11 July 2024

Dear Joan,

Thanks for your decision letter and execution report. I've studied your execution report and have managed to resolve all the issues you flagged. I believe comprehensively. Most of the issues were due to insufficiently detailed comments and commentary in the Notebooks about the correct mapping from the code output to the paper tables. I apologize for that. The revised Notebooks include more supplemental labelling of code output, additional commentary and notes in the markdown boxes and so on. Each component of printed output is now accompanied by an explicit note of where it appears in the paper (e.g., "Table 4, Column 6").

These changes should now make it possible for you to verify that there is indeed a one-to-one mapping from the code output to the numbers reported in the paper tables.

Note that for the Monte Carlo results I did use a fixed random seed (np.random.seed(seed=361)). When I re-executed the code on my computer today, I reproduced the results reported in the paper exactly. I am not sure if this will be true if you are using a slightly different version of Python and/or different machine. Regardless, what you get should be very close to what is reported.

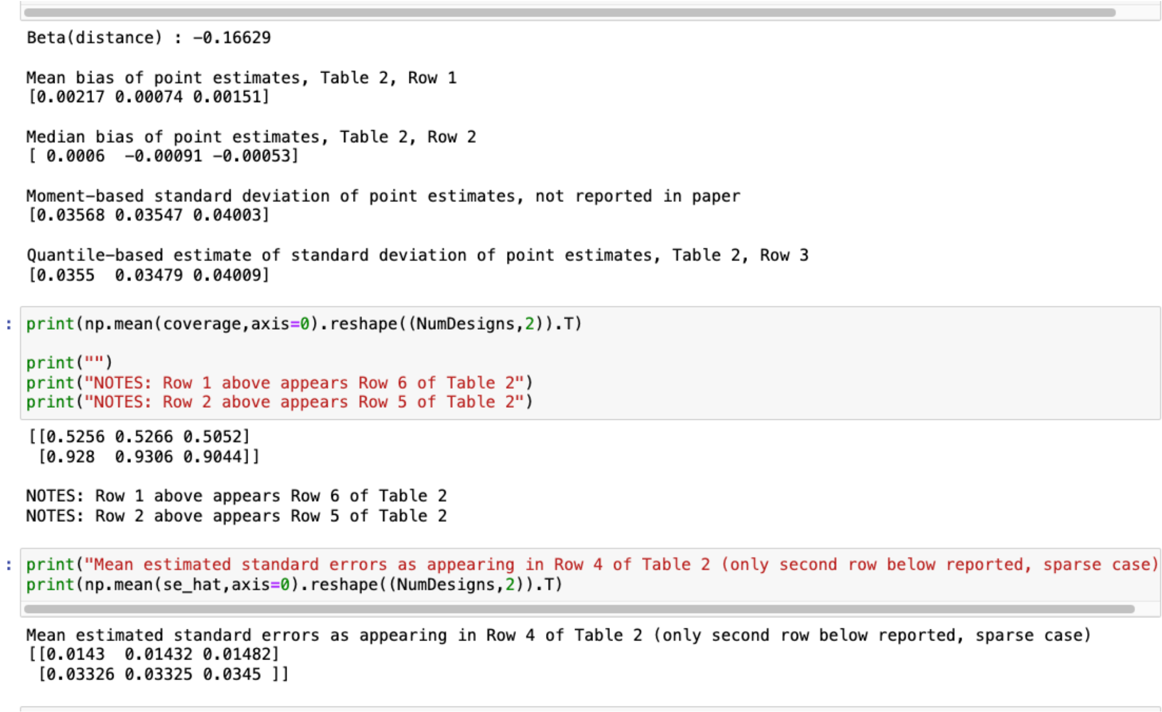
I also fixed two editing errors, which should further clarify two of your specific questions (see below for details on this).

Let me now respond to your numbered points in sequence.

2. Firstly, please provide a clearer mapping from results in the notebook to the paper, you print out mean, median and SE results for tables 2,3,4.

All the results reported in Tables 1 and 2 of the paper do indeed appear in the "Sparse_Network_Asymptotics_Notebook_Empirical_Illustration" file as claimed. The reference to Table 2 in the "Sparse_Network_Asymptotics_Notebook_Monte_Carlo" file was an editing error related to a change in table numbering that occurred across revisions. This typo has been fixed. Sorry about the confusion.

That said, you can be forgiven for having trouble finding the Table 2 results in the Notebook, as this output was not especially well formatted and/or commented. I have tried to make the mapping from code output into paper tables more explicit and overt in this revision of the replication package. I hope this makes things easier for users (and you). For example, all Table 2 output is now mapped from the code output to the paper as follows:



```
Beta(distance) : -0.16629

Mean bias of point estimates, Table 2, Row 1
[0.00217 0.00074 0.00151]

Median bias of point estimates, Table 2, Row 2
[ 0.0006  -0.00091 -0.00053]

Moment-based standard deviation of point estimates, not reported in paper
[0.03568 0.03547 0.04003]

Quantile-based estimate of standard deviation of point estimates, Table 2, Row 3
[0.0355  0.03479 0.04009]
```

```
print(np.mean(coverage,axis=0).reshape((NumDesigns,2)).T)

print("")
print("NOTES: Row 1 above appears Row 6 of Table 2")
print("NOTES: Row 2 above appears Row 5 of Table 2")
```

```
[[0.5256 0.5266 0.5052]
 [0.928  0.9306 0.9044]]

NOTES: Row 1 above appears Row 6 of Table 2
NOTES: Row 2 above appears Row 5 of Table 2
```

```
print("Mean estimated standard errors as appearing in Row 4 of Table 2 (only second row below reported, sparse case)
print(np.mean(se_hat,axis=0).reshape((NumDesigns,2)).T)
```

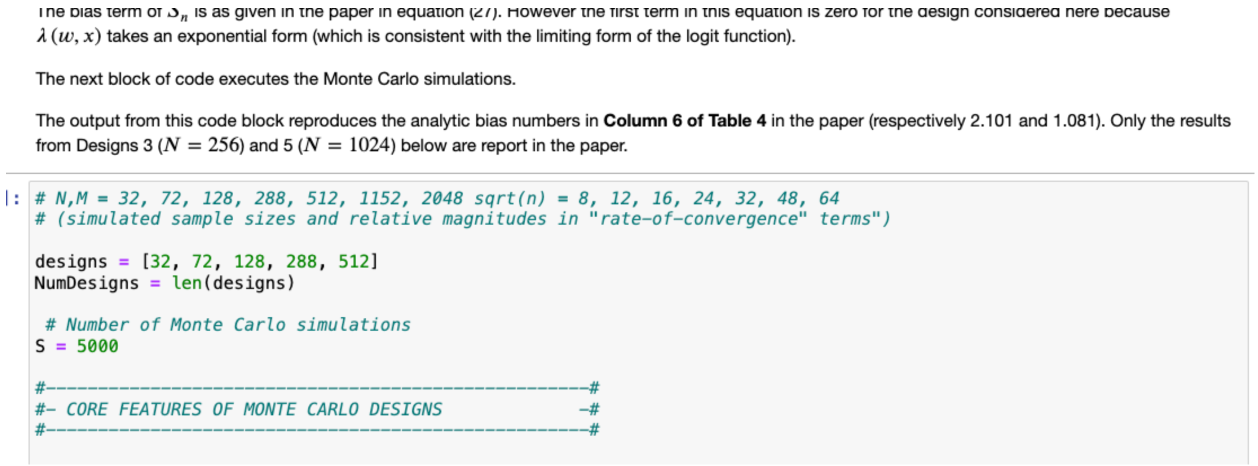
```
Mean estimated standard errors as appearing in Row 4 of Table 2 (only second row below reported, sparse case)
[[0.0143  0.01432 0.01482]
 [0.03326 0.03325 0.0345 ]]
```

Hopefully this makes it a bit easier to map from the code output to what is presented in the official paper tables.

The results reported in Table 4 also fully appear in the output associated with “Sparse_Network_Asymptotics_Notebook_Monte_Carlo” file. Although, again, you can be forgiven for having trouble finding them (sorry)! I have now added additional commentary to the printed code output as well as to the various the markdown boxes in the Notebook to make the mapping explicit.

Note the analytic score bias calculations that appear in Column 6 of Table 4 are a bit harder to find. They appear in the execution output associated with the initial Monte Carlo run. Here are some screen shots showing this.

First a note flagging for the reader where to look for the bias expressions:



The bias term of S_n is as given in the paper in equation (27). However the first term in this equation is zero for the design considered here because $\lambda\,(w, x)$ takes an exponential form (which is consistent with the limiting form of the logit function).

The next block of code executes the Monte Carlo simulations.

The output from this code block reproduces the analytic bias numbers in **Column 6 of Table 4** in the paper (respectively 2.101 and 1.081). Only the results from Designs 3 ($N = 256$) and 5 ($N = 1024$) below are report in the paper.

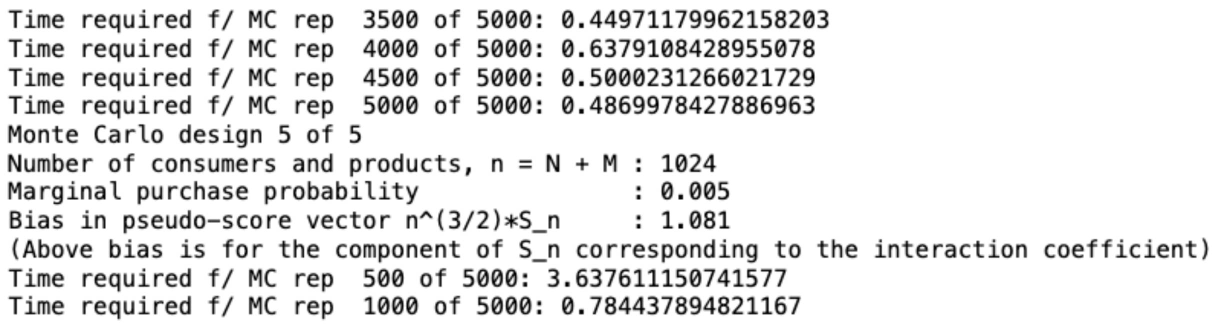
```
# N,M = 32, 72, 128, 288, 512, 1152, 2048 sqrt(n) = 8, 12, 16, 24, 32, 48, 64
# (simulated sample sizes and relative magnitudes in "rate-of-convergence" terms")

designs = [32, 72, 128, 288, 512]
NumDesigns = len(designs)

 # Number of Monte Carlo simulations
S = 5000

#----------------------------------------------------#
#- CORE FEATURES OF MONTE CARLO DESIGNS             -#
#----------------------------------------------------#
```

Second, an example of the actual printed bias expression (This one appears in Row 7, Column 6 of Table 4):



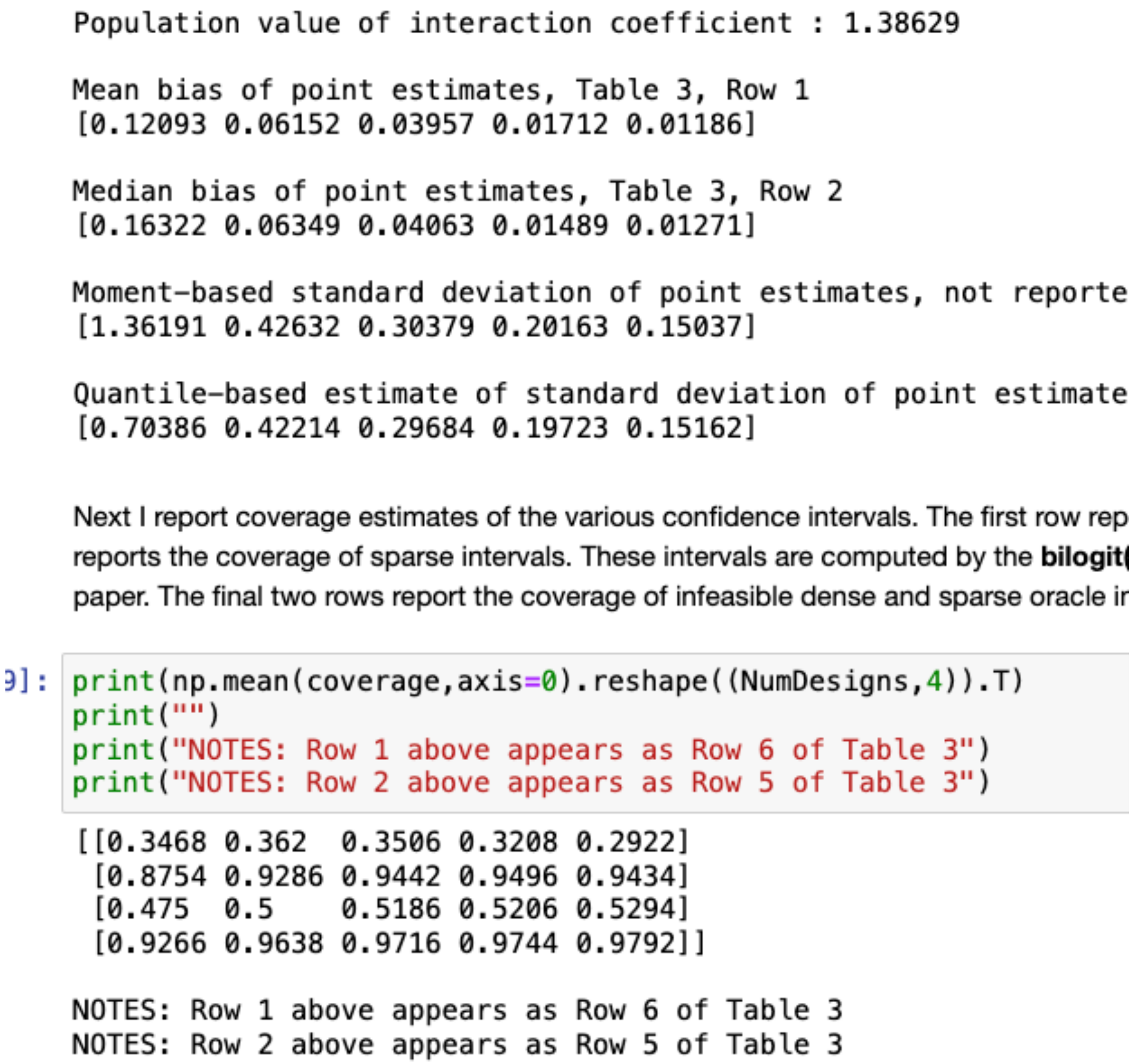
```
Time required f/ MC rep  3500 of 5000: 0.44971179962158203
Time required f/ MC rep  4000 of 5000: 0.6379108428955078
Time required f/ MC rep  4500 of 5000: 0.5000231266021729
Time required f/ MC rep  5000 of 5000: 0.4869978427886963
Monte Carlo design 5 of 5
Number of consumers and products, n = N + M : 1024
Marginal purchase probability               : 0.005
Bias in pseudo-score vector n^(3/2)*S_n     : 1.081
(Above bias is for the component of S_n corresponding to the interaction coefficient)
Time required f/ MC rep  500 of 5000: 3.637611150741577
Time required f/ MC rep  1000 of 5000: 0.784437894821167
```

3. Secondly, in the ReadMe you state that results may be different up to a simulation error, that is acceptable, however as you can see in the following screenshot the results differ for some by a more significant margin. Have I misread the table, or is there another reason?

There are two reasons for the discrepancy.

First, in the code I sent you I had inadvertently set the number of Monte Carlo simulations to 500 (I likely did this as I was debugging and checking the code to make sure it executed prior to sending off the replication package). If you executed the code without changing this number from 500 to 5000 (which I assume was the case), then it seems quite likely that simulation error would be *much larger* than what I claimed it should be in, for example, the notes to the tables. This is because in the paper I report results based on 5000 simulations. With this correction made Tables 3 and 4 now replicate exactly (as expected).

For example, the Table 3 results are shown in the code output below and they match what is reported in the paper exactly.



```
Population value of interaction coefficient : 1.38629

Mean bias of point estimates, Table 3, Row 1
[0.12093 0.06152 0.03957 0.01712 0.01186]

Median bias of point estimates, Table 3, Row 2
[0.16322 0.06349 0.04063 0.01489 0.01271]

Moment-based standard deviation of point estimates, not reporte
[1.36191 0.42632 0.30379 0.20163 0.15037]

Quantile-based estimate of standard deviation of point estimate
[0.70386 0.42214 0.29684 0.19723 0.15162]
```

Next I report coverage estimates of the various confidence intervals. The first row rep
reports the coverage of sparse intervals. These intervals are computed by the **bilogit(**
paper. The final two rows report the coverage of infeasible dense and sparse oracle ir

```
9]: print(np.mean(coverage,axis=0).reshape((NumDesigns,4)).T)
    print("")
    print("NOTES: Row 1 above appears as Row 6 of Table 3")
    print("NOTES: Row 2 above appears as Row 5 of Table 3")
```

```
[[0.3468 0.362  0.3506 0.3208 0.2922]
 [0.8754 0.9286 0.9442 0.9496 0.9434]
 [0.475  0.5    0.5186 0.5206 0.5294]
 [0.9266 0.9638 0.9716 0.9744 0.9792]]

NOTES: Row 1 above appears as Row 6 of Table 3
NOTES: Row 2 above appears as Row 5 of Table 3
```

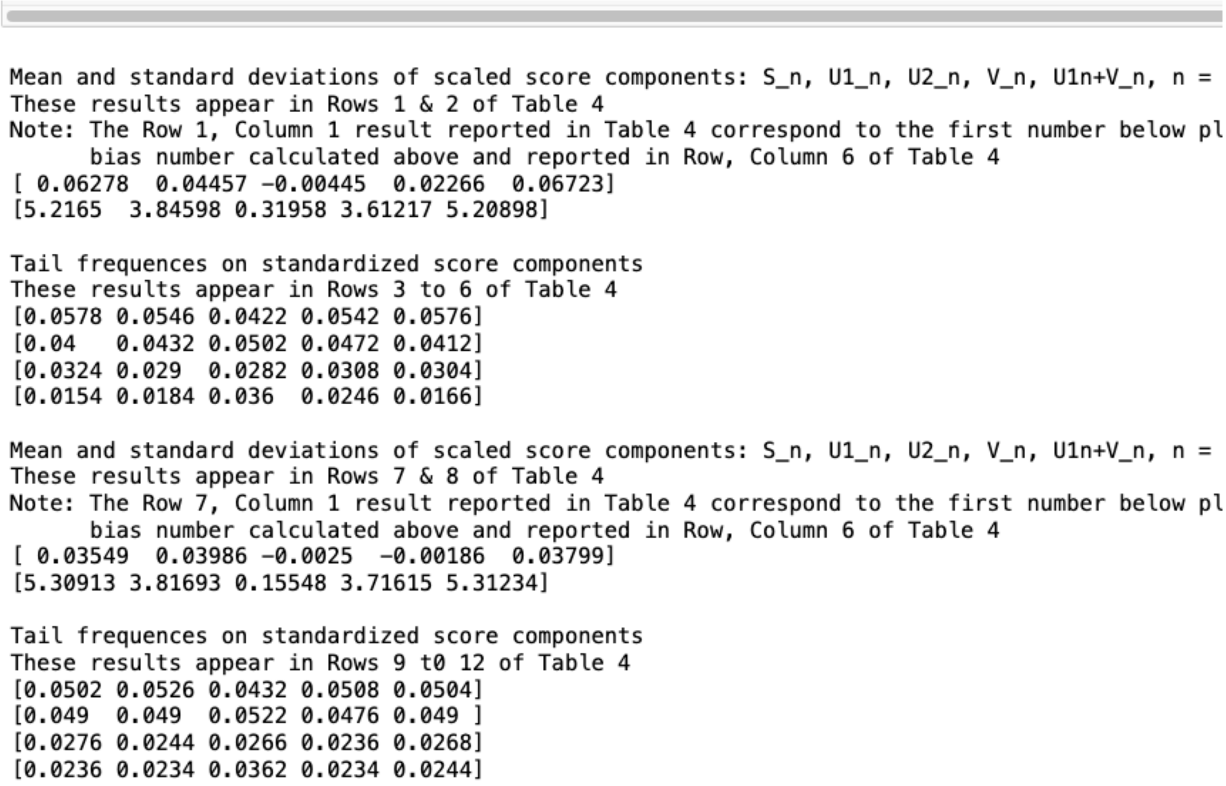
Note the precise difference you pointed out in your letter actually involved a comparison of a moment-based standard deviation estimate (which is not reported in the paper) with the quantile-based estimate (which is reported in the paper). So that particular comparison was affected by both high simulation error (due to me not setting the number of simulations correctly) and an "apples to oranges”" issue.

The moment-based estimate (not reported in the paper) is now overtly flagged as such in the code output, while the quantile-based estimate (reported in the paper and described in the notes to Tables 2 and 3) is also flagged. See the notes to Tables 2 and 3 and the revised notebooks.

As a side note....I routinely report quantile-based estimates of standard deviations because it is non-uncommon for an estimate to lack certain moments in finite samples, while such moments generally exist in the asymptotic distribution (LIML is a famous example). This is a trick I learned from my advisor, Gary Chamberlain. That said, the moment- and quantile-based standard deviation estimates are fairly close except for the n=64 case you flagged.

Although I reported mean bias in the paper, I personally prefer median bias estimates (also reported) for similar reasons.

For completeness the code output replicating Table 4 is as follows:



```
Mean and standard deviations of scaled score components: S_n, U1_n, U2_n, V_n, U1n+V_n, n =
These results appear in Rows 1 & 2 of Table 4
Note: The Row 1, Column 1 result reported in Table 4 correspond to the first number below pl
      bias number calculated above and reported in Row, Column 6 of Table 4
[ 0.06278  0.04457 -0.00445  0.02266  0.06723]
[5.2165  3.84598 0.31958 3.61217 5.20898]

Tail frequences on standardized score components
These results appear in Rows 3 to 6 of Table 4
[0.0578 0.0546 0.0422 0.0542 0.0576]
[0.04   0.0432 0.0502 0.0472 0.0412]
[0.0324 0.029  0.0282 0.0308 0.0304]
[0.0154 0.0184 0.036  0.0246 0.0166]

Mean and standard deviations of scaled score components: S_n, U1_n, U2_n, V_n, U1n+V_n, n =
These results appear in Rows 7 & 8 of Table 4
Note: The Row 7, Column 1 result reported in Table 4 correspond to the first number below pl
      bias number calculated above and reported in Row, Column 6 of Table 4
[ 0.03549  0.03986 -0.0025  -0.00186  0.03799]
[5.30913 3.81693 0.15548 3.71615 5.31234]

Tail frequences on standardized score components
These results appear in Rows 9 t0 12 of Table 4
[0.0502 0.0526 0.0432 0.0508 0.0504]
[0.049  0.049  0.0522 0.0476 0.049 ]
[0.0276 0.0244 0.0266 0.0236 0.0268]
[0.0236 0.0234 0.0362 0.0234 0.0244]
```

This too matches what is reported in the paper exactly. Note that the number in Row 1, Column 1 in Table 4 equals 0.06278 (listed in the output above) + 2.101 (analytic bias, computer earlier in the notebook). A similar process delivers the Row 7, Column 1 number in Table 4.

In any case, to re-iterate, when I re-executed the code today, the numbers I got for Tables 3 and 4 matched those reported in the paper exactly. While, for the reasons noted earlier, they may not replicate exactly for you, they should now be very close (since the number of simulations has been corrected to equal 5000, as opposed to 500, simulation error should be modest). Sorry for that goof.

I hope the additional comments in the markdown boxes and added labelling of the actual code output also helps you connect the dots more easily. Again, sorry that was not clearer at the outset.

Thanks for your work in looking at my code. I hope the changes in the included archive clarify/correct the issues you flagged. I know this is time consuming on your part. I am appreciative of your care and thoughtfulness.

The corrected replication archive is attached. I have also pushed all these corrections to the GitHub “netrics” repo…so they are easy to find.

Take care.

Bryan