International Core Journal of Engineering Volume 7 Issue 9, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0051

Apply Learned Sketch to Iterative Hessian Sketch

Jiangwei Chen[1], Bai Lin Ji[2](Canada), Yuankai Sun[3], Meier Chen[4]

[1]School of Spatial Information and Digital Technology, Wuhan University, Wuhan, 430072, China;

[2]School of Sentinel Secondary School, West Vancouver, V7S 2R2, Canada;

[3]School of Computer Science, Wuhan University, Wuhan, 430072, China;

[4]School of Engineering and Computer Science, University of the Pacific Stockton, 95211, USA.

Abstract

The goal of this experiment is to study Randomize Sketch Methods and combine the Learned Sketch with Hessian Sketch to produce optimization results that are both fast and accurate. Assess different types of Random Sketch Methods then take the most accurate one and apply Iterative Hessian Sketch method to minimize the function: $\mathbf{x_{OPT}} = \mathbf{argmin_{x\epsilon C}} \frac{1}{2} \| \mathbf{Ax} - \mathbf{b} \|_2^2$, The new method Iterative Hessian Sketch, which uses a random projection dimension proportional to the statistical complexity of the least-squares minimizer. This method is tested both on unconstrained least square problem and LASSO. Finally compare the test results to other famous Sketch Methods in terms of accuracy.

Keywords

Random Projection; Lasso; Convex Optimization; Count Sketch; Learned Sketch; Iterative Hessian Sketching.

1. Introduction

Optimizing a problem with limited complexity and fast procedure is achieved through many means. One aspect of this includes the Iterative Hessian Sketch, and the *Learned Sketch* which will be the focus of this experiment. The goal is to optimize Unconstrained Least Squares problem with Learned Sketch and then apply Iterative Hessian Sketch to achieve fast and accurate linear regression [1].

$$x_{OPT} = argmin_{x \epsilon C} \frac{1}{2} ||Ax - b||_2^2$$

2. Experiments

In normal IHS, the S matrix is generated by several random distributions. IHS exploits random approximations to $A^T A$ by the quadratic form $(SA)^T SA$. IHS uses a new randomly generated sketch in each step [2]. Due to its randomness, this leads that its acceleration effect on the calculation is also random. Therefore, the question arises whether it's possible to use Learn Sketch and then apply IHS in each step of iteration to produce a more stable calculation acceleration effect.

The Iterative Hessian Sketching (IHS) approach exploits the quadratic program formulation of and uses random projections to accelerate tons of computations in the problem setup [3]. IHS samples a random linear transformation $S \in R^{m \times n}$ from a sufficiently well-behaved distribution of matrices with $m << n$:

$$x^{t+1} = argmin_{x \in C} \frac{1}{2} ||S^{t+1} A(x - x^t)||_2^2 - < A^T (b - Ax^t), x - x^t > \quad (1)$$

There are several ways to generate a random sketch:

Gaussian Sketch: Sample a matrix G whose entries are iid normal, $G_{ij} \sim N(0,1)$, and define the sketch matrix S by scaling $S = G/\sqrt{m}$.

Count Sketch: Initialise $S = 0_{m,n}$ and for every column i of S choose a row h(i) uniformly at random. Set $S_{h(i),i}$ to either +1 or -1 with equal probability [4].

Sparse Jogson-Lindentrauss Transform (SJLT): The sparse embedding S with column sparsity (number of nonzeros per column) parameter s is constructed by row-wise concatenating s independent CountSketch transforms, each of dimension m/s*n.

For learning sketch, the Learn Sketch value is kept the same in the Sketch Vector through Mini-Batch, one of Gradient-Descent Algorithms

"Algorithm 1. LEARN-SKETCH: gradient-descent algorithnm for optimizing sketch value"

Require: $A_{\text{train}} = \{A_1, \ \ldots \ , \ A_{N_{train}}\}$ $where \ A_i \in R^{n\times d}$, learning rate α

1: Initialize $\vec{p}, \vec{v}$ randomly

2: for $i = 1$ to num_grad_steps do

3: form S using $\vec{p}, \vec{v}$

4: sample $batch$ A_{batch} from A_{train}

5: $\vec{v} \leftarrow \vec{v} - \alpha \dfrac{\partial L(S, A_{batch})}{\partial \vec{v}}$

6: end for

The loss function is defined as:

$$\text{loss}(S) = \| S(Ax-b) \|_2$$

Then we can get the learned sketch S* to apply in IHS formula (1).

After many iterations, we get the optimal x*.

3. Main Results

Two experiments were conducted in this study. The first experiment is where we apply the Learned Sketch method to an unconstrained least square regression, and the other one is applying Learned Sketch method to the lasso regression. The prediction error in relation to the number of iterations. In our test cases we choose the size of our data to be m where $\mathrm{m} = \mathrm{y} \times \mathrm{d}$ ($\mathrm{y} = 5, 10, 15, \ldots$). As shown in figure one, the test results show that out of all other sketch methods, the learned Sketch 10 and the Learned Sketch 15 have the lowest predicted error. Figure two is a closer comparison between the results. Strangely, the Lasso regression did not perform as well as expected. The learned sketch method had the second highest prediction error, just behind Count Sketch. This is not an ideal outcome since the goal is to minimize the error [5].

Volume 7 Issue 9, 2021
DOI: 10.6919/ICJE.202109_7(9).0051

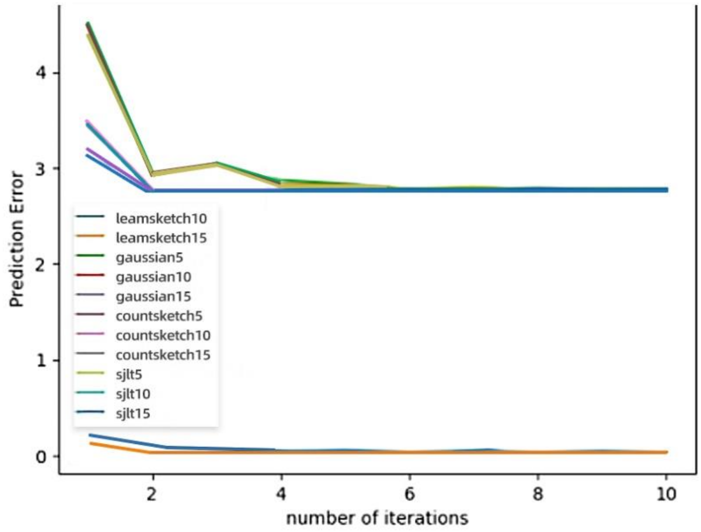

Figure 1. After the same number of iterations, LearnSketch methods perform much better than others on unconstrained regression problem.

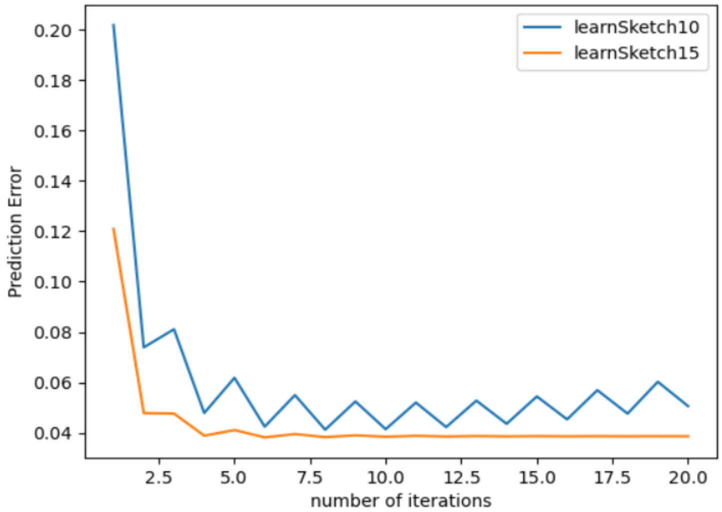

Figure 2. Prediction error of LearnSketch10 and LearnSketch15 of each iteration on unconstrained regression.

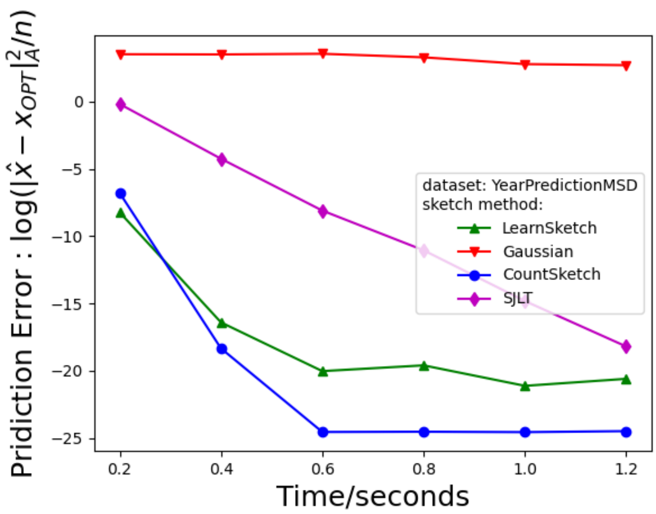

Figure 3. Prediction error of LearnSketch and other method vs real runtime on LASSO problem.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202109_7(9).0051

4. Conclusion

Despite the effort to create a fast and accurate optimization method, the result of this experiment was not ideal. The unconstrained regression worked as expected whereas Lasso Regression had a deficient result. This could be the outcome of insufficient data, and iteration. Perhaps, in the future Lasso Regression problems can be solved using the algorithm listed in this paper.

Acknowledgments

Thanks to the CIS teachers and teaching assistants for their help during this period. At the same time, we would also like to thank the professor for providing us with a large amount of information, as well as Suggestions for revision.

References

[1] Cormode G and Dickens C. Public on October 30, 2019. Iterative Hessian Sketch in Input Sparsity. Machine Learning (stat.ML) p3-4.

[2] Pilanci M and Martin J. Public on November 3, 2014. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. P11-12.

[3] Yi Li, Honghao Lin, David P. Woodruff (2021). Learning-Augmented Sketches for Hessians. https:// openreview. net/forum?id=Lnomatc-1s

[4] Simin Liu, Tianrui Liu, Ali Vakilian, Yulin Wan, David P. Woodruff (July 20, 2020). Learned Sketches for Randomized Numerical Linear Algebra. P3-4. https://arxiv.org/abs/2007.09890

[5] Meifan Zhang, Hongzhi Wang, Jianzhong Li, Hong Gao. Learned sketches for frequency estimation. Inf. Sci. 507: 365-385 (2020) https://dblp.org/rec/journals/isci/ZhangWLG20.html