Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models

Tiago P. Peixoto[*]
Institut für Theoretische Physik, Universität Bremen, Hochschulring 18, D-28359 Bremen, Germany

(Received 17 October 2013; published 13 January 2014)

We present an efficient algorithm for the inference of stochastic block models in large networks. The algorithm can be used as an optimized Markov chain Monte Carlo (MCMC) method, with a fast mixing time and a much reduced susceptibility to getting trapped in metastable states, or as a greedy agglomerative heuristic, with an almost linear $O(N \ln^2 N)$ complexity, where N is the number of nodes in the network, independent of the number of blocks being inferred. We show that the heuristic is capable of delivering results which are indistinguishable from the more exact and numerically expensive MCMC method in many artificial and empirical networks, despite being much faster. The method is entirely unbiased towards any specific mixing pattern, and in particular it does not favor assortative community structures.

DOI: 10.1103/PhysRevE.89.012804 PACS number(s): 89.75.Hc, 02.50.Tt, 89.70.Cf

I. INTRODUCTION

The use of generative models to infer modular structure in networks has been gaining increased attention in recent years [1–12], due to its more general character and because it allows the use of a more principled methodology when compared to more common methods, such as modularity maximization [13]. The most popular generative model being used for this purpose is the so-called stochastic block model [14–17], where the nodes in the network are divided into B blocks, and a $B \times B$ matrix specifies the probabilities of edges existing between nodes of each block. This simple model generalizes the notion of "community structure" [18] in that it accommodates not only assortative connections but also arbitrary mixing patterns, including, for example, bipartite, and core-periphery structures. In this context, the task of detecting modules in networks is converted into a process of statistical inference of the parameters of the generative model given the observed data [1–12], which allows one to make use of the robust framework of statistical analysis. Among the many advantages which this approach brings is the capacity of separating noise from structure, such that no spurious communities are found [19–25], increased resolution in the detection of very small blocks based on refined model selection methods [26], and the identification of fundamental limits in the detection of modular structure [27–31]. However, one existing drawback in the application of statistical inference is the lack of very efficient algorithms, in particular for networks with a very large number of blocks, with a performance comparable to some popular heuristics available for modularity-based methods [32,33]. Here we present some efficient techniques of performing statistical inference on large networks, which are partially inspired by the modularity-based heuristics but where special care is taken not to restrict the procedure to purely assortative block structures and to control the total number of blocks B such that detailed model selection criteria can be used. Furthermore, the method presented functions either as a greedy heuristic, with a fast $O(N \ln^2 N)$ algorithmic complexity, or as full-fledged Monte Carlo method, which saturates the detectability range of arbitrary modular structure at the expense of larger running times.

This paper is divided as follows. In Sec. II the stochastic block model is defined together with the maximum likelihood inference procedure. Section III presents an optimized Markov chain Monte Carlo (MCMC) method which is capable of reaching equilibrium configurations more efficiently than more unsophisticated approaches. In Sec. IV the MCMC techniques are complemented with an agglomerative heuristic which successfully avoids metastable states resulting from starting from random partitions and can be used on its own as an efficient and high-quality inference method. In this session we also compare the heuristic to the full MCMC method for synthetic networks. In Sec. V we compare both methods with several empirical networks. We finally conclude in Sec. VI with a discussion.

II. THE STOCHASTIC BLOCK MODEL

The stochastic block model ensemble [14–17] is composed of N nodes, divided into B blocks, with e_{rs} edges between nodes of blocks r and s (or, for convenience of notation, twice that number if $r = s$). For many empirical networks, much better results are obtained if degree variability is included inside each block, as in the so-called degree-corrected block model [8], in which one additionally specifies the degree sequence $\{k_i\}$ of the graph as an additional set of parameters.

The detection of modules consists in inferring the most likely model parameters which generated the observed network. One does this by finding the best partition $\{b_i\}$ of the nodes, where $b_i \in [1,B]$ is the block membership of node i, in the observed network G, which maximizes the posterior likelihood $\mathcal{P}(G|\{b_i\})$. Because each graph with the same edge counts e_{rs} are equally likely, the posterior likelihood is $\mathcal{P}(G|\{b_i\}) = 1/\Omega(\{e_{rs}\},\{n_r\})$, where e_{rs} and n_r are the edge and node counts associated with the block partition $\{b_i\}$, and $\Omega(\{e_{rs}\},\{n_r\})$ is the number of different network realizations. Hence, maximizing the likelihood is identical to minimizing the microcanonical entropy [34] $\mathcal{S}(\{e_{rs}\},\{n_r\}) = \ln \Omega(\{e_{rs}\},\{n_r\})$, which can be computed [35] as

$$\mathcal{S}_t = \frac{1}{2}\sum_{rs} n_r n_s H_b\left(\frac{e_{rs}}{n_r n_s}\right), \tag{1}$$

[*]tiago@itp.uni-bremen.de

for the traditional model and

$$\mathcal{S}_c \simeq -E - \sum_k N_k \ln k! - \frac{1}{2}\sum_{rs} e_{rs} \ln\left(\frac{e_{rs}}{e_r e_s}\right) \tag{2}$$

for the degree corrected variant, where $E = \sum_{rs} e_{rs}/2$ is the total number of edges, N_k is the total number of nodes with degree k, $e_r = \sum_s e_{rs}$ is the number of half-edges incident on block r, and $H_b(x) = -x\ln x - (1-x)\ln(1-x)$ is the binary entropy function and it was assumed that $n_r \gg 1$.

These models can be generalized for directed networks, for which corresponding expressions for the entropies are easily obtained [19,35]. The methods described in this paper are directly applicable for directed networks as well.

Although minimizing $\mathcal{S}_{t/c}$ allows one to find the most likely partition into B blocks, it cannot be used to find the best value of B itself. This is because the minimum of $\mathcal{S}_{t/c}$ is a strictly decreasing function of B, since larger models can always incorporate more details of the observed data, providing a better fit. Indeed, if one minimizes $\mathcal{S}_{t/c}$ over all B values one will always obtain the trivial $B = N$ partition where each node is in its own block, which is not a useful result. The task of identifying the best value of B in a principled fashion is known as model selection, which attempts to separate actual structure from noise and avoids overfitting. In the current context this can be done in a variety of ways, such as using the minimum description length (MDL) criterion [19,20] or performing Bayesian model selection (BMS) [7,21–25]. In Ref. [26] a high-resolution model selection method is presented, which is based on MDL and a hierarchy of nested stochastic block models describing the network topology at multiple scales and is capable of discriminating blocks with sizes significantly below the so-called "resolution limit" present in other model selection procedures and other community detection heuristics such as modularity optimization [36]. In Ref. [26] it is also shown that BMS and MDL deliver identical results if the same model constraints are imposed. However, in order to perform model selection, one first needs to find optimal partitions of the network for given values of B, which is the subproblem which we consider in detail in this work. Therefore, in the remainder of this paper we will assume that the value of B is a fixed parameter, unless otherwise stated, but the reader should be aware that this value itself can be determined at a later step via model selection, as described, e.g., in Refs. [19,26].

Given a value of B, directly obtaining the partition $\{b_i\}$ which minimizes $\mathcal{S}_{t/c}$ is, in general, not tractable, since it requires testing all possible partitions, which is only feasible for very small networks. Instead, one must rely on approximate or stochastic procedures which are guaranteed to sample partitions with a probability given as a function of $\mathcal{S}_{t/c}$, as described in the following section.

III. MARKOV CHAIN MONTE CARLO

The MCMC approach consists in modifying the block membership of each node in a random fashion and accepting or rejecting each move with a probability given as a function of the entropy difference $\Delta\mathcal{S}_{t/c}$. If the acceptance probabilities are chosen appropriately and the process is ergodic, i.e., all possible network partitions are accessible, and detailed balance is preserved, i.e., the moves are reversible, after a sufficiently long equilibration time, each observed partition must occur with the desired probability proportional to $\mathcal{P}(G|\{b_i\}) = e^{-\mathcal{S}_{t/c}}$. In this sense, this process is exact, since it is guaranteed to eventually produce the partitions with the desired probabilities, after a sufficient long equilibration (or mixing) time. In practice, the situation is more nuanced, since equilibration times may be very long, and one may not able to sample from a good approximation of the desired distribution, and different ways of implementing the Markov chain leads to different mixing times. The simplest approach one can take is to attempt to move each vertex into one of the B blocks with equal probability. This easily satisfies the requirements of ergodicity and detailed balance but can be very inefficient. This is particularly so in the case where the value of B is large and the block structure of the network is well defined, such that the vertex will belong to very few of the B blocks with a nonvanishing probability, which means that most random moves will simply be rejected. A better approach has been proposed in Ref. [19], which we present here in a slightly generalized fashion and consists in attempting to move a vertex from block r to s with a probability given by

$$p(r \to s|t) = \frac{e_{ts} + \epsilon}{e_t + \epsilon B}, \tag{3}$$

where t is the block label of a randomly chosen neighbor and $\epsilon > 0$ is a free parameter (note that by making $\epsilon \to \infty$ we recover the fully random moves described above). Equation (3) means that we attempt to guess the block membership of a given node by inspecting the block membership of its neighbors and by using the currently inferred model parameters to choose the most likely blocks to which the original node belongs (see Fig. 1). It should be observed that this move imposes no inherent bias; in particular, it does not attempt to find assortative structures in preference to any other, since it depends fully on the matrix e_{rs} currently inferred. For any choice of $\epsilon > 0$, this move proposal fulfills the ergodicity condition but not detailed balance. However, this can be enforced in the usual Metropolis-Hastings fashion [37,38] by accepting each move with a probability a given by

$$a = \min\left\{e^{-\beta\Delta\mathcal{S}_{t/c}}\frac{\sum_t p_t^i p(s \to r|t)}{\sum_t p_t^i p(r \to s|t)}, 1\right\}, \tag{4}$$

where p_t^i is the fraction of neighbors of node i which belong to block t, and $p(s \to r|t)$ is computed after the proposed $r \to s$

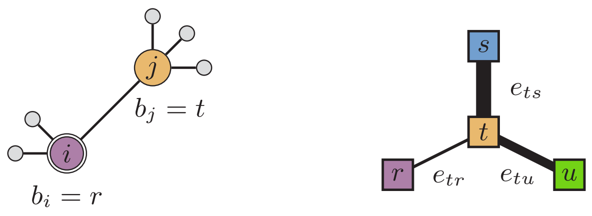

FIG. 1. (Color online) Left: Local neighborhood of node i belonging to block r, and a randomly chosen neighbor j belonging to block t. Right: Block multigraph, indicating the number of edges between blocks, represented as the edge thickness. In this example, the attempted move $b_i \to s$ is made with a larger probability than either $b_i \to u$ or $b_i \to r$ (no movement), since $e_{ts} > e_{tu}$ and $e_{ts} > e_{tr}$.

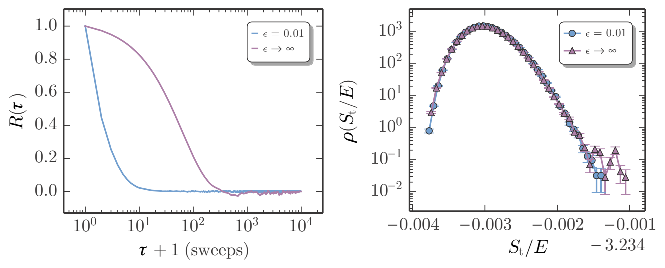

FIG. 2. (Color online) Left: Autocorrelation function $R(\tau)$, for a PP model with $c = 0.8$ and $B = 100$, for a network of size $N = 10^4$ and $\langle k\rangle = 10$, and two values of the parameter ϵ, where for $\epsilon \to \infty$ we have fully random moves. The curves were averaged for 100 independent network realizations. Right: PDF of the values of $\mathcal{S}_t/E$ obtained for $T = 2 \times 10^4$ consecutive sweeps for 100 independent network realizations, for different ϵ values, showing the same distribution.

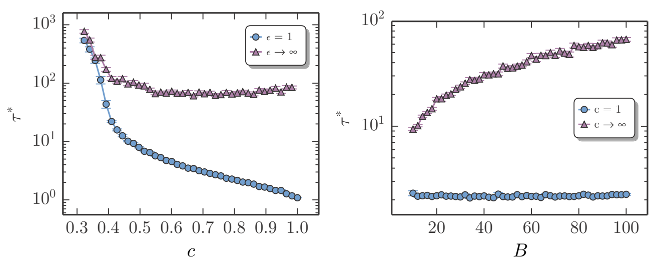

FIG. 3. (Color online) Left: Correlation time τ^* as a function of the model parameter c, for different values of ϵ, $N = 10^4$, $\langle k\rangle = 10$, $B = 100$, averaged over 40 independent network realizations. Right: Correlation time τ^* as a function of the number of blocks B, for different values of ϵ, for $N = 100 \times B$, $\langle k\rangle = 10$, $c = 0.8$, averaged over 40 independent network realizations.

move (i.e., with the new values of e_{rt}), whereas $p(r \to s|t)$ is computed before. The parameter β in Eq. (4) is an inverse temperature, which can be used to escape local minima or to turn the algorithm into a greedy heuristic, as discussed below.

The moves with probabilities given by Eq. (3) can be implemented efficiently. We simply write $p(r \to s|t) = (1 - R_t)e_{ts}/e_t + R_t/B$, with $R_t = \epsilon B/(e_t + \epsilon B)$. Hence, in order to sample s we proceed as follows: (1) A random neighbor j of the node i being moved is selected, and its block membership $t = b_j$ is obtained; (2) the value s is randomly selected from all B choices with equal probability; (3) with probability R_t it is accepted; (4) if it is rejected, a randomly chosen edge adjacent to block t is selected, and the block label s is taken from its opposite endpoint. This simple procedure selects the value of s with a probability given by $\sum_t p_t^i p(r \to s|t)$ and requires only a small number of operations, which is independent either on B or the number of neighbors the node i has. The only requirement is that we keep a list of edges which are adjacent to each block, which incurs an additional memory complexity of $O(E)$. To decide whether to accept the move, we need to compute the value of a, which can be done in $O(k_i)$ time, which is the same number of operations which is required to compute $\Delta \mathcal{S}_{t/c}$.[1] Therefore, an entire MCMC sweep of all nodes in the network requires $O(E)$ operations, independent of B.

To test the behavior of this approach, we examine a simple example known as the planted partition (PP) model [39]. It corresponds to an assortative block structure given by $e_{rs} = 2E[\delta_{rs}c/B + (1 - \delta_{rs})(1 - c)/B(B - 1)]$, $n_r = N/B$, and $c \in [0,1]$ is a free parameter which controls the assortativity strength. In this example, the algorithm above leads to much faster mixing times, as can be seen in the left part of Fig. 2, which shows the autocorrelation function

$$R(\tau) = \frac{\sum_{t=1}^{T-\tau}(\mathcal{S}_{t/c}(t) - \langle \mathcal{S}_{t/c}\rangle)(\mathcal{S}_{t/c}(t+\tau) - \langle \mathcal{S}_{t/c}\rangle)}{(T-\tau)\sigma^2_{\mathcal{S}_{t/c}}}, \quad (5)$$

where $\mathcal{S}_{t/c}(t)$ is the entropy value after t MCMC sweeps and T is the total number of sweeps, computed after a sufficiently long transient has been discarded. For the particular choice of parameters chosen for Fig. 2, the autocorrelation time is of the order of 10 sweeps with the optimized moves, and of the order of 100 sweeps with the fully random variant. Despite the difference in the mixing time, both methods sample from the same distribution, as shown in Fig. 2 (right).

The improvement for smaller ϵ values is more prominent as the block structure becomes more well defined, as can be seen in Fig. 3, which shows the autocorrelation time τ^*, defined here as

$$\tau^* = \sum_{\tau=0}^{T'} R(\tau), \quad (6)$$

where T' is the largest value of τ for which $R(\tau) \geqslant 0$. In Fig. 3 (left) are shown the values of τ^* depending on c, from which one can see that the relative improvement on the mixing time can be up to two orders of magnitude for the chosen value of $B = 100$. As the value of c approaches the detectability threshold (see below), the autocorrelation time diverges, as is typical of second-order phase transitions, and the relative advantage of the optimized moves diminishes. However, for most of the parameter range where the blocks are detectable, the mixing time with the optimized moves seems independent on the actual number of blocks, as shown in Fig. 3, where a fixed block size $N/B = 100$ was used and B was varied. One can see that for the optimized moves the mixing time remains constant, whereas for the fully random moves it increases steadily with B.

Although the optimized moves above provide a considerable improvement over the fully random alternative whenever the number of blocks B becomes large, there remains an important problem when applying it. Namely, the mixing time can be heavily dependent on how close one starts from the typical partitions which are obtained after equilibration. Since one does not know this, one often starts with a random partition. However, this is very far from the equilibrium states,

[1]For sparse networks with $e_{rs} \ll n_r n_s$, we may write $\mathcal{S}_t \cong E - \frac{1}{2}\sum_{rs} e_{rs}\ln e_{rs} + \sum_r e_r \ln n_r$, and note that to compute the change in entropy we need to modify at most $4k$ terms in the first sum and 2 terms in the second, if we change the membership of a node with degree k. The same argument holds for $\mathcal{S}_c$.

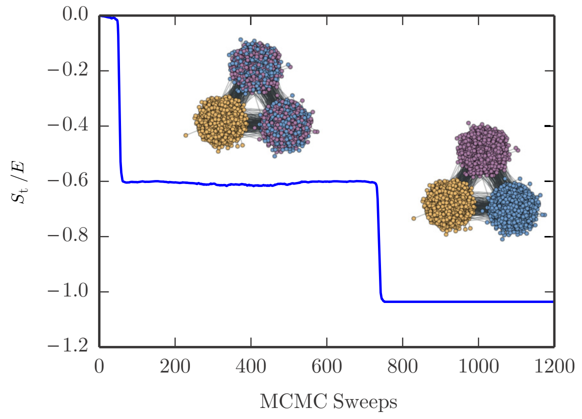

FIG. 4. (Color online) Evolution of the MCMC for a network sampled from the PP model with $N = 10^4$, $\langle k \rangle = 10$, $B = 3$, and $c = 0.99$, starting from a fully random partition of the nodes. The networks show a representative snapshot of the state of the system before and after the last drop in $\mathcal{S}_t$.

and if the block structure is sufficiently strong, this can lead to metastable configurations, where the block structure is only partially discovered, as shown in Fig. 4, for a network with $B = 3$.[2] The main problem is that not only does it take a long time to escape such metastable states, but also, by observing the values of $S_{t/c}$ alone, one may arrive at the wrong conclusion that the Markov chain has equilibrated. For example, in the simulation shown in Fig. 4, it took many hundreds of sweeps for the final drop in $\mathcal{S}_t$ to occur, and, before this, the time series is difficult to distinguish from an equilibrated chain. This problem is exacerbated if the average block size N/B increases, which can be frustrating since one would like to consider these scenarios to be easier than for smaller block sizes. In order to avoid this problem, we propose the agglomerative heuristic described in the next session, which can be used as a privileged starting point for the Markov chain or as an approximate inference tool on its own.

IV. AGGLOMERATIVE HEURISTIC

In order to avoid the metastable states described previously, we explore the fact that they are more likely to occur if the block sizes are large, since otherwise the quenched topological fluctuations present in the network will offer a smaller free-energy barrier which needs to be overcome. Therefore, a more promising approach is to attempt to find the best configuration for some $B' > B$ and then use that configuration to obtain a better estimate for one with B blocks.[3] This can be done by

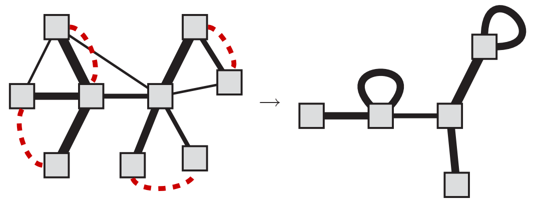

FIG. 5. (Color online) Representation of the block merges used in the agglomerative heuristic. Each square node is a block in the original graph, and the merges (represented as red dashed lines) correspond simply to block membership moves.

merging blocks together progressively, as shown in Fig. 5. We implement this by constructing a block (multi-)graph, where the blocks themselves are the nodes (weighted by the block sizes) and the edge counts e_{rs} are the edge multiplicities between each block node. In this representation, a block merge is simply a block membership move of a block node, where initially each node is in its own block. The choice of moves is done with same probability as before, i.e., via Eq. (3). In order to select the best merges, we attempt n_m moves for each block node and collectively rank the best moves for all nodes according to $\Delta S_{t/c}$. From this global ranking, we select the best $B' - B$ merges to obtain the desired partition into B blocks. However, if the value of N/B' itself is too large, we face again the same problem as before. Therefore we proceed iteratively by starting with $B_1 = N$, and selecting $B_{i+1} = B_i/\sigma$, until we reach the desired B value, where $\sigma > 1$ controls how greedily the merges are performed. To diminish the effect of bad merges done in the earlier steps, we also allow individual node moves between each merge step, by applying the MCMC steps above to the original network, with $\beta \to \infty$. The complexity of each agglomerative step is $O[n_m E + N \ln(B_i - B_{i-1}) + \tau E]$, which incorporates the search for the merge candidates, the ranking of the $B_i - B_{i-1}$ best merges, and the movement of the individual nodes, where τ is the necessary amount of sweeps to reach a local minimum. Since we have in total $\ln(N/B)/\ln\sigma$ merge steps, with the slowest one being the first with $B_1 = N$, we have an overall complexity of $O\{[(n_m + \tau)E + N \ln N] \times \ln N/\ln\sigma\} \sim O(N \ln^2 N)$, if we assume that $B \ll N$[4] and that the graph is sparse with $E \sim O(N)$.

Despite its greedy nature, we found that this approach is capable of almost always avoiding the metastable configurations described previously and often comes very close or even exactly to the planted partition (see Fig. 6).

The parameters n_m, σ, and ϵ allow one to choose an appropriate trade-off between quality and speed. The best results are obtained for large n_m and small σ; however, these need not to be chosen fully independently. We found that setting n_m to a "reasonable" value such as 10 or 100 and selecting σ to be 2, 1.1, or 1.01 allows one to probe the full quality range of the algorithm (see below). The choice of the value ϵ is interesting, since making $\epsilon = 0$ allows one to preserve certain graph invariants throughout the whole

[2]The occurrence of these metastable states is independent of the optimized moves and happens also for the fully random $\epsilon \to \infty$ moves.

[3]Note that we cannot simply set $B' > B$ and perform the same MCMC sweeps, expecting to obtain a partition into B blocks, since the values of $\mathcal{S}_{t/c}$ obtained for larger B values are always smaller. Differently from other community detection approaches, such as modularity optimization, here we are forced to control the value of B explicitly, which we can determine at a later step via a model selection procedure, as discussed previously.

[4]This is a worst-case scenario. If $B \sim N$, then the complexity reduces to $O(N \ln N)$.

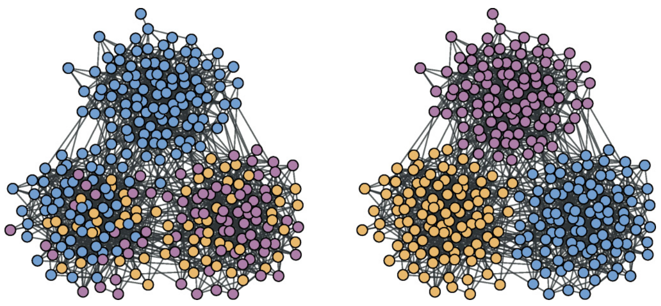

FIG. 6. (Color online) Left: An example of a typical partition obtained by starting with a random $B = 3$ configuration and applying only greedy moves until no further improvement is possible for a PP network with $N = 300$, $\langle k\rangle = 10$, and $c = 0.9$. Right: A typical outcome for the same network, with the greedy agglomerative algorithm described in the text.

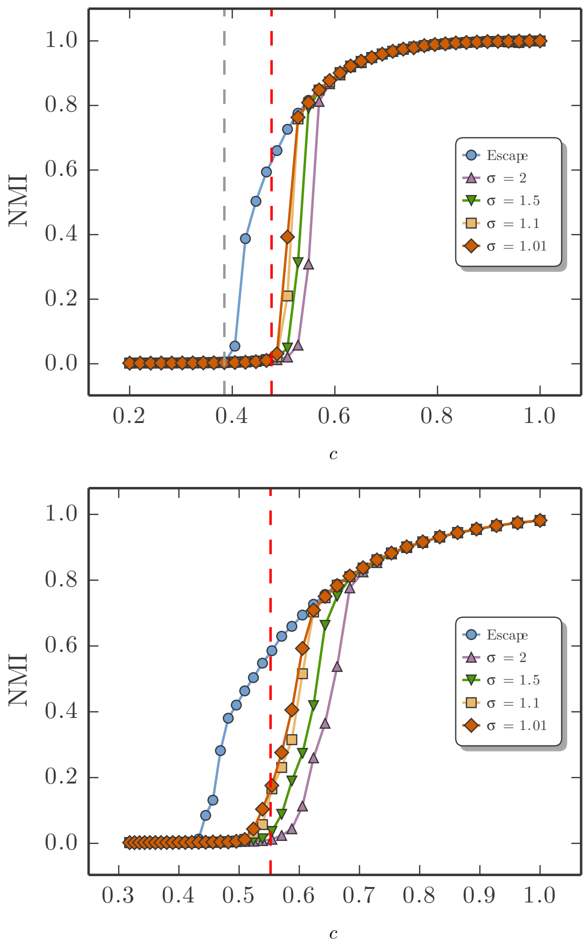

FIG. 7. (Color online) Normalized mutual information (NMI) (see footnote 6, page 6) between the planted and the inferred partitions for (top) the PP model and (bottom) the circular multipartite model described in the text, as a function of the modular strength c, for $N = 10^4$ and $B = 10$. The "Escape" curves correspond to MCMC equilibrations starting from the planted partition, and the remaining curves to the greedy agglomerative heuristic with ratio σ shown in the legend, and $n_m = 10$. All curves are averaged over 20 independent network realizations. The grey vertical dashed line corresponds to the detectability threshold c^* for the PP model, and the red dashed line to the MDL model selection threshold of Eq. (7).

procedure. Since at the first merging step when $B_i = N$ the e_{rs} matrix is simply the adjacency matrix, the membership moves with $\epsilon = 0$ cannot merge nodes which belong to different components or to different partitions in bipartite networks. It is easy to see that this property is preserved for later merging steps as well, so they are fully reflected in the final block structure. We find that very often this is a desired property and leads to better block partitions. In situations where it is not desired, it can be disabled by setting $\epsilon > 0$.

The algorithm above can be turned into a more robust MCMC method by making $\beta = 1$ in the intermediary phase between each merge step and waiting sufficiently long for the Markov chain to equilibrate. This is a slower, but more exact, counterpart to the greedy heuristic variant, which is less susceptible to getting trapped in the metastable states discussed previously. If one wishes to find the minimum of $\mathcal{S}_{t/c}$, one can make $\beta \to \infty$ after the chain has equilibrated, either abruptly (as we do in the results presented in this paper), or slowly via simulated annealing [40].

We can assess the quality of the heuristic method by comparing with known bounds on the detectability of the PP model. If we have that $N/B \gg 1$, it can be shown that for $\langle k\rangle < [(B-1)/(cB-1)]^2$ [27–29], it is not possible to detect the planted partition with any method. To emphasize the applicability of the method for dissortative (or arbitrary) topologies, we also analyze a circular multipartite block model, with $e_{rs} = 2E[(\delta_{r,s-1} + \delta_{r,s+1})c/2B + (1-c)/B^2]$, where c controls the strength of the modular structure, and periodic boundaries are assumed. In both cases we compare the agglomerative heuristic with MCMC results starting from the true partition, which represents the best possible case. As can be seen in Fig. 7, the results from the optimal MCMC and the heuristic are identical for up to some values of c which are larger than the actual detectability threshold. Thus the greedy method falls short of saturating the detectable parameter region but behaves badly only for a relatively small range of c, below which it becomes much harder (but not impossible) to distinguish the observed network from a random graph. To give a more precise idea of the extent to which the graphs in this region deviate from a random topology, we compare with a model selection threshold based on the minimum description length (MDL) principle [19],

$$\langle k\rangle > \frac{2\ln B}{I_{t/c}}, \qquad (7)$$

with $I_{t/c} = (S^r_{t/c} - S_{t/c})/E$, where $S^r_{t/c}$ is the entropy for a fully random graph, with $e_{rs} = 2En_rn_s/N^2$ (or $e_{rs} = e_re_s/2E$ for the degree-corrected case), and $E \gg B^2$ was assumed. This criterion is useful when we do not know the correct value of B and, hence, cannot rely on minimizing $S_{t/c}$ alone, since it would always result in a $B = N$ partition. If this condition is not fulfilled, the inferred partition (even if exact) is discarded in favor of a fully random graph, since the model parameters in this case cannot be used to provide a more compact description of the network. From Fig. 7 we see that this threshold lies very close to the region where the agglomerative algorithm is incapable of discovering the optimal partition. Hence, in

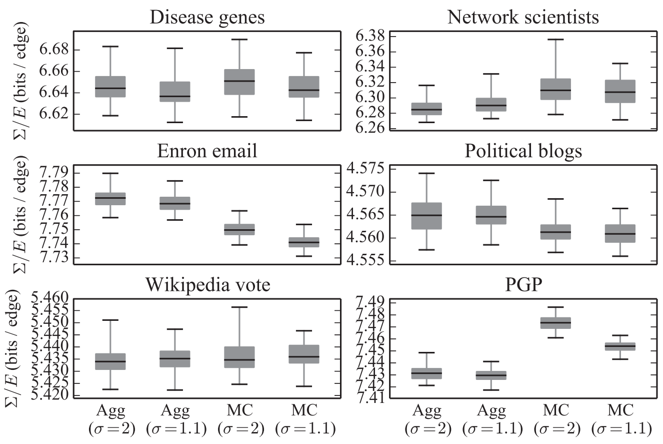

FIG. 8. Description length Σ for different empirical networks collected for 100 independent runs of the MCMC algorithm (MC) and the agglomerative heuristic (Agg) for different agglomeration ratios σ.

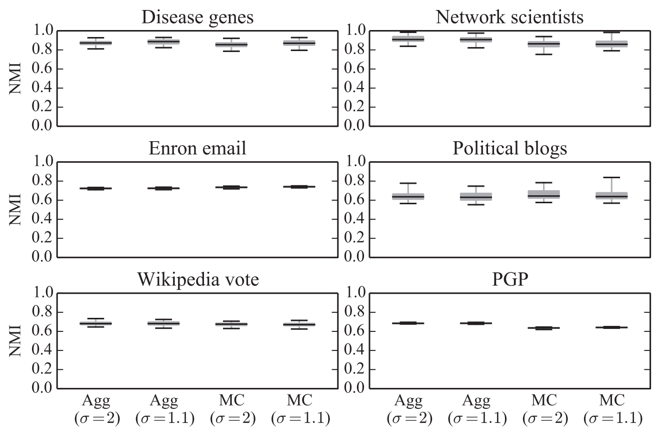

FIG. 9. Normalized mutual information (NMI) between the best overall partition and each one collected for 100 independent runs of the MCMC algorithm (MC) and the agglomerative heuristic (Agg) for different agglomeration ratios σ.

situations where model selection needs to be performed, any significant improvement to the quality of the algorithm would be ultimately discarded, at least in these specific examples. In other situations, where an increased precision close to the detectability transition is desired, the heuristic should be used only as a component of the full-fledged MCMC procedure with $\beta = 1$, as described above, which should be able to eventually reach the optimal configurations but requires longer running times.

V. PERFORMANCE ON EMPIRICAL NETWORKS

We have analyzed a few empirical networks to assess the behavior of the algorithm in realistic situations. We have chosen the following networks: The largest component of coauthorships in network science [41] ($N = 379$, $E = 914$, undirected), the human disease gene network [42] ($N = 903$, $E = 6760$, undirected), the political blog network [43] ($N = 1222$, $E = 19\,021$, directed), the Wikipedia vote network [44] ($N = 8298$, $E = 103\,689$, directed), the Enron email network [45,46] ($N = 36\,692, E = 367\,662$, undirected), the largest strong component of the PGP network [47] ($N = 39\,796$, $E = 301\,498$, directed), the IMDB film actor network [19] ($N = 372\,547$, $E = 1\,812\,312$, undirected), and the Berkeley-Stanford web graph [46] ($N = 654\,782$, $E = 7\,499\,425$, directed). In all cases we used the degree-corrected model. Since for these networks the most appropriate value of B is unknown, we performed model selection using the MDL criterion as described in Ref. [19], where we find the partition which minimizes the description length $\Sigma = \mathcal{L}_{t/c} + \mathcal{S}_{t/c}$, where $\mathcal{L}_{t/c}$ is the amount of information necessary to describe the model parameters, which increases with B.[5] For the networks with moderate size we were capable of comparing the results with the agglomerative heuristic to those of the more time-consuming MCMC method. Figures 8 and 10 show the values of Σ after several runs of each algorithm. It can be observed that the results obtained with both methods seem largely indistinguishable for some networks (disease genes, network scientists, and Wikipedia votes), whereas the MCMC algorithm leads to better results for others (Enron email, political blogs) and, interestingly, to worse results for the PGP network. The better results for MCMC are expected, but the worse result for the PGP network is not. We can explain this by pointing out that for that network the average value of $\mathcal{S}_c$ obtained with MCMC for $\beta = 1$ noticeably differs from the minimum possible value. Since we used an abrupt cooling to $\beta \to \infty$, the MCMC is more likely to get trapped in a local minimum than the agglomerative heuristic, which is never allowed to heat up to the $\beta = 1$ configurations. MCMC would probably match, or even improve, the heuristic results if, e.g., simulated annealing would be used to reach the $\beta \to \infty$ region. However, this serves as an example of at least one scenario where the agglomerative heuristic can lead to even better results, despite being much faster than MCMC.

Perhaps a more meaningful comparison among the different results is to determine how the obtained partitions differ from each other. This is shown in Figs. 9 and 10, where the normalized mutual information (NMI)[6] between the best partition across all runs of all algorithms and every other partition found is compared for the two algorithms. Despite leading to different Σ values, the typical partitions found for each algorithm seem equally far from the (approximated)

[5] As mentioned previously, a more refined MDL method presented in Ref. [26] computes $\mathcal{L}_{t/c}$ via a hierarchical sequence of stochastic block models, which provides better resolution at the expense of some additional complexity. But since our objective here is to compare methods of finding partitions, not model selection, we opt for the simpler criterion.

[6] The NMI is defined as $2I(\{b_i\},\{c_i\})/[H(\{b_i\}) + H(\{c_i\})]$, where $I(\{b_i\},\{c_i\}) = \sum_{rs} p_{bc}(r,s) \ln (p_{bc}(r,s)/p_b(r)p_c(s))$, and $H(\{x_i\}) = -\sum_r p_x(r) \ln p_x(r)$, where $\{b_i\}$ and $\{c_i\}$ are two partitions of the network.

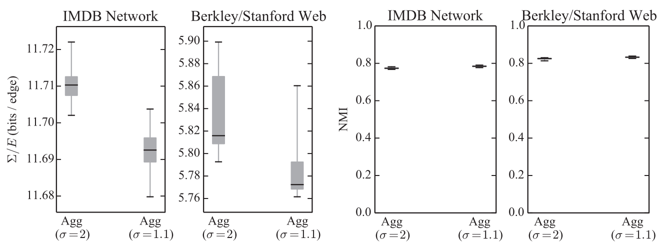

FIG. 10. Description length Σ for different empirical networks, as well the normalized mutual information (NMI) between the best overall partition and each one, collected for 100 independent runs of the agglomerative heuristic, for different agglomeration ratios σ.

global maximum, so the difference in Σ can be attributed to minor differences in the partitions. From this we can conclude that the agglomerative heuristic delivers results comparable to MCMC for many empirical networks, while being significantly faster.

Note that the NMI values in Fig. 9 are overall reasonably high, indicating that the partitions are much more similar than different; however, they are almost never 1, or very close to it, except for the smallest networks. This seems to point to a certain degree of degeneracy of optimal partitions, similar to those reported in Ref. [48] for methods based on modularity maximization. A more detailed analysis of this is needed, but we leave it to future work.

VI. CONCLUSION

We have presented an optimized MCMC method[7] for inferring stochastic block models in large networks, which possesses an improved mixing time due to optimized proposed node membership moves, and an agglomerative procedure which strongly reduces the likelihood of getting trapped in undesired metastable states. By increasing the inverse temperature to $\beta \to \infty$, this method is turned into an agglomerative heuristic, with a fast algorithmic complexity of $O(N \ln^2 N)$ in sparse networks. We have shown that although the heuristic does not fully saturate the detectability range of the MCMC method, it tends to find indistinguishable partitions for a very large range of parameters of the generative model, as well as for many empirical networks. The method also allows for detailed control of the number of blocks B being inferred, which makes it suitable to be used in conjunction with model selection techniques [19–26].

The heuristic method is comparable to the agglomerative algorithm of Clauset *et al.* [32] (and variants thereof, e.g., Refs. [49–51]), which has the same overall complexity but is restricted to finding purely assortative block structures, based on modularity optimization, and is strictly agglomerative, whereas the algorithm presented here permits individual node moves between the blocks at every stage, which allows for the correction of bad merges done in the earliest stages. It can also be compared to the popular method of Blondel *et al.* [33], which is not strictly agglomerative, but it is also restricted to assortative structures and is based on modularity, although it is typically faster than either the method of Clauset *et al.* and the method presented here.

Both the MCMC method and the greedy heuristic compare favorably to many statistical inference methods which depend on obtaining the full marginal probability π_r^i that node i belongs to block r [27,28,52]. Although this gives more detailed information on the network structure, it does so at the expense of much increased algorithmic complexity. For instance, the belief propagation approach of Refs. [27,28,52], although it possesses strong optimal properties, requires $O(NB^2)$ operations per update sweep, in addition to an $O(EB)$ memory complexity. Since in realistic situations the desired value of B is likely to scale with some power of N, this approach quickly becomes impractical and hinders its application to very large networks, in contrast to the log-linear complexity in N (independent of B) with the method proposed in this paper.

[7]An efficient C++ implementation of the algorithm described here is freely available as part of the graph-tool Python library at http://graph-tool.skewed.de.

[1] M. B. Hastings, Phys. Rev. E **74**, 035102 (2006).
[2] D. Garlaschelli and M. I. Loffredo, Phys. Rev. E **78**, 015101 (2008).
[3] M. E. J. Newman and E. A. Leicht, Proc. Natl. Acad. Sci. USA **104**, 9564 (2007).
[4] J. Reichardt and D. R. White, Eur. Phys. J. B **60**, 217 (2007).
[5] J. M. Hofman and C. H. Wiggins, Phys. Rev. Lett. **100**, 258701 (2008).
[6] P. J. Bickel and A. Chen, Proc. Natl Acad. Sci. USA **106**, 21068 (2009).
[7] R. Guimerà and M. Sales-Pardo, Proc. Natl. Acad. Sci. USA **106**, 22073 (2009).
[8] B. Karrer and M. E. J. Newman, Phys. Rev. E **83**, 016107 (2011).
[9] B. Ball, B. Karrer, and M. E. J. Newman, Phys. Rev. E **84**, 036103 (2011).
[10] J. Reichardt, R. Alamino, and D. Saad, PLoS ONE **6**, e21282 (2011).
[11] Y. Zhu, X. Yan, and C. Moore, arXiv:1205.7009.
[12] E. B. Baskerville, A. P. Dobson, T. Bedford, S. Allesina, T. M. Anderson, and M. Pascual, PLoS Comput. Biol. **7**, e1002321 (2011).
[13] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).
[14] P. W. Holland, K. B. Laskey, and S. Leinhardt, Soc. Networks **5**, 109 (1983).
[15] S. E. Fienberg, M. M. Meyer, and S. S. Wasserman, J. Am. Stat. Assoc. **80**, 51 (1985).
[16] K. Faust and S. Wasserman, Soc. Networks **14**, 5 (1992).
[17] C. J. Anderson, S. Wasserman, and K. Faust, Soc. Networks **14**, 137 (1992).

[18] S. Fortunato, Phys. Rep. **486**, 75 (2010).
[19] T. P. Peixoto, Phys. Rev. Lett. **110**, 148701 (2013).
[20] M. Rosvall and C. T. Bergstrom, Proc. Natl. Acad. Sci. USA **104**, 7327 (2007).
[21] J.-J. Daudin, F. Picard, and S. Robin, Stat. Comput. **18**, 173 (2008).
[22] M. Mariadassou, S. Robin, and C. Vacher, Ann. Appl. Stat. **4**, 715 (2010), mathematical Reviews number (MathSciNet): MR2758646.
[23] C. Moore, X. Yan, Y. Zhu, J.-B. Rouquier, and T. Lane, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11 (ACM, New York, 2011), pp. 841–849.
[24] P. Latouche, E. Birmele, and C. Ambroise, Stat. Model. **12**, 93 (2012).
[25] E. Côme and P. Latouche, arXiv:1303.2962.
[26] T. P. Peixoto, arXiv:1310.4377.
[27] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. Lett. **107**, 065701 (2011).
[28] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. E **84**, 066106 (2011).
[29] E. Mossel, J. Neeman, and A. Sly, arXiv:1202.1499.
[30] J. Reichardt and M. Leone, Phys. Rev. Lett. **101**, 078701 (2008).
[31] D. Hu, P. Ronhovde, and Z. Nussinov, Philos. Mag. **92**, 406 (2012).
[32] A. Clauset, M. E. J. Newman, and C. Moore, Phys. Rev. E **70**, 066111 (2004).
[33] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech.: Theor. Exp. (2008) P10008.
[34] G. Bianconi, Phys. Rev. E **79**, 036114 (2009).
[35] T. P. Peixoto, Phys. Rev. E **85**, 056122 (2012).
[36] S. Fortunato and M. Barthélemy, Proc. Natl. Acad. Sci. USA **104**, 36 (2007).
[37] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).
[38] W. K. Hastings, Biometrika **57**, 97 (1970).
[39] A. Condon and R. M. Karp, Random Struct. Algor. **18**, 116 (2001).
[40] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Science **220**, 671 (1983).
[41] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).
[42] K. I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A. L. Barabási, Proc. Natl. Acad. Sci. USA **104**, 8685 (2007).
[43] L. A. Adamic and N. Glance, in *Proceedings of the 3rd International Workshop on Link Discovery*, LinkKDD '05 (ACM, New York, 2005), p. 36.
[44] J. Leskovec, D. Huttenlocher, and J. Kleinberg, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10 (ACM, New York, 2010), p. 1361.
[45] B. Klimt and Y. Yang, in *First Conference on Email and Anti-Spam (CEAS), 2004 Proceedings, Mountain View, CA July 30 and 31, 2004*, http://ceas.cc/2004/168.pdf.
[46] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, arXiv:0810.1355.
[47] O. Richters and T. P. Peixoto, PLoS ONE **6**, e18384 (2011).
[48] B. H. Good, Y.-A. de Montjoye, and A. Clauset, Phys. Rev. E **81**, 046106 (2010).
[49] K. Wakita and T. Tsurumi, arXiv:cs/0702048.
[50] P. Schuetz and A. Caflisch, Phys. Rev. E **77**, 046112 (2008).
[51] P. Schuetz and A. Caflisch, Phys. Rev. E **78**, 026112 (2008).
[52] X. Yan, J. E. Jensen, F. Krzakala, C. Moore, C. R. Shalizi, L. Zdeborova, P. Zhang, and Y. Zhu, arXiv:1207.3994.