Simulation results

Chanwoo Lee, August 25, 2021

1 Preliminary simulation setting

The first simulation considers the following generating models listed in Table 1

Model id	$f(x, y, z)$
1	xyz
2	$\lvert x-y\rvert + \lvert y-z\rvert + \lvert z-x\rvert$
3	$\frac{x^2+y^2+z^2}{\exp(\cos(1/(x^2+y^2+z^2)))}$
4	$\log(1+\max(x,y,z))$
5	$x^3+y^3+z^3-xy-yz-zx$

Table 1: List of generating models for testing.

2 Constant approximation vs polynomial approximation

In the first simulation, I compared the performance between constant approximation (degree = 0) versus polynomial approximation (degree¿0). It turns out that the number of group k is important factor for the performance. I set two different ways to set k

$$k_1 = d^{-1/3}, \quad \text{and} \quad k_2 = d^{-m/(m+2)}.$$

k_1 is the number of group set in `polytensor.R` while k_2 is the theoretical order that guarantees the rate in theorems. Since we do not know the smoothness of the unknown function, I set $\alpha = 1$ here. So I compare block-wise polynomial approximation from degree 0 to degree 3 with the block size k_1 and k_2 when the tensor dimension is $d \in \{50, 100\}$ across 5 different simulation setting in Table 1. I perform the same simulations 10 times for the stable comparison. Figure 1 shows the MSE versus degree of polynomial used in the approximation. It turns out that constant block approximation with k_2 block size performs the best in many scenerios while no big differences are observed regardless of polynomial degrees when the block size k_1 is fitted. Therefore, I decided to use either constant approximation with k_2 or degree 3 polynomial approximation with k_1 in the future simulations.

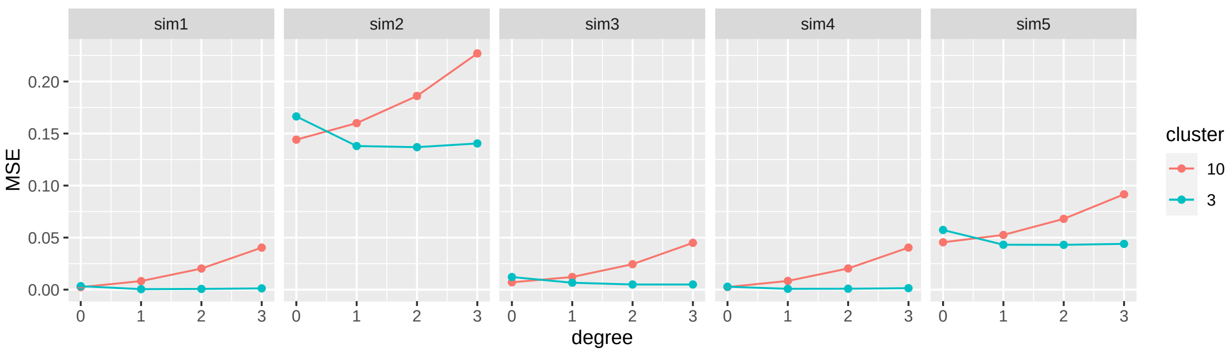
3 Preliminary comparison with alternative methods

In this section I compare the other alternative methods with ours. Candidates for alternative methods are as follow

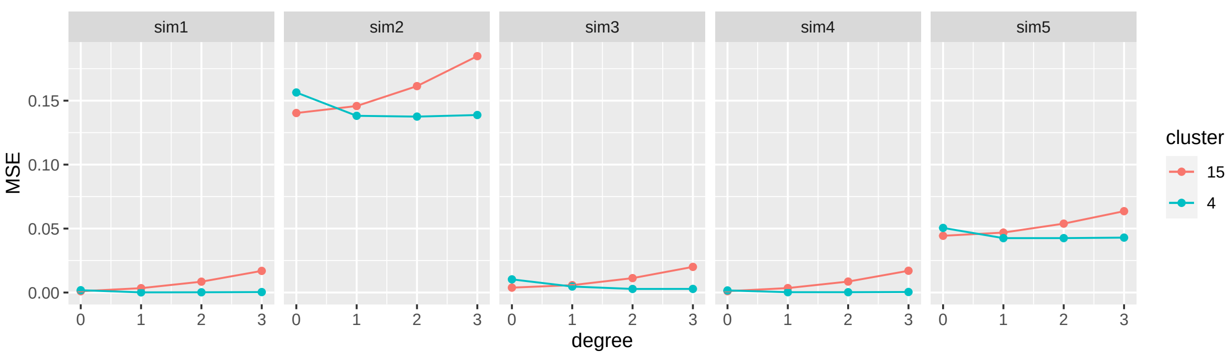
1. `LSE_k1`: Least square estimation method based on [1] with the number of block size k_1.
2. `LSE_k2`: Least square estimation method based on [1] with the number of block size k_2.
3. `Spectral`: Spectral method based on [2] with matricization.

Candidates for our method have

1. `Bordac_k1`: Borda count estimation with constant approximation and k_1 block size.



(a) When dimension = 50



(b) When dimension = 100

Figure 1: MSE versus degree of polynomial used in the approximation across 5 different models. Blue lines are fitted with the block-size k_1 while red lines with the block-size k_2.

2. `Bordac_k2`: Borda count estimation with constant approximation and k_2 block size.
3. `Bordap_k1`: Borda count estimation with polynomial degree 3 approximation and k_1 block size.
4. `Bordap_k2`: Borda count estimation with polynomial degree 3 approximation and k_2 block size.

We compare alternative methods in 5 different simulation setting with 10 repetition Figure 2 shows the simulatiuon results. In Model 2 and 5, our performance is worse than the spectral method. Possible explanation is that this two setting does not hold our main assumption: monotonic increasing property. Notice that Model 1 has $f(x,y,z) = |x-y| + |y-z| + |z-x|$ and Model 5 has $f(x,y,z) = x^3 + y^3 + z^3 - xy - yz - zx$. Therefore, I replace Model 2 and 5 by models that satisfy monotonicity in the main simulation.

In addition, among our methods, `Bordac_k2` and `Bordap_k1` have the almost same performance. So I choose the these two to use in main simulation.

4 Main simulation

In the main simulation, I use the modified simulation settings described in Table 2.

Model id	$f(x,y,z)$
1	xyz
2	$\frac{1}{3}(x+y+z)$
3	$\frac{x^2+y^2+z^2}{\exp(\cos(1/(x^2+y^2+z^2)))}$
4	$\log(1+\max(x,y,z))$
5	$\exp\left(-(\min(x,y,z)+\sqrt{x}+\sqrt{y}+\sqrt{z}\right)$

Table 2: List of generating models for testing.

I plot MSE versus tensor dimension $d \in \{10, \ldots, 100\}$ according to 5 different methods: Borda count estimation with constant approximation with k_2 (`Borda_c`), Borda count estimation with degree 3 polynomial approximation with k_1 (`Borda_p`), LSE with k_1 (`LSE_1`), LSE with k_2 (`LSE_1`), and Spectral method (`Spectral`). Figure 3 shows that the LSE method actually very unstable and its performance heavily depends on the model. I think I need to recheck the LSE algorithm [1] for the fair comparison.

In addition, we can see our method always outperforms other alternative methods while two our methods, (`Borda_c`) and a(`Borda_p`), show the similar performance.

References

[1] Krishnakumar Balasubramanian. Nonparametric modeling of higher-order interactions via hypergraphons. *arXiv preprint arXiv:2105.08678*, 2021.

[2] Jiaming Xu. Rates of convergence of spectral methods for graphon estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5433–5442. PMLR, 10–15 Jul 2018.

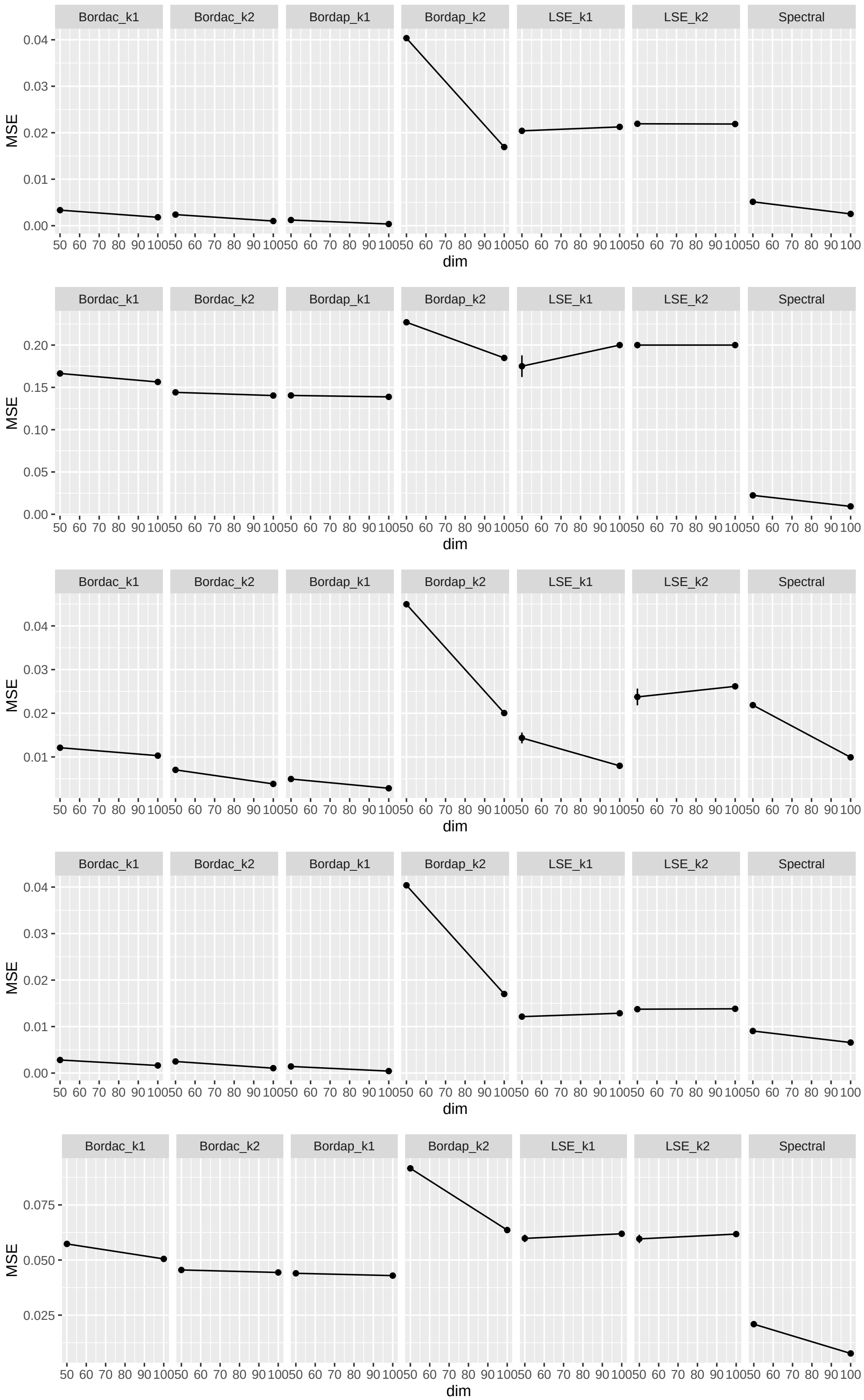


Figure 2: Preliminary simulation results: MSE versus tensor dimension $d \in \{50, 100\}$.

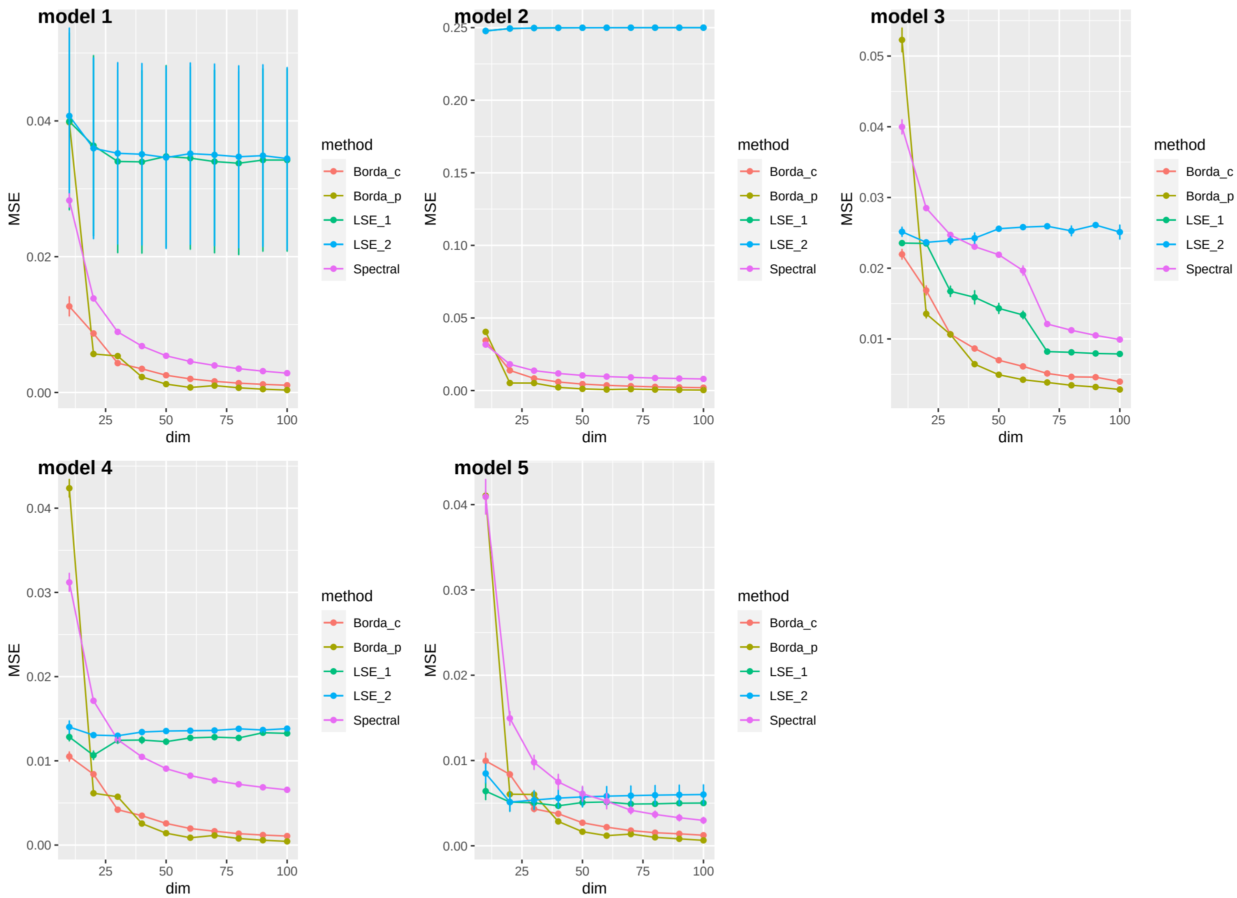


Figure 3: MSE versus tensor dimension $d \in \{10, \ldots, 100\}$ according to different methods