# Lightweight Stochastic Optimization for Minimizing Finite Sums with Infinite Data

**Shuai Zheng** [1] **James T. Kwok** [1]

## Abstract

Variance reduction has been commonly used in stochastic optimization. It relies crucially on the assumption that the data set is finite. However, when the data are imputed with random noise as in data augmentation, the perturbed data set becomes essentially infinite. Recently, the stochastic MISO (S-MISO) algorithm is introduced to address this expected risk minimization problem. Though it converges faster than SGD, a significant amount of memory is required. In this paper, we propose two SGD-like algorithms for expected risk minimization with random perturbation, namely, stochastic sample average gradient (SSAG) and stochastic SAGA (S-SAGA). The memory cost of SSAG does not depend on the sample size, while that of S-SAGA is the same as those of variance reduction methods on unperturbed data. Theoretical analysis and experimental results on logistic regression and AUC maximization show that SSAG has faster convergence rate than SGD with comparable space requirement, while S-SAGA outperforms S-MISO in terms of both iteration complexity and storage.

## 1. Introduction

Machine learning tasks are often cast as optimization problems with some data distributions. In regularized risk minimization with $n$ training samples, one minimizes:

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta) + g(x), \tag{1}$$

where $\theta$ is the model parameter, $\ell_i$ is the loss due to sample $i$, and $g$ is a regularizer. In this paper, we assume that $\ell_i$ and $g$ are smooth and convex. Stochastic gradient descent (SGD) (Robbins & Monro, 1951) and its variants (Nemirovski et al., 2009; Xiao, 2010; Duchi et al., 2011; Bottou et al., 2016) are flexible, scalable, and widely used for this problem. However, SGD suffers from large variance due to sampling noise. To alleviate this problem, the stepsize has to be decreasing, which slows convergence.

By exploiting the finite-sum structure in (1), a class of variance-reduced stochastic optimization methods have been proposed recently (Le Roux et al., 2012; Johnson & Zhang, 2013; Shalev-Shwartz & Zhang, 2013; Mairal, 2013; Defazio et al., 2014a;b). Based on the use of control variates (Fishman, 1996), they construct different approximations to the true gradient so that its variance decreases as the optimal solution is approached.

In order to capture more variations in the data distribution, it is effective to obtain more training data by injecting random noise to the data samples (Decoste & Schölkopf, 2002; van der Maaten et al., 2013; Paulin et al., 2014). Theoretically, it has been shown that random noise improves generalization (Wager et al., 2014). In addition, artificially corrupting the training data has a wide range of applications in machine learning. For example, additive Gaussian noise can be used in image denoising (Vincent et al., 2010) and provides a form of $\ell_2$-type regularization (Bishop, 1995); dropout noise serves as adaptive regularization that is useful in stabilizing predictions (van der Maaten et al., 2013) and selecting discriminative but rare features (Wager et al., 2013); and Poisson noise is of interest to count features as in document classification (van der Maaten et al., 2013).

With the addition of noise perturbations, (1) becomes the following expected risk minimization problem:

$$\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} \mathbf{E}_{\xi_i}[\ell_i(\theta;\xi_i)] + g(x), \tag{2}$$

where $\xi_i$ is the random noise injected to function $\ell_i$, and $\mathbf{E}_{\xi}$ denotes expectation w.r.t. $\xi_i$. Because of the expectation, the perturbed data can be considered as infinite, and the finite data set assumption in variance reduction methods is violated. In this case, each function in problem (2) can only be accessed via a stochastic first-order oracle, and the main optimization tool is SGD.

[1]Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. Correspondence to: Shuai Zheng <szhengac@cse.ust.hk>.

*Proceedings of the $35^{th}$ International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. 

Despite its importance, expected risk minimization has received very little attention. One very recent work for this is the stochastic MISO (S-MISO) (Bietti & Mairal, 2017). While it converges faster than SGD, S-MISO requires $O(nd)$ space, where $d$ is the feature dimensionality. This significantly limits its applicability to big data problems. The N-SAGA algorithm (Hofmann et al., 2015) can also be used on problems with infinite data. However, its asymptotic error is nonzero.

In this paper, we focus on the linear model. By exploiting the linear structure, we propose two SGD-like variants with low memory costs: stochastic sample average gradient (SSAG) and stochastic SAGA (S-SAGA). In particular, the memory cost of SSAG does not depend on the sample size $n$, while S-SAGA has a memory requirement of $O(n)$, which matches the stochastic variance reduction methods on unperturbed data (Le Roux et al., 2012; Shalev-Shwartz & Zhang, 2013; Defazio et al., 2014a;b). Similar to S-MISO, the proposed algorithms have faster convergence than SGD. Moreover, the convergence rate of S-SAGA depends on a constant that is typically smaller than that of S-MISO. Experimental results on logistic regression and AUC maximization with dropout noise demonstrate the efficiency of the proposed algorithms.

**Notations**. For a vector $x$, $\|x\| = \sqrt{\sum_i x_i^2}$ is its $\ell_2$-norm. For two vectors $x$ and $y$, $x^T y$ denotes its dot product.

## 2. Related Work

In this paper, we consider the linear model. Given samples $\{x_1, \ldots, x_n\}$, with each $x_i \in \mathbb{R}^d$, the regularized risk minimization problem in (1) can be written as:

$$\min_\theta \frac{1}{n} \sum_{i=1}^n \phi_i(x_i^T \theta) + g(\theta), \tag{3}$$

where $\hat{y}_i \equiv x_i^T \theta$ is the prediction on sample $i$, and $\phi_i$ is a loss. For example, logistic regression corresponds to $\phi_i(\hat{y}_i) = \log(1 + \exp(-y_i \hat{y}_i))$, where $\{y_1, \ldots, y_n\}$ are the training labels; and linear regression corresponds to $\phi_i(\hat{y}_i) = (y_i - \hat{y}_i)^2$.

### 2.1. Learning with Injected Noise

To make the predictor robust, one can inject i.i.d. random noise $\xi_i$ to each sample $x_i$ (van der Maaten et al., 2013). Let the perturbed sample be $\hat{x}_i \equiv \psi(x_i, \xi_i)$. The following types of noise have been popularly used: (i) additive noise (Bishop, 1995; Wager et al., 2013): $\hat{x} = x + \xi$, where $\xi$ comes from a zero-mean distribution such as the normal or Poisson distribution; and (ii) dropout noise (Srivastava et al., 2014): $\hat{x} = \xi \circ x$, where $\circ$ denotes the element-wise product, $\xi \in \{0, 1/(1-p)\}^d$, $p$ is the dropout probability, and each component of $\xi$ is an independent draw from a scaled Bernoulli$(1-p)$ random variable. With random perturbations, (3) becomes the following expected risk minimization problem:

$$\min_\theta F(\theta) \equiv \frac{1}{n} \sum_{i=1}^n \mathbf{E}_{\xi_i}[\phi_i(\hat{x}_i^T \theta)] + g(\theta). \tag{4}$$

As the objective contains an expectation, computing the gradient is infeasible as infinite samples are needed. As an approximation, SGD uses the gradient from a single sample. However, this has large variance.

In this paper, we make the following assumption on $f_i(\theta; \xi_i) \equiv \phi_i(\hat{x}_i^T \theta) + g(\theta)$ in (4). Note that this implies $\phi_i(\hat{x}_i^T \theta)$ and $F$ are also $L$-smooth.

**Assumption 1.** *Each $f_i(\theta; \xi_i)$ is $L$-smooth w.r.t. $\theta$, i.e., there exists constant $L$ such that $\|\nabla f_i(\theta; \xi_i) - \nabla f_i(\theta'; \xi_i)\| \le L\|\theta - \theta'\|, \forall \theta, \theta'$.*

### 2.2. Variance Reduction

In stochastic optimization, control variates have been commonly used to reduce the variance of stochastic gradients (Fishman, 1996). In general, given a random variable $X$ and another highly correlated random variable $Y$, a variance-reduced estimate of $\mathbf{E}X$ can be obtained as

$$X - Y + \mathbf{E}Y. \tag{5}$$

In stochastic optimization on problem (3), the gradient $\phi_i'(x_i^T \theta) x_i$ of the loss evaluated on sample $x_i$ is taken as $X$. When the training set is finite, various algorithms have been recently proposed so that $Y$ is strongly correlated with $\phi_i'(x_i^T \theta) x_i$ and $\mathbf{E}Y$ can be easily evaluated. Examples include stochastic average gradient (SAG) (Le Roux et al., 2012), MISO (Mairal, 2013), stochastic variance reduced gradient (SVRG) (Johnson & Zhang, 2013), Finito (Defazio et al., 2014b), SAGA (Defazio et al., 2014a), and stochastic dual coordinate ascent (SDCA) (Shalev-Shwartz & Zhang, 2013).

However, with the expectation in (4), the full gradient (i.e., $\mathbf{E}Y$ in (5)) cannot be evaluated, and variance reduction can no longer be used. Very recently, the stochastic MISO (S-MISO) algorithm (Bietti & Mairal, 2017) is proposed for solving (4). Its convergence rate outperforms that of SGD by having a smaller multiplicative constant. However, S-MISO requires an additional $O(nd)$ space, which prevents its use on large data sets.

## 3. Sample Average Gradient

Let the iterate at iteration $t$ be $\theta_{t-1}$. To approximate the gradient $\nabla F(\theta_{t-1})$ in (4), SGD uses the gradient $g_t = \phi_{i_t}'(\hat{x}_{i_t}^T \theta_{t-1}) \hat{x}_{i_t} + \nabla g(\theta_{t-1})$ evaluated on a single sample $\hat{x}_{i_t}$, where $i_t$ is sampled uniformly from $[n] \equiv$

$\{1, 2, \ldots, n\}$. The variance of $g_t$ is usually assumed to be bounded by a constant, as

$$\mathbf{E}\|g_t - \nabla F(\theta_{t-1})\|^2 \leq \sigma_s^2, \tag{6}$$

where the expectation is taken w.r.t. both the random index $i_t$ and perturbation $\xi_t$ at iteration $t$. Note that the gradient of regularizer $g$ does not contribute to the variance.

### 3.1. Exploiting the Model Structure

#### 3.1.1. Stochastic Sample-Average Gradient (SSAG)

At iteration $t$, the stochastic gradient of the loss $\phi_i(\hat{x}_i^T\theta)$ for sample $\hat{x}_{i_t}$ is $\phi'(\hat{x}_{i_t}^T\theta)\hat{x}_{i_t}$. Thus, the gradient direction is determined by $\hat{x}_{i_t}$, while parameter $\theta$ only affects its scale. With this observation, we consider using $a_t\hat{x}_{i_t}$ as a control variate for $\phi'(\hat{x}_{i_t}^T\theta)\hat{x}_{i_t}$, where $a_t$ may depend on past information but not on $\hat{x}_{i_t}$. Note that the gradient component $\nabla g(\theta)$ is deterministic, and does not contribute to the construction of control variate. Using (5), the resultant gradient estimator is:

$$z_t = (\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1}) - a_t)\hat{x}_{i_t} + a_t\tilde{x}_t + \nabla g(\theta_{t-1}), \tag{7}$$

where $\tilde{x}_t$ is an estimate of $\mathbf{E}[\hat{x}_{i_t}]$. For example, $\tilde{x}_t$ can be defined as

$$\tilde{x}_t = \left(1 - \frac{1}{t}\right)\tilde{x}_{t-1} + \frac{1}{t}\hat{x}_{i_t}, \tag{8}$$

so that $\tilde{x}_t$ can be incrementally updated as $\hat{x}_{i_t}$'s are sampled. As $\hat{x}_{i_t}$'s are i.i.d., by the law of large number, the sample average $\tilde{x}_t$ converges to the expected value $\mathbf{E}[\hat{x}_{i_t}]$.

The following shows that $z_t$ in (7) is a biased estimator of the gradient $\nabla F(\theta_{t-1})$. As $\tilde{x}_t$ converges to $\mathbf{E}[\hat{x}_{i_t}]$, $z_t$ is still asymptotically unbiased.

**Proposition 1.** $\mathbf{E}[z_t] = \nabla F(\theta_{t-1}) + a_t\left(1 - \frac{1}{t}\right)(\tilde{x}_{t-1} - \mathbf{E}[\hat{x}_{i_t}])$.

Note that $\mathbf{E}[\hat{x}_{i_t}] = \frac{1}{n}\sum_{i=1}^n \mathbf{E}_{\xi_i}[\hat{x}_i]$, where $\mathbf{E}_{\xi_i}$ denotes the expectation w.r.t. $\xi_i$. We assume that each $\mathbf{E}_{\xi_i}[\hat{x}_i]$ can be easily computed. This is the case, for example, when the noise is dropout noise or additive zero-mean noise, and $\mathbf{E}_{\xi_i}[\hat{x}_i] = x_i$ (van der Maaten et al., 2013). This suggests replacing $\tilde{x}_t$ in (7) by $\tilde{x} \equiv \frac{1}{n}\sum_{i=1}^n \mathbf{E}_{\xi_i}[\hat{x}_i]$ (which is equal to $\mathbf{E}[\hat{x}_{i_t}]$), leading to the estimator:

$$v_t = (\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1}) - a_t)\hat{x}_{i_t} + a_t\tilde{x} + \nabla g(\theta_{t-1}). \tag{9}$$

The following shows that $v_t$ is unbiased, and also provides an upper bound of its variance.

**Proposition 2.** $\mathbf{E}[v_t] = \nabla F(\theta_{t-1})$, *and* $\mathbf{E}[\|v_t - \nabla F(\theta_{t-1})\|^2] \leq \mathbf{E}[(\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1}) - a_t)^2\|\hat{x}_{i_t}\|^2]$. *The bound is minimized when*

$$a_t = a_t^* \equiv \frac{\mathbf{E}[\phi'(\hat{x}^T\theta_{t-1})\|\hat{x}\|^2]}{\mathbf{E}[\|\hat{x}\|^2]}. \tag{10}$$

For dropout noise and other additive noise with known variance, one can compute $\mathbf{E}_{\xi_i}\|\hat{x}_i\|^2$ for each $i \in [n]$, and then average to obtain $\mathbf{E}[\|\hat{x}\|^2]$. However, evaluating the expectation in the numerator of (10) is infeasible.

Instead, we define $a_t$ as

$$a_t = \tilde{a}_t / s_t \tag{11}$$

for $t \geq 1$, and approximate the expectations in the numerator and denominator by moving averages:

$$\begin{aligned} \tilde{a}_{t+1} &= (1-\beta_t)\tilde{a}_t + \beta_t\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\|\hat{x}_{i_t}\|^2, \\ s_{t+1} &= (1-\beta_t)s_t + \beta_t\|\hat{x}_{i_t}\|^2. \end{aligned}$$

We initialize $a_1 = \tilde{a}_1 = s_1 = 0$, and set $\beta_t \in [0, 1)$.

The resulting algorithm, called stochastic sample-average gradient (SSAG), is shown in Algorithm 1. Compared to S-MISO (Bietti & Mairal, 2017), SSAG is more computationally efficient. It does not require an extra $O(nd)$ memory, and only requires one single gradient evaluation (step 6) in each iteration.

**Algorithm 1** Stochastic sample-average gradient (SSAG).

1: **Input:** $\eta_t > 0$, $\beta_t \in [0, 1)$.
2: **initialize** $\theta_0$; $\tilde{x} \leftarrow \frac{1}{n}\sum_{i=1}^n \mathbf{E}_{\xi_i}[\hat{x}_i]$; $a_1 \leftarrow 0$; $\tilde{a}_1 \leftarrow 0$; $s_1 \leftarrow 0$
3: **for** $t = 1, 2, \ldots$ **do**
4: draw sample index $i_t$ and random perturbation $\xi_t$
5: $\hat{x}_{i_t} \leftarrow \psi(x_{i_t}, \xi_t)$
6: $d_t \leftarrow \phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})$
7: $v_t \leftarrow (d_t - a_t)\hat{x}_{i_t} + a_t\tilde{x} + \nabla g(\theta_{t-1})$
8: $\theta_t \leftarrow \theta_{t-1} - \eta_t v_t$
9: $\tilde{a}_{t+1} \leftarrow (1-\beta_t)\tilde{a}_t + \beta_t d_t\|\hat{x}_{i_t}\|^2$
$s_{t+1} \leftarrow (1-\beta_t)s_t + \beta_t\|\hat{x}_{i_t}\|^2$
$a_{t+1} \leftarrow \tilde{a}_{t+1}/s_{t+1}$
10: **end for**

The following Proposition shows that $a_t$ in (11) is asymptotically optimal for appropriate choices of $\eta_t$ and $\beta_t$.

**Proposition 3.** *If (i)* $\mathbf{E}[\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})^2\|\hat{x}_{i_t}\|^4] < \infty$ *and* $\mathbf{E}[\|\hat{x}_{i_t}\|^4] < \infty$*; (ii)* $\|v_t\| < \infty$*; (iii)* $\eta_t \to 0$, $\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$*; (iv)* $\beta_t \to 0$, $\sum_t \beta_t = \infty$, $\sum_t \beta_t^2 < \infty$*; and (v)* $\eta_t/\beta_t \to 0$*, then*

$$a_t \to a_t^* \quad w.p.1.$$

A simple choice is: $\eta_t = O(1/t^{c_1})$, $\beta_t = O(1/t^{c_2})$, where $1/2 < c_2 < c_1 \leq 1$. The following Proposition quantifies the convergence of $s_t a_t$ to $s_t a_*^t$. In particular, when $c_1 = 1$, the asymptotic bound in (12) is minimized when $c_2 = 2/3$.

**Proposition 4.** *With assumptions (i)-(v) in Proposition 3,* $\eta_t = O(1/t^{c_1})$*, and* $\beta_t = O(1/t^{c_2})$*, we have*

$$\mathbf{E}[s_t^2(a_t - a_t^*)^2] \leq O\left(\max\left\{\frac{1}{t^{c_2}}, \frac{1}{t^{2(c_1-c_2)}}\right\}\right). \tag{12}$$

#### 3.1.2. STOCHASTIC SAGA (S-SAGA)

Recall that in (9), $\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\hat{x}_{i_t}$ plays the role of $X$ in (5), and $a_t\hat{x}_{i_t}$ plays the role of $Y$. However, the corresponding $X$ and $Y$ in (5) can be negatively correlated in some iterations. This is partly because $a_t$ in (9) does not depend on $\hat{x}_{i_t}$, though $a_t\hat{x}_{i_t}$ serves as a control variate for $\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\hat{x}_{i_t}$. Thus, it is better for each sample $\hat{x}_i$ to have its own scaling factor, leading to the estimator:

$$v_t = (\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1}) - a_{i_t})\hat{x}_{i_t} + m_{t-1} + \nabla g(\theta_{t-1}), \quad (13)$$

where $m_{t-1} = \mathbf{E}[a_{i_t}\hat{x}_{i_t}] = \frac{1}{n}\sum_{i=1}^n a_i\mathbf{E}_{\xi_i}[\hat{x}_i]$. Note that $m_t$ can be updated sequentially as:

$$m_t = m_{t-1} + \frac{1}{n}(\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1}) - a_{i_t})\mathbf{E}_{\xi_t}[\hat{x}_{i_t}].$$

Besides, (13) reduces to the SAGA estimator (Defazio et al., 2014a) when the random noise is removed. The whole procedure, which will be called stochastic SAGA (S-SAGA), is shown in Algorithm 2.

**Algorithm 2** Stochastic SAGA (S-SAGA).

**Input:** $\eta_t > 0$.
**initialize** $\theta_0$; $\bar{x}_i \leftarrow \mathbf{E}_{\xi_i}[\hat{x}_i]$ and $a_i \leftarrow \phi'_i(\hat{x}_i^T, \theta_0)$ for all $i \in [n]$; $m_0 = \frac{1}{n}\sum_{i=1}^n a_i\bar{x}_i$
**for** $t = 1, 2, \ldots$ **do**
  draw sample index $i_t$ and random perturbation $\xi_t$
  $\hat{x}_{i_t} \leftarrow \psi(x_{i_t}, \xi_t)$
  $d_t \leftarrow \phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})$
  $v_t \leftarrow (d_t - a_{i_t})\hat{x}_{i_t} + m_{t-1} + \nabla g(\theta_{t-1})$
  $\theta_t \leftarrow \theta_{t-1} - \eta_t v_t$
  $m_t \leftarrow m_{t-1} + \frac{1}{n}(d_t - a_{i_t})\bar{x}_{i_t}$
  $a_{i_t} \leftarrow d_t$
**end for**

S-SAGA needs an additional $O(nd)$ space for $\{a_1, \ldots, a_n\}$ and $\{\mathbf{E}_{\xi_1}[\hat{x}_1], \ldots, \{\mathbf{E}_{\xi_n}[\hat{x}_n]\}$. However, as discussed in Section 3.1.1, $\mathbf{E}_{\xi_i}[\hat{x}_i] = x_i$ for many types of noise. Hence, $\mathbf{E}_{\xi_i}[\hat{x}_i]$'s do not need to be explicitly stored, and the additional space is reduced to $O(n)$. This is significantly smaller than that of S-MISO, which always requires $O(nd)$ additional space.

### 3.2. Convergence Analysis

In this section, we provide convergence results for SSAG and S-SAGA on problem (4).

#### 3.2.1. SSAG

For SSAG, we make the following additional assumptions.

**Assumption 2.** *$F$ is $\mu$-strongly convex, i.e., $F(\theta') \geq F(\theta) + \langle\nabla F(\theta), \theta' - \theta\rangle + \frac{\mu}{2}\|\theta' - \theta\|^2, \forall\theta, \theta'$.*

**Assumption 3.** *$\mathbf{E}[(\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1}) - a_t)^2\|\hat{x}_{i_t}\|^2] \leq \sigma_a^2$ for all $t$.*

Let the minimizer of (4) be $\theta_*$. The following Theorem shows that SSAG has $O(1/t)$ convergence rate, which is similar to SGD (Bottou et al., 2016).

**Theorem 1.** *Assume that $\eta_t = c/(\gamma + t)$ for some $c > 1/\mu$ and $\gamma > 0$ such that $\eta_1 \leq 1/L$. For the $\{\theta_t\}$ sequence generated from SSAG, we have*

$$\mathbf{E}[F(\theta_t)] - F(\theta_*) \leq \frac{\nu_1}{\gamma + t + 1}, \quad (14)$$

*where $\nu_1 \equiv \max\left\{\frac{c^2L\sigma_a^2}{2(c\mu - 1)}, (\gamma + 1)C_1\right\}$, and $C_1 = F(\theta_0) - F(\theta_*)$.*

Note that this $\eta_t$ also satisfies the conditions in Proposition 3. The condition $c > 1/\mu$ is crucial to obtaining the $O(1/t)$ rate. It has been observed that underestimating $c$ can make convergence extremely slow (Nemirovski et al., 2009). When the model is $\ell_2$-regularized, $\mu$ can be estimated by the corresponding regularization parameter.

**Corollary 1.** *To ensure that $\mathbf{E}[F(\theta_t)] - F(\theta_*) \leq \epsilon$, SSAG has a time complexity of $O(n + \kappa C_1/\epsilon + \sigma_a^2\kappa^2/\epsilon)$, where $\kappa \equiv L/\mu$ is the condition number.*

The $O(n)$ term is due to initialization of $m_0$ and amortized over multiple data passes. In contrast, the time complexity for SGD is $O(\kappa C_1/\epsilon + \sigma_s^2\kappa^2/\epsilon)$, where $\sigma_s^2$ is defined in (6) (Bottou et al., 2016). To compare $\sigma_s^2$ with $\sigma_a^2$, we assume that the perturbed samples have finite variance $\sigma_x^2$:

$$\mathbf{E}[\|\hat{x} - \mathbf{E}[\hat{x}]\|^2] = \sigma_x^2.$$

The variance of the SGD estimator $g_t$ can be bounded as

$$\begin{aligned}
&\mathbf{E}[\|g_t - \nabla F(\theta_{t-1})\|^2 \\
&= \mathbf{E}[\|\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\hat{x}_{i_t} - \mathbf{E}[\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\hat{x}_{i_t}]\|^2] \\
&\leq 3\mathbf{E}[\|\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\hat{x}_{i_t} - a_t\hat{x}_{i_t}\|^2] \\
&\quad +3\mathbf{E}[\|a_t\hat{x}_{i_t} - a_t\mathbf{E}[\hat{x}_{i_t}]\|^2 \\
&\quad +3\mathbf{E}[\|a_t\mathbf{E}[\hat{x}_{i_t}] - \mathbf{E}[\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})\hat{x}_{i_t}]\|^2] \\
&\lessapprox 3\sigma_a^2 + 3a_t^2\sigma_x^2.
\end{aligned}$$

Thus, the gradient variance of SGD has two terms, one involving $\sigma_a^2$ and the other involving $a_t\sigma_x^2$. In particular, if the derivative $\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})$ is constant, then $\sigma_a^2 = 0$, and only the perturbed sample variance $\sigma_x^2$ contributes to the gradient variance of SGD. For a large class of functions including the logistic loss and smoothed hinge loss, $\phi'_{i_t}(\hat{x}_{i_t}^T\theta_{t-1})$ is nearly constant in some regions. In this case, we have $a_t^2\sigma_x^2 \approx \sigma_s^2$.

#### 3.2.2. S-SAGA

Besides Assumption 1, we assume the following:

**Assumption 4.** *Each $f_i(\theta; \xi_i)$ is $\mu$-strongly convex, i.e., $f_i(\theta'; \xi_i) \geq f_i(\theta; \xi_i) + \langle\nabla f_i(\theta; \xi_i), \theta' - \theta\rangle + \frac{\mu}{2}\|\theta' - \theta\|^2, \forall\theta, \theta'$.*

**Assumption 5.** *For all $t$, $\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}_{\xi_i,\xi_i'}[(\phi_i'(\hat{x}_i'^T\theta_{t-1}) - \phi_i'(\hat{x}_i^T\theta_{t-1}))^2\|\hat{x}_i\|^2] \le \sigma_c^2$, where $\hat{x}_i = \psi(x_i,\xi_i)$, $\hat{x}_i' = \psi(x_i,\xi_i')$, and $\xi_i'$ is another randomly sampled noise for $x_i$.*

**Theorem 2.** *Assume that $\eta_t = c/(\gamma+t)$ for some $c > 1/\mu$ and $\gamma > 0$ such that $\eta_1 \le 1/(3(\mu n + L))$. For the $\{\theta_t\}$ sequence generated from S-SAGA, we have*

$$\mathbf{E}[\|\theta_t - \theta_*\|^2] \le \frac{\nu_2}{\gamma+t+1}, \tag{15}$$

*where $\nu_2 \equiv \max\left(\frac{4c^2\sigma_c^2}{c\mu-1}, (\gamma+1)C_2\right)$, and $C_2 \equiv \|\theta_0 - \theta_*\|^2 + \frac{2n}{3(\mu n+L)}[F(\theta_0) - F(\theta_*)]$.*

Thus, S-SAGA has a convergence rate of $O(\sigma_c^2\kappa^2/t)$. In comparison, the convergence rate of SGD is $O(\sigma_s^2\kappa^2/t)$. Note that $\sigma_s^2$ in (6) includes variance due to data sampling, while $\sigma_c^2$ above only considers that due to noise. Since data sampling induces a much larger variation than that from perturbing the same sample, typically we have $\sigma_c^2 \ll \sigma_s^2$, and thus S-SAGA has faster convergence.

S-MISO considers the variance of the difference in gradients due to noise:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{E}_{\xi,\xi'}[\|\phi_i'(\hat{x}_i'^T\theta_{t-1})\hat{x}_i' - \phi_i'(\hat{x}_i^T\theta_{t-1})\hat{x}_i\|^2] \le \sigma_m^2,$$

and its convergence rate is $O(\sigma_m^2\kappa^2/t)$. The bounds for $\sigma_m^2$ and $\sigma_c^2$ are similar in form. However, $\sigma_c^2$ can be small when the difference $\phi'(\hat{x}_i^T\theta) - \phi'(\hat{x}_i'^T\theta)$ is small, while it is not the case for $\sigma_m^2$. In particular, when $\phi'(\hat{x}^T\theta)$ is a constant regardless of random perturbations, $\sigma_c^2 = 0$.

The following Corollary considers the time complexity of S-SAGA.

**Corollary 2.** *To ensure that $\mathbf{E}[F(\theta_t)] - F(\theta_*) \le \epsilon$, S-SAGA has a time complexity of $O((n+\kappa)C_2/\epsilon + \sigma_c^2\kappa^2/\epsilon)$.*

**Remark 1.** *In (Bietti & Mairal, 2017), additional speedup can be achieved by first running the algorithm with a constant stepsize for a few epochs, and then applying the decreasing stepsize. This trick is not used here. If incorporated, it can be shown that the $C_2/\epsilon$ term will be improved to $\log(C_2/\bar{\epsilon})$.*

A summary of the convergence results is shown in Table 1. As can be seen, by exploiting the linear model structure, SSAG has a smaller variance constant than SGD ($\sigma_a^2$ vs $\sigma_s^2$) while having comparable space requirement. S-SAGA improves over S-MISO and achieves gains both in terms of iteration complexity and storage.

### 3.3. Acceleration By Iterate Averaging

The complexity bounds in Corollaries 1 and 2 depend quadratically on the condition number $\kappa$. This may be problematic on ill-conditioned problems. To alleviate this problem, one can use iterate averaging (Bietti & Mairal, 2017), which outputs

*Table 1.* Iteration complexity and extra storage of different methods for solving optimization problem (4). For simplicity of comparison, we drop the constant $C$.

| | iteration complexity | space |
|---|---|---|
| SGD | $O(\kappa/\epsilon + \sigma_s^2\kappa^2/\epsilon)$ | $O(d)$ |
| S-MISO | $O((n+\kappa)/\epsilon + \sigma_m^2\kappa^2/\epsilon)$ | $O(nd)$ |
| SSAG | $O(n+\kappa/\epsilon + \sigma_a^2\kappa^2/\epsilon)$ | $O(d)$ |
| S-SAGA | $O((n+\kappa)/\epsilon + \sigma_c^2\kappa^2/\epsilon)$ | $O(n)$ |

$$\bar{\theta}_T \equiv \frac{2}{T(2\gamma+T-1)}\sum_{t=0}^{T-1}(\gamma+t)\theta_t, \tag{16}$$

where $T$ is the total number of iterations. It can be easily seen that (16) can be efficiently implemented in an online fashion without the need for storing $\theta_t$'s:

$$\bar{\theta}_t = (1-\rho_t)\bar{\theta}_{t-1} + \rho_t\theta_{t-1},$$

where $\rho_t = \frac{2(\gamma+t-1)}{t(2\gamma+t-1)}$ and $\bar{\theta}_0 = 0$. As in (Bietti & Mairal, 2017), the following shows that the $\kappa^2$ dependence in both SSAG and S-SAGA (Corollaries 1 and 2) can be reduced to $\kappa$.

**Theorem 3.** *Assume that $\eta_t = 2/(\mu(\gamma+t))$ and $\gamma > 0$ such that $\eta_1 \le 1/(2L)$. For the $\{\theta_t\}$ sequence generated from SSAG, we have*

$$\mathbf{E}[F(\bar{\theta}_T)] - F(\theta_*) \le \frac{\mu\gamma(\gamma-1)}{T(2\gamma+T-1)}\|\theta_0-\theta_*\|^2 + \frac{4\sigma_a^2}{\mu(2\gamma+T-1)}.$$

The stepsize $\eta_t = 2/(\mu(\gamma+t))$ and condition $\eta_1 \le 1/2L$ together implies that $\gamma = O(\kappa)$. Thus, when $T \ll \gamma$, the first term, which depends on $\|\theta_0-\theta_*\|^2$, decays as $1/T$, which is no better than (14). On the other hand, if $T \gg \gamma$, the first term decays at a faster $\kappa/T^2$ rate.

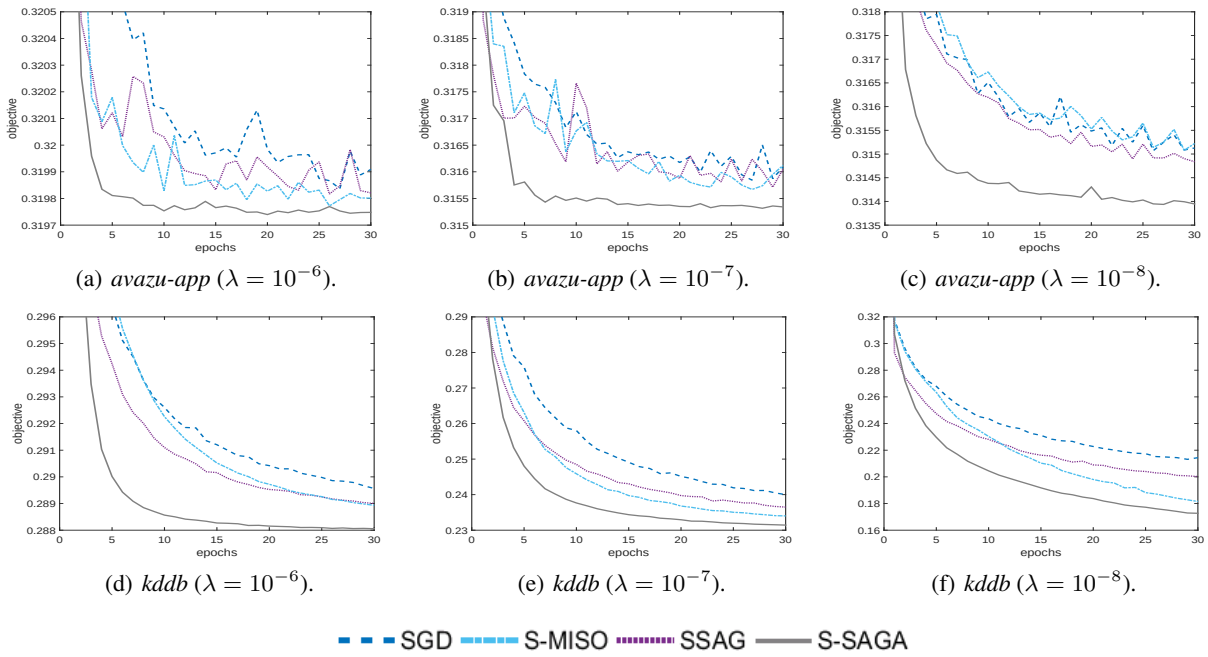
**Corollary 3.** *When $T \gg \gamma$, to ensure that $\mathbf{E}[F(\bar{\theta}_T)] - F(\theta_*) \le \epsilon$, SSAG with iterate averaging has a time complexity of $O(n + \sqrt{\kappa C_3}/\sqrt{\epsilon} + \sigma_a^2\kappa/\epsilon)$, where $C_3 = \|\theta_0 - \theta_*\|^2$.*

Similarly, we have the following for S-SAGA.

**Theorem 4.** *Assume that $\eta_t = c/(\gamma+t)$ for some $c > 1/\mu$ and $\gamma > 0$ such that $\eta_1 \le 1/(7(\mu n + L))$. For the $\{\theta_t\}$ sequence generated from S-SAGA, we have*

$$\mathbf{E}[F(\bar{\theta}_T) - F(\theta_*)] \le \frac{\mu\gamma(\gamma-1)}{2T(2\gamma+T-1)}C_4 + \frac{32\sigma_c^2}{\mu(2\gamma+T-1)}, \tag{17}$$

*where $C_4 \equiv 3\|\theta_0-\theta_*\|^2 + \frac{4n}{7(\mu n+L)}[F(\theta_0) - F(\theta_*)]$.*




(a) *avazu-app* ($\lambda = 10^{-6}$). (b) *avazu-app* ($\lambda = 10^{-7}$). (c) *avazu-app* ($\lambda = 10^{-8}$).

(d) *kddb* ($\lambda = 10^{-6}$). (e) *kddb* ($\lambda = 10^{-7}$). (f) *kddb* ($\lambda = 10^{-8}$).

*Figure 1.* Convergence with the number of epochs (logistic regression with dropout). The regularization parameter $\lambda$ of $\ell_2$ regularizer is varied from $10^{-6}$ to $10^{-8}$. The dropout rate is fixed to 0.3.

The condition $\eta_1 \leq 1/(7(\mu n + L))$ is satisfied when $\gamma = O(n + \kappa)$. Thus, the second term in $C_4$ is scaled by $4n/(7(\mu n + L)) = O(n/(\mu\gamma))$. These implies that the first term in (17) decays as $n/T$ when $T \ll \gamma$. On the other hand, when $T \gg \gamma$, the first term decays as $n(n + \kappa)/T^2$. Thus, iterate averaging does not provide S-SAGA with much acceleration as compared to SSAG.

The following Corollary considers the case where $n = O(\kappa)$ (Johnson & Zhang, 2013).

**Corollary 4.** *Assume that $n = O(\kappa)$. When $T \gg \gamma$, to ensure that $\mathbf{E}[F(\bar{\theta}_T)] - F(\theta_*) \leq \epsilon$, S-SAGA with iterate averaging has a time complexity of $O(n + \sqrt{(n+\kappa)C_4}/\sqrt{\epsilon} + \sigma_c^2\kappa/\epsilon)$.*

## 4. Experiments

In this section, we perform experiments on logistic regression (Section 4.1) and AUC maximization (Section 4.2).

### 4.1. Logistic Regression with Dropout

Consider the $\ell_2$-regularized logistic regression model with dropout noise, with dropout probability $p = 0.3$. This can be formulated as the following optimization problem:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \mathbf{E}_{\hat{\xi}_i}[\log(1 + \exp(-y_i \hat{z}_i^T \theta))] + \frac{\lambda}{2}\|\theta\|^2, \quad (18)$$

where $\hat{z}_i = \psi(z_i, \xi_i)$, $z_i$ is the feature vector of sample $i$, and $y_i$ the corresponding class label. We vary $\lambda \in \{10^{-6}, 10^{-7}, 10^{-8}\}$. The smaller the $\lambda$, the higher the condition number. Experiments are performed on two high-dimensional data sets from the LIBSVM archive (Table 2).

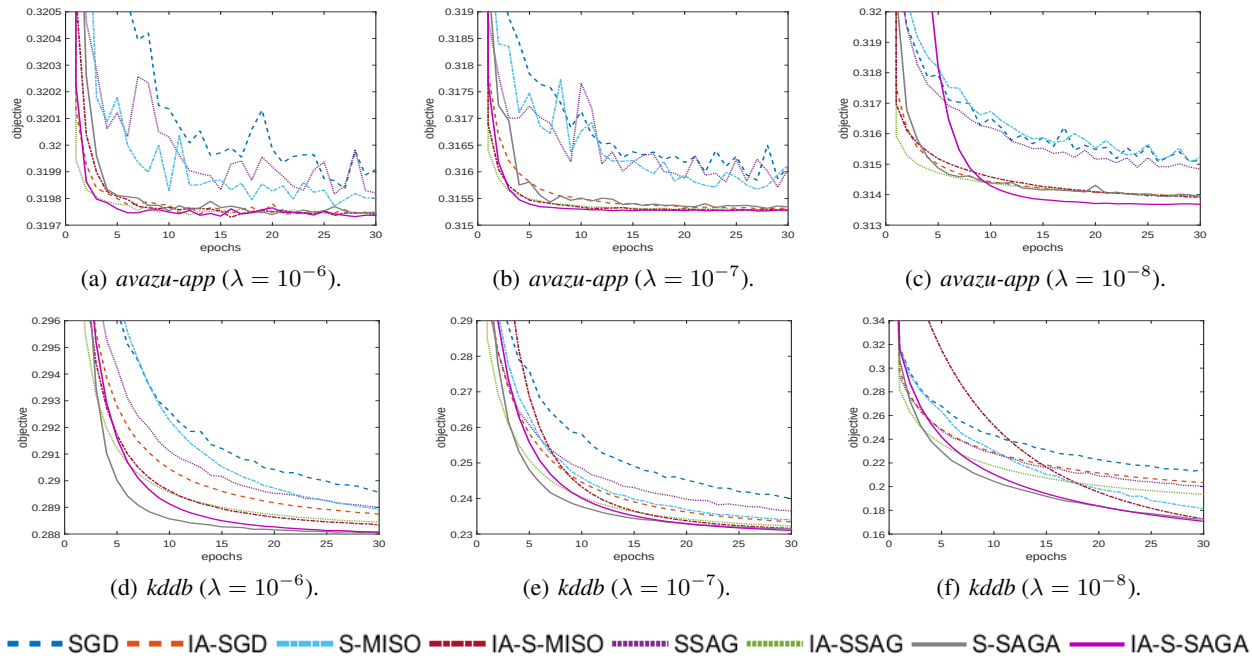
*Table 2.* Data sets used in the logistic regression experiment.

| | #training | #testing | dimensionality |
|---|---|---|---|
| *avazu-app* | 12,642,186 | 1,953,951 | 1,000,000 |
| *kddb* | 19,264,097 | 748,401 | 29,890,095 |

#### 4.1.1. Comparison with SGD and S-MISO

The proposed SSAG and S-SAGA are compared with SGD and S-MISO. From Proposition 4, we use a slightly larger $\beta_t = t^{-0.75}$ for better non-asymptotic performance. As mentioned in the theorems, the stepsize schedule is $\eta_t = c/(\gamma + t)$. We fix $c = 2/\lambda$ for SGD, SSAG, S-SAGA, and $c = 2n$ for S-MISO as suggested in (Bietti & Mairal, 2017). We then select $\gamma$ from a number of possible values (e.g., powers of tens and five times powers of tens) by monitoring the training objective. To reduce statistical variability, results are averaged over five repetitions.

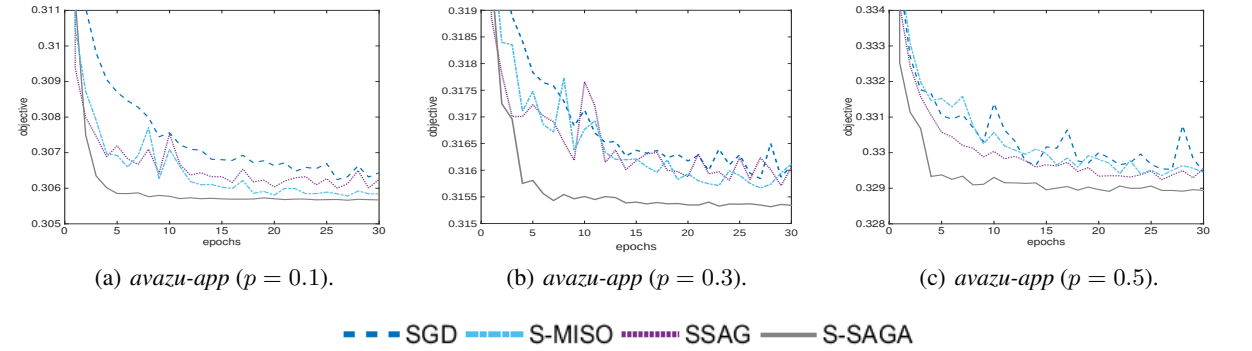
As all methods under comparison have the same iteration complexities, Figure 1 shows convergence of the training objective with the number of epochs. The expectation in (18) is estimated from 5 perturbed samples. As can be seen, S-SAGA significantly outperforms all the others. In par-




(a) *avazu-app* ($\lambda = 10^{-6}$). (b) *avazu-app* ($\lambda = 10^{-7}$). (c) *avazu-app* ($\lambda = 10^{-8}$).

(d) *kddb* ($\lambda = 10^{-6}$). (e) *kddb* ($\lambda = 10^{-7}$). (f) *kddb* ($\lambda = 10^{-8}$).

*Figure 2.* Convergence with the number of epochs (both methods with and without iterate averaging are included). The experiment is performed on the same task as in Figure 1 but with more algorithms included.




(a) *avazu-app* ($p = 0.1$). (b) *avazu-app* ($p = 0.3$). (c) *avazu-app* ($p = 0.5$).

*Figure 3.* Convergence with the number of epochs (logistic regression with dropout). The dropout probability is varied from $0.1$ to $0.5$.

ticular, it reaches a much lower objective value when the condition number is large ($\lambda = 10^{-8}$). SSAG and S-MISO have similar convergence behavior and converge faster than SGD. However, S-MISO requires much more memory than SSAG. A comparison of the additional memory (relative to SGD) used by each method is shown in Table 3.

*Table 3.* Additional memory (relative to SGD) required by the various algorithms in the logistic regression experiment.

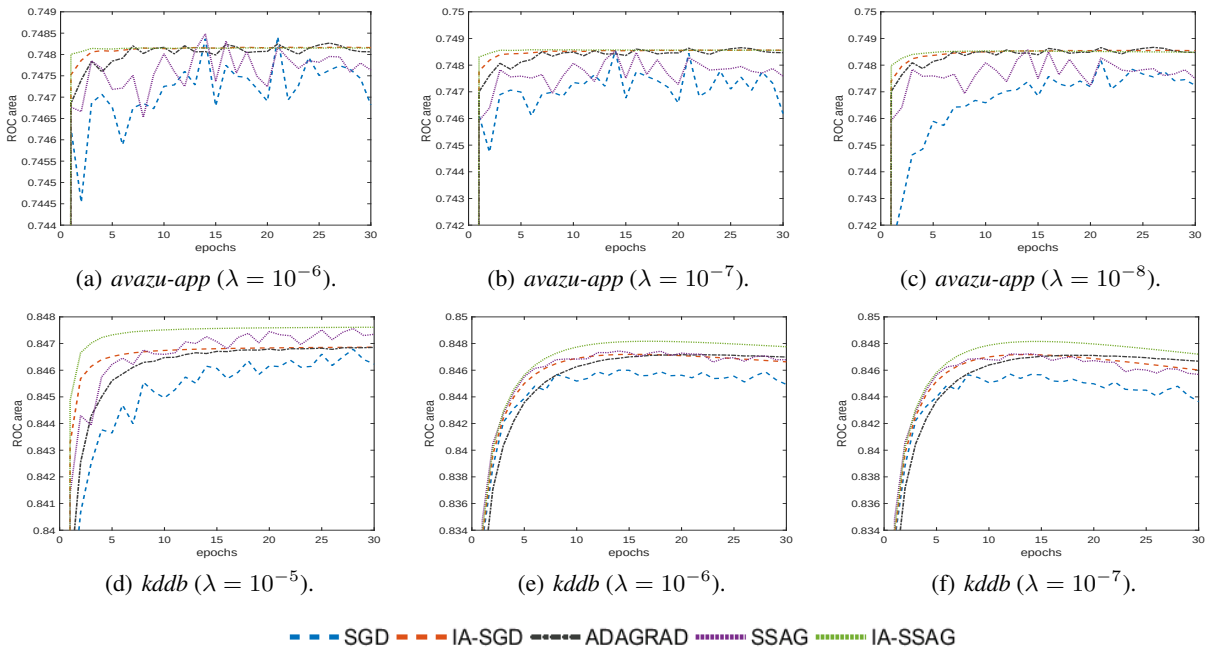
| | S-MISO | SSAG | S-SAGA |
|---|---|---|---|
| *avazu-app* | 3.1GB | 7.6MB | 104.1MB |
| *kddb* | 8.9GB | 147 MB | 375MB |

To see how $a_t$ differs from $a_t^*$ in (10), we perform an experiment using a subset of *covertype* data from the LIBSVM archive. The expectations in $a_t^*$ are again approximated by randomly sampling 5 perturbations for each sample. Empirically, $\max_{t \geq 2} |a_t - a_t^*|/|a_t^*|$ is of the order $0.01$, indicating that $a_t$ is a reasonable estimate even in the early iterations.

#### 4.1.2. Use of Iterate Averaging

Figure 2 adds the convergence results for iterate averaging to Figure 1 ("IA" is prepended to the names of algorithms using iterate averaging). As can be seen, iterate averaging leads to significant improvements for SGD, S-MISO and SSAG, but less prominent improvement for S-SAGA. This agrees with the discussions in Section 3.3. Moreover, when the condition number is high, SSAG has similar convergence as IA-SGD on *kddb*. This demonstrates that SSAG is more robust to large condition number.

Overall, when memory is not an issue, S-SAGA is preferred for problems with small or medium condition num-




(a) *avazu-app* ($\lambda = 10^{-6}$). (b) *avazu-app* ($\lambda = 10^{-7}$). (c) *avazu-app* ($\lambda = 10^{-8}$).

(d) *kddb* ($\lambda = 10^{-5}$). (e) *kddb* ($\lambda = 10^{-6}$). (f) *kddb* ($\lambda = 10^{-7}$).

*Figure 4.* Convergence of AUC with the number of epochs.

bers, while IA-S-SAGA can be better for problems with large condition numbers. If memory is limited, IA-SSAG is recommended.

#### 4.1.3. Varying the Dropout Probability

In this section, we study how the strength of the dropout noise affects convergence. We use the *avazu-app* data set, and fix $\lambda = 10^{-7}$. The dropout probability $p$ is varied in $\{0.1, 0.3, 0.5\}$. Note that a larger dropout probability leads to larger noise variance. Figure 3 shows that S-SAGA is very robust to different noise levels, while S-MISO performs much worse when the dropout probability increases. This demonstrates the theoretical result in Theorem 2 that S-SAGA has a smaller variance constant, while SGD and SSAG are not sensitive to $p$.

### 4.2. AUC Maximization with Dropout

In this section, we consider maximization of the AUC (i.e., area under the ROC curve). This is equivalent to ranking the positive samples higher than the negative samples (Sculley, 2009). It can be formulated as minimizing the following objective with the squared hinge loss:

$$\frac{1}{n^+ n^-} \sum_{y_i=1, y_j=0} \mathbf{E}_{\xi_i,\xi_j}[\max(0, 1 - (\hat{z}_i - \hat{z}_j)^T\theta)^2] + \frac{\lambda}{2}\|\theta\|^2,$$

where $n^+, n^-$ are the numbers of samples belonging to the positive and negative class, respectively. We again use the data sets in Table 2, and inject dropout noise with dropout probability $p = 0.3$.

Even without noise perturbation, AUC maximization is infeasible for existing variance reduction methods. Methods such as SAG and SAGA need $O(n^+n^-)$ space. SVRG trades space with time, and takes $O(n^+n^-)$ time. With dropout noise injected, S-MISO requires even more space, namely, $O(n^+n^-d)$. S-SAGA requires $O(n^+n^-)$ space, and so is also impractical. Thus, in the following, we only compare SGD, SSAG and their variants with iterate averaging. As a further baseline, we also compare with ADAGRAD (Duchi et al., 2011), which performs SGD with an adaptive learning rate.

Figure 4 shows the results. IA-SSAG is always the fastest, and has the highest AUC on *kddb*. ADAGRAD and IA-SGD have comparable AUC with IA-SSAG on *avazu-app*, but not on *kddb*. ADAGRAD is faster than SGD and SSAG on *avazu-app*, but slower than SSAG on *kddb*. On *kddb*, SSAG has comparable performance with IA-SGD, and is better when $\lambda = 10^{-5}$.

## 5. Conclusion

In this paper, we proposed two SGD-like algorithms for finite sums with infinite data when learning with the linear model. The key is to exploit the linear structure in the construction of control variates. Convergence results on strongly convex problems are provided. The proposed methods require small memory cost. Experimental results demonstrate that the proposed algorithms outperform the state-of-the-art on large data sets.

## References


Bietti, A. and Mairal, J. Stochastic optimization with variance reduction for infinite datasets with finite-sum structure. In *Advances in Neural Information Processing Systems*, pp. 1622–1632, 2017.

Bishop, C. M. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. Preprint arXiv:1606.04838, 2016.

Decoste, D. and Schölkopf, B. Training invariant support vector machines. *Machine learning*, 46(1-3):161–190, 2002.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 2116–2124, 2014a.

Defazio, A., Domke, J., and Caetano, T. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1125–1133, 2014b.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Fishman, G. S. *Monte Carlo: Concepts, Algorithms and Applications*. Springer, 1996.

Hofmann, T., Lucchi, A., Lacoste-Julien, S., and McWilliams, B. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems*, pp. 2305–2313, 2015.

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

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

Mairal, J. Optimization with first-order surrogate functions. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

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

Paulin, M., Revaud, J., Harchaoui, Z., Perronnin, F., and Schmid, C. Transformation pursuit for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3646–3653, 2014.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Sculley, D. Large scale learning to rank. In *NIPS 2009 Workshop on Advances in Ranking*, 2009.

Shalev-Shwartz, S. and Zhang, T. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

van der Maaten, L., Chen, M., Tyree, S., and Weinberger, K. Learning with marginalized corrupted features. In *International Conference on Machine Learning*, pp. 410–418, 2013.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11 (12):3371–3408, 2010.

Wager, S., Wang, S., and Liang, P. S. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pp. 351–359, 2013.

Wager, S., Fithian, W., Wang, S., and Liang, P. S. Altitude training: Strong bounds for single-layer dropout. In *Advances in Neural Information Processing Systems*, pp. 100–108, 2014.

Xiao, L. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.