Some simulations

Chanwoo Lee, July 11, 2021

1 Simulation settings

I generate the networks from the four graphons listed in Table 1, selected to have different features in different combinations of monotonic degree and the rank. Graphon 1 has $K = \lfloor \log n \rfloor$ blocks with different within block edge probabilities, which all dominate the low between-block probability. Graphon 2 lacks node degree monotonicity with small rank while Graphon 4 has monotonic degree with full rank. Notice that graphon 4 has repeating structures as in our previous project. Graphon 3 has difficult characteristics to estimate because it has no monotonic degrees and high rank. Table 2 shows the hypergraphons that extends graphons in Table 1.

Graphon	Function $f(x, y)$	Monotone degrees	Rank
1	$f(x,y) = \begin{cases} k/(K+1) & \text{if } x, y \in ((k-1)/K, k/K) \\ 0.4/(K+1) & \text{otherwise; } K = \lfloor \log n \rfloor \end{cases}$	Yes	$\lfloor \log n \rfloor$
2	$f(x,y) = \sin(5\pi(x+y)+1)/2 + 0.5$	No	3
3	$f(x,y) = \min((x^2+y^2)/3^{\cos(1/(x^2+y^2))}, 1)$	No	Full
4	$f(x,y) = \min(x, y)$	Yes	full

Table 1: Synthetic graphons for matrix case

Graphon	Function $f(x, y, z)$	Monotone degrees	Rank
1	$f(x,y,z) = \begin{cases} k/(K+1) & \text{if } x, y, z \in ((k-1)/K, k/K) \\ 0.4/(K+1) & \text{otherwise; } K = \lfloor \log n \rfloor \end{cases}$	Yes	$\lfloor \log n \rfloor$
2	$f(x,y,z) = \sin(5\pi(x+y+z)+1)/2 + 0.5$	No	Low
3	$f(x,y,z) = \min((x^2+y^2+z^2)/3^{\cos(1/(x^2+y^2+z^2))}, 1)$	No	Full
4	$f(x,y,z) = \min(x, y, z)$	Yes	full

Table 2: Synthetic graphon for tensor case

Good simulation setup!

what's the difference between these two under the matrix setting?

For matrix case, I compare three different methods: sort-and-smoothing (SAS), squre spectral (Spectral), and High-order spectral (Hspectral) methods. For tensor case, I include Stochastic block method (SBM) with HSC initialization. Since our `tbmClustering` algorithm is only available on 3-order tensor, I skipped the comparison with this method on matrix case.

2 Simulation results

First, I visualize the true probability matrix versus estimated one in matrix case when dimension $d = 50$. Figure 1 shows that SAS method is only working great for graphon 4, which has monotonic degrees. The other two methods perform moderately well in all settings and Spectral method works better than high-order spectral method consistently.

Second, I compare the mean squared errors with matrix/tensor dimension. Figure 2 shows that generally, Spectral method performs the best in matrix case while Hspectral method has the best performance in tensor case. In graphon 4 model, SAS method is always the best. This is because graphon 4 model matches well with the assumptions for SAS method. It seems that the performance

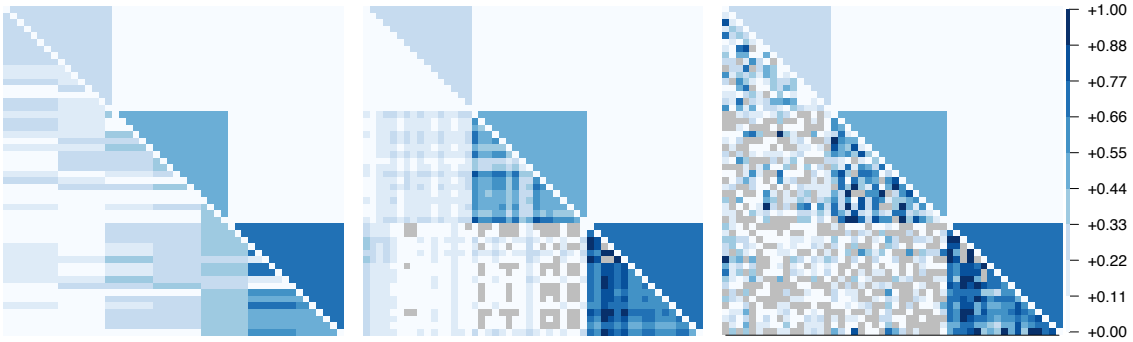
of SBM is bad considering the algorithmic time complexity. One possible explanation is that we set the rank $k = d^{\frac{m}{m+2}}$, which other SAS and Hspectral method take. However, optimal k for SBM is $k = d^{\frac{m}{m+\alpha}}$. why alpha?

Distinguish two SBMs:
1) Empirical SBM: local optimizer returned from our algorithm
2) Oracle SBM: global optimizer defined based on argmax(objective function).

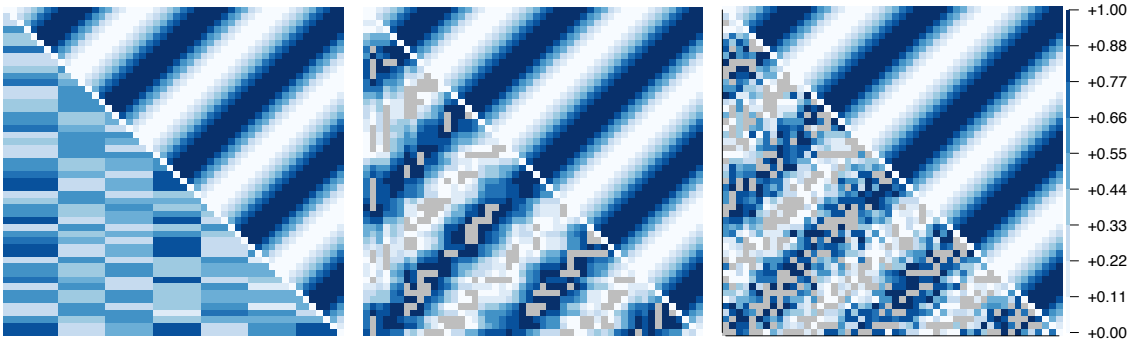
Oracle SBM is the best estimate; however, there is no way to obtain it in practice.
Empirical SBM is the one you included in the figure.
The suboptimality is due to the higher error variance than what would be for oracle SBM.

In simulation, we can ``estimate'' the oracle SBM as follows:
1. Simulate hypergraphon under fixed designs.
2. Keep (x,y,z) in original (increasing) orders —> the ground truth sorting.
In application, we will have no access to true sorting. In simulation, we unmask true sorting for the purpose of comparison.
3. Run SAS by using the ground truth sorting in place of estimated sorting.

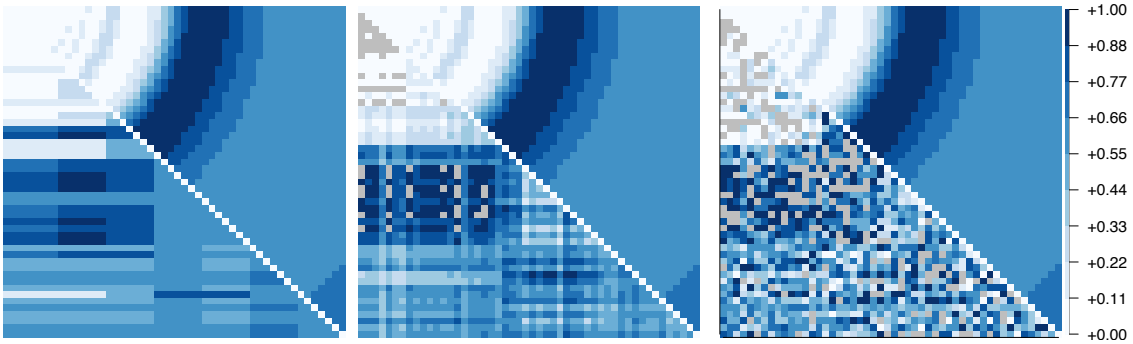
Add oracle SBM in comparison —> serves as the golden criteria benchmark.



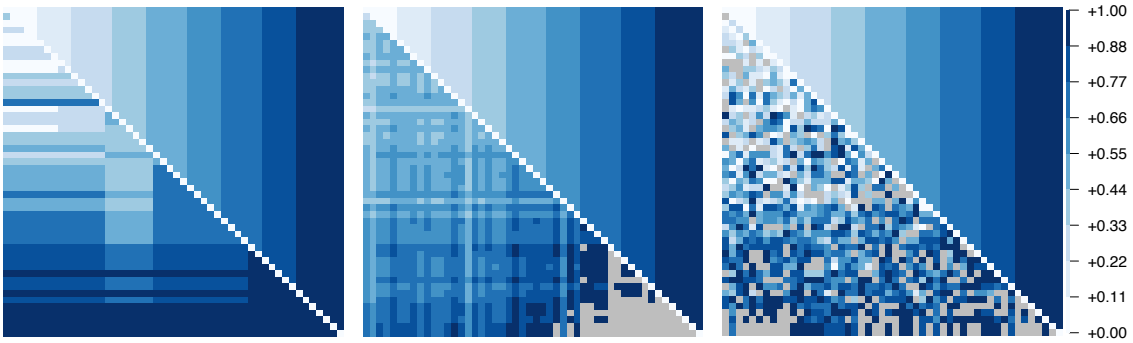
(a) Graphon 1



(b) Graphon 2



(c) Graphon 3



(d) Graphon 4

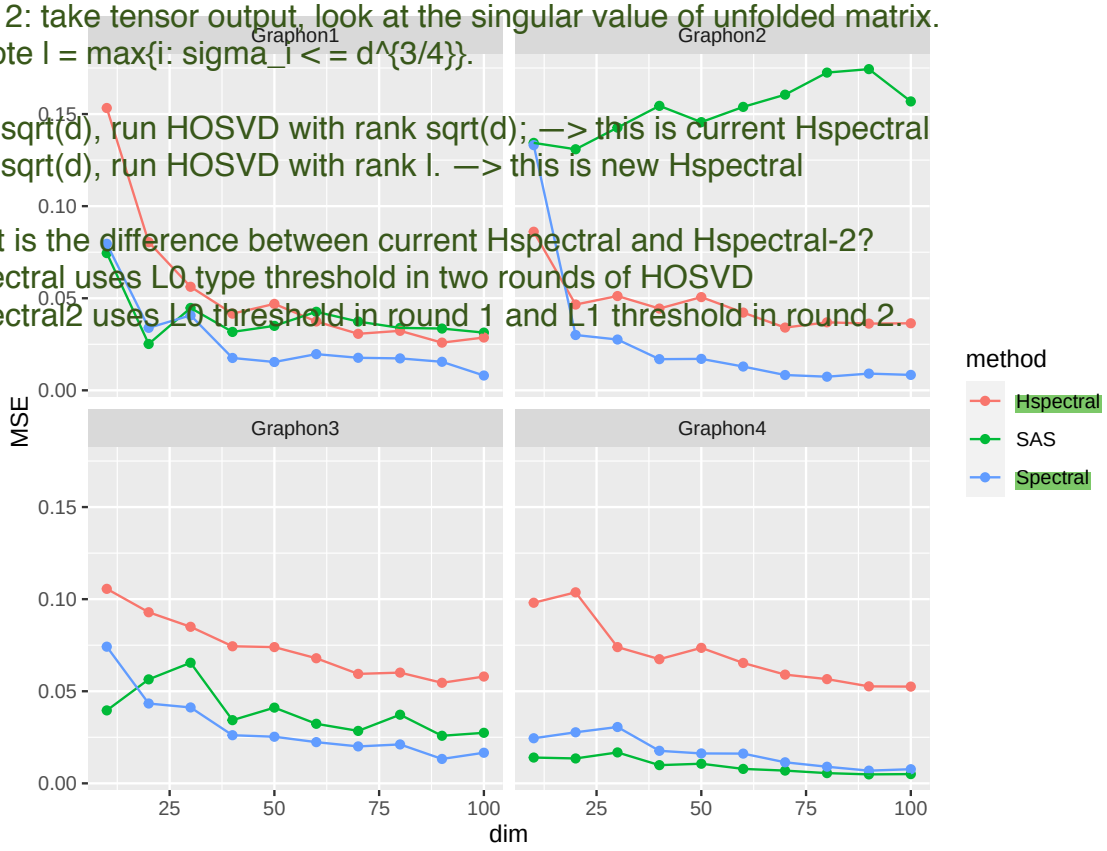
I like your visualization.

Figure 1: Estimated probability matrices for graphons 1-4, shown in rows 1-4. Column 1: true P (upper) and SAS method (lower). Column 2 true P (upper) and Spectral method (lower). Column 3 true$P$ (upper) and High-order spectral method (lower). Gray colored entries have the values outside of the range $[0, 1]$.

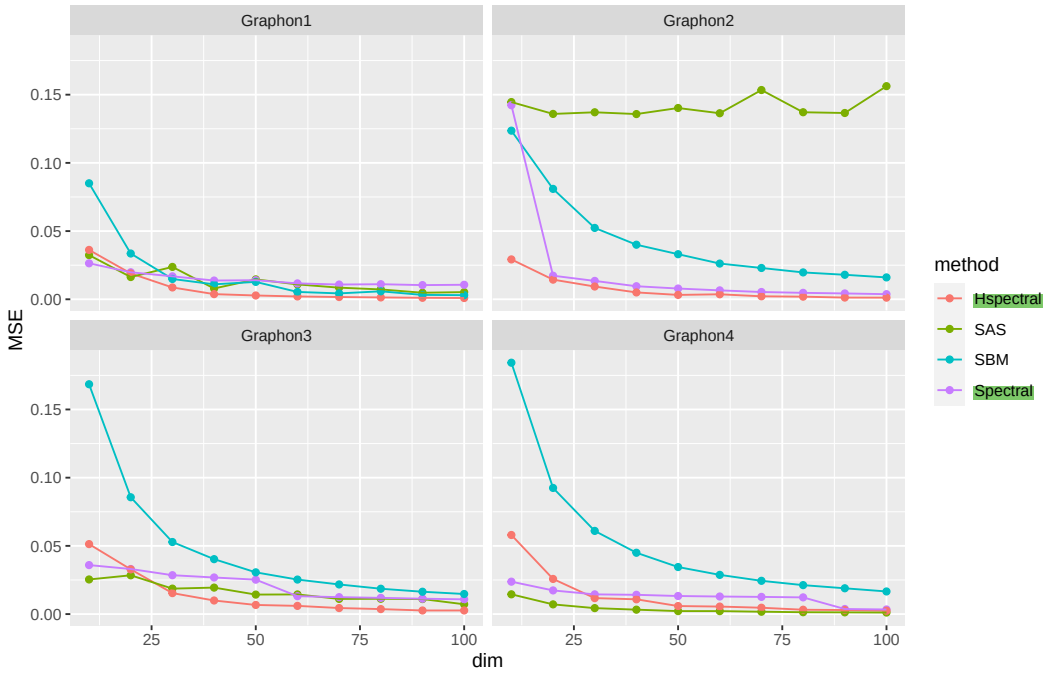
Add another Hspectral-2:
Step 1. run HOSVD with rank sqrt(d)
Step 2: take tensor output, look at the singular value of unfolded matrix.
Denote l = max{i: sigma_i < = d^{3/4}}.

If l > sqrt(d), run HOSVD with rank sqrt(d); —> this is current Hspectral
If l < sqrt(d), run HOSVD with rank l. —> this is new Hspectral

What is the difference between current Hspectral and Hspectral-2?
Hspectral uses L0 type threshold in two rounds of HOSVD
Hspectral2 uses L0 threshold in round 1 and L1 threshold in round 2.



(a) Mean squared error of estimated probability matrices versus matrix dimension of each method on four different graphons.



(b) Mean squared error of estimated probability tensors versus tensor dimension of each method on four different graphons.

Figure 2: Mean squared error of estimated probability versus dimension. Stochastic block method is included in tensor case.