Stochastic Quasi Newton methods

Duc Nguyen

I. Abstract

Stochastic quasi-newton method are second order optimization method that The following report will look

II. Literature reviews

A. Stochastic Optimization

Stochastic methods have gained traction as the method of choice in solving large-scale optimization problems in machine learning and scientific computing. In stochastic optimization, we look at the following composite objective function

$$f(w) = \frac{1}{N}\sum_{i=1}^{N} f_i(w)$$

For the purpose of our convergence analysis, we will assume that each function f_i is strongly convex. Most of the methods that are widely used in practice are variants of stochastic first order method or stochastic gradient descent. Vanilla stochastic gradient descent can be summarized as:

Algorithm 1 Stochastic Gradient Descent

```
For k = 1, 2, ...
    **Randomly select** $i_k$ between 1 and $N$
    Update $w^{k+1} = w^k - \alpha^k \nabla f_{i_k}(w^k)$
```

Due to the random nature of the algorithm, the convergence analysis of stochastic gradient descent shows that with a constant step size α^k, the algorithm converges linearly to a local neighborhood around x^* but not x^* itself. With a decreasing step-size, the algorithm converges sublinearly to x^*.

B. SVRG

Stochastic Variance Reduced Gradient builds on vanilla SGD and incorporates full gradient information to obtain a better estimates that allows linear convergence with constant step-size. The algorithm makes a tradeoff between per iteration computation cost and overall convergence rate. SVRG can be summarized as follows:

Algorithm 2 Stochastic Variance Reduced Gradient Descent (SVRG)

```
For k = 1, 2, ...
    Choose $x^0 = w^k$
    For j = 1, 2,..., m
        Randomly choose $i$ in $1, ..., N$
        $x^j = x^{j-1} - \alpha[\nabla f_i(x^{j-1}) - \nabla f_i(w^k) + \nabla f(w^k)]$
    $w^{k+1} = z^m$
```

In fact, most state of the art methods have been first order methods as well as their improvements and modifications. In recent years, there have been a push towards stochastic second order methods. Second methods, which make use of second order information about the objective function, can potentially yield better convergence rate than first order method.

Among second order methods, Newton's method is the algorithm with the best theoretical convergence rate. However, computing the Hessian matrix in large scale problem is intractable and computing the hessian inverse vector product is another bottle neck of the algorithm. BFGS is a class of second order method invented to circumvent this expensive computation. However, it has to store a $n \times n$ Hessian approximations which is not appropriate for problems in high dimensions. Its limited memory variant L-BFGS is often used instead thanks to its linear memory usage.

C. Sublinearly convergent L-BFGS

The following algorithm outlines the steps in a Stochastic Quasi-newton algorithm with sub-linear convergence.

Algorithm 3 Sublinear convergent- SQN M constraint on memory storage; b size of random subset to approximate gradient; b_H size of subset to approximate hessian

```
Initialize r = 0 (number of corrections), $H_0 = I$
For k = 1, 2, ...
    Choose a random subset $S$ of size $b$ and compute $\nabla f_S(w^k)$
    if $r < 1$ do a gradient descent update
    else
        $w^{k+1} = w^k - \alpha^k H_r \nabla f_S(x)$
    if $mod(k, L) == 0$
        Compute $\bar{w}_r = \frac{1}{L}\sum_{i=k-L+1}^{k} w^i$
        If $r > 1$
            Choose random subset of size $b_H$ to compute $\nabla^2 f_{S_H}(\bar{w}_r)$
            $s_r = \bar{w}_r - \bar{w}_{r-1}$
            $y_r = \nabla f_{S_H}(\bar{w}_r) s_r$
            Compute limited memory $H_r$
```

D. SVRG-L-BFGS

The following algorithm outlines the steps in a linearly convergent stochastic quasi newton that builds on the idea of SVRG. Note that this algorithm looks very similar to the stochastic quasi-newton algorithm outlined above. However, the main difference is the inclusion of the SVRG step that computes the variance reduced gradient instead of the simple average stochastic gradient in the sublinearly convergent L-BFGS. We will see in convergence analysis that this allows the algorithm to have linear convergence rate with a constant step size while keeping the same computational complexity per iteration as the sublinearly stochastic quasi-newton method outlined in the previous subsection.

Algorithm 4 linearly convergent SQN M constraint on memory storage; b size of random subset to approximate gradient; b_H size of subset to approximate hessian; η constant step size

```
Initialize r = 0 (number of corrections), $H_0 = I$
For k = 1, 2, ...
    Compute full gradient $\mu_k = \triangledown f(w_k)$
    Set $x_0 = w_k$
    For t = 0, ... , m-1
        Choose a random subset $S$ of size $b$ and compute $\triangledown f_S(x_t)$
        Compute reduced variance gradient $v_t = \triangledown f_S(x_t) - \triangledown f_S(w_k) + \mu_k$
        Update $x_{t+1} = x_t - \eta H_t v_t$
        if $\mod(t, L) == 0$
            r++
            $\bar{x}_r = \frac{1}{L} \sum_{i=t-L+1}^{t} x_i$
            Choose random subset of size $b_H$ to compute $\triangledown^2 f_{S_H}(\bar{x}_r)$
            $s_r = \bar{x}_r - \bar{x}_{r-1}$
            $y_r = \triangledown f_{S_H}(\bar{x}_r) s_r$
            Compute limited memory $H_r$
    Choose $w^{k+1} = x_{m-1}$
```

III. Convergence Analysis

In this section, we will look at the convergence analysis of the two algorithms and see the improvements made introduced by SVRG.

A. Sublinearly convergent L-BFGS

The analysis for the sublinearly convergent L-BFGS assumes that each function f_i is convex but doesn't have to be strongly convex. s

B. Linearly convergent L-BFGS

Assumption 1. The function f_i is convex and twice continuously differentiable for all i. f can be made strongly convex by adding a regularizer. Hence, the analysis assumes that f is strongly convex

Assumption 2. There exist positive constants λ and Λ such that $\lambda I \leq \nabla^2 f_S(w) \leq \Lambda I$ for all i.

From these assumptions, we can derive the following lemmas:

Lemma 1. Given $B_r = H_r^{-1}$, then for some constants d and M, the following holds:

$$tr(B_r) \leq (d+M)\Lambda$$
$$det(B_r) \geq \frac{\lambda^{d+M}}{((d+M)\Lambda)^M}$$

Lemma 2. There exists constants $0 < \gamma < \Gamma$ such that $\gamma I \preceq H_r \preceq \Gamma I$

where

$$\gamma = \frac{1}{(d+M)\Lambda} \qquad \Gamma = \frac{((d+M)\Lambda)^{d+M-1}}{\lambda^{d+M}}$$

Lemma 3. $\|\nabla f(x)\|^2 \geq 2\lambda(f(x) - f(w^*)$. This follows naturally from the strong convexity assumption on f.

Lemma 4. With $v_t = \nabla f_S(x_t) - \nabla f_S(w_k) + \nabla f(w_k)$, then:

$$E_{k,t}[\|v_t\|^2] \leq 4\Lambda(f(x_t) - f(w^*) + f(w_k) - f(w^*))$$

Conclusion from these lemmas and assumptions, the main result shows that:

$$\mathbb{E}[f(w) - f(w^*)] \leq C^k \, \mathbb{E}[f(w_0) - f(w^*)]$$

where the convergence rate C is given as

$$\frac{1/(2m\eta) + \eta\Gamma\Lambda^2}{\gamma\lambda - \eta\Gamma^2\Lambda^2} < 1$$

IV. Numerical Experiments

For our experiments we look at the following composite objective function:

$$F(w) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}w^T A_i w - b_i^T w$$

where A_i is a symmetric and positive definite matrix. The codes to generate the objective function were included and modifiable with different choices of N and n. Hence, this is a strongly convex objective function and all sub functions are also strongly convex.

V. Results

VI. Discussion

VII. Conclusion

References

[1] R. H. Byrd, S.L. Hansen, Jorge Nocedal, Y. Singer *A stochastic quasi-newton method for large scale optimization*.

[2] Philip Moritz, Robert Nishihara, Mihael I.Jordan *A linearly convergent stochastic L-BFGS algorithm*

VIII. Appendix

A. Proof for Lemma

B. Proof for Lemma

C. Proof for Lemma

D. Proof for Lemma