

Checking algorithm

Zhuoyan Xu

After I check algorithm, I find the result in my previous note and what I reported on last week's meeting is not wrong, but in a extreme case. I'll explain in this note.

In following setting, U is ground truth, i.e. $\text{logit}(\text{tensor})$.

1 Unsupervised

I dig into the condition and some issue I ran into last week, and I find something interesting.

The erratic pattern shows because of the generated value I set. This is basically how I generated values in simulation.

```
whole_shape = rep(20,3) ; core_shape = rep(4,3)
d1 = whole_shape[1] ; d2 = whole_shape[2] ; d3 = whole_shape[3]
r1 = core_shape[1] ; r2 = core_shape[2] ; r3 = core_shape[3]

####----- generate data
set.seed(37)
A = randortho(d1)[,1:r1]
B = randortho(d2)[,1:r2]
C = randortho(d3)[,1:r3]

### G: sd = 1
G = as.tensor(array(data = rnorm(r1*r2*r3,sd = 1),dim = core_shape))

U = ttl(G,list(A,B,C),ms = c(1,2,3))@data

ts = rbinom(d1*d2*d3,1,prob = as.vector( 1/(1 + exp(-U)) ) )
ts = as.tensor(array(ts,dim = c(d1,d2,d3)))@data
```

The different G (core tensor) result in different result. I use rnorm to generate data, and all the elements in G is Gaussian with mean zero.

1.1 $\text{sd} = 1$

When I set $\text{sd} = 1$, the results are not ideal.

It might because sometimes I think it converge, but actually it didn't. (Since the improvement for logLik for each update may not show a monotone decreasing trend when updating times increasing). To make sure I set updating times up to 100. The logLik is shown like:

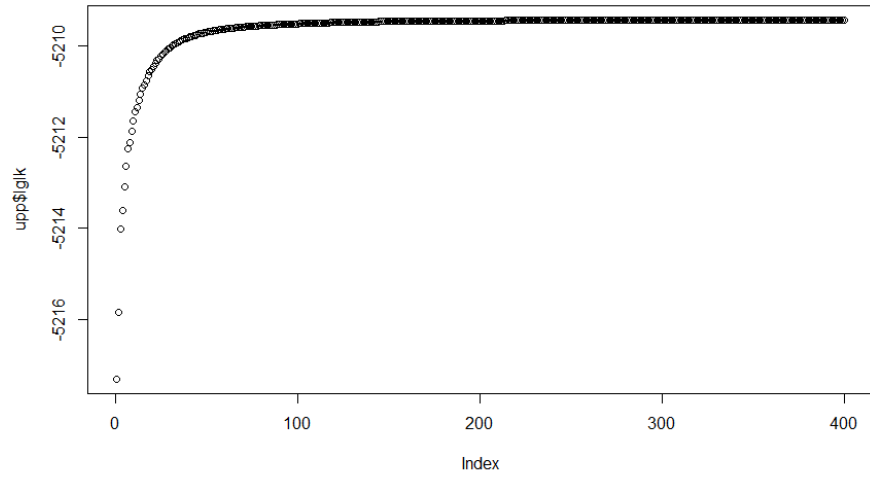


Figure 1: logLik

I plot one of the slices of the U and U-hat(they basically shows all the same pattern). It shows a erratic pattern. Like this:

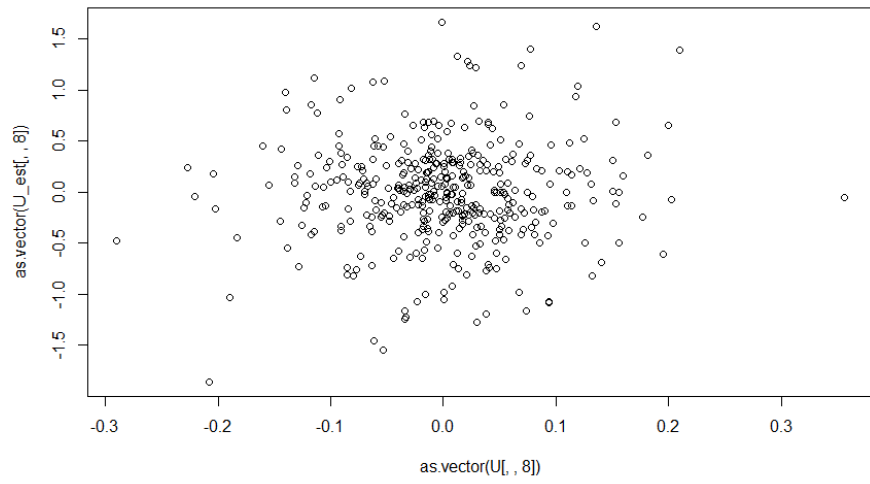


Figure 2: U slice in real value

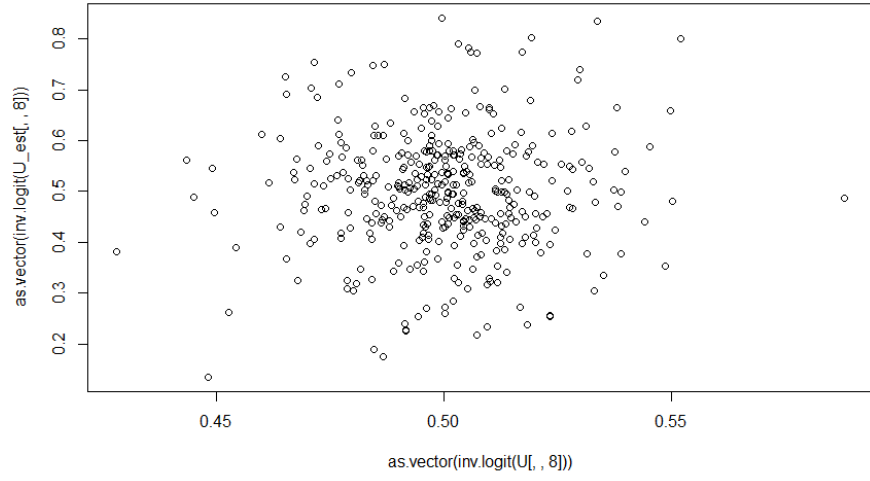


Figure 3: U slicer in logic scale

The U and U-hat tensor is shown like:

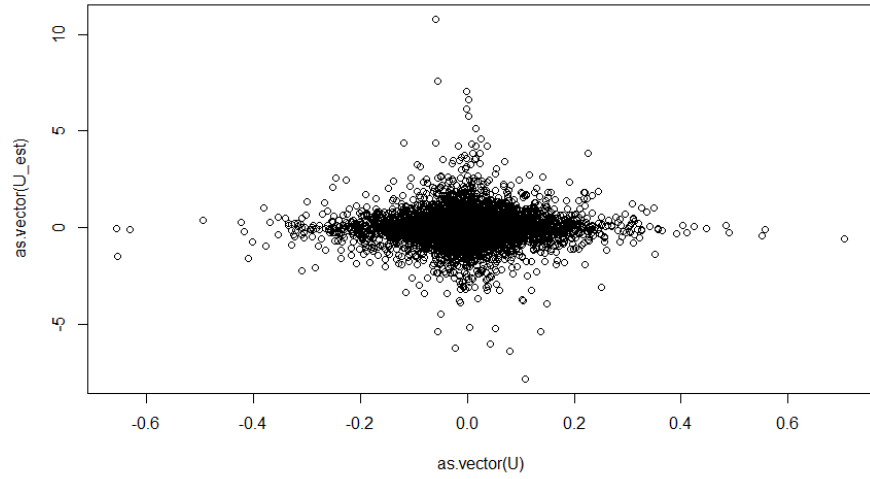


Figure 4: U tensor in real scale

1.2 $sd = 10$ or 20

I plot one of the slices of the U and U-hat(they basically shows all the same pattern). It makes more sense. Like this:

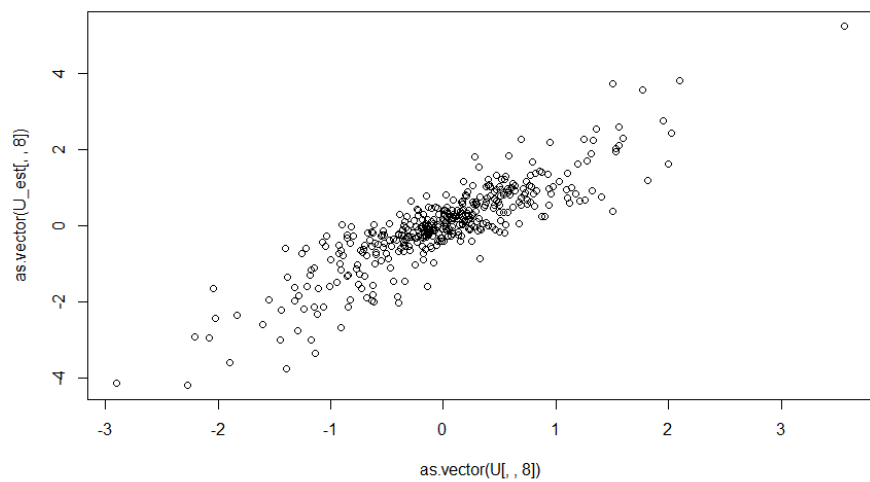


Figure 5: U slice in real value

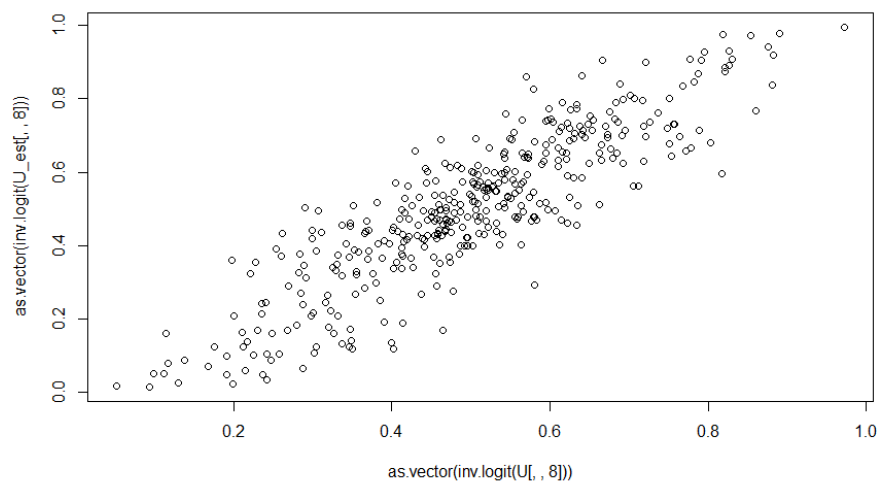


Figure 6: U slicer in logic scale

The U and U-hat tensor is shown like:

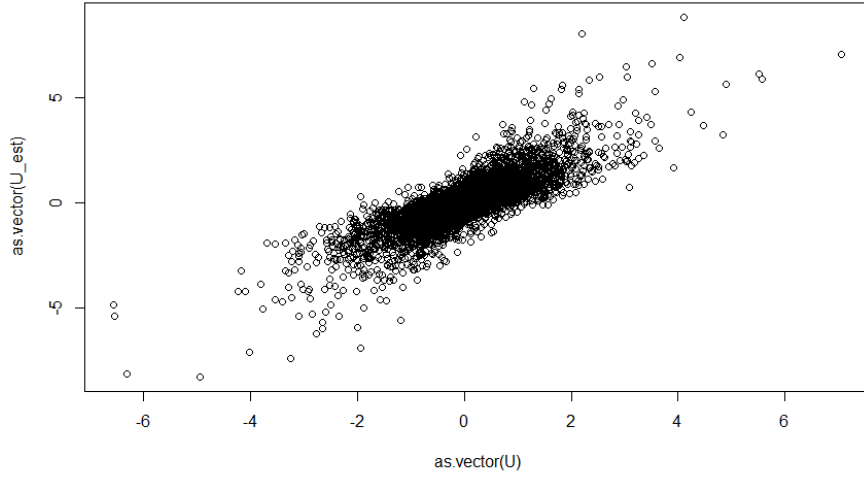


Figure 7: U tensor in real scale

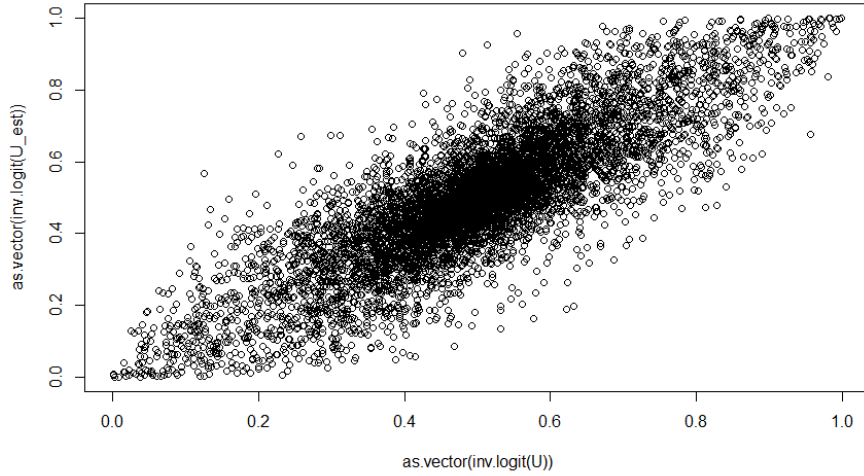


Figure 8: U tensor in real scale

And under this condition, logLik shows it converges faster than $sd=1$.

1.3 $sd = 100$

I plot one of the slices of the U and U-hat(they basically shows all the same pattern). It's more erratic than $sd=10$ or 20 . And MSE is higher, say 1961873(idea case would be 4000

6000). The case is still convergent(by logLik). But I think the result is OK, the bad MSE it's just because more noise. Since the slices comparison makes sense, as shown below:

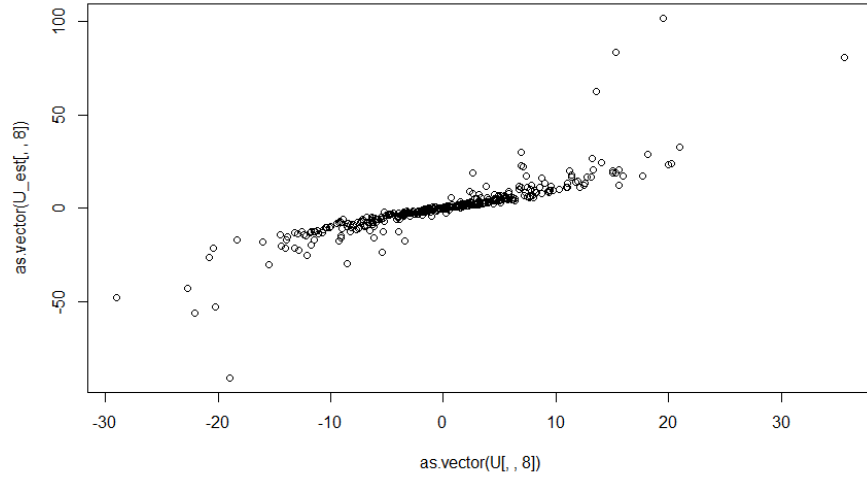


Figure 9: U slice in real value

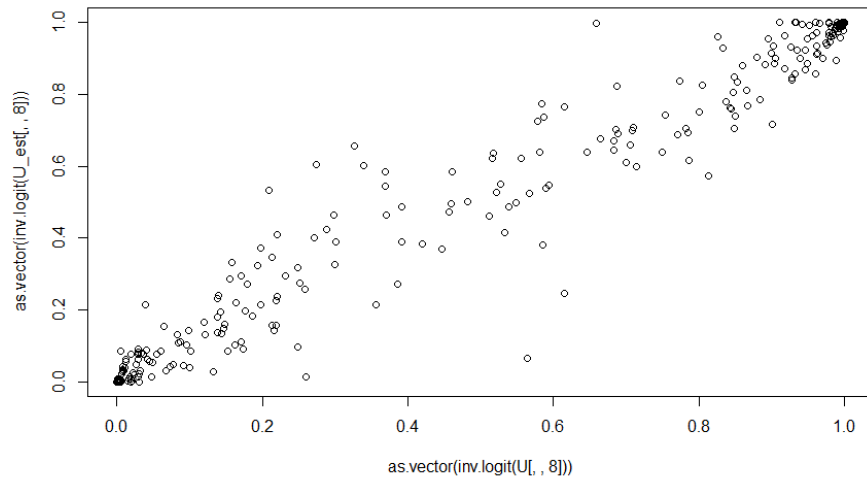


Figure 10: U slicer in logic scale

The U and U -hat tensor is shown like:

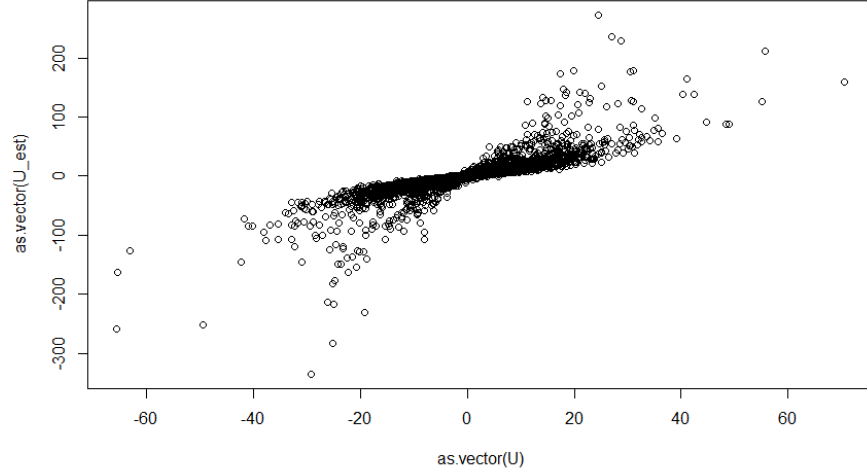


Figure 11: U tensor in real scale

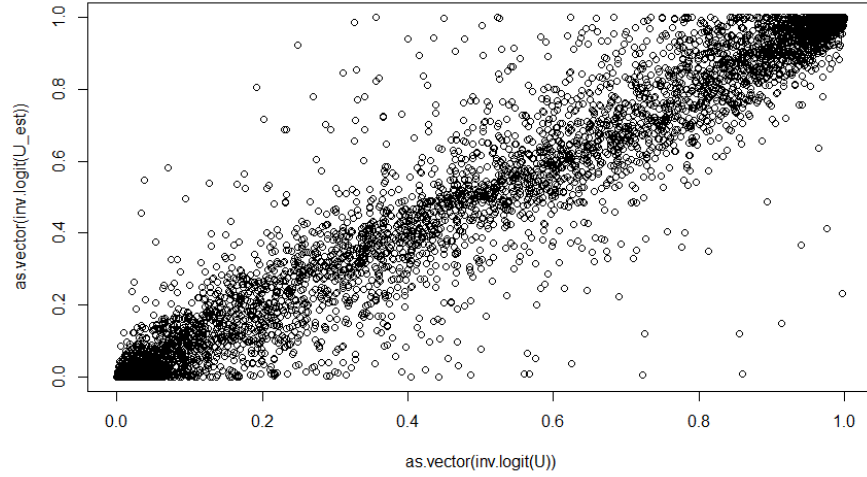


Figure 12: U tensor in real scale

1.4 Uniform distribution to Generate core tensor

Then I set `runif` to generate core tensor.

1.4.1 [0,1]

When I set `unif[0,1]`, the results are not ideal.

It might because sometimes I think it converge, but actually it didn't. (Since the improvement for `logLik` for each update may not show a monotone decreasing trend when updating times increasing). To make sure I set updating times up to 100. The `logLik` is shown like:

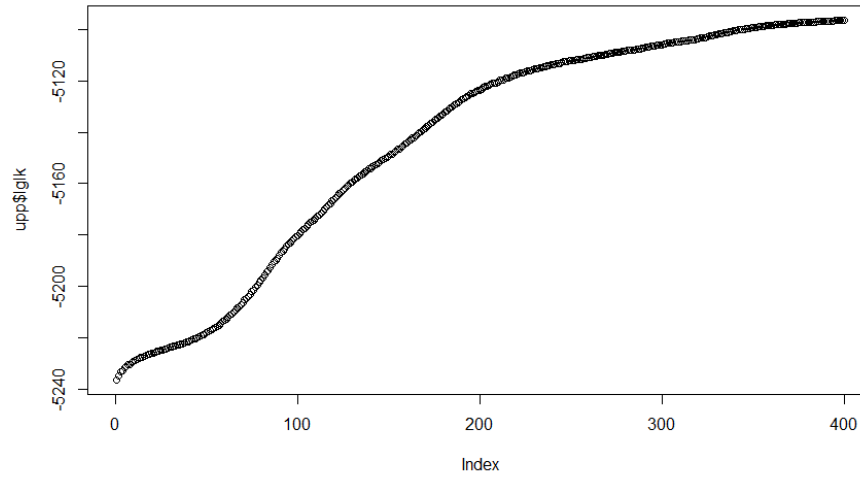


Figure 13: `logLik`

The MSE of \hat{U} is 2.930119×10^{19} .

I plot one of the slices of the U and \hat{U} (they basically show all the same pattern). It shows an erratic pattern. Like this:

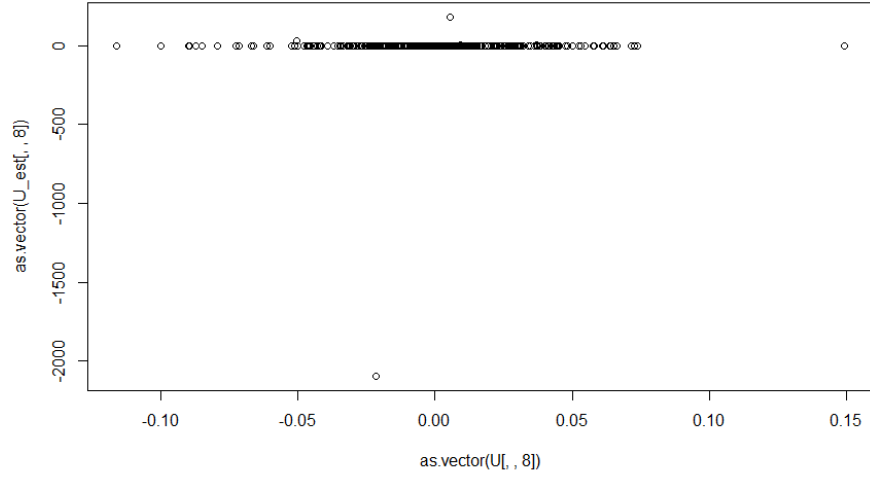


Figure 14: U slice in real value

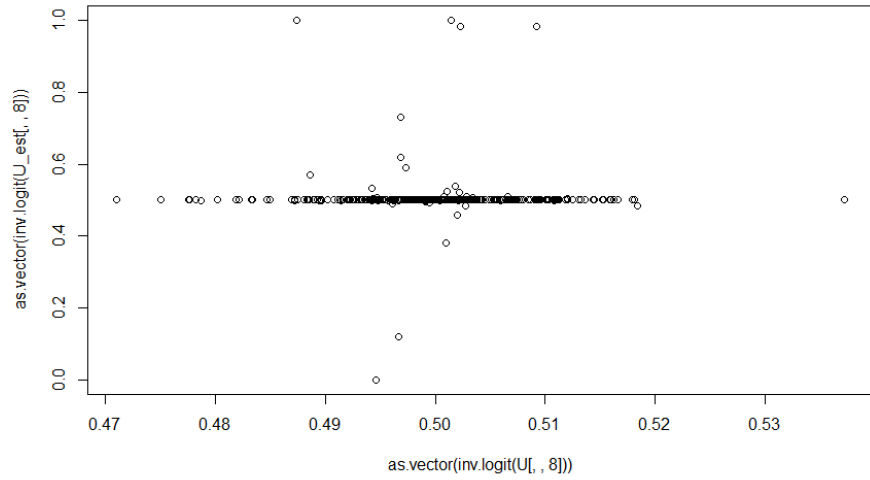


Figure 15: U slicer in logic scale

The U and U-hat tensor is shown like:

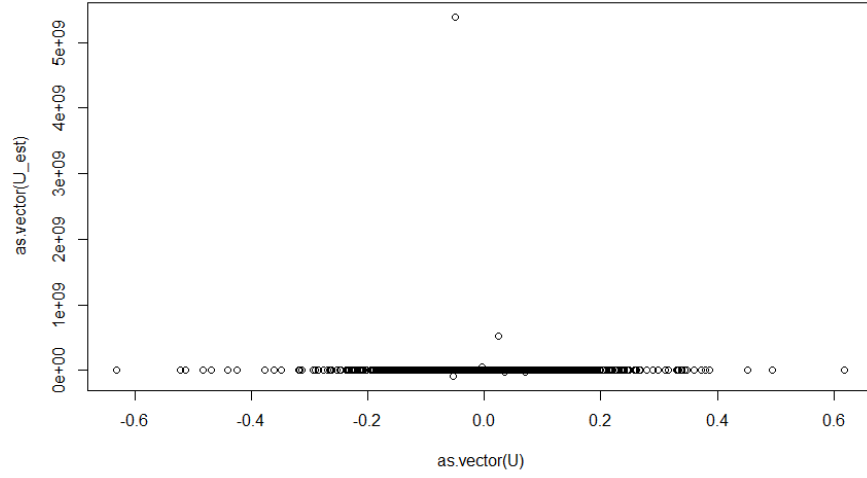


Figure 16: U tensor in real scale

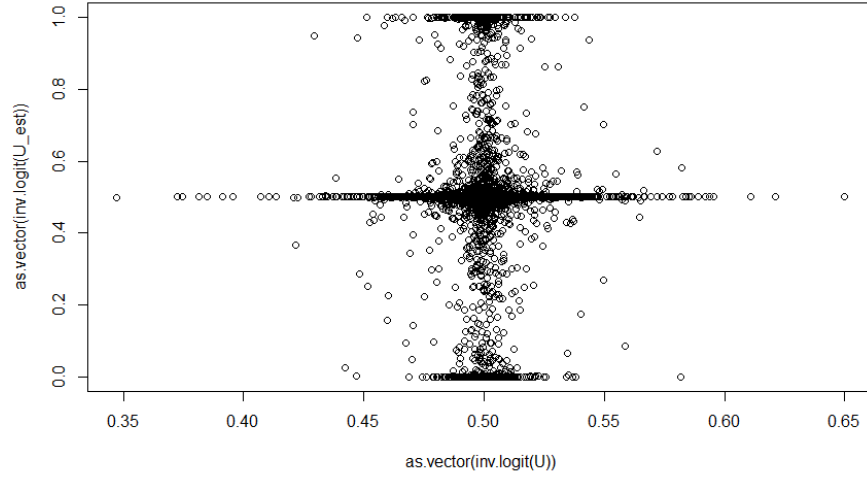


Figure 17: U tensor in log scale

1.4.2 [-10,10]

When I set uniform to be $[-10,10]$, it basically shows the same result as $\text{rnorm}(0,10)$, except relatively lower convergence than Gaussian case.

1.5 Check with initialization

In this case, it shows the same as the result without initialization. (The result is in my previous note[*Check_Algorithm_and_some_Evidence_Theory*])

2 Supervised

Still, I set

$$\begin{aligned}d_1 &= d_2 = d_3 = 20 \\ r_1 &= r_2 = r_3 = 2\end{aligned}$$

First I start with simple version, I set covariate matrix to be $\text{diag}(20)$, identity matrix with 20 rows/columns.

Then the supervised(U-hat1) and unsupervised(U-hat2) should have same result.

The Frobenious norm of two U-hats is $8.088389\text{e-}05$.

The tensor of U-hat1 and U-hat2 is:

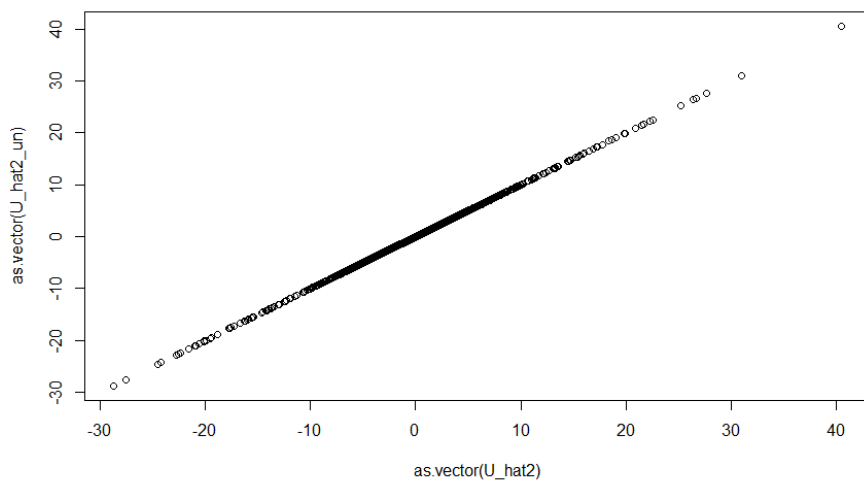


Figure 18: 2 U-hat tensor in real scale

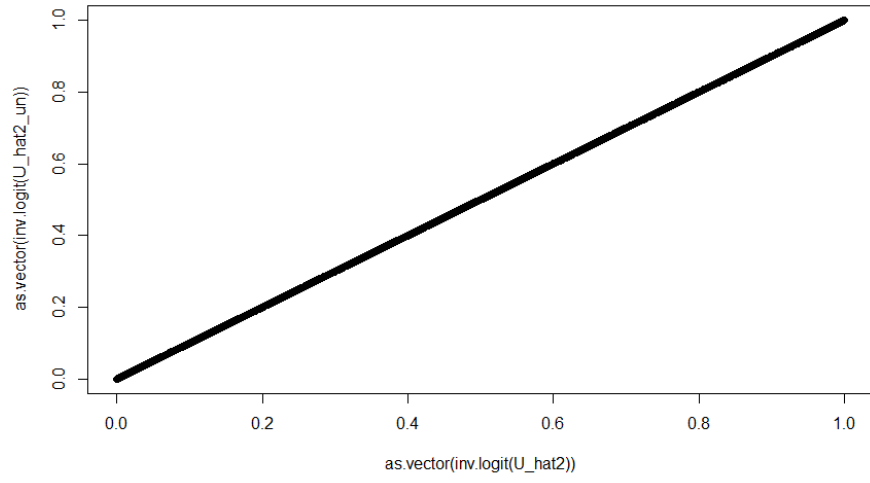


Figure 19: 2 U-hat tensor in log scale

The comparison of one of factor matrix is(others are the same):

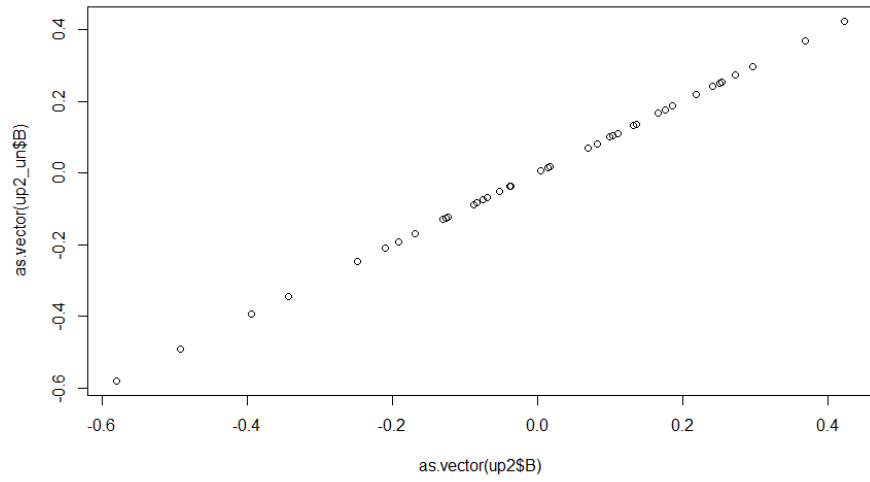


Figure 20: 2 factor matrices

The comparison of core tensor is:

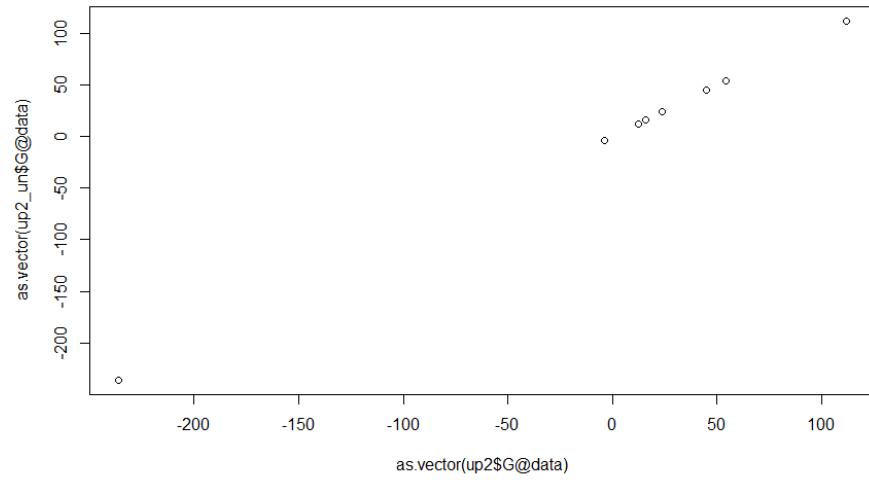


Figure 21: core tensor