Referee Response for JCGS Submission JCGS-21-341

We thank the referees for their time in reviewing this paper and providing valuable comments and suggestions, which have all been taken into account for this revision. Most of the changes are in the simulation and application sections. In particular, we add a simulation study of “disassortative” PABMs and clarify the choices made in the data preprocessing.

The following sections are individual referee comments and our responses (highlighted in blue).

Referee 1 Comments

Main comments

The interest of this paper is to provide new perspectives on PABM (theoretical and algorithmic). The theoretical results are sound. However, the algorithmic results shown in the simulations are not really impressive since they are done in settings that do not appear very challenging. Indeed, they are done with fixed numbers of communities which are quite low (no more than 4) and only with assortative structure (larger probabilities of connections within a community than between two communities). Therefore, I think the paper needs a more challenging simulation study and the question of the choice of the number of communities needs to be addressed at least with some heuristic that works in simulation studies if theoretical results are not reachable. Moreover, some points on the exposition could be made clearer. I have specific comments below where I ask for some clarifications and detail how the simulation study could be improved in my opinion.

We thank the referee for the feedback. The concerns addressed here are answered below. In particular, we make sure to address the concerns about the simulation studies.

Specific comments and questions

1. p4. What is a hollow graph?
   We changed the terminology a bit here and throughout the paper and specify that the analysis assumes G is an unweighted and undirected graph with no self-loops (no edge directly from itself to itself), so A is binary and symmetric with zeros on the diagonal.

2. p4. G as a graph is non-oriented or undirected and A is symmetric.
   We have made minor edits clarifying the structure of G and A.

3. p4. Please define the set $[n]$.
   We added $[n] = \{1, 2, ..., n\}$.

4. p4. Definition 1. Do you consider a probability distribution on $z_1, ..., z_n$? Please give the range of variation of (i, j) when you define P_{ij}.
   The range of (i, j) is now included in the definition. In regard to the probability distribution on the labels $z_1, ..., z_n$, none of our theorems use this, although some do require that each n_k be sufficiently large, so it did not factor into our definitions and analyses. More specifically, the convergence rate of Theorem 5 depends on each n_k, and for convergence, we require that $n_k \to \infty$. While none of the theorems explicitly require this, in practice, we also require that each n_k is large enough for the latent vectors in the k^th community to span its subspace.

5. p4. line 50. Instead of “increasing values of the community assignments”, you could write “reorganized by community memberships”.
   This has been changed.

6. p5. Definition 2. I do not understand the definition of the subset $\mathcal{X}$. Is it $x \in \mathcal{X}$ if for all $y \in \mathbb{R}^d$ we have $x^\top I_{p,q} y$ or is $\mathcal{X}$ a subset on $(\mathbb{R}^d)^2$? Please clearly state that $d = p + q$ (not stated before Definition 3).
   We now define the subset as $\mathcal{X} = \{x, y \in \mathbb{R}^d : x^\top I_{p,q} y \in [0, 1]\}$ with $d = p + q$.

7. p6. Remark 2. What are $\hat{Z}$, V, and D?
   Remark 2 now includes that D is the diagonal matrix of the p most positive and q most negative eigenvalues of A (or P), V is the matrix of corresponding eigenvectors, and $\hat{Z}$ is the embedding $V|D|^{1/2}$.

8. top of p7. $\tilde{P}$ instead of P.
   We have corrected the typo.

9. p7 line 40-46. I don't understand the interest of this paragraph.
   The purpose of the additional material in the proof of Theorem 1 is to provide intuition for the arbitrary K case by starting with $K = 2$. In particular, for the $K = 2$ case, we show that $\tilde{P} = X \Pi X^\top$ where Π is a permutation matrix with 2 fixed points and 1 cycle of order 2, and the rows of X consist of vectors that lie on two 2-dimensional orthogonal subspaces. This then generalizes to larger K such that the rows of X are vectors that lie on K K-dimensional orthogonal subspaces, and Π is a permutation matrix with K fixed points and $K^2 - K = K(K-1)$ cycles of order 2. Then the eigenvalues of Π are broken up by the following: K 1's corresponding to the K fixed points and $K(K-1)$ pairs of $+1$ and -1 corresponding to the $K(K-1)$ cycles of order 2, resulting in $K(K+1)/2$ $+1$'s and $K(K-1)/2$ -1's, i.e., the diagonal eigenvalue matrix of Π is exactly $I_{K(K+1)/2, K(K-1)/2}$. This gives us the embedding dimension ($\mathbb{R}^{K^2}$ broken up into $K(K+1)/2$ "positive" and $K(K-1)/2$ "negative" dimensions according to the eigenvalues of Π) as well as the structure (K K-dimensional subspaces as described by the rows of X). We have added additional details to the paper to address this.

10. p8, line 14-21. I don't see why the permutation given by Π has K fixed points.
    The purpose of Example 1 is to illustrate this explicitly. Comparing matrices X and Y, we can see that they share three columns: the first, fifth, and ninth. The other columns are permuted. The proof of Theorem 1 then generalizes this to arbitrary K.

11. Section 3.1. I found it confusing that community detection algorithms are detailed when P is assumed to be known and then methods for estimating P are recalled. A sentence in the beginning of this section to expose what is presented could help the reader.
    The goal of estimation is not to estimate P but to identify the community labels and estimate the popularity parameters. This is not immediately obvious even if we know P beforehand, although having P does make estimation much easier. In the introduction to section 3 (before 3.1), we have added the following:
    "In our methods, the data that are observed for estimation is the adjacency matrix, $A \sim \text{PABM}(\{\lambda^{(k\ell)}\}_K, \rho_n)$, along with an assumed number of communities, K. To motivate our methods, we also consider community detection and parameter estimation in the case where we know the edge probability matrix, P, beforehand, noting that community memberships and popularity parameters are not immediately discernible from P itself. After establishing methods for community detection and parameter estimation from P, we use the consistency property of the ASE [4, 3] to demonstrate that the same methods work for A almost surely as $n \to \infty$."

12. p10, line 12. $c^{(i)}$ is not defined.
    $c^{(i)}$ is the i^{th} entry of column vector c. This has been added to the paper.

13. p10. I didn't understand the exposition of the SSC algorithm.
    We have added more detail to our description of the SSC algorithm. In particular, we describe the intuition/motivation behind the algorithm.
    The description of SSC now reads:
    "The SSC algorithm can be described as follows: Given $X \in \mathbb{R}^{n \times d}$ with vectors $x_i^\top \in \mathbb{R}^d$ as rows of X, the optimization problem $c_i = \arg\min_c \|c\|_1$ subject to $x_i = X^\top c$ and $c^{(i)} = 0$, where $c^{(i)}$ is the i^{th} entry of c, is solved for each $i \in [n]$. The solutions are collected into matrix $C = [c_1 \mid \cdots \mid c_n]^\top$

to construct an affinity matrix $B = |C| + |C^\top|$. If each x_i lies exactly on one of K subspaces, B describes an undirected graph consisting of *at least* K disjoint subgraphs, i.e., $B_{ij} = 0$ if x_i, x_j lie on different subspaces. The intuition here is that vectors that lie on the same subspace can be described as linear combinations of each other, assuming the number of vectors in the subspace is greater than the dimensionality of the subspace. Then once sparsity is enforced, for each c_i, its j^{th} element $c_i^{(j)}$ is zero if x_j belongs to a subspace that doesn't contain x_i, resulting in $B_{ij} = 0$. Thus, for each c_i, $c_i^{(j)}$ is zero if x_i and x_j belong to different subspaces and may be nonzero if they do. If X instead represents points near K subspaces with some noise, then this property may only hold approximately and a final graph partitioning step may be required (e.g., edge thresholding or spectral clustering)."

14. p11, line 31. "… SSC on the spectral embedding of A". Could you support this assertion by references?
We have added, "... SSC on the rows of the spectral embedding of A, since they will still lie approximately on the K subspaces [3]."

15. p12, line 13. It could help recall what the z_i, z_j are.
We have changed z_i to "community labels".

16. p14. Do you think that the OSC algorithm can retrieve other kinds of structure than assortative ones?
We added a third simulation study with $\lambda_{kk} \sim Beta(1, 2)$ and $\lambda_{kl} \sim Beta(2, 1)$, resulting in an average within-community edge probability of $1/3$ and between-community edge probability of $2/3$. Results of this simulation study are very similar to that of the first two and consistent with the theory.
On a related note, it is our view that the distinction between assortative and disassortative is not very important for the PABM, at least when it is viewed as a type of GRDPG. In the case of the SBM and DCBM, it can be shown that the rank of P (and therefore the number of embedding dimensions) is equal to K, but assortativity or disassortativity determines how many positive and negative eigenvalues P has [3]. In the case of the PABM, the number of positive eigenvalues is always $K(K + 1)/2$ and the number of negative eigenvalues is always $K(K - 1)/2$, (assuming each community spans its respective subspace).

17. Sections 4 and 5. Why do you use either the ARI criterion or the misclassification rate to assess the recovery of the nodes clustering?
The first to datasets in section 5 refer to datasets analyzed by other PABM papers. For the Leeds Butterfly dataset, while the paper with which we are comparing [2] provides enough details to transform and preprocess the data to match their analysis, they do not provide code to run their clustering algorithm. Therefore, we use their reported benchmark value, which is in the form of ARI. In order to make an apples-to-apples comparison, we report the ARI for OSC and SSC-ASE here as well. For all other analyses (simulation and real data), we use misclassification error (whether rate or count).

18. Do you consider sparsity in the simulated adjacency matrices?
Sparsity was not considered in our analyses and algorithms.

19. Why do you limit your simulation to assortative structure? Is it possible to see the results with a larger value of K? To what extent does the value of K impact the computational burden for the algorithms?
Regarding disassortativity, as mentioned in previous comments, we have added a third simulation study to address this. Performance of the algorithms tested is similar to the other two simulation studies and consistent with the theory. Regarding K, our algorithms require each n_k to be large enough to span its subspace of dimension K, so we focused mostly on smaller values of K. Cursory numerical experiments suggest that OSC and SSC-ASE behave similarly for larger K provided that n is large enough.
Regarding computational complexity, because both OSC and SSC-ASE may have an arbitrary final processing step, we will focus on the complexity in constructing the affinity matrix B. For OSC, this involves spectral decomposition followed by matrix multiplication of $n \times K^2$ and $K^2 \times n$ matrices, so the complexity is $O(n^3 + n^2 K^2)$. For SSC-ASE, we again start with spectral decomposition, followed by n LASSO problems with a design matrix of size $K^2 \times (n - 1)$, so the complexity is $O(n^3 + n^2 K^4 + n K^6)$ [1]. SSC-A involves solving n LASSO regression problems each with design matrices of size $n \times (n - 1)$, so the complexity here is $O(n^4)$. When $n \gg K^2$, we can see that OSC and SSC-ASE are both $O(n^3)$. In practice, we have found that K does not affect runtimes too much.

20. Again, in your application on the Leeds butterfly dataset, why did you limit your analysis to $K = 4$?
For the Leeds butterfly dataset, we wanted to compare results of OSC and SSC-ASE to an analysis performed by another group studying the PABM [2]. In their results, they describe how they removed some of the nodes and discretized the edges, which we replicate in our analysis.

21. Please specify in the text that the proofs for Theorems 3, 4, and 5 are given in the Appendix.
We have added this to the end of the introduction section.

Typos

1. p 10, line 16 "if each x_i lieS ..."
We have corrected this typo.

Referee 2 Comments

The first part of the paper shows that any PABM is a GRDPG and gives a constructive proof of the result. In particular, the proof gives the construction of the latent positions of the corresponding GRDPG.

The second part of the paper investigates the implications of this relation to the GRDPG (and the identified latent positions) on estimation algorithms for the PABM. New algorithms for community detection and parameter estimation are proposed. In particular, a new community detection method, named orthogonal spectral clustering (OSC), is proposed that directly exploits the result of Theorem 1. Furthermore, an improvement of the sparse subspace clustering (SSC) algorithm is proposed consisting of applying SSC to the adjacency spectral embedding of the network (ASE), referred to as SSC-ASE. For both algorithms it is shown that the subspace detection property holds and that the convergence rate of the detection error is faster than for classical SSC. Likewise, it is shown that parameter estimation can be improved. A numerical study illustrates the performance of the algorithms in comparison with existing methods. It shows that OSC performs the best, and also that SSC-ASE improves the results with respect to classical SSC in some scenarios. In the application on real datasets, OSC and SSC-ASE provide good results, but do not achieve the performance of modularity maximization (MM).

The topic of the paper is interesting as it concerns a recent graph model and the results are valuable. Contributions are two-fold: theoretic and computational. The paper is well written, and the presentation of the results and the proofs is very clear.

I only have some minor comments and a couple of questions:

- I have a general issue with latent position models like the GRDPG due to the nonidentifiability of the latent space. As stated in the paper, latent positions are not unique, and not even the dimension of the latent space is identifiable. Theorem 1 (or its proof) provide one of all possible latent positions. A general question is what is the consequence of this specific choice. Does the latent space (or its dimension) have an impact on the clustering results? Indeed, I lack intuition about what type of latent space would be optimal for community detection or estimation in the PABM. The space with the smallest possible dimension? Do the authors have an idea of the properties of the latent space that would give the best results for community detection? Related to this issue, is the latent space of the proof of Theorem 1 the smallest possible latent space to relate PABM to the GRDPG?
We thank the referee for the questions. While it is true that there are an infinite number of latent configurations that generate the PABM, based on Theorem 1 and GRDPG results by Rubin-Delanchy et al. [3], we can conclusively say the following:
1. The embedding dimension is K^2 with $K(K+1)/2$ positive and $K(K-1)/2$ negative eigenvalues,
2. The latent structure will always consist of K K-dimensional subspaces—in the configuration outlined by Theorem 1, these subspaces are orthogonal, but the multiplication by arbitrary $Q \in \mathbb{O}(p, q)$ that results in the same P may skew the subspaces making them no longer orthogonal (although they will still be K K-dimensional subspaces),
3. $B = nVV^\top$, where V is the $n \times K^2$ matrix of eigenvectors of P, will always be such that $B_{ij} = 0$ if and only if vertices i and j are in different communities.

4. The rows of V are such that the inner product of the i^{th} and j^{th} rows is zero if and only if vertices i and j are in different communities.

Therefore, the dimensionality of the latent configuration is not in question, only what kind of effect multiplication by unidentifiable Q has on the configuration. The OSC algorithm circumvents the nonidentifiability of the latent configuration by only considering the entries of $B = nVV^\top$. SSC-ASE does rely on the latent configuration, but we show that the orthogonality property holds for the particular embedding we perform for this algorithm.

That said, looking back at the latent configuration in Theorem 1, we can see that a K-dimensional (not K^2-dimensional) latent configuration is sufficient for clustering. If we are able to obtain this exact embedding, this would greatly reduce the number of dimensions required for the clustering In practice, this isn't always possible due to the nonidentifiability issue (in particular, determining which of the K out of K^2 eigenvectors should be used, although we have found via numerical experiments that community detection is often possible with the 2^{nd} to the $(K+1)^{st}$ embedding dimensions).

- This is clearly beyond the scope of the paper, but I wonder if there is a straightforward way to adapt the proposed methods to directed models or weighted graphs or any other types of graphs than binary undirected ones?

  For directed graphs, the ASE provides a consistent estimator and thus should also work here [4]. For weighted graphs, if A_{ij} can be an estimator for P_{ij} (e.g., if $A_{ij} \sim \text{Poisson}(P_{ij})$), then all of the theory should transfer over to those models as well, and we believe that the theory can be adapted to other models where $g(A_{ij})$ is an estimator for P_{ij}. We hope to address these topics in future research.

- In the simulation study, SSC-ASE works well for $K \geq 3$, but not for $K = 2$. The authors explain the bad performance by the use of gaussian mixtures in the last clustering step. However, I don't understand why this has an impact on the performance for $K = 2$, but not for $K \geq 3$.

  We are not entirely clear on why this happens for $K = 2$, but when comparing the normalized Laplacian eigenmaps of the affinity matrix B for various K, we see that when $K > 2$, the points appear much closer to Gaussian mixtures, whereas with $K = 2$, there is some separation by community, but they do not appear as Gaussian mixtures, at least for the simulated data. One possible reason for this may be because we failed to find the right hyperparameter for SSC-ASE.

  Under ideal circumstances, we would be able to circumvent this issue altogether via the subspace detection property, but SDP does not necessarily guarantee a partitioning of the graph into exactly K disjoint subgraphs but rather *at least* K disjoint subgraphs, and in our simulations, we found that setting ϑ to the appropriate value to obtain SDP often results in greater than K subgraphs, which is why we ended up choosing an "incorrect" ϑ for SSC and then performing the final clustering step.

  We also note that in our simulations, while SSC-ASE and SSC-A in the $K = 2$ case and MM-Louvain for all K output estimated labels such that the number of mislabeled vertices increases with n, the proportion of mislabeled vertices still decreases. For instance, in section 4.1 for, the error count for SSC-ASE for $K = 2$ is around 20 when $n = 128$ and around 100 when $n = 4096$, resulting in an error rate of around 16% for $n = 128$ and 2% for $n = 4096$.

- In the application section, MM performs better than OSC and SSC-ASE, while in the simulation study the opposite is observed. What is the reason for the different behavior?

  In the application section, we analyze data that are not necessarily generated by the PABM (or any known model). While all three algorithms are specific to the PABM, they might all behave differently to the different ways in which the model is misspecified. There is also the question of hyperparameter tuning, especially in the case of SSC-ASE, and we spent a bit more time tweaking the hyperparameters, resulting in performance that is a bit closer to that of MM.

- Typo on p. 10: Xc should be $X^\top c$. Likewise, $X_{-i}c$ is to be replaced with $X_{-i}^\top c$.

  We have corrected the typo.

References

[1] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32 (2):407 – 499, 2004. doi: 10.1214/009053604000000067.

[2] M. Noroozi, R. Rimal, and M. Pensky. Estimation and clustering in popularity adjusted block model. *Journal of the Royal Statistical Society, Series B.*, 2021+.

[3] P. Rubin-Delanchy, J. Cape, M. Tang, and C. E. Priebe. A statistical interpretation of spectral embedding: the generalised random dot product graph. arXiv: 1709.05506, 2017.

[4] D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107:1119–1128, 2012.