Stochastic Proximal Gradient Algorithm and it's application to sum of least squares

Aniket Anand Deshmukh
Department of EECS
University of Michigan
Ann Arbor, MI 48105
aniketde@umich.edu

Abstract

In this project, I explore and implement a class of iterative gradient algorithms to solve a minimization of sum of two functions. These methods can be viewed as an extension of the classical gradient algorithm and are attractive due to their adequateness for solving large-scale problems. I implemented the fast iterative shrinkage-thresholding algorithm (FISTA) which is proven to be significantly better than classical gradient algorithms, both theoretically and practically [1]. But for very large scale problem this method can be slow. So I further implemented a stochastic proximal gradient algorithm (SPGA) where gradient is approximated using monte carlo techniques[2]. This algorithm is proven to converge under some conditions with the same convergence rate as their deterministic counterpart. To illustrate, I implement this algorithm for sum of least squares problem which is motivated from the real optimization challenge.

1 Introduction

In this project I deal with the optimization problem of the form

$$\min_x F(x) = f(x) + g(x) \tag{1}$$

where f is a smooth function and g is a possibly non-smooth convex penalty term. We come across such optimization problem in wide range of applications such as signal and image processing, controls, statistical inference, etc. It is important to solve these problems in real time. There have been lot of literature in past few years for solving related problems [3,4,5,6,7,8]. The goal of this project is to make such a problem computationally tractable and I address one such a specific problem.

The rest of the report is organized as follows. Section 2 gives the general gradient methods and section 3 continues with a general problem statement and important concepts that are needed in the report. Section 4 describes the FISTA and section 5 describes the SPGA [2]. Section 6 describes the specific problem statement that I'm trying to address and synthetic data used for the experiments. Section 7 describes the experiments and results are discussed in section 8. Finally, I conclude in section 9.

2 Gradient Methods

Consider the unconstrained minimization problem of a continuously differentiable function $f : R^n \to R$:

$$\min_x \{f(x) : x \in R^n\}. \tag{2}$$

gradient algorithm is one of the simplest method that can solve such a problem.

$$x_o \in R^n, x_k = x_{k-1} - t_k \nabla f(x_{k-1}), \tag{3}$$

where $t_k \geq 0$ is a step size. This gradient iteration can be solved using proximal regularization [1], which is written as:

$$x_k = \underset{x}{argmin}\{f(x_{k-1}) + \langle x - x_{k-1}, \nabla f(x_{k-1})\rangle + \frac{1}{2t_k}\|x - x_{k-1}\|^2\} \tag{4}$$

Adopting the same idea for sum of 2 functions:

$$x_k = \underset{x}{argmin}\{f(x_{k-1}) + \langle x - x_{k-1}, \nabla f(x_{k-1})\rangle + \frac{1}{2t_k}\|x - x_{k-1}\|^2 \ + g(x)\} \tag{5}$$

In the case of $l-1$ norm this can be further reduced by ignoring the constant terms and it can be solved using shrinkage thresholding as in [1].

3 General Problem Statement and Concepts

Let's consider the general problem:

$$\min_x F(x) = f(x) + g(x) \tag{6}$$

Following assumptions made throughout this report:

- $g : R^n \to R$ is a continuous convex function which is possibly nonsmooth.
- $f : R^n \to R$ is a smooth convex function which is continuously differential with Lipschitz continuous gradient:

$$\|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\| \tag{7}$$

for every $x, y \in R^n$. Here $L(f)$ denote the Lipschitz constant of ∇f.

- Problem in 6 is solvable i.e. $X_* := argminF \neq \phi$ and for $x^* \in X_*$ set $F_* := F(x^*)$

Now, $F(x) = f(x) + g(x)$ can be approximated using:

$$Q_L(x, y) = f(y) + \langle x - y, \nabla f(y)\rangle + \frac{L}{2}\|x - y\|^2 + g(x) \tag{8}$$

which admits the unique minimizer:

$$\rho_L(y) = argmin\{Q_L(x, y) : x \in R^n\} \tag{9}$$

Ignoring constant terms optimization becomes:

$$\rho_L(y) = \underset{x}{argmin}\{g(x) + \frac{L}{2}\|x - (y - \frac{1}{L}\nabla f(y)\|^2\} \tag{10}$$

One can solve this using iterative soft thresholding, if $g(x)$ is $l-1$ norm, where each iterate is:

$$x_k = \rho_L(x_{k-1}) \tag{11}$$

3.1 Important Lemma

Few important theorems, and lemma are provided here without proof. One can see detailed proof here [1].

Lemma Let $f : R^n \to R$ be a continuously differential function with Lipschitz continuous gradient and Lipschitz constant $L(f)$. Then for any $L \geq L(f)$

$$f(x) \leq f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2}\|x - y\|^2 \tag{12}$$

for every $x, y \in R^n$.

Lemma For any $y \in R^n$, one has $z = \rho_L(y)$ if and only if there exists $\gamma(y) \in \delta g(z)$, the subdifferential of $g(\dot{})$, such that

$$\nabla f(y) + L(z - y) + \gamma(y) = 0 \tag{13}$$

Lemma Let $y \in R^n$ and $L \geq 0$ be such that

$$F(\rho_L(y)) \leq Q(\rho_L(y), y) \tag{14}$$

then for any $x \in R^n$,

$$F(x) - F(\rho_L(y)) \geq \frac{L}{2}\|\rho_L(y) - y\|^2 + L\langle y - x, \rho_L(y) - y \rangle \tag{15}$$

Because of above results, we can minimize Q_L instead of F as in equation 9. This is also a class of Majorize-minimize algorithm where we minimize some convex majorizer instead of actual function. Now we see a FISTA in the following section.

4 Fast Iterative Soft Thresholding

FISTA solves the general problem described by equation 6. **Theorem** Let $\{x_k\}, \{y_k\}$ be generated by following FISTA algorithm. Then for any $k \geq 1$

$$F(x) - F(x*) \leq \frac{2L(f)\|x_0 - x*\|^2}{(k+1)^2} \forall x^* \in X_* \tag{16}$$

For proof please see [1].

Algorithm 1 FISTA with constant stepsize

1: Input: $L = L(f)$ - A lipschitz constant of ∇f.
2: Step 0: Take $y_1 = x_0 \in R^n, t_1 = 1$
3: **for** $k \geq 1$ until convergence: **do**
4: $\quad x_k = \rho_L(y_k) = argmin_x\{g(x) + \frac{L}{2}\|x - (y_k - \frac{1}{L}\nabla f(y_k))\|^2\}$
5: $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
6: $\quad y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$
7: **end for**

5 Stochastic Proximal Gradient

Now, what if in the original problem (equation 6) ∇f becomes intractable? One source of intractability arises when huge datasets are handled. For example in the case of learning problem, f could be a loss function and in that case f can be written as $f = \sum_{i=1}^{N} f_i$ where N is a sample size. In this case, computation of f and ∇f can be a difficult task. But these computations can be reduced using Monte Carlo estimation of f and ∇f. This algorithm

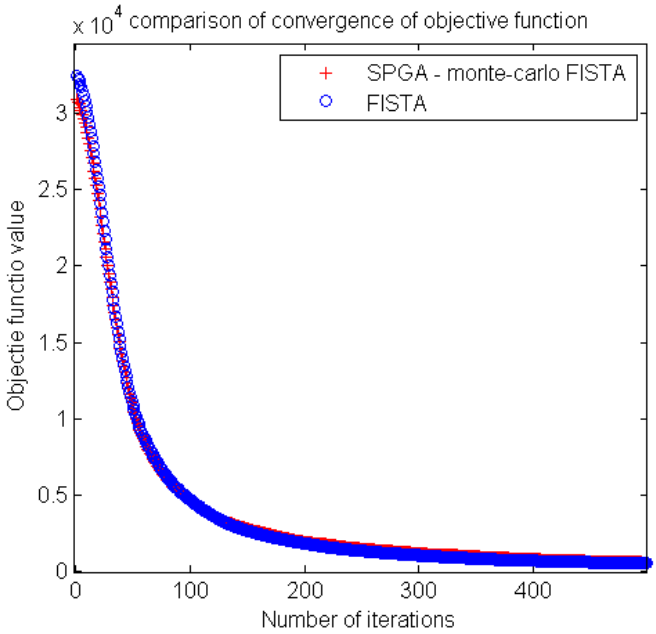

Figure 1: FISTA vs SPGA

is guaranteed to converge under some conditions [2]. The comparison of convergence between two algorithms (for the problem described in the next section) is shown the figure 1. The data is generated as explained in the section 7. For the purpose of this experiment $N = 500, L = 3, d = 3$, and $\lambda = 0.01$. Monte Carlo batch sizes were used as described in the datasets section. For FISTA, entire dataset is used to calculate the gradient. One can see that there is almost no difference between rate of convergence. Also, cost per iteration for SPGA is much lesser than that of FISTA.

Algorithm 2 SPGA constant stepsize

Input: $L = L(f)$ - A lipschitz constant of ∇f.
Step 0: Take $y_1 = x_0 \in R^n, t_1 = 1$
for $k \geq 1$ until convergence: **do**
    Choose Monte Carlo batch size
    Estimate $\hat{\nabla} f$ using above Monte Carlo batch
    $x_k = \rho_L(y_k) = argmin_x\{g(x) + \frac{L}{2}||x - (y_k - \frac{1}{L}\hat{\nabla} f(y_k))||^2\}$
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $y_{k+1} = x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1})$
end for

6 Application - Problem Statement

Given data: $\{p_n, q_n\}_{n=1}^N$, $p_n, q_n \in R^d$ and a function $\rho : R \to R$ Find w such that

$$\underset{w}{argmin}\, J(w) = \underset{w}{argmin} \sum_{j=1}^{N} ||q_j - \sum_{l=1}^{L}\sum_{k=1}^{N} w_{kl} \nabla_{p_k} \rho(\frac{||p_j - p_k||}{s_l})||^2 \tag{17}$$

where $\rho(z) = \frac{1}{1+||z||^m}, m > d$.

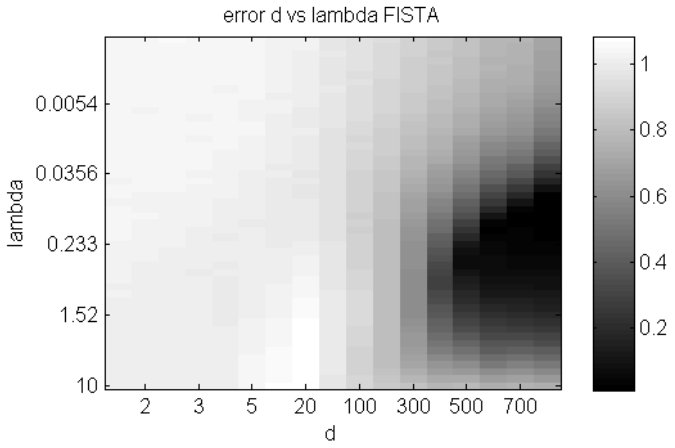

Figure 2: FISTA

Let $w = [w_1, w_2, ..., w_{NL}]^T_{NL\times 1}$, $X_j = [X_{1j}, X_{2j}, ..., X_{NLj}]^T_{d\times NL}$, where $X_{ij} = \nabla_{p_k}\rho(\frac{\|p_j - p_k\|}{s_l})$ and $l = \lceil i/L \rceil$

So, optimization function is

$$\underset{w}{argmin}\, J(w) = \underset{w}{argmin} \sum_{j=1}^{N} \|q_j - X_j w\|^2 \tag{18}$$

We also want w to be sparse. So imposing sparsity condition to the above optimization function:

$$\underset{w}{argmin}\, J(w) = \underset{w}{argmin} \sum_{j=1}^{N} \|q_j - X_j w\|^2 + \lambda\|w\|_1 \tag{19}$$

where λ is a regularization parameter.

Using section 2 and section 3:

$$w_k = \underset{w}{argmin}\, J(w) = \underset{w}{argmin}\{g(w) + \frac{L}{2}\|w - (w_{k-1} - \frac{1}{L}\nabla f(w_{k-1}))\|^2\} \tag{20}$$

$$w_k = \underset{w}{argmin}\{\lambda|w|_1 + \frac{L}{2}\|w - (w_{k-1} - \frac{1}{L}\sum_{j=1}^{N}(-2X_j^T q_j + X_j^T X_j w_{k-1}))\|^2\} \tag{21}$$

Now this can be solved using either algorithm 1 or algorithm 2. Almost similar problems are solved in [9,10].

7 synthetic data and Experiments

- First, synthetic dataset to test FISTA is generated. Each element of X_j is drawn from $N(0, 1)$ independently. Then each column of X_j is normalized. Then w is generated from uniform distribution $[-10, 10]$. Then q is calculated using $X_j w + 0.1 * N(0, 1)$ and w is estimated using FISTA. For this experiment - $N = 1, L = 2000, w \in R^{2000}, X_j \in R^{d\times 2000}$. d and λ are varied in this experiment. Results are shown in the figure 2.

- X_j, w, and q_j are generated in same way as above. But now we vary L, d, and λ for one experiment and then LN, and d for the second one. Sparsity in w is kept at 10%. Monte Carlo batch sizes are defined as: for first iteration: $(N/log(N))$, increment in

each iteration: ($log(N)$) and maximum size: $N/2$. Results are shown in figure 3 and 4.

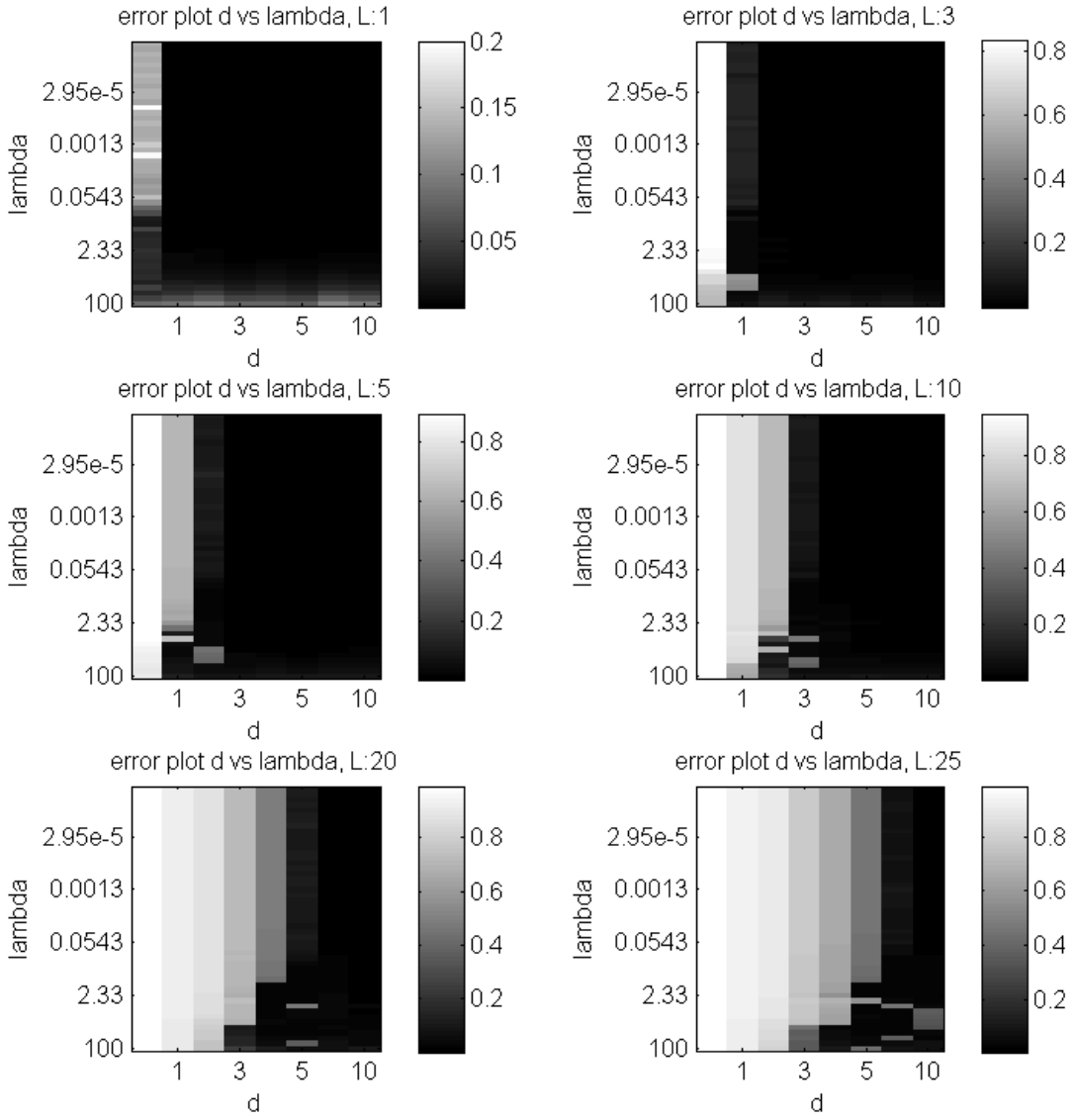

Figure 3: SPGA-different-L-d-λ

8 Results

From figure 1, one can see the convergence rate for both SPGA and it's deterministic counterpart is almost same. So, one can get huge speed ups using Monte Carlo techniques (for the setting of the problem addressed in this project). Figure 2 gives the result for FISTA (relative error). For very small d, relative error is almost one. This is because we have very less data (N = 1) and 2000 variables to estimate. But as d becomes greater than 600,

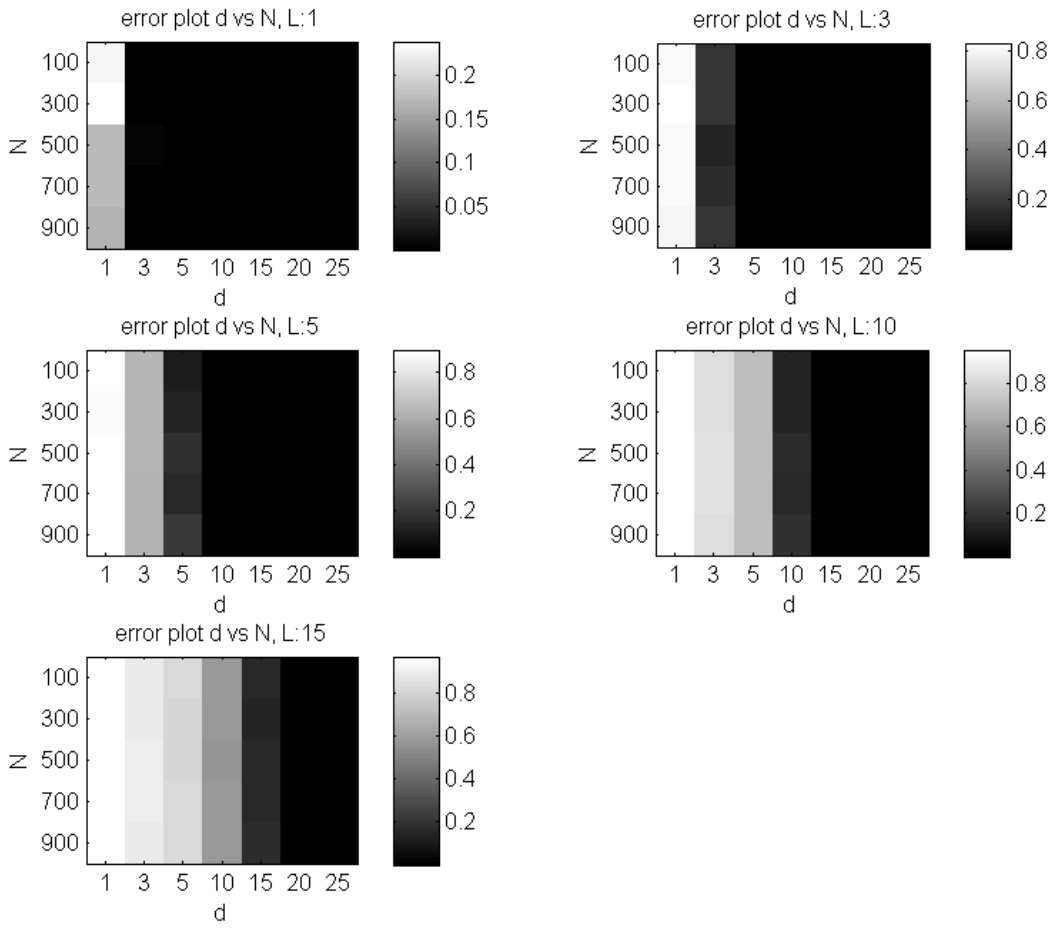

Figure 4: SPGA-different-L-d-N

we get error close to zero for some range of λ. So we need data (or number of equations) close to number of variables which we want to estimate. Figure 3 gives the relative error for different λ, d, and L. N is kept constant here at 100. As L increases number of variables increase (as $w \in R^{NL}$). It is evident from the figure 3 that as L increases we have more and more error for small d. As d increases number of equations/data points increase, so we get better results. One thing that looks surprising is λ doesn't have lot of effect in this case. Relative error is almost same for all λ. In figure 4, N, d, and L are varied. Again like λ, change in N doesn't have lot of effect. Though this is not surprising here because both variables and number of equations depend linearly on N. It is again evident that as L increases error increases.

9 Conclusions and future work

In this project proximal gradient methods have been studied. I implemented FISTA [1] and SPGA [2] and studied variuos aspects through experiments. In the problem that I'm addressing it looks like more d, less L is favorable from the accuracy point of view. My immediate next goal is to apply this to actual data (Experimenting with that will need more time, as it's huge). Recent paper [6] also looks promising for the problems that I'm trying to

address and I will incorporate results for that too. The analysis from synthetic data will be useful for experimenting with actual data but there could be changes as actual data is more structured and there is some pattern (one can observe that from equation 17). This project helped me in exploring the stochastic optimization techniques which is really helpful in my research - "Scalabale machine learning algorithms". This direction of research is exciting and looks like there are many concepts I can pick up and develop in this area.

References

[1] Beck, Amir, and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." SIAM Journal on Imaging Sciences 2, no. 1 (2009): 183-202.

[2] Atchade, Yves F., Gersende Fort, and Eric Moulines. "On stochastic proximal gradient algorithms." arXiv preprint arXiv:1402.2365 (2014).

[3] Rosasco, Lorenzo, Silvia Villa, and Bang Cng VÅ. "Convergence of stochastic proximal gradient algorithm." arXiv preprint arXiv:1403.5074 (2014).

[4] Rosasco, Lorenzo, Silvia Villa, and Bang Cng VÅ. "A Stochastic forward-backward splitting method for solving monotone inclusions in Hilbert spaces." arXiv preprint arXiv:1403.7999 (2014).

[5] Zhang, Ziming, and Venkatesh Saligrama. "RAPID: Rapidly Accelerated Proximal Gradient Algorithms for Convex Minimization." arXiv preprint arXiv:1406.4445 (2014).

[6] Xiao, Lin, and Tong Zhang. "A proximal stochastic gradient method with progressive variance reduction." SIAM Journal on Optimization 24, no. 4 (2014): 2057-2075.

[7] Lin, Qihang, Zhaosong Lu, and Lin Xiao. "An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization." arXiv preprint arXiv:1407.1296 (2014).

[8] Nitanda, Atsushi. "Stochastic Proximal Gradient Descent with Acceleration Techniques." In Advances in Neural Information Processing Systems, pp. 1574-1582. 2014.

[9] Solomon, Justin, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. "Earth mover's distances on discrete surfaces." ACM Transactions on Graphics (TOG) 33, no. 4 (2014): 67.

[10] Xue, Guoliang, and Yinyu Ye. "An efficient algorithm for minimizing a sum of Euclidean norms with applications." SIAM Journal on Optimization 7, no. 4 (1997): 1017-1036.