Large Scale Optimization with Proximal Stochastic Newton-Type Gradient Descent

Ziqiang Shi(✉) and Rujie Liu

Fujitsu Research and Development Center,Beijing, China
{shiziqiang,rjliu}@cn.fujitsu.com

Abstract. In this work, we generalized and unified two recent completely different works of Jascha [10] and Lee [2] respectively into one by proposing the **prox**imal s**to**chastic **N**ewton-type gradient (PROXTONE) method for optimizing the sums of two convex functions: one is the average of a huge number of smooth convex functions, and the other is a nonsmooth convex function. Our PROXTONE incorporates second order information to obtain stronger convergence results, that it achieves a linear convergence rate not only in the value of the *objective* function, but also for the *solution*. The proofs are simple and intuitive, and the results and technique can be served as a initiate for the research on the proximal stochastic methods that employ second order information. The methods and principles proposed in this paper can be used to do logistic regression, training of deep neural network and so on. Our numerical experiments shows that the PROXTONE achieves better computation performance than existing methods.

1 Introduction and Problem Statement

In this work, we consider the problems of the following form:

$$\min_{x\in\mathbb{R}^p} f(x) := \frac{1}{n}\sum_{i=1}^{n} g_i(x) + h(x), \tag{1}$$

where g_i is a smooth convex loss function associated with a sample in a training set, and h is a non-smooth convex penalty function or a regularizer. Let $g(x) = \frac{1}{n}\sum_{i=0}^{n} g_i(x)$. We assume the optimal value $f^\star$ is attained at some optimal solution $x^\star$, not necessarily unique. Problems of this form often arise in machine learning, such as the least-squares regression, the Lasso, the elastic net, the logistic regression, and deep neural network.

For optimizing (1), the standard and popular *proximal full gradient method* (ProxFG) uses iterations of the form

$$x^{k+1} = \arg\min_{x\in\mathbb{R}^p} \left\{ \nabla g(x_k)^T x + \frac{1}{2\alpha_k}\|x - x_{k-1}\|^2 + h(x) \right\}, \tag{2}$$

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 691–704, 2015.
DOI: 10.1007/978-3-319-23528-8_43

where α_k is the step size at the k-th iteration. Under standard assumptions the sub-optimality achieved on iteration k of the ProxFG method with a constant step size is given by

$$f(x^k) - f(x^*) = O(\frac{1}{k}).$$

When f is strongly-convex, the error satisfies [11]

$$f(x^k) - f(x^*) = O((\frac{L - \mu_g}{L + \mu_h})^k),$$

where L is the Lipschitz constant of $f(x)$, μ_g, and μ_h are the convexity parameters of $g(x)$ and $h(x)$ respectively. These notations mentioned here will be detailed in Section 1.1. This result in a linear convergence rate, which is also known as a geometric or exponential rate because the error is cut by a fixed fraction on each iteration.

Unfortunately, the ProxFG methods can be unappealing when n is large or huge because its iteration cost scales linearly in n. When the number of components n is very large, then each iteration of (2) will be very expensive since it requires computing the gradients for all the n component functions g_i, and also their average.

To overcome this problem, researchers proposed the *proximal stochastic gradient descent* methods (ProxSGD), whose main appealing is that they have an iteration cost which is independent of n, making them suited for modern problems where n may be very large. The basic ProxSGD method for optimizing (1), uses iterations of the form

$$x_k = \text{prox}_{\alpha_k h}\big(x_{k-1} - \alpha_k \nabla g_{i_k}(x_{k-1})\big), \tag{3}$$

where at each iteration an index i_k is sampled uniformly from the set $\{1, ..., n\}$. The randomly chosen gradient $\nabla g_{i_k}(x_{k-1})$ yields an unbiased estimate of the true gradient $\nabla g(x_{k-1})$ and one can show under standard assumptions that, for a suitably chosen decreasing step-size sequence $\{\alpha_k\}$, the ProxSGD iterations have an expected sub-optimality for convex objectives of [1]

$$\mathbb{E}[f(x^k)] - f(x^*) = O(\frac{1}{\sqrt{k}})$$

and an expected sub-optimality for strongly-convex objectives of

$$\mathbb{E}[f(x^k)] - f(x^*) = O(\frac{1}{k}).$$

In these rates, the expectations are taken with respect to the selection of the i_k variables.

Besides these first order method, there is another group of methods, called *proximal Newton-type* methods, which converge much faster, but need more memory and computation to obtain the second order information about the objective function. These methods are always limited to small-to-medium scale problems that require a high degree of precision. For optimizing (1), proximal Newton-type methods [2] that incorporate second order information use iterations of the form $x^{k+1} \leftarrow x^k + \Delta x^k$, here Δx^k is obtained by

$$\Delta x^k = \arg\min_{d\in\mathbb{R}^p} \nabla g(x^k)^T d + \frac{1}{2} d^T H_k d + h(x^k + d), \tag{4}$$

where H_k denotes an approximation to $\nabla^2 g(x_k)$. According to the strategies for choosing H_k, we obtain different method, such as *proximal Newton method* (ProxN) when we choose H_k to be $\nabla^2 g(x^k)$; *proximal quasi-Newton method* (ProxQN) when we build an approximation to $\nabla^2 g(x_k)$ using changes measured in ∇g according to a quasi-Newton strategy [2]. Indeed if we compared (4) with (2), it can be seen ProxN is the ProxFG with scaled proximal mappings.

Based on the related background introduced above, now we can describe our approaches and findings. The primary contribution of this work is the proposal and analysis of a new algorithm that we call the proximal stochastic Newton-type gradient (PROXTONE, pronounced /prok stone/) method, a stochastic variant of the ProxN method. The PROXTONE method has the low iteration cost as that of ProxSGD methods, but achieves the convergence rates like the ProxFG method stated above. The PROXTONE iterations take the form $x^{k+1} \leftarrow x^k + t_k \Delta x^k$, where Δx^k is obtained by

$$\Delta x^k \leftarrow \arg\min_{d} d^T(\nabla_k + H_k x^k) + \frac{1}{2} d^T H_k d + h(x^k + d), \tag{5}$$

here $\nabla_k = \frac{1}{n}\sum_{i=1}^n \nabla_k^i$, $H_k = \frac{1}{n}\sum_{i=1}^n H_k^i$, and at each iteration a random index j and corresponding H_{k+1}^j is selected, then we set

$$\nabla_{k+1}^i = \begin{cases} \nabla g_i(x^{k+1}) - H_{k+1}^i x^{k+1} & \text{if } i = j, \\ \nabla_{k+1}^i & \text{otherwise.} \end{cases}$$

and $H_{k+1}^i \leftarrow H_k^i$ $(i \neq j)$.

That is, like the ProxFG and ProxN methods, the steps incorporates a gradient with respect to each function; but, like the ProxSGD method, each iteration only computes the gradient with respect to a single example and the cost of the iterations is independent of n. Despite the low cost of the PROXTONE iterations, we show in this paper that the PROXTONE iterations have a linear convergence rate for strongly-convex objectives, like the ProxFG method. That is, by having access to j and by keeping a memory of the approximation for the Hessian matrix computed for the objective funtion, this iteration achieves a faster convergence rate than is possible for standard ProxSGD methods.

Besides PROXTONE, there are a large variety of approaches available to accelerate the convergence of ProxSGD methods, and a full review of this immense literature would be outside the scope of this work. Several recent work considered various special cases of (1), and developed algorithms that enjoy the linear convergence rate, such as ProxSDCA [8], MISO [3], SAG [7], ProxSVRG [11], SFO [10], and ProxN [2]. All these methods converge with an exponential rate in the value of the objective function, except that the ProxN achieves superlinear rates of convergence for the *solution*, however it is a batch mode method. Shalev-Shwartz and Zhang's ProxSDCA [8,9] considered the case where the component functions have the form $g_i(x) = \phi_i(a_i^T x)$ and the Fenchel conjugate functions of ϕ_i and h can be computed efficiently. Schimidt et al.'s SAG [7] and Jascha et al.'s SFO [10] considered the case where $h(x) \equiv 0$.

Different from above related methods, our PROXTONE is a extension of the SFO and ProxN to a proximal stochastic Newton-type method for solving the more *general* nonsmooth (compared to ProxSDCA, SAG and SFO) class of problems defined in (1). PROXTONE makes connections between two completely different approaches. It achieves a linear convergence rate not only in the value of the objective function, but also for the *solution*. We now outline the rest of the study. Section 2 presents the main algorithm and gives an equivalent form in order for the ease of analysis. Section 3 states the assumptions underlying our analysis and gives the main results; we first give a linear convergence rate in *function value* (weak convergence) that applies for any problem, and then give a strong linear convergence rate for the *solution*, however with some additional conditions. We report some experimental results in Section 4 and provide concluding remarks in Section 5.

1.1 Notations and Assumptions

In this paper, we assume the function $h(x)$ is lower semi-continuous and convex, and its effective domain, $\mathrm{dom}(h) := \{x \in \mathbb{R}^p \mid h(x) < +\infty\}$, is closed. Each $g_i(x)$, for $i = 1, \ldots, n$, is differentiable on an open set that contains $\mathrm{dom}(h)$, and their gradients are Lipschitz continuous, that is, there exist $L_i > 0$ such that for all $x, y \in \mathrm{dom}(h)$,

$$\|\nabla g_i(x) - \nabla g_i(y)\| \le L_i \|x - y\|. \tag{6}$$

Then from the Lemma 1.2.3 and its proof in Nesterov's book [5], for $i = 1, \ldots, n$, we have

$$|g_i(x) - g_i(y) - \nabla g_i(y)^T (x - y)| \le \frac{L_i}{2} \|x - y\|^2. \tag{7}$$

A function $f(x)$ is called μ-strongly convex, if there exist $\mu \ge 0$ such that for all $x \in \mathrm{dom}(f)$ and $y \in \mathbb{R}^p$,

$$f(y) \ge f(x) + \xi^T (y - x) + \frac{\mu}{2} \|y - x\|^2, \quad \forall \xi \in \partial f(x). \tag{8}$$

The *convexity parameter* of a function is the largest μ such that the above condition holds. If $\mu = 0$, it is identical to the definition of a convex function. The strong convexity of $f(x)$ in (1) may come from either $g(x)$ or $h(x)$ or both. More precisely, let $g(x)$ and $h(x)$ have convexity parameters μ_g and μ_h respectively, then $\mu \ge \mu_g + \mu_h$. From Lemma B.5 in [3] and (8), we have

$$f(y) \ge f(x^*) + \frac{\mu}{2} \|y - x^*\|^2. \tag{9}$$

2 The PROXTONE Method

In this section we present the Proximal Stochastic Newton-type Gradient Descent (PROXTONE) algorithm for solving problems of the form (1). There are two key steps in the algorithm: (step 2) the regularized quadratic model (5) is solved to give a search direction; (step 4) the component function $g_j(x)$ is sampled randomly and the regularized quadratic model (5) is updated using this selected function. Once these key steps have been performed, the current point x_k is updated to give a new point x_{k+1}, and the process is repeated.

We summarize the PROXTONE method of (5) in Algorithm 1, while a thorough description of each of the key steps in the algorithm will follow in the rest of this section. It can be easily checked that if $n = 1$, then it becomes the determined proximal Newton-type methods proposed by Lee and Sun et al. [2] for minimizing composite functions:

$$\min_{x \in \mathbb{R}^p} f(x) := g(x) + h(x) \tag{10}$$

by (4). Thus PROXTONE is indeed a generalization of ProxN [2].

Algorithm 1. PROXTONE: A generic PROXimal sTOchastic NEwton-type gradient descent method

```
Input: start point $x^0 \in \text{dom } f$; for $i \in \{1, 2, .., n\}$, let $H_{-1}^i = H_0^i$ be a positive definite approximation to the Hessian of $g_i(x)$ at $x^0$, $\nabla_{-1}^i = \nabla_0^i = \nabla g_i(x^0) - H_0^i x^0$; and $\nabla_0 = \frac{1}{n}\sum_{i=1}^n \nabla_0^i$, $H_0 = \frac{1}{n}\sum_{i=0}^n H_0^i$.
1: repeat
2: Solve the subproblem for a search direction:$\triangle x^k \leftarrow \arg\min_d d^T(\nabla_k + H_k x^k) + \frac{1}{2} d^T H_k d + h(x^k + d)$.
3: Update: $x^{k+1} = x^k + \triangle x^k$.
4: Sample $j$ from $\{1, 2, .., n\}$, use the $\nabla g_j(x^{k+1})$ and $H_{k+1}^j$, which is a positive definite approximation to the Hessian of $g_j(x)$ at $x^{k+1}$, to update the $\nabla_{k+1}^i$ $(i \in \{1, 2, .., n\})$: $\nabla_{k+1}^j \leftarrow \nabla g_j(x^{k+1}) - H_{k+1}^j x^{k+1}$, while leaving all other $\nabla_{k+1}^i$ and $H_{k+1}^i$ unchanged: $\nabla_{k+1}^i \leftarrow \nabla_k^i$ and $H_{k+1}^i \leftarrow H_k^i$ $(i \neq j)$ ; and finally obtain $\nabla_{k+1}$ and $H_{k+1}$ by $\nabla_{k+1} \leftarrow \frac{1}{n}\sum_{i=1}^n \nabla_{k+1}^i$, $H_{k+1} \leftarrow \frac{1}{n}\sum_{i=1}^n H_{k+1}^i$.
5: until stopping conditions are satisfied.
Output: $x^k$.
```

It is also a generalization of recent work by Jascha [10], whose SFO is the special case of our PROXTONE with $h(x) \equiv 0$. Our algorithm in Jascha's style is summarized in Algorithm 2 which is equivalent to the original PROXTONE. To see the equivalence, keep in mind that $G^k(x)$ in Algorithm 2 is a quadratic function, we only need to check the following equations:

$$\nabla^2 G^k(x) = \frac{1}{n}\sum_{i=1}^n H_k^i \quad \text{and} \quad \nabla G^k(x) = \frac{1}{n}\sum_{i=1}^n \nabla g_i(x) + \frac{1}{n}\sum_{i=1}^n (x - x^k)^T H_k^i,$$

and

$$\nabla_k + H_k x^k = \frac{1}{n}\sum_{i=1}^{n}[\nabla g_i(x^{\theta_{i,k}}) + (x^k - x^{\theta_{i,k-1}})^T H^i_{\theta_{i,k}}]. \tag{11}$$

In following analysis, we will not distinguish these two forms of PROXTONE from each other.

Algorithm 2. PROXTONE in a form that is easy to analyze

```
**Input**: start point $x^0 \in \operatorname{dom} f$; for $i \in \{1,2,..,n\}$, let $g^0_i(x) = g_i(x^0) + (x - x^0)^T \nabla g_i(x^0) + \frac{1}{2}(x - x^0)^T H^i_0 (x - x^0)$, where the notation $H^i_0$ $(i \in \{1,2,..,n\})$ are totally the same as they in Algorithm 1; and $G^0(x) = \frac{1}{n}\sum_{i=1}^{n} g^0_i(x)$.
**repeat**
    Solve the subproblem for new approximation of the solution:
        $$x^{k+1} \leftarrow \arg\min_x \big[G^k(x) + h(x)\big]. \tag{12}$$
    Sample $j$ from $\{1,2,..,n\}$, and update the quadratic models or surrogate functions:
        $$g^{k+1}_j(x) = g_j(x^{k+1}) + (x - x^{k+1})^T \nabla g_j(x^{k+1}) + \frac{1}{2}(x - x^{k+1})^T H^i_{k+1}(x - x^{k+1}), \tag{13}$$
    while leaving all other $g^{k+1}_i(x)$ unchanged: $g^{k+1}_i(x) \leftarrow g^k_i(x)$ $(i \neq j)$; and $G^{k+1}(x) = \frac{1}{n}\sum_{i=1}^{n} g^{k+1}_i(x)$.
**until** stopping conditions are satisfied.
**Output**: $x^k$.
```

To better understand this method, we make the following illustration and observations.

2.1 The Regularized Quadratic Model in Algorithm 2

For fixed $x \in \mathbb{R}^p$, we define a regularized piecewise quadratic approximation of $f(x)$ as follows:

$$G^k(x) + h(x) = \frac{1}{n}\sum_{i=1}^{n} g^k_i(x) + h(x)$$

where $g^k_i(x)$ is the quadratic model for $g_i(x)$

$$\begin{aligned} &g^k_i(x)\\ =&g_i(x^{\theta_{i,k}}) + (x - x^{\theta_{i,k}})^T \nabla g_i(x^{\theta_{i,k}}) + \frac{1}{2}(x - x^{\theta_{i,k}})^T H^i_{\theta_{i,k}}(x - x^{\theta_{i,k}}), \end{aligned} \tag{14}$$

here $\theta_{i,k}$ is a random variable which have the following conditional probability distribution in each iteration:

$$\mathbb{P}(\theta_{i,k} = k|j) = \frac{1}{n} \quad \text{and} \quad \mathbb{P}(\theta_{i,k} = \theta_{i,k-1}|j) = 1 - \frac{1}{n}, \tag{15}$$

and $H^i_{\theta_{i,k}}$ is any positive definite matrix, which possibly depends on $x^{\theta_{i,k}}$. Then at each iteration the search direction is found by solving the subproblem (12).

One of the crucial ideas of this algorithm is that the component function to be used for updating the search direction at each iteration is chosen randomly. This allows the function to be selected very quickly. After the component function $g_j(x)$ selected and updated by (13), while leaving all other $g_i^{k+1}(x)$ unchanged.

2.2 The Hessian Approximation

Arguably, the most important feature of this method is that the regularized quadratic model (12) incorporates second order information in the form of a positive definite matrix H^i_k. This is key because, at each iteration, the user has complete freedom over the choice of H^i_k. A few suggestions for the choice of H^i_k include: the simplest option is $H^i_k = I$ that no second order information is employed; $H^i_k = \nabla^2 g_i(x_k)$ provides the most accurate second order information, but it is (potentially) more computationally expensive to work with.

3 Convergence Analysis

In this section we provide convergence theory for the PROXTONE algorithm. Under the standard assumptions, we now state our convergence result.

Theorem 1. *Suppose $\nabla g_i(x)$ is Lipschitz continuous with constant $L_i > 0$ for $i = 1, ..., n$, and $L_i I \preceq mI \preceq H^i_k \preceq MI$ for all $i = 1, ..., n$, $k \geq 1$. $h(x)$ is strongly convex with $\mu_h \geq 0$. Let $L_{max} = \{L_1, ..., L_n\}$, then the PROXTONE iterations satisfy for $k \geq 1$:*

$$\mathbb{E}[f(x^k)] - f^* \leq \frac{M + L_{max}}{2}[\frac{1}{n}\frac{M + L_{max}}{2\mu_h + m} + (1 - \frac{1}{n})]^k \|x^* - x^0\|^2. \tag{16}$$

The ideas of the proof is closed related to that of MISO by Mairal [3] and for completeness we give a simple version in the appendix.

We have the following remarks regarding the above result:

- In order to satisfy $\mathbb{E}[f(x^k)] - f^* \leq \epsilon$, the number of iterations k needs to satisfy
$$k \geq (\log \rho)^{-1} \log\big[\frac{2\epsilon}{(M + L_{max})\|x^* - x^0\|^2}\big],$$
where $\rho = \frac{1}{n}\frac{M+L_{max}}{2\mu_h+m} + (1 - \frac{1}{n})$.
- Inequality (16) gives us a reliable stopping criterion for the PROXTONE method.

At this moment, we see that the expected quality of the output of PROXTONE is good. However, in practice we are not going to run this method many times on the same problem. What is the probability that our single run can give us also a good result. Since $f(x^k) - f^* \geq 0$, Markov's inequality and Theorem 1 imply that for any $\epsilon > 0$,

$$\text{Prob}\Big(f(x^k) - f^* \geq \epsilon\Big) \;\leq\; \frac{\text{E}[f(x^k) - f^*]}{\epsilon} \;\leq\; \frac{(M + L_{max})\rho^k \|x^* - x^0\|^2}{2\epsilon}.$$

Thus we have the following high-probability bound.

Corollary 1. *Suppose the assumptions in Theorem 1 hold. Then for any $\epsilon > 0$ and $\delta \in (0,1)$, we have*

$$\text{Prob}\big(f(x^k) - f(x^\star) \leq \epsilon\big) \geq 1 - \delta,$$

provided that the number of iterations k satisfies

$$k \geq \log\left(\frac{(M + L_{max})\|x^* - x^0\|^2}{2\delta\epsilon}\right) \Big/ \log\left(\frac{1}{\rho}\right).$$

Based on Theorem 1 and its proof, we give a deeper and stronger result that the PROXTONE achieves a linear convergence rate for the solution.

Theorem 2. *Suppose $\nabla g_i(x)$ and $\nabla^2 g_i$ are Lipschitz continuous with constant $L_i > 0$ and $K_i > 0$ respectively for $i = 1, ..., n$, $h(x)$ is strongly convex with $\mu_h \geq 0$. Let $L_{max} = \{L_1, ..., L_n\}$ and $K_{max} = (1/n)\sum_{i=1}^n L_i$. If $H^i_{\theta_{i,k}} = \nabla^2 g_i(x^{\theta_{i,k}})$ and $L_i I \preceq mI \preceq H^i_k \preceq MI$, then PROXTONE converges exponentially to $x^\star$ in expectation:*

$$\begin{aligned}&\mathbb{E}[\|x^{k+1} - x^\star\|] \\ \leq& \Big(\frac{K_{max} + 2L_{max}}{m}\frac{M + L_{max}}{2\mu_h + m} + \frac{2L_{max}}{m}\Big)\Big[\frac{1}{n}\frac{M + L_{max}}{2\mu_h + m} + (1 - \frac{1}{n})\Big]^{k-1}\|x^* - x^0\|^2.\end{aligned}$$

In order to satisfy $\mathbb{E}[\|x^{k+1} - x^\star\|] \leq \epsilon$, the number of iterations k needs to satisfy

$$k \geq (\log\rho)^{-1}\log\Big[\frac{\epsilon}{C\|x^* - x^0\|^2}\Big],$$

where ρ is as before and $C = \frac{K_{max}+2L_{max}}{m}\frac{M+L_{max}}{2\mu_h+m} + \frac{2L_{max}}{m}$.

Due to the Markov's inequality, Theorem 2 implies the following result.

Corollary 2. *Suppose the assumptions in Theorem 2 hold. Then for any $\epsilon > 0$ and $\delta \in (0,1)$, we have*

$$\text{Prob}\big(\|x^{k+1} - x^\star\| \geq \epsilon\big) \geq 1 - \delta,$$

provided that the number of iterations k satisfies

$$k \geq \log\left(\frac{((K_{max} + 2L_{max})(M + L_{max}) + 2L_{max}(2\mu_h + m))\|x^* - x^0\|^2}{m(2\mu_h + m)\delta\epsilon}\right) \Big/ \log\left(\frac{1}{\rho}\right).$$

4 Numerical Experiments

The technique proposed in this paper has wide applications, it can be used to do least-squares regression, the Lasso, the elastic net, and the logistic regression. Furthermore the principle of PROXTONE can also be applies to do nonconvex optimization problems, such as training of deep convolutional network and so on.

In this section we present the results of some numerical experiments to illustrate the properties of the PROXTONE method. We focus on the sparse regularized logistic regression problem for binary classification: given a set of training examples $(a_1, b_1), \ldots, (a_n, b_n)$ where $a_i \in \mathbb{R}^p$ and $b_i \in \{+1, -1\}$, we find the optimal predictor $x \in \mathbb{R}^p$ by solving

$$\min_{x \in \mathbb{R}^p} \quad \frac{1}{n} \sum_{i=1}^{n} \log\bigl(1 + \exp(-b_i a_i^T x)\bigr) + \lambda_1 \|x\|_2^2 + \lambda_2 \|x\|_1,$$

where λ_1 and λ_2 are two regularization parameters. We set

$$g_i(x) = \log(1 + \exp(-b_i a_i^T x) + \lambda_1 \|x\|_2^2, \qquad h(x) = \lambda_2 \|x\|_1, \tag{17}$$

and

$$\lambda_1 = 1E - 4, \qquad \lambda_2 = 1E - 4.$$

In this situation, the subproblem (12) become a lasso problem, which can be effectively and accurately solved by the proximal algorithms [6].

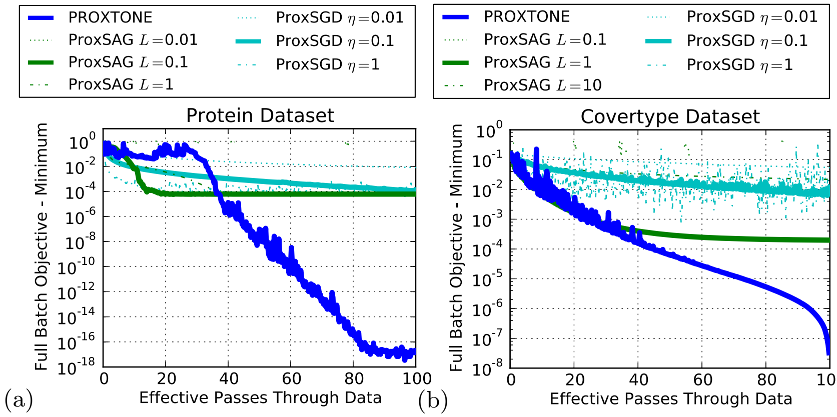

Fig. 1. A comparison of PROXTONE to competing optimization techniques for two datasets. The bold lines indicate the best performing hyperparameter for each optimizer.

We used some publicly available data sets. The *protein* data set was obtained from the KDD Cup 2004[1]; the covertype data sets were obtained from the LIBSVM Data[2].

[1] http://osmot.cs.cornell.edu/kddcup

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

The performance of PROXTONE is compared with some related algorithms:

- ProxSGD: We used a constant step size that gave the best performance among all powers of 10.
- ProxSAG: This is a proximal version of the SAG method, with the trailing number providing the Lipschitz constant.

The results of the different methods are plotted for the first 100 effective passes through the data in Figure 1. The PROXTONE iterations seem to achieve the best of all.

5 Conclusions

This paper introduces a proximal stochastic method called PROXTONE for minimizing regularized finite sums. For nonsmooth and strongly convex problems, we show that PROXTONE not only enjoys the same linear rates as those of MISO, SAG, ProxSVRG and ProxSDCA, but also showed that the *solution* of this method converges in exponential rate too. There are some directions that the current study can be extended. In this paper, we have focused on the theory and the convex experiments of PROXTONE; it would be meaningful to also make clear the implementation details and do the numerical evaluation to nonconvex problems [10]. Second, combine with randomized block coordinate method [4] for minimizing regularized convex functions with a huge number of varialbes/coordinates. Moreover, due to the trends and needs of big data, we are designing distributed/parallel PROXTONE for real life applications. In a broader context, we believe that the current paper could serve as a basis for examining the method on the proximal stochastic methods that employ second order information.

Appendix

In this Appendix, we give the proofs of the two propositions.

A Proof of Theorem 1

Since in each iteration of the PROXTONE, we have (14) and (15), that yields

$$\mathbb{E}[\|x^* - x^{\theta_{i,k}}\|^2] = \frac{1}{n}\mathbb{E}[\|x^* - x^k\|^2] + (1 - \frac{1}{n})\mathbb{E}[\|x^* - x^{\theta_{i,k-1}}\|^2]. \tag{18}$$

Since $0 \preceq H^i_{\theta_{i,k}} \preceq MI$ and $\nabla^2 g^k_i(x) = H^i_{\theta_{i,k}}$, by Theorem 2.1.6 of [5] and the assumption, $\nabla g^k_i(x)$ and $\nabla g_i(x)$ are Lipschitz continuous with constant M and L_i respectively, and further $\nabla g^k_i(x) - \nabla g_i(x)$ is Lipschitz continuous with constant $M + L_i$ for $i = 1, \dots, n$. This together with (7) yieds

$$|[g^k_i(x) - g_i(x)] - [g^k_i(y) - g_i(y)] - \nabla[g^k_i(y) - g_i(y)]^T(x - y)| \le \frac{M + L_i}{2}\|x - y\|^2.$$

Applying the above inequality with $y = x^{\theta_{i,k}}$, and using the fact that $\nabla[g_i^k(x^{\theta_{i,k}})] = \nabla[g_i(x^{\theta_{i,k}})]$ and $g_i^k(x^{\theta_{i,k}}) = g_i(x^{\theta_{i,k}})$, we have

$$|g_i^k(x) - g_i(x)| \leq \frac{M + L_i}{2}\|x - x^{\theta_{i,k}}\|^2.$$

Summing over $i = 1, \ldots, n$ yields

$$[G^k(x) + h(x)] - [g(x) + h(x)] \leq \frac{1}{n}\sum_{i=1}^{n}\frac{M + L_i}{2}\|x - x^{\theta_{i,k}}\|^2. \tag{19}$$

Then by the Lipschitz continuity of $\nabla g_i(x)$ and the assumption $L_i I \preceq mI \preceq H_k^i$, we have

$$\begin{aligned} &g_i(x) \\ &\quad\leq g_i(x^{\theta_{i,k}}) + \nabla g_i(x^{\theta_{i,k}})^T(x - x^{\theta_{i,k}})| + \frac{L_i}{2}\|x - x^{\theta_{i,k}}\|^2 \\ &\quad\leq g_i(x^{\theta_{i,k}}) + (x - x^{\theta_{i,k}})^T\nabla g_i(x^{\theta_{i,k}}) + \frac{1}{2}(x - x^{\theta_{i,k}})^T H_{\theta_{i,k}}^i(x - x^{\theta_{i,k}}) = g_i^k(x), \end{aligned}$$

and thus, by summing over i yields $g(x) \leq G^k(x)$, and further by the optimality of x^{k+1}, we have

$$\begin{aligned} f(x^{k+1}) &\leq G^k(x^{k+1}) + h(x^{k+1}) \leq G^k(x) + h(x) \\ &\leq f(x) + \frac{1}{n}\sum_{i=1}^{n}\frac{M + L_i}{2}\|x - x^{\theta_{i,k}}\|^2 \end{aligned} \tag{20}$$

Since $mI \preceq H_{\theta_{i,k}}$ and $\nabla^2 g_i^k(x) = H_{\theta_{i,k}}$, by Theorem 2.1.11 of [5], $g_i^k(x)$ is m-strongly convex. Since $G^k(x)$ is the average of $g_i^k(x)$, thus $G^k(x) + h(x)$ is $(m + \mu_h)$-strongly convex, we have

$$\begin{aligned} f(x^{k+1}) + \frac{m + \mu_h}{2}\|x - x^{k+1}\|^2 &\leq G^k(x^{k+1}) + h(x^{k+1}) + \frac{m + \mu_h}{2}\|x - x^{k+1}\|^2 \\ &\leq G^k(x) + h(x) \\ &= f(x) + [G^k(x) + h(x) - f(x)] \\ &\leq f(x) + \frac{1}{n}\sum_{i=1}^{n}\frac{M + L_i}{2}\|x - x^{\theta_{i,k}}\|^2. \end{aligned}$$

By taking the expectation of both sides and let $x = x^*$ yields

$$\mathbb{E}[f(x^{k+1})] - f^* \leq \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\frac{M + L_i}{2}\|x^* - x^{\theta_{i,k}}\|^2] - \mathbb{E}[\frac{m + \mu_h}{2}\|x^* - x^{k+1}\|^2].$$

We have

$$\begin{aligned} &\frac{\mu_h}{2}\|x^{k+1} - x^*\|^2 \leq \mathbb{E}[f(x^{k+1})] - f^* \\ &\qquad\qquad \leq \mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\frac{M + L_{max}}{2}\|x - x^{\theta_{i,k}}\|^2] - \mathbb{E}[\frac{m + \mu_h}{2}\|x - x^{k+1}\|^2]. \end{aligned}$$

thus

$$\|x^{k+1}-x^*\|^2 \le \frac{M+L_{max}}{2\mu_h+m}\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^*-x^{\theta_{i,k}}\|^2]. \tag{21}$$

then we have

$$\begin{aligned}\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^*-x^{\theta_{i,k}}\|^2] &= \frac{1}{n}\|x^k-x^*\|^2+(1-\frac{1}{n})\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^*-x^{\theta_{i,k-1}}\|^2]\\ &\le \frac{1}{n}\|x^k-x^*\|^2+(1-\frac{1}{n})\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^*-x^{\theta_{i,k-1}}\|^2]\\ &\le [\frac{1}{n}\frac{M+L_{max}}{2\mu_h+m}+(1-\frac{1}{n})]\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^*-x^{\theta_{i,k-1}}\|^2]\\ &\le [\frac{1}{n}\frac{M+L_{max}}{2\mu_h+m}+(1-\frac{1}{n})]^k\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^*-x^{\theta_{i,0}}\|^2]\\ &\le [\frac{1}{n}\frac{M+L_{max}}{2\mu_h+m}+(1-\frac{1}{n})]^k\|x^*-x^0\|^2.\end{aligned}$$

Thus we have $\mathbb{E}[f(x^{k+1})]-f^* \le \frac{M+L_{max}}{2}[\frac{1}{n}\frac{M+L_{max}}{2\mu_h+m}+(1-\frac{1}{n})]^k\|x^*-x^0\|^2$.

B Proof of Theorem 2

We first examine the relations between the search directions of ProxN and PROXTONE.

By (4), (5) and Fermat's rule, Δx^k_{ProxN} and Δx^k are also the solutions to

$$\begin{aligned}\Delta x^k_{ProxN} &= \arg\min_{d\in\mathbb{R}^p}\nabla g(x^k)^T d+(\Delta x^k_{ProxN})^T H_k d+h(x^k+d),\\ \Delta x^k &= \arg\min_{d\in\mathbb{R}^p}(\nabla_k+H_k x^k)^T d+(\Delta x^k)^T H_k d+h(x^k+d).\end{aligned}$$

Hence Δx^k and Δx^k_{ProxN} satisfy

$$\begin{aligned}&\nabla g(x^k)^T\Delta x^k+(\Delta x^k_{ProxN})^T H_k\Delta x^k+h(x^k+\Delta x^k)\\ \ge&\nabla g(x^k)^T\Delta x^k_{ProxN}+(\Delta x^k_{ProxN})^T H_k\Delta x^k_{ProxN}+h(x^k+\Delta x^k_{ProxN})\end{aligned}$$

and

$$\begin{aligned}&(\nabla_k+H_k x^k)^T\Delta x^k_{ProxN}+(\Delta x^k)^T H_k\Delta x^k_{ProxN}+h(x^k+\Delta x^k_{ProxN})\\ &\quad\ge(\nabla_k+H_k x^k)^T\Delta x^k+(\Delta x^k)^T H_k\Delta x^k+h(x^k+\Delta x^k).\end{aligned}$$

We sum these two inequalities and rearrange to obtain

$$\begin{aligned}&(\Delta x^k)^T H_k\Delta x^k-2(\Delta x^k_{ProxN})^T H_k\Delta x^k+(\Delta x^k_{ProxN})^T H_k\Delta x^k_{ProxN}\\ &\quad\le(\nabla_k+H_k x^k-\nabla g(x^k))^T(\Delta x^k_{ProxN}-\Delta x^k).\end{aligned}$$

The assumptions $mI \preceq H_{\theta_{i,k}}$ yields that $mI \preceq H_k$, together with (11) we have

$$\begin{aligned} & m\|\Delta x^k - \Delta x^k_{ProxN}\|^2 \\ & \quad \leq \|\frac{1}{n}\sum_{i=1}^{n}(\nabla g_i(x^{\theta_{i,k}}) - \nabla g_i(x^k) - (x^{\theta_{i,k}} - x^k)^T H^i_{\theta_{i,k}})\|\|(\Delta x^k - \Delta x^k_{ProxN})\|. \end{aligned} \tag{22}$$

Since $\nabla^2 g_i$ is Lipschitz continuous with constant $K_i > 0$, by Lemma 1.2.4 of [5] we have

$$\|\nabla g_i(x^{\theta_{i,k}}) - \nabla g_i(x^k) - (x^{\theta_{i,k}} - x^k)^T H^i_{\theta_{i,k}}\| \leq \frac{K_i}{2}\|x^{\theta_{i,k}} - x^k\|^2. \tag{23}$$

Then from (22) and (23), we have

$$\|\Delta x^k - \Delta x^k_{ProxN}\| \leq \frac{K_{max}}{2mn}\sum_{i=1}^{n}\|x^{\theta_{i,k-1}} - x^k\|^2. \tag{24}$$

Since the ProxN method converges q-quadratically (cf. Theorem 3.3 of [2]),

$$\begin{aligned} & \|x^{k+1} - x^\star\| \\ & \quad \leq \|x^k + \Delta x^k_{ProxN} - x^\star\| + \|\Delta x^k - \Delta x^k_{ProxN}\| \\ & \quad \leq \frac{K_{max}}{m}\|x^k - x^\star\|^2 + \|\Delta x^k - \Delta x^k_{ProxN}\|. \end{aligned} \tag{25}$$

Thus from (24) and (25), we have almost surely that

$$\begin{aligned} & \|x^{k+1} - x^\star\| \\ & \quad \leq \frac{K_{max}}{m}\|x^k - x^\star\|^2 + \frac{L_{max}}{2mn}\sum_{i=1}^{n}\|x^{\theta_{i,k-1}} - x^k\|^2 \\ & \quad \leq \frac{K_{max}}{m}\|x^k - x^\star\|^2 + \frac{L_{max}}{mn}\sum_{i=1}^{n}2\|x^{\theta_{i,k-1}} - x^*\|^2 + \frac{L_{max}}{mn}\sum_{i=1}^{n}2\|x^* - x^k\|^2. \end{aligned}$$

Then by (21), we have

$$\|x^{k+1} - x^\star\| \leq (\frac{K_{max} + 2L_{max}}{m}\frac{M + L_{max}}{2\mu_h + m} + \frac{2L_{max}}{m})\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\|x^{\theta_{i,k}} - x^*\|^2]$$

which yieds

$$\begin{aligned} \|x^{k+1} - x^\star\| \leq (\frac{K_{max} + 2L_{max}}{m}\frac{M + L_{max}}{2\mu_h + m} + \frac{2L_{max}}{m})[\frac{1}{n}\frac{M + L_{max}}{2\mu_h + m} \\ + (1 - \frac{1}{n})]^k\|x^* - x^0\|^2. \end{aligned}$$

References

1. Bertsekas, D.P.: Incremental gradient, subgradient, and proximal methods for convex optimization: a survey. Optimization for Machine Learning 2010, 1–38 (2011)
2. Lee, J., Sun, Y., Saunders, M.: Proximal newton-type methods for convex optimization. In: Advances in Neural Information Processing Systems, pp. 836–844 (2012)
3. Mairal, J.: Optimization with first-order surrogate functions. arXiv preprint arXiv:1305.3120 (2013)
4. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization **22**(2), 341–362 (2012)
5. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer, Boston (2004)
6. Parikh, N., Boyd, S.: Proximal algorithms. Foundations and Trends in Optimization **1**(3), 123–231 (2013)
7. Schmidt, M., Roux, N.L., Bach, F.: Minimizing finite sums with the stochastic average gradient. arXiv preprint arXiv:1309.2388 (2013)
8. Shalev-Shwartz, S., Zhang, T.: Proximal stochastic dual coordinate ascent. arXiv preprint arXiv:1211.2717 (2012)
9. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss. The Journal of Machine Learning Research **14**(1), 567–599 (2013)
10. Sohl-Dickstein, J., Poole, B., Ganguli, S.: Fast large-scale optimization by unifying stochastic gradient and quasi-newton methods. In: Proceedings of the 31st International Conference on Machine Learning (ICML 2014), pp. 604–612 (2014)
11. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. arXiv preprint arXiv:1403.4699 (2014)