Towards stability and optimality in stochastic gradient descent

Panos Toulis
Harvard University

Dustin Tran
Harvard University

Edoardo M. Airoldi
Harvard University

Abstract

Iterative procedures for parameter estimation based on stochastic gradient descent (SGD) allow the estimation to scale to massive data sets. However, they typically suffer from numerical instability, while estimators based on SGD are statistically inefficient as they do not use all the information in the data set. To address these two issues we propose an iterative estimation procedure termed *averaged implicit* SGD *(*AI-SGD*)*. For statistical efficiency AI-SGD employs averaging of the iterates, which achieves the Cramér-Rao bound under strong convexity, i.e., it is asymptotically an optimal unbiased estimator of the true parameter value. For numerical stability AI-SGD employs an implicit update at each iteration, which is similar to updates performed by proximal operators in optimization. In practice, AI-SGD achieves competitive performance with state-of-the-art procedures. Furthermore, it is more stable than averaging procedures that do not employ proximal updates, and is simple to implement as it requires fewer tunable hyperparameters than procedures that do employ proximal updates.

1 Introduction

The majority of problems in statistical estimation can be cast as finding the parameter value $\theta_\star \in \Theta$ such that

$$\theta_\star = \arg\min_{\theta\in\Theta} \mathbb{E}\left(L(\theta, \xi)\right), \tag{1}$$

where the expectation is with respect to the random variable $\xi \in \Xi \subseteq \mathbb{R}^d$ that represents the data, $\Theta \subseteq \mathbb{R}^p$ is the parameter space, and $L : \Theta \times \Xi \to \mathbb{R}$ is a loss function. A popular procedure for solving Eq. (1) is stochastic gradient descent (SGD) (Zhang, 2004; Bottou, 2004, for example), where a sequence θ_n approximates $\theta_\star$, and is updated iteratively, one data point at a time, through the iteration

$$\theta_n = \theta_{n-1} - \gamma_n \nabla L(\theta_{n-1}, \xi_n), \tag{2}$$

where $\{\xi_1, \xi_2, \ldots\}$ is a stream of i.i.d. realizations of ξ, and $\{\gamma_n\}$ is a non-increasing sequence of positive real numbers, known as the learning rate. The nth iterate θ_n in SGD (2) can be viewed as an *estimator* of $\theta_\star$. To evaluate such estimators it is typical to consider three properties: convergence rate and numerical stability, by studying the mean-squared errors $\mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$; and statistical efficiency, by studying the limit $n\mathrm{Var}\left(\theta_n\right)$ as $n \to \infty$.

While computationally efficient, the SGD procedure (2) suffers from numerical instability and statistical inefficiency. Regarding stability, SGD is sensitive to specification of the learning rate γ_n since the mean-squared errors can diverge arbitrarily when γ_n is misspecified with the respect to problem parameters, e.g., the convexity and Lipschitz parameters of the loss function (Benveniste et al., 1990; Moulines and Bach, 2011). Several solutions have been proposed to resolve this issue, e.g., using projections or gradient clipping. However, they are typically heuristic and, thus, hard to generalize. Regarding statistical efficiency, SGD loses statistical information. In fact, the amount of information loss depends on the misspecification of γ_n with respect to the spectral gap of the matrix $\mathbb{E}\left(\nabla^2 L(\theta_\star, \xi)\right)$ (Toulis et al., 2014; Toulis and Airoldi, 2015), also known as the Fisher information matrix. To resolve this issue second-order information needs to be leveraged, but this sacrifices the computational simplicity of SGD procedures.

In this paper, we aim for the ideal combination of numerical stability, computational simplicity, and statistical efficiency using the following iterative procedure:

AI-SGD

$$\theta_n = \theta_{n-1} - \gamma_n \nabla L(\theta_n, \xi_n), \tag{3}$$

$$\bar{\theta}_n = (1/n)\sum_{i=1}^{n} \theta_i. \tag{4}$$

Our proposed procedure, termed *averaged implicit* SGD *(*AI-SGD*)*, is comprised of two inner procedures. The first procedure employs updates given in Eq. (3), which are *implicit* because the iterate θ_n appears on both sides of the equation. Procedure (3), also known as *implicit* SGD (Toulis et al., 2014), aims to stabilize the updates of the

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 41.

classic SGD procedure (2). In fact, implicit SGD can be motivated as the limit of a sequence of classic SGD procedures. To see this, first fix the sample history $\mathcal{F}_{n-1} = \{\theta_0, \xi_1, \xi_2, \ldots, \xi_{n-1}\}$. Then, the classic (not implicit) SGD procedure is $\theta_n = \theta_{n-1} - \gamma_n \nabla L(\theta_{n-1}, \xi_n) \triangleq \theta_n^{(1)}$. If we "trust" $\theta_n^{(1)}$ to be a better estimate of $\theta_\star$ than θ_{n-1}, then we can use $\theta_n^{(1)}$ instead of θ_{n-1} in computing the loss function at data point ξ_n. This leads to a revised procedure $\theta_n = \theta_{n-1} - \gamma_n \nabla L(\theta_n^{(1)}, \xi_n) \triangleq \theta_n^{(2)}$. Likewise, we can use $\theta_n^{(2)}$ instead of $\theta_n^{(1)}$, and so on. If we repeat this argument ad infinitum, then we get the following sequence of improved SGD procedures,

$$\begin{aligned}
\theta_n^{(1)} &= \theta_{n-1} - \gamma_n \nabla L(\theta_{n-1}, \xi_n), \\
\theta_n^{(2)} &= \theta_{n-1} - \gamma_n \nabla L(\theta_n^{(1)}, \xi_n), \\
\theta_n^{(3)} &= \theta_{n-1} - \gamma_n \nabla L(\theta_n^{(2)}, \xi_n), \\
&\cdots \\
\theta_n^{(\infty)} &= \theta_{n-1} - \gamma_n \nabla L(\theta_n^{(\infty)}, \xi_n). \qquad (5)
\end{aligned}$$

In the limit, assuming a unique fixed point is reached almost surely, the final procedure of sequence (5) can be rewritten as $\theta_n = \theta_{n-1} - \gamma_n \nabla L(\theta_n, \xi_n)$, which is identical to implicit SGD. Thus, implicit SGD can be viewed as a repeated application of classic SGD, where we keep updating the same iterate θ_{n-1} using the same data point ξ_n until a fixed point is reached. This idea is related to the so-called *self-consistency principle* in statistics; many statistical estimation methods, such as Expectation-Maximization or Rao-Blackwellisation, can be obtained by invoking the self-consistency principle (Tarpey and Flury, 1996).

The stability improvement achieved by implicit updates can be motivated by the following argument. Assume for simplicity that L is strongly convex almost surely with parameter $\mu > 0$. Then, we can derive a recursive inequality on the mean-squared error of implicit SGD from a rewrite of the implicit SGD procedure (3) as follows,

$$\begin{aligned}
\theta_n + \gamma_n \nabla L(\theta_n, \xi_n) &= \theta_{n-1}, \\
||\theta_n - \theta_\star||^2 + 2\gamma_n (\theta_n - \theta_\star)^\intercal \nabla L(\theta_n, \xi_n) &\leq ||\theta_{n-1} - \theta_\star||^2, \\
(1 + \gamma_n \mu)||\theta_n - \theta_\star||^2 &\leq ||\theta_{n-1} - \theta_\star||^2, \\
||\theta_n - \theta_\star||^2 &\leq \frac{1}{1 + \gamma_n \mu} ||\theta_{n-1} - \theta_\star||^2,
\end{aligned}$$

which implies that $||\theta_n - \theta_\star||^2$ is contracting almost surely. In contrast, the classic SGD procedure does not share this contracting property.

While the implicit update of Eq. (3) aims to achieve stability, the averaging of the iterates in Eq. (4) aims to achieve optimal statistical efficiency. Ruppert (1988) gave a nice intuition on why iterate averaging can lead to statistical optimality. When the learning rate is $\gamma_n \propto n^{-1}$, then $\bar{\theta}_n - \theta_\star$ is a weighted average of n error variables $\nabla L(\theta_{i-1}, \xi_i)$, which therefore are significantly autocorrelated. However, when $\gamma_n \propto n^{-\gamma}$ with $\gamma \in (0, 1)$, then $\bar{\theta}_n - \theta_\star$ is the average of $n^\gamma \log n$ error variables, which become uncorrelated in the limit. Thus, averaging improves the estimation accuracy.

1.1 Related work

The implicit update (3) is equivalent to solving

$$\theta_n = \arg\min_{\theta \in \Theta} \left\{ \frac{1}{2\gamma_n} ||\theta - \theta_{n-1}||^2 + L(\theta, \xi_n) \right\}. \qquad (6)$$

Arguably, the first method that used an update similar to (6) for estimation was the normalized least-mean squares filter of Nagumo and Noda (1967), used in signal processing. This update is also used by the *incremental proximal method* in optimization (Bertsekas, 2011), and has shown superior performance to classic SGD both in theory and applications (Bertsekas, 2011; Toulis et al., 2014; Défossez and Bach, 2015; Toulis and Airoldi, 2014). Overall, implicit updates lead to similar convergence rates as classic SGD updates, but are significantly more stable. This stability can also be motivated from a Bayesian interpretation of Eq. (6), where θ_n is the posterior mode of a model with the standard multivariate normal $\mathcal{N}(\theta_{n-1}, \gamma_n I)$ as the prior, $L(\theta, \cdot)$ as the log-likelihood, and ξ_n as the nth data sample.

A statistical analysis of procedure (3) without averaging was done by Toulis et al. (2014) who derived the asymptotic variance $\mathrm{Var}\,(\theta_n)$ of θ_n, and provided an algorithm to efficiently solve the fixed-point equation (3) for θ_n in the family of generalized linear models. In the online learning literature, Kivinen et al. (2006) and Kulis and Bartlett (2010) have also analyzed implicit updates in terms of regret; Schuurmans and Caelli (2007) have further applied implicit procedures on learning with kernels. Notably, the implicit update (6) is related to the importance weight updates proposed by Karampatziakis and Langford (2010), but the two update forms are not equivalent, and can be combined in practice (Karampatziakis and Langford, 2010, Section 5).

Assuming that the expected loss ℓ is known, instead of update (6) we could use the update

$$\theta_n^+ = \arg\min_{\theta \in \Theta} \left\{ \frac{1}{2\gamma_n} ||\theta - \theta_{n-1}||^2 + \ell(\theta) \right\}. \qquad (7)$$

In optimization this mapping from θ_{n-1} to θ_n^+ in Eq. (7) is known as a *proximal operator*, and the procedure is a special instance of the proximal point algorithm (Rockafellar, 1976). Thus, implicit SGD involves mappings that are stochastic versions of mappings from proximal operators. The stochastic proximal gradient algorithm (Singer and Duchi, 2009; Parikh and Boyd, 2013; Rosasco et al., 2014) is related but different to implicit SGD. In contrast

to implicit SGD, the stochastic proximal gradient algorithm first makes a classic SGD update (forward step) and then an implicit update (backward step). Only the forward step is stochastic whereas the backward proximal step is not. This may increase convergence speed but may also introduce instability due to the forward step. Interest on proximal operators has surged in recent years because they are non-expansive and converge with minimal assumptions. Furthermore, they can be applied on non-smooth objectives, and can easily be combined in modular algorithms for optimization in large-scale and distributed settings (Parikh and Boyd, 2013). The idea has also been generalized through splitting algorithms (Lions and Mercier, 1979; Beck and Teboulle, 2009; Singer and Duchi, 2009; Duchi et al., 2011). Krakowski et al. (2007) and Nemirovski et al. (2009) have shown that proximal methods can fit better in the geometry of the parameter space Θ, and Toulis and Airoldi (2014) have made a connection to shrinkage methods in statistics.

Two recent procedures based on stochastic proximal updates are PROX-SVRG (Xiao and Zhang, 2014) and PROX-SAG (Schmidt et al., 2013, Section 6). The main idea in both methods is to periodically compute an estimate of the full gradient averaged over all data points in order to reduce the variance of stochastic gradients. This requires a finite data setting, whereas AI-SGD also applies to streaming data. Moreover, the periodic calculations in PROX-SVRG are controlled by additional hyperparameters, and the periodic calculations in PROX-SAG require storage of the full gradient at every iteration. AI-SGD differs because it employs averaging to achieve statistical efficiency, has no additional hyperparameters or major storage requirements, and thus it has a simpler implementation.

Averaging of the iterates in Eq. (4) is the other key component of AI-SGD. Averaging was proposed and analyzed in the stochastic approximation literature by Ruppert (1988) and Bather (1989). Polyak and Juditsky (1992) substantially expanded the scope of the averaging method by proving asymptotic optimality of the classic SGD procedure with averaging under suitable assumptions. Their results showed clearly that slowly-convergent stochastic approximations (achieved, for example, when the learning rates are large) need to be averaged. Recent work has analyzed classic SGD with averaging (Zhang, 2004; Xu, 2011; Shamir and Zhang, 2012; Bach and Moulines, 2013) and has shown their superiority in numerous learning tasks.

1.2 Overview of results

In this paper, we study the iterates θ_n and use the results to study $\bar{\theta}_n$ as an estimator of $\theta_\star$. Under strong convexity of the expected loss, we give upper bounds for the squared errors $\mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$ and $\mathbb{E}\left(\bar{\theta}_n - \theta_\star||^2\right)$ in Theorem 1 and Theorem 2, respectively.

Two main results are derived from our theoretical analysis. First, $\bar{\theta}_n$ achieves the Cramér-Rao bound, i.e., no other unbiased estimator of $\theta_\star$ can do better in the limit, which is equivalent to the optimal $\mathcal{O}(1/n)$ rate of convergence for first-order procedures. Second, AI-SGD is significantly more stable to misspecification of the learning rate relative to classic averaged SGD procedures with respect to the learning problem parameters, e.g., convexity and Lipschitz constants. Finally, we perform experiments on several standard machine learning tasks, which show that AI-SGD comes closer to combining stability, optimality, and simplicity than other competing methods.

2 Preliminaries

Notation. Let $\mathcal{F}_n = \{\theta_0, \xi_1, \xi_2, \ldots, \xi_n\}$ denote the filtration that process θ_n (3) is adapted to. The norm $||\cdot||$ will denote the L_2 norm. The symbol $\triangleq$ indicates a definition, and the symbol $\stackrel{\text{def}}{=}$ denotes "equal by definition". For example, $x \triangleq y$ defines x as equal to known variable y, whereas $x \stackrel{\text{def}}{=} y$ denotes that the value of x is equal to the value of y, by definition. We will not use this formalism when defining constants. For two positive sequences a_n, b_n, we write $b_n = \mathcal{O}(a_n)$ if there exists a fixed $c > 0$ such that $b_n \leq ca_n$, for all n; also, $b_n = o(a_n)$ if $b_n/a_n \to 0$. When a positive scalar sequence a_n is monotonically decreasing to zero, we write $a_n \downarrow 0$. Similarly, for a sequence X_n of vectors or matrices, $X_n = \mathcal{O}(a_n)$ denotes that $||X_n|| = \mathcal{O}(a_n)$, and $X_n = o(a_n)$ denotes that $||X_n|| = o(a_n)$. For two matrices A, B, $A \preceq B$ denotes that $B - A$ is nonnegative-definite; $\mathrm{tr}(A)$ denotes the trace of A.

We now introduce the main assumptions pertaining to the theory of this paper.

Assumption 1. *The loss function $L(\theta, \xi)$ is almost-surely differentiable. The random vector ξ can be decomposed as $\xi = (x, y)$, $x \in \mathbb{R}^p, y \in \mathbb{R}^d$, such that*

$$L(\theta, \xi) \equiv L(x^\intercal \theta, y). \tag{8}$$

Assumption 2. *The learning rate sequence $\{\gamma_n\}$ is defined as $\gamma_n = \gamma_1 n^{-\gamma}$, where $\gamma_1 > 0$ and $\gamma \in (1/2, 1]$.*

Assumption 3 (Lipschitz conditions)**.** *For all $\theta_1, \theta_2 \in \Theta$, a combination of the following conditions is satisfied almost-surely:*

(a) The loss function L is Lipschitz with parameter λ_0, i.e.,

$$|L(\theta_1, \xi) - L(\theta_2, \xi)| \leq \lambda_0 ||\theta_1 - \theta_2||,$$

(b) The map ∇L is Lipschitz with parameter λ_1, i.e.,

$$||\nabla L(\theta_1, \xi) - \nabla L(\theta_2, \xi)|| \leq \lambda_1 ||\theta_1 - \theta_2||,$$

(c) The map $\nabla^2 L$ is Lipschitz with parameter λ_2, i.e.,

$$||\nabla^2 L(\theta_1, \xi) - \nabla^2 L(\theta_2, \xi)|| \leq \lambda_2 ||\theta_1 - \theta_2||.$$

Assumption 4. *The observed Fisher information matrix, $\hat{\mathcal{I}}(\theta) \triangleq \nabla^2 L(\theta, \xi)$, has non-vanishing trace, i.e., there exists $\phi > 0$ such that $\mathrm{tr}(\hat{\mathcal{I}}(\theta)) \geq \phi$, almost-surely, for all $\theta \in \Theta$. The expected Fisher information matrix, $\mathcal{I}(\theta) \triangleq \mathbb{E}\left(\hat{\mathcal{I}}(\theta)\right)$, has minimum eigenvalue $0 < \underline{\lambda_f} \leq \phi$, for all $\theta \in \Theta$.*

Assumption 5. *The zero-mean random variable $W_\theta \triangleq \nabla L(\theta, \xi) - \nabla \ell(\theta)$ is square-integrable, such that, for a fixed positive-definite Σ,*

$$\mathbb{E}\left(W_{\theta_\star} W_{\theta_\star}^\intercal\right) \preceq \Sigma.$$

Remarks. Assumption 1 puts a constraint on the loss function, which is not very restrictive because numerous machine learning models indeed depend on the parameter θ through a linear combination with features x. A notable exception includes loss functions with a regularization term. Although it is easy to add regularization to AI-SGD we will not do so in this paper because AI-SGD works well without it, since the proximal operator (6) already regularizes the estimate θ_n towards θ_{n-1}. In experiments, regularization neither improved nor worsened AI-SGD (see supplementary material for more details). Assumption 2 on learning rates and Assumption 5 are standard in the literature of stochastic approximations, dating back to the original paper of Robbins and Monro (1951) in the one-dimensional parameter case.

Assumptions on Lipschitz gradients, namely Assumption 3(b) and Assumption 3(c), have been relaxed in classical stochastic approximation theory (Benveniste et al., 1990, for example). However, these two Lipschitz conditions are commonly used in order to simplify the non-asymptotic analysis (Moulines and Bach, 2011). Assumption 3(a) is less standard in classic SGD literature but has so far been standard in the limited literature on implicit SGD (Bertsekas, 2011). It is also an open problem whether a clean stability result similar to Theorem 1 can be derived under Assumption 3(b) instead of Assumption 3(a). We discuss this issue after the proof of Theorem 1 in the supplementary material.

Assumption 4 makes two claims. The first claim on the observed Fisher information matrix is a relaxed form of strong convexity for the loss $L(\theta, \xi)$. However, in contrast to strong convexity, this claim allows several eigenvalues of $\nabla^2 L$ to be zero. The second claim of Assumption 4 is equivalent to strong convexity of the expected loss $\ell(\theta)$. From a statistical perspective, strong convexity posits that there is information in the data for all elements of $\theta_\star$. This assumption is necessary to derive bounds on the errors $\mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$, and has been used to show optimality of classic SGD with averaging (Polyak and Juditsky, 1992; Ljung et al., 1992; Xu, 2011; Moulines and Bach, 2011).

Overall, our assumptions are weaker than the assumptions in the limited literature on implicit SGD. For example, Bertsekas (2011, Assumptions 3.1, 3.2) assumes almost-sure bounded gradients $\nabla L(\theta, \xi)$ in addition to Assumption 3(a). We discuss more details in the supplementary material after the proof of Theorem 1.

3 Theory

In this section we present our theoretical analysis of AI-SGD. All proofs are given in the supplementary material. The main technical challenge in analyzing implicit SGD (3) is that unlike the typical analysis with classic SGD (2), the error ξ_n is not conditionally independent of θ_n. This implies that $\mathbb{E}\left(\nabla L(\theta_n, \xi_n) | \, \theta_n\right) \neq \ell(\theta_n)$, which makes it no longer possible to use the convexity properties of ℓ to analyze the errors $\mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$, as it is common in the literature.

As mentioned earlier, to circumvent this issue other authors have made strict assumptions of almost-sure bounded gradients or strong convexity (Bertsekas, 2011). In this paper, we rely on weaker conditions, namely the Lipschitz assumptions 3(a)-3(c), which are also used in non-implicit procedures. Our proof strategy relies on a master lemma (Lemma 3 in supplementary material) for the analysis of recursions that appear to be typical in implicit procedures. This result is novel to our best knowledge, and it can be useful in future research on implicit procedures.

3.1 Computational efficiency

Our first result enables efficient computation of the implicit update (3). In general, this can be expensive due to solving the fixed-point equation of the implicit update at every iteration. We reduce this multidimensional equation to an equation of only one dimension. Furthermore, under almost-sure convexity of the loss function, efficient search bounds for the one-dimensional fixed-point equation are available. This result generalizes an earlier result in efficient computation of implicit updates on generalized linear models (Toulis et al., 2014, Algorithm 1).

Definition 1. *Suppose that Assumption 1 holds. For observation $\xi = (x, y)$, the first derivative with respect to the natural parameter $x^\intercal \theta$ is denoted by $L'(\theta, \xi)$, and is defined as*

$$L'(\theta, \xi) \triangleq \frac{\partial L(\theta, \xi)}{\partial (x^\intercal \theta)} \stackrel{\text{def}}{=} \frac{\partial L(x^\intercal \theta, y)}{\partial (x^\intercal \theta)}. \tag{9}$$

Similarly, $L''(\xi, \theta) \triangleq \frac{\partial L'(\theta, \xi)}{\partial (x^\intercal \theta)}$.

Lemma 1. *Suppose that Assumption 1 holds, and consider functions L', L'' from Definition 1. Then, almost-surely,*

$$\nabla L(\theta_n, \xi_n) = s_n \nabla L(\theta_{n-1}, \xi_n); \tag{10}$$

the scalar s_n satisfies the fixed-point equation,

$$s_n\kappa_{n-1} = L'\left(\theta_{n-1} - s_n\gamma_n\kappa_{n-1}x_n, \xi_n\right), \qquad (11)$$

where $\kappa_{n-1} \triangleq L'(\theta_{n-1}, \xi_n)$. Moreover, if $L''(\theta, \xi) \geq 0$ almost-surely for all $\theta \in \Theta$, then

$$s_n \in \begin{cases} [\kappa_{n-1}, 0) & \text{if } \kappa_{n-1} < 0, \\ [0, \kappa_{n-1}] & \text{otherwise.} \end{cases}$$

Remarks. Lemma 1 has two parts. First, it shows that the implicit update can be performed by obtaining s_n from the fixed-point Eq. (10), and then using $\nabla L(\theta_n, \xi_n) = s_n \nabla L(\theta_{n-1}, \xi_n)$ in the implicit update (3). The fixed-point equation can be solved through a numerical root-finding procedure (Kivinen et al., 2006; Kulis and Bartlett, 2010; Toulis et al., 2014). Second, when the loss function is convex, then narrow search bounds for s_n are available. This property holds, for example, when the loss function is the negative log-likelihood in an exponential family.

3.2 Non-asymptotic analysis

Our next result is on the mean-squared errors $\mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$. These errors show the stability and convergence rates of implicit SGD and are used in combination with bounds on errors $\mathbb{E}\left(||\theta_n - \theta_\star||^4\right)$ to derive bounds on the errors $\mathbb{E}\left(||\bar{\theta}_n - \theta_\star||^2\right)$ of the averaged procedure.[1]

Theorem 1. *Suppose that Assumptions 1, 2, 3*(a)*, and 4 hold. Define $\delta_n \triangleq \mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$, and constants $\Gamma^2 = 4\lambda_0^2 \sum \gamma_i^2 < \infty$, $\epsilon = (1 + \gamma_1(\phi - \underline{\lambda_f}))^{-1}$, and $\lambda = 1 + \gamma_1\underline{\lambda_f}\epsilon$. Also let $\rho_\gamma(n) = n^{1-\gamma}$ if $\gamma \neq 1$ and $\rho_\gamma(n) = \log n$ if $\gamma = 1$. Then, there exists constant $n_0 > 0$ such that, for all $n > 0$,*

$$\delta_n \leq (8\lambda_0^2\gamma_1\lambda/\underline{\lambda_f}\epsilon)n^{-\gamma} + e^{-\log\lambda\cdot\rho_\gamma(n)}[\delta_0 + \lambda^{n_0}\Gamma^2].$$

Remarks. According to Theorem 1, the convergence rate of the implicit iterates θ_n is $\mathcal{O}(n^{-\gamma})$. This matches earlier results on rates of classic SGD (Benveniste et al., 1990; Moulines and Bach, 2011). The most important difference, however, is that the implicit procedure discounts the initial conditions δ_0 at an exponential rate, regardless of the specification of the learning rate. As shown by Moulines and Bach (2011, Theorem 1), in classic SGD there exists a term $\exp(\lambda_1^2\gamma_1^2 n^{1-2\gamma})$ in front of the initial conditions, which can be catastrophic if the learning rate parameter γ_1 is misspecified. In contrast, the implicit iterates are unconditionally stable, i.e., any specification of the learning rate will lead to a stable discounting of the initial conditions.

[1] The bounds for the fourth moments $\mathbb{E}\left(||\theta_n - \theta_\star||^4\right)$ are given in the supplementary material because they rely on the same intermediate results as for $\mathbb{E}\left(||\theta_n - \theta_\star||^2\right)$.

Theorem 2. *Consider the* AI-SGD *procedure* (4)*, and suppose that Assumptions 2, 3*(a)*, 3*(c)*, 4, and 5 hold. Then,*

$$\begin{aligned}(\mathbb{E}\left(||\bar{\theta}_n - \theta_\star||^2\right))^{1/2} \leq &\left(\mathrm{tr}(\nabla^2\ell(\theta_\star)^{-1}\Sigma\nabla^2\ell(\theta_\star)^{-1})/n\right)^{1/2} \\ &+ \mathcal{O}(n^{-1+\gamma/2}) + \mathcal{O}(n^{-\gamma}) \\ &+ \mathcal{O}(\exp(-\log\lambda\cdot n^{1-\gamma}/2).\end{aligned}$$

Remarks. The full version of Theorem 2, which includes all constants, is given in the supplementary material. Even in its shortened form, Theorem 2 delivers three main results. First, the iterates $\bar{\theta}_n$ attain the Cramér-Rao lower bound, i.e., any other unbiased estimator of $\theta_\star$ cannot have lower MSE than $\bar{\theta}_n$. From an optimization perspective, $\bar{\theta}_n$ attains the rate $\mathcal{O}(1/n)$, which is optimal for first-order methods (Nesterov, 2004). This result matches the asymptotic optimality of averaged iterates from classic SGD procedures, which has been proven by Polyak and Juditsky (1992).

Second, the remaining rates are $\mathcal{O}(n^{-2+\gamma})$ and $\mathcal{O}(n^{-2\gamma})$. This implies the optimal choice $\gamma = 2/3$ for the exponent of the learning rate. It extends the results of Ruppert (1988), and more recently by Xu (2011), and Moulines and Bach (2011), on optimal exponents for classic SGD procedures.

Third, as with non-averaged implicit iterates in Theorem 1, the averaged iterates $\bar{\theta}_n$ have a decay of the initial conditions regardless of the specification of the learning rate parameter. This stability property is inherited from the underlying implicit SGD procedure (3) that is being averaged. In contrast, averaged iterates of classic SGD procedures can diverge numerically because arbitrarily large terms can appear in front of initial conditions (Moulines and Bach, 2011, Theorem 3). We demonstrate this stability in the experiment of Section 4.1.

4 Experiments

In this section, we show that AI-SGD achieves comparable, and sometimes superior, results to other methods while combining statistical efficiency, stability, and simplicity. In our experiments, we compare our procedure to the following procedures:

- SGD: Classic stochastic gradient descent in its standard formulation (Sakrison, 1965; Zhang, 2004), which employs the update $\theta_n = \theta_{n-1} - \gamma_n\nabla L(\theta_{n-1}, \xi_n)$.
- IMPLICIT SGD: Stochastic gradient descent procedure introduced in Toulis et al. (2014) which employs implicit update (3) without averaging. It is robust to misspecification of the learning rate but also exhibits slower convergence in practice relative to classic SGD.
- ASGD: Averaged stochastic gradient descent procedure with classic updates of the iterates (Xu, 2011; Shamir

and Zhang, 2012; Bach and Moulines, 2013). This is equivalent to AI-SGD where the update (3) is replaced by the classic step $\theta_n = \theta_{n-1} - \gamma_n \nabla L(\theta_{n-1}, \xi_n)$.

- PROX-SVRG: A proximal version of the stochastic gradient descent procedure with progressive variance reduction (SVRG) (Xiao and Zhang, 2014).
- PROX-SAG: A proximal version of the stochastic average gradient (SAG) procedure (Schmidt et al., 2013). While its theory has not been formally established, PROX-SAG has shown similar convergence properties to PROX-SVRG in practice.
- ADAGRAD: A stochastic gradient descent procedure with a form of diagonal scaling to adapt the learning rate (Duchi et al., 2011).

Note that PROX-SVRG and PROX-SAG are applicable only to fixed data sets and not to the streaming setting. Therefore the theoretical linear convergence rate of these methods refers to convergence to an empirical minimizer (e.g., maximum likelihood, or maximum a-posteriori if there is regularization), and not to the ground truth $\theta_\star$. On the other hand, AI-SGD can be applied to both data settings.

We also note that ADAGRAD, and similar adaptive schedules, (Tieleman and Hinton, 2012; Kingma and Ba, 2015) effectively approximate the natural gradient $\mathcal{I}(\theta)^{-1}\nabla L(\theta, \xi)$ by using a multi-dimensional learning rate. These learning rates have the added advantage of being less sensitive than one-dimensional rates to tuning of hyperparameters, and can be combined in practice with AI-SGD.

4.1 Statistical efficiency and stability

We first demonstrate the theoretical results on the stability and statistical optimality of AI-SGD. To do so, we follow a simple normal linear regression example from Bach and Moulines (2013). Let $N = 10^6$ be the number of observations and $p = 20$ be the number of features. Let $\theta_\star = (0, 0, \ldots, 0)^\intercal$ be the ground truth. The random variable ξ is decomposed as $\xi_n = (x_n, y_n)$, where the feature vectors $x_1, \ldots, x_N \sim \mathcal{N}_p(0, H)$ are i.i.d. normal random variables, and H is a randomly generated symmetric matrix with eigenvalues $1/k$, for $k = 1, \ldots, p$. The outcome y_n is sampled from a normal distribution as $y_n \mid x_n \sim \mathcal{N}(x_n^\intercal \theta_*, 1)$, for $n = 1, \ldots, N$. Our loss function is defined as the squared residual, i.e., $L(\theta, \xi_n) = (y_n - x_n^\intercal \theta)^2$, and thus $\ell(\theta) = \mathbb{E}\left(L(\theta, \xi)\right) = (\theta - \theta_\star)^\intercal H (\theta - \theta_\star)$.

We choose a constant learning rate $\gamma_n \equiv \gamma_1$ according to the average radius of the data $R^2 = \mathrm{trace}(H)$, and for both ASGD and AI-SGD we collect iterates θ_n, $n = 1, \ldots, N$, and keep the average $\bar{\theta}_n$. In Figure 1, we plot $\ell(\bar{\theta}_n)$ for each iteration for a maximum of N iterations in log-log space.

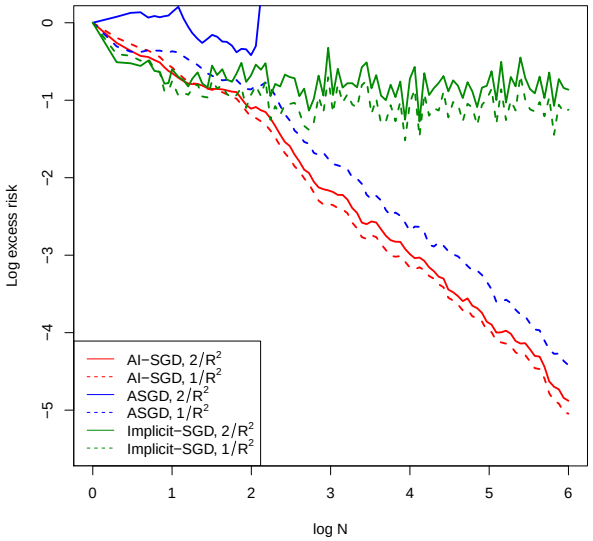

Figure 1: Loss of AI-SGD, ASGD, and IMPLICIT SGD, on simulated multivariate normal data with $N = 10^6$ observations, $d = 20$ features. The plot shows that AI-SGD achieves stability regardless of the specification of the learning rate $\gamma_n \equiv \gamma_1$. In contrast, ASGD diverges when the learning rate is only slightly misspecified (e.g., solid blue line).

Figure 1 shows that AI-SGD performs on par with ASGD for the rates at which ASGD is known to be optimal. However, the benefit of the implicit procedure (3) in AI-SGD becomes clear as the learning rate increases. Notably, AI-SGD remains stable for learning rates that are above the theoretical threshold, i.e., when $\gamma_1 > 1/R^2$, whereas ASGD diverges above that threshold, e.g., when $\gamma_1 = 2/R^2$. This stable behavior is also exhibited in IMPLICIT SGD, but IMPLICIT SGD converges at a slower rate than AI-SGD, and thus does not combine stability with statistical efficiency. This behavior is also reflected for AI-SGD when using decaying learning rates, e.g., $\gamma_n \propto 1/n$.

4.2 Classification error

We now conduct a study of AI-SGD's empirical performance on standard benchmarks of large-scale linear classification. For brevity, we display results on four data sets although we have seen similar results on eight additional ones (see the supplementary material for more details).

Table 1 displays a summary of the data sets. The COVTYPE data set (Blackard, 1998) consists of forest cover types in which the task is to classify class 2 among 7 forest cover types. DELTA is synthetic data offered in the PASCAL Large Scale Challenge (Sonnenburg et al., 2008) and we apply the default processing offered by the challenge organizers. The task in RCV1 is to classify documents belonging

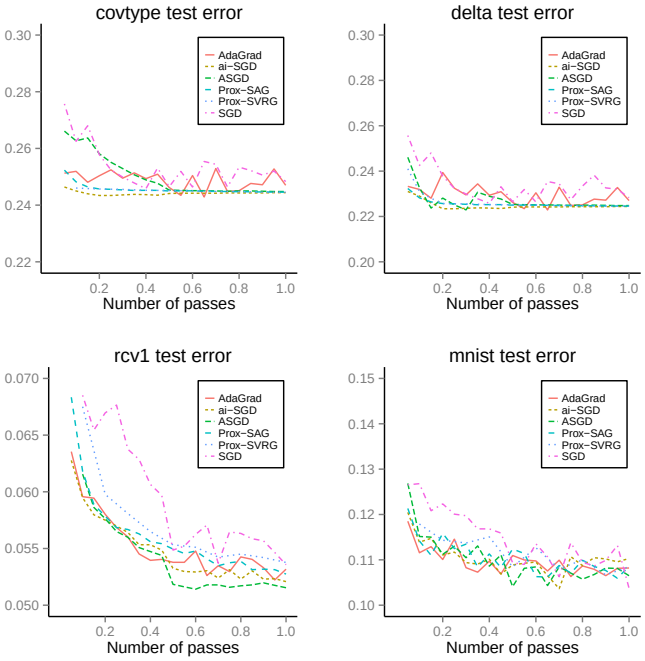

Figure 2: Large scale linear classification with log loss on four data sets. Each plot indicates the test error of various stochastic gradient methods over a single pass of the data.

to class CCAT in the text dataset (Lewis et al., 2004), where we apply the standard preprocessing provided by Bottou (2012). In the MNIST data set (Le Cun et al., 1998) of images of handwritten digits, the task is to classify digit 9 against all others.

For AI-SGD and ASGD, we use the learning rate $\gamma_n = \eta_0(1+\eta_0 n)^{-3/4}$ prescribed in Xu (2011), where the constant η_0 is determined through preprocessing on a small subset of the data. Hyperparameters for other methods are set based on a computationally intensive grid search over the entire hyperparameter space: this includes step sizes for PROX-SAG, PROX-SVRG, and ADAGRAD, and the inner iteration count for PROX-SVRG. For all methods we use L_2 regularization with parameter λ which varies for each data set, and which is also used in Xu (2011).

The results are shown in Figure 2. We see that AI-SGD achieves comparable performance with the tuned proximal methods PROX-SVRG and PROX-SAG, as well as ADAGRAD. Interestingly, ADAGRAD exhibits a larger variance in its estimate than the proximal methods. This comes from the less known fact that the learning rate in ADAGRAD is a suboptimal approximation of the Fisher information, and hence it is statistically inefficient.

4.3 Sensitivity analysis

We examine the inherent stability of the aforementioned procedures by perturbing their hyperparameters. That is, we perform sensitivity analysis by varying any hyperparameters that the user must tweak in order to fine tune the convergence of each procedure. We do so for hyperparameters in ASGD (the learning rate), PROX-SVRG (proximal step size η and inner iteration m), and AI-SGD (the learning rate).

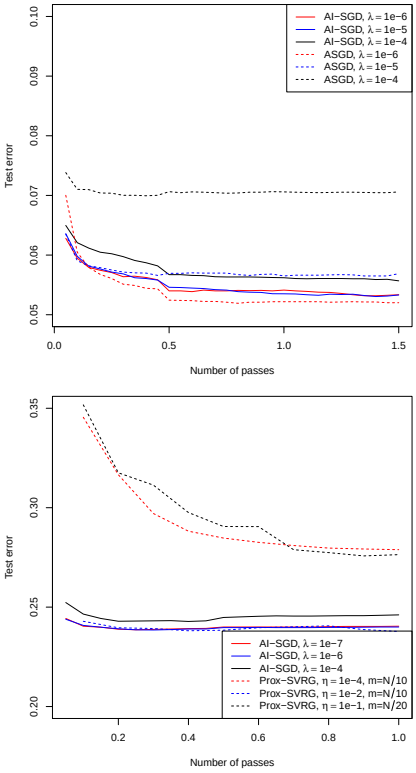

Figure 3: Top: Logistic regression on the RCV1 dataset, performing sensitivity analysis of AI-SGD and ASGD for the choice of regularization parameter λ. Bottom: linear SVM on the covtype dataset, performing sensitivity analysis of AI-SGD and PROX-SVRG, in which PROX-SVRG has additional hyperparameters η according to the step size of the proximal update and m according to the inner iteration count.

The results are shown in Figure 3. As we decrease the regularization parameter, ASGD performs increasingly worse. While it may converge, the test error can be arbitrarily large. On the other hand, AI-SGD always converges and is not affected by regularization. When the regularization parameter is about $1/N$, i.e., when $\lambda < 1\text{E-}6$, ASGD remains stable and can compare with AI-SGD. Similar results hold when perturbing η and m in PROX-SVRG, as AI-SGD does not require specification of such hyperparameters.

	description	type	features	training set	test set	λ
covtype	forest cover type	sparse	54	464,809	116,203	10^{-6}
delta	synthetic data	dense	500	450,000	50,000	10^{-2}
rcv1	text data	sparse	47,152	781,265	23,149	10^{-5}
mnist	digit image features	dense	784	60,000	10,000	10^{-3}

Table 1: Summary of data sets and the L_2 regularization parameter, following the settings in Xu (2011).

5 Conclusion

We propose a statistical learning procedure, termed AI-SGD, and investigate its theoretical and empirical properties. AI-SGD combines simple stochastic proximal steps, also known as implicit updates, with iterate averaging and larger step-sizes. The proximal steps allow AI-SGD to be significantly more stable compared to classic SGD procedures, with or without averaging of the iterates; this stability comes at virtually no computational cost for a large family of machine learning models. Furthermore, the averaging of the iterates lead AI-SGD to be statistically optimal, i.e., the variance of the iterate $\bar{\theta}_n$ of AI-SGD achieves the minimum Cramér-Rao lower bound, under strong convexity. Last but not least, AI-SGD is as simple to implement as classic SGD. In comparison, other stochastic proximal procedures, such as PROX-SVRG or PROX-SAG, require tuning of hyperparameters that control periodic calculations over the entire dataset, and possibly storage of the full gradient.

References

Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Advances in Neural Information Processing Systems*, pages 773–781, 2013.

JA Bather. *Stochastic approximation: A generalisation of the Robbins-Monro procedure*, volume 89. Mathematical Sciences Institute, Cornell University, 1989.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Albert Benveniste, Pierre Priouret, and Michel Métivier. Adaptive algorithms and stochastic approximations. 1990.

Dimitri P Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical programming*, 129(2):163–195, 2011.

Jock Blackard. *Comparison of neural networks and discriminant analysis in predicting forest cover types*. PhD thesis, Department of Forest Sciences, Colorado State University, 1998.

Léon Bottou. Stochastic learning. In *Advanced lectures on machine learning*, pages 146–168. Springer, 2004.

Leon Bottou. Stochastic Gradient Descent Tricks. In *Neural Networks: Tricks of the Trade*, volume 1, pages 421–436. 2012.

Alexandre Défossez and Francis Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 205–213, 2015.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Nikos Karampatziakis and John Langford. Online importance weight aware updates. *arXiv preprint arXiv:1011.1576*, 2010.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

Jyrki Kivinen, Manfred K Warmuth, and Babak Hassibi. The p-norm generalization of the lms algorithm for adaptive filtering. *Signal Processing, IEEE Transactions on*, 54(5):1782–1793, 2006.

Krzysztof A Krakowski, Robert E Mahony, Robert C Williamson, and Manfred K Warmuth. A geometric view of non-linear on-line stochastic gradient descent. *Author website*, 2007.

Brian Kulis and Peter L Bartlett. Implicit online learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 575–582, 2010.

Yann Le Cun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998.

David Lewis, Yiming Yang, Tony Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *The Journal of Machine Learning Research*, 5: 361–397, 2004.

Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

Lennart Ljung, Georg Ch Pflug, and Harro Walk. *Stochastic approximation and optimization of random systems*, volume 17. Springer, 1992.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

Jin-Ichi Nagumo and Atsuhiko Noda. A learning method for system identification. *Automatic Control, IEEE Transactions on*, 12(3):282–287, 1967.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):123–231, 2013.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

Lorenzo Rosasco, Silvia Villa, and Bang Công Vũ. Convergence of stochastic proximal gradient algorithm. *arXiv preprint arXiv:1403.5074*, 2014.

David Ruppert. Efficient estimators from a slowly convergent robbins-monro process. Technical report, School of Operations Research and Industrial Engineering, Cornell University, 1988.

David J Sakrison. Efficient recursive estimation; application to estimating the parameters of a covariance function. *International Journal of Engineering Science*, 3(4): 461–483, 1965.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. Technical report, HAL 00860051, 2013.

Li Cheng SVN Schuurmans and SW Caelli. Implicit online learning with kernels. *Advances in neural information processing systems*, 19:249, 2007.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. *arXiv preprint arXiv:1212.1824*, 2012.

Yoram Singer and John C Duchi. Efficient learning using forward-backward splitting. In *Advances in Neural Information Processing Systems*, pages 495–503, 2009.

Soeren Sonnenburg, Vojtech Franc, Elad Yom-Tov, and Michele Sebag. Pascal large scale learning challenge, 2008. URL `http://largescale.first.fraunhofer.de`.

Thaddeus Tarpey and Bernard Flury. Self-consistency: a fundamental concept in statistics. *Statistical Science*, pages 229–243, 1996.

T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Panos Toulis and Edoardo M Airoldi. Implicit stochastic gradient descent. *arXiv preprint arXiv:1408.2923*, 2014.

Panos Toulis and Edoardo M. Airoldi. Scalable estimation strategies based on stochastic approximations: classical results and new insights. *Statistics and Computing*, 25 (4):781–795, 2015. doi: 10.1007/s11222-015-9560-y. URL `http://dx.doi.org/10.1007/s11222-015-9560-y`.

Panos Toulis, Jason Rennie, and Edoardo Airoldi. Statistical analysis of stochastic gradient methods for generalized linear models. In *31st International Conference on Machine Learning*, 2014.

Lin Xiao and Tony Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24:2057–2075, 2014.

Wei Xu. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv preprint arXiv:1107.2490*, 2011.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM, 2004.