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 made considerable effort to address the referees' 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. The phrase "hollow graph" is no longer used.

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. In particular we assume throughout the paper that G is undirected and A is symmetric.

3. p4. Please define the set $[n]$.
   We added $[n] = \{1, 2, ..., n\}$ at the beginning of Section 2.1.

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 particular we wrote in Section 2.1 that "$1 \leq i < j \leq n$." In regard to the probability distribution on the labels $z_1, ..., z_n$, none of our theoretical assume anything about the distribution of the $\{z_i\}$; we do, however, occasionally assume that the n_k are sufficiently large for each $k = 1, 2, \ldots, K$. More specifically the convergence rate of Theorem 5 depends on the n_k, i.e., for convergence we require that $n_k \to \infty$ as $n \to \infty$. In practice, we also require that each n_k is large enough so that the latent vectors in the kth community span its subspace; this is a rather mild requirement, e.g., if the latent vectors in the kth community lies in a general position (are linearly independent), then we only require $n_k \geq K$.

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 denote $d = p + q$ in Definition 2 and define $\mathcal{X}$ as a subset of $\mathbb{R}^d$ such that for any $x \in \mathcal{X}$ and $y \in \mathcal{X}$, we have $x^\top I_{p,q} y \in [0, 1]$. One simple way to construct such a set $\mathcal{X}$ is to first find a collection of K vectors $\mathcal{S} = \{\nu_1, \nu_2, \ldots, \nu_K\} \subset \mathbb{R}^d$ where $\nu_i^\top I_{p,q} \nu_j \in [0, 1]$ for all i, j and then define $\mathcal{X}$ as

$$\mathcal{X} = \mathrm{conv}(\mathcal{S}) = \{x = \sum_i \lambda_i \nu_i \colon \lambda_i \geq 0 \text{ for all } i, \sum_i \lambda_i = 1\}.$$

7. p6. Remark 2. What are $\hat{Z}$, V, and D?
We had rewrote Remark 2 to more clearly described the matrices $\hat{Z}$, V and D. To avoid confusion, we now use $\hat{D}$ to denote the diagonal matrix whose diagonal entries are the $d = p + q$ largest eigenvalues (in modulus) of A and we use $\hat{V}$ to denote the $n \times d$ matrix whose columns are the corresponding eigenvectors. We then use $\hat{Z} = \hat{V}|\hat{D}|^{1/2}$ (where the $|\cdot|$ operation is applied elementwise) to denote the adjacency spectral embedding (ASE) of A into $\mathbb{R}^d$; that is to say, the ith row of $\hat{Z}$ represents an estimate of the latent position X_i (up to some non-identifiability transformation $Q \in \mathbb{O}(p, q)$). We also reorganized the paper slightly so that the definition and remarks about the non-identifiability of the latent positions X in a GRDPG are presented before discussing ASE.

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 first present the proof in the special case when $K = 2$ so as to build intuition for the general case of $K \geq 2$. More specificially for $K = 2$ 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 latent vectors lie in the union of two 2-dimensional orthogonal subspaces, i.e., the rows of X consist of vectors that lie in $\mathcal{S}_1 \cup \mathcal{S}_2$ where $\mathcal{S}_1$ and $\mathcal{S}_2$ are both 2-dimensional subspaces of $\mathbb{R}^4$ with $\mathcal{S}_1 \cap \mathcal{S}_2 = \{0\}$ and $x^\top y = 0$ for all $x \in \mathcal{S}_1, y \in \mathcal{S}_2$. This then generalizes to larger K such that the rows of X are now vectors that lie in the union of K orthogonal subspaces $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \ldots \mathcal{S}_K$ where each $\mathcal{S}_k$ is a K-dimensional subspace of $\mathbb{R}^{K^2}$, and Π is a permutation matrix with K fixed points and $K(K-1)$ cycles of order 2. The permutation matrix Π then has $K(K-1)/2$ eigenvalues equal to -1 (due to the $K(K-1)$ cycles of order 2) and $K(K+1)/2$ eigenvalues equal to 1 (due to the $K(K-1)$ cycles of order 2 together with the K fixed points). We can thus write $\Pi = U I_{K(K+1)/2, K(K-1)/2} U^\top$. We have revised the paper to more clearly present the above observations. See also Example 1 on page 9 of the revised manuscript and the response to comment 10 below.

10. p8, line 14-21. I don't see why the permutation given by Π has K fixed points.
Example 1 illustrates these fixed points explicitly for $K = 3$. More specifically, comparing the matrices X and Y, we see that they have three columns in common: the first, fifth, and ninth. The other columns are permuted. The proof of Theorem 1 then generalizes this to arbitrary K. In particular by the construction of X, each column of X has exactly one non-zero block corresponding to some vector $\lambda_{k\ell} \in \mathbb{R}^k$. The fixed points of Π are then the columns with indices $r(K+1)+1$ for $0 \leq r \leq K-1$; these columns contain the vectors λ_{kk} for $1 \leq k \leq K$. The cycles of order 2 swap the columns of X containing the vectors $\lambda_{k\ell}$ and $\lambda_{\ell k}$ for the $\binom{K}{2}$ pairs (k, ℓ) with $k \neq \ell$.

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.
We thank the referee for pointing out this issue. We had revised the manuscript to include the line "In our methods, the data that are observed is only the adjacency matrix $A \sim \mathrm{PABM}(\{\lambda^{(k\ell)}\}_K, \rho_n)$

along with an assumed number of communities, K. To motivate our methods, we first 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 ASE to demonstrate that the same methods work for A almost surely as $n \to \infty$." Indeed, the assumption of known P helps motivate Algorithm 1 through Algorithm 3 and the theoretical results in Theorem 2. Theorem 3 and Theorem 4 then follows by leveraging the consistency properties for adjacency spectral embedding to show that community recovery and parameters estimation using A is asymptotically equivalent to community recover and paraemters estimation using the true but unknown P.

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 details to our description of SSC to provide further 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 $C = \begin{bmatrix} c_1 \mid \cdots \mid c_n \end{bmatrix}^\top$ 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. Thus, for each c_i, $c_i^{(j)}$ is zero whenever x_i and x_j belong to different subspaces and may be nonzero otherwise. If X instead represents points near K subspaces with some noise, then this property will 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?
There are no references for this, as it is, to the best of our knowledge, a novel idea. We emphasize that while SSC has been proposed for community detection for PABM (see e.g., [4]), these work apply SSC directly on the rows of A (which are binary vectors in $\mathbb{R}^n$) rather than on the ASE of A (which are real-valued vectors in $\mathbb{R}^d$ where $d \ll n$). The motivation behind our use of SSC comes from Theorem 1 wherein we show that the latent vectors for PABM lie on the union of orthogonal subspaces in $\mathbb{R}^d$ and hence, by the consistency properties of ASE, the rows of the estimated eigenvectors $\hat{V}$ also lies close to the union of orthogonal subspaces in $\mathbb{R}^d$. In contrast, while the rows of P do lie exactly on the union of K orthogonal subspaces in $\mathbb{R}^n$, the rows of A (which are now noisy binary vectors) could be quite far from the corresponding rows of P and hence need not lie close to the union of any K orthogonal subspaces in $\mathbb{R}^n$

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?
The OSC algorithm works for any PABM parameters setting and is thus agnostic to the particular structure (such as assortativity vs disassortativity). We had revised the manuscript to include an additional simulation demonstrating this claim (see Section 4.3). In particular Section 4.3 considers a PABM setting wherein λ_{kk} is a vector in $\mathbb{R}^{n_k}$ whose elements are i.i.d Beta$(1, 2)$ random variables and $\lambda_{k\ell}$ is a vector in $\mathbb{R}^{n_k}$ whose elements are i.i.d Beta$(2, 1)$ random variables. This result in an average within-community edge probability of $1/3$ and a between-community edge probability of $2/3$. The results for this simultation setting are very similar to those presented in the original manuscript and is also consistent with the theoretical results in the paper.
On a related note, it is our view that the distinction between assortative and disassortative is not always meaningful for the PABM. Indeed, unlike the SBM, in the PABM each vertex is free to have a higher affinity to its own community or to other communities. For example, suppose vertex i belongs to community 1. Then the average probability of an edge between vertex i and another vertex in

community 1 is $n^{-1}\rho_n\lambda_{i1}\sum_j \lambda_{j1}$ while the average probability of an edge between vertex i and another vertex in community $k \neq 1$ is $n^{-1}\rho_n\lambda_{ik}\sum_j \lambda_{j1}$ and thus, on average, vertex i have more affinity with vertices in community $k \neq 1$ whenever $\lambda_{i1} < \lambda_{ik}$

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 two datasets that we analyzed in section 5 had also been analyzed in previous papers on the PABM. In particular the Leeds Butterfly dataset was analyzed in [4] and while [4] provides enough details for us to preprocess the data to match their analysis, they do not provide code to run their clustering algorithm (which is based on SSC applied to the rows of A). Therefore we used their reported benchmark values, which are given in terms of ARI. In order to make an apples-to-apples comparison, we also report the ARI for OSC and SSC-ASE here as well. For all other analyses (simulation and real data), we use the misclassification error rate/count.

18. Do you consider sparsity in the simulated adjacency matrices?
We have added another simulation in the supplemental materials. In this simulation, we fix $n = 2048$ and $K = 3$ and vary $\rho \in (0, 1)$.

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?
As we discussed in the response to comment 14, we have added a third simulation to address the issue of disassortativity (see Section 4.3 of the revised manuscript). The performance of all the algorithm for the setting in Section 4.3 is qualitatively similar to those presented earlier in Section 4.1 and Section 4.2 and is 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.
As for the computational complexity of the algorithms, the main bottleneck for OSC and SSC-ASE is the construction of the affinity matrix B. For OSC, this involves a (truncated) spectral decomposition to extract the K^2 largest eigenvalues in modulus (and their corresponding eigenvectors) followed by a matrix multiplication of $n \times K^2$ and $K^2 \times n$ matrices, so the time complexity is of the order $O(n^2K^2)$. For SSC-ASE (applying SSC on the rows of the adjacency spectral embedding $\hat{V}$) 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 $O(n^2K^2 + n^2K^4 + nK^6)$ [1]. SSC-A (applying SSC on the rows of A) as proposed in [4] 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 $O(n^2K^2)$. In practice the choice of parameter K does not affect runtimes too much. For the simulation studies we only considered $K \leq 4$ as values of $K \geq 5$ will in general, require larger values of n to achieve comparable estimation accuracy as $K \leq 4$.

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 our results using OSC and SSC-ASE to an earlier analysis using SSC on the rows of A (see [4]). In their analysis they describe how they removed some of the nodes (corresponding to the other butterfly species) so that the resulting dataset only include the $K = 4$ butterfly species as considered in our paper.

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. [5], 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 $\hat{B} = n\hat{V}\hat{V}^\top$ as $n\hat{V}\hat{V}^\top$ is, by the asymptotic properties of adjacency spectral embedding, close to $nVV^\top$ and $nVV^\top$ is always unique (the orthogonal projection onto a column space is always unique). Similarly, SSC-ASE also circumvents this non-identifiability by only using the rows of $n^{1/2}\hat{V}$ as two rows $\hat{v}_i$ and $\hat{v}_j$ of $\hat{V}$ belonging to different subspaces, i.e., different communities, will also have $n\hat{v}_i^\top \hat{v}_j \approx 0$.
  That said, looking back at the latent configuration in Theorem 1, we can see that a K-dimensional (not K^2-dimensional) embedding is sufficient for clustering, i.e., the embedding corresponding to the diagonal blocks $\lambda_{11}, \lambda_{22}, \ldots, \lambda_{KK}$. If we are able to find this embedding then this would greatly reduce the number of dimensions required for the community recovery/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 is the main difficulty. Nevertheless we have found via numerical experiments that community detection is often possible using only the eigenvectors corresponding to the 2^{nd} to the $(K+1)^{\text{st}}$ largest eigenvalues of A.

- 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 [7]. 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 $K = 2$, while there is some separation by community, they do not appear as Gaussian mixtures. One possible reason for this may be because we failed to find the right value for the hyperparameter ϑ for SSC-ASE.
  More specifically, under ideal circumstances, we would be able to circumvent the issue of selecting the hyperparameter ϑ by using the theoretical value that ensures, with high probability, the rows of $\hat{V}$ satisfy the subspace detection property (SDP). However, as other researchers had pointed out (see e.g., [3, 2], the fact that SDP is satisfied does not guarantee that the affinity matrix B represents a graph with K connected components but rather a graph with *at least* K connected component. Indeed, we found in our simulations that setting ϑ to the theoretical values that guarantees SDP often results in B having more than K connected components. This is why we choose to do spectral clustering on B, i.e., we first compute the normalized Laplacian embeddings using B before clustering the rows of this embedding using Gaussian mixtures modeling.
  Finally we note that in our simulations the proportion of mislabeled vertices does decreases as n increases. For instance, in section 4.1, if $K = 2$ then the average number of mis-clustered vertices for SSC-ASE is around 20 when $n = 128$ and around 100 when $n = 4096$; this correspond to an error rate of around 16% and 2%, respectively.

- 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 behave differently to the different ways in which the model is misspecified. Finally, we note that Section 5 analyzed three real datasets. For the first dataset (Leeds butterfly) we didn't run the MM algorithm as the original implementation of MM used in [6] is extremely slow when $K = 4$. For the second dataset (British MP's), the accuracy of MM is slightly better than that of OSC. Finally, for the last dataset (Karantaka villages), the accuracy of MM is slightly worse than that of OSC.

- 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] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:171–184, 2013.

[3] B. Nasihatkon and R. Hartley. Graph connectivity in sparse subspace clustering. In *Computer Vision and Pattern Recognition*, pages 2137–2144, 2011.

[4] M. Noroozi, R. Rimal, and M. Pensky. Estimation and clustering in popularity adjusted block model. *Journal of the Royal Statistical Society, Series B.*, 83:293–317, 2021.

[5] 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.

[6] S. Sengupta and Y. Chen. A block model for node popularity in networks with community structure. *Journal of the Royal Statistical Society, Series B.*, 80:365–386, 2018.

[7] 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.