Subsampled Hessian Newton Methods for Supervised Learning

EE5121: Convex Optimization (Jul-Nov '23)

Group 1: Haricharan B (EP21B015) & Akshay Dharmaraj C (EP20B007)

Abstract

The **Newton Method** for minimizing objective functions converges quadratically, but is **computationally expensive** due to the calculation of the Hessian. We use a subset of data to calculate a **Hessian approximation**, and adjust the search direction to be a weighted sum of the approximate Newton directions of the current and previous iterations.

Introduction

In supervised learning, the cost function is the average of the cost of data-points $(x_i, y_i) \in \mathcal{X}$, as follows: $F(w; x, y) = \frac{1}{m}\sum_{i=1}^{m} f(w; x_i, y_i) + Regularization.$

Logistic loss function for binary classification

$$F(w; x, y) = \frac{C}{l}\sum_{i=1}^{l} \log\left(1 + e^{-y_i w^T x_i}\right) + \frac{1}{2}w^T w$$

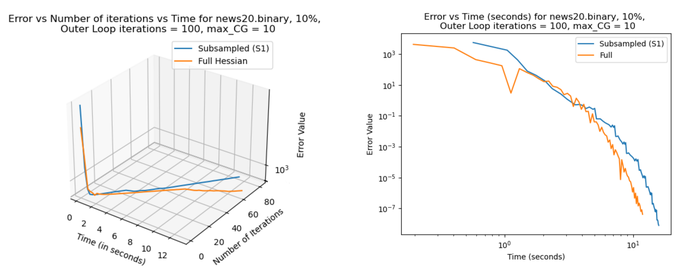
The descent direction p_k is obtained by solving $\nabla^2 F_k p_k = -\nabla F_k$ using the **Conjugate Gradient** Method to find p, but we replace the Hessian $\nabla^2 F_k$ by a stochastic approximation from a set S_k:

$$H_{S_k} = \frac{1}{|S_k|}\sum_{i \in S_k} \nabla^2 f(w; x_i, y_i)$$

Previous Work (S1)

Richard H. Byrd *et al.* (2015) proposed using a **random subsample** of some percentage S_k of the data-points to **compute the Hessian** at each step.

Find approx. Hessian → CG to find direction → Backtracking line search



This model (S1) **performs similar or worse** than the Exact Hessian on the news20 data (described later), because:

1. Running time ≈ NumIterations × TimePerIteration
2. The descent direction proposed here is **sub-optimal due to sampling**
3. Takes **greater number of iterations** to converge

The model (S2 & S3)

Chien-Chih Wang *et al.* (2019) suggest the following:

- Picking a **better initial value of** α, i.e. $\alpha_k = -\frac{\nabla f(w_k)^T d_k}{d_k^T H_{S_k} d_k}$ (S2)
- Pick the descent direction to be a **linear combination of current and previous Newton's directions**: $p_k = \beta_{k1} d_k + \beta_{k2} \bar{d}_k$, where $\bar{d}_k = d_{k-1}$. We find β_{k1} and β_{k2} by exact line search (S3)

We calculate β_{k1} and β_{k2} by setting $\nabla_\beta f(\beta) = 0$.

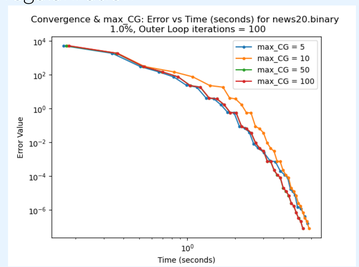
Intuition on why $\bar{d}_k = d_{k-1}$ works: By using a linear combination as above, we use **second-order information**, which is better than what we used before!

The Algorithm

Initialize Weights $w \leftarrow 0$, CG max limit max_{CG}, initial sample $S_0 \subseteq \mathcal{X}$
while $\nabla J_X(w_k) \neq 0$ **do**
  Evaluate $J_X(w_k), \nabla J_X(w_k)$
  Solve $\nabla^2 J_{S_k}(w_k) d_k = -\nabla^2 J_{X_k}(w_k)$ using Conjugate gradient method.
  $\bar{d}_k \leftarrow d_{k-1}$
  $p_k \leftarrow \beta_{k1} d_k + \beta_{k2} \bar{d}_k$, with β calculated exactly
  Update $w_{k+1} \leftarrow w_k + \alpha_k p_k$, with α_k as above.
  Choose a new S_{k+1} stochastically
end while

Simulation Results on Logistic Regression

Datasets: news20 & phishing, two different binary classification datasets tackled using logistic regression with L2-regularization

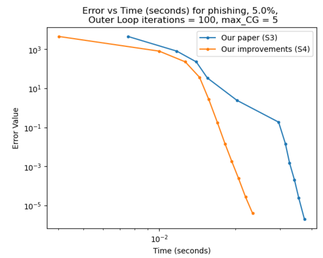


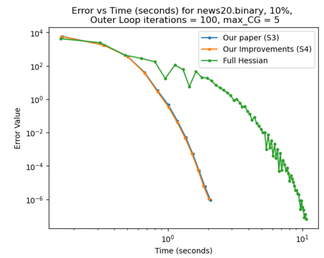
Replicating the results in the paper, we draw the following conclusions:

1. This model (S3) **performs better** than just subsampling naively (S1)
2. This model converges **more smoothly** than other methods
3. We observe the **strong correlation** between max_{CG} **and convergence**

Further Improvements and Simulations (S4)

Instead of the descent direction $\bar{d}_k = d_{k-1}$, $\bar{d}_k = -\nabla F_k$ is suggested in the paper, leading to the descent direction being a **linear combination** of the **steepest descent direction and current approximate Newton direction**. We find this gradient exactly, *not* stochastically.





- **phishing dataset:** S4 produces better results, believed to be because number of features << data-points
- **news20 dataset:** S4 produces similar results, believed to be because number of features ≈ data-points

Conclusions

- **Superiority** of this model S3 over the prior model S1 and the Full Hessian, showing **20%-40% reduction** in **running time**
- The **percentage** of the dataset used in the stochastic approximation is affected strongly max_{CG}, i.e. percentage $\propto \frac{1}{max_{CG}}$
- **Proof of concept** of the suggestion, by simulating the gradient in the descent direction in S4, thus validating the hypothesis in the paper

Why does all this matter?

This paper improves the machinery to work on answering **yes-no questions** using logistic regression in medical screening, quality control in industries, and so on. It also tackles the **theoretical aspect**, improving our knowledge repository on optimization techniques using the Newton's method.

References

[1] Richard H. Byrd, Gillian M. Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 2011.

[2] Chien-Chih Wang, Chun-Heng Huang, and Chih-Jen Lin. Subsampled hessian newton methods for supervised learning. *Neural Computation*, 2015.