Introduction to ADaaa

Kaiwen Jin

香港科技大学

2024-11-26

Spectral Density

It has two kinds of forms.

$$1. \quad A(x) = \sum_{k=1}^{N} \delta(x - x_i)$$

$$2. \quad A(x) \in \mathcal{S}, \quad A(z) \in \mathbb{H}(\mathbb{C})$$

Now we nonly consider the second kind.

Matsubara Green Function

Define 1. Matsubara Green function

$$G(z) = \int_{\mathbb{R}} \frac{A(x)}{z-x} dx \quad Imz > 0,$$

$$G(w) = \lim_{\eta \to 0^+} \int \frac{A(x)}{w + i\eta - x} dx = P.V. \int \frac{A(x)}{w-x} dx - i\pi A(w)$$

$$A(w) = -\frac{1}{\pi} Im(G(w))$$

Theorem 1.

$$\lim_{z \to w} \int \frac{A(x)}{z-x} dx = P.V. \int \frac{A(x)}{w-x} dx - i\pi * sgn(\eta) A(w)$$

Proposition 1.1. The green function $G(z)$ on $Imz > 0$ can be analytically continued to the whole complex plane $\mathbb{C}$.

It means that $G(z)$ has no pole on $\mathbb{C}$.

2. Complex Differentiation

Definition 2. For a function

$$f : \mathbb{C} \to \mathbb{C}, \quad f(x, y) = u(x, y) + iv(x, y), \quad u, v \in C^\infty$$

We define that

$$df = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy = \frac{\partial f}{\partial z} dz + \frac{\partial f}{\partial z^*} dz^*$$

In above formula,

$$dz = dx + idy, \quad dz^* = dx - idy$$

$$\frac{\partial}{\partial z} = \frac{1}{2}\left(\frac{\partial}{\partial x} - i\frac{\partial}{\partial y}\right), \quad \frac{\partial}{\partial z^*} = \frac{1}{2}\left(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}\right)$$

2. Complex Differentiation

Proposition 2.1

$$\left(\frac{\partial f}{\partial z}\right)^* = \left(\frac{\partial f^*}{\partial z^*}\right)$$

Proposition 2.2

$$dg(f) = \left(\frac{\partial g}{\partial f}\frac{\partial f}{\partial z} + \frac{\partial g}{\partial f^*}\frac{\partial f^*}{\partial z}\right)dz + \left(\frac{\partial g}{\partial f}\frac{\partial f}{\partial z^*} + \frac{\partial g}{\partial f^*}\frac{\partial f^*}{\partial z^*}\right)dz^*$$

3. AAA algorithm

It's a interesting algorithm because it's most important idea is not barycentric but deviding a set of points into 2 parts. One of them is insert points set and the second is checking points set.

The reason is you can't decide the weight just with all ponits as insert points.

Denote chosen points as A, and set of waiting points as B. Assume that

$$A = \{z_1, .., z_n\}, \quad B = \left\{z_{\{n+1\}}, .., z_m\right\}$$

Now consider:

$$L = \left(\frac{G(z_j) - G(z_k)}{z_j - z_k}\right)_{jk}$$

Then we get a sub matrix L_n from it by getting the first n columns and $n + 1, ., m$ rows.

3. AAA algorithm

$$
\begin{array}{cc}
 & \\
\text{m-n}\left\{ \right. & \begin{bmatrix} 0 & & & \\ & \cdots & & \\ & \dfrac{G(iw_j) - G(iw_k)}{z_j - z_k} & \cdots & \\ & & & 0 \end{bmatrix} \\
 & \underbrace{\qquad\qquad}_{\text{n}}
\end{array}
$$

3. AAA algorithm

Now for

$$G(z) \approx \frac{N_{n(z)}}{D_{n(z)}}$$

$$N_{n(z)} = \sum_{j=1}^{n} \frac{w_j G(z_j)}{z - z_j}, \quad D_{n(z)} = \sum_{j=1}^{n} \frac{w_j}{z - z_j}$$

We have

$$(GD_n - N_n)(B) = L_n w$$

$$\Rightarrow \min_w \|(GD_n - L_n)(B)\|_{L^2} = \min_w \|L_n w\| = \min \sigma(L_n)$$

3. AAA algorithm

Then we can use svd to find such min $\sigma(L_n)$ and related w.

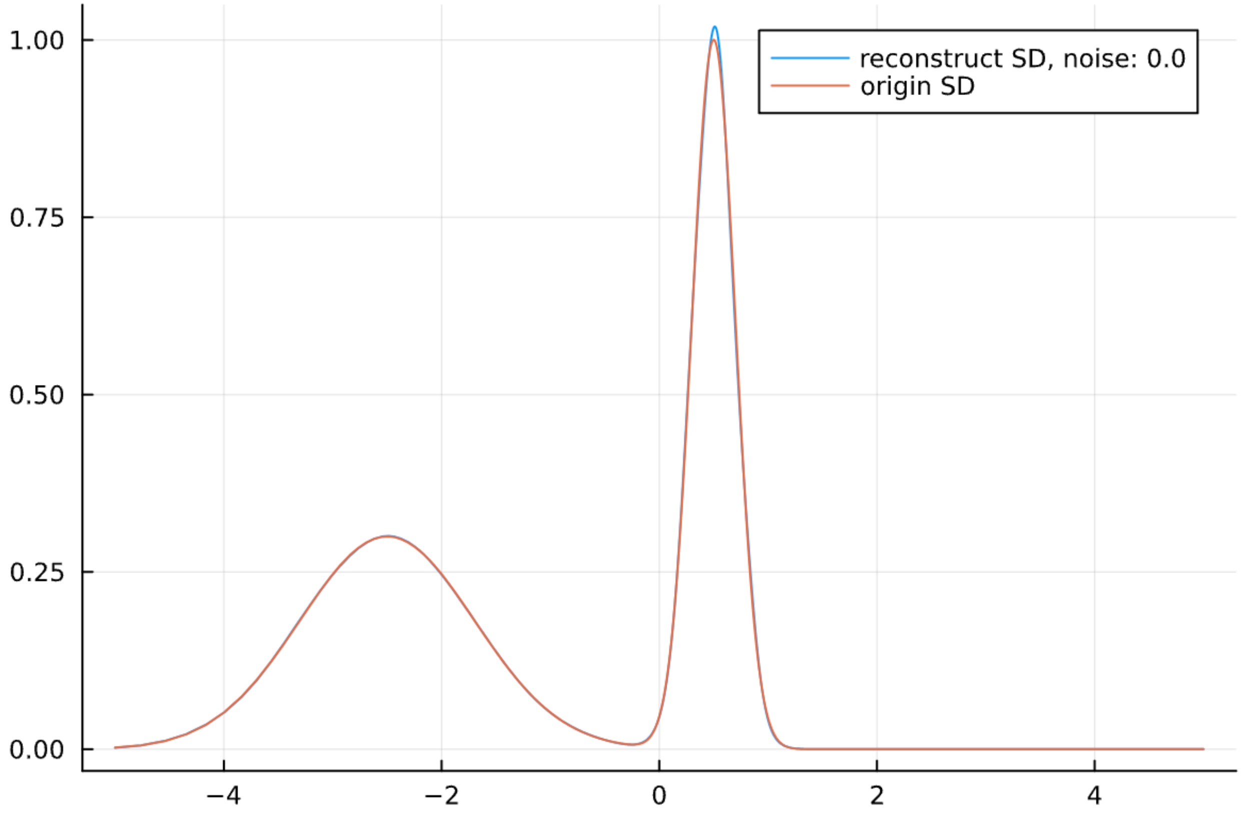
Then we chose

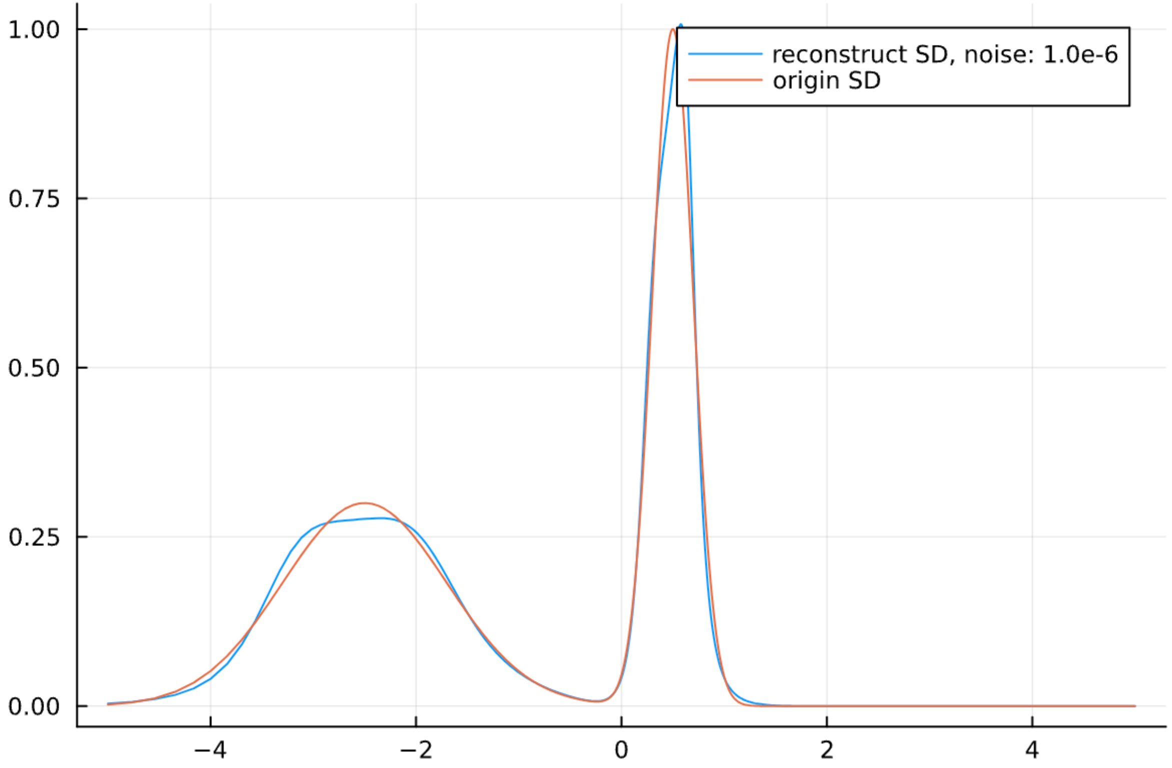
$$z_{new} = argmax_{z\in B} \quad \|G(z) - \frac{N_{n(z)}}{D_{n(z)}}\|$$

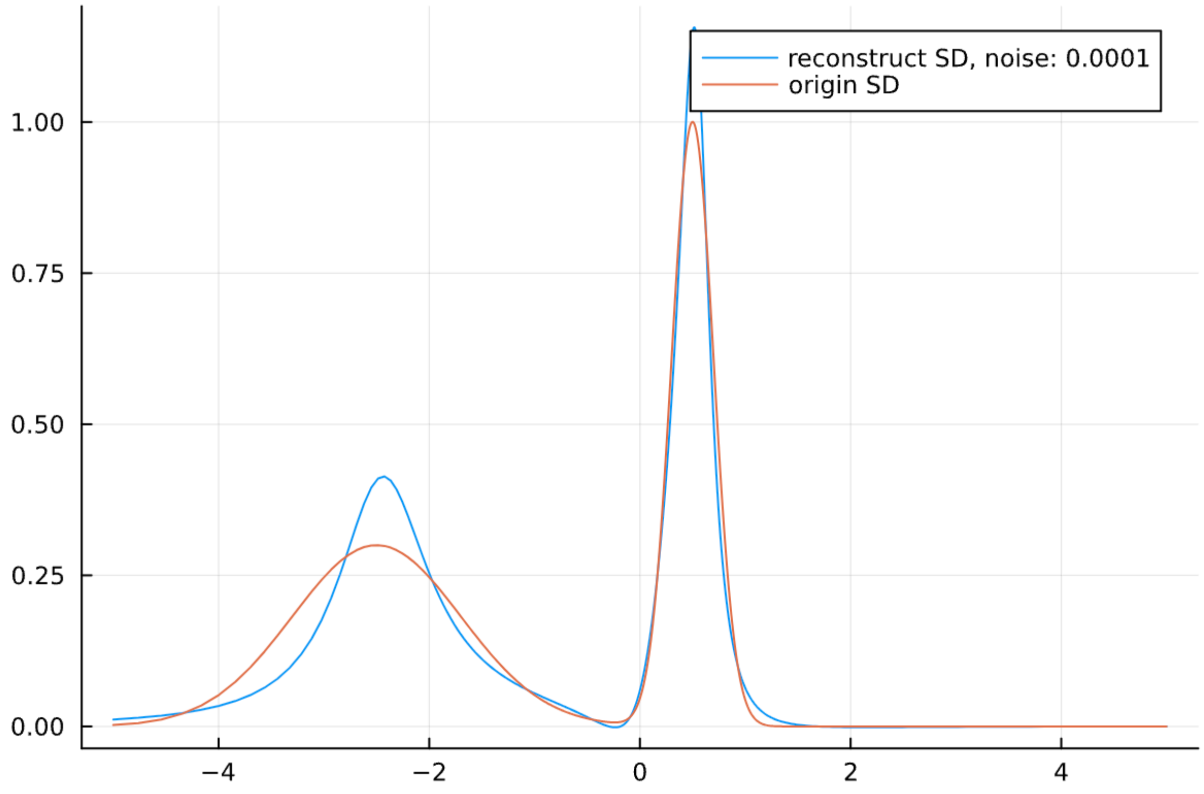
Add z_{new} into A and delete it from B and continue iteration.

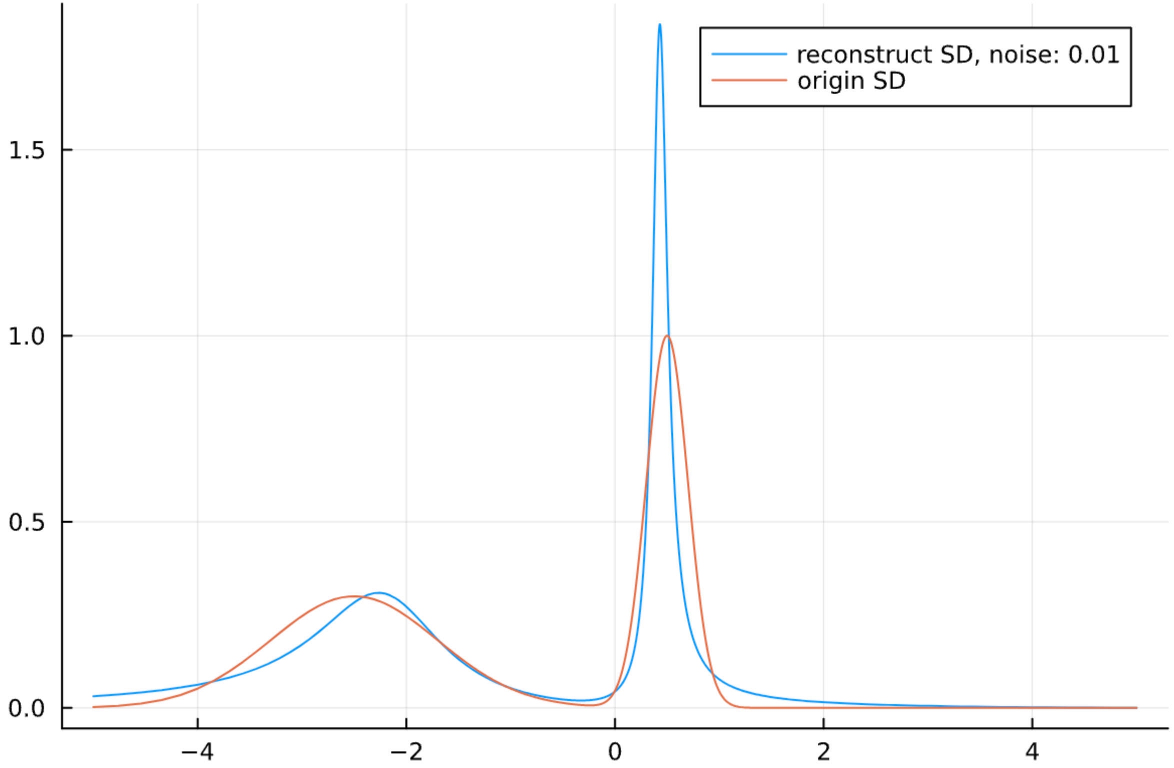
Get $G(z)$ and then we can reconstruct $A(w)$

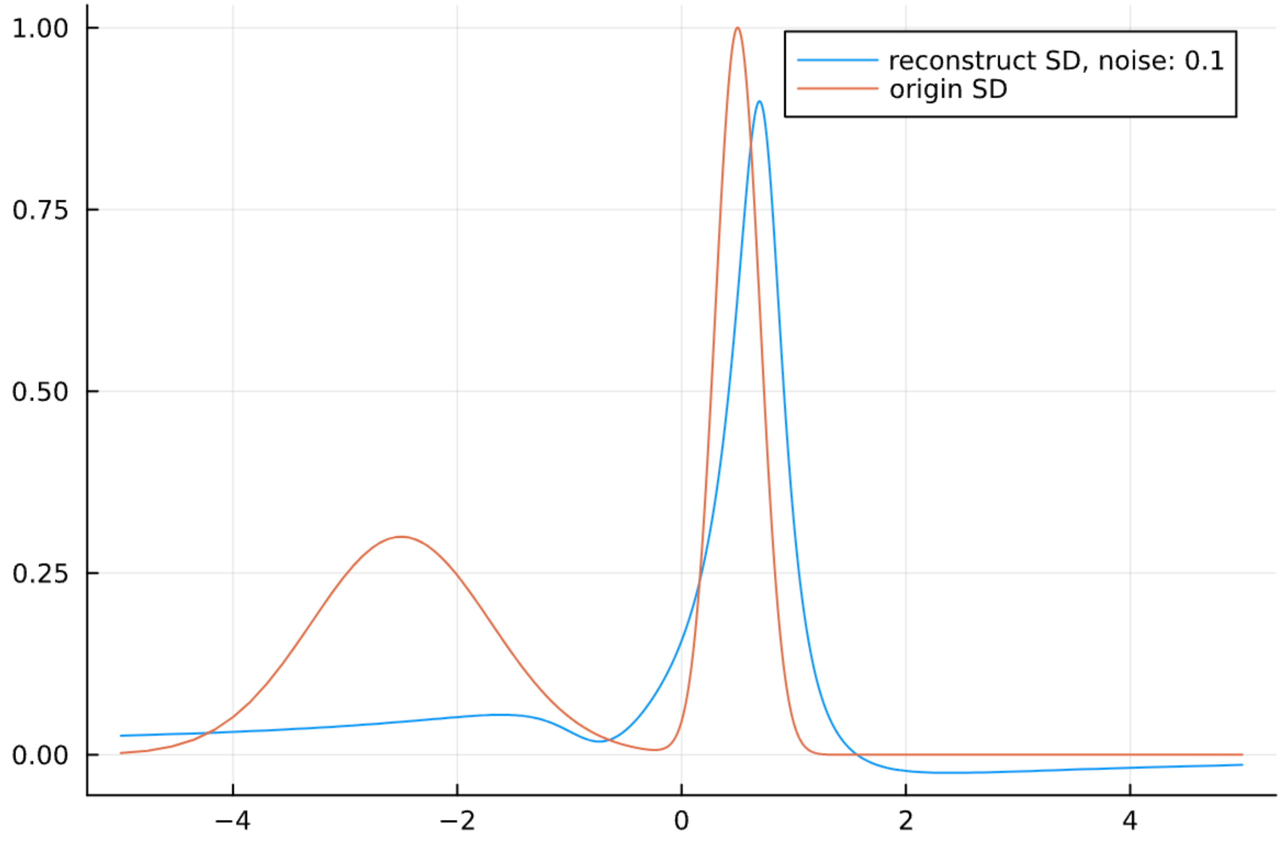
Pure signal without noise, with 1e-6 noise and larger 1e-4 noise, 1e-2 noise and 1e-1 noise


reconstruct SD, noise: 0.0
origin SD
1.00
0.75
0.50
0.25
0.00
−4
−2
0
2
4


reconstruct SD, noise: 1.0e-6
origin SD
1.00
0.75
0.50
0.25
0.00
−4
−2
0
2
4


reconstruct SD, noise: 0.0001
origin SD
1.00
0.75
0.50
0.25
0.00
−4
−2
0
2
4


reconstruct SD, noise: 0.01
origin SD
1.5
1.0
0.5
0.0
−4
−2
0
2
4

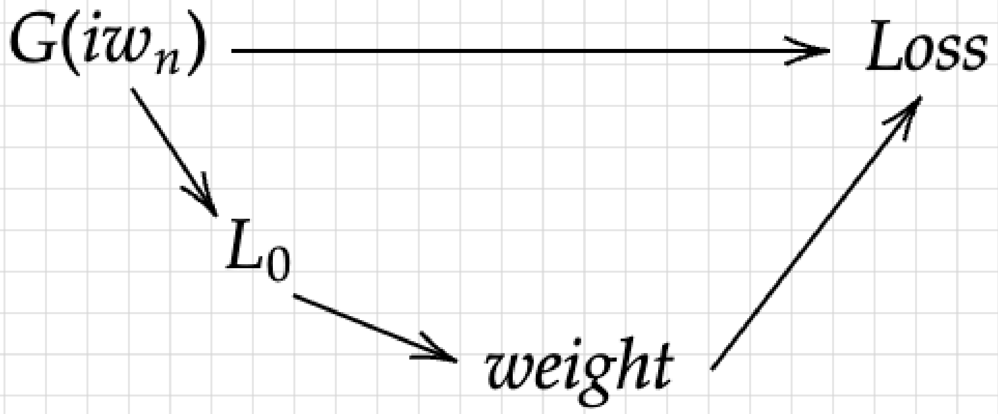

1.00
0.75
0.50
0.25
0.00
−4
−2
0
2
4
reconstruct SD, noise: 0.1
origin SD

4. Calculate ∇ Loss

Given $G_0(iwn)$ and wn, the way we calcuate Loss function is as following chart and the Loss function is defined as

$$Loss(G(iwn), weight) = \|A(x) - A_0(x)\|_2$$

4. Calculate ∇ Loss



$L0 : sub\ matrix\ of\ Lowner\ matrix\ generated\ by\ G(iw_n)$

4. Calculate ∇ Loss

Finite difference (FD) works awful for calculating derivative of

$$Loss(G, w)$$

So we use AD to calculate $\nabla Loss$

But because L_0 is an ill-condition number matrix and the log of its condition number is approximately proportional to the number of $G(iwn)$, formulas of calculating compelx SVD performs disastrously. So we directly use FD and infact

$$\frac{weight(G_0 + \varepsilon) - weight(G_0)}{\varepsilon}$$

performs very stably as ε changes.

Then with proposition 2.2 we have

4. Calculate ∇ Loss

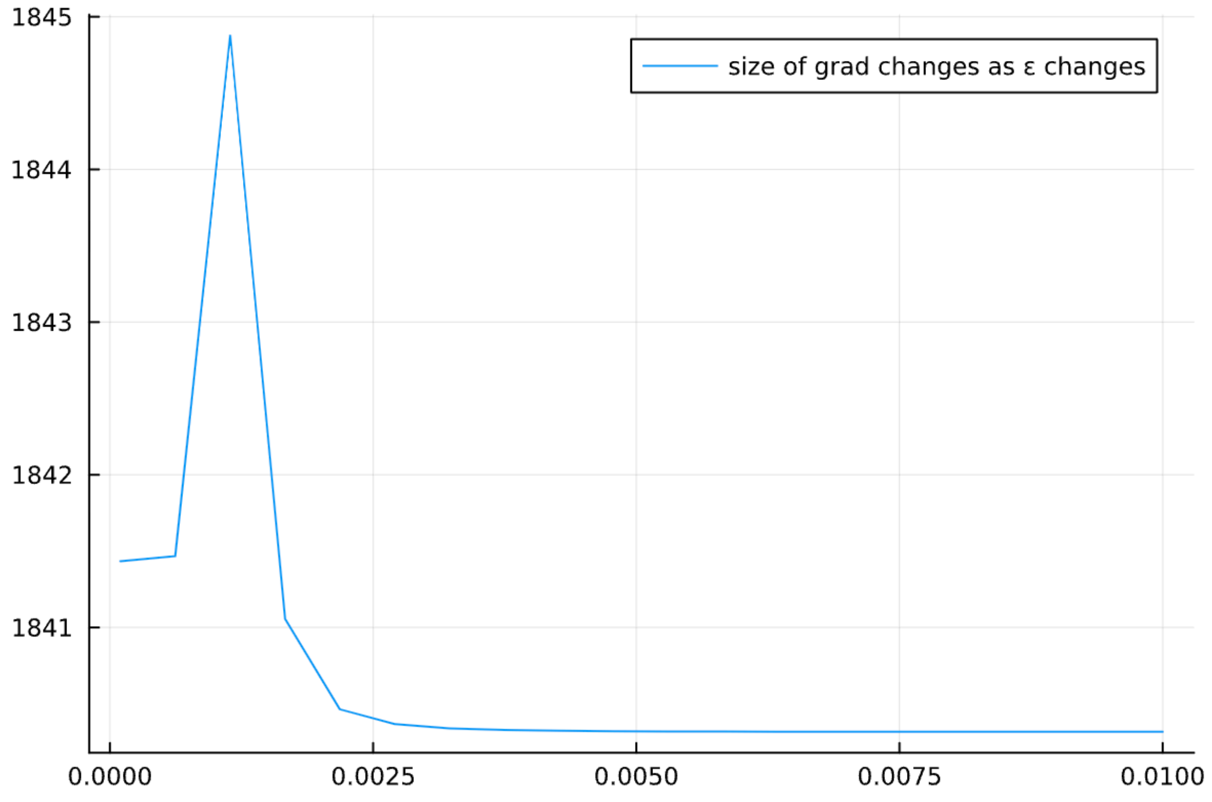
Theorem 4.

$$\nabla_G Loss(G, w(G)) = 2\left(\frac{\partial L}{\partial G}\right)^* = \nabla_1 L + 2\left(\frac{\partial L}{\partial w}\right)^\dagger * \left(\frac{Jw}{JG}\right)^* + 2\left(\frac{\partial L}{\partial w}\right)^T * \frac{Jw}{JG^*}$$

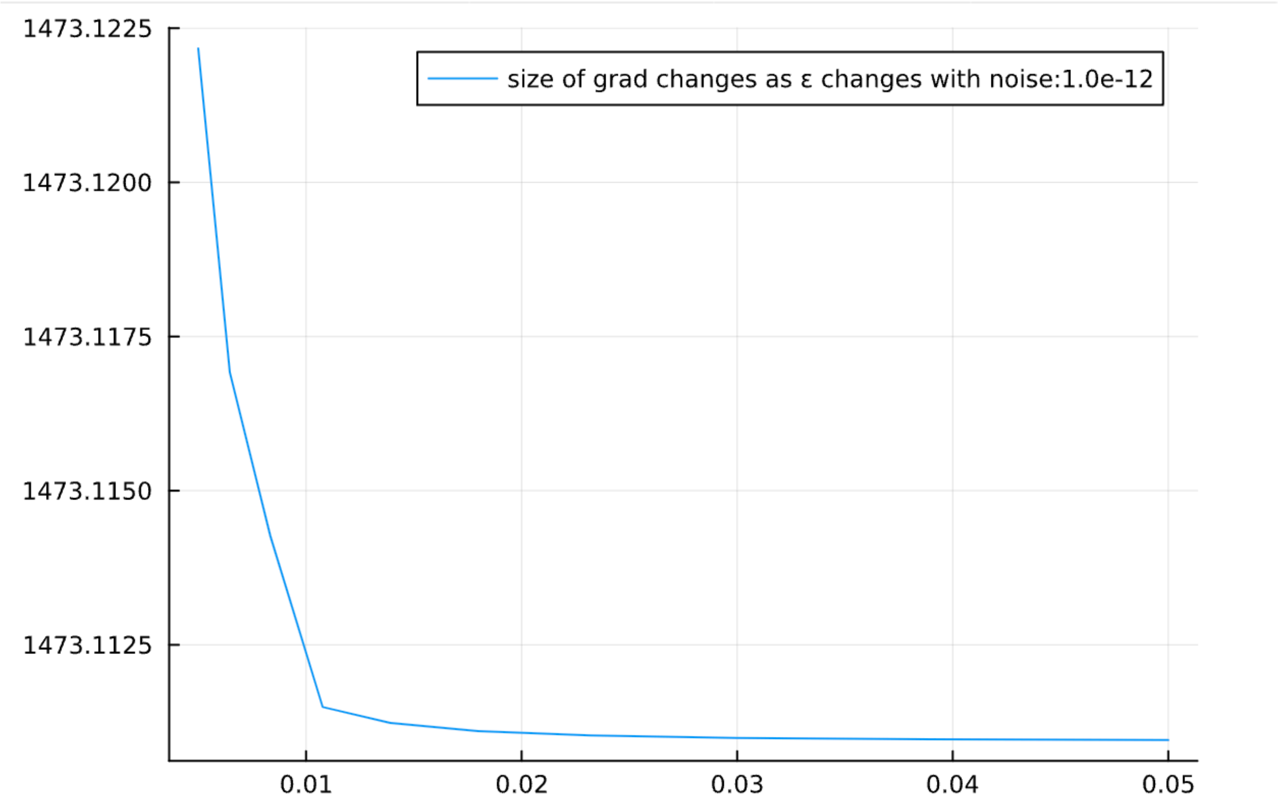
Here $\frac{Jy}{Jx}$ means the jacobian matrix.

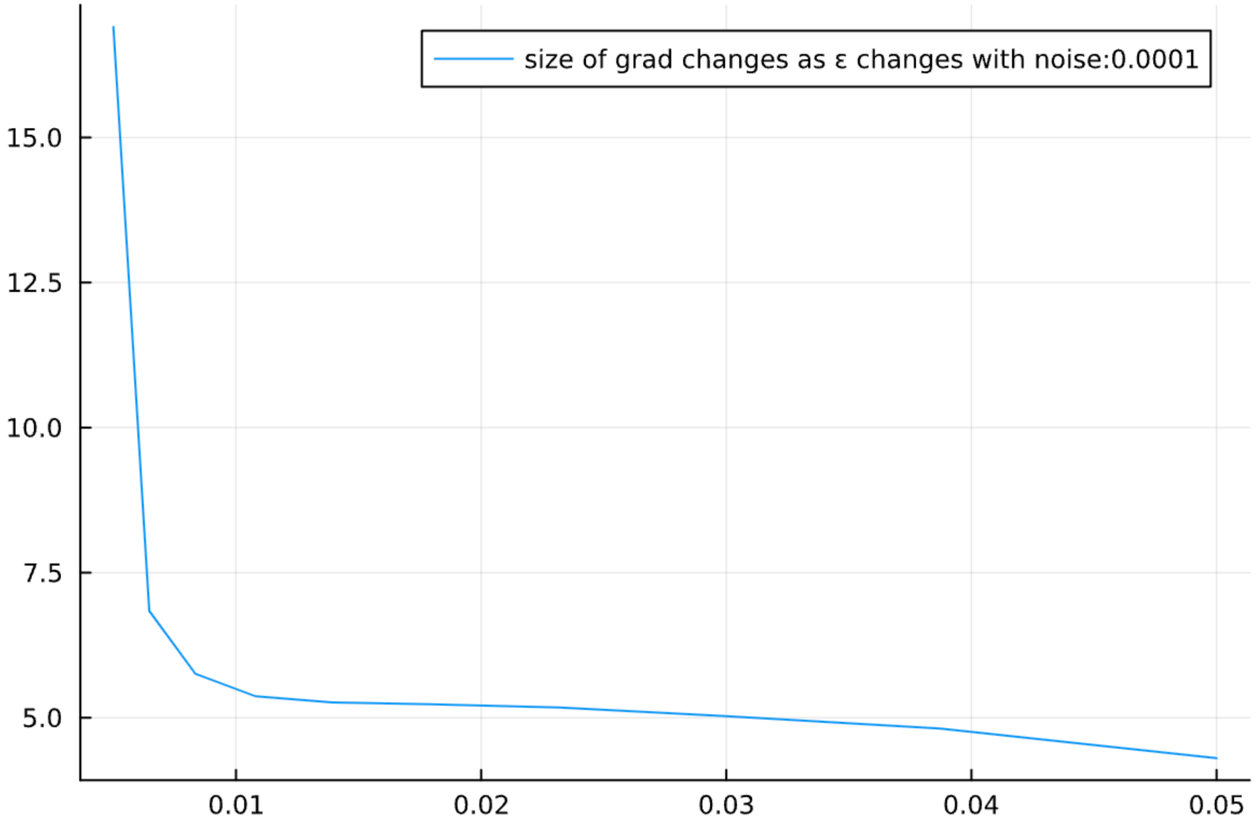
We have checked the correctness of the formula we get, and the stablity of calculating $\frac{Jw}{JG}$ by FD. Now we show the variation of $|\nabla L(G)|$ as ε in FD changes.

When no noise, it's stable around 1840:


1845
1844
1843
1842
1841
0.0000
0.0025
0.0050
0.0075
0.0100
size of grad changes as ε changes

But if we add noise even if a little, the AD value break totally. In this condition the only we to get a suitable gradient is calculate a series of ε and choose a piece of relatively stable statistic.


size of grad changes as ε changes with noise:1.0e-12
1473.1225
1473.1200
1473.1175
1473.1150
1473.1125
0.01
0.02
0.03
0.04
0.05


size of grad changes as ε changes with noise:0.0001
15.0
12.5
10.0
7.5
5.0
0.01
0.02
0.03
0.04
0.05

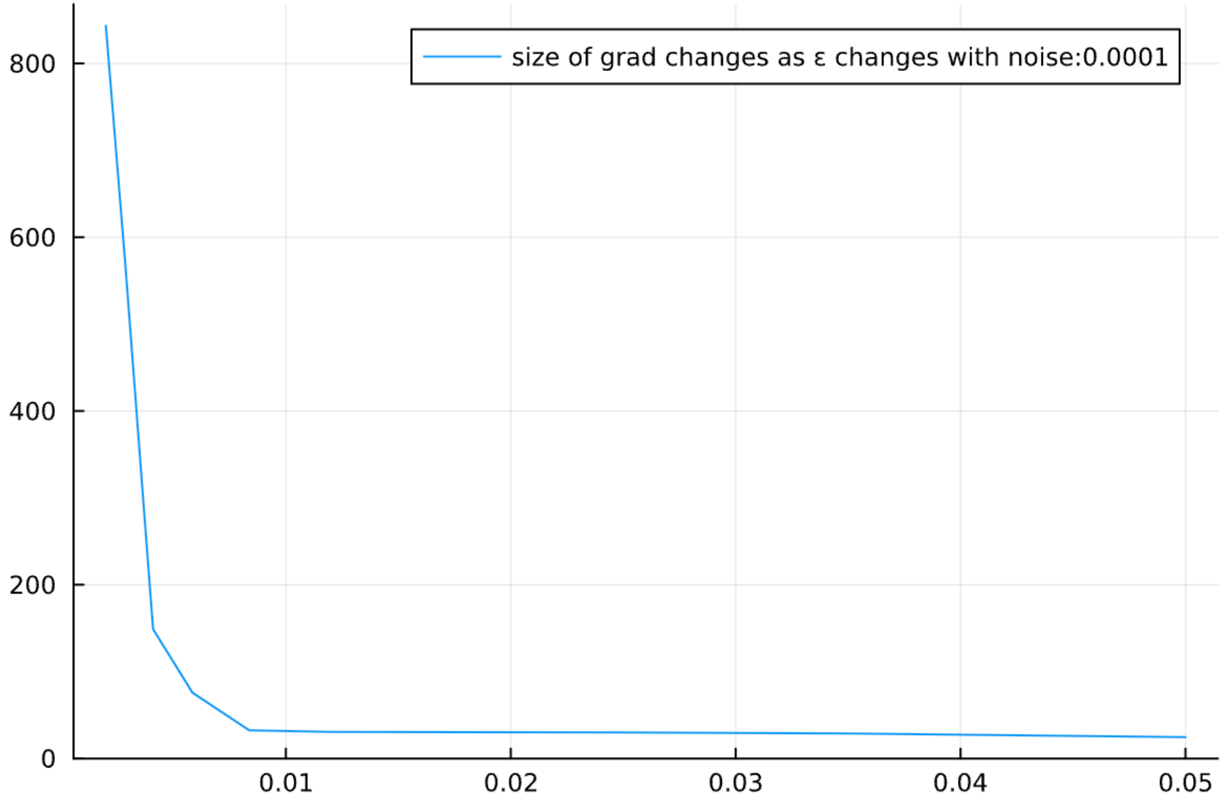
I gauss that the root cause is aaa algorithm is non-differentiable. The evidence is that, for

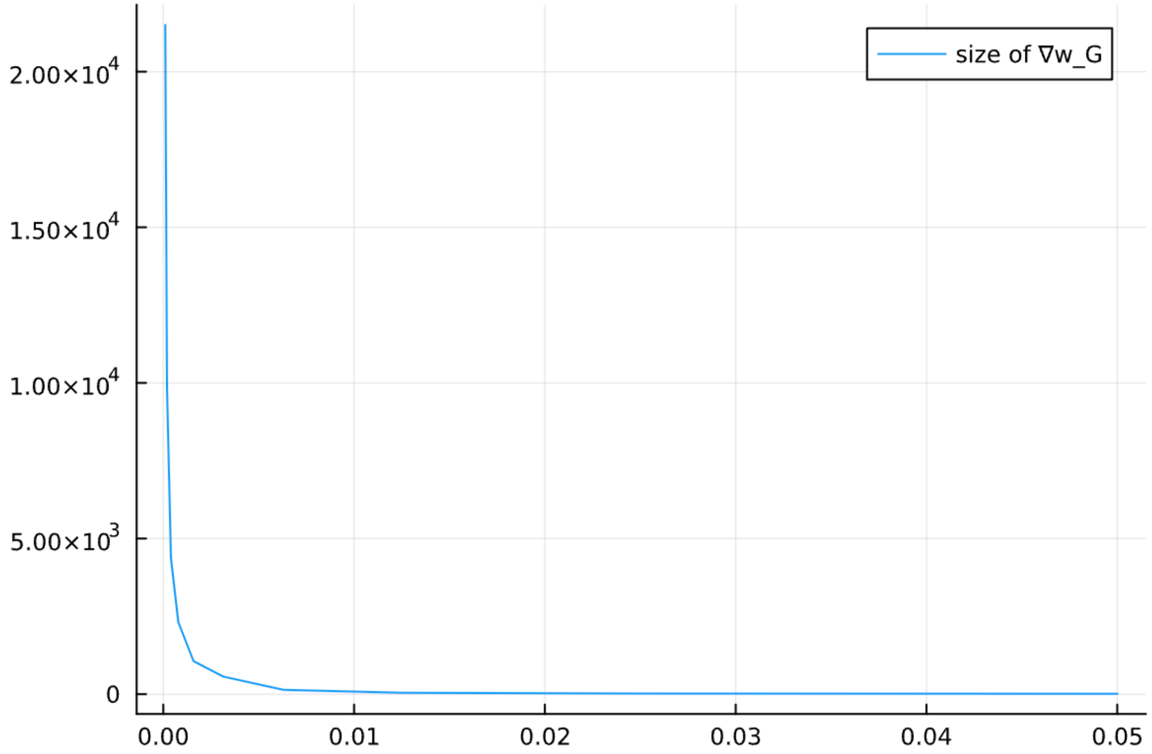
$$Giwn \to L_0$$

Even if no noise and we have only 8 $Giwn$ points, when the condition number is only about 1.1e6, but the value of

$$\|\nabla_G L_0(G)\|$$

is still greatly unstable no matter in a wide or narrow range.


size of grad changes as ε changes with noise:0.0001
800
600
400
200
0
0.01
0.02
0.03
0.04
0.05


size of ∇w_G
2.00×10^4
1.50×10^4
1.00×10^4
5.00×10^3
0
0.00
0.01
0.02
0.03
0.04
0.05

The solution I can think of is give a tight and differentiable upper bound. And apply AD on this estimulation.