Accelerating Minibatch Stochastic Gradient Descent using Stratified Sampling

Peilin Zhao
Department of Statistics
Rutgers University
Piscataway, NJ, 08854, USA
peilin.zhao@rutgers.edu

Tong Zhang
Baidu Inc. & Rutgers University
Piscataway, NJ, 08854, USA
tzhang@stat.rutgers.edu

Abstract

Stochastic Gradient Descent (SGD) is a popular optimization method which has been applied to many important machine learning tasks such as Support Vector Machines and Deep Neural Networks. In order to parallelize SGD, minibatch training is often employed. The standard approach is to uniformly sample a minibatch at each step, which often leads to high variance. In this paper we propose a stratified sampling strategy, which divides the whole dataset into clusters with low within-cluster variance; we then take examples from these clusters using a stratified sampling technique. It is shown that the convergence rate can be significantly improved by the algorithm. Encouraging experimental results confirm the effectiveness of the proposed method.

1   Introduction

Stochastic Gradient Descent has been extensively studied in the machine learning community [11, 2, 8, 6, 5, 9, 3]. At every step, a typical stochastic gradient descent method will sample one training example uniformly at random from the training data, and then update the model parameter using the sampled example. In order to parallelize SGD, the standard approach is to employ minibatch training, which samples multiple examples uniformly at each step. The uniformly sampled minibatch stochastic gradient is an unbiased estimation of the true gradient [11, 6, 9, 2], but the resulting estimator may have relatively high variance. This will negatively affect the convergence rate of the underlying optimization procedure. Instead of using uniform sampling of the training data, we propose to divide the whole dataset into clusters and perform stratified sampling that minimizes an upperbound of the variance. We show that the proposed algorithm can significantly reduce the stochastic variance, which will then improve convergence.

The key idea in the proposed approach is to perform stratified sampling and construct the corresponding unbiased stochastic gradient estimators that minimize an upperbound of the stochastic variance. To this end, we analyze the relationship between the variance of stochastic gradient estimator and the sampling strategy. We show that to minimize the variance, the optimal sampling strategy should roughly minimize a sum of all the weighted standard deviations of gradients corresponding to the subdatasets. Our theoretical analysis shows that under certain conditions, the proposed sampling method can significantly improve the convergence rate. This result is empirically verified by experiments.

The rest of this paper is organized as follows. Section 2 reviews the related work. In section 3, we study minibatch stochastic gradient descent with stratified sampling. The empirical evaluations are presented in Section 4. Section 5 concludes the paper.

2   Related Work

Stochastic Gradient Descent has been extensively studied in the traditional stochastic approximation literature [4]; however the results are often asymptotic. In recent years, finite sample convergence rate of SGD for solving linear prediction problems have been studied by a number of authors [11, 7]. In general SGD can achieve a convergence rate of $O(1/\sqrt{T})$ for convex loss functions, and a convergence rate of $O(\log T/T)$ for strongly convex loss functions, where T is the number of iterations of the algorithm. More recently, researchers have improved the previous bound for the strongly convex loss function case to $O(1/T)$ by using α-Suffix Averaging [6], which means that the algorithm will return the average of the last α fraction of the previously obtained sequence of predictors. A similar result can be obtained via a polynomial decay averaging strategy [9].

Although SGD has been extensively studied, most of the existing work only considered the uniform sampling scheme during the entire learning process, which will result in an unbiased estimator with high variance. To explicitly reduce the variance, some new stochastic gradient algorithms have been proposed [3, 10, 12]. In [3], the authors constructed an unbiased stochastic estimator of the full gradient called SVRG, with the property that the resulting variance will approach zero asymptotically in the finite training example case. In [10], the authors developed a variance reduction approach with control variates formed using low-order moments. However they still employ uniform sampling during the training process.

Instead of using uniform sampling, stochastic gradient descent with importance sampling was studied in [12], where a nonuniform sampling distribution is constructed to reduce the variance of the stochastic gradient estimator. This paper considers a different variance reduction method using stratified sampling for minibatch SGD training. This idea is complementary to previously proposed variance reduction methods such as SVRG in [3] and importance sampling in [12]. In fact, these methods can be combined.

3   Minibatch SGD with Stratified Sampling

3.1   Preliminaries
We briefly introduce some key definitions and a property of convex functions that are useful throughout the paper (for details, please refer to [1]).

Definition 1. *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is H-strongly convex, if for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, we have*

$$\phi(\mathbf{u}) \geq \phi(\mathbf{v}) + \nabla\phi(\mathbf{v})^\top(\mathbf{u} - \mathbf{v}) + \frac{H}{2}\|\mathbf{u} - \mathbf{v}\|^2,$$

where $\|\cdot\|$ is a norm.

Definition 2. *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is L-Lipschitz, if for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, we have*

$$|\phi(\mathbf{u}) - \phi(\mathbf{v})| \leq L\|\mathbf{u} - \mathbf{v}\|.$$

Definition 3. *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is $(1/\gamma)$-smooth if it is differentiable and its gradient is $(1/\gamma)$-Lipschitz, or, equivalently for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, we have*

$$\phi(\mathbf{u}) \leq \phi(\mathbf{v}) + \nabla\phi(\mathbf{v})^\top (\mathbf{u} - \mathbf{v}) + \frac{1}{2\gamma}\|\mathbf{u} - \mathbf{v}\|^2.$$

Property 1. *If a function $\phi : \mathbb{R}^d \to \mathbb{R}$ is convex (0-strongly convex) and $(1/\gamma)$-smooth, then for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, we have*

$$\langle \nabla\phi(\mathbf{u}) - \nabla\phi(\mathbf{v}), \mathbf{u} - \mathbf{v} \rangle \geq \gamma\|\nabla\phi(\mathbf{u}) - \nabla\phi(\mathbf{v})\|^2,$$

which is known as co-coercivity of $\nabla\phi$ with parameter γ.

Throughout this paper, we will denote $\|\cdot\|_2$ as $\|\cdot\|$ for simplicity.

3.2 Problem Setting

In this paper, we will focus on the standard multiclass classification task, for which a set of examples is given as $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where each $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional instance and $y_i \in \{1, 2, \ldots, m\}$ is the class label assigned to $\mathbf{x}_i$. Given this set of examples, we learn a classifier $\mathbf{w}$ to predict the label y of $\mathbf{x} \in \mathbb{R}^d$. To learn the classifier, a loss function $\ell(\mathbf{w}; \mathbf{x}, y)$ will be introduced to penalize the deviation of the prediction of $\mathbf{w}$ on $\mathbf{x}$ from the true label y. In this problem setting, our goal is to find an approximate solution of the following optimization problem

$$\min_{\mathbf{w} \in \mathbb{R}^d} P(\mathbf{w}) \qquad P(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^n \phi_i(\mathbf{w}),$$

where $\phi_i(\mathbf{w}) = \ell(\mathbf{w}, \mathbf{x}_i, y_i) + \frac{\lambda}{2}\|\mathbf{w}\|^2$, and λ is a regularization parameter.

To solve the above optimization, a standard method is gradient descent, which can be described by the following update rule for $t = 1, 2, \ldots$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla P(\mathbf{w}_t) = \mathbf{w}_t - \frac{\eta_t}{n}\sum_{i=1}^n \nabla\phi_i(\mathbf{w}_t).$$

However, at each step, gradient descent requires the calculation of n derivatives, which is expensive when n is very large. A popular modification is Stochastic Gradient Descent (SGD): at each iteration $t = 1, 2, \ldots$, we draw i_t uniformly randomly from $[n] := \{1, \ldots, n\}$, and let

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla\phi_{i_t}(\mathbf{w}_t).$$

Because $\mathbb{E}[\nabla\phi_{i_t}(\mathbf{w}_t)|\mathbf{w}_t] = \frac{1}{n}\sum_{i=1}^n \nabla\phi_i(\mathbf{w}_t) = \nabla P(\mathbf{w}_t)$, the expectation $\mathbb{E}[\mathbf{w}_{t+1}|\mathbf{w}_t]$ equals the full gradient. The advantage of stochastic gradient is that each step only relies on a single derivative $\nabla\phi_{i_t}(\mathbf{w}_t)$, and thus the computational cost is $1/n$ that of the standard gradient descent. However, a disadvantage of the method is that the randomness introduces variance, which is caused by the fact that $\nabla\phi_{i_t}(\mathbf{w}_t)$ equals the gradient $\nabla P(\mathbf{w}_t)$ in expectation but each $\nabla\phi_i(\mathbf{w}_t)$ is different. In particular, if it has a large variance, then the convergence will be slow.

3

For example, consider the case that each $\phi_i(\mathbf{w})$ is $(1/\gamma)$-smooth, then we have

$$
\begin{aligned}
\mathbb{E}P(\mathbf{w}_{t+1}) &= \mathbb{E}P\big(\mathbf{w}_t - \eta_t \nabla \phi_{i_t}(\mathbf{w}_t)\big) \le \mathbb{E}P(\mathbf{w}_t) - \eta_t \|\mathbb{E}\nabla P(\mathbf{w}_t)\|^2 + \frac{\eta_t^2}{2\gamma}\mathbb{E}\|\nabla \phi_{i_t}(\mathbf{w}_t)\|^2 \\
&\le \mathbb{E}P(\mathbf{w}_t) - \eta_t(1 - \frac{\eta_t}{2\gamma})\mathbb{E}\|\nabla P(\mathbf{w}_t)\|^2 + \frac{\eta_t^2}{2\gamma}\mathbb{V}(\nabla \phi_{i_t}(\mathbf{w}_t)),
\end{aligned}
$$

where the variance is

$$
\mathbb{V}(\nabla \phi_{i_t}(\mathbf{w}_t)) = \mathbb{E}\|\nabla \phi_{i_t}(\mathbf{w}_t) - \nabla P(\mathbf{w}_t)\|^2.
$$

From the above inequality, we can observe that the smaller the variance, the more reduction on the objective function we have. To reduce the variance, a typical method is to i.i.d. uniformly sample a mini-batch of indices $\mathcal{B}_t = \{i_t^j \in \{1, 2, \ldots, n\} | j = 1, 2, \ldots, b\}$ with replacement from the set of indices, and then update the classifier as

$$
\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{b} \sum_{s \in \mathcal{B}_t} \nabla \phi_s(\mathbf{w}_t),
$$

which equals GD update in expectation. For this update method, the reduction on the objective value can be similarly computed as:

$$
\mathbb{E}P(\mathbf{w}_{t+1}) \le \mathbb{E}P(\mathbf{w}_t) - \eta_t(1 - \frac{\eta_t}{2\gamma})\mathbb{E}\|\nabla P(\mathbf{w}_t)\|^2 + \frac{\eta_t^2}{2\gamma b}\mathbb{V}(\nabla \phi_{i_t^1}(\mathbf{w}_t))
$$

using the fact that every $s \in \mathcal{B}_t$ is i.i.d. uniformly sampled from $\{1, 2, \ldots, n\}$. Although the variance is reduced to $1/b$ of the one for using a single uniformly sampled example, the computation becomes b times that of the standard SGD. Nevertheless, minibatch training is needed to parallelize the SGD algorithm and has been commonly used in practice. In the next subsection, we will show that the variance of minibatch SGD can be significantly reduced if we employ an appropriate stratified sampling strategy instead of uniform sampling.

3.3 Algorithm
The main idea is as follows. At the t-th step, we will use some clustering method to separate the training set of indices $[n]$ into k clusters $\mathcal{C}_1^t, \mathcal{C}_2^t, \ldots, \mathcal{C}_k^t$, where $\mathcal{C}_i^t \subset [n]$, $\cup_{i=1}^k \mathcal{C}_i^t = [n]$, $\mathcal{C}_j^t \cap \mathcal{C}_i^t = \emptyset$ $\forall i \ne j$ and $|\mathcal{C}_i^t| = n_i^t$ (we also assume that each cluster contains only one class label). Given these k clusters, we will independently sample k subsets $\mathcal{B}_1^t, \mathcal{B}_2^t, \ldots, \mathcal{B}_k^t$ from $\mathcal{C}_1^t, \mathcal{C}_2^t, \ldots, \mathcal{C}_k^t$, respectively, where each index $s \in \mathcal{B}_i^t$ is i.i.d. uniformly sampled from $\mathcal{C}_i^t$, with $|\mathcal{B}_i^t| = b_i^t$, $\sum b_i^t := b$ and $\mathcal{B}^t = \cup_{i=1}^k \mathcal{B}_i^t$. Given these sampled indices, the proposed algorithm works as follows

$$
\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{n} \sum_{i=1}^k \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \nabla \phi_s(\mathbf{w}_t),
$$

which equals the GD update in expectation, since

$$
\begin{aligned}
\mathbb{E}\left[\frac{1}{n} \sum_{i=1}^k \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \nabla \phi_s(\mathbf{w}_t) | \mathbf{w}_t\right] &= \frac{1}{n} \sum_{i=1}^k \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \mathbb{E}_t \nabla \phi_s(\mathbf{w}_t) \\
&= \frac{1}{n} \sum_{i=1}^k \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \frac{1}{n_i^t} \sum_{s \in \mathcal{C}_i^t} \nabla \phi_s(\mathbf{w}_t) = \nabla P(\mathbf{w}_t).
\end{aligned}
$$

4

Assume the update methods for $\mathcal{C}_1^t, \mathcal{C}_2^t, \ldots, \mathcal{C}_k^t$ and $b_1^t, b_2^t, \ldots, b_k^t$ are provided, we can summarize the proposed method in Algorithm 1.

Algorithm 1 Stochastic Gradient Descent with Stratified Sampling (SGD-ss)

Input: Clusters Number k, Minibatch Size b.
Initialize: $\mathbf{w}_1 = 0$, $\{\mathcal{C}_1^1, \mathcal{C}_2^1, \ldots, \mathcal{C}_k^1\}$, $b_1^1, b_2^1, \ldots, b_k^1$, such that each $\mathcal{C}_i^1$ contains only one label.
for $t = 1, 2, \ldots, T$ **do**
  for $s = 1, 2, \ldots, k$ **do**
    $\mathcal{B}_s^t = \emptyset$;
    for $r = 1, 2 \ldots, b_s^t$ **do**
      Uniformly sample $i_r \in \mathcal{C}_s^t$ and $\mathcal{B}_s^t = \mathcal{B}_s^t \cup \{i_r\}$;
    end for
  end for
  Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{n} \sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \nabla \phi_s(\mathbf{w}_t)$;
  Update $\mathcal{C}_1^{t+1}, \mathcal{C}_2^{t+1}, \ldots, \mathcal{C}_k^{t+1}$ such that each $\mathcal{C}_i^{t+1}$ contains only one label;
  Update $b_1^{t+1}, b_2^{t+1}, \ldots, b_k^{t+1}$;
end for

For the proposed algorithm, the reduction on the objective function can be similarly computed as follows:

$$\mathbb{E}P(\mathbf{w}_{t+1}) \leq \mathbb{E}P(\mathbf{w}_t) - \eta_t(1 - \frac{\eta_t}{2\gamma})\mathbb{E}\|\nabla P(\mathbf{w}_t)\|^2 + \frac{\eta_t^2}{2\gamma}V(\mathcal{C}_1^t, \ldots, \mathcal{C}_k^t, b_1^t, \ldots, b_k^t),$$

where $V(\mathcal{C}_1^t, \ldots, \mathcal{C}_k^t, b_1^t, \ldots, b_k^t) = \mathbb{V}\left(\frac{1}{n} \sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \nabla \phi_s(\mathbf{w}_t)\right)$.

According to the above analysis, to maximize the reduction on the objective function, the optimal clusters and minibatch distribution can be obtained by solving the following optimization problem:

$$\min_{\mathcal{C}_1^t, \ldots, \mathcal{C}_k^t, b_1^t, \ldots, b_k^t} V(\mathcal{C}_1^t, \ldots, \mathcal{C}_k^t, b_1^t, \ldots, b_k^t) = \frac{1}{n^2} \sum_{i=1}^{k} (\frac{n_i^t}{b_i^t})^2 \sum_{s \in \mathcal{B}_i^t} \mathbb{V}(\nabla \phi_s(\mathbf{w}_t))$$

$$= \frac{1}{n^2} \sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{C}_i^t} \|\nabla \phi_s(\mathbf{w}_t) - \frac{1}{n_i^t} \sum_{r \in \mathcal{C}_i^t} \nabla \phi_r(\mathbf{w}_t)\|^2, \quad (1)$$

which can be considered as a dynamically weighted k-means problem, where the weights of gradients in the same cluster are the same and optimized with the clusters simultaneously.

Although, this sampling method can minimize the variance of the stochastic estimator, it requires the calculation of n derivatives and requires running a clustering algorithm for every iteration, which is clearly impractical. To address this issue, we assume that $\partial_{\mathbf{w}}\ell(\mathbf{w}; \mathbf{x}, y)$ is L-Lipschitz in $\mathbf{x}$ for fixed $\mathbf{w}$ and y. Now assume further that each cluster contains only one class label: $y_s = y_r$

$\forall s, r \in \mathcal{C}_i^t$. Under these assumptions, we can relax the previous expression of variance as follows:

$$V(\mathcal{C}_1^t, \ldots, C_k^t, b_1^t, \ldots, b_k^t)$$

$$= \frac{1}{n^2} \sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{C}_i^t} \left\| \partial_{\mathbf{w}} \ell(\mathbf{w}_t; \mathbf{x}_s, y_s) - \frac{1}{n_i^t} \sum_{r \in \mathcal{C}_i^t} \partial_{\mathbf{w}} \ell(\mathbf{w}_t; \mathbf{x}_r, y_r) \right\|^2$$

$$\leq \frac{1}{n^2} \sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{C}_i^t} \left\| \partial_{\mathbf{w}} \ell(\mathbf{w}_t; \mathbf{x}_s, y_s) - \partial_{\mathbf{w}} \ell(\mathbf{w}_t; \mu_i^t, y_s) \right\|^2$$

$$\leq \frac{L^2}{n^2} \sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{C}_i^t} \| \mathbf{x}_s - \mu_i^t \|^2,$$

where $\mu_i^t = \sum_{s \in \mathcal{C}_i^t} \mathbf{x}_s / n_i^t$. This relaxation inspires us to find an iteration-independent sampling strategy where $\mathcal{C}_i^t = \mathcal{C}_i$, $n_i^t = n_i$, and $b_i^t = b_i$, $\forall t, i$, by solving the following optimization problem, which corresponds to an upperbound of (1)

$$\min_{\mathcal{C}_i, b_i, y_s = y_r, \forall s, r \in \mathcal{C}_i} \frac{L^2}{n^2} \sum_{i=1}^{k} \frac{n_i}{b_i} \sum_{s \in \mathcal{C}_i} \left\| \mathbf{x}_s - \frac{1}{n_i} \sum_{r \in \mathcal{C}_i} \mathbf{x}_r \right\|^2. \tag{2}$$

The solution of this optimization problem can be pre-calculated and used at every iteration. Given $\mathcal{C}_1, \ldots, \mathcal{C}_k$, it is easy to verify that the optimal $b_1, \ldots, b_k$ can be calculated as (b_i is relaxed to take non-integer values):

$$b_i = \frac{b n_i \sqrt{v_i}}{\sum_{j=1}^{n} n_j \sqrt{v_j}}, \quad v_i = \frac{1}{n_i} \sum_{s \in \mathcal{C}_i} \left\| \mathbf{x}_s - \frac{1}{n_i} \sum_{r \in \mathcal{C}_i} \mathbf{x}_r \right\|^2, \tag{3}$$

and we can simplify (2) to the following optimization problem

$$\min_{\mathcal{C}_i, y_s = y_r, \forall s, r \in \mathcal{C}_i} \sum_{i=1}^{k} n_i \sqrt{\frac{1}{n_i} \sum_{s \in \mathcal{C}_i} \left\| \mathbf{x}_s - \frac{1}{n_i} \sum_{r \in \mathcal{C}_i} \mathbf{x}_r \right\|^2}. \tag{4}$$

which can be solved by a k-means style alternating optimization algorithm. An even simpler method is to use the standard k-means algorithm separately for each class label to obtain the clusters $\{\mathcal{C}_i\}$.

3.4 Analysis

This section provides a convergence analysis of the proposed algorithm. Before presenting the results, we introduce the notation:

$$\mathbf{w}^* = \min_{\mathbf{w}} P(\mathbf{w}),$$

which implies $\mathbf{w}^*$ is the optimal solution. Given this notation, we begin our analysis with a technical lemma.

Lemma 1. *Suppose $P(\mathbf{w})$ is H-strongly convex and $(1/\gamma)$-smooth over $\mathbb{R}^d$. If $\eta_t \in (0, \gamma]$, then the proposed algorithm satisfies the following inequality for any $t \geq 1$,*

$$\mathbb{E}\left[P(\mathbf{w}_{t+1}) - P(\mathbf{w}^*) \right] \leq \frac{1}{2\eta_t} \mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] - \frac{H}{2} \mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \eta_t \mathbb{E} V_t,$$

where

$$V_t = \mathbb{V}\Big(\frac{1}{n}\sum_{i=1}^{k}\frac{n_i^t}{b_i^t}\sum_{s\in\mathcal{B}_i^t}\nabla\phi_s(\mathbf{w}_t)\Big).$$

Proof. To simplify the analysis, we denote $\mathbf{g}_t = \frac{1}{n}\sum_{i=1}^{k}\frac{n_i^t}{b_i^t}\sum_{s\in\mathcal{B}_i^t}\nabla\phi_s(\mathbf{w}_t)$, and

$$\delta_t = \Big\langle \frac{1}{n}\sum_{i=1}^{k}\frac{n_i^t}{b_i^t}\sum_{s\in\mathcal{B}_i^t}\nabla\phi_s(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \Big\rangle - \Big[\frac{1}{n}\sum_{i=1}^{k}\frac{n_i^t}{b_i^t}\sum_{s\in\mathcal{B}_i^t}(\phi_s(\mathbf{w}_t) - \phi_s(\mathbf{w}^*)) + \frac{H}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2\Big].$$

Given these notations, we can first derive

$$\|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_t - \eta_t\mathbf{g}_t - \mathbf{w}^*\|^2 = 2\Big\langle \eta_t\mathbf{g}_t, \mathbf{w}_t - \mathbf{w}^* \Big\rangle - \|\eta_t\mathbf{g}_t\|^2$$

$$= 2\eta_t\delta_t + 2\eta_t\Big[\frac{1}{n}\sum_{i=1}^{k}\frac{n_i^t}{b_i^t}\sum_{s\in\mathcal{B}_i^t}(\phi_s(\mathbf{w}_t) - \phi_s(\mathbf{w}^*)) + \frac{H}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2\Big] - \|\eta_t\mathbf{g}_t\|^2.$$

Taking expectation of the above equality, and using the fact

$$\mathbb{E}\delta_t = \Big\langle \nabla P(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \Big\rangle - \Big[P(\mathbf{w}_t) - P(\mathbf{w}^*) + \frac{H}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2\Big] \geq 0,$$

we can obtain

$$\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] \geq 2\eta_t\mathbb{E}\Big[P(\mathbf{w}_t) - P(\mathbf{w}^*) + \frac{H}{2}\|\mathbf{w}_t - \mathbf{w}^*\|^2\Big] - \eta_t^2\mathbb{E}\|\mathbf{g}_t\|^2.$$

In addition, using the fact that P is $(1/\gamma)$-smooth and $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\mathbf{g}_t$, we can derive

$$\mathbb{E}P(\mathbf{w}_{t+1}) \leq \mathbb{E}P(\mathbf{w}_t) - \eta_t\mathbb{E}\|\nabla P(\mathbf{w}_t)\|^2 + \frac{\eta_t^2}{2\gamma}\mathbb{E}\|\mathbf{g}_t\|^2.$$

Combining the above two inequalities, we can get

$$\mathbb{E}[P(\mathbf{w}_{t+1}) - P(\mathbf{w}^*)]$$
$$\leq \frac{1}{2\eta_t}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] - \frac{H}{2}\mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 + (\frac{\eta_t}{2} + \frac{\eta_t^2}{2\gamma})\mathbb{E}\|\mathbf{g}_t\|^2 - \eta_t\mathbb{E}\|\nabla P(\mathbf{w}_t)\|^2.$$

Combing the above inequality with the facts $\mathbb{E}\|\mathbf{g}_t\|^2 = \mathbb{V}(\mathbf{g}_t) + (\|\mathbb{E}\mathbf{g}_t\|)^2 = \mathbb{V}(\mathbf{g}_t) + \|\nabla P(\mathbf{w}_t)\|^2$, and $\eta_t \in (0, \gamma]$ will conclude the the proof of this lemma. $\square$

Given the above lemma, we will prove a convergence result for the proposed algorithm, when $P(\mathbf{w})$ is H-strongly convex and $(1/\gamma)$-smooth.

Theorem 1. *Suppose $P(\mathbf{w})$ is H-strongly convex and $(1/\gamma)$-smooth. If we set $\eta_t = 1/(a + Ht)$ where $a \geq 1/\gamma - H$, then the proposed algorithm will satisfy the following inequality for all T,*

$$\inf_{t\in[T]} \mathbb{E}P(\mathbf{w}_{t+1}) - P(\mathbf{w}^*) \leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}P(\mathbf{w}_{t+1}) - P(\mathbf{w}^*) \leq \frac{1}{T}\Big[\frac{a}{2}\|\mathbf{w}^*\|^2 + \mathbb{E}\sum_{t=1}^{T}\frac{V_t}{a + Ht}\Big].$$

where $V_t = \mathbb{V}\Big(\frac{1}{n}\sum_{i=1}^{k}\frac{n_i^t}{b_i^t}\sum_{s\in\mathcal{B}_i^t}\nabla\phi_s(\mathbf{w}_t)\Big).$

Proof. Firstly, it is easy to verify $\eta_t \in (0, \gamma]$, $\forall t \geq 1$. Because $P(\mathbf{w})$ and η_t satisfy the assumptions in Lemma 1, we have

$$\mathbb{E}\Big[P(\mathbf{w}_{t+1}) - P(\mathbf{w}^*)\Big] \leq \frac{1}{2\eta_t}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] - \frac{H}{2}\mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \eta_t\mathbb{E}V_t,$$

Summing the above inequality over $t = 1, 2, \ldots, T$, and using $\eta_t = 1/(a + Ht)$, we get

$$\sum_{t=1}^{T}\mathbb{E}P(\mathbf{w}_{t+1}) - \sum_{t=1}^{T}P(\mathbf{w}^*)$$

$$\leq \sum_{t=1}^{T}\frac{a + Ht}{2}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2] - \frac{H}{2}\sum_{t=1}^{T}\mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \mathbb{E}\sum_{t=1}^{T}\frac{V_t}{a + Ht}$$

$$= \frac{a}{2}\mathbb{E}\|\mathbf{w}_1 - \mathbf{w}^*\|^2 - \frac{a + HT}{2}\mathbb{E}\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 + \mathbb{E}\sum_{t=1}^{T}\frac{V_t}{a + Ht} \leq \frac{a}{2}\|\mathbf{w}^*\|^2 + \mathbb{E}\sum_{t=1}^{T}\frac{V_t}{a + Ht}.$$

Dividing the above inequality with T will conclude the theorem. $\square$

If $V_t = 0$ for all $t \in [T]$, then the above theorem will give a $O(1/T)$ convergence bound, which is the same as the convergence bound of gradient descent for a convex (0-strongly convex) and smooth objective function. However, if the objective function is H-strongly convex with $H > 0$, the convergence bound of gradient descent is $O(c^T)$ with $c \in (0, 1)$, which is significantly better than the bound in this theorem. Therefore in addition to the above theorem, we also prove a linear convergence bound when $V_t = 0$, $\forall t \in [T]$. First we prove a technical lemma as follows.

Lemma 2. *Suppose $P(\mathbf{w})$ is H-strongly convex and $(1/\gamma)$-smooth over $\mathbb{R}^d$. Then we have the following inequality for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$,*

$$\langle \nabla P(\mathbf{u}) - \nabla P(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle \geq \frac{H/\gamma}{H + 1/\gamma}\|\mathbf{u} - \mathbf{v}\|^2 + \frac{1}{H + 1/\gamma}\|\nabla P(\mathbf{u}) - \nabla P(\mathbf{v})\|^2.$$

Proof. To simplify the notations, we define

$$f(\mathbf{w}) = P(\mathbf{w}) - \frac{H}{2}\|\mathbf{w}\|^2.$$

Firstly, $f(\mathbf{w})$ is convex (0-strongly convex), since the following inequality holds for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$,

$$f(\mathbf{u}) - [f(\mathbf{v}) + \langle \nabla f(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle] = P(\mathbf{u}) - \frac{H}{2}\|\mathbf{u}\|^2 - [P(\mathbf{v}) - \frac{H}{2}\|\mathbf{v}\|^2 + \langle \nabla P(\mathbf{v}) - H\mathbf{v}, \mathbf{u} - \mathbf{v}\rangle]$$

$$= P(\mathbf{u}) - [P(\mathbf{v}) + \langle \nabla P(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{H}{2}\|\mathbf{u} - \mathbf{v}\|^2] \geq 0,$$

where the final inequality used the H-strongly convexity of $P(\mathbf{w})$.

Secondly, $f(\mathbf{w})$ is $(1/\gamma - H)$-smooth, since the following inequality holds for any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$,

$$f(\mathbf{u}) - [f(\mathbf{v}) + \langle \nabla f(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{1/\gamma - H}{2}\|\mathbf{u} - \mathbf{v}\|^2]$$

$$= P(\mathbf{u}) - \frac{H}{2}\|\mathbf{u}\|^2 - [P(\mathbf{v}) - \frac{H}{2}\|\mathbf{v}\|^2 + \langle \nabla P(\mathbf{v}) - H\mathbf{v}, \mathbf{u} - \mathbf{v}\rangle + \frac{1/\gamma - H}{2}\|\mathbf{u} - \mathbf{v}\|^2]$$

$$= P(\mathbf{u}) - [P(\mathbf{v}) + \langle \nabla P(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \frac{1}{2\gamma}\|\mathbf{u} - \mathbf{v}\|^2] \leq 0,$$

8

where the final inequality used the fact $P(\mathbf{w})$ is $(1/\gamma)$-smooth.

Because $f(\mathbf{w})$ is convex and $(1/\gamma - H)$-smooth, according to the Property 1, the co-coercivity of $f(\mathbf{w})$ with parameter $\frac{1}{1/\gamma - H}$ gives

$$\langle (\nabla P(\mathbf{u}) - H\mathbf{u}) - (\nabla P(\mathbf{v}) - H\mathbf{v}), \mathbf{u} - \mathbf{v} \rangle \geq \frac{1}{1/\gamma - H} \|(\nabla P(\mathbf{u}) - H\mathbf{u}) - (\nabla P(\mathbf{v}) - H\mathbf{v})\|^2.$$

Re-arranging the above inequality conclude the proof of this lemma. □

Given the above lemma, we can prove another bound for the proposed algorithm as follows, when $P(\mathbf{w})$ is H-strongly convex and $(1/\gamma)$-smooth.

Theorem 2. *Suppose $P(\mathbf{w})$ is H-strongly convex and $(1/\gamma)$-smooth. If we set $\eta_t = \eta \in (0, \frac{2}{H+1/\gamma}]$, then the proposed algorithm will satisfy the following inequality for all T,*

$$\mathbb{E}P(\mathbf{w}_{T+1}) - P(\mathbf{w}^*) \leq \frac{\alpha(\eta)^T}{2\gamma} \|\mathbf{w}^*\|^2 + \frac{\eta^2}{2\gamma} \sum_{t=1}^{T} \alpha(\eta)^{T-t} \mathbb{E}V_t,$$

where $\alpha(\eta) = 1 - \frac{2\eta H/\gamma}{H+1/\gamma} \in [(\frac{H-1/\gamma}{H+1/\gamma})^2, 1)$ and $V_t = \mathbb{V}\left(\frac{1}{n}\sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \nabla \phi_s(\mathbf{w}_t)\right).$

Proof. Using the fact $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}_t$ where $\mathbf{g}_t = \frac{1}{n}\sum_{i=1}^{k} \frac{n_i^t}{b_i^t} \sum_{s \in \mathcal{B}_i^t} \nabla \phi_s(\mathbf{w}_t)$, we have

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \|\mathbf{w}_t - \eta \mathbf{g}_t - \mathbf{w}^*\|^2 = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}^* \rangle + \eta^2 \|\mathbf{g}_t\|^2.$$

Taking expectation of the above inequality, using $\mathbb{E}\|\mathbf{g}_t\|^2 = \mathbb{V}(\mathbf{g}_t) + \|\mathbb{E}\mathbf{g}_t\|^2 = \mathbb{V}(\mathbf{g}_t) + \|\nabla P(\mathbf{w}_t)\|^2$ and using Lemma 2, will derive

$$\mathbb{E}\|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 = \mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 - 2\eta\mathbb{E}\langle \nabla P(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}^* \rangle + \eta^2 \mathbb{E}[V_t + \|\nabla P(\mathbf{w}_t)\|^2]$$

$$\leq (1 - \frac{2\eta H/\gamma}{H+1/\gamma})\mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \eta(\eta - \frac{2}{H+1/\gamma})\mathbb{E}\|\nabla P(\mathbf{w}_t)\|^2 + \eta^2 \mathbb{E}V_t$$

$$\leq \alpha(\eta)\mathbb{E}\|\mathbf{w}_t - \mathbf{w}^*\|^2 + \eta^2 \mathbb{E}V_t,$$

where the final inequality used $\eta \in (0, \frac{2}{H+1/\gamma}]$. Using the above inequality iteratively, we can derive

$$\mathbb{E}\|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 \leq \alpha(\eta)\mathbb{E}\|\mathbf{w}_T - \mathbf{w}^*\|^2 + \eta^2 \mathbb{E}V_T \leq \alpha(\eta)\left[\alpha(\eta)\mathbb{E}\|\mathbf{w}_{T-1} - \mathbf{w}^*\|^2 + \eta^2 \mathbb{E}V_{T-1}\right] + \eta^2 \mathbb{E}V_T$$

$$\leq \quad \ldots \leq \alpha(\eta)^T \mathbb{E}\|\mathbf{w}_1 - \mathbf{w}^*\|^2 + \alpha(\eta)^{T-1}\eta^2 \mathbb{E}V_1 + \ldots + \alpha(\eta)\eta^2 \mathbb{E}V_{T-1} + \eta^2 \mathbb{E}V_T$$

$$= \quad \alpha(\eta)^T \mathbb{E}\|\mathbf{w}_1 - \mathbf{w}^*\|^2 + \eta^2 \sum_{t=1}^{T} \alpha(\eta)^{T-t} \mathbb{E}V_t.$$

In addition using the facts $P(\mathbf{w})$ is $(1/\gamma)$-smooth and $\nabla P(\mathbf{w}^*) = 0$, we get

$$P(\mathbf{w}) - P(\mathbf{w}^*) \leq \langle \nabla P(\mathbf{w}^*), \mathbf{w} - \mathbf{w}^* \rangle + \frac{1}{2\gamma}\|\mathbf{w} - \mathbf{w}^*\|^2 = \frac{1}{2\gamma}\|\mathbf{w} - \mathbf{w}^*\|^2.$$

Combining the above two inequalities concludes the proof. □

If $V_t = 0$ for all $t \in [T]$, then the above theorem will given a convergence bound of $O(\alpha(\eta)^T)$ with $\alpha(\eta) \in [(\frac{H-1/\gamma}{H+1/\gamma})^2, 1)$, which is the same with convergence bound of gradient descent for a H-strongly ($H > 0$) convex and smooth objective function.

4   Experiments

In this section, we evaluate the empirical performance of the proposed algorithm by comparing it to SGD with uniform sampling.

4.1   Experimental Setup

To extensively examine the performance, we test all the algorithms on a number of benchmark datasets from web machine learning repositories. Table 1 shows the details of the datasets used in our experiments. All of these datasets can be downloaded from the LIBSVM website [1]. These datasets were chosen fairly randomly in order to cover various sizes of datasets.

Table 1: Details of the datasets in our experiments.

Dataset	# class	Training size	Testing size	# features	minibatch size
covtype.binary	2	523,124	57,888	54	10
letter	26	15,000	5,000	16	26
mnist	10	60,000	10,000	780	10
pendigits	10	7,494	3,498	16	13
usps	10	7,291	2,007	256	48

To make a fair comparison, all algorithms adopted the same setup in our experiments. In particular, we performed L2-regularized multiclass logistic regression (convex optimization) to tackle these classification tasks. The regularization parameters of multiclass logistic regression is set as 10^{-5}, 10^{-4}, 10^{-4}, 10^{-3} and 10^{-3} for *covtype.binay*, *letter*, *mnist*, *pendigits*, and *usps* respectively. For SGD with uniform sampling and stratified sampling, the step size for the t-th round is set as $\eta_t = 1/(\lambda t)$ for all the datasets. The minibatch sizes, given in Table!1, were chosen as the same value for SGD and SGD-ss on every dataset. For SGD-ss, we used a stratified sampling strategy by solving the optimization problems (3) and (4). Note that instead of using (4), the clusters $\{\mathcal{C}_i\}$ can also be obtained via the simpler k-means method without performance degradation.

All the experiments were conducted by fixing 5 different random seeds for each datasets. All the results were reported by averaging over these 5 runs. We evaluated the algorithms' performance by measuring *primal objective value* on training dataset, i.e., $P(\mathbf{w}_t)$. In addition, to examine the performance of resulting classifiers on test datasets, we also evaluated the *test error rate*. Finally, we also compared the actual variances of stochastic gradient estimators of the two algorithms to check whether the proposed algorithm can effectively reduce the variance.

4.2   Performance Evaluation

Figure 1 summarizes the experimental results on the dataset *covtype.binary*. First, Figure 1 (a) shows the primal objective values of SGD-ss in comparison to that of SGD with uniform sampling. We can observe that SGD-ss converges faster and is much more stable than SGD. Because these two algorithm adopted the same minibatch size and learning rates, this observation clearly implies that the proposed stratified sampling strategy is more effective to reduce the variance of the stochastic

---

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/

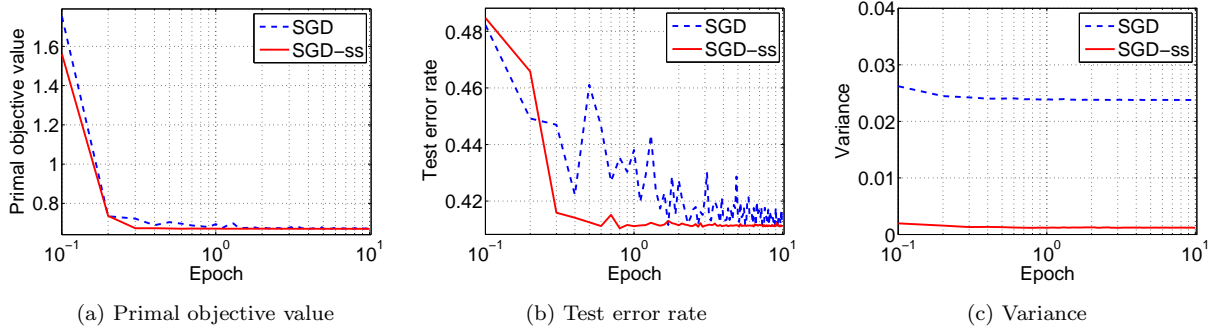(a) Primal objective value   (b) Test error rate   (c) Variance

Figure 1: Multiclass logistic regression (convex) on **covtype.binary**. Epoch for the horizontal axis is the number of iterations times the minibatch size divided by the training data size.

gradient estimator than uniform sampling. Second, Figure 1 (b) provides test error rates of the two algorithms, where we observe that SGD-ss achieves significantly smaller and stable test error rates than that of SGD. This shows that the proposed stratified sampling approach is effective in improving the testing performance and reducing its variance. Third, according to Figure 1 (c), we observe that the variance for SGD-ss is significantly smaller than that of SGD with uniform sampling. This again demonstrates the effectiveness of the proposed sampling strategy to reduce the variance of the unbiased stochastic gradient estimators.

Figure 2 shows more L2-regularized logistic regression results in terms of primal objective value, test error rate and variance of the stochastic gradients. Overall, SGD-ss is clearly superior to SGD with uniform sampling, which demonstrates the effectiveness of the proposed sampling strategy.

5   Conclusion

This paper studies stratified sampling to reduce the variance for Stochastic Gradient Descent method. We not only provided a dynamic stratified sampling strategy but also provided a fixed stratified sampling strategy. We showed that the convergence rate for the training process can be significantly reduced by the proposed strategy. We have also conducted an extensive set of experiments to compare the proposed method with the traditional uniform sampling strategy. Promising empirical results validated the effectiveness of our technique.
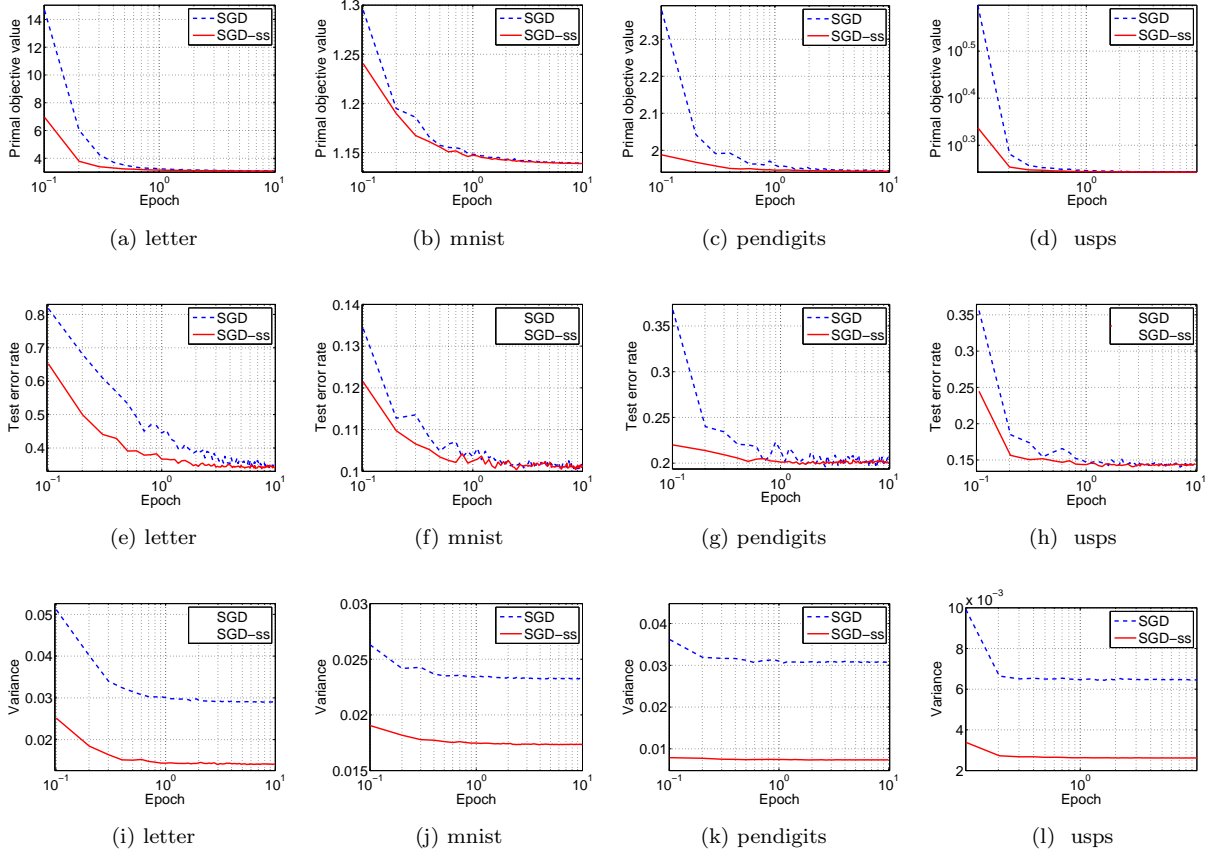
11

Figure 2: More results on multiclass logistic regression (convex) on **letter**, **mnist**, **pendigits**, and **usps**. Epoch for the horizontal axis is the number of iterations times the minibatch size divided by the training data size. The first row summarized the primal objective value of the algorithms on these four datasets. The second row summarized the test error rates of the algorithms on these four datasets. The final row summarized the variances of the stochastic gradient estimators of the algorithms on these four datasets.

References

[1] Jonathan Borwein and Adrian Lewis. *Convex analysis and nonlinear optimization: theory and examples*, volume 3. Springer, 2006.

[2] John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, 10:2899–2934, 2009.

[3] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[4] Harold J Kushner and George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer, 2003.

[5] Mehrdad Mahdavi, Tianbao Yang, Rong Jin, Shenghuo Zhu, and Jinfeng Yi. Stochastic gradient descent with only one projection. In *NIPS*, pages 503–511, 2012.

[6] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

[7] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pages 807–814, 2007.

[8] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Math. Program.*, 127(1):3–30, 2011.

[9] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 71–79, 2013.

[10] Chong Wang, Xi Chen, Alex J. Smola, and Eric P. Xing. Variance reduction for stochastic gradient optimization. In *NIPS*, pages 181–189, 2013.

[11] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.

[12] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling. *CoRR*, abs/1401.2753, 2014.