Cluster-size constrained network partitioning

Konstantin Avrachenkov, Maksim Mironov

Cluster-size constrained network partitioning

Konstantin Avrachenkov
INRIA Sophia Antipolis, France
k.avrachenkov@inria.fr

Maksim Mironov
Moscow Institute of Physics and Technology
Department of Discrete Mathematics, Russia
maksim.mironov@phystech.edu

***Abstract*—In this paper we consider a graph clustering problem with a given number of clusters and approximate desired sizes of the clusters. One possible motivation for such task could be the problem of databases or servers allocation within several given large computational clusters, where we want related objects to share the same cluster in order to minimize latency and transaction costs. This task differs from the original community detection problem. To solve this task, we adopt some ideas from Glauber Dynamics and Label Propagation Algorithm. At the same time we consider no additional information about node labels, so the task has the nature of unsupervised learning. We propose an algorithm for the problem, show that it works well for a large set of parameters of Stochastic Block Model (SBM) and theoretically show that its running time complexity for achieving almost exact recovery is of $O(n \cdot \bar{d} \cdot \omega)$ for the mean-field SBM with $\bar{d}$ being the average degree and ω tending to infinity arbitrary slow. Other significant advantage of the proposed approach is its local nature, which means it can be efficiently distributed with no scheduling or synchronization.**

I. Introduction

Community detection in networks is a very important topic which has numerous applications in social network analysis, computer science, telecommunications and bioinformatics, and has attracted the effort of many researchers. Let us just mention main classes of methods for network partitioning. The first very large class is based on spectral elements of the network matrices such as adjacency matrix and Laplacian (see e.g. the surveys [1, 2] and references therein). The second class of methods is based on the use of random walks (see e.g. [3, 4, 5, 6, 7] for the most representative works in this research direction). Finally, the third class of methods to network partitioning is based on the optimization of some objective such as likelihood [8, 9], modularity [10] or energy function [11, 12]. Interestingly, many mentioned approaches can also be viewed as particular instances of hedonic game [13]. The approach developed in this paper is also based on the energy function optimization.

We focus on the problem of graph partitioning given the desired number of clusters and their approximate sizes as an input. This problem statement can be useful for example when it is needed to split some objects in a predetermined number of clusters with respect to their relations or pack the graph into several given folds trying to minimize some global metric. At the same time we focus on algorithms which do updates based only on local (node neighbourhood) information. In this paper we consider partitioning into two folds, though the extension for the larger number of folds can be naturally derived by applying the "one-vs-rest" technique.

From formal side, we suppose that there is an unavailable ground truth: each node has its original cluster number, 1 or -1, which we want to determine. Similarly to [11], as a global objective function we use the energy borrowed from the Ising model. Let $n = n_1 + n_2$ be the total number of nodes with n_1, n_2 denoting the desired cluster sizes. Given the graph $G = (V, E)$, we introduce a set of configurations $\mathbf{\Sigma} \subset \{-1, 1\}^V$, such that for every configuration $\sigma \in \mathbf{\Sigma}$ every node $v \in V$ has label $\sigma(v)$ from $\{-1, 1\}$. Since we consider constraints on the clusters' sizes, each $\sigma \in \mathbf{\Sigma}$ consists of exactly n_1 labels 1 and of exactly n_2 labels -1. For convenience, we call vertices labeled with "1" as black and vertices labeled with "-1" as white. The global energy $\varepsilon(\sigma)$ of a configuration is then defined as follows:

$$\varepsilon(\sigma) = -\sum_{\{u,v\} \in E} \sigma(u)\sigma(v). \tag{1}$$

The global energy is then small when the number of edges with same colour nodes is large and the number of edges with different colour nodes is small. Alongside the global energy, we shall also use the local energy of a node v in configuration σ:

$$\varepsilon(\sigma, v) = -\sigma(v) \sum_{w \sim v} \sigma(w). \tag{2}$$

In the steps of our algorithms (to be described in detail a bit further) we shall do updates based on local energies and keep the numbers of black and white nodes as invariants, which is a major difference from Gibbs sampler and Glauber dynamics [11, 12]. The main motivation for doing this is to keep the current configuration cluster-size constrained and do not hit configurations colored in the unique color, keeping the algorithm of only local nature. We shall indicate how we can organize an efficient and distributed implementation of our algorithms.

II. SBM and mean-field SBM

Let us introduce the *Stochastic Block Model (SBM)* as the basic model for graphs with clustered structure. In that model we have a random graph $G_{sbm} = (V, E)$ with two blocks V_1, V_2, where $V = V_1 \sqcup V_2$ and for each pair of nodes $\{v, u\}$ an edge is drawn independently according to

$$\mathrm{P}\left(\{v, u\} \in E\right) = \begin{cases} p_1, & v, u \in V_1, \\ p_2, & v, u \in V_2, \\ q, & \text{overwise}, \end{cases}$$

with n_1, n_2, p_1, p_2, q possibly being functions of n. Let us suppose that the block sizes differ by a factor α, i.e.,

$$|V_1| = n_1,\ |V_2| = n_2,\ n_2/n_1 = \alpha \geq 1.$$

Cases when $\alpha = 1$ and $\alpha > 1$ are different, since for the first case there are two configurations giving us exact recovery (black-white and white-black configurations), whereas for the second case there is only one such configuration. This is a very important observation, which we shall discuss later in detail.

Let

$$\mathcal{G} = \{0,1\}^{\binom{n_1+n_2}{2}}$$

be a set of all possible graphs and let μ be a probability distribution over $\mathcal{G}$ derived from the SBM model.

The *mean-field SBM* is going to be used to obtain some theoretical insights, especially about the complexity of the proposed algorithms. In the mean-field SBM we have a complete deterministic graph $G_{mf} = (V_1 \sqcup V_2, E)$ and each edge has a weight according to its probability in the SBM graph, i.e., p_1, p_2 or q.

We consider only the case when the nodes have the same expected degree, namely, we assume that

$$p_1 + \alpha q = \alpha p_2 + q. \tag{3}$$

If the expected degrees differ for different blocks then the clustering task is trivial. Here we use exact and not asymptotic equality for simplicity of statements.

When $p_1 = p_2 = q$ then the SBM graph collapses to the Erdős-Rényi $G(n, p_1)$ graph where all edges are drawn independently with probability p_1 and no reconstruction is possible. Let us recall some facts about regimes in the SBM model [14].

- Erdős-Rényi $G(n,p)$ (as a block in SBM) is connected if and only if $p = c\ln(n)/n$ and $c > 1$.
- Exact recovery (i.e., right labeling of all nodes) in G_{sbm} for $\alpha = 1$ and $p_1 = p_2 = a(n)\ln(n)/n, q = b(n)\ln(n)/n$ is possible if $(a(n) + b(n))/2 > 1 + \sqrt{a(n)b(n)}$ and is not possible if $(a(n) + b(n))/2 < 1 + \sqrt{a(n)b(n)}$.
- Almost exact recovery (i.e., right labeling of $1 - o(1)$ share of nodes) in G_{sbm} with $p_1 = p_2 = a(n)/n, q = b(n)/n$ is possible if and only if

$$\frac{(a(n) - b(n))^2}{(a(n) + b(n))} \to \infty.$$

We split all configurations from $\mathbf{\Sigma}$ into groups $\Sigma^{(i)}$, such that $\sigma^{(i)} \in \Sigma^{(i)}$ denotes arbitrary configuration that has exactly i black nodes in the first block. Then, we consider the energy of configuration $\sigma^{(i)}$ as a function of i. Taking $j = n - i$ and averaging, we obtain:

$$\varepsilon^{(i)} := \mathrm{E}_\mu \varepsilon\left(\sigma^{(i)}\right) =$$

$$= (-1) \cdot ((n_1 - 2i)^2 \frac{p_1}{2} + (n_2 - 2j)^2 \frac{p_2}{2} +$$

Algorithm 1

Require: $n_1 \leqslant n_2$, G has $n = n_1 + n_2$ nodes
Require: relative error $0 < \delta < 1/2$
function ALGORITHM 1(n_1, n_2, G, δ)
  initialize σ with n_1 labels 1 chosen at random, other n_2 labels set with -1
  $\alpha \leftarrow n_2/n_1$
  $t \leftarrow 0$
  $T \leftarrow (1+\alpha)n/\delta$
  while $t < T$ **do**
    choose two random nodes v, u of different labels
    calculate sum s of local energies for v, u
    if $s > 0$ **then**
      $swap(\sigma[v], \sigma[u])$
    $t+ = 1$
  return σ

Fig. 1. One-round basic algorithm

$$+(n_1 - 2i)(n_2 - 2j)q + (n_1 p_1 + n_2 p_2))(1 + o(1)). \tag{4}$$

It is easy to see, that the expectation with respect to μ is exactly the corresponding weighted sum over all edges for the mean-field SBM.

We also need to define events, or subsets of configurations, which give us desired solution of the partitioning task. We introduce an event A_δ with δ denoting a relative error, which corresponds to configurations, where at least $(1-\delta)n_1$ nodes from V_1 are labeled with 1 (that would mean not more that δn_1 nodes with label 1 happen to be in V_2). Formally, we can write

$$A_\delta = \bigsqcup_{i \geqslant (1-\delta)n_1} \Sigma^{(i)}.$$

For $\alpha > 1$ the event A_δ corresponds to desired partitioning, though for $\alpha = 1$ we also need to define

$$B_\delta = A_\delta \sqcup A_{1-\delta},$$

which consists of both white-black and black-white optimal configurations and configurations close to them.

III. DESCRIPTION OF THE ALGORITHMS

Let us describe and discuss in detail two proposed algorithms. The first, basic algorithm is presented in Algorithm 1.

As one can see, Algorithm 1 strictly maintains the number of nodes of each color. The stopping rule here is defined by the number of steps derived in Proposition IV.3. Other possible natural stopping rules might depend on the number of steps without significant updates of the objective function.

One more approach for stopping rule works if we are aware of the graph structure like in SBM graphs with known parameters. For the SBM graph we know the expected global energy value and thus we can use the following stopping rule: if the current global energy obtained in an update of the algorithm is close to the expected global minimum in SBM, we stop the algorithm.

Algorithm 2

Require: $n_1 \leqslant n_2$, G has $n = n_1 + n_2$ nodes
Require: relative error $0 < \delta < 1/2$
function ALGORITHM 2(n_1, n_2, G, δ)
  $\sigma \leftarrow$ ALGORITHM 1(n_1, n_2, G, δ)
  $v \leftarrow 0$
  labels $\leftarrow \sigma$
  while $v < n$ **do**
    $s \leftarrow \sum_{u \sim v} labels[u]$ / $(labels[u] == 1?\ n_1 : n_2)$
    if $s \leqslant 0$ **then**
      $\sigma[v] = -1$
    else
      $\sigma[v] = 1$
    $v+ = 1$
  return σ

Fig. 2. Two-rounds algorithm

A significant advantage of the proposed algorithms consists in the nature of their local updates. The updates of the algorithms can be distributed over many machines with shared memory with no need for synchronization. Two updates of the algorithm are dependent if there is an edge between one of two nodes chosen at one update and one of two nodes chosen at the next update. More generally, if we have k machines making updates at the same time, then for independence there must not exist $2k \cdot (2k-2)/2 = 2k(k-1)$ such edges. If edge probability tends to zero and average degree is sub-linear, which is typically the case in real world graphs, then the probability that such $2k(k-1)$ edges do not exist is lower bounded by $(1-p)^{2k(k-1)}$ for $p = \max(p_1, p_2, q)$ and that expression tends to 1 given that k is a constant, i.e., with high probability these k machines make update independently for large graphs.

It is easy to notice, that in SBM setting Algorithm 1 is likely to make an effective update of the objective function (the objective can either decrease or stagnate), if there are many nodes with wrong labels. This reminds us about the coupon collector problem. We shall elaborate on this connection in the next section.

In order to deal with slow performance at the end of the algorithm run, we can apply the following heuristic: after running Algorithm 1 with any chosen stopping rule, we can iterate through all nodes and label each node independently based on the majority labeling of its neighbours. Thus, we come up with the second round of the algorithm, see Algorithm 2, given that the first round provides us some high quality but not perfect partitioning. We note that such two-rounds scheme is common practice to achieve exact recovery in graph clustering (see e.g., [15]).

The other possible heuristic inspired by the coupon collector setting might be the following. We can choose nodes to update not at random but based on their local energies: at each step we choose one black and one white nodes with the largest local energy and swap their labels. In order to do that we introduce into the algorithm Cartesian tree with the node number as a *key* and with the local energy as a *value*. After two nodes v, u are chosen and updated we need to update their local energies and the local energies of their neighbours, updating the Cartesian tree. This takes $O((\deg(v) + \deg(u)) \cdot \log(n))$. Therefore, the total asymptotic running time is only factor $\log(n)$ larger. However, the expected number of updates is supposed to be much smaller. We leave this modification as a subject of future research.

IV. Running time for mean-field SBM

First of all we need to note here the following. The mean-field SBM is clearly different from the authentic SBM. Nevertheless, studying the running time of the algorithm for the mean-field model gives intuition concerning the running time for the SBM and, as it will be shown in the next sections, the two models behave very similarly in numerical experiments.

Recall, that the intuition behind the use of the energy function is that it measures the quality of clustering: a lower energy means there is a smaller number of edges between nodes colored differently and more edges with the nodes sharing the same color.

Let us consider the energy as an objective of the discrete optimisation problem, where the steps of the optimisation procedure generate a sequence of configurations ending in a configuration with a local optimum of the energy. We first establish some properties of the expected energy.

Proposition IV.1. *Let $\alpha > 1$, let nodes in G_{sbm} share the same expected degree. If $(p_1+p_2)/2 > q$, then for the expected energy $\varepsilon^{(i)} = \mathrm{E}\varepsilon(\sigma^{(i)})$, as for a function of i, the following properties hold:*

1) *$\varepsilon^{(i)}$ with i from the discrete segment $[0, n_1]$ has a unique global minimum at $i = n_1$ (in that case all nodes are labeled correctly);*
2) *$\varepsilon^{(i)}$ with i from the continuous segment $[0, n_1]$ has exactly two local minima, associated with $i = 0$ and $i = n_1$;*
3) *finally, $\varepsilon^{(n_1)} - \varepsilon^{(0)} \geqslant 4(\alpha-1)n_1^2(p_2-q)(1+o(1))$.*

Proof. From (4) and $n_2 = \alpha n_1$, we can derive that

$$\varepsilon^{(i)} = \mathrm{E}\varepsilon(\sigma^{(i)}) = -(a \cdot i^2 + b \cdot i + c), \tag{5}$$

with $a = 4((p_1+p_2)/2 - q)$ and

$$b = 2(\alpha-2)n_1p_2 - 2n_1p_1 - 2(\alpha-2)n_1q + 2n_1q.$$

By the conditions of the proposition, we conclude that $a > 0$ and thus $\varepsilon^{(i)}$, as a continuous function of i, is a parabola with tails going down. So, on the segment $i \in [0, n_1]$ there might be either one or two local minima of $\varepsilon^{(i)}$ potentially given by the endpoints of the segment.

Again, using (4), we obtain

$$(-1) \cdot (\varepsilon^{(n_1)} - \varepsilon^{(0)}) = \frac{n_1^2}{2}(p_1 + \alpha^2 p_2 - 2\alpha q)(1+o(1)) -$$

$$-\frac{n_1^2}{2}(p_1 + (\alpha-2)^2 p_2 + 2(\alpha-2)q)(1+o(1)) \geqslant$$

$$\geqslant 4(\alpha-1)n_1^2(p_2-q)(1+o(1)).$$

Now in order to finish the proof we need to show that there cannot be a unique local minimum, i.e., we need to show that $-b/2a > 0$ (i.e., $-b > 0$). Suppose the opposite, $-b \leqslant 0$. Then,

$$-b \leqslant 0 \Leftrightarrow 2(\alpha-2)n_1p_2-2n_1p_1-2(\alpha-2)n_1q+2n_1q \geqslant 0 \Leftrightarrow$$
$$\Leftrightarrow (\alpha-2)p_2 - p_1 - (\alpha-2)q + q \geqslant 0 \Leftrightarrow$$
$$\Leftrightarrow (\alpha-2)p_2 - p_1 \geqslant (\alpha-3)q.$$

Recall, nodes share the same expected degree, which means that the equation (3) holds. Thus, we have

$$p_1 = \alpha p_2 - (\alpha-1)q$$

and

$$(\alpha-3)q \leqslant (\alpha-2)p_2 - (\alpha p_2 - (\alpha-1)q) \Leftrightarrow$$
$$\Leftrightarrow -2q \leqslant -2p_2 \Leftrightarrow q \geqslant p_2.$$

Using $q \geqslant p_2$ and equation (3) we have

$$p_1 \leqslant p_2$$

and so $(p_1 + p_2) \leqslant 2q$ which disagrees with the conditions of the proposition. Therefore, under the conditions of the proposition, there are always two local minima. □

Note that the statement about exactly two local minima has rather negative implications. Namely, there is no chance to have a parameter setting such that the application of our algorithm to the mean-field model will for sure produce convergence to the global optimum. There is always a chance that the algorithm converges to the non-desired local minimum. However, a good news is that, as it is stated in the proposition, the values of the global energy in two optima differ by a significant margin. It is also possible to show in some regimes, that the difference of $\Theta(n^2(p_2-q))$ is enough to distinguish optima in G_{sbm} with probability tending to 1 due to high concentration of the energy around its expected value.

Next, we consider the algorithm complexity in terms of its expected running time in different scenarios for the mean-field SBM.

Proposition IV.2. *Let $p_1+p_2 > 2q$. Then, for the graph G_{mf} the expected number of updates T for Algorithm 1 to hit a local minimum (any of them) is bounded by*

$$\mathrm{E}T \leqslant \frac{\pi^2}{12}n^2.$$

Proof. Recall that we have two local minima with $i = 0$ and $i = n_1$. Each step of the greedy algorithm does not increase the global energy and in some cases strictly decreases it by moving configuration $\sigma \in \Sigma^{(i)}$ from class $\Sigma^{(i)}$ to $\Sigma^{(i+1)}$ or $\Sigma^{(i-1)}$ depending on the tail of parabola parameter i happen to occur in.

The move is effective if two nodes $\{v_1, v_2\}$, chosen uniformly at random, are such that $v_1 \in V_1, v_2 \in V_2$ and $\sigma(v_1) \neq \sigma(v_2)$. And depending on the parabola tail all further steps will move parameter i in the same direction (in G_{sbm} graph that is not true due to local topology). Without loss of generality, we consider the case moving towards $i = n_1$, namely, $i > -b/2a$ with a, b specified in (5). Then, in order to do effective step, we need two nodes of different labels chosen at random at the current step to be in different blocks and both of them should be labeled wrongly.

This is now equivalent to the well-studied coupon collector problem. Let $T(\sigma)$ be a random variable denoting the number of steps needed to achieve $i = n_1$. That is, we need to hit all wrongly labeled nodes starting the algorithm from an arbitrary configuration σ. Then,

$$T(\sigma) \leqslant \sum_{i=\lfloor -b/2a \rfloor}^{n_1-1} T_i \leqslant \sum_{i=0}^{n_1-1} T_i, \tag{6}$$

where T_i is the number of steps needed to hit two wrongly labeled nodes from different blocks and to swap their labels (regardless the values of the local energies). From the definition of $\Sigma^{(i)}$, the probability of choosing the needed $\{v_1, v_2\}$ is equal to

$$2 \cdot \frac{n_1 - i}{n_1 + n_2} \cdot \frac{n_1 - i}{n_1 + n_2} = \frac{2}{(1+\alpha)^2} \frac{(n_1-i)^2}{n_1^2}.$$

T_i being a geometrical random variable, yields

$$\mathrm{E}T_i = \frac{(1+\alpha)^2 n_1^2}{2} \cdot \frac{1}{(n_1-i)^2}.$$

Then, we can write

$$\mathrm{E}T(\sigma) \leqslant \sum_{i=0}^{n_1-1} \mathrm{E}T_i = \frac{(1+\alpha)^2 n_1^2}{2} \sum_{i=0}^{n_1-1} \frac{1}{(n_1-i)^2} =$$
$$= \frac{(1+\alpha)^2 n_1^2}{2} \sum_{i=1}^{n_1} \frac{1}{i^2} \leqslant \frac{(1+\alpha)^2 n_1^2 \pi^2}{12} = \frac{n^2\pi^2}{12}.$$

In the case of moving towards $i = 0$ we can consider $j = n_1 - i$ and collect j white labels in first block instead of i black ones. Then, the calculations using this change of variables will be just the same. □

As it was mentioned before, the algorithm reduces to the coupon collector problem, so the probability to make a right update, when the current error is small, is small itself. However, we can apply a so-called $80/20$ Pareto rule and do not all, but almost all, needed work in a much smaller number of steps. We formally state this in the next proposition.

Proposition IV.3. *Let $0 < \delta = \delta(n) < 1/2$ and the conditions of Proposition IV.1 hold. Then, the expected number of steps T needed to hit the B_δ is such that*

$$\mathrm{E}T < \frac{(1+\alpha)}{\delta}n.$$

Proof. First, let us consider the case of moving towards $i = n_1$ and estimate the expected number of steps to hit $n_1(1-\delta)$ black labels in the first block. Using the same notations as before, we can write

$$\sum_{i=0}^{n_1(1-\delta)} \mathrm{E}T_i = \frac{(1+\alpha)^2 n_1^2}{2} \sum_{i=0}^{n_1(1-\delta)} \frac{1}{(n_1-i)^2} <$$

$$< \frac{(1+\alpha)^2 n_1^2}{2} \sum_{i=n_1\delta}^{\infty} \frac{1}{i^2} < \frac{(1+\alpha)^2 n_1^2}{2} \int_{n_1\delta-1}^{\infty} \frac{dx}{x^2} =$$

$$= \frac{(1+\alpha)^2 n_1^2}{2n_1\delta - 2} < \frac{(1+\alpha)^2 n_1^2}{2n_1\delta/2} = \frac{(1+\alpha)n}{\delta}.$$

The steps to converge to the other optimum $i = 0$ can be upper bounded as before with the change of variable: $j = n_1 - i$. □

When blocks are of different sizes ($\alpha > 1$), Algorithm 1 can converge to any of the local optima depending on the tail of the parabola where the initial configuration happens to be initiated. However, these optima can be distinguished from each other. Thus, in the case of convergence to the wrong optimum, we need to run the algorithm once again starting from another initial configuration. The probability that an initial configuration leads to the wrong optimum depends on the number of black labels in the first block. Easy to notice, that $-b/2a$ matches the expectation of the number of black nodes in the first block if n_1 black nodes were distributed over $n = n_1 + n_2$ nodes uniformly at random. In other words, let each of n_1 black nodes in the initial configuration fall into the first block with probability $n_1/(n_1+n_2)$ independently. Given that the initial number of black labels in the first block is a binomial random variable and that the mean and median for the binomial random variable are equal with respect to integer part, the desired probability to occur within range $[-b/2a, n_1]$ is almost $1/2$. That means that with high probability we need constant (around 2) times to re-run the algorithm from scratch in order to converge to a vicinity of the desired minimum ($i = n_1$) and to hit the event A_δ.

When blocks are of the same size ($\alpha = 1$), $\varepsilon^{(i)}$, as a function of i, is symmetric with the center of symmetry $i = n_1/2$. Since both $i \sim 0$ and $i \sim n_1$ we consider as good solutions and, obviously, both of them and only them provide local and global minima of energy with the same value, we can converge to any of them. Therefore, the event B_δ, introduced earlier, is the desired solution. The above can be summarized formally as follows:

Proposition IV.4. *Let $\alpha = 1, p_1 = p_2$ and $p_1 > q$. Then, the expected number of steps T to obtain the exact recovery by Algorithm 1 in the mean-field SBM is upper bounded by*

$$\mathrm{E}T \leqslant \frac{\pi^2}{12} n^2.$$

Proposition IV.5. *Let $\alpha = 1, p_1 = p_2, p_1 > q$ and $\delta = o(1)$. Then, the expected number of steps T to obtain the almost exact recovery in the mean-field SBM is upper bounded by*

$$\mathrm{E}T \leqslant \frac{2}{\delta} n.$$

V. Simulations

Let us study the application of the proposed algorithm to the authentic Stochastic Block Model and compare the results with spectral clustering which is considered as one of the state of the art methods (see [2, 14, 16]). We consider the regimes

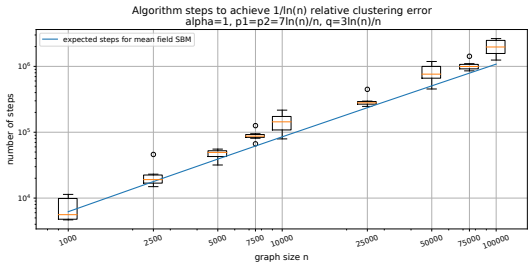

Fig. 3. Running time comparison for SBM and mean-field SBM.

were $p_1 = a\ln(n)/n, q = b\ln(n)/n$ and p_2 is automatically derived from (3) for some constants $a > 0, b > 0$ (for $\alpha = 1$, p_2 is equal to p_1). In this section for all simulations we use Algorithm 2 with the desired relative error $\delta = 1/(2\ln(n))$ and hence $T = 4 \cdot n\ln(n)$.

A. Symmetric case, $\alpha = 1$

First, we want to compare running times of the algorithm for G_{sbm} and G_{mf} graphs. We have theoretical estimates on the running time on G_{mf} (see the previous section). In contrast with the mean-field model, in the authentic SBM Algorithm 1 does not always increase (decrease) the number of black nodes in the first cluster monotonously converging to a local minimum of energy. In the stochastic case, some steps might go in the undesired direction producing a random walk on a segment $[0, n_1]$, were the "walker" corresponds to the class i of configuration $\sigma \in \Sigma^{(i)}$. Nevertheless, it appears that in most regimes the derived upper bound for the expected running time is close to the actual running time in the authentic SBM model.

For the first set of simulations we took $\alpha = 1, a = 7, b = 3$, which represents a difficult setting since the exact recovery for this regime is not possible according to conditions mentioned is Section II. For each graph of size n from the list, we have measured the number of steps needed to achieve $\delta = 1/(2\ln(n))$ relative clustering error. We repeat the experiment several times for each n. We have compared the output with the theoretical estimation provided for the mean-field model. The result is presented in Fig. 3 with log-log scale. It can be seen that for non-trivial regime the running time is very much as expected from the mean-field model.

At the same time on Fig. 4 we can see trajectories of the share of black nodes in the first cluster during the simulations of Algorithm 2 on two different graphs (difficult and very difficult), each with 50000 nodes. In the first plot, trajectories tend to 0 or 1, since the both are global optima of the objective; both are desired solutions since the clusters are balanced. Spikes at the end of each line correspond to the second round of the algorithm. Considering the first plot, we can distinguish three phases of the algorithm. Phase 1: the algorithm accumulates a critical shift from the equilibrium. In this phase accuracy changes slow, there is a small chance to do right update given the configuration is balanced. This is the most tricky phase – it is difficult to analyze the graph

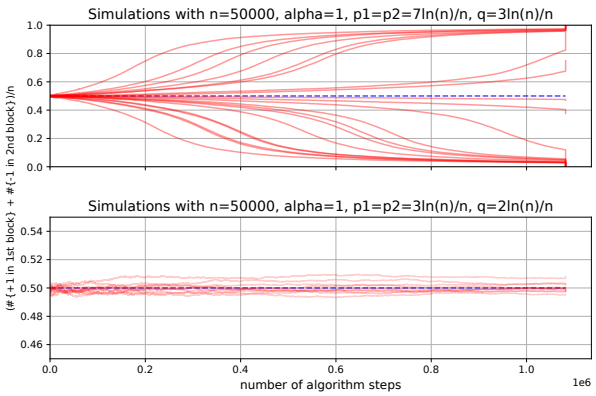

Fig. 4. Trajectories of the share of black labels in the first cluster ($\alpha = 1$).

topology and to estimate the number of steps needed here. Phase 2: judging by significant progress, the algorithm does most of the work here. In this phase the accuracy changes fast, right pair of nodes are likely to be updated in the correct way. Finally, phase 3: the algorithm is trying to hit wrongly-labeled nodes when there are only few of them. A right pair of nodes is almost surely updated in the correct way, but the probability to hit such pair is low since the error is low. In this phase the accuracy again changes slowly.

It is nice to observe that the upper bound for the mean-field model given in Proposition IV.5 seems to work for the authentic SBM as well. Another interesting remark is that the trajectories in the first plot never cross 50%-line, which confirms the algorithm's strong dependency on the initial configuration. Moreover, in the authentic SBM, the updates might be done in both right and wrong directions depending on the local graph topology. However, in the first plot of Fig. 4 we see smooth lines with no backward movements, when the task is not extremely difficult. In contrast, in the second plot, the trajectories look more like random walks tightly concentrated around the 50%-line, which corresponds to a random guess.

Next, we want to compare Algorithm 1 with Algorithm 2. We do that in several regimes: more or less difficult regimes, large or small graphs. The results are presented in Fig. 5; each bar corresponds to 5 independent simulations for a common random graph.

As expected, two-round algorithm significantly improves the accuracy of clustering. We would also like to note that we should not look at the average accuracy, but rather at the maximal accuracy, since we have the energy as a measure of clustering quality and can distinguish the best results from the rest. We also have some simulations that ended up in almost random partitioning, which means some of random initial configurations happened to be very poor.

Next we compare the accuracy of spectral clustering (python realization from sklearn library) with the accuracy of Algorithm 2 for the SBM with graph size $n = 10000$ and different parameters a and b (see Fig. 6, Fig. 7 and Fig. 8). In order to do so, we choose a trial with the best energy out

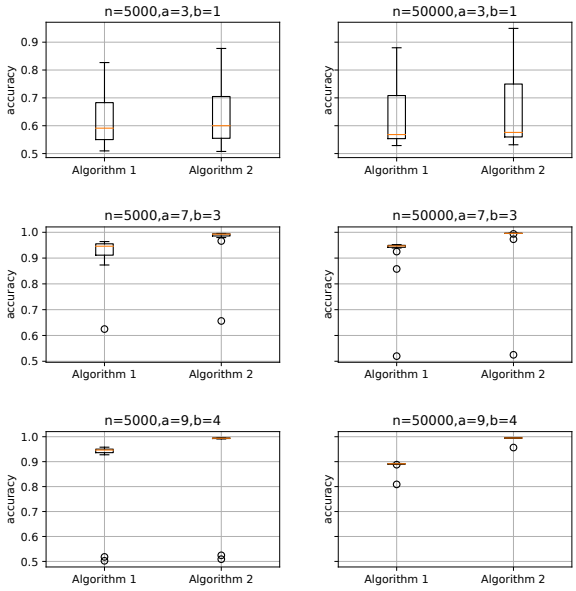

Fig. 5. Accuracy comparison of Algorithm 1 and Algorithm 2.

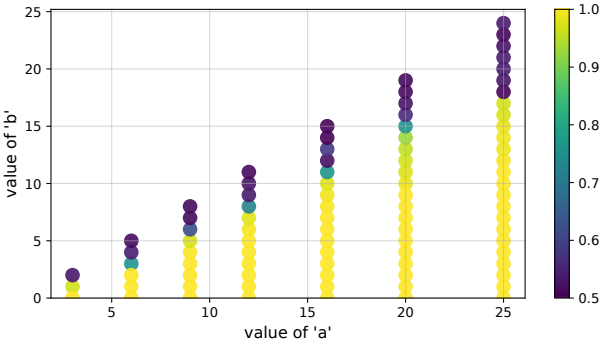

Fig. 6. Accuracy of Algorithm 2 in different regimes with $n = 10000, \alpha = 1$.

of 3 trials for Algorithm 2 and then measure its accuracy. As we can see, there are differences only in some extreme regimes where spectral clustering performs better, while in the majority of regimes the clustering result is perfect for both algorithms. We emphasize here, that spectral clustering has worst case time complexity $O(n^3)$ [17]. Often in practice spectral clustering has very good performance but to the best of our knowledge we do not know about any rigorous results about the average time complexity of spectral clustering. In contrast, our algorithm has the proven average number of algorithm steps $O(n \ln(n))$ for the mean-field SBM, with each step complexity of $O(\ln(n))$ in the original SBM graph given each node has logarithmic degree in the considered regimes.

B. Asymmetric case, $\alpha > 1$

Let us also briefly discuss some simulation results for the case of unbalanced blocks. We have chosen $\alpha = 3$ for the simulations as a significant but not extreme value and $n = 10000$. Spectral clustering algorithms work poorly on

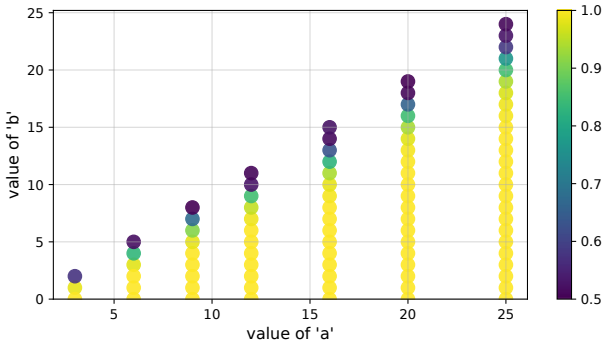

Fig. 7. Accuracy of Spectral Clustering in different regimes with $n = 10000, \alpha = 1$.

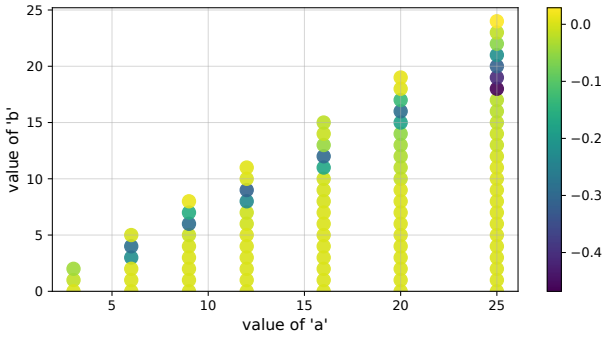

Fig. 8. Accuracy of Algorithm 2 minus accuracy of Spectral Clustering in different regimes with $n = 10000, \alpha = 1$.

such unbalanced graphs, so we only present the result of running Algorithm 2 in different regimes. For $\alpha = 3$ a natural initialization would be to distribute $n_1 = 2500$ black labels and $n_2 = 7500$ white labels uniformly at random having the average accuracy of $n_1^2/(n_1 + n_2) = 62.5\%$. As discussed in the previous sections, globally there are two minima of the expected energy, close to which the algorithm can possibly converge. In the case $\alpha = 3$, the desired optimum gives as before 100% accuracy and the other gives the accuracy of 50% since it corresponds to the case when all nodes from the small cluster are colored incorrectly. Having that said, we have the results presented in Fig. 9. Each point corresponds to a simulation with the best energy out of 5 trials.

As we can see, the algorithm works for a significant range of parameters a and b, though it is not very stable and more independent trials should be applied for each graph. Dark points surrounded by light ones mean that all 5 initial configurations were not good enough, i.e., they happened to occur on the wrong tail of the energy curve and thus led the algorithm to the wrong minimum. The trajectories of the accuracy in this case are presented in Fig. 10. Let us take a closer look at the second round of Algorithm 2. The results of good trajectories are practically improved to 100% accuracy. Whereas for the wrong trajectories accuracy falls below 50% which might look surprising at the first sight. This happens because of the weighted re-labeling provided by line 6 in

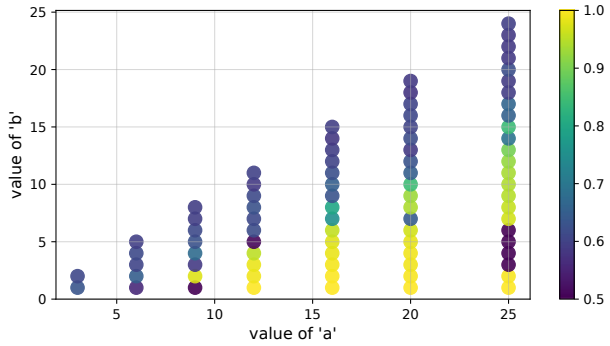

Fig. 9. Accuracy of Algorithm 2 in different regimes with $n = 10000, \alpha = 3$.

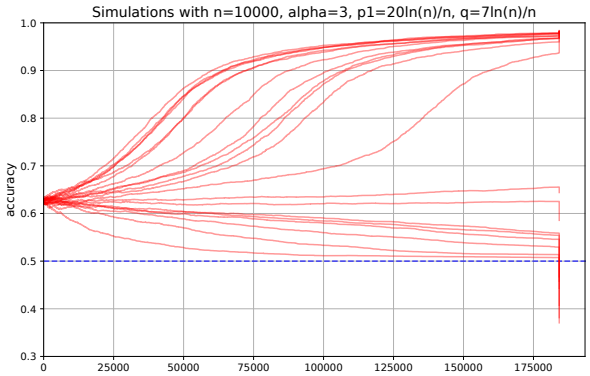

Fig. 10. Trajectories of accuracy with unbalanced clusters.

Algorithm 2.

VI. Conclusion and Future Research

To summarize, we see the following advantages of the proposed approach:

- the running time complexity of the algorithm is around $\bar{d} \cdot n \ln(n)$ for the SBM graphs;
- the algorithm can be effectively distributed over any number of machines with shared memory and with no need in synchronization;
- the algorithm can have a natural stopping rule based on the value of the desired global energy;
- the resulting accuracy is comparable with spectral clustering algorithm within large area of parameters;
- the algorithm works with high accuracy even in the case of unbalanced clusters;
- the approach can be customized with different objective functions.

At the same time, we have noticed the following deficiencies:

- the output of the algorithm is not reproducible, it is a result of a random process;
- the result and the quality of the algorithm strongly depends on the initial configuration;
- for the extremely difficult problems it works worse than the spectral clustering in the case of balanced clusters.

For the future research we see a number of opportunities. First, establishing theoretical results for the authentic SBM is of a significant interest. Second, different upgrades of the algorithm are possible, such as: at each step nodes could be chosen not at random but, e.g., according to the local energy; the second round of the algorithm might be different, some intermediate steps might be added in order to expand this approach for the case with no cluster-size constraints. Finally, different objectives (such as in [10] and [12]) can be used.

ACKNOWLEDGMENT

The simulations and numerical experiments part of the research was funded by RFBR, project number 20-31-90023. The other parts of the work has been done with support of Inria - Nokia Bell Labs "Distributed Learning and Control for Network Analysis" project. This is the author version of the paper accepted to ICPR 2020.

REFERENCES

[1] E. Abbe, "Community detection and stochastic block models." *Foundations and Trends in Communications and Information Theory*, vol. 14, no. 1-2, pp. 1–170, 2018.

[2] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, p. 395–416, 2007.

[3] K. Avrachenkov, V. Dobrynin, D. Nemirovsky, S. Pham, and E. Smirnova, "Pagerank based clustering of hypertext document collections." *Proceedings of the 31st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20–24*, 2008.

[4] K. Avrachenkov, M. Chamie, and G. Neglia, "Graph clustering based on mixing time of random walks." *IEEE International Conference on Communications, ICC 2014*, 2014.

[5] M. Chen, J. Liu, and X. Tang, "Clustering via random walk hitting time on directed graphs," in *AAAI*, vol. 8, 2008, pp. 616–621.

[6] M. Newman, "A measure of betweenness centrality based on random walks," *Social Networks*, vol. 27(1), p. 39–54, 2005.

[7] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *ISCIS*, 2005.

[8] M. E. Newman, "Equivalence between modularity optimization and maximum likelihood methods for community detection," *Physical Review E*, vol. 94, no. 5, p. 052315, 2016.

[9] V. V. Mazalov, "Comparing game-theoretic and maximum likelihood approaches for network partitioning," in *Transactions on Computational Collective Intelligence XXXI*. Springer, 2018, pp. 37–46.

[10] M. E. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[11] M. Blatt, S. Wiseman, and E. Domany, "Clustering data through an analogy to the Potts model," in *Advances in neural information processing systems*, 1996, pp. 416–422.

[12] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical review E*, vol. 74, no. 1, p. 016110, 2006.

[13] K. E. Avrachenkov, A. Y. Kondratev, V. V. Mazalov, and D. G. Rubanov, "Network partitioning algorithms as cooperative games," *Computational social networks*, vol. 5, no. 1, pp. 1–28, 2018.

[14] E. Abbe and C. Sandon, "Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, 2015, pp. 670–688.

[15] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE: Transactions on Information Theory*, vol. 62, no. 1, 2016.

[16] L. Su, W. Wang, and Y. Zhang, "Strong consistency of spectral clustering for stochastic block models," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 324–338, 2020.

[17] J. W. Demmel, O. A. Marques, B. N. Parlett, and C. Vömel, "Performance and accuracy of lapack's symmetric tridiagonal eigensolvers," *SIAM Journal on Scientific Computing*, vol. 30, no. 3, pp. 1508–1526, 2008.