Stochastic Orthant-Wise Limited-Memory Quasi-Newton Methods

Jianqiao Wangni
Tsinghua University
Beijing, China
zjnqha@gmail.com

Abstract

The $\ell_1$-regularized sparse model has been popular in machine learning society. The orthant-wise quasi-Newton (OWL-QN) method is a representative fast algorithm for training the model. However, the proof of the convergence has been pointed out to be incorrect by multiple sources, and up until now, its convergence has not been proved at all. In this paper, we propose a stochastic OWL-QN method for solving $\ell_1$-regularized problems, both with convex and non-convex loss functions. We address technical difficulties that have existed many years. We propose three alignment steps which are generalized from the original OWL-QN algorithm, to encourage the parameter update be orthant-wise. We adopt several practical features from recent stochastic variants of L-BFGS and the variance reduction method for subsampled gradients. To the best of our knowledge, this is the first orthant-wise algorithms with comparable theoretical convergence rate with stochastic first order algorithms. We prove a linear convergence rate for our algorithm under strong convexity, and experimentally demonstrate that our algorithm achieves state-of-art performance on $\ell_1$-regularized logistic regression and convolutional neural networks.

Introduction

Originating from Newton’s method, the second order algorithms are widely-used for scientific computing and optimization. They capture more useful curvature information than first-order algorithms. They are theoretically guaranteed to converge faster, especially when the data dimension is relatively small. In high dimensions, BFGS, a representative quasi-Newton method, is able to circumvent the inversion of the Hessian matrix by logging optimizer trajectory. From an application perspective, the machine learning community has been favouring $\ell_1$-regularized sparse learning. Therefore many algorithms were proposed to speed up the training process. Due to the non-smoothness of the $\ell_1$ regularizer, the sparse model can not directly benefit from the fast convergence of second order algorithms (Tseng and Yun 2009)(Rodomanov and Kropotov 2016). The Orthant-Wise Limited memory Quasi-Newton (OWL-QN)((Andrew and Gao 2007)) method solved this problem differently by generalizing the L-BFGS and adopting three gradient eq-narrayment steps, which made the parameters remain in the same orthant after updates, and achieved state-of-art performance in $\ell_1$-regularized logistic regression (Yu et al. 2010) and sparse inverse covariance estimation (Oztoprak et al. 2012). Despite its great performance in practice, the convergence proof of OWL-QN is pointed out to be incorrect by (Yu et al. 2010)(Yuan et al. 2010)(Schmidt et al. 2009)(Byrd et al. 2012). (Gong and Ye 2015) pointed out the problems in the existing convergence analysis and proposed a modified OWL-QN (mOWL-QN) algorithm with theoretical convergence.

The success of stochastic first order algorithms (Bottou 2010)(Defazio, Bach, and Lacoste-Julien 2014)(Shalev-Shwartz and Zhang 2013) has led to some progresses toward stochastic second order algorithms. As opposed to batch algorithms, stochastic ones update parameters by processing data mini-batches with higher frequency and reduce the loss function in expectation. (Schraudolph et al. 2007) proposed using subsampled gradients. The regularized BFGS method (Mokhtari and Ribeiro 2014) further modified the algorithm by adding an identity matrix regularization to the Hessian matrix to guarantee the positive-definiteness. (Moritz, Nishihara, and Jordan 2016) proposed a linearly convergent stochastic quasi-Newton method based on SVRG (Johnson and Zhang 2013), which adds a full gradient on a stale point to the subsampled gradients, and balances the gradient expectation by subtracting subsampled stale gradients evaluated on the same subset. (Gower, Goldfarb, and Richtárík 2016) additionally adopt block BFGS update from (Schnabel 1983) and the consequent convergence rate is faster.

In this paper, we proposed a novel stochastic quasi-Newton methods for $\ell_1$ regularized problems. The algorithm is very easy to implement based on L-BFGS, adapts different data by multiple strategies, and achieves a state-of-art convergence rate with respect to dataset passes. The improvement is significant since the memory bandwidth is frequently the bottleneck nowadays.

Preliminaries

We study the composite objective function $P(x)$ on $\mathbb{R}^D$ as

\[ P(x) = F(x) + R(x), \]
where $F$ is the average of $\mu$-strongly convex loss functions and $R$ is the $\ell_1$ regularization,

$$F(x) = \frac{1}{N} \sum_{i=1}^{N} f_n(x), \quad R(x) = \lambda \|x\|_1.$$  

We will revisit some algorithms built on the assumption.

**Assumption 1.** Each loss function $f_n : \mathbb{R}^D \to \mathbb{R}$ is twice differentiable, and has $L$-Lipschitz continuous gradient, such that $\nabla^2 f_n(x) \preceq L I_D$, where $I_D \in \mathbb{R}^{D \times D}$ is an identity matrix.

The proximal gradient optimizer for the problem in Eq. (1) sequentially finds a minimum on a quadratic expansion at the point $x_{m,k-1} \in \mathbb{R}^D$ in the $k$-th step update,

$$x_k = \arg \min_{x \in \mathbb{R}^D} \nabla F(x_{m,k-1})^T x + \frac{1}{2\eta} \|x - x_{m,k-1}\|^2 + R(x),$$

where $\eta$ is the stepsize. The proximal gradient method can be written compactly as

$$x_k = \text{prox}_{\eta R}(x_{m,k-1} - \eta \nabla F(x_{m,k-1})),$$  

where $\text{prox}_{\eta R}(y) = \arg \min_{x \in \mathbb{R}^D} \frac{1}{2\eta} \|x - y\|^2 + \eta R(x)$. For our case $R(x) = \lambda \|x\|_1$, we have $\text{prox}_{\eta R}(x) = \sigma(x) \odot \max(\|x\| - \lambda, 0)$, where $\sigma$ is the element-wise product.

**Orhant-Wise Limited-Memory Quasi-Newton Method (OWL-QN).** We denote the sign function $\sigma()$ as follows: $\sigma(x)_i = 1$, if $x_i > 0$; $\sigma(x)_i = 1$, if $x_i < 0$ and $\sigma(x)_i = 0$, otherwise. The gradient eqnarrayment function $\pi : \mathbb{R}^D \to \mathbb{R}^D$ (Andrew and Gao 2007) was element-wisely defined as

$$\pi_i(x; y) = \begin{cases} x_i, & \text{if } \sigma(x)_i = \sigma(y)_i, \\
0, & \text{otherwise}, \end{cases}$$  

where $y \in \mathbb{R}^D$ is referred as a reference point, and $y_i$ is the $i$-th element of $y$. For notation simplicity, we define an element-wise operator $\psi : \mathbb{R}^D \to \mathbb{R}^D$ as

$$\psi_i(v; x; \lambda) = \begin{cases} v + \lambda, & \text{if } x_i > 0, \\
v - \lambda, & \text{if } x_i < 0, \\
v_i + \lambda, & \text{if } x_i = 0, v_i + \lambda < 0, \\
v_i - \lambda, & \text{if } x_i = 0, v_i - \lambda > 0, \\
0, & \text{otherwise}. \end{cases}$$

OWL-QN eqnarrayments the pseudo-gradient $\nabla F(x)$ based on current gradient $\nabla F(x)$ and weights $x$,

$$\nabla F(x) = \psi(\nabla F(x); x; \lambda).$$

Then, based on an approximated quadratic expansion at point $x_{m,k-1}$ of Eq. (1), the OWL-QN finds the direction $d_k$ by minimizing the expansion function as

$$d_k = \arg \min_{d \in \mathbb{R}^D} F(x_{m,k-1}) + \nabla F(x_{m,k-1})^T d + \frac{1}{2} d^T B_k d$$

$$= -H_k \nabla F(x_{m,k-1}),$$

where $B_k$ is the approximated Hessian at $x = x_{m,k-1}$ and $H_k = B_k^{-1}$. The inverse Hessian approximation $H_k$ is constructed from the curvature pairs $(s_j, y_j)$, where $s_j = x_j - x_{j-1}$ and $y_j = \nabla F(x_j) - \nabla F(x_{j-1})$ for $j \leq k$. Denoting $\rho_j = 1/s_j^T y_j$, initializing $H_k^0 = (s_j^T y_j/\|y_j\|^2) I_D$ and setting $H_k = H_k^0$, then OWL-QN recursively compute

$$H_{k+1} = (I - \rho_j s_j y_j^T H_k^0) H_k^{-1} (I - \rho_j s_j y_j^T) + \rho_j s_j y_j^T$$

for $k-M+1 \leq j \leq k$, as standard L-BFGS. The OWL-QN applies $H_k$ to obtain better direction $d_k$, then takes the second eqnarrayment on the direction $d_k$ to encourage the direction $p_k$ be orthant-wise,

$$p_k = \pi(-d_k; y_k), \quad \text{where } y_k = \nabla F(x_k).$$

After this, OWL-QN makes the third eqnarrayment, which explicitly restrict the updated weights $x_{m,k-1} - \alpha_k p_k$ to be in the same orthant with $x_{m,k-1}$, as

$$x_k = \pi(x_{m,k-1} - \alpha_k p_k; x_{m,k-1})$$

where the best stepsize $\alpha_k$ is obtained by line-search to satisfy the Wolfe condition.

**Stochastic Block L-BFGS.** We define a subsampled loss function $f_n$ evaluated on a subset in the $k$-th step, 

$$f_k(x) = \frac{1}{|S_k|} \sum_{n \in S_k} f_n(x), \quad \text{where } S_k \subset \{1, \ldots, N\}.$$  

The stochastic block BFGS method (Gower, Goldfarb, and Richtárik 2016), applies the weighted projection on $H_{k-1}$ to obtain $H_k$,

$$H_k = \arg \min_{H \in \mathbb{R}^{D \times D}} ||(H - H_{k-1}) \nabla^2 f_k(x_{m,k-1})||_F^2,$$

s.t. $H \nabla^2 f_k(x_{m,k-1}) D_k = D_k; H = H^T$,  

where $|| \cdot ||_F$ is the Frobenius norm. The matrix $D_k$ is random sampled and hereby to introduce stochasticity into the algorithm. The update is

$$H_k = D_k \Delta_k D_k^T + (I - D_k \Delta_k Y_k^T) H_{k-1} (I - Y_k \Delta_k D_k),$$

where $\Delta_k = (D_k^T Y_k)^{-1}$ and $Y_k = \nabla^2 f(x_k) D_k$.

**Stochastic OWL-QN.** The difficulty of generalizing stochastic L-BFGS methods to non-smooth objective function mainly rises from the two points: controlling the variance of the quasi-Newton directions based on an inaccurate first order gradients, and preserve the convergence property when the stochastic gradients approach the true gradients. The former issue is shared by all previous stochastic BFGS literatures (Byrd et al. 2016; Moritz, Nishihara, and Jordan 2016; Gower, Goldfarb, and Richtárík 2016; Wang et al. 2016), even with smooth functions. In these papers, the variance of stochastic second order gradients is loosely bounded. Therefore they failed to obtain a better convergence rate than SVRG and its variants (Reddi et al. 2016; [Allen-Zhu and Hazan 2016]; [Allen-Zhu and Yuan 2016]; [Reddi et al. 2016]). Although this drawback is critical, it is not in the scope of this paper. The latter issue is also troublesome, since the convergence proof of the original OWL-QN is still not accomplished. The reason for such a difficulty was discussed in
(Gong and Ye 2015), and their solution was to add a gradient descent step to ensure convergence. As for the pure second order methods, orthant-wise updates have not been proved convergence at all. In this paper, we will give an OWL-QN type algorithm that empirically outperforms the proximal-SVRG, with comparable theoretical convergence rate.

We denote the stochastic first-order gradient in the \( k \)-th step as \( v_k \), and the calculated second-order one as \( q_k \), then
\[
E[P(x_k)] = E[P(x_{m,k-1} - \eta q_k)] 
\]
\[
\leq F(x_{m,k-1}) - \eta \nabla F(x_{m,k-1})^T v_k + \frac{\eta^2}{2} E[||q_k||^2] + R(x_k) \tag{17} 
\]

The key to establish an optimizer that converges fast both in theory and application is to make \( q_k \) as an unbiased estimate of \( \nabla P(x_{m,k-1}) \) and control the variance \( ||q_k||^2 \). However, as the original OWL-QN methods suggest that
\[
p_k = \pi(H_k(v_k + \partial R(x_{m,k-1})), v_k + \partial R(x_{m,k-1})) \tag{18} 
\]
\[
q_k = \pi(x_{m,k-1} - \eta q_k, x_{m,k-1}) \tag{19} 
\]
as the algorithm approaches a stationary point, and the \( v_k \) approaches the true gradient, as \( v_k \rightarrow \nabla F(x_{m,k-1}) \), and under an assumption that \( 0 < \gamma \leq H_k \), the key proposition in the OWL-QN paper, as
\[
v_k^T q_k \geq v_k^T p_k \geq v_k^T d_k = v_k^T H_k v_k \geq \gamma ||d_k||^2, \tag{20} 
\]
which is important for upper bounding the second negative term in Eq. (16), does not hold. The analysis encourages us to modify the gradient eqnarraymant operation to assure theoretical convergence.

The Proposed Algorithm
Here we introduce our Stochastic Orthant-Wise Limited-Memory Quasi-Newton algorithm (SOWL-QN). The main improvement over OWL-QN is using a subsampled gradient to replace the full gradient. We define a subsampled loss function \( f_k \) evaluated on a subset in the \( k \)-th step, we first evaluate the subsampled gradient \( \nabla f_k \) as Eq. (13) and make the first eqnarraymant \( \nabla f_k \),
\[
\hat{f}_k(x_{m,k-1}) = \psi(\nabla f_k(x_{m,k-1}), x_{m,k-1}, \lambda), \tag{21} 
\]
where \( \lambda \) is the regularization parameter in Eq. (1). We then calculate the variance reduced version \( v_k \) of \( \nabla f_k \) out of the same considerations as \( H_k \),
\[
v_k = \nabla f_k(x_{m,k-1}) - \nabla f_k(\tilde{x}) + \nabla F(\tilde{x}), \tag{22} 
\]
where \( \tilde{x} \) is a stale point used in SVRG family algorithms. Although a more straightforward choice is to set \( v_k = \hat{f}_k(x_{m,k-1}) \), but this will introduce variance that is hard to control and compromise the convergence rate, since the subgradient \( \hat{f}_k \).

Next, we take the second eqnarraymant step, using the pseudo-gradient \( \hat{f}_k(x_{m,k-1}) \) as a reference point, which is independent of \( v_k \), and its sign is uniformly distributed once the data is normalized to have zero mean, therefore this will not introduce additional bias into the stochastic gradient,
\[
p_k = \pi(H_k v_k, \hat{f}_k(x_{m,k-1})). \tag{23} 
\]

The aforementioned calculation does not explicitly involves the gradient of \( R(x) \), excepts for eqnarraymant reference, this is due to that we are avoiding introducing additional variance. So, to make the overall objective function decreases over iterations, we have to make an additional modification that involves regularizations, in next steps. Here we propose to define a novel eqnarraymant operator \( \pi(\cdot) \), that \( \pi(x; y) = 0 \) if and only if \( \pi(x; y) = 0 \), and \( \pi(x; y) = x \) for otherwise. The difference is that we allow \( (H_k v_k) \) takes arbitrary value if \( -\lambda < \nabla f(x) < \lambda \) and \( (x_k) = 0 \). We also define a novel eqnarraymant operator \( \phi \),
\[
\phi(x; y; a) = \begin{cases} 
0, & \text{if } (\pi(x; y) = 0) \text{ or } (|x| < a) \\
\pi(x; y) - a & \text{sign}(x), & \text{otherwise}
\end{cases} \tag{24} 
\]

It is a generalization of \( \pi(\cdot) \) operator which reduces to the latter one when \( a \rightarrow 0 \), and it has additional suppression on the absolute value of large elements and zero restriction on small elements. In the convex case, we take a forward step on the obtained direction \( p_k \) scaled with stepsize \( \eta \), and take the third eqnarraymant as
\[
q_k = \phi(x_{m,k-1} - \eta p_k; x_{m,k-1}; \eta \lambda) - x_{m,k-1}. \tag{25} 
\]

Different from OWL-QN, we allow the update of each element of \( x_{m,k-1} \) from zero to any orthant validated, and this modification shows improvement in practice. We also empirically test the original operator, as \( x_k = \phi(x_{m,k-1} - \eta p_k; x_{m,k-1}; 0) \), the results are relatively worse than our proposed one in convex case, but comparable in non-convex case. We describe the whole framework as Algorithm 1 and the matrix vector multiplication update in Algorithm 2.

The update frequency of the stale point is controlled by a parameter \( m \), which also indicates the times of evaluating

---

**Algorithm 1 Stochastic OWL-QN**

**Input** \( x_0 \in \mathbb{R}^D \), stepsize \( \eta > 0 \), and stale points update frequency \( m \).

**Initialize** \( k = 0 \), \( H_0 = I \), \( t = 0 \).

**repeat**

Compute a full gradient \( \nabla F(\tilde{x}_t) \) on the stale point.

**repeat**

Sample a random mini-batch \( S_k \subset \{1, \cdots, N\} \), \( \Phi \) the eqnarrayment \( \hat{f}_k(x_{m,k-1}) \). \( \Phi \)

Compute the subsampled gradient \( v_k \). \( \Phi \)

Sample a matrix \( D_k \in \mathbb{R}^{D \times r} \) that \( \text{rank}(D_k) = r \), compute \( Y_k = \nabla^2 f(x_k) D_k \) and \( D_k^T Y_k \) and compute \( \Delta_k = (D_k^T Y_k)^{-1} \) by Cholesky factorization. \( \Phi \)

Compute the direction \( d_k = H_k v_k \) via Algorithm 2, \( \Phi \)

eqnarrayment the direction \( p_k = \pi(d_k; \hat{f}_k(x_k)) \). \( \Phi \)

eqnarrayment the update \( x_k = \phi(x_{m,k-1} - \eta p_k; x_{m,k-1}; \eta \lambda) \). \( \Phi \)

Set \( k = k + 1 \).

until \( k \% m = 0 \)

**Option I:** Set \( \tilde{x}_{t+1} = \frac{1}{m} \sum_{j=k-m}^{k-1} x_j \).

**Option II:** Set \( \tilde{x}_{t+1} = x_j \), where \( j \) is selected uniformly at random from \( [m] = \{k-m, k-m+1, \cdots, k\} \).

Set \( t = t + 1 \).

until Reaching maximum outer iterations \( t = T \), or converging.

---

**Algorithm 2 Stochastic OWL-QN**

**Input** \( x_0 \in \mathbb{R}^D \), stepsize \( \eta > 0 \), and stale points update frequency \( m \).

**Initialize** \( k = 0 \), \( H_0 = I \), \( t = 0 \).

**repeat**

Compute a full gradient \( \nabla F(\tilde{x}_t) \) on the stale point.

**repeat**

Sample a random mini-batch \( S_k \subset \{1, \cdots, N\} \), \( \Phi \) the eqnarrayment \( \hat{f}_k(x_{m,k-1}) \). \( \Phi \)

Compute the subsampled gradient \( v_k \). \( \Phi \)

Sample a matrix \( D_k \in \mathbb{R}^{D \times r} \) that \( \text{rank}(D_k) = r \), compute \( Y_k = \nabla^2 f(x_k) D_k \) and \( D_k^T Y_k \) and compute \( \Delta_k = (D_k^T Y_k)^{-1} \) by Cholesky factorization. \( \Phi \)

Compute the direction \( d_k = H_k v_k \) via Algorithm 2, \( \Phi \)

eqnarrayment the direction \( p_k = \pi(d_k; \hat{f}_k(x_k)) \). \( \Phi \)

eqnarrayment the update \( x_k = \phi(x_{m,k-1} - \eta p_k; x_{m,k-1}; \eta \lambda) \). \( \Phi \)

Set \( k = k + 1 \).

until \( k \% m = 0 \)

**Option I:** Set \( \tilde{x}_{t+1} = \frac{1}{m} \sum_{j=k-m}^{k-1} x_j \).

**Option II:** Set \( \tilde{x}_{t+1} = x_j \), where \( j \) is selected uniformly at random from \( [m] = \{k-m, k-m+1, \cdots, k\} \).

Set \( t = t + 1 \).

until Reaching maximum outer iterations \( t = T \), or converging.
subsampling gradients with the same full gradient $\nabla F(x)$. Then we proceed to calculate the gradient $v_k$ by the inverse Hessian $H_k$. The inverse Hessian $H_k$ is updated by every $M$ iterations, and uses $M$ saved $(s_j, y_j)$ curvature pairs (L-BFGS) (Byrd et al. 2016) or $(D_j, Y_j, \Delta_j)$ curvature triples (BL-BFGS) (Gower, Goldfarb, and Richtárik 2016). Algorithm 2 consumes $M(r(4D + 2r) + \Delta_j)$ arithmetic operations in total. Following the setting in (Gower, Goldfarb, and Richtárik 2016) that $r \leq \sqrt{D}$, its cost is approximately $O(D^r\sqrt{r} \times M)$.

The calculation of $D_j^i Y_j$ and the Cholesky factorization results in an additional $O(r^2D)$ plus $O(rD)$ operations. The gradient equation only consumes $O(D)$ operations, which is negligible compared to the major costs. The matrix $D_j$ is randomly sampled and hereby to introduce randomness into the algorithm.

Algorithm 2 Block L-BFGS update

**Input** $v_k \in \mathbb{R}^D, D_j, Y_j \in \mathbb{R}^{D \times r} \text{ and } \Delta_j \in \mathbb{R}^{r \times r}$ from Algorithm 1. for $j \in \{k + 1 - M, \ldots, k\}$. Initialize $v' = v_k, j = k$.

Repeat

1. $v = \Delta_j D_j^T v', v' = v' - Y_j v', j = j - 1$, until $j = k - M + 1$.
2. $v = \Delta_j Y_j^T v', v' = v' + D_j(\alpha_j - \beta_j), j = j + 1$, until $j = k$.

**Convergence Analysis**

Before proceeding to the main theorem, we cite the following lemmas as important building blocks.

**Lemma 1.** Under the condition of Assumption 1, we define $x_\ast = \arg\min_x f(x)$, and define $v_k$ as Eq. (22), then

$$E[\|\nabla f_k(x) - \nabla f_k(x_\ast)\|^2] \leq 2L[P(x) - P(x_\ast)], \quad \text{(26)}$$

$$E[\|v_k\|^2] \leq 4L[P(x_{m,k-1}) - P(x_\ast) + P(\bar{x}) - P(x_\ast)], \quad \text{(27)}$$

**Lemma 2.** For any convex function $R$ on $\mathbb{R}^D$, and $x, y \in \mathbb{R}^D$, it holds $\|\rho \text{prox}_\eta(y) - \text{prox}_\eta(y)\| \leq \|x - y\|$, and $\eta$ is a positive constant. **Remark.** This is the nonexpansiveness of proximal mapping which can be found in (Rockafellar 2015) section 31.

**Lemma 3.** Suppose that Assumption 1 holds and each loss function is $\mu$-strongly convex. Then there exist constants $0 \leq \gamma < \Gamma$ such that $H_k$ satisfies $\gamma I \leq H_k \leq \Gamma I, \quad \forall k \geq 1$. **Remark.** The bound for the BFGS update can be found in (Moritz, Nishihara, and Jordan 2016).

**Lemma 4.** That $1/((D + M)L) \leq \gamma, \Gamma \leq 1/(D + M)L \delta^{D + M - 1/\mu}$, and the bound for the block BFGS update can be found in (Gower, Goldfarb, and Richtárik 2016).

**Theorem 1.** For convex function $R$ on $\mathbb{R}^D$, and $x, y \in \mathbb{R}^D$, $\|\phi(x; z; y\lambda) - \phi(y; z; y\lambda)\| \leq \|x - y\|$.

**Theorem 2.** For any composite function $P(x) = F(x) + R(x)$, where $F(x)$ is $\mu$-strongly convex and it has $L$-Lipschitz continuous gradient, $\phi(x)$ as a reference vector, $R(x)$ is convex. We define $x^- = x - \eta p, x^+ = \phi(x^-; x; y\lambda), q = \frac{1}{\eta}(x - x^+)$, where $p = \pi(p; \phi(x))$, and a proximal full gradient update (without explicit calculation) $\bar{x} \in \mathbb{R}^D$ as

$$\bar{x} = \text{prox}_{\eta R}(x - \eta \tau H \nabla F(x), \phi(x)), \quad \text{(28)}$$

and define $p \in \mathbb{R}^D, g \in \mathbb{R}^D, \Delta \in \mathbb{R}^D$ by

$$g_i = \begin{cases} q_i, (\sigma(x^-) - \sigma(x)) = 0 \\ p_i, (\sigma(x^-) - \sigma(x)) \neq 0, \end{cases} \quad \Delta = g - H \nabla F(x), \quad \text{(29)}$$

we define $\eta$ as a stepsize that $0 < \eta \leq 1/L$, then we have for any $y \in \mathbb{R}^D, x \in \mathbb{R}^D$ and $x^+ \in \mathbb{R}^D$,

$$\eta\|\Delta\|^2 - \Delta^T(x - y) + P(y) \quad \text{(30)}$$

$$\geq P(x^+) + q^T(y - x) + \frac{\eta}{2}\|q\|^2 + \frac{\mu}{2}\|y - x\|^2. \quad \text{(31)}$$

**Theorem 3.** Under the condition in Theorem 2, there is $\|g\| \leq \|p\|$. **Theorem 4.** We define a function as $Q(x) = \mathbb{E}[x - P(x_\ast)]$, where $x_\ast = \arg\min_x P(x)$. Under Assumption 1 and suppose each loss function is $\mu$-strongly convex, the Stochastic OWL-QN method in Algorithms 1 with step-size $0 < \theta < 1/(6L^2\gamma^2)$, the sequence of stale points $\{\hat{x}_\mu\}_{\mu=1}^\infty$ uniformly converges to the global optima $x_\ast$, in expectation, as

$$\mathbb{E}[Q(\hat{x}_\mu)] \leq \rho^\mu Q(\hat{x}_0), \quad \text{where } \rho = \frac{2 + 8\mu L^2\eta^2}{2\mu\eta(1 - 6L^2\eta^2)\delta^2}. \quad \text{(32)}$$

**Remarks.** When we set $\eta = \theta/(L^2\gamma^2)$, where $\theta$ is a relative small constant, we have $\rho \approx L^2\mu^2(\theta(1 - \theta/6\delta) + 6\theta / (1 - 6\theta))$, when $m$ is of the same order as $L^2\mu^2$. When $\theta < 1/6$, then $\rho$ is a constant in $(0, 1)$. To evaluate stochastic gradients $m$ times per outer-iteration, the computational complexity is $O((N + L^2\mu^2/\delta) \log(1/\epsilon))$ for obtaining an $\epsilon$-accurate solution.

**Theorem 5.** For the conditions in Theorem 3, but without the convexity assumption, if we define $y = \phi(x - \eta q; x; y\lambda)$, then there is

$$P(y) \leq P(z) + (y - z)^T(\nabla F(x) - g) + \left(\frac{L^2}{2}\right)\|y - x\|^2$$

$$+ \left(\frac{L^2}{2}\right)\|z - x\|^2 - \frac{\mu}{2}\|y - z\|^2,$$

for all $z \in \mathbb{R}^D$. **Remarks.** The proof is essentially the same with Theorem 3 without the convexity inequality.

**Theorem 6.** For loss functions of $L$-smoothness, which may be non-convex, suppose that Lemma 2 holds for the BFGS update matrix. By utilizing Algorithm 1, if we set the mini-batch size $M \leq N/2^3$, and step size as $\eta = M^3/2/(3LN)$, and the maximum outer iterations $T$ to be a multiple of $m$,

$$\mathbb{E}[\|q_k\|^2] \leq \frac{18L^2N^2(P(x_0) - P(x_\ast))}{M^{3/2}(3N - 2M^{3/2})Tm} \quad \text{(32)}$$

for all $z \in \mathbb{R}^D$. **Remarks.** When we set $\eta = \theta/(3L^2)$, and $m = N^{1/3}$, and $T$ to be a multiple of $m$, then we have $\mathbb{E}[\|q_k\|^2] \leq 18Q(x_0)/T$. To obtain a $\epsilon$-accurate solution, the overall computational complexity is $O((N + N^{2/3})/\epsilon)$. 
Numerical Experiments

We implement our methods in MATLAB, based on code generously provided by the authors of (Gower, Goldfarb, and Richtarik 2016). We verify the algorithm’s efficiency by $\ell_2$ and $\ell_1$ regularized logistic regression for classification task.

\[
P(x) = \frac{1}{N} \sum_n \log\left[\frac{1}{1 + \exp(-a_n^T x b_n)}\right] + \lambda_2 \|x\|_2^2 \tag{33}\]

\[
+ \lambda_1 \|x\|_1, \quad a_n \in \mathbb{R}^D, \quad b_n \in \{-1, 1\}. \tag{34}
\]

We use datasets from ((Chang and Lin 2011), including covtype, rcv1 and news20. The data dimensions span from 50 to 1M to test adaptiveness. The dataset information $N$, $D$ and the regularization parameters $\lambda_1$, $\lambda_2$ are noted along with figures. We compare our algorithm with other linearly convergent ones, including first order proximal-SVRG described in ((Xiao and Zhang 2014)), the proximal gradient version of SB-BFGS in ((Gower, Goldfarb, and Richtarik 2016)), proximal gradient version of SL-BFGS in ((Moritz, Nishihara, and Jordan 2016)) (referred by author initials MNJ). The convergence of quasi-Newton proximal gradient algorithms has been proven in ((Luo et al. 2016)) but without any experiments. We refer to them with prefix proximal. SOWL-QN type ones with prefix owlqn. We later remove the $\ell_2$ norm regularizer of the aforementioned logistic regression problem, for demonstrating non-strongly convex cases. In this setting, we also implement the SOWL-QN with two kinds of random sketching strategies (Gower, Goldfarb, and Richtarik 2016), as a) MNJ, which means no sketching is used, or $D_k$ is an identity matrix, b) gauss, which samples $D_k$ i.i.d from Gaussian distribution. c) prev, which stacks $M$ previous unequarranged direction vectors as $D_k = [d_{k+1-M}^T, \cdots, d_k^T]$.

For all the methods, we set $M = 5$ as the number of used curvature triples/pairs and the updating frequency of $H_k$, use the same subsampling size $|S_k| = \sqrt{N}$ for the gradient and Hessian update. The stepsize is grid searched within the span of $\eta \in \{10^{-3}, 5 \times 10^{-3}, 10^{-4}, \cdots, 10^{-12}\}$ for the best one. We used $m = N/|S_k|$ for the number of inner iterations, so that the algorithms fully scan over the dataset before recalculating the full gradient and updating the stale point. It is also worthwhile to mention that we tried modifying the eqnarrangement operator further, but did not get better results. For covtype, we plot the loss function w.r.t the datapasses and $t$-axis, and plot the results. For the non-convex case, we plot the loss functions $P(x_i)$ changes over datapasses as in Figure.(3). SOWL-QN method runs considerably faster than proximal-SVRG, by the nature of second order methods. The gauss and prev strategies perform slightly different in two cases, but both outperform MNJ, showing the effectiveness of the sketching technique under this setting. For the non-strongly convex case, we plot the loss functions $P(x_i)$ changes over datapasses as in Figure.(3). SOWL-QN method runs considerably faster than proximal-SVRG, by the nature of second order methods. The gauss and prev strategies perform slightly different in two cases, but all outperform MNJ, showing the effectiveness of the sketching technique under this setting. The MNJ type algorithms perform faster for high-dimensional data, since they have lighter computation workload, we show their convergence behavior with multiple stepsizes. In the non-convex case, SOWL-QN for neural networks converges considerably faster than the stochastic proximal gradient methods, as shown in Figure.(4). For the neural network case, SOWL-QN outperforms the proximal-SGD by more when the $\ell_1$ parameter is larger. This also agrees our intuition, since SOWL-QN is specifically designed to work well with strong $\ell_1$ regularizer.

As we see, although by the nature of orthant-wise eqnarrayment, a large percent of elements in gradients $v_k$ and directions $p_k$ are forced to zero during optimization, the SOWL-QN is able to converge with much larger stepsize than the corresponding proximal version, which leads to much faster convergence, especially in high dimensional setting. In general, SOWL-QN with or without sketching matrices perform better than its counterpart. This proves that the our proposed eqnarrayment operator does improve the gradient to better direction towards the global optima, making the algorithm considerably stabler with negligible extra arithmetic operations.

Conclusion

We proposed SOWL-QN, the first stochastic quasi-Newton method that specializing in solving $\ell_1$ regularized sparse models with experimental proof. Comparing to first order algorithms, the major feature is reducing dataset scans during optimization, trading computation for bandwidth. As in the past decades, the computation power grew much faster than the memory bandwidth, and it it expected that this trend will continue. We improve the variance reduced stochastic block BFGS by three novel eqnarrayment operators, which are modified by practice. We prove linear convergence of the algorithm, and the experiments show that the SOWL-QN outperforms other linear convergent algorithms.
The term that there exist a sub-gradient condition (1) and (2) for each element, we can conclude \( \sigma = 0 \) or \( \sigma(x_i) = \sigma(y_i) = \sigma(z_i) \), there is
\[
\phi_i(x; z; \eta \lambda) = \sigma(x_i) \max(|x_i| - \eta \lambda, 0) = (\text{prox}_{\eta R}(x_i)), \quad (35)
\]
reducing to a case included in Lemma 2; (2) else, there is \( x_i \neq 0 \), without loss of generality, we assume \( \sigma(x_i) = \sigma(z_i) \) and \( \sigma(y_i) \neq \sigma(z_i) \), so \( \phi_i(y_i; z; \eta \lambda) = \phi_i(0; z; \eta \lambda) \), then
\[
||\phi_i(x; z; \eta \lambda) - \phi_i(0; z; \eta \lambda)|| \leq ||x_i - 0|| \leq ||x_i - y_i||; \quad (36)
\]
(3) else, there is \( x_i \neq 0, \sigma(x_i) \neq \sigma(z_i) \) and \( \sigma(y_i) \neq \sigma(z_i) \), so \( \phi_i(x_i; z; \eta \lambda) - \phi_i(y_i; z; \eta \lambda) = 0 \). Combining the inequalities for all elements, we get the proof.

**Proof of Theorem 2.**

We define a subgradient of \( R \) as \( \xi \in \partial R(x^+) \), for each element \( \xi_i, 1 \leq i \leq D \), (1) if \( \sigma(x_i^+) \sigma(x_i) \neq -1 \), we can obtain \( x_i^+ = \sigma(x_i^-) \max(|x_i^-| - \eta \lambda, 0) \), which is exactly the optima of
\[
x_i^+ = \arg\min_{y_i} \frac{1}{2} ||y_i - (x_i - \eta p_i)||^2 + \eta R(y_i), \quad (37)
\]
then by the optimality property of \( x^+ \) and the definition \( q_i = (x_i - x_i^-)/\eta \), it holds
\[
x_i^+ - (x_i - \eta p_i) + \eta q_i = 0 \Rightarrow \xi_i = q_i - p_i, \quad (38)
\]
(2) else, there is \( \sigma(x_i^+) \sigma(x_i) = -1 \), then \( x_i^+ = 0 \), leading to \( \xi_i = 0 \), which is rewritten as \( \xi_i = q_i - p_i \). Combining condition (1) and (2) for each element, we can conclude that there exist a sub-gradient \( \xi = q - g \in \partial R(x^+) \).

The function \( P(x) \) is lower bounded by the strong convexity of \( F \) and \( R \), for any \( x_i \in \mathbb{R}^D \),
\[
P(y) = F(y) + R(y) \geq F(x) + \nabla F(x)^T (y - x) + \mu ||y - x||^2 + R(x^+) + \xi^T (y - x^+), \quad (39)
\]
The term \( F(x) \) is further bounded by its smoothness,
\[
F(x) \geq F(x^+) - \nabla F(x)^T (x^+ - x) - \frac{L}{2} ||x^+ - x||^2. \quad (40)
\]

Therefore there is
\[
P(y) \geq F(x^+) - \nabla F(x)^T (x^+ - x) - \frac{L}{2} ||x^+ - x||^2 + \nabla F(x)^T (y - x) + \frac{\mu}{2} ||y - x||^2 + R(x^+) + \xi^T (y - x^+) \geq P(x^+) - \nabla F(x)^T (x^+ - x) - \frac{L}{2} \eta^2 ||q||^2 + \nabla F(x)^T (y - x) + \frac{\mu}{2} ||y - x||^2 + \xi^T (y - x^+) \quad (42)
\]
where in the last inequality we used and \( x^+ - x = -\eta q \). For the production terms on the right-hand side, we have
\[
-\nabla F(x)^T (x^+ - x) + \nabla F(x)^T (y - x) + \xi^T (y - x^+) = \nabla F(x)^T (y - x^+) + (q - g)^T (y - x^+) = q^T (y - x^+) + (q - \nabla F(x))^T (x^+ - y) < q^T (y - x^+) + \xi^T (y - x^+) \quad (43)
\]
where we used \( \xi = q - g, \Delta = g - \nabla F(x) \) and \( x^+ - x^+ = \eta q \) in the first, third, last equalities respectively. Substituting Eq. (42) to Eq. (43), we obtain Eq. (40).

**Proof of Theorem 3.**

We apply the Cauchy-Schwarz inequality for each element \( \Delta_i(x_i^+ - y_i), \forall 1 \leq i \leq D \),
\[
-\Delta_i(x_i^+ - y_i) = -\Delta_i(x_i^+ - \bar{x}_i) - \Delta_i(x_i^+ - \bar{x}_i) \leq ||\Delta_i|| ||x_i^+ - \bar{x}_i|| - \Delta_i(x_i^+ - \bar{x}_i) \quad (44)
\]
Then we apply Lemma 2 if \( \sigma(x_i^-) \sigma(x_i) \neq -1 \), and apply Theorem 1 if \( \sigma(x_i^-) \sigma(x_i) = -1 \), we have \( ||x_i^+ - \bar{x}_i|| \leq \eta ||\Delta_i|| \) by the definition of \( g \) and \( \Delta \). Combining them, we get the first inequality.

**Proof of Theorem 4.**

We prove the theorem by analyse how the Euclidean distance between \( x_k \) and \( x_s \) changes over iterations. We define an auxiliary function
\[
\mathcal{L}_k = ||x_k - x_m||^2 - ||x_m - x_k||^2, \quad (45)
\]
where we use \( x_k = x_{m,k-1} - \eta q_k \). Then we apply Theorem 2 with substitution that \( x = x_{m,k-1}, v = q_k, \)
\[ x^+ = x_k, \quad q = q_k, \quad \text{and } y = x_*, \quad \text{and auxiliary variables}, \]
\[ \bar{p} = \pi(H_k \nabla F(x_{m,k-1}) \otimes \bar{f}_k(x_{m,k-1})) \]
\[ -q^T_k(x_{m,k-1} - x_*) + \eta\|q_k\|^2 \leq P(x_*) - P(x_k) \quad (47) \]
\[ -\frac{\mu}{2}\|x_{m,k-1} - x_*\|^2 - \Delta_k^T(x_k - x_*) \quad (48) \]

Now we take expectation on both sides of the above inequality with respect to \(S_k\) to obtain
\[ E[\mathcal{L}_k] \leq -2\eta E[\langle q_k \rangle] + 2\eta^2 E[\|\Delta_k\|^2] - 2\eta E[\Delta_k^T(\bar{x}_k - x_*)] \quad (49) \]

Recalling the definition of the gradient eqnarrayment operations, we have
\[ E[\Delta_k] = E[\langle q_k \rangle] - E[H_k \nabla F(x_{m,k-1})] = 0. \]

We note that both \(\bar{x}_k\) and \(x_*\) are independent of the random set \(S_k\), therefore they are not correlated with the term \(\Delta_k\).
\[ E[\Delta_k^T(\bar{x}_k - x_*)] = E[\Delta_k]^T(\bar{x}_k - x_*) = 0. \quad (50) \]
Recalling the definition of $p_k$ and Lemma 3, there is
\[ \mathbb{E}[||p_k||^2] \leq \mathbb{E}[||H_k v_k||^2] \leq \Gamma^2 \mathbb{E}[||v_k||^2], \quad (51) \]
by the definition of spectral norm. We write the term $\mathbb{E}[||\Delta_k||^2] = \mathbb{E}[||g_k - H_k \nabla F(x_{m,k-1})||^2]$, and apply Theorem 3 that $||g_k||^2 \leq ||p_k||^2$, then
\[ \mathbb{E}[||\Delta_k||^2] = \mathbb{E}[||g_k||^2 - 2g_k^T H_k \nabla F(x_{m,k-1}) + ||H_k \nabla F(x_{m,k-1})||^2] \leq \mathbb{E}[||p_k||^2 - 2(p_k + \xi_k)^T \nabla F(x_{m,k-1}) + \Gamma^2 \||\nabla F(x_{m,k-1})||^2]. \]
where $\xi_k = g_k - p_k$ is dependent of data subset $S_k$, therefore is independent of $\nabla F(x_{m,k-1})$
\[ \mathbb{E}[||\Delta_k||^2] \leq \mathbb{E}[||p_k||^2 - 2p_k^T H_k \nabla F(x_{m,k-1}) + \Gamma^2 ||\nabla F(x_{m,k-1})||^2]. \]
Notice that $p_k - H_k v_k$ is a random variable with zero means, so
\[ \mathbb{E}[p_k^T H_k \nabla F(x_{m,k-1})] = \nabla F(x_{m,k-1})^T H_k^T H_k \nabla F(x_{m,k-1}) \geq \gamma^2 \||\nabla F(x_{m,k-1})||^2 \]
Substituting this into term $||\Delta_k||^2$, then there is
\[ \mathbb{E}[||\Delta_k||^2] \leq \Gamma^2 \mathbb{E}[||v_k||^2] + 2 \Gamma^2 \mathbb{E}[||\nabla F(x_{m,k-1})||^2] \leq 6 \Gamma L^2 Q_{k-1} + 4 \Gamma^2 \mathbb{E}[Q(x)], \quad (53) \]
where we apply Lemma 1 to bound the term $||v_k||$ and $||\nabla F(x_{m,k-1})||$. Substituting Eq.(53) into Eq.(45),
\[ \mathcal{L}_k \leq -2 \eta Q_k + 12 \Gamma^2 L^2 \mathbb{E}[Q(x)] + 8 \Gamma^2 \eta^2 \mathbb{E}[Q(x)]. \]
We take the optimization trajectory from with $(t-1)$-th step for analysis, where the state point $\bar{x}_{t-1}$ were used, then
\[ x_0 = \hat{x} = \bar{x}_{t-1}, \quad \bar{x}_t = \frac{\xi_t}{\alpha} : \sum_{k=1}^{m} x_k. \]
By combining the inequalities in Eq.(55), at iterations $k = 1, \ldots, m$, we obtain
\[ \mathbb{E}[||x_m - x_0||^2 + 2 \eta Q_m + \alpha \sum_{k=1}^{m-1} Q_k] \leq ||x_0 - x_0||^2 + \beta Q(x_0) + m \mathbb{E}[Q(x)], \quad (56) \]
where $\alpha = 2 \eta (1 - 6 \Gamma L^2 \eta)$ and $\beta = 8 \Gamma^2 \eta^2$. Notice that $2 \eta (1 - 6 \Gamma L^2 \eta) < 2 \eta$ by our setting, and $x_0 = \bar{x}_{t-1}$, so we have
\[ \alpha \sum_{k=1}^{m} Q_k \leq ||\bar{x}_{t-1} - x_0||^2 + \beta (m + 1) Q(x_0). \]
By the strong convexity of $P$ and definition of $\bar{x}_t$, we have
\[ P(\bar{x}_t) \leq \frac{1}{m} \sum_{k=1}^{m} P(x_k), ||\bar{x}_{t-1} - x_0||^2 \leq \frac{2}{\mu} \cdot Q(\bar{x}_{t-1}). \]
Substituting it back, we have $\alpha m \mathbb{E}[Q(\bar{x}_t)] \leq (\frac{2}{\mu} + \beta (m + 1)) \mathbb{E}[Q(\bar{x}_{t-1})]$. Recall that $\eta < 1/(6 \Gamma L^2)$, there is $\alpha > 0$, dividing both sides of the above inequality by $\alpha m$, we arrive at
\[ Q(\bar{x}_t) \leq (\frac{2}{\mu \alpha m} + \beta \frac{(m + 1)}{\alpha m}) Q(\bar{x}_{t-1}). \]
Define $\rho = \frac{2}{\mu \alpha m} + \beta \frac{(m + 1)}{\alpha m}$, and apply the above inequality recursively, we get $Q(\bar{x}_t) \leq \rho^t Q(\bar{x}_0)$, which is our main theorem.

**Proof of Theorem 6.**
We define two auxiliary variables,
\[ \bar{p}_k = \pi (H_k \nabla F(x_{m,k-1}); \phi f(x_{m,k-1})), \quad (61) \]
\[ \bar{x}_k = \phi (x_{m,k-1} - \eta \bar{p}_k, x_{m,k-1}, \eta \lambda), \quad (62) \]
which are dependent of $x_{m,k-1}$ but independent of random set $S_k$, then we apply Theorem 5, and substitute $y = \bar{x}_k, z = x_{m,k-1}, d = p$, to get
\[ \mathbb{E}[P(\bar{x}_k)] \leq \mathbb{E}[P(x_{m,k-1}) + (\frac{L}{2} - \frac{1}{2 \eta}) ||\bar{x}_k - x_{m,k-1}||^2 \leq \eta \mathbb{E}[\bar{x}_k - x_{m,k-1}||^2]], \quad (63) \]
\[ \mathbb{E}[P(x_k)] \leq \mathbb{E}[P(x_{m,k-1}) + (\frac{L}{2} - \frac{1}{2 \eta}) ||\bar{x}_k - x_{m,k-1}||^2 \leq \eta \mathbb{E}[\bar{x}_k - x_{m,k-1}||^2]], \quad (64) \]
Combining the two inequalities above, we get
\[ \mathbb{E}[P(x_k)] \leq \mathbb{E}[P(x_{m,k-1}) + (\bar{x}_k - x_{m,k-1})^T H_k \nabla F(x_{m,k-1}) - v_k + (\frac{L}{2} - \frac{1}{2 \eta}) ||\bar{x}_k - x_{m,k-1}||^2 \leq \eta \mathbb{E}[\bar{x}_k - x_{m,k-1}||^2]], \quad (65) \]
\[ + (\frac{L}{2} + \frac{1}{2 \eta}) ||x_k - x_{m,k-1}||^2 - \frac{1}{2 \eta} ||\bar{x}_k - x_{m,k-1}||^2]], \quad (66) \]
we can bound the inner product term by
\[ (\bar{x}_k - x_{m,k-1})^T H_k \nabla F(x_{m,k-1}) - v_k \leq \frac{1}{2 \eta} \mathbb{E}[||\bar{x}_k - x_{m,k-1}||^2] + \eta \frac{L^2}{2} \mathbb{E}[||\nabla F(x_{m,k-1}) - v_k||^2 \leq \frac{1}{2 \eta} \mathbb{E}[||\bar{x}_k - x_{m,k-1}||^2] + \eta \frac{L^2}{2} \mathbb{E}[||\nabla F(x_{m,k-1}) - \hat{x}||^2]], \quad (68) \]
from the Cauchy-Schwarz inequality, and the smoothness of $F$. Substituting this into Eq.(67), we have
\[ \mathbb{E}[P(x_k)] \leq \mathbb{E}[P(x_{m,k-1}) + (L - \frac{1}{2 \eta}) ||\bar{x}_k - x_{m,k-1}||^2 \leq \frac{1}{2 \eta} \mathbb{E}[||\bar{x}_k - x_{m,k-1}||^2] + \eta \frac{L^2}{2} \mathbb{E}[||\nabla F(x_{m,k-1} - \hat{x}||^2]], \quad (69) \]
we analyse the distance between the current estimation with the state point $\hat{x}$, to be specific,
\[ \mathbb{E}[||x_k - \hat{x}||^2] = \mathbb{E}[||x_k - x_{m,k-1} + x_{m,k-1} - \hat{x}||^2] = \mathbb{E}[||x_k - x_{m,k-1}||^2 + ||x_{m,k-1} - \hat{x}||^2 + 2(x_k - x_{m,k-1})^T (x_{m,k-1} - \hat{x})] \quad (70) \]
Combining these inequalities together, we have
\[
\mathcal{L}_k = \mathbb{E}[P(x_k) + c_k \|x_k - \bar{x}\|^2] \\
= \mathbb{E}[P(x_k)] + c_k \mathbb{E}[\|x_k - x_{m,k-1}\|^2 + \|x_{m,k-1} - \bar{x}\|^2] \\
+ 2\mathbb{E}[(x_k - x_{m,k-1})^T(x_{m,k-1} - \bar{x})] \\
\leq \mathbb{E}[P(x_k)] + \mathbb{E}[c_k(1 + \frac{1}{\beta})\|x_k - x_{m,k-1}\|^2] \\
+ c_k(1 + \beta)\|x_{m,k-1} - \bar{x}\|^2
\]
\[
\leq \mathbb{E}[P(x_{m,k-1}) + (L - \frac{1}{2\eta})\|\bar{x}_k - x_{m,k-1}\|^2 + [c_k(1 + \frac{1}{\beta})] \\
+ \frac{L}{2} - \frac{1}{2\eta}\|x_k - x_{m,k-1}\|^2 \\
+ [c_k(1 + \beta) + \frac{\eta L^2 \Gamma^2}{2M}]\|x_{m,k-1} - \bar{x}\|^2
\]
\[
\leq \mathbb{E}[P(x_{m,k-1}) + (L - \frac{1}{2\eta})\|\bar{x}_k - x_{m,k-1}\|^2 + [c_k(1 + \frac{1}{\beta})] \\
+ \frac{L}{2} - \frac{1}{2\eta}\|x_k - x_{m,k-1}\|^2 \\
+ [c_k(1 + \beta) + \frac{\eta L^2 \Gamma^2}{2M}]\|x_{m,k-1} - \bar{x}\|^2
\]
\[
= \mathcal{L}_{k-1} + (L - \frac{1}{2\eta})\|\bar{x}_{k-1} - x_{m,k-1}\|^2 \\
= \mathcal{L}_{k-1} + (L - \frac{1}{2\eta})\|\bar{x}_{k-1} - x_{m,k-1}\|^2
\]
where the third inequality is due to the following inequality
\[
c_k(1 + \frac{1}{\beta}) + \frac{L}{2} \leq \frac{\eta L^2 \Gamma^2 (1 + \frac{M}{N})^{m-k-1} - 1}{\beta}
\]
\[
= \frac{L \Gamma^2}{6\sqrt{M}}[(1 + \frac{M}{N})^{m-k-1} - 1] \\
\leq \frac{L \Gamma^2}{6\sqrt{M}}[(1 + N)\frac{1}{\sqrt{L}} - 1] \\
\leq \frac{L \Gamma^2}{6\sqrt{M}}[(1 + N)\frac{1}{6\sqrt{M}} - 1] \\
\leq \frac{L \Gamma^2}{6\sqrt{M}}[(e - 1)\frac{1}{6\sqrt{M}}]
\]
where the first inequality is due to \(m = N/M\), and the second inequality is by definition of \(e\), the Euler’s number.

\[
c_k(1 + \frac{1}{\beta}) + \frac{L}{2} \leq \frac{L \Gamma^2}{6\sqrt{M}}[(1 + N)\frac{1}{\sqrt{L}} + \frac{L}{2}] \\
\leq \frac{L \Gamma^2}{3\sqrt{M}}[(1 + N)\frac{1}{\sqrt{L}} + \frac{L}{2}] \\
\leq \frac{3L \Gamma^2}{2\sqrt{M}} = \frac{1}{2\eta}
\]
where the third inequality is by \(N \geq M^{3/2}\). By summing the inequalities from \(k = mt\) to \(k = m(t + 1)\), we have

\[
\mathcal{L}_{m(t+1)} \leq \mathcal{L}_{mt} + \sum_{k=mt}^{m(t+1)} (L - \frac{1}{2\eta})\mathbb{E}[\|\bar{x}_k - x_{m,k-1}\|^2].
\]

And by the definition that \(x_{mt} = \bar{x}\) in this outer iteration, we have

\[
\mathbb{E}[P(\bar{x}_{t+1})] \leq \mathbb{E}[P(\bar{x}_t)] + \sum_{k=mt}^{m(t+1)} (L - \frac{1}{2\eta})\mathbb{E}[\|\bar{x}_k - x_{m,k-1}\|^2].
\]

And by summing up all the outer iterations,

\[
\sum_{k=0}^{m(t+1)} (L - \frac{1}{2\eta})\mathbb{E}[\|\bar{x}_k - x_{m,k-1}\|^2] \\
\leq P(x_0) - \mathbb{E}[P(x_{m(t+1)})] \leq P(x_0) - P(\bar{x})
\]

References

[Allen-Zhu and Hazan 2016] Allen-Zhu, Z., and Hazan, E. 2016. Variance reduction for faster non-convex optimization. In Proceedings of The 33rd International Conference on Machine Learning, 699–707.

[Allen-Zhu and Yuan 2016] Allen-Zhu, Z., and Yuan, Y. 2016. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In Proceedings of The 33rd International Conference on Machine Learning, 1080–1089.

[Andrew and Gao 2007] Andrew, G., and Gao, J. 2007. Scalable training of \(\ell_1\)-regularized log-linear models. In Proceedings of the 24th international conference on Machine learning, 33–40. ACM.

[Bottou 2010] Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT’2010. Springer. 177–186.

[Byrd et al. 2012] Byrd, R. H.; Chin, G. M.; Nocedal, J.; and Oztok, F. 2012. A family of second-order methods for convex \(\ell_1\)-regularized optimization. Mathematical Programming 1–33.

[Byrd et al. 2016] Byrd, R. H.; Hansen, S.; Nocedal, J.; and Singer, Y. 2016. A stochastic quasi-newton method for large-scale optimization. SIAM Journal on Optimization 26(2):1008–1031.

[Chang and Lin 2011] Chang, C.-C., and Lin, C.-J. 2011. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3):27.

[Defazio, Bach, and Lacoste-Julien 2014] Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Advances in Neural Information Processing Systems, 1646–1654.

[Gong and Ye 2015] Gong, P., and Ye, J. 2015. A modified orthant-wise limited memory quasi-newton method with convergence analysis. In Proceedings of the 32nd International Conference on Machine Learning (ICML-15), 276–284.

[Gower, Goldfarb, and Richtárik 2016] Gower, R. M.; Goldfarb, D.; and Richtárik, P. 2016. Stochastic block bfgs: Squeezing more curvature out of data. 276–284.

[J. Reddi et al. 2016] J. Reddi, S.; Sra, S.; Poczos, B.; and Smola, A. 2016. Fast stochastic methods for nonsmooth nonconvex optimization. arXiv:1605.06900.

[Johnson and Zhang 2013] Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in Neural Information Processing Systems, 315–323.
[Luo et al. 2016] Luo, L.; Chen, Z.; Zhang, Z.; and Li, W.-J. 2016. A proximal stochastic quasi-newton algorithm. *arXiv preprint arXiv:1602.00223*.

[Mokhtari and Ribeiro 2014] Mokhtari, A., and Ribeiro, A. 2014. Res: Regularized stochastic bfgs algorithm. *IEEE Transactions on Signal Processing* 62(23):6089–6104.

[Moritz, Nishihara, and Jordan 2016] Moritz, P.; Nishihara, R.; and Jordan, M. 2016. A linearly-convergent stochastic l-bfgs algorithm. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 249–258.

[Oztoprak et al. 2012] Oztoprak, F.; Nocedal, J.; Rennie, S.; and Olsen, P. A. 2012. Newton-like methods for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems*, 755–763.

[Reddi et al. 2016] Reddi, S. J.; Hefny, A.; Sra, S.; Poczos, B.; and Smola, A. 2016. Stochastic variance reduction for nonconvex optimization. *arXiv preprint arXiv:1603.06160*.

[Schraudolph et al. 2007] Schraudolph, N. N.; Yu, J.; Günter, S.; et al. 2007. A stochastic quasi-newton method for online convex optimization. In *AISTATS*, volume 7, 436–443.

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

[Tseng and Yun 2009] Tseng, P., and Yun, S. 2009. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming* 117(1-2):387–423.

[Yuan et al. 2010] Yuan, G.-X.; Chang, K.-W.; Hsieh, C.-J.; and Lin, C.-J. 2010. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *Journal of Machine Learning Research* 11(Nov):3183–3234.