Automatic Differentiation of Sketched Regression

Hang Liao

CMU-CS-20-119

August 2020

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
David P. Woodruff, Chair
Zico Kolter

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: Theory, Deep Learning, Automatic Differentiation, Sketching, Least Squares Regression

Abstract

In this work, we explore the possibility of applying sketching, or dimensionality reduction, in the least squares regression (LLS) problem in differentiable programming settings. To motivate automatic differentiation (AD) for systems with a sketched regression component, we need to answer the following questions: do we yield similar derivatives (AD transformations) in differentiable programming systems with LLS and sketched LLS? In practice, does a system containing sketched LLS converge faster than the same system with LLS in training? How close are the results after convergence? To answer them, we first provide a bound on the operator norm of a sketched pseudoinverse matrix product, which is useful when analyzing the derivatives of sketched regression. We then give analysis on the approximation errors of derivatives in two proposed ways of sketched regression. Finally, we run experiments on both synthetic and real-world datasets to test the performance of our sketching methods.

Acknowledgments

I would like to give very special thanks to my advisor, Dr. David Woodruff, for introducing me to the research area of sketching and streaming algorithms, and for his advice, guidance and encouragement. It was very fortunate to work with him.

I would like to thank Dr. Vamsi Potluru for his guidance and patience while working on the subject of sketched regression in differentiable programming setting. I would like to thank both Dr. Vamsi Potluru and Dr. Barak Pearlmutter for their advice and suggestions on the AISTATS paper's video talk. I would also like to thank Dr. Zico Kolter for his helpful lectures and assignments on automatic differentiation.

Special thanks to Dr. Anupam Gupta, Dr. William Hrusa, Dr. Vamsi Potluru, Dr. Wilfried Sieg, and Dr. David Woodruff for supporting my graduate school applications. Special thanks to Dr. David Woodruff and Dr. Zico Kolter for being part of my thesis committee and for their invaluable advice on this document.

Last but foremost, I would not have the opportunity to study at CMU without the care and support from my parents. I am very grateful to my family for their continuous support and encouragement during my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

[1] A sketch is a sub-linear space data structure that can answer certain types of queries in the original data with approximation guarantees. Since the space usage of the sketch is sub-linear to the input size, a lot of times we benefit from a small sketch size and therefore achieve run time acceleration. The method to convert the original data into a sketch is called sketching.

Consider the well-known linear least squares regression (LLS) problem. A least squares regression takes in a matrix A of size $n \times d$ ($n \gg d$) and a vector b of size $n \times 1$, and aims to find $\text{argmin}_y \left\|Ay - b\right\|_2$. For a sketched least squares regression (LLS$_S$), we convert LLS to $\text{argmin}_y \left\|SAy - Sb\right\|_2$ by multiplying a sketching matrix S[2]. Suppose S has size $m \times n$ with $m < n$. The original problem takes in a matrix of size $n \times d$ and a vector of size n, whereas the sketched problem involves a matrix of size $m \times d$ and a vector of size m. Note the problem size becomes a lot smaller given SA can be calculated fast and $m \ll n$. Moreover, for sketching matrices that form $(1 \pm \epsilon)$ ℓ_2 subspace embeddings, it is guaranteed that $\|Ay_S^* - b\|_2 = (1 \pm \epsilon)\|Ay^* - b\|_2$ [3], where y^* is the solution to the unsketched problem and y_S^* is the solution to the sketched problem. There has been extensive research on the type and size of the sketching matrix that can be used as the sketching matrix of LLS$_S$. See [15] for a survey.

In this paper, however, we are not going to design new sketching matrices. Instead, we mainly use the sketching matrix property as a black box and try to answer the question that whether LLS$_S$ can replace LLS in a differentiable programming system which internally uses LLS. Although LLS$_S$ provides a good approximation to $\text{argmin}_y \left\|Ay - b\right\|_2$, the derivatives are not necessarily close to those of LLS. As a result, after training with gradient descent method, a neural network containing LLS$_S$ might converge differently and yield higher test loss compared with the same neural network with LLS.

To address these concerns, we first provide necessary background on sketching and automatic differentiation. After exploring a novel bound on a sketched pseudoinverse matrix product, we analyze the two proposed ways of sketched regression, namely the "Regular Sketch" and the "Partial Sketch", how the forward and reverse AD transformations of LLS$_S$ can be expressed, and the approximation error bound on the transformations. Finally, we test the actual loss and running

[1]This work is based on [8].

[2]Informally, we say a matrix is a sketching matrix if we apply the matrix and the input with certain operations to get the sketch.

[3]We use $\|Sx\|_2^2 = (1 \pm \epsilon)\|x\|_2^2$ to denote $(1 - \epsilon)\|x\|_2^2 \leq \|Sx\|_2^2 \leq (1 + \epsilon)\|x\|_2^2$.

speed of our sketched regression in the diffentiable programming context on both synthetic and real-world datasets to see if the sketching methods enjoy the favorable accuracy and complexity.

Chapter 2

Preliminaries

2.1 Sketching in Least Squares Regression

We use $\|\cdot\|_2$ as the operator norm for matrices, $\|\cdot\|_F$ as the Frobenius norm for matrices. and $\|\cdot\|$, $\|\cdot\|_2$ interchangeably as the ℓ_2 norm for vectors.

Definition 2.1.1. *(Least Squares Regression, or Linear Regression, ℓ_2 Regression) Given an $n \times d$ matrix A with $n \gg d$, an $n \times 1$ vector b, output a vector x that minimizes $\|Ax - b\|_2$.*

Throughout the document, we assume $n \gg d$ with $\text{rank}(A) = d$. We can view each row of the matrix A as a data point and each column as a feature. For real world data, typically there are far more number of data points than the number of features. This motivates us to only consider the case when $n > d$.

It is well-known that for a matrix A, $\text{argmin}_x \|Ax - b\|_2 = A^+ b$ where A^+ is the pseudoinverse of A. If A has full column rank, $A^+ = (A^\mathrm{T} A)^{-1} A^\mathrm{T}$.

Definition 2.1.2. *(ℓ_2-Subspace Embedding) Let V be a fixed d-dimensional subspace in $\mathbb{R}^n$. A matrix S is a $(1 \pm \epsilon)$ ℓ_2-subspace embedding for V if for all $x \in V$, $\|Sx\|_2^2 = (1 \pm \epsilon)\|x\|_2^2$. Equivalently, fix A to be an $n \times d$ matrix with column space V, an $m \times n$ matrix S is a $(1 \pm \epsilon)$ ℓ_2-subspace embedding for V if for all $x \in \mathbb{R}^d$, $\|SAx\|_2^2 = (1 \pm \epsilon)\|Ax\|_2^2$.*

Since we are working in the ℓ_2 norm throughout the document, we sometimes omit the ℓ_2 for simplicity. When we say a matrix S is a subspace embedding for a matrix A, we mean S is a $(1 \pm \epsilon)$ ℓ_2-subspace embedding for the column space of A.

A limitation with the subspace embedding is that we need to know the subspace we are trying to embed beforehand. This motivates the following definition.

Definition 2.1.3. *(Oblivious ℓ_2-Subspace Embedding) A matrix S is a (d, ϵ, δ) oblivious ℓ_2-subspace embedding if for any fixed d-dimensional subspace $V \in \mathbb{R}^n$, for all $x \in V$, $\|Sx\|_2^2 = (1 \pm \epsilon)\|x\|_2^2$ with probability at least $1 - \delta$. Equivalently, a matrix S is a (d, ϵ, δ) oblivious ℓ_2-subspace embedding if for any $n \times d$ matrix A, for all $x \in \mathbb{R}^d$, $\|SAx\|_2^2 = (1 \pm \epsilon)\|Ax\|_2^2$ with probability at least $1 - \delta$.*

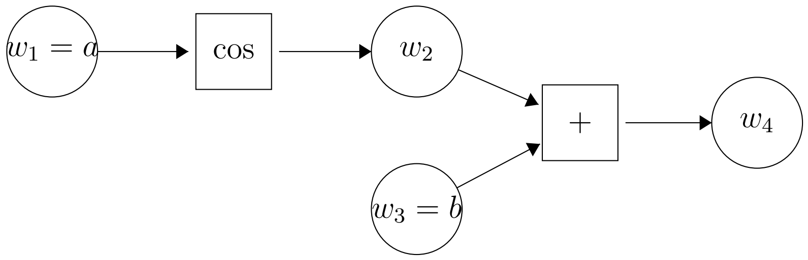
Let U be a matrix with orthonormal columns and v be a unit vector. $\|SUv\|_2^2 = (1 \pm \epsilon)\|Uv\|_2^2$ implies $(SUv)^\mathrm{T}(SUv) - (Uv)^\mathrm{T}(Uv) \leq \epsilon$ and can be further simplified to $\|U^\mathrm{T} S^\mathrm{T} SU - I\|_2 < \epsilon$. This also implies the singular values of $U^\mathrm{T} S^\mathrm{T} SU$ are in $[1 - \epsilon, 1 + \epsilon]$, and the singular values of SU are in $[1 - \epsilon, 1 + \epsilon]$ for small ϵ. These properties are useful in later chapters.

We use OSE to denote the oblivious ℓ_2-subspace embedding for short. In literature (and this document), sometimes "oblivious" are dropped for convenience, and "subspace embedding" is used interchangeably with "oblivious ℓ_2-subspace embedding".

Definition 2.1.4. *(Singular Value Decomposition) Any $n \times d$ matrix A can be decomposed as $U\Sigma V^{\mathrm{T}}$, where U is an $n \times d$ matrix with orthonormal columns, Σ is a $d \times d$ diagonal matrix with singular values on its diagonal, V is a $d \times d$ orthogonal matrix.*

2.2 Notations for Automatic Differentiation

- Automatic Differentiation, or AD for short, refers to the derivative evaluation algorithms. Forward and reverse mode AD transformations refer to the actual derivative(s).
- Standard notation from AD community is used. Consider an independent variable x and a dependent variable y. We use $\dot{x}$ for an infinitesimal perturbation of x, and $\dot{y}$ for $\frac{\partial y}{\partial x}$ (notation used in forward mode AD). We use $\bar{x}$ for $\frac{\partial y}{\partial x}$ (notation used in reverse mode AD). See [1] for a thorough survey on automatic differentiation.
- We can write a computation process as a directed acyclic graph with circles representing variables and squares representing operators. Say we compute $g(x, y) = f(x) + y$ where $f(z) = \cos z$. Then the computation of $g(a, b)$ for values a, b can be represented by the following graph:



  Computation graphs can be helpful in the computation of forward and reverse mode AD transformations.

2.3 The Problem

Formally, given an $n \times d$ matrix A (with $n \gg d$), an $n \times 1$ vector b and an $m \times n$ sketching matrix S, we want to show a tight bound on the sketched regression's approximation error of forward and reverse mode AD transformations.

2.4 Motivation

Sketching is a useful technique to accelerate linear regression. However, to our best knowledge, the relationship between the derivatives of the sketched and unsketched linear regression is yet unknown. At the same time, there is burgeoning interest in extending deep learning (aka differentiable programming) to allow more complicated building boxes to appear inside the computational process being differentiated. This includes optimization of various sorts (learning-to-learn, bi-level optimization), AD of fixed point computations [2] recently rediscovered in the ML community [7] and even AD of discrete optimization processes [10]. In a nutshell, if a subroutine is useful, people will want to use it in programs they write, and it is increasingly desired that we are able to differentiate the programs we write. This has resulted in a systematic effort to explore how to efficiently calculate appropriate derivatives of a variety of numerical processes. Here, we consider differentiating a linear regression subroutine—in particular, linear regression computed using sketching methods.

2.5 Related Work

There is a recent work on sketching for speeding up distributed communication of gradients [6]. More specifically, the work focuses on shortening the run time of distributed stochastic gradient descent methods using CountSketch matrices.

In addition to gradient sketching, Hessian sketching has also been considered since second-order methods have better convergence rate compared with first-order methods. For instance, [5] is a recent work on Hessian sketching for serverless systems.

Instead of designing a general scheme of solving various kinds of problems with the traditional “sketch and solve” approach, our work focuses on the least squares regression, explicitly reveals the derivative computation steps, and studies the errors resulted from sketching in derivatives and the overall network.

Chapter 3

A bound on the Opertor Norm of Sketched Pseudoinverse

We need theorem 3.0.1 for proving theorem 3.0.2.

Theorem 3.0.1. *(By Cohen et al. [3]) Let $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times p}$ with $n \gg d, n \gg p$. For sketching matrix S drawn from $(\epsilon, \delta, 2k)$-OSE,*

$$\|A^{\mathrm{T}}S^{\mathrm{T}}SB - A^{\mathrm{T}}B\|_2 \leq \epsilon\sqrt{(\|A\|_2^2 + \|A\|_F^2/k)(\|B\|_2^2 + \|B\|_F^2/k)}$$

holds with probability $1-\delta$.

Theorem 3.0.2 says the pseudoinverse and sketched pseudoinverse terms cannot differ by much in terms of the opertor norm, and the difference is small compared to the norm of the pseudoinverse itself. Since we apply sketching on the least squares regression problem, we will encounter pseudoinverse and sketched pseudoinverse terms in derivative computations. In particular, terms like $(SA)^{+}(SB)$, where B is not a vector, come into the picture. Under these circumstances, the bound in theorem 3.0.2 becomes useful as we need to bound the difference between the unsketched and sketched pseudoinverse terms. This result also extends previous results in which B is considered as a vector [12] [4].

The intuition for proving the bound is that after singular value decomposition, $(SA)^{+}SB$ and $A^{+}B$ have similar structures[1]. For the parts of the structures that differ between the two terms, there are enough tools to bound them. The former term has a $U^{\mathrm{T}}S^{\mathrm{T}}SU$ factor and the latter term has a corresponding factor I. It is easy to bound the difference as $\|U^{\mathrm{T}}S^{\mathrm{T}}SU - I\|_2 \leq \epsilon$. Also note the former term consists of a $U^{\mathrm{T}}S^{\mathrm{T}}SB$ factor, and in the latter term the corresponding factor is $U^{\mathrm{T}}B$. These two terms are exactly in the form of the left-hand side of theorem 3.0.1.

Theorem 3.0.2. *Let $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{n\times p}$ with $n \gg d, n \gg p$. Let S be a matrix drawn from a distribution D with $(\epsilon, \delta, 2k)$-OSE property for some $k \geq d$. With probability $1-\delta$,*

$$\|(SA)^{+}SB\|_2 \leq \|A^{+}B\|_2 + O(\epsilon)\sqrt{(1 + d/k)(\|B\|_2^2 + \|B\|_F^2/k)}/\sigma_{min}(A)$$

[1] $(SA)^{+}SB = V\Sigma^{-1}(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}S^{\mathrm{T}}SB$ and $A^{+}B = V\Sigma^{-1}U^{\mathrm{T}}B$.

Proof. Let $U\Sigma V^{\mathrm{T}}$ be the SVD of A. By Triangle Inequality,

$$\begin{aligned}
\|(SA)^{+}SB\|_2 &= \|(SA)^{+}SB - A^{+}B + A^{+}B\|_2 \\
&\leq \|A^{+}B\|_2 + \|(SA)^{+}SB - A^{+}B\|_2 \\
&= \|A^{+}B\|_2 + \|V\Sigma^{-1}(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}S^{\mathrm{T}}SB - V\Sigma^{-1}U^{\mathrm{T}}B\|_2 \\
&= \|A^{+}B\|_2 + \|V\Sigma^{-1}\|_2\|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}S^{\mathrm{T}}SB - U^{\mathrm{T}}B\|_2 \\
&\leq \|A^{+}B\|_2 + \|\Sigma^{-1}\|_2\|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}S^{\mathrm{T}}SB - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}B \\
&\quad + (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}U^{\mathrm{T}}B - U^{\mathrm{T}}B\|_2 \\
&\leq \|A^{+}B\|_2 + \|\Sigma^{-1}\|_2(\|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|U^{\mathrm{T}}S^{\mathrm{T}}SB - U^{\mathrm{T}}B\|_2 \\
&\quad + \|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1} - I_d\|_2\|U^{\mathrm{T}}B\|_2)
\end{aligned}$$

Recall the singular values of $U^{\mathrm{T}}S^{\mathrm{T}}SU$ are in $[1-\epsilon, 1+\epsilon]$ when S is an OSE. Therefore the singular values of $(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}$ are also roughly in $[1-\epsilon, 1+\epsilon]$ for small ϵ, and the singular values of $(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1} - I$ are in $[-\epsilon, \epsilon]$. Also by theorem 3.0.1, we have $\|U^{\mathrm{T}}S^{\mathrm{T}}SB - U^{\mathrm{T}}B\|_2 \leq \sqrt{(1+d/k)(\|B\|_2^2 + \|B\|_F^2/k)}$. Now we can continue the proof as follows:

$$\begin{aligned}
\|(SA)^{+}SB\|_2 &\leq \|A^{+}B\|_2 + \|\Sigma^{-1}\|_2(\|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|U^{\mathrm{T}}S^{\mathrm{T}}SB - U^{\mathrm{T}}B\|_2 \\
&\quad + \|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1} - I_d\|_2\|U^{\mathrm{T}}B\|_2) \\
&\leq \|A^{+}B\|_2 + \|\Sigma^{-1}\|_2(1+\epsilon)\epsilon\sqrt{(1+\frac{d}{k})(\|B\|_2^2 + \frac{\|B\|_F^2}{k})} + \epsilon\|\Sigma^{-1}\|_2\|U^{\mathrm{T}}B\|_2 \\
&\leq \|A^{+}B\|_2 + 2\epsilon\|\Sigma^{-1}\|_2\sqrt{(1+\frac{d}{k})(\|B\|_2^2 + \frac{\|B\|_F^2}{k})} + \epsilon\|\Sigma^{-1}\|_2\|B\|_2 \\
&\leq \|A^{+}B\|_2 + 3\epsilon\|\Sigma^{-1}\|_2\sqrt{(1+\frac{d}{k})(\|B\|_2^2 + \frac{\|B\|_F^2}{k})} \\
&\leq \|A^{+}B\|_2 + \frac{3\epsilon}{\sigma_{\min}(A)}\sqrt{(1+\frac{d}{k})(\|B\|_2^2 + \frac{\|B\|_F^2}{k})}
\end{aligned}$$

□

Notice that if $k/\epsilon^2 \gg d$, we have $\epsilon\sqrt{1+d/k} \ll 1$, and the term $\|B\|_2/\epsilon_{min}(A)$ roughly equals to $\|A^{+}B\|_2$, which translates to $\|(SA)^{+}SB\|_2 \leq (1+c)\|A^{+}B\|_2$ where c is a constant depends on k, d, ϵ and is small when $k/\epsilon^2 > d$. In words, we have shown that $\|(SA)^{+}SB\|_2/\|A^{+}B\|_2 \approx 1$ in practice.

We simultaneously get a lower bound on $\|(SA)^{+}SB\|_2$ as well. If we interchange $(SA)^{+}SB$ and $A^{+}B$ in the proof, as every term is in norm, we would get

$$\|A^{+}B\|_2 \leq \|(SA)^{+}SB\|_2 + O(\epsilon)\sqrt{(1+d/k)(\|B\|_2^2 + \|B\|_F^2/k)}/\sigma_{\min}(A).$$

Chapter 4

Forward and Reverse Mode AD of Sketched Regression

- **Regular Sketch:** We define the "Regular Sketch" to be the scheme that approximates $\text{argmin}_x \|Ax - b\|$ with $\text{argmin}_x \|SAx - Sb\| = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}S^{\mathrm{T}}Sb$ and computes the AD transformations accordingly for both the forward and reverse mode.
- **Partial Sketch:** We define the "Partial Sketch" to be the scheme that uses $(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}b$ as primal, and for both forward and reverse mode, sketches the $(A^{\mathrm{T}}A)^{-1}$ term only. We call it "Partial Sketch" because the derivatives yielded by this method is an approximation of the real derivative if the sketched solution is given by $(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}b$. This sketch is desirable because theoretically it still accelerates the computation, as finding $(A^{\mathrm{T}}A)^{-1}$ is the most expensive operation in calculating the AD transformations. More importantly, this sketch makes it easier to analyze the AD transformations. Sketching $A^{\mathrm{T}}A$ only is first considered by Pilanci et al. [9].

We summarize the results for the forward and reverse AD transformations.

Type	**Primal**	**Forward Transform**
Regular	$y = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$	$\dot{y} = (A^{\mathrm{T}}A)^{-1}(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b} - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y)$
"Regular Sketch"	$y_S = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}S^{\mathrm{T}}Sb$	$\dot{y}_S = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}(\dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{b} - (\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A})y_S)$
"Partial Sketch"	$y_D = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}b$	$\dot{y}_D = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b} - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y_D)$

Table 4.1: Forward Mode AD Transformations.

The computation details can be found in the following section.

Type	**Primal**	**Reverse Transform**
Regular	$y = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$	$\bar{A} = -A(A^{\mathrm{T}}A)^{-1}\bar{y}y^{\mathrm{T}} - Ay\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} + b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1}$ $\bar{b} = A(A^{\mathrm{T}}A)^{-1}\bar{y}$
"Regular Sketch"	$y_S = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S^{\mathrm{T}}b$	$\bar{A}_S = -S^{\mathrm{T}}SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_S{y_S}^{\mathrm{T}} - S^{\mathrm{T}}SAy_S\bar{y}_S{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} + S^{\mathrm{T}}Sb\bar{y}_S{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}$ $\bar{b}_S = S^{\mathrm{T}}SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_S$
"Partial Sketch"	$y_D = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}b$	$\bar{A}_D = -A(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_D{y_D}^{\mathrm{T}} - Ay_D\bar{y}_D{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} + b\bar{y}_D{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}$ $\bar{b}_D = A(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_D$

Table 4.2: Reverse Mode AD Transformations.

4.1 AD Transformations

For completeness, we include the computations of AD transformations for the least squares regression and two ways of the sketched regression.

4.1.1 Forward mode AD

Least Squares Regression

First we create a computation graph for computing $y(A, b) = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b$ as shown in figure 1. Then with matrix derivative rules, we have the following:

$$
\begin{aligned}
w_1 &= A \\
w_2 &= b \\
w_3 &= A^{\mathrm{T}} = {w_1}^{\mathrm{T}} \\
w_4 &= A^{\mathrm{T}}A = w_3 w_1 \\
w_5 &= (A^{\mathrm{T}}A)^{-1} = w_4^{-1} \\
w_6 &= A^{\mathrm{T}}b = w_3 w_2 \\
w_7 &= (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b = w_5 w_6
\end{aligned}
\quad \Longrightarrow \quad
\begin{aligned}
\dot{w}_1 &= \dot{A} \\
\dot{w}_2 &= \dot{b} \\
\dot{w}_3 &= {\dot{w}_1}^{\mathrm{T}} \\
\dot{w}_4 &= \dot{w}_3 w_1 + w_3 \dot{w}_1 \\
\dot{w}_5 &= -{w_4}^{-1}\dot{w}_4 {w_4}^{-1} \\
\dot{w}_6 &= \dot{w}_3 w_2 + w_3 \dot{w}_2 \\
\dot{w}_7 &= \dot{w}_5 w_6 + w_5 \dot{w}_6
\end{aligned}
$$

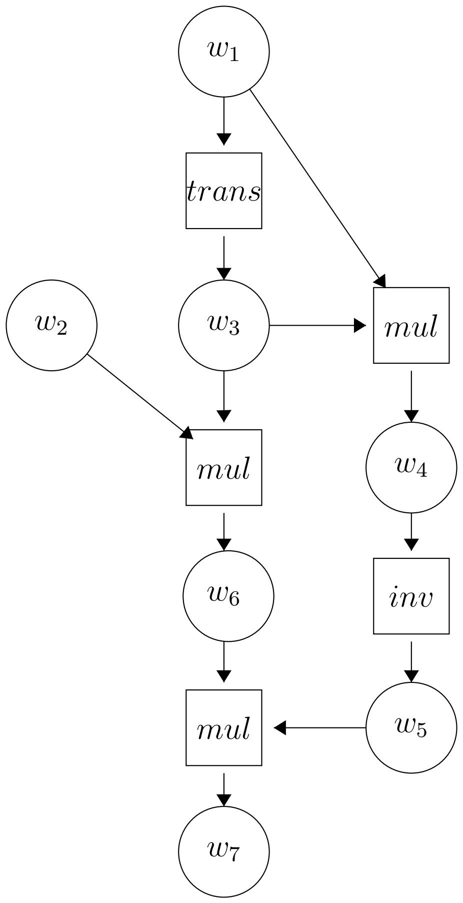


Figure 4.1: Computation Graph of Least Squares Regression

Expand $\dot{y}$:

$$\begin{aligned}
\dot{w}_7 &= \dot{w}_5 w_6 + w_5 \dot{w}_6 \\
&= -{w_4}^{-1}\dot{w}_4 {w_4}^{-1} w_6 + w_4^{-1}(\dot{w}_3 w_2 + w_3 \dot{w}_2) \\
&= -(A^{\mathrm{T}}A)^{-1}\dot{w}_4 (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b + (A^{\mathrm{T}}A)^{-1}(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) \\
&= -(A^{\mathrm{T}}A)^{-1}(\dot{w}_3 w_1 + w_3 \dot{w}_1)(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b + (A^{\mathrm{T}}A)^{-1}(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) \\
&= -(A^{\mathrm{T}}A)^{-1}[(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b - (\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b})] \\
&= (A^{\mathrm{T}}A)^{-1}[(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})w_7] \\
&= (A^{\mathrm{T}}A)^{-1}[(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y]
\end{aligned}$$

Regular Sketch

With computation graph figure 2 and matrix derivative rules, we have the following:

$$\begin{aligned}
&w_1 = A & & \dot{w}_1 = \dot{A} \\
&w_2 = b & & \dot{w}_2 = \dot{b} \\
&w_3 = S w_1 & & \dot{w}_3 = S\dot{w}_1 \\
&w_4 = S w_2 & & \dot{w}_4 = S\dot{w}_2 \\
&w_5 = {w_3}^{\mathrm{T}} & \Longrightarrow \quad & \dot{w}_5 = {\dot{w}_3}^{\mathrm{T}} = {\dot{w}_1}^{\mathrm{T}} S^{\mathrm{T}} \\
&w_6 = A^{\mathrm{T}}S^{\mathrm{T}}SA = w_5 w_3 & & \dot{w}_6 = \dot{w}_5 w_3 + w_5 \dot{w}_3 \\
&w_7 = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} = w_6^{-1} & & \dot{w}_7 = -{w_6}^{-1}\dot{w}_6 {w_6}^{-1} \\
&w_8 = A^{\mathrm{T}}S^{\mathrm{T}}Sb = w_5 w_4 & & \dot{w}_8 = \dot{w}_5 w_4 + w_5 \dot{w}_4 \\
&w_9 = w_7 w_8 & & \dot{w}_9 = \dot{w}_7 w_8 + w_7 \dot{w}_8
\end{aligned}$$

Expand $\dot{y}_S$:

$$\begin{aligned}
\dot{w}_9 &= \dot{w}_7 w_8 + w_7 \dot{w}_8 \\
&= -{w_6}^{-1}\dot{w}_6 {w_6}^{-1} w_8 + {w_6}^{-1}(\dot{w}_5 w_4 + w_5 \dot{w}_4) \\
&= -{w_6}^{-1}(\dot{w}_5 w_3 + w_5 \dot{w}_3){w_6}^{-1} w_8 + {w_6}^{-1}({\dot{w}_1}^{\mathrm{T}} S^{\mathrm{T}} w_4 + w_5 S \dot{w}_2) \\
&= -{w_6}^{-1}({\dot{w}_1}^{\mathrm{T}} S^{\mathrm{T}} w_3 + w_5 S\dot{w}_1){w_6}^{-1} w_8 + {w_6}^{-1}({\dot{w}_1}^{\mathrm{T}} S^{\mathrm{T}} w_4 + w_5 S \dot{w}_2) \\
&= {w_6}^{-1}({\dot{w}_1}^{\mathrm{T}} S^{\mathrm{T}} w_4 + w_5 S \dot{w}_2 - ({\dot{w}_1}^{\mathrm{T}} S^{\mathrm{T}} w_3 + w_5 S\dot{w}_1){w_6}^{-1} w_8) \\
&= (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\big(\dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{b} - (\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A})w_7 w_8\big) \\
&= (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\big(\dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{b} - (\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A})y_S\big)
\end{aligned}$$

Partial Sketch

The actual $\dot{y}_D$ corresponds to the primal $y_D = (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}b$ is $(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b} - (\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A})y_D)$. However, as we discussed how the "Partial Sketch" works, we simply sketch any $(A^{\mathrm{T}}A)^{-1}$ term in $\dot{y}$ to approximate $\dot{y}_D$. Recall $\dot{y} = (A^{\mathrm{T}}A)^{-1}[(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y]$. We only sketch the $A^{\mathrm{T}}A$ term to get $(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}[(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y]$ and use it as $\dot{y}_D$. See the previous discussion for more details.

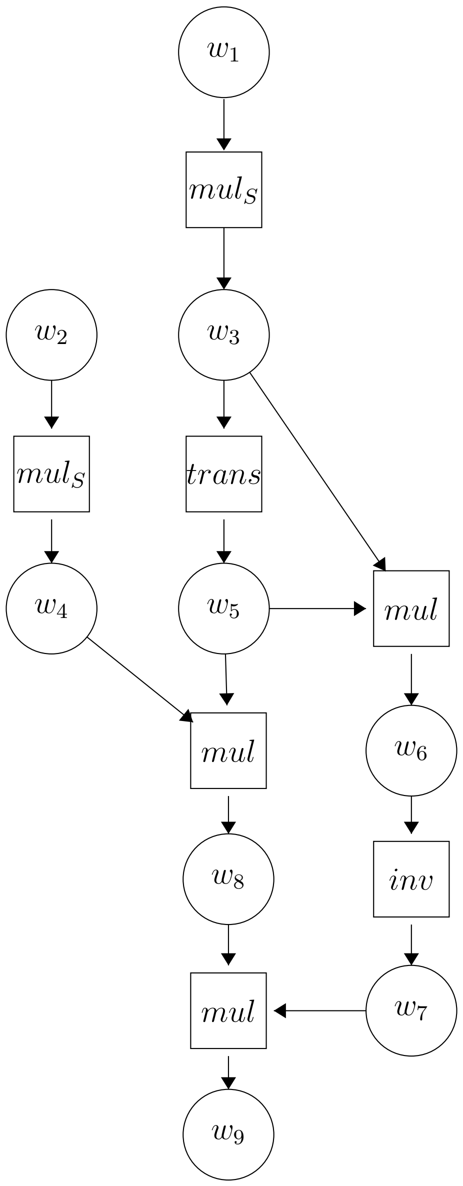


Figure 4.2: Computation Graph of “Regular Sketch”

4.1.2 Reverse Mode AD

We include the computation of AD transformations in the reverse mode for the linear regression and two ways of the sketched regression. The same computation graph for forward mode is used. For reverse mode AD transformations, we want to find both $\bar{A}$ and $\bar{b}$.

Least Squares Regression

See figure 1 for the computation graph of the least squares regression.

$$
\begin{aligned}
w_1 &= A \\
w_2 &= b \\
w_3 &= A^{\mathrm{T}} = {w_1}^{\mathrm{T}} \\
w_4 &= A^{\mathrm{T}}A = w_3 w_1 \\
w_5 &= (A^{\mathrm{T}}A)^{-1} = w_4^{-1} \\
w_6 &= A^{\mathrm{T}}b = w_3 w_2 \\
w_7 &= (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b = w_5 w_6
\end{aligned}
\quad \Longrightarrow \quad
\begin{aligned}
\bar{w_7} &= \bar{w_7} \\
\bar{w_6} &= {w_5}^{\mathrm{T}}\bar{w_7} \\
\bar{w_5} &= \bar{w_7}{w_6}^{\mathrm{T}} \\
\bar{w_4} &= -w_5\bar{w_5}w_5 \\
\bar{w_3} &= \bar{w_6}{w_2}^{\mathrm{T}} + \bar{w_4}{w_1}^{\mathrm{T}} \\
\bar{w_2} &= {w_3}^{\mathrm{T}}\bar{w_6} \\
\bar{w_1} &= {\bar{w_3}}^{\mathrm{T}} + {w_3}^{\mathrm{T}}\bar{w_4}
\end{aligned}
$$

Expand $\bar{A}$ and $\bar{b}$:

$$
\begin{aligned}
\bar{A} &= \bar{w_1} \\
&= {\bar{w_3}}^{\mathrm{T}} + {w_3}^{\mathrm{T}}\bar{w_4} \\
&= (\bar{w_6}{w_2}^{\mathrm{T}} + \bar{w_4}{w_1}^{\mathrm{T}})^{\mathrm{T}} + A(-w_5\bar{w_5}w_5) \\
&= w_2{\bar{w_6}}^{\mathrm{T}} + w_1{\bar{w_4}}^{\mathrm{T}} - Aw_5\bar{w_5}w_5 \\
&= w_2({w_5}^{\mathrm{T}}\bar{w_7})^{\mathrm{T}} + w_1(-w_5\bar{w_5}w_5)^{\mathrm{T}} - Aw_5\bar{w_5}w_5 \\
&= w_2{\bar{w_7}}^{\mathrm{T}}w_5 - w_1w_5{\bar{w_5}}^{\mathrm{T}}w_5 - Aw_5\bar{w_5}w_5 \\
&= b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}{\bar{w_5}}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}\bar{w_5}(A^{\mathrm{T}}A)^{-1} \\
&= b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}w_6{\bar{w_7}}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}\bar{w_7}{w_6}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} \\
&= b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}\bar{y}b^{\mathrm{T}}A(A^{\mathrm{T}}A)^{-1} \\
&= b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - Ay\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}\bar{y}y^{\mathrm{T}}
\end{aligned}
$$

$$
\begin{aligned}
\bar{b} &= \bar{w_2} \\
&= {w_3}^{\mathrm{T}}\bar{w_6} \\
&= A{w_5}^{\mathrm{T}}\bar{w_7} \\
&= A(A^{\mathrm{T}}A)^{-1}\bar{w_7} \\
&= A(A^{\mathrm{T}}A)^{-1}\bar{y}
\end{aligned}
$$

Regular Sketch

See figure 2 for the computation graph of the "Regular Sketch".

$$
\begin{aligned}
w_1 &= A & \qquad\qquad \bar{w_9} &= \bar{y}_S \\
w_2 &= b & \bar{w_8} &= {w_7}^{\mathrm{T}}\bar{w_9} \\
w_3 &= Sw_1 & \bar{w_7} &= \bar{w_9}{w_8}^{\mathrm{T}} \\
w_4 &= Sw_2 & \bar{w_6} &= -w_7\bar{w_7}w_7 \\
w_5 &= {w_3}^{\mathrm{T}} & \Longrightarrow \qquad \bar{w_5} &= \bar{w_8}{w_4}^{\mathrm{T}} + \bar{w_6}{w_3}^{\mathrm{T}} \\
w_6 &= A^{\mathrm{T}}S^{\mathrm{T}}SA = w_5w_3 & \bar{w_4} &= {w_5}^{\mathrm{T}}\bar{w_8} \\
w_7 &= (A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} = w_6^{-1} & \bar{w_3} &= \bar{w_5}^{\mathrm{T}} + {w_5}^{\mathrm{T}}\bar{w_6} \\
w_8 &= A^{\mathrm{T}}S^{\mathrm{T}}Sb = w_5w_4 & \bar{w_2} &= S^{\mathrm{T}}\bar{w_4} \\
w_9 &= w_7w_8 & \bar{w_1} &= S^{\mathrm{T}}\bar{w_3}
\end{aligned}
$$

Expand $\bar{A}_S$ and $\bar{b}_S$:

$$
\begin{aligned}
\bar{A}_S &= \bar{w_1} \\
&= S^{\mathrm{T}}\bar{w_3} \\
&= S^{\mathrm{T}}(\bar{w_5}^{\mathrm{T}} + {w_5}^{\mathrm{T}}\bar{w_6}) \\
&= S^{\mathrm{T}}((\bar{w_8}{w_4}^{\mathrm{T}} + \bar{w_6}{w_3}^{\mathrm{T}})^{\mathrm{T}} - {w_5}^{\mathrm{T}}w_7\bar{w_7}w_7) \\
&= S^{\mathrm{T}}((w_4\bar{w_8}^{\mathrm{T}} + w_3\bar{w_6}^{\mathrm{T}}) - {w_5}^{\mathrm{T}}w_7\bar{w_9}{w_8}^{\mathrm{T}}w_7) \\
&= S^{\mathrm{T}}((w_4({w_7}^{\mathrm{T}}\bar{w_9})^{\mathrm{T}} + w_3(-w_7\bar{w_7}w_7)^{\mathrm{T}}) - {w_5}^{\mathrm{T}}w_7\bar{w_9}{w_8}^{\mathrm{T}}w_7) \\
&= S^{\mathrm{T}}((w_4\bar{w_9}^{\mathrm{T}}w_7 - w_3{w_7}^{\mathrm{T}}\bar{w_7}^{\mathrm{T}}{w_7}^{\mathrm{T}}) - {w_5}^{\mathrm{T}}w_7\bar{w_9}{w_8}^{\mathrm{T}}w_7) \\
&= S^{\mathrm{T}}(w_4\bar{w_9}^{\mathrm{T}}w_7 - w_3{w_7}^{\mathrm{T}}(\bar{w_9}{w_8}^{\mathrm{T}})^{\mathrm{T}}{w_7}^{\mathrm{T}}) - {w_5}^{\mathrm{T}}w_7\bar{w_9}{w_8}^{\mathrm{T}}w_7) \\
&= S^{\mathrm{T}}((Sb\bar{y}_S{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} - SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}A^{\mathrm{T}}S^{\mathrm{T}}Sb\bar{y}_S{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} \\
&\quad - SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_Sb^{\mathrm{T}}S^{\mathrm{T}}SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}) \\
&= S^{\mathrm{T}}Sb\bar{y}_S{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} - S^{\mathrm{T}}SAy_S\bar{y}_S{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} - S^{\mathrm{T}}SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_S{y_S}^{\mathrm{T}}
\end{aligned}
$$

$$
\begin{aligned}
\bar{b}_S &= \bar{w_2} \\
&= S^{\mathrm{T}}\bar{w_4} \\
&= S^{\mathrm{T}}{w_5}^{\mathrm{T}}\bar{w_8} \\
&= S^{\mathrm{T}}SA{w_7}^{\mathrm{T}}\bar{w_9} \\
&= S^{\mathrm{T}}SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{w_9} \\
&= S^{\mathrm{T}}SA(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_S
\end{aligned}
$$

Partial Sketch

We simply sketch every $(A^{\mathrm{T}}A)^{-1}$ term in $\bar{A}, \bar{b}$ to approximate $\bar{A}_D, \bar{b}_D$ respectively. Recall $\bar{A} = b\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - Ay\bar{y}^{\mathrm{T}}(A^{\mathrm{T}}A)^{-1} - A(A^{\mathrm{T}}A)^{-1}\bar{y}y^{\mathrm{T}}$, $\bar{b} = A(A^{\mathrm{T}}A)^{-1}\bar{y}$. We set $\bar{A}' = b\bar{y}_D{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} -$

$Ay_D\bar{y}_D{}^{\mathrm{T}}(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1} - A(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_D y_D{}^{\mathrm{T}}$ and use it as $\bar{A}_D$, $\bar{b}' = A(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}\bar{y}_D$ and use it as $\bar{b}_D$. Check the previous discussion for more details.

4.2 Bounds on Sketched Regression AD Transformations

For concision, we use M, M_S as the shorthanded expressions for $A^{\mathrm{T}}A$, $A^{\mathrm{T}}S^{\mathrm{T}}SA$ respectively. We also make the assumption that $\bar{y} = \bar{y}_S = \bar{y}_D$ to simplify the analysis.

Lemma 4.2.1. *If S is drawn from a subspace embedding, with probability $1-\delta$, we have*

$$\begin{aligned}
\|AM^{-1}\|_2 &= \|\Sigma^{-1}\|_2 \\
\|AM_S^{-1}\|_2 &= (1 \pm \epsilon)\|\Sigma^{-1}\|_2 \\
\|M^{-1} - M_S^{-1}\|_2 &\leq \epsilon\|\Sigma^{-1}\|_2^2 \\
\|M^{-1} - M_S^{-1}\|_F &\leq \epsilon\|\Sigma^{-1}\|_2\|\Sigma^{-1}\|_F \\
\|M^{-1} + M_S^{-1}\|_2 &\leq (2+\epsilon)\|\Sigma^{-1}\|_2^2 \\
\|AM^{-1} - AM_S^{-1}\|_2 &\leq \epsilon\|\Sigma^{-1}\|_2 \\
\|AM^{-1} - AM_S^{-1}\|_F &\leq \epsilon\|\Sigma^{-1}\|_F
\end{aligned}$$

Proof. Consider the SVD of matrix $A = U\Sigma V^{\mathrm{T}}$. We have $M = A^{\mathrm{T}}A = V\Sigma U^{\mathrm{T}}U\Sigma V^{\mathrm{T}} = V\Sigma^2V^{\mathrm{T}}$ and $M^{-1} = V\Sigma^{-2}V^{\mathrm{T}}$. Additionally, $AM^{-1} = U\Sigma V^{\mathrm{T}}V\Sigma^{-2}V^{\mathrm{T}} = U\Sigma^{-1}V^{\mathrm{T}}$ yielding $\|AM^{-1}\|_2 = \|\Sigma^{-1}\|_2$.

We can simplify the expression $AM_S^{-1} = U\Sigma V^{\mathrm{T}}(V\Sigma U^{\mathrm{T}}S^{\mathrm{T}}SU\Sigma V^{\mathrm{T}})^{-1} = U\Sigma V^{\mathrm{T}}V\Sigma^{-1}(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\Sigma^{-1}V^{\mathrm{T}} = U(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\Sigma^{-1}V^{\mathrm{T}}$ and bound its norm $\|U(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\Sigma^{-1}V^{\mathrm{T}}\|_2 = (1\pm\epsilon)\|\Sigma^{-1}\|_2$ with probability at least $1-\delta$.

The remaining bounds follow:

$$\begin{aligned}
\|M^{-1} - M_S^{-1}\|_2 &= \|V\Sigma^{-1}(I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1})\Sigma^{-1}V^{\mathrm{T}}\|_2 \\
&\leq \|\Sigma^{-1}\|_2\|I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|\Sigma^{-1}\|_2 \\
&\leq \epsilon\|\Sigma^{-1}\|_2^2
\end{aligned}$$

$$\begin{aligned}
\|M^{-1} - M_S^{-1}\|_F &= \|V\Sigma^{-1}(I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1})\Sigma^{-1}V^{\mathrm{T}}\|_F \\
&\leq \|\Sigma^{-1}\|_2\|I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|\Sigma^{-1}\|_F \\
&\leq \epsilon\|\Sigma^{-1}\|_2\|\Sigma^{-1}\|_F
\end{aligned}$$

$$\begin{aligned}
\|M^{-1} + M_S^{-1}\|_2 &= \|V\Sigma^{-1}(I + (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1})\Sigma^{-1}V^{\mathrm{T}}\|_2 \\
&\leq \|\Sigma^{-1}\|_2\|I + (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|\Sigma^{-1}\|_2 \\
&\leq (2+\epsilon)\|\Sigma^{-1}\|_2^2
\end{aligned}$$

$$\begin{aligned}
\|AM^{-1} - AM_S^{-1}\|_2 &= \|U(I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1})\Sigma^{-1}V^{\mathrm{T}}\|_2 \\
&\leq \|I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|\Sigma^{-1}V^{\mathrm{T}}\|_2 \\
&\leq \epsilon\|\Sigma^{-1}\|_2
\end{aligned}$$

$$\begin{aligned}\|AM^{-1} - AM_S^{-1}\|_F &= \|U(I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1})\Sigma^{-1}V^{\mathrm{T}}\|_F \\ &\leq \|I - (U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1}\|_2\|\Sigma^{-1}V^{\mathrm{T}}\|_F \\ &\leq \epsilon\|\Sigma^{-1}\|_F\end{aligned}$$

□

Lemma 4.2.2. *(By Sarlos and others [12] [11]) With our previous notations, we have*

$$\|y - y_S\|_2 \leq O(\epsilon)\min_x\|Ax - b\|_2\|A^+\|_2 = \frac{O(\epsilon)}{\sigma_{\min}(A)}\min_x\|Ax - b\|_2$$

4.2.1 Forward mode AD

Regular Sketch

Lemma 4.2.3. *Given a matrix S that is an (ϵ, δ, d)-OSE, we can bound $\|\dot{y}-\dot{y}_S\|_2$ with probability $1 - \delta$ as follows.*

Proof.

$$\begin{aligned}\|\dot{y} - \dot{y}_S\| &= \|M^{-1}(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b} - (\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y) \\ &\quad - M_S^{-1}(\dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{b} - (\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA + A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A})y_S)\| \\ &\leq \|M^{-1}\dot{A}^{\mathrm{T}}b - M_S^{-1}\dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb\| + \|M^{-1}A^{\mathrm{T}}\dot{b} - M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{b}\| \\ &\quad + \|M^{-1}\dot{A}^{\mathrm{T}}Ay - M_S^{-1}\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| + \|M^{-1}A^{\mathrm{T}}\dot{A}y - M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A}y_S\|\end{aligned}$$

We use the triangle inequality of version $\|AB - CD\|_2 \leq \|A - C\|_2\|B + D\|_2 + \|A + C\|_2\|B - D\|_2$ for arbitrary matrices in the following equations.

The first difference term can be written as

$$\begin{aligned}\|M^{-1}\dot{A}^{\mathrm{T}}b - M_S^{-1}\dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb\| &\leq \|M^{-1} - M_S^{-1}\|_2\|\dot{A}^{\mathrm{T}}b + \dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb\| \\ &\quad + \|M^{-1} + M_S^{-1}\|_2\|\dot{A}^{\mathrm{T}}b - \dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb\| \\ &\leq \epsilon\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}b + \dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb\| \\ &\quad + (2 + \epsilon)\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}b - \dot{A}^{\mathrm{T}}S^{\mathrm{T}}Sb\|\end{aligned}$$

Following up with the second difference term, where we use lemma 4.2.2.

$$\|M^{-1}A^{\mathrm{T}}\dot{b} - M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{b}\| \leq O(\epsilon)\min_x\|Ax - \dot{b}\|_2\|A^+\|_2$$

The third difference term can be bounded as follows:

$$\begin{aligned}
\|M^{-1}\dot{A}^{\mathrm{T}}Ay - M_S^{-1}\dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| &\le \|M^{-1} - M_S^{-1}\|_2\|\dot{A}^{\mathrm{T}}Ay + \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| \\
&\quad + \|M^{-1} + M_S^{-1}\|_2\|\dot{A}^{\mathrm{T}}Ay - \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| \\
&\le \epsilon\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}Ay + \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| \\
&\quad + (2+\epsilon)\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}Ay - \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| \\
&\le \epsilon\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}Ay + \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SAy_S\| \\
&\quad + (2+\epsilon)\|\Sigma^{-1}\|_2^2(\|\dot{A}^{\mathrm{T}}A\|_2\|y - y_S\| \\
&\quad + \|\dot{A}^{\mathrm{T}}A - \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA\|_2\|y_S\|_2)
\end{aligned}$$

Note that the second term of the bound vanishes when S is the identity matrix. We can further apply theorem 3.0.1 on $\|\dot{A}^{\mathrm{T}}A - \dot{A}^{\mathrm{T}}S^{\mathrm{T}}SA\|_2$.

The last difference term can be bounded using the result in theorem 3.0.2 and lemma 4.2.2 as follows:

$$\begin{aligned}
\|M^{-1}A^{\mathrm{T}}\dot{A}y - M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A}y_S\| &\le \|M^{-1}A^{\mathrm{T}}\dot{A}y - M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A}y\| \\
&\quad + \|M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A}y - M_S^{-1}A^{\mathrm{T}}S^{\mathrm{T}}S\dot{A}y_S\| \\
&\le O(\epsilon)\frac{\min_x\|Ax - \dot{A}y\|_2}{\sigma_{\min}(A)} \\
&\quad + O(\epsilon)\|(SA)^{+}S\dot{A}\|_2\frac{\min_x\|Ax - b\|_2}{\sigma_{\min}(A)}
\end{aligned}$$

$$\text{where } \|(SA)^{+}S\dot{A}\|_2 \le \|\Sigma^{-1}\|_2\Big(\|\dot{A}\|_2 + O(\epsilon)\sqrt{(1+d/k)(\|\dot{A}\|_2^2 + \|\dot{A}\|_F^2/k)}\Big)$$

Combining all four terms gives us an approximation bound. □

Partial Sketch

Lemma 4.2.4. *For any matrix S that is an (ϵ,δ,d)-OSE, we have with probability $1-\delta$,*

$$\|\dot{y} - \dot{y}_D\| \le \epsilon\|\Sigma^{-1}\|_2^2(\|\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}\| + \|(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y\| + \|M_S^{-1}(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})\|_2\|A^{\mathrm{T}}b\|)$$

Proof. Let $G = \dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A}$.

$$\begin{aligned}
\|\dot{y} - \dot{y}_D\| &= \|(M^{-1} - M_S^{-1})(\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}) + M_S^{-1}(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y_D - M^{-1}(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y)\| \\
&\le \epsilon\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}\| + \|(M^{-1} - M_S^{-1})Gy\| + \|M_S^{-1}G(y - y_D)\| \\
&\le \epsilon\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}\| + \epsilon\|\Sigma^{-1}\|_2^2\|Gy\| + \|M_S^{-1}G(M^{-1}A^{\mathrm{T}}b - M_S^{-1}A^{\mathrm{T}}b)\| \\
&\le \epsilon\|\Sigma^{-1}\|_2^2\|\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}\| + \epsilon\|\Sigma^{-1}\|_2^2\|Gy\| + \epsilon\|M_S^{-1}G\|_2\|\Sigma^{-1}\|_2^2\|A^{\mathrm{T}}b\| \\
&= \epsilon\|\Sigma^{-1}\|_2^2(\|\dot{A}^{\mathrm{T}}b + A^{\mathrm{T}}\dot{b}\| + \|(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})y\| + \|M_S^{-1}(\dot{A}^{\mathrm{T}}A + A^{\mathrm{T}}\dot{A})\|_2\|A^{\mathrm{T}}b\|)
\end{aligned}$$

□

4.2.2 Reverse Mode AD

Regular Sketch

Lemma 4.2.5. *With probability* $1-\delta$*, the approximation error for the term* $\bar{A}_S$ *can be bounded as follows.*

Proof. Since the terms in $\bar{A}$ and $\bar{A}_S$ have one to one correspondence, we bound the approximation error with three parts $\|\bar{A}-\bar{A}_S\|_F \leq P_1+P_2+P_3$:

$$\begin{aligned}
P_1 &= \|b\bar{y}^{\mathrm{T}}M^{-1} - S^{\mathrm{T}}Sb\bar{y}_S{}^{\mathrm{T}}M_S^{-1}\|_F \\
&\leq \|I - S^T S\|_2\|b\bar{y}^{\mathrm{T}}M_S^{-1}\|_F + \|b\bar{y}^{\mathrm{T}}(M^{-1}-M_S^{-1})\|_F \\
&\leq \|I - S^T S\|_2\|b\bar{y}^{\mathrm{T}}M_S^{-1}\|_F + \epsilon\|\Sigma^{-1}\|_2\|\Sigma^{-1}\|_F\|b\bar{y}^{\mathrm{T}}\|_2
\end{aligned}$$

$$\begin{aligned}
P_2 &= \|AM^{-1}\bar{y}y^{\mathrm{T}} - S^{\mathrm{T}}SAM_S^{-1}\bar{y}y_S{}^{\mathrm{T}}\|_F \\
&\leq \|AM^{-1}\bar{y}y^{\mathrm{T}} - AM_S^{-1}\bar{y}y^{\mathrm{T}}\|_F + \|AM_S^{-1}\bar{y}y^{\mathrm{T}} - S^{\mathrm{T}}SAM_S^{-1}\bar{y}y_S{}^{\mathrm{T}}\|_F \\
&\leq \epsilon\|\Sigma^{-1}\|_2\|\bar{y}\|\|y\| + \|(I-S^{\mathrm{T}}S)AM_S^{-1}\bar{y}y^{\mathrm{T}}\|_F \\
&\leq \epsilon\|\Sigma^{-1}\|_2\|\bar{y}\|\|y\| + \|I-S^{\mathrm{T}}S\|_2\|AM_S^{-1}\bar{y}y^{\mathrm{T}}\|_F
\end{aligned}$$

$$\begin{aligned}
P_3 &= \|Ay\bar{y}^{\mathrm{T}}M^{-1} - S^{\mathrm{T}}SAy_S\bar{y}_S{}^{\mathrm{T}}M_S^{-1}\|_F \\
&\leq \|Ay\bar{y}^{\mathrm{T}}M^{-1} - Ay_S\bar{y}^{\mathrm{T}}M_S^{-1}\|_F + \|Ay_S\bar{y}_S{}^{\mathrm{T}}M_S^{-1} - S^{\mathrm{T}}SAy_S\bar{y}_S{}^{\mathrm{T}}M_S^{-1}\|_F \\
&\leq \|Ay\bar{y}^{\mathrm{T}}M^{-1} - Ay_S\bar{y}^{\mathrm{T}}M_S^{-1}\|_F + \|I-S^{\mathrm{T}}S\|_2\|Ay_S\bar{y}^{\mathrm{T}}M_S^{-1}\|_F \\
&\leq \|A(y-y_S)\|\|\bar{y}\|\|\Sigma^{-1}\|_2\|\Sigma^{-1}\|_F + \|I-S^{\mathrm{T}}S\|_2\|Ay_S\bar{y}^{\mathrm{T}}M_S^{-1}\|_F
\end{aligned}$$

Note P_1, P_2, P_3 can be large because of the $\|I-S^{\mathrm{T}}S\|$ term. □

Lemma 4.2.6. *With probability* $1-\delta$*, the approximation error for the term* $\bar{b}_S$ *can be bounded as follows:*

$$\|\bar{b}-\bar{b}_S\|_2 \leq \|\Sigma^{-1}\|_2\|\bar{y}\|_2(\epsilon + (1+\epsilon)\|I-S^{\mathrm{T}}S\|_2)$$

Proof.

$$\begin{aligned}
\|\bar{b}-\bar{b}_S\|_2 &= \|AM^{-\mathrm{T}}\bar{y} - S^{\mathrm{T}}SAM_S{}^{-\mathrm{T}}\bar{y}_S\|_2 \\
&= \|AM^{-1}\bar{y} - AM_S^{-1}\bar{y} + AM_S^{-1}\bar{y} - S^{\mathrm{T}}SAM_S^{-1}\bar{y}\|_2 \\
&\leq \|AM^{-1} - AM_S^{-1}\|_2\|\bar{y}\|_2 + \|I-S^{\mathrm{T}}S\|_2\|AM_S^{-1}\|_2\|\bar{y}\|_2 \\
&\leq \epsilon\|\Sigma^{-1}\|_2\|\bar{y}\|_2 + \|I-S^{\mathrm{T}}S\|_2\|AM_S^{-1}\|_2\|\bar{y}\|_2 \\
&\leq \|\Sigma^{-1}\|_2\|\bar{y}\|_2(\epsilon + (1+\epsilon)\|I-S^{\mathrm{T}}S\|_2)
\end{aligned}$$

□

Note the bound is tight when $S = I$. However, $\|I-S^{\mathrm{T}}S\|_2$ can be large.

Partial Sketch

Lemma 4.2.7. *With probability at least* $1 - \delta$*, the approximation error for the term* $\bar{A}$ *can be bounded by* $\epsilon \cdot \mathsf{poly}(\|A\|_F, \|A^{-1}\|_F, \|b\|_2, \|\bar{y}\|_2)$.

Proof. The approximation error can be split into 3 terms such that $\|\bar{A} - \bar{A}_D\|_F \leq Q_1 + Q_2 + Q_3$ where:

$$\begin{aligned} Q_1 &= \|b\bar{y}^{\mathrm{T}} M^{-1} - b\bar{y}_D{}^{\mathrm{T}} M_S^{-1}\|_F \\ &\leq \epsilon \|b\bar{y}^{\mathrm{T}}\|_F \|\Sigma^{-1}\|_2 \|\Sigma^{-1}\|_F \end{aligned}$$

$$\begin{aligned} Q_2 &= \|AM^{-1}\bar{y}y^{\mathrm{T}} - AM_S^{-1}\bar{y}y_D{}^{\mathrm{T}}\|_F \\ &= \|A(M^{-1} - M_S^{-1})\bar{y}y^{\mathrm{T}} + AM_S^{-1}\bar{y}(y^{\mathrm{T}} - y_D{}^{\mathrm{T}})\|_F \\ &\leq \|A(M^{-1} - M_S^{-1})\bar{y}y^{\mathrm{T}}\|_F + \|AM_S^{-1}\bar{y}(y - y_D)^{\mathrm{T}}\|_F \\ &\leq \epsilon \|\Sigma^{-1}\|_2 \|\bar{y}y^{\mathrm{T}}\|_F + \|AM_S^{-1}\|_2 \|\bar{y}(y - y_D)^{\mathrm{T}}\|_F \\ &\leq \epsilon \|\Sigma^{-1}\|_2 \|\bar{y}\| \|y\| + \|AM_S^{-1}\|_2 \|\bar{y}\| \|AM^{-1} - AM_S^{-1}\|_2 \|b\| \\ &\leq \epsilon \|\Sigma^{-1}\|_2 \|\bar{y}\| \|y\| + \|AM_S^{-1}\|_2 \|\bar{y}\| \epsilon \|\Sigma^{-1}\|_2 \|b\| \\ &\leq \epsilon \|\bar{y}\| \|\Sigma^{-1}\|_2 (\|y\| + \|AM_S^{-1}\|_2 \|b\|) \\ &\leq \epsilon \|\bar{y}\| \|\Sigma^{-1}\|_2 (\|y\| + (1+\epsilon)\|\Sigma^{-1}\|_2 \|b\|) \end{aligned}$$

$$\begin{aligned} Q_3 &= \|Ay\bar{y}^{\mathrm{T}} M^{-1} - Ay_D\bar{y}^{\mathrm{T}} M_S^{-1}\|_F \\ &= \|Ay\bar{y}^{\mathrm{T}}(M^{-1} - M_S^{-1}) + Ay\bar{y}^{\mathrm{T}} M_S^{-1} - Ay_D\bar{y}^{\mathrm{T}} M_S^{-1}\|_F \\ &\leq \epsilon \|Ay\bar{y}^{\mathrm{T}}\|_2 \|\Sigma^{-1}\|_2 \|\Sigma^{-1}\|_F + \|A(y - y_D)\bar{y}^{\mathrm{T}} M_S^{-1}\|_F \\ &\leq \epsilon \|Ay\bar{y}^{\mathrm{T}}\|_2 \|\Sigma^{-1}\|_2 \|\Sigma^{-1}\|_F + \epsilon \|A\|_2 \|\Sigma^{-1}\|_2 \|b\| \|\bar{y}^{\mathrm{T}} M_S^{-1}\| \\ &\leq \epsilon \|A\|_2 \|\Sigma^{-1}\|_2 (\|y\bar{y}^{\mathrm{T}}\|_2 \|\Sigma^{-1}\|_F + \|b\| \|\bar{y}^{\mathrm{T}} M_S^{-1}\|) \end{aligned}$$

Note that Q_1, Q_2, Q_3 are $O(\epsilon)$. □

Lemma 4.2.8. *With probability* $1 - \delta$*, the reverse mode approximation error for the term* $\bar{b}_D$ *satisfies*

$$\|\bar{b} - \bar{b}_D\|_2 \leq \epsilon \|\Sigma^{-1}\|_2 \|\bar{y}\|_2$$

Proof.

$$\begin{aligned} \|\bar{b} - \bar{b}_D\|_2 &= \|AM^{-1}\bar{y} - AM_S^{-1}\bar{y}_S\|_2 \\ &= \|U(I - (U^{\mathrm{T}} S^{\mathrm{T}} SU)^{-1})\Sigma^{-1} V^{\mathrm{T}} \bar{y}\|_2 \\ &\leq \epsilon \|U\|_2 \|\Sigma^{-1}\|_2 \|\bar{y}\|_2 \\ &\leq \epsilon \|\Sigma^{-1}\|_2 \|\bar{y}\|_2 \end{aligned}$$

□

In summary, we provide theoretical bounds on the approximation errors of forward and reverse mode AD transformations. For forward mode AD transformations, the errors for both the "Regular Sketch" and the "Partial Sketch" are in $O(\epsilon)$. However, for reverse mode AD transformations, the errors for the "Regular Sketch" consist of the term $\|I - S^{\mathrm{T}}S\|_2$ which can be large under certain circumstances. Say if S is a CountSketch matrix, $\|I - S^{\mathrm{T}}S\|_2$ can be in $O(n)$. In reverse mode, the bound for AD transformations of the "Partial Sketch" is straightforward and all in $O(\epsilon)$.

An intuitive explanation about the tight derivative bounds of the "Partial Sketch" is that, when combining the terms, we can factor out all the terms that are not $(A^{\mathrm{T}}A)^{-1}$ or $(A^{\mathrm{T}}S^{\mathrm{T}}SA)^{-1}$ as they are common terms. We can then use SVD on A with the inequality $\|(U^{\mathrm{T}}S^{\mathrm{T}}SU)^{-1} - I\|_2 \leq \epsilon$ to factor out a constant term ϵ. For the "Regular Sketch" however, because of how derivative rules work, we end up getting $I - S^{\mathrm{T}}S$ term if we factor out common terms. We cannot simply bound the operator norm of a term with an $I - S^{\mathrm{T}}S$ factor by adding the norm operator on $I - S^{\mathrm{T}}S$ as its norm grows with its dimension for certain sketching matrices. At the same time, the approximation errors of the ADs of the "Regular Sketch" do not have the ϵ factor.

Thus, we expect a better performance of the "Partial Sketch" than the "Regular Sketch" in terms of approximating derivatives when replacing the least squares regression. We also expect the "Partial Sketch" outperforms the "Regular Sketch" when incorporated in a larger deep learning system as reverse mode ADs are used in back-propagation and the "Partial Sketch" has lower approximation errors in ADs.

Chapter 5

Experiments

We consider both synthetic and deep learning experiments to highlight the performance of our proposed sketching methods in approximating derivatives and speeding up training time.

For the synthetic experiment, we plot the approximation error of AD transformations of the two proposed approaches for obtaining forward and reverse mode AD in figure 5.1.

- We generate $A \in \mathbb{R}^{100000 \times 100}$ and $b \in \mathbb{R}^{100000 \times 1}$ with entries to be drawn uniformly random in $[0, 1)$.
- We set $\bar{y}[i] = 1$ if $y[i] > 0$ and $\bar{y}[i] = -1$ if $y[i] < 0$ (use sign as cost function).
- We set $\dot{A}$ and $\dot{b}$'s each entry to be to be drawn in i.i.d. $N(0, 1)$ then multiplied with 10^{-4}.
- For the forward mode, we plot $\|\dot{y} - \dot{y}_S\|_2$ in yellow and $\|\dot{y} - \dot{y}_D\|_2$ in blue.
- For the reverse mode, we plot $\|\bar{b} - \bar{b}_S\|_2$ in yellow and $\|\bar{b} - \bar{b}_D\|_2$ in blue.
- Three families of sketching matrices: Gaussian, CountSketch and subsampled randomized Hadamard transform (SRHT) are applied.

We observe that the "Partial Sketch" is more accurate, which is consistent with our previous results.

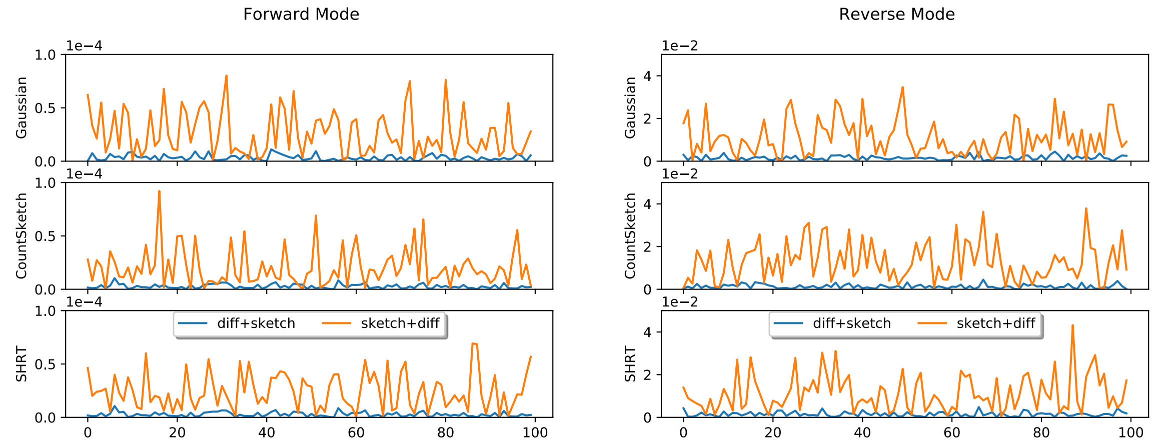


Figure 5.1: Numerical Observations in Synthetic Experiment

In figure 5.1 we use old terminology "diff+sketch" and "sketch+diff" in place of the "Par-

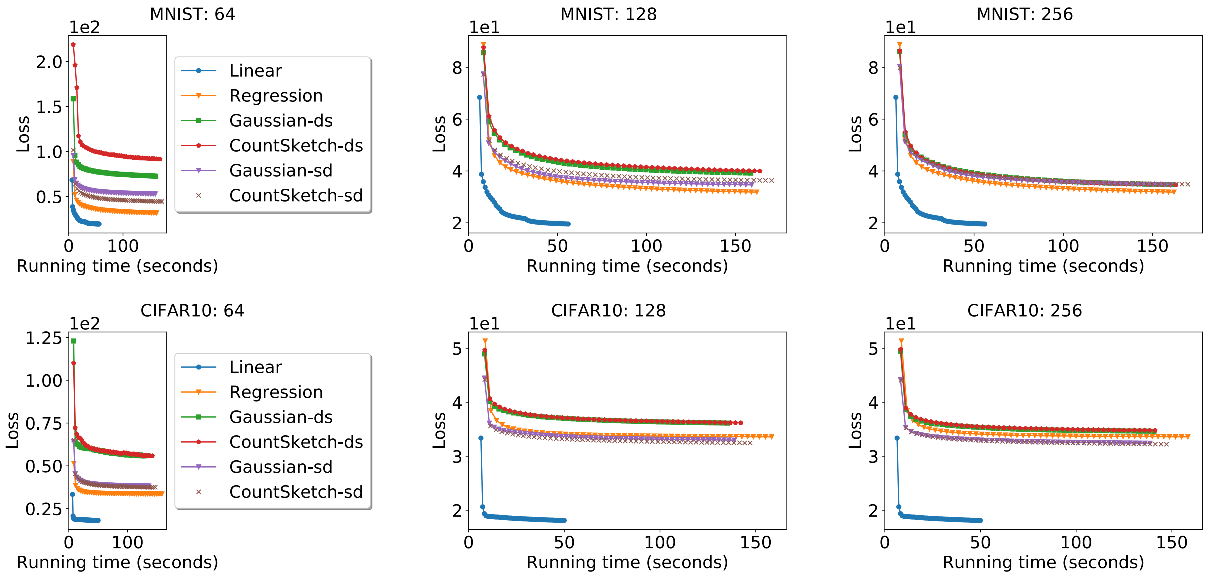


Figure 5.2: Training loss in linear and regression layers on MNIST and CIFAR10 datasets with $64, 128$ and 256 rank features run on a NVIDIA GTX980 GPU.

tial Sketch" and the "Regular Sketch". Note we adapt the latter terminology throughout this document for less confusion caused by naming. In figure 5.2 we use **ds** and **sd** as shorthanded notation for "diff+sketch" and "sketch+diff" respectively, which should be replaced by **ps** and **rs** that stand for the "Partial Sketch" and the "Regular Sketch" respectively.

For the deep learning experiments, we consider the following real-world datasets:

MNIST: $60,000$ handwritten digits of shape 28×28 for training and $10,000$ for testing.

CIFAR10: $60,000$ images in 10 classes of which $10,000$ are for testing.

We use an autoencoder for showcasing our sketched regression layer. We consider the standard encoder decoder framework with the encoder consisting of a linear layer mapping to 64 dimensions followed by a ReLu layer. The decoder is built with a linear layer mapping from 64 to 128 dimensions followed by a ReLU and a second linear layer mapping from 128 dimensions to the input dimension, followed by a $\tanh$ layer. In our experiments, we replace the linear layer of the encoder by the linear least squares regression modules, both unsketched and sketched, which have been considered in this work.

The linear layer takes in input data x, then applies a linear transformation to x with the formula $y = Ax + b$ in which A, b will be learned. The regression layer takes in input data x, outputs $y = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}x$ in which A will be learned throughout the training.

Sketching		**MNIST**			**CIFAR10**		
		64	128	256	64	128	256
RG	**ps**	0.16	0.08	0.08	0.21	0.09	0.08
	rs	0.11	0.07	0.08	0.11	0.08	0.08
CS	**ps**	0.15	0.10	0.09	0.14	0.09	0.09
	rs	0.10	0.09	0.08	0.09	0.08	0.08

Table 5.1: Test loss on the MNIST and CIFAR10 datasets after convergence of the sketching algorithms using random Gaussian (RG) and CountSketch matrices (CS) with "Partial Sketch" or **ps**, and "Regular Sketch" or **rs**.

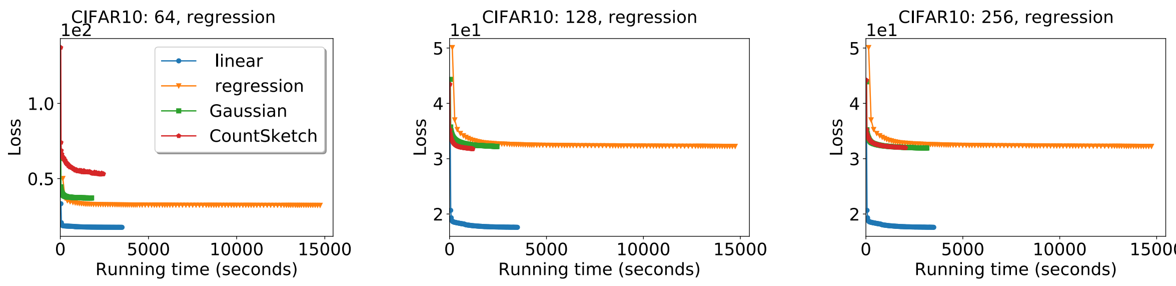


Figure 5.3: Training loss in linear and regression layers on CIFAR10 dataset with 64, 128 and 256 rank features on multicore settings (CPU).

The result running on GPU is shown in figure 5.2. Notice that the regression layer tends to result in a higher loss and running time compared with the linear layer, and the sketched regression layer hardly get any speedup over the plain regression layer. This is probably due to the fact that we have not taken advantage of GPU capabilities for implementing the sketching operations. We will discuss this oddity in the last chapter. Also, surprisingly the "Regular Sketch" seems to result in better performance in terms of training loss than the "Partial Sketch" approach, as shown numerically in table 5. And with higher rank features (for example 128, 256), the sketched regression model achieves a lower loss.

We note that sketching methods provide a significant speedup over the plain regression layer in the CPU setting as shown in figure 5.3. Here we only consider the "Partial Sketch" version though it should also apply to the "Regular Sketch".

Chapter 6

Discussions and Future Work

We conclude that a sketched regression layer can be a good replacement of a linear regression layer in the context of differentiable programming. Both methods provide a good speedup compared to unsketched linear regression in CPU setting. We believe sketch methods would also have a good performance in GPU setting if we fix our code to take advantage of GPU in sketching computations.

A big loose end in this work, suggested by Dr. Zico Kolter, is that the regression layer is a special kind of the linear layer and replacing the linear layer with the regression layer without making assumption to the experiment datasets typically yields worse performance in terms of model accuracy. Both the linear layer and the regression layer are performing a linear transformation to the input, and the regression layer assumes certain characteristics of the transformation (with form $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$). In this sense, the linear layer captures more scenarios and generally achieves lower loss than the regression layer. Therefore, it is not realistic to expect the regression layer to outperform the linear layer in most type of neural networks. However, the regression layer can be useful with problems where the matrix $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ is structured. Many kinds of structured matrices are closed under arithmetic operations like summation, product, transpose and inverse [14]. If we restrict the matrix A to be structured in training, $(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}$ will end up being a structured matrix. Moreover, the training time can be shortened with the regression layer as shown in 5.3. We give a few examples of structured matrices that are commonly considered below.

- A can be innately represented with the sum of two low rank matrices $\theta_1 A_1 + \theta_2 A_2$ where $\theta_1, \theta_2 \in \mathbb{R}^{n\times 1}$ and $A_1, A_2 \in \mathbb{R}^{1\times d}$.
- A can be a Vandermonde matrix. An $n \times d$ Vandermonde matrix has the form

$$\begin{bmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{d-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{d-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_n & a_n^2 & \dots & a_n^{d-1} \end{bmatrix}$$

  It has the property that computing $A \cdot x$ for an arbitrary vector $x \in \mathbb{R}^d$ takes time $O(n \log n)$ [13], which can further accelerate the running time of the regression layer when $d > O(\log n)$.

- A can be a Toeplitz matrix. An $n \times d$ Toeplitz matrix has the form that $A_{i,j} = A_{i+1,j+1}$. It takes time $O(nd \log n)$ to compute $A^{\mathrm{T}}A$ [13].
- A can be Hankel matrix. An $n \times d$ Hankel matrix has the form that for $i \leq j$, $A_{i,j} = A_{i+k,j-k}$ for all $0 \leq k \leq j - i$.

In these cases, we expect the regression layer to have lower training time and be more robust than the linear layer. Possible applications include the cubic splines interpolation problem, the polynomial interpolation problem and fitting the d-th order autoregression model [13]. These applications correspond to the first three structured matrices we listed above respectively. For future work, we can conduct experiments using datasets with such features.

Due to the aforementioned reasons, we cannot conclude which sketching method generally performs better in deep learning setting with the current experiment. In fact, although the test loss of the "Partial Sketch" scheme is higher compared with that of the "Regular Sketch" scheme, we still believe the "Partial Sketch" should outperform the "Regular Sketch" as it gives a better estimation of the derivatives by the synthetic experiment. One possible improvement of the "Partial Sketch" method is to change the primal to that of the "Regular Sketch", as the primal of the "Regular Sketch" theoretically performs better than the primal of the "Partial Sketch".

Another potential improvement is that we can further accelerate the speed of a sketched regression layer running on sparse datasets using CountSketch matrix.

Last but not least, we can extend the sketched linear regression problem to sketched regression with different norms, apply sketching to other problems in differentiable programming, or even find unbiased gradient estimates with sketching method.

Bibliography

[1] Atilim Gunes Baydin, Barak A. Pearlmutter, and Alexey Andreyevich Radul. Automatic differentiation in machine learning: a survey. *CoRR*, abs/1502.05767, 2015. URL `http://arxiv.org/abs/1502.05767`. 2.2

[2] Bruce Christianson. Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326, 1994. doi: 10.1080/10556789408805572. URL `https://doi.org/10.1080/10556789408805572`. 2.4

[3] Michael B. Cohen, Jelani Nelson, and David P. Woodruff. Optimal approximate matrix product in terms of stable rank. *CoRR*, abs/1507.02268, 2015. URL `http://arxiv.org/abs/1507.02268`. 3.0.1

[4] Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *CoRR*, abs/0710.1435, 2007. URL `http://arxiv.org/abs/0710.1435`. 3

[5] Vipul Gupta, Swanand Kadhe, Thomas A. Courtade, Michael W. Mahoney, and Kannan Ramchandran. Oversketched newton: Fast convex optimization for serverless systems. *CoRR*, abs/1903.08857, 2019. URL `http://arxiv.org/abs/1903.08857`. 2.5

[6] Nikita Ivkin, Daniel Rothchild, Enayat Ullah, Vladimir Braverman, Ion Stoica, and Raman Arora. Communication-efficient distributed SGD with sketching. *CoRR*, abs/1903.04488, 2019. URL `http://arxiv.org/abs/1903.04488`. 2.5

[7] Younghan Jeon, Minsik Lee, and Jin Young Choi. Differentiable forward and backward fixed-point iteration layers, 2020. 2.4

[8] Hang Liao, Barak Pearlmutter, Vamsi Potluru, and David Woodruff. Automatic differentiation of sketched regression. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4367–4376, Online, 26–28 Aug 2020. PMLR. URL `http://proceedings.mlr.press/v108/liao20a.html`. 1

[9] Mert Pilanci and Martin J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *CoRR*, abs/1411.0347, 2014. URL `http://arxiv.org/abs/1411.0347`. 4

[10] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=`

BkevoJSYPB. 2.4

[11] Eric Price, Zhao Song, and David P. Woodruff. Fast regression with an \$\ell_\infty\$ guarantee. *CoRR*, abs/1705.10723, 2017. URL http://arxiv.org/abs/1705.10723. 4.2.2

[12] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 143–152, 2006. 3, 4.2.2

[13] Xiaofei Shi and David P. Woodruff. Sublinear time numerical linear algebra for structured matrices. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 4918–4925. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33014918. URL https://doi.org/10.1609/aaai.v33i01.33014918. 6

[14] Anna T. Thomas, Albert Gu, Tri Dao, Atri Rudra, and Christopher Ré. Learning compressed transforms with low displacement rank. *CoRR*, abs/1810.02309, 2018. URL http://arxiv.org/abs/1810.02309. 6

[15] David P. Woodruff. Sketching as a tool for numerical linear algebra. *CoRR*, abs/1411.4357, 2014. URL http://arxiv.org/abs/1411.4357. 1