University of California, Davis

STA 243 Final Report

Author:
Xialin Sang 917780316
Bohao Zhou 917796070
Ruichen Xu 917858772

June 12, 2020

1 Introduction

Numerical optimization algorithms generally use the derivative information of the objective function, such as the first and second derivatives. If the first derivative is used, it is called a first-order optimization algorithm. If the second derivative is used, it is called a second-order optimization algorithm.

As for the first-order optimization algorithm, the gradient descent method searches in the opposite direction of the gradient, using the information of the first derivative of the function. Because this direction is the fastest descent direction of the current position, so it is also called the Steepest Descent Method. When the objective function is a convex function, the solution of the gradient descent method is a global solution. The closer the steepest descent method is to the target value, the smaller the step size and the slower the advance.

In machine learning, based on the basic gradient descent method, two gradient descent methods have been developed, namely, the random gradient descent method and the batch gradient descent method.

Compared with the first-order optimization algorithm, the second-order optimization algorithm using gradient and Hessian information has a higher convergence rate in theory and practice. For example, Newton's method is a second-order optimization algorithm, and its key idea is to perform Taylor expansion on the function.

The Newton method not only needs to calculate the Hessian matrix, but also needs to calculate the inverse of the Hessian matrix. When the amount of data is relatively small, the operation speed will not be greatly affected. However, when the amount of data is large, especially in deep neural networks, calculating the Hessian matrix and its inverse matrix is very time-consuming. From the overall effect, the optimization speed of Newton method is not as fast as gradient descent algorithm. Therefore, most of the current optimization strategies of neural network loss functions are based on gradient descent. It is worth mentioning that, for the shortcomings of Newton's method, there have been some improved algorithms. Such improved algorithms are collectively called quasi-Newton algorithms. The more representative ones are BFGS(Broyden Fletcher Goldfarb Shanno Algorithm) and L-BFGS(Limited Memory BFGS Algorithm)[7].

The BFGS algorithm uses an approximate method to calculate the inverse of the Hessian matrix, which effectively improves the calculation speed[5]. However, the entire Hessian approximate inverse matrix still needs to be stored, and the space cost is relatively large. The L-BFGS algorithm is an improvement to the BFGS algorithm. It does not need to store the Hessian approximate inverse matrix, but directly obtains the search direction of the current round through the iterative algorithm, and the space cost is greatly reduced.

1.1 Related work

Pilanci and Wainwright [1] proposed a Newton method using sketching techniques to solve the problem that the number of variables d is small but the number of data points n can be extremely large, called Newton Sketch. This paper introduces the use of random projection method and sampling Hessian function[4] for Newton iteration method, which project high-dimensional large data sets to low-dimensional data sets, and the efficient information is remained during the dimensional reduction process. After achieving excellent approximate linear effect, the complexity of Newton iteration has been greatly reduced. It can be widely used in large-scale linear programming and quadratic programming, etc, such as logistic regression.

2 Methodology

In this section, we will discuss the methodology and it main idea of this algorithm. Let's define a minimize objective function $\arg\min_{x\in\Omega} f(x)$, where Ω is the sample space which may constrained or unconstrained.

2.1 Newton's algorithm

The standard form of Newton's algorithm use the second order Taylor expansion. In the standard Newton's algorithm, given a current iterate $x^t \in \Omega$, where Ω is the sample space of x^t and $\Omega \subseteq \mathbb{R}^d$, it generate the new iterate x^{t+1} with the following formula.

$$x^{t+1} = \arg\min_{x\subseteq\Omega}\{\frac{1}{2} < x-x^t, \nabla^2 f(x^t)(x-x^t) > + < \nabla f(x^t), x-x^t >\} \tag{1}$$

where $< \cdot,\cdot >$ means the inner product of two vectors. Suppose that we have available a Hessian matrix square root $\nabla^2 f(x)^{\frac{1}{2}}$, it is a $n \times d$ matrix which have the property $(\nabla^2 f(x)^{\frac{1}{2}})^T \nabla^2 f(x)^{\frac{1}{2}} = \nabla^2 f(x)$, for $n \geq rank(\nabla^2 f(x))$. At this present, let's consider a function of the form $f(x) = g(Ax)$ where $A \in \mathbb{R}^{n\times d}$ and the function $g : \mathbb{R}^n \to \mathbb{R}$ can be written as $g(Ax) = \sum_{i=1}^n g_i(< a_i, x >)$. In this case, a square root of Hessian matrix is given by $n \times d$ matrix $\nabla^2 f(x)^{\frac{1}{2}} = diag\{g_i^{''}(< a_i, x >)\}_{i=1}^n A$.

In terms of this notation, the ordinary Newton update can be re-written as

$$x^{t+1} = \arg\min_{x\subseteq\Omega}\{\frac{1}{2}||\nabla^2 f(x^t)^{\frac{1}{2}}(x-x^t)||_2^2 + < \nabla f(x^t), x-x^t >\} \tag{2}$$

and we can use those formula to find the globe minimize value in the convex function and convex sample space.

2.2 Newton Sketch algorithm

2.2.1 Fully Sketch Newton

In this section, we will discuss the fully sketch newton method. For a sketch matrix $S \in \mathbb{R}^{m\times n}$, it is an *isotropic sketch matrix*, satisfying the relation $\mathbb{E}[S^T S] = I_n$. For the first step, we need to choose the sketch dimension m. Then, the Newton sketch algorithm generates a sequence of iterates $\{x^t\}_0^\infty$ according to the recursion.

$$x^{t+1} = \arg\min_{x\subseteq\Omega}\{\frac{1}{2}||S^t\nabla^2 f(x^t)^{\frac{1}{2}}(x-x^t)||_2^2 + < \nabla f(x^t), x-x^t >\} \tag{3}$$

where $S^t \in \mathbb{R}^{m\times n}$ is an *independent realization of a sketching matrix*. When the problem is unconstrained, i.e., $\Omega = \mathbb{R}^d$ and the matrix $(\nabla^2 f(x^t)^{\frac{1}{2}})^T (S^t)^T S^t \nabla^2 f(x^t)^{\frac{1}{2}}$ is invertible, the Newton sketch update takes the simpler form to

$$x^{t+1} = x^t - ((\nabla^2 f(x^t)^{\frac{1}{2}})^T (S^t)^T S^t \nabla^2 f(x^t)^{\frac{1}{2}})^{-1} \nabla f(x^t) \tag{4}$$

2.2.2 Partially Sketched Newton

Given an additive decomposition of the form $f(x) = f_0(x) + g(x)$, we perform a sketch of the Hessian $\nabla^2 f_0(x)$ but remain the exact form of the Hessian $\nabla^2 g(x)$. This conducts the partially sketched update

$$x^{t+1} = \arg\min_{x\in\Omega}\{\frac{1}{2}(x-x^t)^T Q^t (x-x^t) + < \nabla f(x^t), x-x^t >\} \tag{5}$$

where $Q^t = (S^t \nabla^2 f_0(x^t)^{\frac{1}{2}})^T S^t \nabla^2 f_0(x^t)^{\frac{1}{2}} + \nabla^2 g(x^t)$

The main idea of *Sketch Newton algorithm* is used a sketch matrix $S \in \mathbb{R}^{m \times n}$ where $m \ll n$ to reduce the dimension of original Hessian matrix $\nabla^2 f(x)$. The sketch dimension m can be chosen to be substantially smaller than n, in this case the sketched Newton updates will be much cheaper than a standard Newton update.

The intuition of the *Sketch Newton algorithm* is that: in the iterate x^{t+1}, the random objective function

$$\Phi(x; S^t) = \frac{1}{2}||S^t \nabla^2 f(x^t)^{\frac{1}{2}}(x - x^t)||_2^2 + < \nabla f(x^t), x - x^t >$$

whose expectation is $\mathbb{E}(\Phi(x; S^t))$, taking average over the isotropic sketch matrix S^t, is equal to the *Standard Newton objective*

$$\Phi(x) = \frac{1}{2}||\nabla^2 f(x^t)^{\frac{1}{2}}(x - x^t)||_2^2 + < \nabla f(x^t), x - x^t >$$

In this case, this algorithm can be seen as the stochastic form of Newton update.

3 Theoretical Results

3.1 sketch matrix

From the a theoretical prospective, the sub-Gaussian sketches has obvious concentration property. However, from the computational complexity perspective, the sub-Gaussian sketches is not so perfect. Because we need to consider the problem of multiplying sketch matrix and aim matrix at each step, the computational complexity we need is $\mathbf{O}(nmd)$, which is not a easy job. But when we turn to randomized orthonormal systems (ROS) sketch for help, these matrices care based on Fourier or Hadamard, so the computational complexity can be reduced to $\mathbf{O}(n \log n)$. I do not know how to show that, but when writing the codes, for sketch matrix, we just put elements from N(0,1) and then divide $\sqrt{m}$

3.2 Local convergence analysis using strong convexity

We now following from the article to show that the sketch Newton methods will guarantee the local convergence. We mainly focus on what size of m we need to keep this property.

Based on this paper, we study this question from the perspective of geometry. Naturally, the tangent cone from this optimum occurs. Considering the constraint set $\mathbf{C}$, and the minimizer for the aim function denoted by $x^* :=_{x \in \mathscr{C}} f(x)$, then based on the paper, the tangent cone is defined by:

$$\mathscr{K} := \{\nabla \in |x^* t \nabla for \quad some \quad t > 0\}$$

Then we need to consider the Gaussian width, which may help us make sure the size of compact set in $\mathbb{R}^d$. Here is a example,

$$\mathscr{W}(\mathscr{L}) := \mathbb{E}_g \left[\max_{z \in \mathscr{L}} |\langle g, z \rangle| \right]$$

, where $\mathscr{L}$ is a compact set in $\mathbb{R}^d$, and $g \in \mathbb{R}^d$, which elements are from N(0,1).

The author says that the set we focus on in this paper is ⌋ that for a given cone $\mathscr{K}$, the intersection of this cone with a unit Euclidean sphere $\mathscr{S}^{d-1}$. And more interestingly, Gaussian width of this intersection set is at most $\sqrt{d}$, and general which would be very small. Finally, for a example, if the dimension of $\| = r$ is small than d, we have $\mathscr{W}(\mathscr{K} \cap \mathscr{S}^{d-1}) \leq \sqrt{r}$. Then considering the sketch dimension satisfy the following:

$m \geq \frac{c}{\varepsilon^2} \max_{x\in\mathscr{C}} \mathscr{W}^2\left(\nabla^2 f(x)^{1/2}\mathscr{K}\right)$, and the author mainly consider the cone-constrained eigenvalues of the Hessian, which is

$$\gamma = \inf_{z\in\mathscr{K}\cap\mathscr{S}^{d-1}} \left\langle z, \nabla^2 f(x^*)\right) z\Big\rangle, \quad \text{and} \quad \beta = \sup_{z\in\mathscr{K}\cap\mathscr{S}^{d-1}} \left\langle z, \nabla^2 f(x^*)\right) z\Big\rangle$$

And interestingly, if the case is unconstrained, the γ ,and *beta* will be the minimum and maximum of the eigenvalue of hessian matrix.

And then we introduce the theorem 1 in this paper. (The concept and representation of theorem is directly from the article.)

Theorem 1 (Local convergence of Newton Sketch). For given parameters $\delta, \varepsilon \in (0,1)$, consider the Newton sketch updates based on an initialization x^0 such that $\left\|x^0 - x^*\right\|_2 \leq \delta\frac{\gamma}{8L}$ and a sketch dimension m satisfying the lower bound as above, Then with probability at least $1 - c_1 e^{-c_2 m}$, the ℓ_2 -error satisfies the recursion

$$\left\|x^{t+1} - x^*\right\|_2 \leq \varepsilon\frac{\beta}{\gamma}\left\|x^t - x^*\right\|_2 + \frac{4L}{\gamma}\left\|x^t - x^*\right\|_2^2$$

One of the most interesting of this theorem is applicable twice-differential f with σ and β as below.

Then we focus on the ε in this theorem, when it is a constant. Then the linear-quadratic convergence will be shown directly from theorem 1 with the $x^t - x^*$. And when the norm of the error is large enough, the rate is initially quadratic, that is $\|x^t - x^*\|_2 \sim \frac{4L}{\gamma}\|x^t - x^*\|_2^2$. But the error is not so big, less than 1, something changes. That is that $\|x^t - x^*\|_2 \sim \frac{\varepsilon\beta}{\gamma}\|x^t - x^*\|_2$, which is linear convergences.

All in all, the most important meaning of theorem 1 is that for a sketch matrix, m dimension, we may get the linear-quadratic convergence.

Then from the theorem 1 we can get the Corollary 1, the Corollary is directly from the theorem 1.

Corollary 1. **Consider the Newton sketch iterates using the iteration-dependent sketching accuracy $\varepsilon(t) = \frac{1}{\log(1+t)}$. Then with the same probability as in Theorem** 1, **we have**

$$\left\|x^{t+1} - x^*\right\|_2 \leq \frac{1}{\log(1+t)}\frac{\beta}{\gamma}\left\|x^t - x^*\right\|_2 + \frac{4L}{\gamma}\left\|x^t - x^*\right\|_2^2$$

and consequently, super-linear convergence is obtained-namely, $\lim_{t\to\infty} \frac{\left\|x^{t+1}-x^*\right\|_2}{\left\|x^t-x^*\right\|_2} = 0$

In the Corollary 1, the ε is not a constant. And thus corollary indicates that for a sketch matrix, m dimension, we may get the super-linear convergence. But we need to pay attention, the sketch matrix is influenced by the time, which may trouble us when coding and computing.

3.3 Newton sketch for self-concordant functions

3.3.1 Unconstrained cases

The function we consider in this part is closed convex self-concordant function, $|\phi'''(x)| \leq 2(\phi''(x))^{3/2}$. In the article use $\phi_{x,y} := f(x+ty)$ to extend the definition of self-concordant function from $\mathbb{R}\to\mathbb{R}$ to $\mathbb{R}^d \to \mathbb{R}$. The concordant function is usual in our daily life, such as linear function, quadratic function and negative logarithm. More interesting, the concord function will keep under addition and affine transformation.

Theorem 2. Let f be a strictly convex self-concordant function. Given a sketching matrix $S \in \mathbb{R}^{m\times n}$ with $m \geq \frac{c_3}{\varepsilon^2}\max_{x\in\mathscr{C}} \operatorname{rank}\left(\nabla^2 f(x)\right) = \frac{c_3}{\varepsilon^2}d$, the number of total iterations T for obtaining an δ approximate solution in function value via Algorithm 1 is bounded by

$$T = \frac{f\left(x^0\right) - f\left(x^*\right)}{\nu} + 0.65\log_2\left(\frac{1}{16\delta}\right)$$

with probability at least $1 - c_1 N e^{-c_2 m}$

From the theorem 2, this theorem is powerful when we consider the unconstrained cases. Because the bounded provided by the theorem 2 is not a matter of function f and problem parameters.

From the algorithm 1, the computational complexity for newton sketch is most $\nVdash + \ltimes \log$, which is smaller than the computational complexity of newton, $\ltimes$. In our experiment we get the decent result, the time of newton sketch converges is smaller than other cases.

3.3.2 Newton Sketch with self-concordant barriers

In this part, we add a convex self-concordant barrier function g in this part, which is the same to say that we consider $min_{x\in\mathbb{R}^d}\{f_0(x)+g(x)\}$. The author provides two ways to resolve this problem, for some simple cases, we just take g as usual part and then sketch this function as usual; in other cases, the g is the separable part, we just sketch the first part like the part 2 we point out.

Then the algorithm 2 we show also in the appendix. Specially, the choice of sketch dimension in this situation is based on tangent cone relative to iterations. As above, the sketch dimension need to satisfy the lower bound: $m^t \geq \frac{c_3}{\varepsilon^2}\max_{x\in\mathscr{C}} \mathscr{W}^2\left(\nabla^2 f(x)^{1/2}\mathscr{K}^t\right)$, where, $\nu = ab\frac{\eta^2}{1+\left(\frac{1+\varepsilon}{1-\varepsilon}\right)\eta}$ where $\eta = \frac{1}{8}\frac{1-\frac{1}{2}\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 - a}{\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^3}$, obviously, which only depends on the ε and line search parameter.

Theorem 3. Let $f:\mathbb{R}^d\to\mathbb{R}$ **be a convex and self-concordant function, and let** $g:\mathbb{R}^d\to\mathbb{R}\cup\{+\infty\}$ **be a convex and self-concordant barrier for the convex set** $\mathscr{C}$**. Suppose that we implement Algorithm 2 with sketch dimensions** $\{m^t\}_{t\geq 0}$ **satisfying the lower bound (19). Then taking**

$$N = \frac{f\left(x^0\right) - f\left(x^*\right)}{\nu} + 0.65\log_2\left(\frac{1}{16\delta}\right) \quad \textbf{iterations}$$

suffices to obtain δ**-approximate solution in function value with probability at least** $1 - c_1 N e^{-c_2 m}$

Form the theorem 3, the self-concordant barrier functions can be optimizer by newton sketch. This conclusion is very useful, because in our real life, we need to consider the Lasso or ridge regression, which include the penalty. So this method and this algorithm can hugely increase newton sketch scope.

3.3.3 Sketching with interior point method

In this part, we consider the problem which satisfy: $\min_{x\in\mathbb{R}^d} f_0(x)$ subject to $g_j(x)\leq 0$. Then with the barrier method, the question can be transformed to :$\widehat{x}(\tau) := \arg\min_{x\in\mathbb{R}^d}\left\{\tau f_0(x) - \sum_{j=1}^r \log\left(-g_j(x)\right)\right\}$. Then consider the algorithm 3 in the appendix, there are two ways to deal with this problem, one is to use algorithm 1 with fully newton sketch, the other is to use algorithm 2 with partial sketch. The both two methods can guarantee the convergence but different computational complexity. The following theorem is about the barrier method relative to both two methods. **Theorem 4 (Newton Sketch complexity for interior point methods). For a given target accuracy** $\delta\in(0,1)$ **and any** $\mu > 1$, **the total number of Newton Sketch iterations required to obtain a** δ **-accurate solution using Algorithm 3 is at most**

$$\left\lceil \frac{\log\left(r/\left(\tau^0\delta\right)\right)}{\log\mu}\right\rceil\left(\frac{r(\mu-1-\log\mu)}{\gamma} + 0.65\log_2\left(\frac{1}{16\delta}\right)\right)$$

Following from theorem 4, when μ is set a constant which is the upper bound, then we need to conduct $\mathbf{O}(\sqrt{r})$ iterations. But it is just an idea. However, it is hard to get such kind of *mu* in practice. From the author, generally, we fix $\mu\in[2,100]$. In experiment, we usually need the Newton iterations is independent with other parameters to make the experiment well done, and get the fair result. The most important part of theorem 4 is to help us get the faster interior points solves with rigorous worst-case complexity results. Finally, I will introduce some applications of algorithm 3 in the appendix.

4 Experimental Details

We will apply this method only on Logistic Regression because of the limitation of computing resource. The goal of this experiment is to compare the consuming time and the convergence ability of Sketch Newton and Standard Newton and the influence of different reduced dimensions to convergence ability of Sketch Newton.

4.1 Set up

4.1.1 Set up for comparing two algorithms

For Sketch Newton algorithm, the reduced dimension in this project which we selected is 100. We simulated a data matrix $X \in \mathbb{R}^{N\times K}$ from Standard Normal Distribution $\mathbb{N}(0,1)$, where N represents the samples numbers and K denotes the coefficients number. In this project, we selected $N = 1000$ and $K = 100$.We randomly established the labels of each samples from Bernoulli distribution. The probability is $p = 0.5$ for label 1. The initial coefficients $\vec{\beta}^0 \in \mathbb{R}^K$ randomly build from Standard Normal Distribution $\mathbb{N}(0,1)$. The learning rate for Sketch Newton is $1e^{-7}$. **For Standard Newton algorithm**, we used the same data matrix, same labels and same initial coefficients. The learning rate for Standard Newton is $1e^{-6}$. The max iteration of those two algorithms are all 25000 times.

4.1.2 Set up for exploring different reduced dimensions

We selected $N = 500$ and $K = 50$ for this data matrix. The reduced dimensions which we selected are $[5, 25, 45, 65, 85, 105]$. We randomly established the labels of each samples from Bernoulli distribution. The probability is $p = 0.5$ for label 1. The initial coefficients $\vec{\beta}^0 \in \mathbb{R}^K$ randomly build from Standard Normal Distribution $\mathbb{N}(0,1)$. The learning rate for those Sketch Newton are all $1e^{-3}$. The max iterations are all set 25000.

4.2 Results

The comparison between Sketch Newton and Standard Newton and the effect of different reduced dimensions are showed in the plots (Because the limitation of pages, those plots are showed in appendix). We used the mean MSE loss for the predict vector and truth labels to present the convergence ability.

From figure 1 we can know that there contains fluctuation in the Sketch Newton method and the convergence ability of Sketch Newton method is much better than the Standard Newton method in the same iterations even though the learning rate of Sketch Newton is 10 times smaller than Standard Newton. From the theory 3.3.1 we can know that this is in our expectation. The consuming time of Sketch Newton is 46 minutes with 56 seconds. The consuming time of Standard Newton is 1 hour 10 minutes with 53 seconds. Those phenomenons indicate that this Sketch Newton algorithm transcends the Standard Newton method prodigious, regardless on the consuming time or the convergence ability.
From figure 2 we can know that with different reduced dimension, the convergence ability is raised when reduced dimension increased and then declined when reduced dimension becomes huge. The best reduced dimension in those experiments is 45. If the reduced dimensions too small, the information contains in the sketched Hessian matrix is small. So, it is hard to estimate the truth value that the convergence ability is low. However, with the dimension increasing, the raising rate of information is decreased but the random factor in Sketch Newton algorithm will effect the estimation ability more seriously. So, the convergence ability will decrease slightly.

5 Appendix A: The algorithm for theoretical part

5.1 The algorithm: 1

Algorithm 1 Unconstrained Newton Sketch with backtracking line search

Input: Starting point x^0, tolerance $\delta > 0, (a,b)$ line-search parameters, sketching matrices $\{S^t\}_{t=0}^{\infty} \in \mathbb{R}^{m\times n}$ 1: Compute approximate Newton step Δx^t and approximate Newton decrement $\lambda(x)$

$$\Delta x^t := \arg\min_{\Delta} \left\langle \nabla f\left(x^t\right), \Delta \right\rangle + \frac{1}{2}\left\| S^t\left(\nabla^2 f\left(x^t\right)\right)^{1/2}\Delta\right\|_2^2$$
$$\widetilde{\lambda}_f\left(x^t\right) := \nabla f(x)^T \Delta x^t$$

2: Quit if $\tilde{\lambda}\left(x^t\right)^2/2 \le \delta$ 3: Line search: choose μ : while $f\left(x^t+\mu\Delta x^t\right) > f\left(x^t\right) + a\mu\lambda\left(x^t\right), \quad \mu \leftarrow b\mu$ 4: Update: $x^{t+1} = x^t + \mu\Delta x^t$ Output: minimizer x^t, optimality gap $\lambda\left(x^t\right)$

5.2 The algorithm: 2

Algorithm 2 Newton Sketch with self-concordant barriers

Input: Starting point x^0, constraint $\mathscr{C}$, corresponding barrier function g such that $f = f_0 + g$, tolerance $\delta > 0, (\alpha,\beta)$ line-search parameters, sketching matrices $S^t \in \mathbb{R}^{m\times n}$ 1: Compute approximate Newton step Δx^t and approximate Newton decrement $\lambda(x)$

$$\Delta x^t := \arg\min_{x^t+\Delta\in\mathscr{C}} \left\langle \nabla f\left(x^t\right), \Delta \right\rangle + \frac{1}{2}\left\| S^t\left(\nabla^2 f_0\left(x^t\right)\right)^{1/2}\Delta\right\|_2^2 + \frac{1}{2}\Delta^T\nabla^2 g\left(x^t\right)\Delta$$
$$\widetilde{\lambda}_f\left(x^t\right) := \nabla f(x)^T \Delta x^t$$

2: Quit if $\tilde{\lambda}\left(x^t\right)^2/2 \le \delta$ 3: Line search: choose μ : while $f\left(x^t+\mu\Delta x^t\right) > f\left(x^t\right) + \alpha\mu\lambda\left(x^t\right), \quad \mu \leftarrow \beta\mu$ 4: Update: $x^{t+1} = x^t + \mu\Delta x^t$ Output: minimizer x^t, optimality gap $\lambda\left(x^t\right)$

5.3 The algorithm: 3

Algorithm 3 Interior point methods using Newton Sketch

Input: Strictly feasible starting point x^0, initial parameter τ^0 s.t. $\tau := \tau^0 > 0, \mu > 1$, tolerance $\delta > 0$ 1: Centering step: Compute $\widehat{x}(\tau)$ by Newton Sketch with backtracking line-search initialized at x using Algorithm 1 or Algorithm 2 2: Update $x := \widehat{x}(\tau)$ 3: Quit if $r/\tau \le \delta$ 4: Increase τ by $\tau := \mu\tau$ Output: minimizer $\widehat{x}(\tau)$

6 Appendix B: The result of experiments

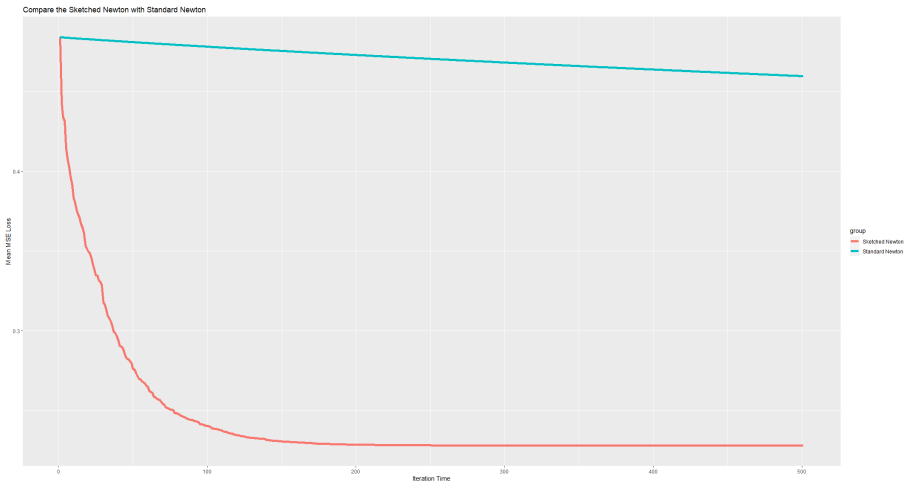

Figure 1: Compare the sketch newton with standard newton

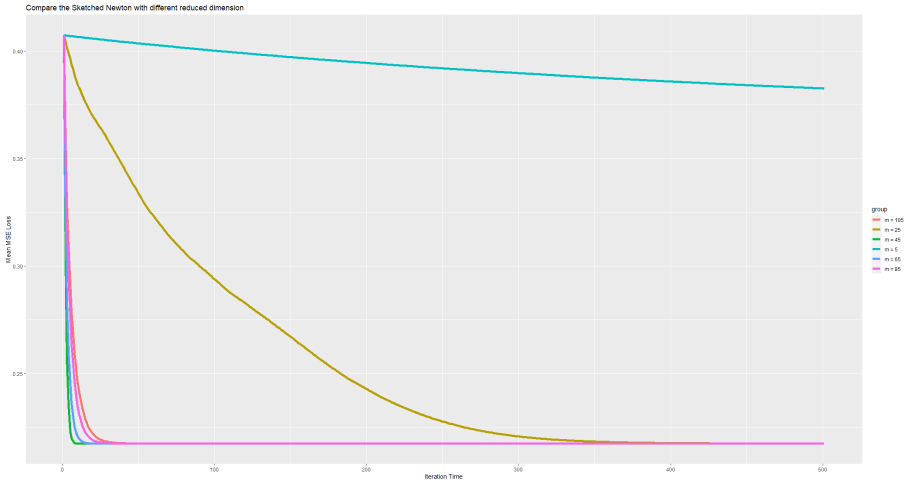

Figure 2: Compare standard newton with different reduced dimensions

7 Appendix C: Summary of Applications

In this section, we summarize various forms of Hessian structures from different optimization problems when applying Newton Sketch. Under barrier method with Newton Sketch, it shows how flexible in sketching when the objective and/or the constraints contain more than one term[1].

7.1 Estimation in generalized linear models

Under the conditions of (constrained) maximum likelihood estimation for a generalized linear model(GLM), the problem can be written as:

$$\min_{x\in C}\{\sum_{i=1}^{n}\psi(<a_i,x>,y_i)\} \tag{6}$$

, where $\psi:\mathbb{R}\to\mathbb{R}$ is a given convex function from the probabilistic model, and $C\subseteq\mathbb{R}^d$ is a closed set.
Then, we try to use Newton sketch algorithm on this optimization problem (1). Given the current iterate x^t, calculating the next iteration x^{t+1} requires solving a constrained quadratic program:

$$\min_{x\in C}\{\frac{1}{2}\left\|Sdiag(\psi^{''}(<a_i,x^t>,y_i))^{\frac{1}{2}}A(x-x^t)\right\|_2^2+\sum_{i=1}^{n}<x,\psi^{'}(<a_i,x^t>,y_i)>\} \tag{7}$$

Under the ℓ_1-constrained case, we need to find a suitable sketch dimension m. Based on our choice, the minimal eigenvalue of the ℓ_1-restricted of the data matrix A^TA is shown as below:

$$\gamma_s^-(A):=\min_{\substack{\|z\|_2=1\\ \|z\|_1<2\sqrt{s}}}\|Az\|_2^2$$

The inequality $\gamma_s^-(A)\geq\lambda_{min}(A^TA)$ is always true.
Moreover, the certain quantities that depend on the function ψ are:

$$\psi_{min}^{''}:=\min_{x\in C}\min_{i=1,\ldots,n}\psi^{''}(<a_i,x>,y_i);$$

$$\psi_{max}^{''}:=\max_{x\in C}\max_{i=1,\ldots,n}\psi^{''}(<a_i,x>,y_i) \tag{8}$$

, where $a_i\in\mathbb{R}^d$
Under the above set-up, assuming the optimal solution x^* has cardinality at most $\|x^*\|_0\leq s$, then it canbe shown that it suffices to take a sketch size[1]

$$m=c_0\frac{\psi_{max}^{''}}{\psi_{min}^{''}}\frac{\max_{j=1,\ldots,d}\left\|A_j\right\|_2^2}{\gamma_s^-(A)}slogd \tag{9}$$

, where c_0 is a universal constant. The complexity of Newton sketch per iteration update scales as $O(s^2dlog^2(d))$ using standard Lasso solvers [10].

7.2 Semidefinite programs

Consider the metric learning problem (according to different tasks, learn the metric distance function for a specific task) studied in machine learning. Given feature vectors $a_1,\ldots,a_n\in\mathbb{R}^d$ and their corresponding indicator $y_{ij}\in\{-1,+1\}^n$. When a_i and a_j are same, y_ij will be -1 otherwise for all $i\neq j,i\leq i,j\neq n$. Our goal is to learn a positive semidefinite matrix X which represents a metric, so that the semi-norm

$\|a\|_X = \sqrt{< a, Xa >}$ establishes the proximity measure based on the class label. With ℓ_2-loss, the semi-definite program (SDP) problem can be expressed as following:

$$\min_{X \succcurlyeq 0} \{ \sum_{i \neq j}^{\binom{n}{2}} (< X, (a_i - a_j)(a_i - a_j)^T > - y_{ij})^2 + \lambda trace(X) \} \tag{10}$$

The term $trace(X)$ and the user-adjustable multiplicative pre-factor $\lambda > 0$ are regularization terms used to encourage lower-level solutions. With the standard self-concordant barrier $X \mapsto logdet(X)$ for the PSD(Positive Semidefinite) cone. We can solve the following sequence of sub-problem of the form using barrier method:

$$\min_{X \in \mathbb{R}^{d \times d}} \{ \tau \sum_{i=1}^{n} (< X, a_i a_i^T >, y_i)^2 + \tau \lambda trace(X) - logdet(X) \}$$

Denote $f(vec(X)) = \tau \sum_{i=1}^{n} (< X, a_i a_i^T >, y_i)^2 + \tau \lambda trace(X) - logdet(X)$. Then, the Hessian of $f(vec(X))$ is a $d^2 \times d^2$ matrix given by:

$$\nabla^2 f(vec(X)) = \tau \sum_{i \neq j}^{\binom{n}{2}} vec(A_{ij}) vec(A_{ij})^T + X^{-1} \otimes X^{-1}$$

, where $\otimes$ is the Kronecker product, $A_{ij} := (a_i - a_j)(a_i - a_j)^T$
Then, we use the barrier method with partial Hessian sketch on the first term $\{S_{ij} vec(A_{ij})\}_{i \neq j}$, and compute the exact Hessian for the second term. Since $vec(X) \in \mathbb{R}^{d^2}$, the complexity of Newton Sketch is $\mathbb{O}(m^2 d^2)$

7.3 Portfolio optimization and SVMs

Let us consider the Markowitz formulation of the portfolio optimization problem[9]. The purpose is to find $x \in \mathbb{R}^d$ which belongs to the unit simplex, thereby maximizing the expected return minus the coefficient times the return variance. Denote $\mu \in \mathbb{R}^d$ is a vector corresponding to mean return of the assets, $\Sigma \in \mathbb{R}^{d \times d}$ is a symmetric, PSD matrix, covariance of the returns. Then the optimization problem is shown as below:

$$\max_{X \geq 0, \sum_{j=1}^{d} x_j \leq 1} \{ < \mu, x > - \lambda x^T \Sigma x \} \tag{11}$$

$\Sigma = A^T A$, where the columns of A are time series corresponding to assets normalized by $\sqrt{n}$. Using barrier method, we can figure out (11) by solving the penalized problems (12):

$$\min_{X \in \mathbb{R}} \{ -\tau \mu^T x + \tau \lambda x^T A^T A x - \sum_{i=1}^{d} log(< e_i, x >) - log(1 - < 1, x >) \} \tag{12}$$

, where $e_i \in \mathbb{R}^d$ is the i^{th} element of the canonical basis, and 1 is the row vector of all-ones. Denote $f(x) = -\tau \mu^T x + \tau \lambda x^T A^T A x - \sum_{i=1}^{d} log(< e_i, x >) - log(1 - < 1, x >)$. Then the Hessian matrix is :

$$\nabla^2 f(x) = \tau \lambda A^T A + diag(x_i^2)^{-1} + 11^T$$

Thus, we can sketch the data dependent part of the Hessian through $\tau \lambda SA$. The complexity of Newton Sketch per iteration scales as $O(md^2)$, compared to the complexity of the classical interior points method per step $O(nd^2)$.

7.4 A dual example: Lasso with $d \gg n$

Consider the corresponding dual problem is :

$$\max_{\|A^T w\|_\infty \leq \lambda} \{-\frac{1}{2}\|y-w\|_2^2\}$$

After applying the barrier method, the sequence of problems of the form is shown as below:

$$\min_{w \in \mathbb{R}^n} \{\tau \|y-w\|_2^2 - \sum_{j=1}^{d} log(\lambda - < A_j, w >) - \sum_{j=1}^{d} log(\lambda + < A_j, w >)\} \tag{13}$$

Denote $f(x) = \tau \|y-w\|_2^2 - \sum_{j=1}^{d} log(\lambda - < A_j, w >) - \sum_{j=1}^{d} log(\lambda + < A_j, w >)$. Then Hessian matrix of the problem (13) is

$$\nabla^2 f(w) = \tau I_n + A diag(\frac{1}{(\lambda - < A_j, w >)^2} A^T + A diag(\frac{1}{(\lambda + < A_j, w >)^2}) A^T$$

. Hence, after keeping the exact Hessian of the frst term, we apply the partial sketching to the Hessian of the last two term by:

$$S diag(\frac{1}{(\lambda - < A_j, w >)^2} + \frac{1}{(\lambda + < A_j, w >)^2}) A^T$$

The complexity of above Newton Sketch for $d \gg n$ is $O(dm^2)$, where m is at most d. Compared to the complexity of traditional interior point solvers per iteration is $O(dn^2)$

References

[1] Pilanci, Mert, and Martin J. Wainwright. "Newton sketch: A near linear-time optimization algorithm with linear-quadratic convergence." SIAM Journal on Optimization 27.1 (2017): 205-245.

[2] Y. Nesterov. Introductory Lectures on Convex Optimization. Kluwer Academic Publishers, New York, 2004.

[3] Nesterov, Yurii. "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$." Doklady an ussr. Vol. 269. 1983. pages 543–547, 1983.

[4] Pilanci, Mert, and Martin J. Wainwright. "Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares." The Journal of Machine Learning Research 17.1 (2016): 1842-1879.

[5] Moritz, Philipp, Robert Nishihara, and Michael Jordan. "A linearly-convergent stochastic L-BFGS algorithm." Artificial Intelligence and Statistics. 2016.

[6] P. McCullagh and J.A. Nelder. Generalized linear models. Monographs on statistics and applied probability 37. Chapman and Hall/CRC, New York, 1989.

[7] Moritz, Philipp, Robert Nishihara, and Michael Jordan. "A linearly-convergent stochastic L-BFGS algorithm." Artificial Intelligence and Statistics. 2016.

[8] M. Pilanci and M. J. Wainwright. Randomized sketches of convex programs with sharp guarantees. Technical report, UC Berkeley, 2014. Full length version at arXiv:1404.7203; Presented in part at ISIT 2014.

[9] M. Ledoux and M. Talagrand. Probability in Banach Spaces: Isoperimetry and Processes. Springer-Verlag, New York, NY, 1991.

[10] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ‘1-regularized least squares. IEEE Journal on Selected Topics in Signal Processing, 1(4):606–617, 2007.