) = h'(X_n^\intercal \theta_\star) X_n X_n^\intercal$, and thus the Fisher information matrix is $\mathcal{I}(\theta) = \mathbb{E}(h'(X_n^\intercal \theta) X_n X_n^\intercal)$. Thus, the SGD procedures in equation (1) and in equation (4) can be written as

$$\theta_n^{\mathrm{sgd}} = \theta_{n-1}^{\mathrm{sgd}} + \gamma_n C_n \bigl[Y_n - h\bigl(X_n^\intercal \theta_{n-1}^{\mathrm{sgd}}\bigr)\bigr] X_n, \tag{20}$$

$$\theta_n^{\mathrm{im}} = \theta_{n-1}^{\mathrm{im}} + \gamma_n C_n \bigl[Y_n - h\bigl(X_n^\intercal \theta_n^{\mathrm{im}}\bigr)\bigr] X_n. \tag{21}$$

Implementation of explicit SGD is straightforward. Implicit SGD can be implemented through Algorithm 1. In particular, $\log f(Y; X, \theta) \equiv \ell(X^\intercal \theta; Y)$ with $\ell(\eta; Y) = Y - h(\eta)$. In typical GLMs, $h$ is twice-differentiable and also $h'(\eta) \ge 0$ because it is proportional to the conditional variance of $Y$ given $X$, thus fulfilling Assumption 2.1(b). In the simplified case where $C_n = I$, the identity matrix, for all $n$, Algorithm 1 simplifies to Algorithm 2, which was first derived by Toulis, Airoldi and Rennie (2014). We make extensive experiments using Algorithm 2 in Section 4.2.

3.3. *Cox proportional hazards model.* Here, we apply SGD to estimate a Cox proportional hazards model, which is a popular model in survival analysis [Cox (1972), Klein and Moeschberger (2003)]. Multiple variations of the model exist but for simplicity we will analyze one simple variation that is popular in practice [Davison (2003)]. Consider $N$ individuals, indexed by $i$, with observed survival

**Algorithm 2:** Estimation of GLMs with first-order implicit SGD ($C_n = I$)

```
for all $n \in \{1, 2, \ldots\}$ do
    $r_n \leftarrow \gamma_n [Y_n - h(X_n^\intercal \theta_{n-1}^{\mathrm{im}})]$
    $B_n \leftarrow [0, r_n]$
    if $r_n \leq 0$ then
        $B_n \leftarrow [r_n, 0]$
    end if
    $\xi = \gamma_n [Y_n - h(X_n^\intercal \theta_{n-1}^{\mathrm{im}} + \xi \|X_n\|^2)]$, $\xi \in B_n$
    $\theta_n^{\mathrm{im}} \leftarrow \theta_{n-1}^{\mathrm{im}} + \xi X_n$
end for
```

times $Y_i$, failure indicators $d_i$, and covariates $X_i$. The survival times can be assumed ordered, $Y_1 < Y_2 < \cdots < Y_N$, whereas $d_i = 1$ denotes failure (e.g., death) and $d_i = 0$ indicates censoring (e.g., patient dropped out of study). Given a failure for unit $i$ ($d_i = 1$) at time $Y_i$, the *risk set* $\mathcal{R}_i$ is defined as the set of individuals that could possibly fail at $Y_i$, that is, all individuals except those who failed or were censored before $Y_i$. In our simplified model, $\mathcal{R}_i = \{i, i+1, \ldots, N\}$. Define $\eta_i(\theta) = \exp(X_i^\intercal \theta)$, then the log-likelihood $\ell$ for $\theta$ is given by Davison (2003), Chapter 10,

$$\ell(\theta; X, Y) = \sum_{i=1}^{N} [d_i - H_i(\theta)\eta_i(\theta)] X_i, \tag{22}$$

where $H_i(\theta) = \sum_{j: i \in \mathcal{R}_j} d_j (\sum_{k \in \mathcal{R}_j} \eta_k(\theta))^{-1}$. In an online setting, where $N$ is infinite and data points $(X_i, Y_i)$ are observed one at a time, future observations affect the likelihood of previous ones, as can be seen by inspection of equation (22). Therefore, we apply SGD assuming fixed $N$ to estimate the MLE $\theta_N^{\mathrm{mle}}$. As mentioned in Section 1, our theory in Section 2 can be applied unchanged if we only substitute $\theta_\star$, the true parameter, with the MLE $\theta_N^{\mathrm{mle}}$.

A straightforward implementation of explicit SGD in equation (1) for the Cox model is shown in Algorithm 3. For implicit SGD in equation (4), we have the

**Algorithm 3:** Explicit SGD for Cox proportional hazards model

```
for $n = 1, 2, \ldots$ do
    $i \leftarrow \mathrm{sample}(1, N)$
    $\widehat{H}_i \leftarrow \sum_{j: i \in \mathcal{R}_j} \frac{d_j}{\sum_{k \in \mathcal{R}_j} \eta_k(\theta_{n-1}^{\mathrm{sgd}})}$
    $w_{n-1} \leftarrow [d_i - \widehat{H}_i \eta_i(\theta_{n-1}^{\mathrm{sgd}})]$
    $\theta_n^{\mathrm{sgd}} = \theta_{n-1}^{\mathrm{sgd}} + \gamma_n w_{n-1} C_n X_i$
```

**Algorithm 4:** Implicit SGD for Cox proportional hazards model

```
for $n = 1, 2, \ldots$ do
    $i \leftarrow \text{sample}(1, N)$
    $\widehat{H}_i \leftarrow \sum_{j:i\in\mathcal{R}_j} \frac{d_j}{\sum_{k\in\mathcal{R}_j} \eta_k(\theta_{n-1}^{\text{im}})}$
    $w(\theta) = d_i - \widehat{H}_i \eta_i(\theta)$
    $W_n \leftarrow w(\theta_{n-1}^{\text{im}}) C_n X_i$
    $\lambda_n w(\theta_{n-1}^{\text{im}}) = w(\theta_{n-1}^{\text{im}} + \gamma_n \lambda_n W_n)$
    $\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n W_n$
```

update

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n [d_i - H_i(\theta_n^{\text{im}}) \eta_i(\theta_n^{\text{im}})] X_i, \tag{23}$$

which is similar to the implicit procedure for GLMs in equation (21). However, the log-likelihood term $d_i - H_i(\theta_n^{\text{im}}) \eta_i(\theta_n^{\text{im}})$ does not satisfy the conditions of Assumption 2.1(b) because $H_i(\theta)$ may be increasing or decreasing since it depends on terms $X_j^\intercal \theta$, $j \neq i$. Thus, Theorem 3.1 cannot be applied.

One idea to circumvent this problem is to simply compute $H_i(\cdot)$ on the previous update $\theta_{n-1}^{\text{im}}$ instead of the current $\theta_n^{\text{im}}$. Then update (23) becomes

$$\theta_n^{\text{im}} = \theta_{n-1}^{\text{im}} + \gamma_n [d_i - H_i(\theta_{n-1}^{\text{im}}) \eta_i(\theta_n^{\text{im}})] X_i, \tag{24}$$

which now satisfies Assumption 2.1(b) since $H_i(\theta_{n-1}^{\text{im}})$ is constant with respect to $\theta_n^{\text{im}}$. This idea is implemented in Algorithm 4, but can be more generally applied in models that don't satisfy Assumption 2.1(b); see Section 5 for a discussion.

3.4. *M-estimation.* Given $N$ observed data points $(X_i, Y_i)$ and a convex function $\rho : \mathbb{R} \to \mathbb{R}^+$, the M-estimator is defined as

$$\hat{\theta}^m = \arg\min_\theta \sum_{i=1}^N \rho(Y_i - X_i^\intercal \theta), \tag{25}$$

where it is assumed $Y_i = X_i^\intercal \theta_\star + \varepsilon_i$, and $\varepsilon_i$ are i.i.d. zero mean-valued noise. M-estimators are especially useful in robust statistics [Huber (1964)] because appropriate choice of $\rho$ can reduce the influence of outliers in data. Typically, $\rho$ is twice-differentiable around zero. In this case,

$$\mathbb{E}(\rho'(Y - X^\intercal \hat{\theta}^m) X) = 0, \tag{26}$$

where the expectation is over the empirical data distribution. Thus, according to Section 1, SGD procedures can be applied to approximate the M-estimator $\hat{\theta}^m$. There has been increased interest in the literature for fast approximation of

**Algorithm 5:** Implicit SGD for M-estimation

```
1 for $n = 1, 2, \dots$ do
2     $i \leftarrow \text{sample}(1, N)$
3     $w(\theta) = \rho'(Y_i - X_i^\intercal \theta)$
4     $\lambda_n w(\theta_{n-1}^{\text{im}}) \leftarrow w(\theta_{n-1}^{\text{im}} + \gamma_n \lambda_n w(\theta_{n-1}^{\text{im}}) C_n X_i)$    # implicit update
5     $\theta_n^{\text{im}} \leftarrow \theta_{n-1}^{\text{im}} + \gamma_n \lambda_n w(\theta_{n-1}^{\text{im}}) C_n X_i$
```

M-estimators due to their robustness [Donoho and Montanari (2016), Jain, Tewari and Kar (2014)]. The implicit SGD procedure for approximating M-estimators is defined in Algorithm 5, and is a simple adaptation of Algorithm 1.

Importantly, the conditions of Assumption 2.1(b) are met because $\rho$ is convex, and thus $\rho'' \geq 0$. Thus, Step 4 of Algorithm 5 is a straightforward application of Algorithm 1 by simply setting $\ell'(X_n' \theta_{n-1}^{\text{im}}; Y_n) \equiv \rho'(Y_n - X_n^\intercal \theta_n^{\text{im}})$. The asymptotic variance of $\theta_n^{\text{im}}$ is also easy to derive. If $S = \mathbb{E}(X_n X_n^\intercal)$, $C_n \to C >$ such that $S$ and $C$ commute, $\psi^2 = \mathbb{E}(\rho'(\varepsilon_i)^2)$, and $v(z) = \mathbb{E}(\rho'(\varepsilon_i + z))$, Theorem 2.2 can be leveraged to show that

$$n \operatorname{Var}(\theta_n^{\text{im}}) \to \psi^2 (2v'(0) CS - I)^{-1} CSC. \tag{27}$$

Historically, one of the first applications of explicit stochastic approximation procedures in robust estimation was due to Martin and Masreliez (1975). The asymptotic variance (27) was first derived, only for the explicit SGD case, by Poljak and Tsypkin (1980) using stochastic approximation theory from Nevelson and Khasminskiĭ (1973).

**4. Simulation and data analysis.** Here, we demonstrate the computational and statistical advantages of the SGD estimation procedures in equation (1) and in equation (4). For our experiments we developed a new `R` package, namely `sgd`, which has been published on CRAN. All experiments were conducted on a single laptop running Linux Ubuntu 13.x with 8 cores@2.4 GHz, 16 Gb of RAM memory and 256 Gb of physical storage with SSD technology. A separate set of experiments, which is presented in the supplemental article [Toulis and Airoldi (2017), Section 3], focuses on comparisons of implicit SGD with popular machine learning methods on typical estimation tasks.

4.1. *Numerical results.* In this section, we aim to illustrate the theoretical results of Section 2, namely the result on asymptotic variance (Theorem 2.2) and asymptotic normality (Theorem 2.4) of SGD procedures.

4.1.1. *Asymptotic variance.* In this experiment, we use a normal linear model following Xu (2011). The procedures we test are explicit SGD in equation (1), implicit SGD in equation (4), and AdaGrad in equation (12). For simplicity, we use first-order SGD, that is, $C_n = I$. In the experiment we calculate the empirical variance of said procedures for 25 values of their common learning rate parameter $\gamma_1$ in the interval $[1.2, 10]$. For every value of $\gamma_1$, we calculate the empirical variances through the following process, repeated for 150 times. First, we set $\theta_\star = (1, 1, \dots, 1)^\intercal \in \mathbb{R}^{20}$ as the true parameter value. For iterations $n = 1, 2, \dots, 1500$, we sample covariates as $X_n \sim \mathcal{N}_p(0, S)$, where $S$ is diagonal with elements uniformly on $[0.5, 5]$. The outcome $Y_n$ is then sampled as $Y_n | X_n \sim \mathcal{N}(X_n^\intercal \theta_\star, 1)$. In every repetition, we store the iterate $\theta_{1500}$ for every tested procedure and then calculate the empirical variance of stored iterates over all 150 repetitions.

For any fixed learning rate parameter $\gamma_1$, we set $\gamma_n = \gamma_1 / n$ for implicit SGD and $\gamma_n = \gamma_1$ for AdaGrad. For explicit SGD, we set $\gamma_n = \min\left(0.3, \gamma_1/(n + \|X_n\|^2)\right)$ in order to stabilize its updates. This trick is necessary by the analysis of Section 2.5. In particular, the Fisher information matrix here is $\mathcal{I}(\theta_\star) = \mathbb{E}(X_n X_n^\intercal) = S$, and thus the minimum eigenvalue is $\underline{\lambda_f} = 0.5$ and the maximum is $\overline{\lambda_f} = 5$. Therefore, for stability we require $\gamma_1 < 2/\overline{\lambda_f} = 0.4$ and for fast convergence we require $\gamma_1 > 1/(2\underline{\lambda_f}) = 1$. The two requirements are incompatible, which indicates that explicit SGD can have serious stability issues.

For given $\gamma_1 > 1$, the asymptotic variance of SGD procedures after $n$ iterations is $(1/n)\gamma_1^2(2\gamma_1 S - I)^{-1} S$, by Theorem 2.2. The asymptotic variance of AdaGrad after $n$ iterations is equal to $(\gamma_1/2\sqrt{n}) S^{-1/2}$ by equation (13). The log traces of the empirical variance of the SGD procedures and AdaGrad in this experiment are shown in Figure 1. The $x$-axis corresponds to different values of the learning rate parameter $\gamma_1$, and the $y$-axis corresponds to the log trace of the empirical variance of the iterates for all three different procedures. We also include curves for the theoretical values of the empirical variances.

We see that our theory predicts well the empirical variances of all methods. Explicit SGD performs on par with implicit SGD for moderate values of $\gamma_1$, however, it required a modification in its learning rate to make it work. Furthermore, explicit SGD quickly becomes unstable at larger values of $\gamma_1$ (see, e.g., its empirical variance for $\gamma_1 = 10$), and in several instances, not considered in Figure 1, it numerically diverged. On the other hand, AdaGrad is stable to the specification of $\gamma_1$ and tracks its theoretical variance well. However, it gives inefficient estimators because their variance has order $\mathrm{O}(1/\sqrt{n})$. Implicit SGD effectively combines stability and good statistical efficiency. First, it remains very stable to the entire range of the learning rate parameter $\gamma_1$. Second, its empirical variance is $\mathrm{O}(1/n)$ and is tracks closely the theoretical value predicted by Theorem 2.2 for all $\gamma_1$.

4.1.2. *Asymptotic normality.* In this experiment, we use the normal linear model in the setup of Section 4.1.1 to check the asymptotic normality result of Theorem 2.4. For simplicity, we only test first-order implicit SGD in equation (4) and first-order explicit SGD.

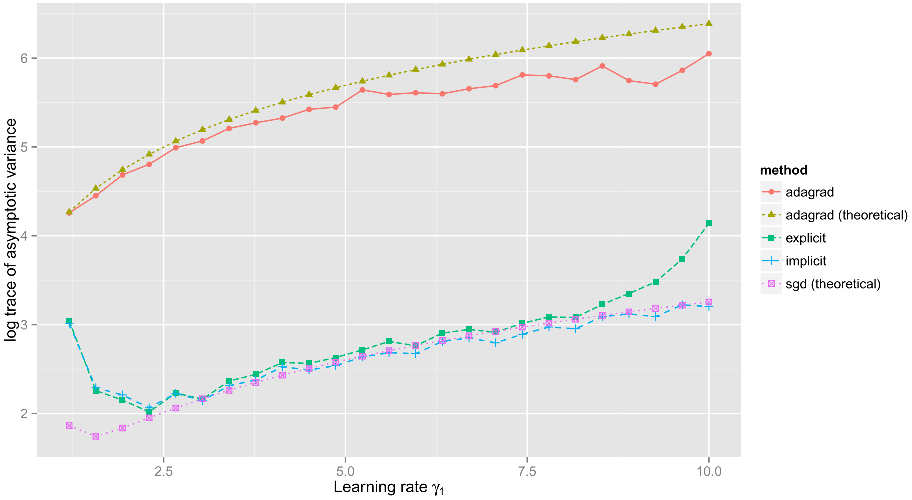

FIG. 1. *Simulation with normal model. The $x$-axis corresponds to learning rate parameter $\gamma_1$; the $y$-axis curves corresponds to log trace of the empirical variance of tested procedures* (*explicit/implicit SGD*, *AdaGrad*). *Theoretical asymptotic variances of SGD and AdaGrad are plotted as well. Implicit SGD is stable and its empirical variance is very close to its asymptotic value. Explicit SGD becomes unstable at large $\gamma_1$. AdaGrad is statistically inefficient but remains stable to large learning rates.*

In the experiment, we define a set of learning rates $(0.5, 1, 3, 5, 6, 7)$. For every learning rate, we take 400 samples of $N(\theta_N - \theta_\star)^\intercal \Sigma^{-1}(\theta_N - \theta_\star)$, where $N = 1200$ and $\theta_N$ denotes either $\theta_N^{\text{sgd}}$ or $\theta_N^{\text{im}}$. The matrix $\Sigma$ is the asymptotic variance matrix in Theorem 2.4, and $\theta_\star = 10\exp(-2 \cdot (1, 2, \dots, p))$, is the true parameter value. We use the ground-truth values both for $\Sigma$ and $\theta_\star$, as we are only interested to test normality of the iterates in this experiment. We also tried $p = 5, 10, 100$ as the parameter dimension. Because the explicit SGD procedure was very unstable across experiments we only report results for $p = 5$. Results on the implicit procedure for larger $p$ are given in the supplemental article [Toulis and Airoldi (2017)], where we also include results for a logistic regression model.

By Theorem 2.4 for implicit SGD, and by classical normality results for explicit SGD [Fabian (1968), Ljung, Pflug and Walk (1992)], the quadratic form $N(\theta_N - \theta_\star)^\intercal \Sigma^{-1}(\theta_N - \theta_\star)$ is a chi-squared random variable with $p$ degrees of freedom. Thus, for every procedure we plot this quantity against independent samples from a $\chi_p^2$ distribution and visually check for deviations. As before, we tried to stabilize explicit SGD as much as possible by setting $\gamma_n = \min(0.3, \gamma_1/(n + \|X_n\|^2))$. This worked in many iterations, but not for all. Iterations for which explicit SGD diverged were not considered. For implicit SGD, we simply set $\gamma_n = \gamma_1/n$ without additional tuning.

The results of this experiment are shown in Figure 2. The vertical axis on the grid corresponds to different values of the learning rate parameter $\gamma_1$, and the hori-

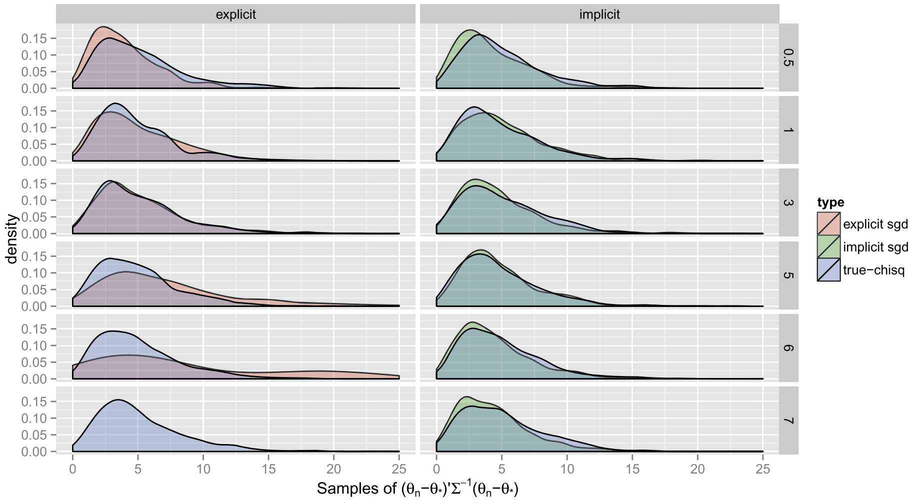

FIG. 2. *Simulation with normal model. The $x$-axis corresponds to the SGD procedure* (*explicit or implicit*) *for various values of the learning rate parameter*, $\gamma_1 \in \{0.5, 1, 3, 5, 7\}$. *The histograms* (*$x$-axis*) *for the SGD procedures are* 500 *replications of SGD where at each replication we only store the quantity* $N(\theta_N - \theta_\star)^\intercal \Sigma^{-1}(\theta_N - \theta_\star)$, *for every method* ($N = 1200$); *the theoretical covariance matrix* $\Sigma$ *is different for every learning rate and is given in Theorem* 2.2. *The data generative model is the same as in Section* 4.1.1. *We observe that implicit SGD is stable and follows the nominal chi-squared distribution. Explicit SGD becomes unstable at larger* $\gamma_1$ *and its distribution does not follow the nominal one well. In particular, the distribution of* $N(\theta_N^{\mathrm{sgd}} - \theta_\star)^\intercal \Sigma^{-1}(\theta_N^{\mathrm{sgd}} - \theta_\star)$ *becomes increasingly heavy-tailed as the learning rate parameter gets larger, and eventually diverges for* $\gamma_1 \geq 7$.

zontal axis has histograms of $N(\theta_N - \theta_\star)^\intercal \Sigma^{-1}(\theta_N - \theta_\star)$, and also includes samples from a $\chi_5^2$ distribution for visual comparison.

We see that the distribution $N(\theta_N^{\mathrm{im}} - \theta_\star)^\intercal \Sigma^{-1}(\theta_N^{\mathrm{im}} - \theta_\star)$ of the implicit iterates follows the nominal chi-squared distribution. This also seems to be unaffected by the learning rate parameter. However, the distribution of $N(\theta_N^{\mathrm{sgd}} - \theta_\star)^\intercal \Sigma^{-1}(\theta_N^{\mathrm{sgd}} - \theta_\star)$ does not follow a chi-squared distribution, except for small learning rate parameter values. For example, as the learning rate parameter increases, the distribution becomes more heavy-tailed (e.g., for $\gamma_1 = 6$), indicating that explicit SGD becomes unstable. Particularly for $\gamma_1 = 7$ explicit SGD diverged in almost all replications, and thus a histogram could not be constructed.

4.2. *Comparative performance analysis.* In this section, we aim to illustrate the performance of implicit SGD estimation against deterministic estimation procedures that are optimal. The goal is to investigate the extent to which implicit SGD can be as fast as deterministic methods, and to quantify how much statistical efficiency needs be sacrificed to accomplish that.

4.2.1. *Experiments with* `glm()` *function.* The built-in function `glm()` in `R` performs deterministic maximum-likelihood estimation through iterative reweighted least squares. In this experiment, we wish to compare computing time and MSE between first-order implicit SGD and `glm()`. Our simulated data set is a simple normal linear model constructed as follows. First, we sample a binary $p \times p$ design matrix $X = (x_{ij})$ such that $x_{i1} = 1$ (intercept) and $P(x_{ij} = 1) = s$ i.i.d., where $s \in (0, 1)$ determines the sparsity of $X$. We set $s = 0.08$ indicating that roughly 8% of the $X$ matrix will be nonzero. We generate $\theta_\star$ by sampling $p$ elements from $(-1, -0.35, 0, 0.35, 1)$ with replacement. The outcomes are $Y_i = X_i^\intercal \theta_\star + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, 1)$ i.i.d., and $X_i = (x_{ij})$ is the $p \times 1$ vector of $i$'s covariates. By GLM properties,

$$\mathcal{I}(\theta_\star) = \mathbb{E}\big(h'(X_i^\intercal \theta_\star) X_i X_i^\intercal\big) = \begin{pmatrix} 1 & s & s & \cdots & s \\ s & s & s^2 & \cdots & s^2 \\ s & s^2 & s & s^2 & \cdots \\ \cdots & s^2 & \cdots & s & \cdots \\ s & s^2 & \cdots & \cdots & s \end{pmatrix}.$$

Slightly tedious algebra can show that the eigenvalues of $\mathcal{I}(\theta_\star)$ are $s(1 - s)$ with multiplicity $(p - 2)$ and the two solutions of $x^2 - A(s)x + B(s) = 0$, where $A(s) = 1 + s + s^2(p - 2)$ and $B(s) = s(1 - s)$. It is thus possible to use the analysis of Section 2.2 and equation (10) to derive a theoretically optimal learning rate. We sample 200 pairs $(p, N)$ for the problem size, uniformly in the ranges $p \sim [10, 500]$ and $N \sim [500, 50{,}000]$, and obtain running times and MSE of the estimates from implicit SGD and `glm()`. Finally, we then run a regression of computing time and MSE against the problem size $(N, p)$.

The results are shown in Table 1. We observe that implicit SGD scales better in both sample size $N$, and especially in the model size $p$. We also observe that this significant computational gain does not come with much efficiency loss. In fact, averaged over all samples, the MSE of the implicit SGD is 10% higher than the MSE of `glm()`, with a standard error of ±0.005. Furthermore, the memory

TABLE 1

*Parameters from regressing computation time and MSE against* $(N, p)$ *in* log-*scale for* `glm()` *and implicit GLM. Computation time for* `glm()` *is roughly* $O(p^{1.47}N)$ *and for implicit SGD, it is* $O(p^{0.2}N^{0.9})$. *Implicit SGD scales better in parameter dimension* $p$, *whereas MSE for both methods are comparable, at the order of* $O(\sqrt{p/N})$

| | **Time (sec)** | | **MSE** | |
|---|---|---|---|---|
| **Method** | **log $p$ (se)** | **log $N$ (se)** | **log $p$ (se)** | **log $N$ (se)** |
| `glm()` function | 1.46 (0.019) | 1.03 (0.02) | 0.52 (0.007) | −0.52 (0.006) |
| implicit SGD | 0.19 (0.012) | 0.9 (0.01) | 0.58 (0.007) | −0.53 (0.006) |

TABLE 2

*Comparison of implicit SGD with* `biglm`. *MSE is defined as* $\|\theta_N - \theta_\star\|/\|\theta_0 - \theta_\star\|$. *Values "*" indicate out-of-memory errors.* `biglm` *was run in combination with the* `ffdf` *package to map big data files to memory. Implicit SGD used a similar but slower ad-hoc method. The table reports computation times excluding file access*

| | | | **Procedure** | | | |
|---|---|---|---|---|---|---|
| | | | **`biglm`** | | **Implicit SGD** | |
| ***p*** | ***N*** | **Size (GB)** | **Time (secs)** | **MSE** | **Time (secs)** | **MSE** |
| 1e2 | 1e5 | 0.021 | 2.32 | 0.028 | 2.4 | 0.028 |
| 1e2 | 5e5 | 0.103 | 8.32 | 0.012 | 7.1 | 0.012 |
| 1e2 | 1e6 | 0.206 | 16 | 0.008 | 14.7 | 0.009 |
| 1e2 | 1e7 | 2.1 | 232 | 0.002 | 127.9 | 0.002 |
| 1e2 | 1e8 | 20.6 | * | * | 1397 | 0.00 |
| 1e3 | 1e6 | 2.0 | * | * | 31.38 | 0.153 |
| 1e4 | 1e5 | 2.0 | * | * | 25.05 | 0.160 |

requirements (not reported in Table 1) are roughly $O(Np^2)$ for `glm()` and only $O(p)$ for implicit SGD.

4.2.2. *Experiments with* `biglm`. The package `biglm` is a popular choice for fitting GLMs with data sets where $N$ is large but $p$ is small.[6] It works in an iterative way by splitting the data set in many parts, and by updating the model parameters using incremental QR decomposition [Miller (1992)], which results in only $O(p^2)$ memory requirement. In this experiment, we compare implicit SGD with `biglm` on larger data sets of Section 4.2.1. with small $p$ and large $N$ such that $Np$ remains roughly constant.

The results are shown in Table 2. We observe that implicit SGD is significantly faster at a very small efficiency loss. The difference is more dramatic at large $p$; for example, when $p = 10^3$ or $p = 10^4$, `biglm` quickly runs out of memory, whereas implicit SGD works without problems.

4.2.3. *Experiments with* `glmnet`. The `glmnet` package in R [Friedman, Hastie and Tibshirani (2010)] is a deterministic optimization algorithm for generalized linear models that uses the elastic net. It performs a component-wise update of the parameter vector, utilizing thresholding from the regularization penalties for more computationally efficient updates. One update over all parameters costs roughly $O(Np)$ operations. Additional computational gains are achieved when the design matrix is sparse because fewer components are updated per each iteration.

[6] See http://cran.r-project.org/web/packages/biglm/index.html for the `biglm` package. `biglm` is part of the High-Performance Computing (HPC) task view of the CRAN project here http://cran.r-project.org/web/views/HighPerformanceComputing.html.

TABLE 3

*Comparing implicit SGD with* `glmnet`. *Table reports running times* (*in secs.*) *and MSE for both procedures. The MSE of* `glmnet` *is calculated as the median MSE over the* 100 *grid values of regularization parameter computed by default* [*Friedman, Hastie and Tibshirani* (2010)]

| | | **Correlation ($\rho$)** | | | |
|---|---|---|---|---|---|
| **Method** | **Metric** | **0** | **0.2** | **0.6** | **0.9** |
| | | $N = 1000$, $p = 10$ | | | |
| `glmnet` | time (sec) | 0.005 | 0.005 | 0.008 | 0.022 |
| | mse | 0.083 | 0.085 | 0.099 | 0.163 |
| `sgd` | time (sec) | 0.011 | 0.011 | 0.011 | 0.011 |
| | mse | 0.042 | 0.042 | 0.049 | 0.053 |
| | | $N = 5000$, $p = 50$ | | | |
| `glmnet` | | 0.058 | 0.067 | 0.119 | 0.273 |
| | | 0.044 | 0.046 | 0.057 | 0.09 |
| `sgd` | | 0.059 | 0.056 | 0.057 | 0.057 |
| | | 0.019 | 0.02 | 0.023 | 0.031 |
| | | $N = 100{,}000$, $p = 200$ | | | |
| `glmnet` | | 2.775 | 3.017 | 4.009 | 10.827 |
| | | 0.017 | 0.017 | 0.021 | 0.033 |
| `sgd` | | 1.475 | 1.464 | 1.474 | 1.446 |
| | | 0.004 | 0.004 | 0.004 | 0.006 |

In this experiment, we compare implicit SGD with `glmnet` on a subset of experiments in the original package release [Friedman, Hastie and Tibshirani (2010)]. In particular, we implement the experiment of Section 5.1 in that paper, as follows. First, we sample the design matrix $X \sim \mathcal{N}_p(0, \Sigma)$, where $\Sigma = b^2 U + I$ and $U$ is the $p \times p$ matrix of ones. The parameter $b = \sqrt{\rho/(1-\rho)}$, where $\rho$ is the target correlation of columns of $X$, is controlled in the experiments. The outcomes are $Y = X\theta_\star + \sigma^2 \varepsilon$, where $\theta_j^* = (-1)^j \exp(-2(j-1)/20)$, and $\varepsilon$ is a standard $p$-variate normal. The parameter $\sigma$ is tuned to achieve a predefined signal-noise ratio. We report average computation times in Table 3 over 10 replications, which expands Table 1 of Friedman, Hastie and Tibshirani (2010).

First, we observe that implicit SGD is consistently faster than the `glmnet` method. In particular, the SGD method scales better at larger $p$ following a sublinear growth as noted in Section 4.2.1. Interestingly, it is also not affected by covariate correlation, whereas `glmnet` gets slower as more components need to be updated at every iteration. For example, with correlation $\rho = 0.9$ and $N = 1e5$, $p = 200$, the SGD method is almost 10× faster.

Second, to compare `glmnet` with implicit SGD in terms of MSE we picked the median MSE produced by the grid of regularization parameters computed by `glmnet`. We picked the median because `glmnet` is a deterministic method and so at the best regularization value its MSE will be lower than the MSE of implicit

SGD. However, implicit SGD seems to perform better against the median performance of `glmnet`. Furthermore, Table 3 indicates a clear trend where, for bigger dimensions $p$ and higher correlation $\rho$, implicit SGD is performing better than `glmnet` in terms of efficiency as well. We obtain similar results in a comparison on a logistic regression model, which we present in Section 3 of the supplemental article [Toulis and Airoldi (2017)].

4.2.4. *Cox proportional hazards.* In this experiment, we test the performance of implicit SGD on estimating the parameters of a Cox proportional hazards model in a setup that is similar to the numerical example of Simon et al. (2011), Section 3.

We consider $N = 1000$ units with covariates $X \sim \mathcal{N}(0, \Sigma)$, where $\Sigma = 0.2U + I$, and $U$ is the matrix of ones. We sample times as $Y_i \sim \text{Expo}(\eta_i(\theta_\star))$, where $\eta_i(\theta) = \exp(X_i^\intercal \theta)$, and $\theta_\star = (\theta_{\star,k})$ is a vector with $p = 20$ elements defined as $\theta_{\star,k} = 2(-1)^{-k} \exp(-0.1k)$. Time $Y_i$ is censored, and thus $d_i = 0$, according to probability $\left(1 + \exp(-a(Y_i - q))^{-1}\right)$, where $q$ is a quantile of choice (set here as $q = 0.8$), and $a$ is set such that $\min\{Y_i\}$ is censored with a prespecified probability (set here as 0.1%). We replicate 50 times the following process. First, we run implicit SGD for $2N$ iterations, and then measure MSE $\|\theta_n^{\text{im}} - \theta_\star\|^2$, for all $n = 1, 2, \ldots, 2N$. To set the learning rates, we use equation (10), where the Fisher matrix is diagonally approximated, through the AdaGrad procedure (12). We then take the 5%, 50% and 95% quantiles of MSE across all repetitions and plot them against iteration number $n$.

The results are shown in Figure 3 (left panel). In the figure, we also plot (horizontal dashed lines) the 5% and 95% quantiles of the MSE of the MLE, assumed to be the best MSE achievable for SGD. We observe that implicit SGD performs well compared to MLE in this small-sized problem. In particular, implicit SGD, under the aforementioned generic tuning of learning rates, converges to the region of optimal MLE in a few thousands of iterations. In experiments with explicit SGD,

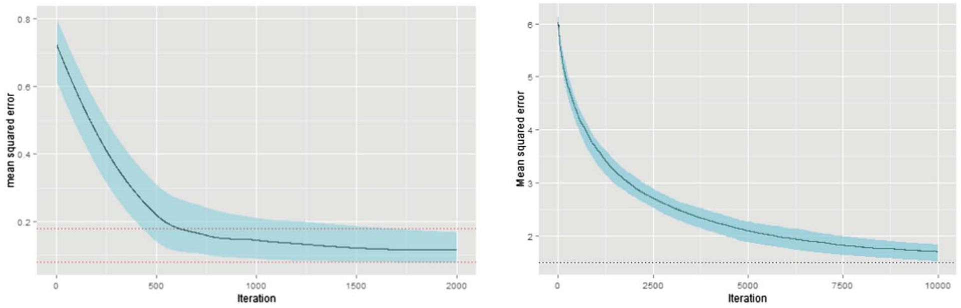

FIG. 3. *Left panel*: 5%–95% *quantile band of implicit SGD estimates* (*in cyan*) *against* 5%–95% *band of the MLE* (*dashed lines*) *for a Cox proportional hazards model* (50 *replications*); *Right panel*: 5%–95% *quantile band of implicit SGD estimates* (*in cyan*) *against median MLE* (*dashed line*) *on an M-estimation task* (100 *replications*).

we were not able to replicate this performance because of numerical instability. We note that there are no standard implementations of explicit SGD for estimating Cox proportional hazards models, to our best knowledge.

4.2.5. *M-estimation.* In this experiment, we test the performance of implicit SGD, in particular Algorithm 5, on a M-estimation problem in a setup that is similar to the simulation example of Donoho and Montanari (2016), Example 2.4.

We set $N = 1000$ data points and $p = 200$ as the parameter dimension. We sample $\theta_\star$ as a random vector with norm $\|\theta_\star\| = 6\sqrt{p}$, and sample the design matrix as $X \sim \mathcal{N}(0, (1/N)I)$. The outcomes are sampled i.i.d. from a contaminated normal distribution, that is, with probability 95%, $Y_n \sim \mathcal{N}(X_n^\intercal \theta_\star, 1)$ and $Y_n = 10$ with probability 5%.

The results over 2000 iterations of implicit SGD are shown in Figure 3 (right panel). In the figure, we plot the 5% and 95% quantiles of MSE of implicit SGD over 100 replications of the experiment. We also plot (horizontal dashed line) the median MSE of the MLE estimator, computed using the `coxph` built-in command of `R`. We observe that SGD converges steadily to the best possible MSE. Similar behavior was observed under various modifications of the simulation parameters.

4.3. *National morbidity-mortality air pollution* (*NMMAPS*) *study.* The NMMAPS study [Samet et al. (2000), Dominici et al. (2002)] analyzed the risks of air pollution to public health. Several cities (108 in the US) are included in the study with daily measurements covering more than 13 years (roughly 5000 days) including air pollution data (e.g., concentration of CO in the atmosphere) together with health outcome variables such as number of respiratory-related deaths.

The original study fitted a Poisson generalized additive model (GAM), separately for each city due to data set size. Recent research [Wood, Goude and Shaw (2015)] has developed procedures similar to `biglm`'s iterative QR decomposition to fit all cities simultaneously on the full data set with approximately $N = 1.2$ million observations and $p = 802$ covariates (7 Gb in size). In this experiment, we construct a GAM model using data from all cities in the NMMAPS study in a process that is very similar (but not identical) to the data set of Wood, Goude and Shaw (2015).

Our final data set has $N = 1{,}426{,}806$ observations and $p = 794$ covariates including all cities in the NMMAPS study (8.6 GB in size), and is fit using a simple first-order implicit SGD procedure with $C_n = I$ and $\gamma_1 = 1$. The runtime for implicit SGD was roughly 120 seconds, which is 6x faster than the 12 minutes reported by Wood, Goude and Shaw (2015) on a similar computer. We cannot directly compare the estimates from the two procedures because the datasets used were different. However, we can compare the estimates of our model with the estimates of `glm()` on a random small subset of the data. For that purpose, we subsampled $N = 50{,}000$ observations and $p = 50$ covariates (19.5 MB in size) and fit the smaller data set using implicit SGD and `glm()`. A scatter plot of the

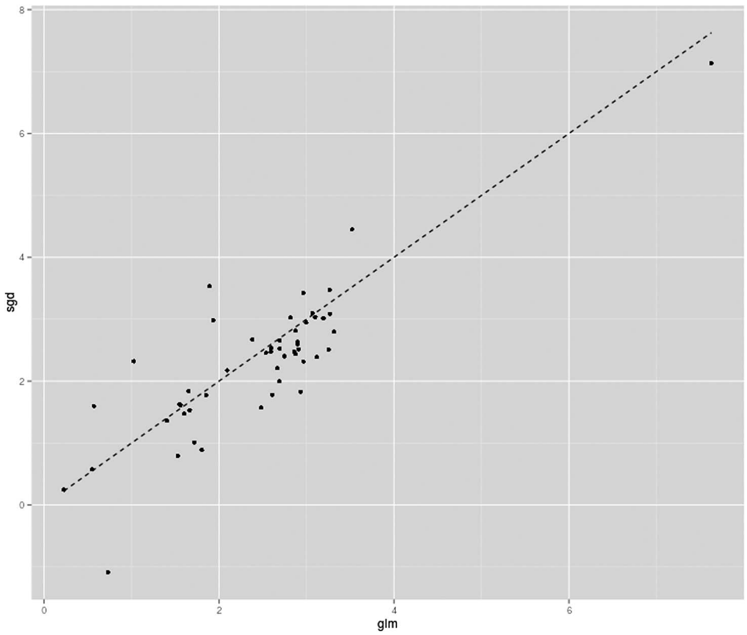

FIG. 4. *Estimates of implicit SGD* ($y$*-axis*) *and* `glm()` ($x$*-axis*) *on a subset of the NMMAPS data set with* $N = 50{,}000$ *observations and* $p = 50$ *covariates, which is roughly* 5% *of the entire data set.*

estimates is shown in Figure 4. The estimates of the implicit of the SGD procedure are very close to MLE, while further replications of the aforementioned testing process revealed the same pattern indicating that implicit SGD converged on all replications.

**5. Discussion.** The theory in Section 2 suggests that implicit SGD is numerically stable and has known asymptotic variance and asymptotic distribution. The experiments in Section 4 show that the empirical properties of SGD are well predicted by theory. In contrast, explicit SGD is unstable and cannot work well without problem-specific tuning. Thus, we conclude that implicit SGD is a principled estimation procedure and is superior to widely-used explicit SGD procedures.

Intuitively, implicit SGD leverages second-order information at every iteration, although second-order quantities do not need to be computed in equation (4). To demonstrate this, we build upon the argument that was first introduced in Section 1 and equation (8), which is repeated below for convenience:

$$\Delta\theta_n^{\mathrm{im}} \approx \left[I + \gamma_n \hat{\mathcal{I}}(\theta_0; X_n, Y_n)\right]^{-1} \Delta\theta_n^{\mathrm{sgd}};$$

here, $\Delta\theta_n^{\mathrm{im}} = \theta_n^{\mathrm{im}} - \theta_0$ and $\Delta\theta_n^{\mathrm{sgd}} = \theta_n^{\mathrm{sgd}} - \theta_0$, and the matrix $\hat{\mathcal{I}}(\theta_0; X_n, Y_n) = -\nabla^2 \log f(Y_n; X_n, \theta)|_{\theta=\theta_0}$ is the observed Fisher information at $\theta_0$. In other words, the implicit procedure is a *shrinked* version of the explicit one, where the shrinkage factor depends on the observed information.

Naturally, the implicit SGD iterate $\theta_n^{\mathrm{im}}$ has also a Bayesian interpretation. In particular, $\theta_n^{\mathrm{im}}$ is the posterior mode of a Bayesian model defined as

$$\begin{aligned} &\theta | \theta_{n-1}^{\mathrm{im}} \sim \mathcal{N}(\theta_{n-1}^{\mathrm{im}}, \gamma_n C_n), \\ &Y_n | X_n, \theta \sim f(\cdot; X_n, \theta). \end{aligned} \tag{28}$$

The explicit SGD update $\theta_n^{\mathrm{sgd}}$ can be written as in equation (28); however, $f$ needs to be substituted with its linear approximation around $\theta_{n-1}^{\mathrm{sgd}}$. Thus, equation (28) provides an alternative explanation why implicit SGD is more principled than explicit SGD. Furthermore, it indicates possible improvements for implicit SGD. For example, the prior in equation (28) could be chosen to fit better the parameter space (e.g., $\theta_\star$ being on the simplex). Krzysztof et al. (2007) and Nemirovski et al. (2008) have argued that appropriate implicit updates can fit better in the geometry of the parameter space, and thus converge faster. Setting up the parameters of the prior is also crucial. Whereas in explicit SGD there is no statistical intuition behind learning rates $\gamma_n$, equation (28) reveals that in implicit SGD the terms $(\gamma_n C_n)^{-1}$ encode the statistical information up to iteration $n$. It follows immediately that it is optimal, in general, to set $\gamma_n C_n = \mathcal{I}(\theta_\star)^{-1}/n$, which is a special case of Theorem 2.2.

The Bayesian formulation of equation (28) also explains the stability of implicit SGD. In Theorem 2.1, we showed that the initial conditions are discounted at an exponential rate, regardless of misspecification of the learning rates. This stability of implicit SGD allows several ideas for improvements. For example, constant learning rates could be used in implicit SGD to speed up convergence toward a region around $\theta_\star$. A sequential hypothesis test could decide on whether $\theta_n^{\mathrm{im}}$ has reached that region or not, and switch to the theoretically optimal $1/n$ rate accordingly. Alternatively, we could run implicit SGD with AdaGrad learning rates and switch to $1/n$ rates when the theoretical $\mathrm{O}(1/\sqrt{n})$ variance of AdaGrad becomes larger than the $\mathrm{O}(1/n)$ variance of implicit SGD. Such schemes using constant rates with explicit SGD are very hard to do in practice because of instability.

Regarding statistical efficiency, a key technical result in this paper is that the asymptotic variance of implicit SGD can be calculated exactly using Theorem 2.2. Optimal learning rates were suggested in equation (10) that depend on the eigenvalues of the unknown Fisher matrix $\mathcal{I}(\theta_\star)$. In this paper, we used second-order procedures of Section 2.2.1 to iteratively estimate the eigenvalues, however better methods are certainly possible and could improve the performance of implicit SGD. For example, it is known that typical iterative methods usually overestimate the largest eigenvalue and underestimate the smallest eigenvalue, in small-to-moderate samples. This crucially affects the behavior of stochastic approximations with learning rates that depend on sample eigenvalues. Empirical Bayes methods have been shown to be superior in iterative estimation of eigenvalues of large matrices [Mestre (2008)], and it would be interesting to apply such methods to design the learning rates of implicit SGD procedures.

Regarding computational efficiency, we developed Algorithm 1 which implements implicit SGD on a large family of statistical models. However, the trick used in fitting the Cox proportional hazards model in Section 3.3 can be more generally applied to models outside this family. For example, assume a log-likelihood gradient of the form $s(X^{\intercal}\theta; Y)G(\theta; X, Y)$, where both its scale $s(\cdot)$ and direction $G(\cdot)$ depend on model parameters $\theta$; this violates conditions of Assumption 2.1(b). The implicit update in equation (4)—where $C_n = I$ for simplicity—would be $\theta_n^{\mathrm{im}} = \theta_{n-1}^{\mathrm{im}} + \gamma_n s(X_n^{\intercal}\theta_n^{\mathrm{im}}; Y_n)G(\theta_n^{\mathrm{im}}; X_n, Y_n)$, which cannot be computed by Algorithm 1. One way to circumvent this problem is to use an implicit update only on the scale and use an explicit update on the direction, that is, $\theta_n^{\mathrm{im}} = \theta_{n-1}^{\mathrm{im}} + \gamma_n s(X_n^{\intercal}\theta_n^{\mathrm{im}}; Y_n)G(\theta_{n-1}^{\mathrm{im}}; X_n, Y_n)$. This form of updates expands the applicability of implicit SGD.

Finally, hypothesis testing and construction of confidence intervals using SGD estimates is an important issue that has remained unexplored. In experiments of Section 4.1.2, we showed that implicit SGD is indeed asymptotically normal in several simulation scenarios. However, as SGD procedures are iterative, there needs to be a rigorous and general method to decide whether SGD iterates have converged to the asymptotic regime. Several methods, such as bootstrapping the data set, could be used for that. Furthermore, conservative confidence intervals could be constructed through multivariate Chebyshev inequalities or other strategies [Marshall and Olkin (1960)].

5.1. *Concluding remarks.* In this paper, we introduced a new stochastic gradient descent procedure that uses implicit updates at every iteration, which we termed implicit SGD. Equation (8) shows, intuitively, that the iterates of implicit SGD are a shrinked version of the standard iterates, where the shrinkage factor depends on the observed Fisher information matrix. Thus, implicit SGD combines the computational efficiency of first-order methods with the numerical stability of second-order methods.

In a theoretical analysis, we derived nonasymptotic upper bounds for the mean-squared errors of implicit SGD iterates, and the asymptotic variance of both explicit and implicit SGD iterates. Our analysis quantifies the efficiency loss of SGD procedures, and suggests principled strategies to calibrate a hyperparameter that is common to both explicit and implicit SGD procedures, known as the learning rate. We illustrated the use of implicit SGD for statistical estimation in generalized linear models, Cox proportional hazards model, and general M-estimation problems.

Viewed as statistical estimation procedures, our results suggest that implicit SGD has the same asymptotic efficiency to explicit SGD. However, the implicit procedure is significantly more stable than the explicit one with respect to misspecification of the learning rate. In general, explicit SGD procedures are sensitive to outliers and to misspecification of the learning rates, making it impossible to apply without problem-specific tuning. In theory and in extensive experiments, implicit procedures emerge as principled iterative estimation methods because they are numerically stable, they are robust to tuning of hyperparameters, and their standard errors are well predicted by theory. Thus, implicit stochastic gradient descent is poised to become a workhorse of estimation from large data sets in statistical practice.

**Acknowledgments.** The authors wish to thank Joe Blitzstein, Leon Bottou, Bob Carpenter, David Dunson, Andrew Gelman, Brian Kulis, Xiao-Li Meng, Natesh Pillai and Neil Shephard for useful comments and discussion. We are grateful to four reviewers, the Associate Editor, and the Editor for offering suggestions that improved the paper.

## SUPPLEMENTARY MATERIAL

**Supplement to "Asymptotic and finite-sample properties of estimators based on stochastic gradients"** (DOI: 10.1214/16-AOS1506SUPP; .pdf). The proofs of all technical results are provided in an online supplement [Toulis and Airoldi (2017)]. There, we also provide numerical results that extend the results in Section 4 of this article—referred to as the "main paper" in the supplement.

## REFERENCES

AMARI, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Comput.* **10** 251–276.

AMARI, S.-I., PARK, H. and FUKUMIZU, K. (2000). Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Comput.* **12** 1399–1409.

BACH, F. and MOULINES, E. (2013). Non-strongly-convex smooth stochastic approximation with convergence rate $o(1/n)$. In *Advances in Neural Information Processing Systems* 773–781.

BATHER, J. A. (1989). Stochastic approximation: A generalisation of the Robbins–Monro procedure. In *Proceedings of the Fourth Prague Symposium on Asymptotic Statistics* (*Prague*, 1988) 13–27. Charles Univ., Prague. MR1051424

BECK, A. and TEBOULLE, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2** 183–202. MR2486527

BENVENISTE, A., MÉTIVIER, M. and PRIOURET, P. (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer, Berlin. MR1082341

BERTSEKAS, D. P. (2011). Incremental proximal methods for large scale convex optimization. *Math. Program.* **129** 163–195. MR2837879

BORDES, A., BOTTOU, L. and GALLINARI, P. (2009). SGD-QN: Careful quasi-Newton stochastic gradient descent. *J. Mach. Learn. Res.* **10** 1737–1754. MR2534877

BORKAR, V. S. (2008). *Stochastic Approximation*. Cambridge Univ. Press, Cambridge. MR2442439

BOTTOU, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'*2010 177–186. Springer, Heidelberg. MR3362066

BYRD, R. H., HANSEN, S. L., NOCEDAL, J. and SINGER, Y. (2016). A stochastic quasi-Newton method for large-scale optimization. *SIAM J. Optim.* **26** 1008–1031. MR3485979

CAPPÉ, O. and MOULINES, E. (2009). On-line expectation-maximization algorithm for latent data models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **71** 593–613. MR2749909

CHEN, H. F., LEI, G. and GAO, A. J. (1988). Convergence and robustness of the Robbins–Monro algorithm truncated at randomly varying bounds. *Stochastic Process. Appl.* **27** 217–231. MR0931029

CHUNG, K. L. (1954). On a stochastic approximation method. *Ann. Math. Stat.* **25** 463–483. MR0064365

COX, D. R. (1972). Regression models and life-tables. *J. Roy. Statist. Soc. Ser. B* **34** 187–220. MR0341758

DAVISON, A. C. (2003). *Statistical Models. Cambridge Series in Statistical and Probabilistic Mathematics* **11**. Cambridge Univ. Press, Cambridge. MR1998913

DEAN, J., CORRADO, G., MONGA, R., CHEN, K., DEVIN, M., MAO, M., SENIOR, A., TUCKER, P., YANG, K., LE, Q. V. et al. (2012). Large scale distributed deep networks. In *Advances in Neural Information Processing Systems* 1223–1231.

DEMPSTER, A. P., LAIRD, N. M. and RUBIN, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B* **39** 1–38. MR0501537

DOMINICI, F., DANIELS, M., ZEGER, S. L. and SAMET, J. M. (2002). Air pollution and mortality: Estimating regional and national dose-response relationships. *J. Amer. Statist. Assoc.* **97** 100–111. MR1963390

DONOHO, D. and MONTANARI, A. (2016). High dimensional robust M-estimation: Asymptotic variance via approximate message passing. *Probab. Theory Related Fields* **166** 935–969. MR3568043

DUCHI, J., HAZAN, E. and SINGER, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **12** 2121–2159. MR2825422

DUCHI, J. and SINGER, Y. (2009). Efficient online and batch learning using forward backward splitting. *J. Mach. Learn. Res.* **10** 2899–2934. MR2579916

EL KAROUI, N. (2008). Spectrum estimation for large dimensional covariance matrices using random matrix theory. *Ann. Statist.* **36** 2757–2790. MR2485012

FABIAN, V. (1968). On asymptotic normality in stochastic approximation. *Ann. Math. Stat.* **39** 1327–1332. MR0231429

FABIAN, V. (1978). On asymptotically efficient recursive estimation. *Ann. Statist.* **6** 854–866. MR0478506

FISHER, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **222** 309–368.

FISHER, R. A. (1925). *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* **33** 1–22.

GEORGE, A. P. and POWELL, W. B. (2006). Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Mach. Learn.* **65** 167–198.

GREEN, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *J. Roy. Statist. Soc. Ser. B* **46** 149–192. MR0781879

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, New York. MR2722294

HENNIG, P. and KIEFEL, M. (2013). Quasi-Newton methods: A new direction. *J. Mach. Learn. Res.* **14** 843–865. MR3049491

HOFFMAN, J. D. and FRANKEL, S. (2001). *Numerical Methods for Engineers and Scientists*. CRC press, Boca Raton, FL.

HUBER, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Stat.* **35** 73–101. MR0161415

JAIN, P., TEWARI, A. and KAR, P. (2014). On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems* 685–693.

KLEIN, J. P. and MOESCHBERGER, M. L. (2003). *Survival Analysis: Techniques for Censored and Truncated Data*. Springer Science & Business Media New York.

KRAKOWSKI, K. A., MAHONY, R. E., WILLIAMSON, R. C. and WARMUTH, M. K. (2007). A geometric view of non-linear on-line stochastic gradient descent. Available at https://users.soe.ucsc.edu/ manfred/pubs/T3.pdf.

LANGE, K. (2010). *Numerical Analysis for Statisticians*, 2nd ed. Springer, New York. MR2655999

LEHMANN, E. L. and CASELLA, G. (1998). *Theory of Point Estimation*, 2nd ed. *Springer Texts in Statistics* **31**. Springer, New York. MR1639875

LIONS, P.-L. and MERCIER, B. (1979). Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* **16** 964–979. MR0551319

LJUNG, L., PFLUG, G. and WALK, H. (1992). *Stochastic Approximation and Optimization of Random Systems*. *DMV Seminar* **17**. Birkhäuser, Basel. MR1162311

MARSHALL, A. W. and OLKIN, I. (1960). Multivariate Chebyshev inequalities. *Ann. Math. Stat.* **31** 1001–1014. MR0119234

MARTIN, R. D. and MASRELIEZ, C. J. (1975). Robust estimation via stochastic approximation. *IEEE Trans. Inform. Theory* **IT-21** 263–271. MR0395111

MESTRE, X. (2008). Improved estimation of eigenvalues and eigenvectors of covariance matrices using their sample estimates. *IEEE Trans. Inform. Theory* **54** 5113–5129. MR2589886

MILLER, A. J. (1992). Algorithm as 274: Least squares routines to supplement those of gentleman. *Applied Statistics* 458–478.

MOULINES, E. and BACH, F. R. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems* 451–459.

NAGUMO, J.-I. and NODA, A. (1967). A learning method for system identification. *IEEE Trans. Automat. Control* **12** 282–287.

NATIONAL RESEARCH COUNCIL (2013). *Frontiers in Massive Data Analysis*. The National Academies Press, Washington, DC.

NELDER, J. A. and WEDDERBURN, R. W. M. (1972). Generalized linear models. *J. Roy. Statist. Soc. Ser. A* **135** 370–384.

NEMIROVSKI, A., JUDITSKY, A., LAN, G. and SHAPIRO, A. (2008). Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.* **19** 1574–1609. MR2486041

NEMIROVSKY, A. S. and YUDIN, D. B. (1983). *Problem Complexity and Method Efficiency in Optimization*. Wiley, New York. MR0702836

NEVELSON, M. B. and KHASMINSKIĬ, R. Z. (1973). *Stochastic Approximation and Recursive Estimation* **47**. Amer. Math. Society, Washington.

PARIKH, N. and BOYD, S. (2013). Proximal algorithms. *Found. Trends Optim.* **1** 123–231.

POLYAK, B. T. and JUDITSKY, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.* **30** 838–855. MR1167814

POLJAK, B. T. and TSYPKIN, JA. Z. (1980). Robust identification. *Automatica J. IFAC* **16** 53–63. MR0571554

POLYAK, B. T. and TSYPKIN, YA. Z. (1979). Adaptive estimation algorithms (convergence, optimality, stability). *Avtomat. i Telemekh.* **3** 71–84. MR0544876

ROBBINS, H. and MONRO, S. (1951). A stochastic approximation method. *Ann. Math. Stat.* **22** 400–407. MR0042668

ROCKAFELLAR, R. T. (1976). Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* **14** 877–898. MR0410483

ROSASCO, L., VILLA, S. and VŨ, B. C. (2014). Convergence of stochastic proximal gradient algorithm. Preprint. Available at arXiv:1403.5074.

RUPPERT, D. (1988). Efficient estimations from a slowly convergent Robbins–Monro process. Technical report, Dept. Operations Research and Industrial Engineering, Cornell Univ., Ithaca, NY.

SACKS, J. (1958). Asymptotic distribution of stochastic approximation procedures. *Ann. Math. Stat.* **29** 373–405. MR0098427

SAKRISON, D. J. (1965). Efficient recursive estimation; application to estimating the parameters of a covariance function. *Internat. J. Engrg. Sci.* **3** 461–483. MR0182082

SAMET, J. M., ZEGER, S. L., DOMINICI, F., CURRIERO, F., COURSAC, I., DOCKERY, D. W., SCHWARTZ, J. and ZANOBETTI, A. (2000). The national morbidity, mortality, and air pollution study. Part II: Morbidity and mortality from air pollution in the United States. *Res*. *Rep*. *Health Eff*. *Inst*. **94** 5–79.

SCHMIDT, M., LE ROUX, N. and BACH, F. (2013). Minimizing finite sums with the stochastic average gradient. Technical report, HAL 00860051.

SHAMIR, O. and ZHANG, T. (2012). Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. Preprint. Available at arXiv:1212.1824.

SIMON, N., FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2011). Regularization paths for Cox's proportional Hazards model via coordinate descent. *J*. *Stat*. *Softw*. **39** 1–13.

SINGER, Y. and DUCHI, J. C. (2009). Efficient learning using forward-backward splitting. In *Advances in Neural Information Processing Systems* 495–503.

SLOCK, D. T. M. (1993). On the convergence behavior of the lms and the normalized lms algorithms. *IEEE Trans*. *Signal Process*. **41** 2811–2825.

TOULIS, P. and AIROLDI, E. M. (2015a). Scalable estimation strategies based on stochastic approximations: Classical results and new insights. *Stat*. *Comput*. **25** 781–795. MR3360492

TOULIS, P. and AIROLDI, E. M. (2015b). Implicit stochastic approximation. Preprint. Available at arXiv:1510.00967.

TOULIS, P. and AIROLDI, E. M. (2017). Supplement to "Asymptotic and finite-sample properties of estimators based on stochastic gradients." DOI:10.1214/16-AOS1506SUPP.

TOULIS, P., AIROLDI, E. M. and RENNIE, J. (2014). Statistical analysis of stochastic gradient methods for generalized linear models. *J*. *Mach*. *Learn*. *Res*. *W&CP* **32 (ICML)** 667–675.

TOULIS, P., TRAN, D. and AIROLDI, E. M. (2016). Towards stability and optimality in stochastic gradient descent. *J*. *Mach*. *Learn*. *Res*. *W&CP* **51 (AISTATS)**.

TRAN, D., TOULIS, P. and AIROLDI, E. M. (2015). Stochastic gradient descent methods for estimation with large data sets. Preprint. Available at arXiv:1509.06459.

WELLING, M. and TEH, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the* 28*th International Conference on Machine Learning* (*ICML*-11) 681–688.

WIDROW, B. and HOFF, M. E. (1960). Adaptive switching circuits. *Defense Technical Information Center*.

WOOD, S. N., GOUDE, Y. and SHAW, S. (2015). Generalized additive models for large data sets. *J*. *R*. *Stat*. *Soc*. *Ser*. *C*. *Appl*. *Stat*. **64** 139–155. MR3293922

XIAO, L. and ZHANG, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM J*. *Optim*. **24** 2057–2075. MR3285905

XU, W. (2011). Towards optimal one pass large scale learning with averaged stochastic gradient descent. Preprint. Available at arXiv:1107.2490.

ZHANG, T. (2004). Solving large scale linear prediction problems using gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning* 116. ACM, New York.

ECONOMETRICS AND STATISTICS
BOOTH SCHOOL OF BUSINESS
UNIVERSITY OF CHICAGO
CHICAGO, ILLINOIS 60637
USA
E-MAIL: panos.toulis@chicagobooth.edu

DEPARTMENT OF STATISTICS
HARVARD UNIVERSITY
CAMBRIDGE, MASSACHUSETTS 02138
USA
E-MAIL: airoldi@fas.harvard.edu