Some simulations

Chanwoo Lee, July 13, 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.

x+y —> [1,…,d]\otime[1,…,1]+[1,…,1]\otimes[1,…,d]

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

ground truth order of x
[1/d, …, 1]

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

For matrix case, I compare three different methods: sort-and-smoothing (SAS), squre spectral (Spectral), and High-order spectral (Hspectral) methods. In matrix case, spectral method collects singular vectors and values that exceeds threshold $\sqrt{d}$

$$\hat{\Theta} = \sum_{i:\, \sigma_i(Y) \geq \sqrt{d}} \sigma_i(Y) u_i v_i^T,$$

where u_i, v_i and $\sigma_i(Y)$ are left and right i-th largest singular vectors and i-th largest singular value of the observation Y respectively. Hspectral method estimates the signal matrix by the following procedure,

$$\begin{aligned}
&\tilde{U} = \text{SVD}_r(Y), \quad \tilde{V} = \text{SVD}_r(Y^T) \\
&\hat{U} = \text{SVD}_r(Y\tilde{V}), \quad \hat{V} = \text{SVD}_r(Y^T\tilde{U}) \\
&\hat{\Theta} = (\hat{U}\hat{U}^T)Y(\hat{V}\hat{V}^T).
\end{aligned}$$

Y = [u1, …,, ud] diag [v1, …, vd]^T

span {[u1, …, ud]^T * Y} —> span{v}

Notice that Spectral method does not require pre-specified rank but it uses truncated SVD while Hspectral needs true rank of signal matrix X as an input.

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 3 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 4 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 $k = d^{\frac{m}{m+2\alpha}}$ where α is a smoothness of the graphon. This is because the statistical error of the SBM estimation is optimized by balancing two source of errors: block approximation error $(1/k^{2\alpha})$ and estimation error (k^m/d^m).

Update: I realized that I used the option `diagP = F` on `tbmClustering` algorithm, which gives us 0 diagonal probability. Since I made graphon functions take non-zero diagonal entries, I changed this option accordingly as `diagP = T`.

Update 2: I added the Spectral 2 method based on the comments. Figure 2 shows Hspectral2 has imporved performance from both Spectral and HSpectral method. In addition, I have checked the performance of Hspectral2 in matrix case. It turns out that Hspectral2 has the exactly the same performance with Spectral method (see Figure 4a), which implies $\max\{i\colon \sigma_i(A) \geq \sqrt{d}\} < \sqrt{d}$.

Spectral:
Y: truncated singular values at sqrt(d)

Hspectral2:
smooth graphon:
F norm of residual ~ d^2\over r^2 —> r~\sqrt{d}

sqrt(d)~spectral norm of residual <= F norm of residual

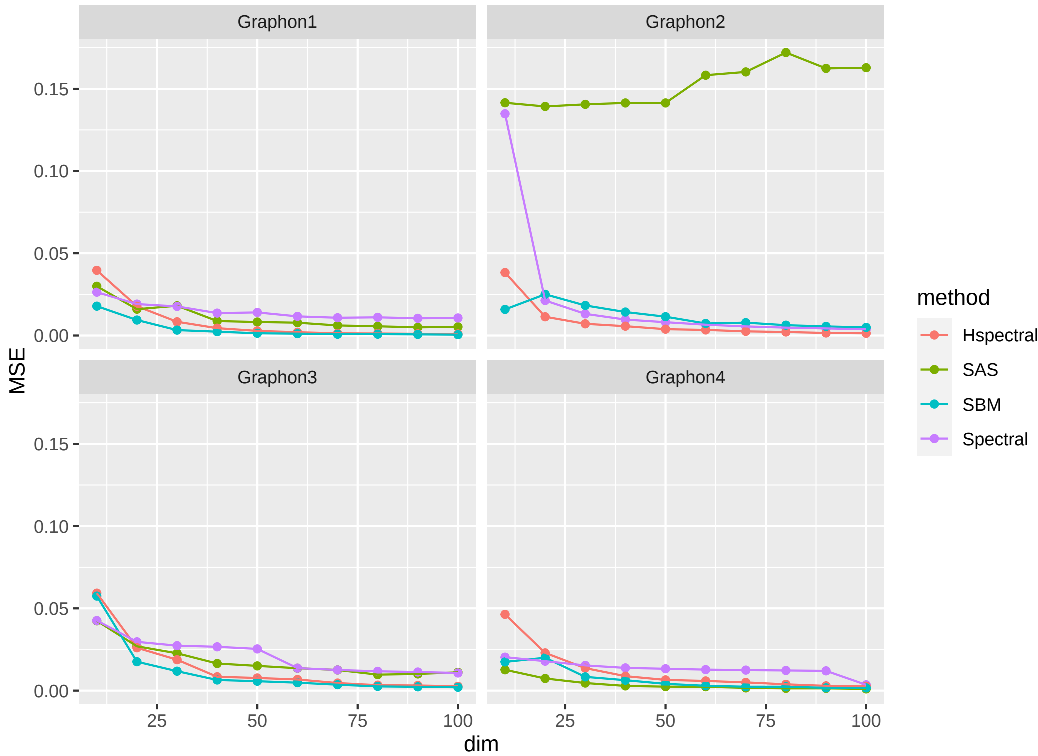


Figure 1: Mean squared error of estimated probability versus dimension. Stochastic block method is updated to take non-zero values at diagonal entries

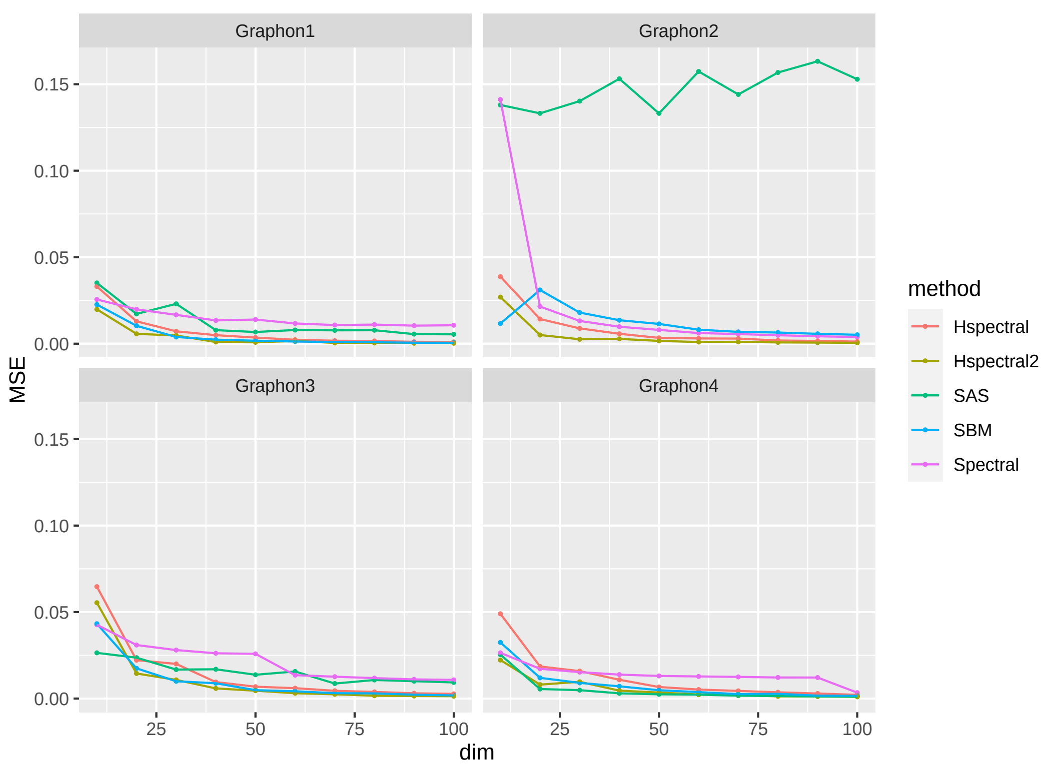
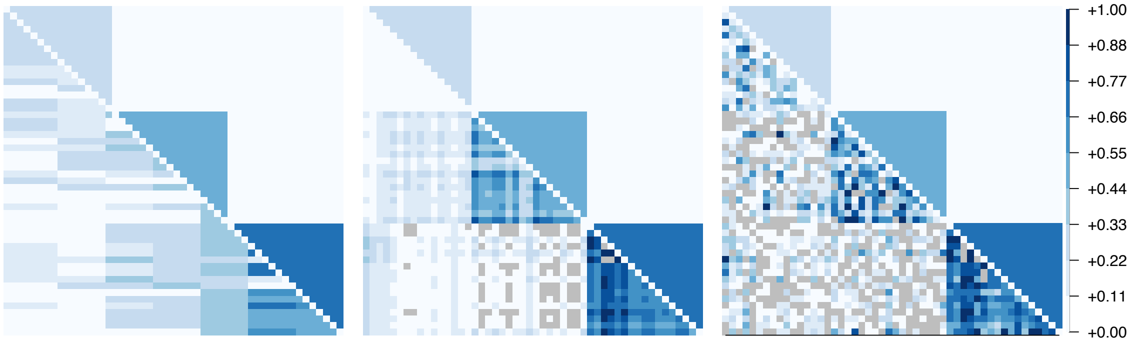
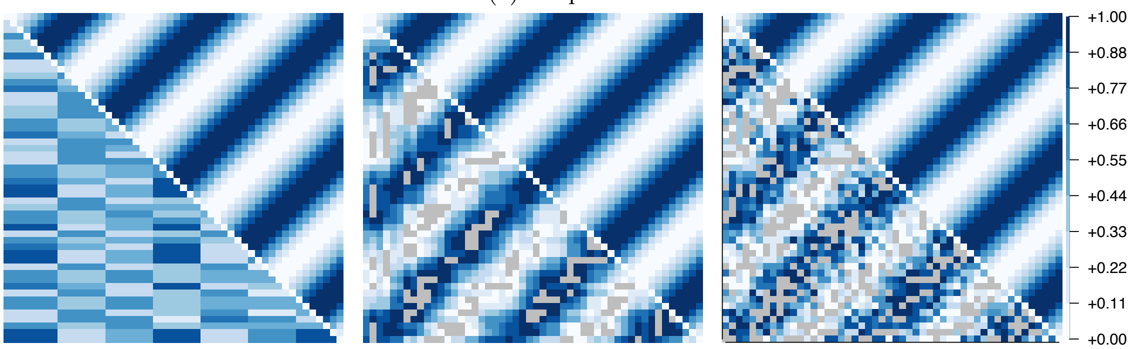


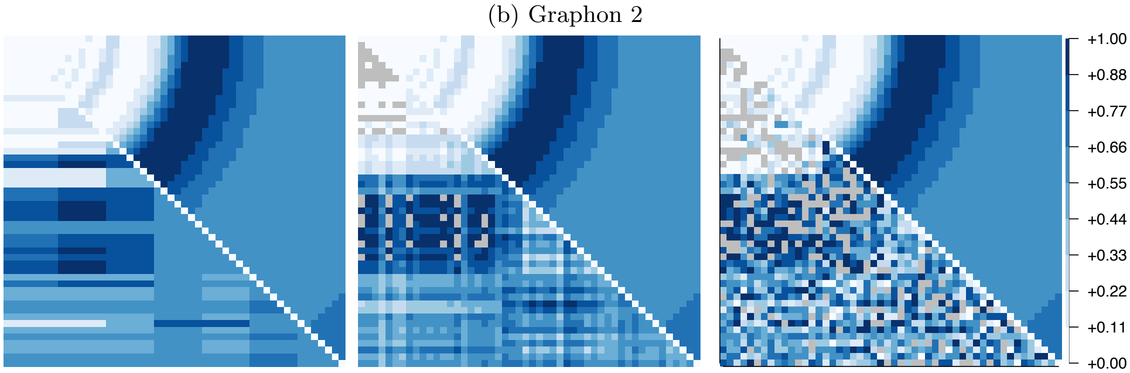
Figure 2: Mean squared error of estimated probability versus dimension. Hspectral2 method is added



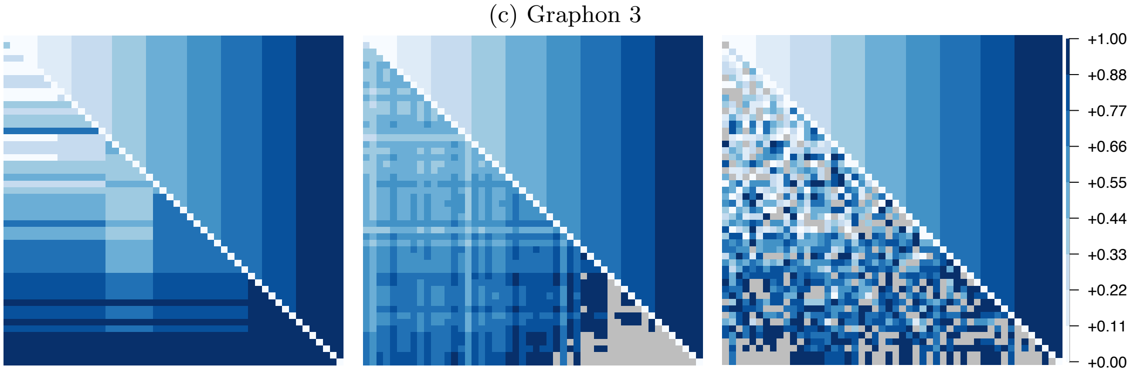
(a) Graphon 1



(b) Graphon 2

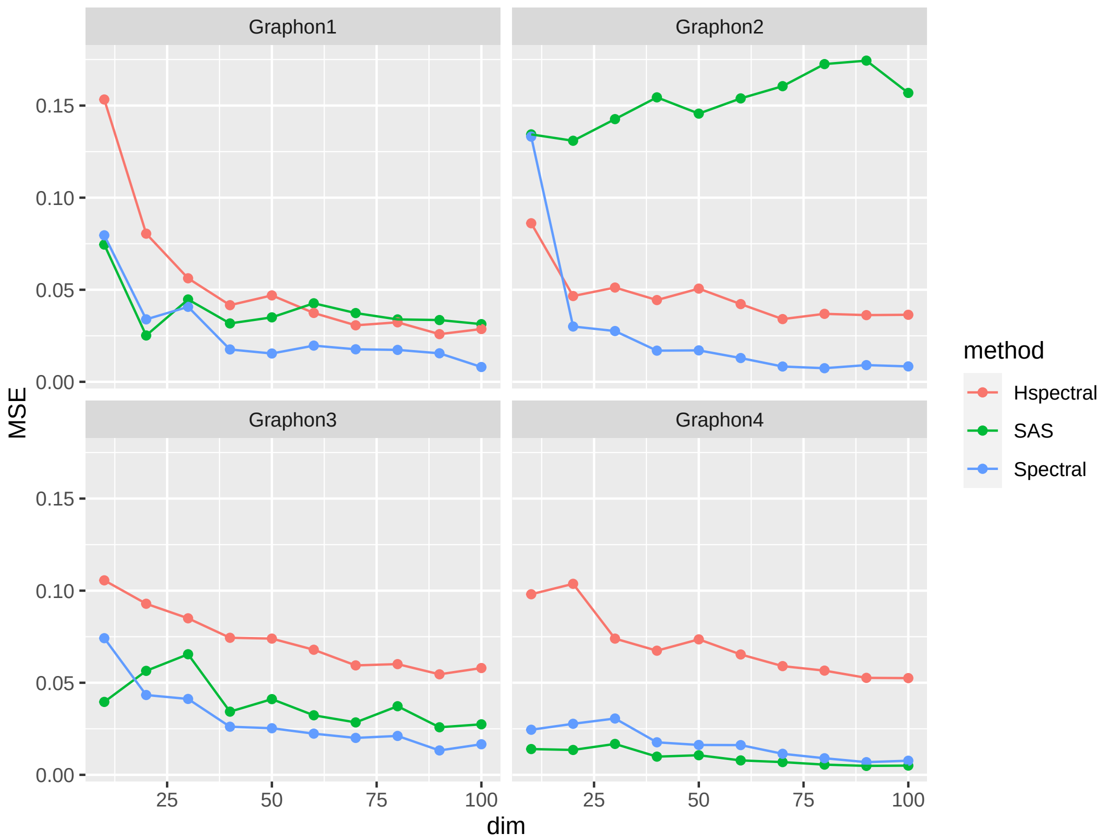


(c) Graphon 3

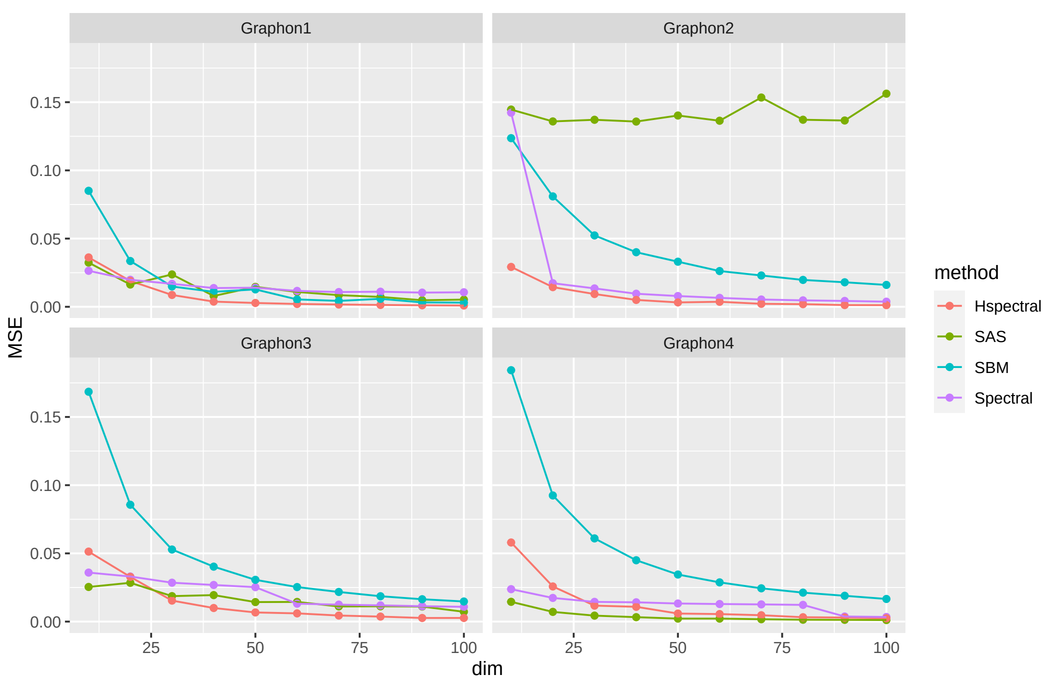


(d) Graphon 4

Figure 3: 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 $[0, 1]$.



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