CS 450 Numerical Analysis (22Fa): Project Report

Xu Ke (3190110360)

Dec. 27, 2022

1 Optimization Problem

An optimization problem is the problem of finding the best solution from all feasible solutions. In this project, we consider the problem of minimizing an average of n functions f_i : $minimize f(\theta) := \frac{1}{n}\sum_{n=1}^{n} f_i(\theta)$ in a batch setting, where n is assumed to be much larger than p. Most machine learning models can be expressed like this, where each function f_i corresponds to an observation, such as logistic regression (LR) and support vector machines (SVM). Many optimization algorithms have been developed to solve this minimization problem using iterative methods, $\theta^{t+1} = \theta^t - \eta_t Q^t \bigtriangledown_\theta f(\theta^t)$ where η_t is the step size and Q^t is a suitable scaling matrix that provides curvature information. The case where Q^t is equal to the identity matrix corresponds to Gradient Descent (GD) which, under smoothness assumptions, achieves linear convergence rate with $O(np)$ per-iteration cost. Second order methods like Newton's Method (NM) can be recovered by taking Q^t to be the inverse Hessian evaluated at the current iterate and may achieve quadratic convergence rates with $O(np^2 + p^3)$ per-iteration cost.

2 Newton's Method Recap

Algorithm 3: Newton's Method

input : $f : \mathbb{R}^n \to \mathbb{R}$ a twice-differentiable function
$\mathbf{x}^{(0)}$ an initial solution
output: $\mathbf{x}^\star$, a local minimum of the cost function f.

begin
  $k \leftarrow 0$;
  while STOP-CRIT **and** $(k < k_{max})$ **do**
    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \boldsymbol{\delta}^{(k)}$;
    with $\boldsymbol{\delta}^{(k)} = -\left(\mathbf{H}_f\left(\mathbf{x}^{(k)}\right)\right)^{-1} \nabla f\left(\mathbf{x}^{(k)}\right)$;
    $k \leftarrow k + 1$;
  return $\mathbf{x}^{(k)}$
end

Fig 1. Newton's Method Algorithm

Newton's method, also known as the Newton–Raphson method, named after Isaac Newton and Joseph Raphson, is a root-finding algorithm which produces successively better approximations to the roots (or zeroes) of a real-valued function. The most basic version starts with a single-variable function f defined for a real variable x, the function's derivative f', and an initial guess x_0 for a root of f. If the function satisfies sufficient assumptions and the initial guess is close, then $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$ is a better approximation of the root than x_0. Geometrically, $(x_1, 0)$ is the intersection of the x-axis and the tangent of the graph of f at $(x_0, f(x_0))$: that is, the improved guess is the unique root of the linear approximation at the initial point. The process is repeated as $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ until a sufficiently precise value is reached. This algorithm (shown in Fig 1. Newton's Method Algorithm) is first in the class of Householder's methods, as taught in CS 450 Numerical Analysis and can be extended to complex functions and to systems of equations.

3 Drawbacks & Solutions Analysis

Though Newton's Method may achieve quadratic convergence rates with $O(np^2 + p^3)$ per-iteration cost, it is insensitive to the condition number of the Hessian and when the number of samples grows larger,

computation of Q^t becomes extremely expensive. A popular line of research tries to construct the matrix Q^t in a way that the update is computationally feasible, yet still provides sufficient second order information. Such attempts resulted in Quasi-Newton methods, in which only gradients and iterates are used in the construction of matrix Q^t, resulting in an efficient update at each step t, which requires $O(np + p^2)$ per-iteration cost. An alternative approach is to use sub-sampling techniques, where scaling matrix Q^t is based on randomly selected set of data points. However, a key challenge is that the sub-sampled Hessian is close to the actual Hessian along the directions corresponding to large eigenvalues (large curvature directions in $f(\theta)$), but is a poor approximation in the directions corresponding to small eigenvalues (flatter directions in $f(\theta)$).

4 NewSamp Method

In order to solve the problems mentioned above, researchers Murat A. Erdogdu and Andrea Montanari from Stanford University proposed a method NewSamp: A Newton method via sub-sampling and eigenvalue thresholding (shown in Fig 2. NewSamp Algorithm) in paper *Convergence rates of sub-sampled Newton methods*, published in NeurIP 2015.

Algorithm 1 NewSamp

Input: $\hat{\theta}^0, r, \epsilon, \{\eta_t, |S_t|\}_t, t = 0.$

1. **Define:** $\mathcal{P}_{\mathcal{C}}(\theta) = \text{argmin}_{\theta' \in \mathcal{C}} \|\theta - \theta'\|_2$ is the Euclidean projection onto $\mathcal{C}$,
   $[\mathbf{U}_k, \mathbf{\Lambda}_k] = \text{TruncatedSVD}_k(\mathbf{H})$ is the rank-k truncated SVD of $\mathbf{H}$ with $(\mathbf{\Lambda}_k)_{ii} = \lambda_i$.
2. **while** $\|\hat{\theta}^{t+1} - \hat{\theta}^t\|_2 \leq \epsilon$ **do**
   Sub-sample a set of indices $S_t \subset [n]$.
   Let $\mathbf{H}_{S_t} = \frac{1}{|S_t|} \sum_{i \in S_t} \nabla^2_\theta f_i(\hat{\theta}^t)$, and $[\mathbf{U}_{r+1}, \mathbf{\Lambda}_{r+1}] = \text{TruncatedSVD}_{r+1}(\mathbf{H}_{S_t})$,
   $\mathbf{Q}^t = \lambda_{r+1}^{-1} \mathbf{I}_p + \mathbf{U}_r \left(\mathbf{\Lambda}_r^{-1} - \lambda_{r+1}^{-1} \mathbf{I}_r\right) \mathbf{U}_r^T$,
   $\hat{\theta}^{t+1} = \mathcal{P}_{\mathcal{C}} \left(\hat{\theta}^t - \eta_t \mathbf{Q}^t \nabla_\theta f(\hat{\theta}^t)\right)$,
   $t \leftarrow t + 1$.
3. **end while**

Output: $\hat{\theta}^t$.

Fig 2. NewSamp Algorithm

At iteration step t, the sub-sampled set of indices, its size and the corresponding sub-sampled Hessian is denoted by S_t, $|S_t|$ and H_{S_t}, respectively. Assuming that the functions f_i's are convex, eigenvalues of the symmetric matrix H_{S_t} are non-negative. Therefore, singular value (SVD) and eigenvalue decompositions coincide. The operation $TruncatedSVDk(H_{S_t})$ is the best rank-k approximation, which means, it takes H_{S_t} as input and returns the largest k eigenvalues in the diagonal matrix with the corresponding k eigenvectors. Final per-iteration cost of NewSamp will be in the form of $O(kp^2)$.

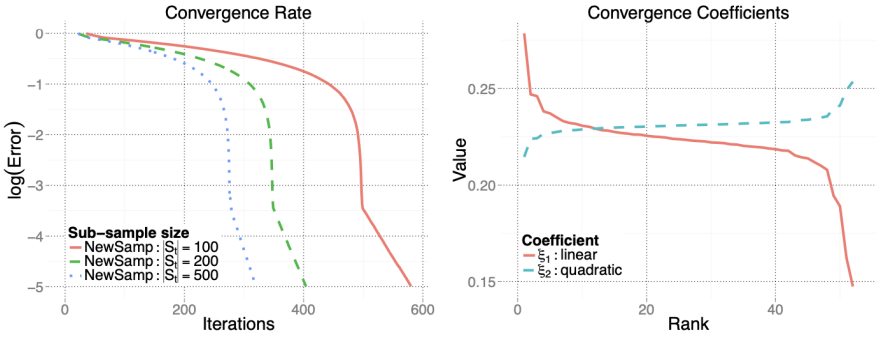

Fig 3. Convergence Behavior

As shown above (Fig 3. Convergence Behavior), the left plot shows the convergence behavior of NewSamp over different sub-sample sizes, the right plot demonstrates how the coefficients of linear and quadratic phases depend on the thresholded rank. We can observe that NewSamp enjoys a quadratic convergence rate at start which transitions into a linear rate in the neighborhood of the minimizer and large sub-samples result in better convergence rates.

Then we will look at the performance of NewSamp through extensive numerical studies. Experiments are done on two optimization problems, namely, Logistic Regression (LR) and Support Vector Machines

(SVM) with quadratic loss. Logistic Regression is a statistical model used to determine if an independent variable has effect on a binary dependent variable. Support Vector Machine finds a hyper-plane that creates a boundary between the types of data.

A thorough comparison of NewSamp with several optimization techniques is shown in Fig. 4. In the case of LR, we observe that stochastic algorithms enjoy fast convergence at start but slows down later as they get close to the true minimizer. The algorithm that comes close to NewSamp in terms of performance is BFGS. In the case of SVM, Newton's method is the closest algorithm to NewSamp, yet in all scenarios, NewSamp outperforms its competitors. For NewSamp, even though the rank thresholding provides a certain level of robustness, we observed that the choice of a good starting point is still an important factor.

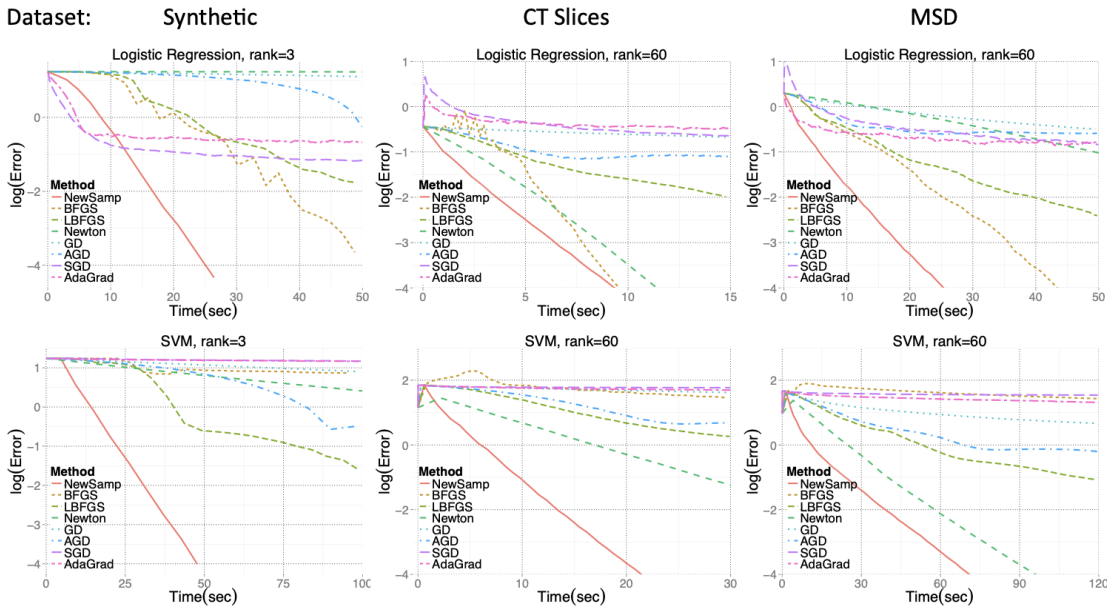

Fig 4. Performance of various optimization methods on different datasets

5 Conclusion

NewSamp, a sub-sampling based second order method utilizing low-rank Hessian estimation, has a composite convergence rate: quadratic at start and linear near the minimizer, and the complexity per iteration of NewSamp is $O(np + |S|p^2)$ with the sample size $|S|$ in the target regime $n >> p$.

6 Confusion & Limitation

Above all, I am quite confused that to the best of my knowledge quadratic at start is a bad feature because this means one should provide a good initial guess as shown in Corollary 3.7 while authors present it as a good feature. I think the major contribution of the paper is its convergence analysis. However, to the best of my knowledge, the integral version of the mean value theorem does not hold for vector-valued (matrix-valued) functions. This flaw affects the subsequent analysis and questions the main contribution of the paper.

Then comes the limitation from my perspective, this sub-sampled matrix is processed to improved its spectrum properties and thus lead to an efficient algorithm, but just in the case when n is much larger than p. Besides, the comparison to more similar techniques would be worthwhile. For example, the algorithm in [Mar10] also sub-samples the Hessian and uses regularization ("damping"). However, in [Mar10] the approximation of the Hessian is not inverted exactly but approximated since only its application to the gradient is needed. It remains unclear whether this could give better results even for the relatively small values of p considered in this paper and whether this additional inexactness influences the convergence rate results. Similarly, [BHNS14] might perform better than the BFGS algorithm used in the experiments.