ISSN: 2710-0588 DOI: 10.29556/FSE.202106_1(3).0014

Learned Sketch and Iterative Hessian Sketch in Linear Algebra

Meier Chen[1], Kangyi Jiang[2]

[1]School of University of the pacific, Stockton, 3601 pacific Ave, CA, USA.

[2]Nanjing Foreign Language School-British Columbia Academy, No.30 Beijing East Road, Nanjing, China.

Abstract

Our goal is to introduce and use two sketching methods, learned sketch and iterative hessian sketch to find a fast way to solve the problem, while still providing fast and accurate data. We also use different types of random sketch methods in which we then take the most accurate one and apply the Iterative Hessian Sketch method to minimize the function: XOPT = argminx ∈ C 1/2||AX-b||2. The new method Iterative Hessian Sketch uses a random projection dimension proportional to the statistical complexity of the least-squares minimizer.

Keywords

Regression Analysis; Count Sketch; Learn Sketch Optimization; Iterative Hessian Sketching (IHS).

1. Introduction

Optimizing a problem with limited complexity and the fast procedure is achieved through many means. One aspect of this includes the literacy Hessian sketch optimization, and the learned sketch which are methods we will be focusing on in this paper. We will introduce the difference between literacy Hessian sketch and learn sketch and introduce some of these branches. And count sketches to reduce latitude by changing the method of proof, and we used Learn sketch to increase the precision so that the value of Iterative Hessian Sketching" (IHS) can be calculated faster.

2. Material and methods:

2.1. Count Sketch

By our construction, the COUNT-MIN Sketch produces a ˜f such that for each fixed j, fj ≤ ˜fj ≤ fj +ε||f−j ||1.[1]The COUNT Sketch, which was actually defined earlier than the simpler COUNT-MIN Sketch, will give a more accurate approximation in that the error will be based on the `2 norm, rather than the `1 norm of f. This will allow us to find (γ, 2) heavy hitters rather than just (γ, 1)-heavy hitters. This streaming algorithm instantiates the following framework.

Find a randomized streaming algorithm whose output is a random variable and it includes the desired expectation[2]. We usually will get high variance, so we need to reduce the variance and turn it into separate samples, and then put them together to get output.

For example

a, b, c, a, b, a .

Then, a: 3, b: 2, c: 1

Let's suppose that we have just one. There are eight possible functions h : {a, b, c} -> {+1, -1}.

By the linearity of expectation, we have E[h(a) X] = E[h(a) Y] + E[h(a) Z] .

ISSN: 2710-0588 DOI: 10.29556/FSE.202106_1(3).0014

```
h  |
abc |  X = counter
----+--------------
+++| +3 +2 +1 =  6
++- | +3 +2 -1 =  4
+--  | +3 -2 -1 =  0
+-+ | +3 -2 +1 =  2
--+  | -3 -2 +1 = -4
 --- | -3 -2 -1 = -6
 -+- | -3 +2 -1 = -2
-++ | -3 +2 +1 =  0
```

calculate expectations

```
               (6 + 4 + 0 + 2) - (-4 + -6 + -2 + 0)
E[h(a) X] = ------------------------------------ = 24/8
```

2.2. Learn Sketch algorithm.

We optimize our learned sparse sketch (SL) using gradient descent, where L(S, A) = kA(SA) +(SB) − Bk 2 F is the regression objective. Then, we find a way to keep the worst-case guarantees of random CS while taking advantage of SL when possible . Therefore, we can efficiently compare the quality of the solution between using random CS (SR) and using SL. Then, we will know which one is better.

Learn-Sketch: gradient-descent algorithm for optimizing sketch values

Require: Atrain = {A1, ..., ANtrain } where Ai ∈ R

n×d, learning rate α

1: Initialize ~p, ~v (defined in 2.1) randomly or use Alg. 6 for ~p

2: for i = 1 to num grad steps do

3: form S using ~v, ~p

4: sample batch Abatch from Atrain

5: ~v ← ~v − α

∂L(S,Abatch)

∂~v

6: end for

2.3. Iterative Hessian Sketching(IHS)

In normal IHS, the S matrix is generated by several random distributions.[3]Iterative Hessian Sketching exploits random approximations to analog telephone adapters by the quadratic form TSA.[3]IHS uses a new randomly generated sketch in each step. Due to its randomness, this leads to its acceleration effect on the calculation is also random[4]. Therefore, we wonder whether we can learn sketch and use the learned sketch 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 (1) and uses random projections to accelerate tons of computations in the problem setup[5]. IHS samples a random linear transformation S ∈ Rm×n from a sufficiently well-behaved distribution of matrices with m << n

ISSN: 2710-0588 DOI: 10.29556/FSE.202106_1(3).0014

$$x^{t+1} = \arg\min_{x \in C} \tfrac{1}{2} \| S^{t+1} A(x - x^t) \|_2^2 - \langle A^T (b - Ax^t), x - x^t \rangle$$

3. Conclusion

We introduce some knowledge about count sketch, learn sketch, and iterative hessian sketch. In this lab, I apply the Learned Sketch method to the lasso regression and the prediction error in relation to the number of iterations. (we also apply the Learned Sketch method to an unconstrained least square regression)

In our test cases, we choose the size of our data to be where. In our test cases, we choose the size of our data to be m where m=y×d (y=5,10,15,...). 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 where we take a closer look at the two learned sketches and compare the results.

There is the test:

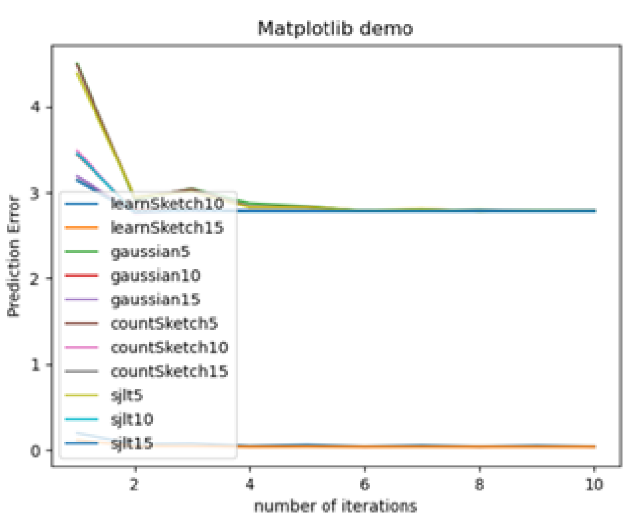

Figure 1. Error range between learn sketch and iterative hessian sketch

This figure shows the error range between learn sketch and iterative hessian sketch. And complete it with the gauss sketch. Then, you will find it is in the same range and prove the theory is worked.

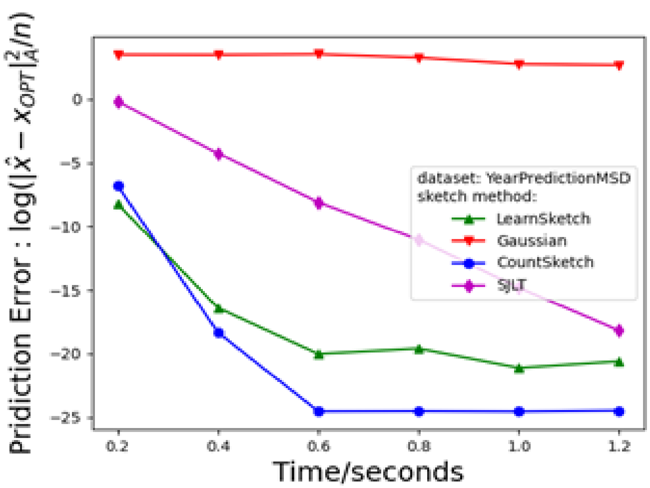

Figure 2. User running time

ISSN: 2710-0588 DOI: 10.29556/FSE.202106_1(3).0014

This figure shows the time when the user running the learnSketch, countsketch.

The figure 1 and figure 2 shows the difference between these sketches and learn sketch has more error than count sketch and prove in this situation, the count sketch is better than the learn sketch.

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] Ben-Eliezer, O., Jayaram, R., Woodruff, D. P., & Yogev, E. (2020, June). A framework for adversarially robust streaming algorithms. In Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (pp. 63-80).

[3] Simon L and Woodruff D. Public on July 20, 2020. On Learned Sketches for Randomized Numerical Linear Algebra. P2-4.

[4] Ozaslan, I. K., Pilanci, M., & Arikan, O. (2019, May). Iterative hessian sketch with momentum. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7470-7474). IEEE.

[5] Pilanci, M., & Wainwright, M. J. (2016). Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. The Journal of Machine Learning Research, 17(1), 1842-1879.