The Maximum Label Propagation Algorithm on Sparse Random Graphs

Charlotte Knierim
ETH Zurich, Switzerland
cknierim@inf.ethz.ch

Johannes Lengler
ETH Zurich, Switzerland
johannes.lengler@inf.ethz.ch

Pascal Pfister
ETH Zurich, Switzerland
ppfister@inf.ethz.ch

Ulysse Schaller
ETH Zurich, Switzerland
ulysses@student.ethz.ch

Angelika Steger
ETH Zurich, Switzerland
angelika.steger@inf.ethz.ch

Abstract

In the Maximum Label Propagation Algorithm (`Max-LPA`), each vertex draws a distinct random label. In each subsequent round, each vertex updates its label to the label that is most frequent among its neighbours (including its own label), breaking ties towards the larger label. It is known that this algorithm can detect communities in random graphs with planted communities if the graphs are very dense, by converging to a different consensus for each community. In [17] it was also conjectured that the same result still holds for sparse graphs if the degrees are at least $C \log n$. We disprove this conjecture by showing that even for degrees n^{ε}, for some $\varepsilon > 0$, the algorithm converges without reaching consensus. In fact, we show that the algorithm does not even reach almost consensus, but converges prematurely resulting in orders of magnitude more communities.

2012 ACM Subject Classification Mathematics of computing → Random graphs; Theory of computation → Distributed algorithms

Keywords and phrases random graphs, distributed algorithms, label propagation algorithms, consensus, community detection

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.58

Category RANDOM

1 Introduction

In the last years, opinion exchange dynamics on graphs and networks has received much attention, see [19] for an excellent survey. Apart from the desire to improve our understanding of social processes, opinion exchange dynamics have also found applications in the fields of distributed computing and network analysis. For example, opinion exchange dynamics like the 3-majority protocol or the 2-choice dynamics have been proposed as simple distributed solutions to the basic problems of consensus forming, majority detection, and plurality consensus in distributed networks [2, 3, 5, 7, 10, 12, 13].

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019).
Editors: Dimitris Achlioptas and László A. Végh; Article No. 58; pp. 58:1–58:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Label propagation algorithms (LPA) are a certain kind of opinion exchange dynamics which have been used for community detection in networks [4, 14, 15, 21, 23]. Despite their great practical importance (see the surveys [4, 15, 23]), and although obtaining theoretical bounds on success and speed of LPAs was proposed as an important research question [1, 8, 18], rigorous theoretical analyses of LPAs have only appeared recently. The first such study by Kothapalli, Pemmaraju and Sardeshmukh [17] investigated an algorithm called `Max-LPA`. In this algorithm, each vertex starts with a random label in the interval $[0, 1]$. In each round, every vertex switches its label to the majority label in its neighbourhood (including its own label), breaking ties towards larger labels. In [17] `Max-LPA` was studied on an Erdős-Rényi model with planted communities.[1] In this model, the vertex set is partitioned as $V = V_1 \dot\cup \ldots \dot\cup V_k$, where all sets V_k have a certain minimal size. Then every edge inside of one of the sets V_i is inserted independently with some probability p_i, and every edge between different partite sets is inserted independently with probability $p' \ll p := \min_i\{p_i\}$. The study [17] considered dense cases, e.g. for $|V_i| = \Omega(n)$ they set $p = \Omega(n^{-1/4+\varepsilon})$ and $p' = O(p^2)$. For this case they show that `Max-LPA` successfully recovers the communities, i.e., it converges quickly to a state where for each i the labels within the set V_i are all identical, and any two distinct sets V_i have different labels. The authors of [17] conjectured that their conditions on p are very far from tight, and that it should suffice to require $p \geq C \log n/n$ (i.e., expected degrees of $C \log n$ instead of $\Omega(n^{3/4+\varepsilon})$), if there is a sufficient gap between p and p'.

The conjecture from [17] has been considered in at least two subsequent papers [6, 9], both of whom have tried to obtain results for LPA algorithms in sparser cases, see Section 1.1 below. However, the conjecture remained unresolved, and the behaviour of `Max-LPA` and other LPAs generally remained poorly understood in the sparse case. In this paper we show that the conjecture from [17] is actually false in a strong sense. Even in the extreme case of just one rather dense community, i.e. $p' = 0$, and $p = n^{-1+\varepsilon}$, `Max-LPA` fails to label the communities consistently. In fact, with high probability the algorithm gets stuck with all label classes having size $o(n)$. For this negative result, note that the case of just one community corresponds to running the `Max-LPA` process on a classical Erdős-Rényi graphs $G_{n,p}$, in which each edge is inserted independently with probability p. We thus phrase our main result for Erdős-Rényi graphs $G_{n,p}$ with expected degree $d := np \leq n^\varepsilon$.

▶ **Theorem 1.** *There exists constants $C > 0$ and $\varepsilon > 0$ such that for any $C \log n/n \leq p \leq n^{-1+\varepsilon}$, the `Max-LPA` process on an Erdős-Rényi graph $G_{n,p}$ terminates with $\Omega(d^3)$ different label classes each of size at most $O\left(n/d^3\right)$, where $d := np$ denotes the expected degree of a vertex in $G_{n,p}$.*

Note that our result does not immediately extend to smaller values of p, as the property "`Max-LPA` finds consensus on G" is not monotone.

A rigorous analysis of LPAs is challenging due to the high influence of dependencies. In [9] this was stated as: "The absence of substantial theoretical progress in the analysis of LPAs is largely due to the lack of techniques for handling the interplay between the non-linearity of the local update rules and the topology of the graph." In order to prove our result we need to combine local analysis of the dynamics with global properties of the graph, in particular by inventing a discharging technique for showing that no set of size $O(n/d^3)$ can propagate much further without the help of (suitably defined) unstable vertices. We hope that this technique will also turn out to be useful for the analysis of other LPAs.

[1] also called *clustered Erdős-Rényi graph* or *stochastic block model*

1.1 Other related work

There is a huge body of experimental work on LPAs for community detection, and we refer the reader to surveys [4, 15, 23]. Basic properties of `Max-LPA` were analysed in [20], and in particular it was shown that the algorithm always converges to a stable configuration or to a limiting cycle of length 2. An experimental comparison of `Max-LPA` with other tie-breaking techniques was performed in [8], with the conclusion that `Max-LPA` typically converges faster than other tie-breaking rules, but that repeated runs on the same graph produces slightly less consistent outcomes than other LPAs.

As outlined above, there are only very few rigorous mathematical results on LPAs. In an attempt to improve on results from [17], Cruciani, Natale, and Scornavacca [9] studied the 2-choices dynamics as a "sampling" approximation of an LPA. In this dynamics, a vertex updates its label to the majority label among its own and the labels of two randomly drawn neighbours, breaking ties towards its own label. For this variant of an LPA, [9] studied clustered regular graphs with two clusters, i.e., graphs with fixed degree d within each of the clusters, and fixed degree $d' \ll d$ between the clusters. While both, the algorithm and the setting, differ slightly from [17], the authors compare their results with [17] and improve the (average) degree d from $d = n^{3/4+\varepsilon}$ to $d = n^{1/2}$. The proof idea relies on expansion properties of the graph, and $n^{1/2}$ is a natural barrier for this approach.

In [6], Clementi et al. analysed the Erdős-Rényi model with two planted communities, for an LPA which uses a different tie-breaking rule than `Max-LPA`: it breaks ties randomly. For this variant they claim that a repeated application of this LPA with random tie breaking can successfully recover the communities with high probability, even for densities as sparse as $p = \Theta(1/n)$. However, their result has a catch: the analysis is only done for a *dynamic* version of the random graph model, in which all edges are re-drawn in every round. This simplifies the analysis considerably, since it removes dependencies between different rounds. In particular, the state of the algorithm at any time can be completely described by the *number* of vertices of each label in each of the partite sets, whereas in the static case the structural information is essential. As we show in this paper, the behaviour of the `Max-LPA` in the sparse case reveals that it fails precisely because of the structural properties of the label classes. In other words, the behaviour in the static and the dynamic case are completely different, and an analysis of the dynamic case in general will not explain the behaviour of the static case.

1.2 Some intuition on the `Max-LPA` process

In this section we provide some intuition for Theorem 1. Recall that we assume that, by definition of the `Max-LPA` process, each $v \in V$ chooses it original label uniformly and independently from $[0,1]$. With high probability all vertices will therefore have different labels. Throughout our paper we will thus assume without loss of generality that in the beginning of the process the labels of all vertices are pairwise different.

Consider a random graph $G_{n,p}$ and recall that we denote by $d = pn$ its average degree. Then almost all vertices will have a neighbour which holds one of the n/d highest labels (here and in the sequel of this overview we ignore polylog(d) factors). After the first round we thus expect that (almost) all but the largest n/d labels are extinct, i.e., are not present any more at any vertex.

Let us thus have a closer look at how such a high label evolves in the first few rounds of the `Max-LPA` process. Hence, consider a vertex v with a label ℓ that belongs to the n/d largest labels. We expect that v pushes ℓ to some of its neighbours. Depending on the size of ℓ, the number of neighbours which receive ℓ in round 1 can range from very few to almost all neighbours (but this is not our concern at the moment). In order to understand what can happen in subsequent rounds, we need to distinguish two cases: (i) v receives in round 1 a new label $\ell' > \ell$ or (ii) v keeps its initial label ℓ in round 1. In case (i), whenever the label ℓ got pushed onto enough neighbours of v in round 1, vertex v will get back its initial label ℓ in round 2. In this case quite a number of different scenarios can happen in further rounds, as for example v can receive back its initial label not only in round 2, but also in other subsequent rounds if the label ℓ was pushed down far enough into the r-neighbourhood of u.

In case (ii), on the contrary, v is already in a quite stable situation, as v and many of its neighbours have the same label. More precisely, in order for v to change its label in any of the subsequent rounds, (almost) all neighbours of v which receive label ℓ in round 1 need to lose ℓ again in one of the subsequent rounds. For any such neighbour $u \in N(v)$ this can either happen if, at some point in time, say t, the initial label of u is pushed back onto u or the neighbourhood of u will contain a different label $\ell' \neq \ell$ on two or more of its vertices. Note that, for most neighbours of v, the first case is unlikely, as the vast majority of neighbours of v initially held one of the $n - n/d$ labels which are extinct after the first round. The latter case, on the other hand, can only happen if u is in a cycle of length at most $2(t+1)$ (here we use that we assumed that in the beginning all vertices have pairwise different labels). Note that for each constant $t \in \mathbb{N}$ there exists a constant $\varepsilon = \varepsilon(t) > 0$ so that the random graph $G_{n,p}$ with $\omega(1/n) \leq p \leq n^{-1+\varepsilon}$ has the property that all but a negligible number of vertices are *not* contained in a cycle of length at most t. Hence, almost all vertices $v \in V$ which keep their label in round 1, together with their neighbours $u \in N(v)$ which receive the label of v in round 1 and initially held one of the $n - n/d$ smallest labels, are in a quite stable configuration already after the first round.

After the first round only a constant fraction of vertices are in such a stable configuration. However, by analysing the subsequent rounds of the process in a similar fashion we can show that after the first few rounds, all but n/d^4 vertices are in such a stable configuration.

Once we have shown that all but n/d^4 vertices are in stable configurations, we still have to argue that the n/d^4 non-stable vertices will not be able to break up these stable configurations. To handle this case we use a discharging argument to show that if a label class grows to size n/d^3, then this label class induces a subgraph with density at least $3/2$. As a random graph $G_{n,p}$ with $\log(n)/n \leq p \leq n^{\varepsilon}$ whp does not contain such a set, we deduce that no label class gets that large. We note that this idea is similar to techniques used to study k-bootstrap percolation on Erdős-Rényi graphs [11], in which an initial set of active vertices successively activates every vertex that has at least k active neighbours. Our case can be viewed as a (hypothetical) $3/2$-bootstrap percolation.

1.3 Notation and terminology

Our graph-theoretic notation is standard and follows that of [22]. In particular, for a graph G, we denote by V and E the set of vertices and edges, respectively. Moreover, $e(G) := |E|$ is the number of edges of G. For any subset $S \subseteq V$ we let $G[S]$ denote the subgraph of G that is induced by the vertices of S. We denote by $E(S)$ the set of edges of $G[S]$ and define $e(S) := |E(S)|$. For any two disjoint subsets $S, T \subset V$ we denote by $E(S, T)$ the set of edges with one endpoint in S and the other in T and define $e(S, T) := |E(S, T)|$. For a vertex $v \in V$, we denote by $N(v)$ the neighbourhood of v, which excludes v, and by $d(v) := |N(v)|$ its degree. For any positive integer $r \geq 2$, the r-neighbourhood of a vertex v, denoted by $N^r(v)$, is the set of vertices that can be reached from v by a path of length at most r (i.e. the r-neighbourhood of a vertex v includes v).

We consider the classical random graph model from Erdős and Rényi. For a positive integer n and $0 \leq p := p(n) \leq 1$. We denote by $G_{n,p}$ the probability space over graphs on n vertices where every possible edge is present with probability p independently of all other edges. We write $d := np$ for the expected degree of a graph $G \sim G_{n,p}$ and $\Delta(G)$ for its maximum degree. We say that an event $\mathcal{E} = \mathcal{E}(n)$ happens *with high probability*, or *whp*, if $\Pr[\mathcal{E}(n)] \to 1$ for $n \to \infty$. We use the notation $\tilde{O}(.)$ to hide polylog(d) factors.

2 Some properties of random graphs

In this section, we provide some results on random graphs which are important in our analysis. First, we state a standard result from random graph theory on degree concentration (the proof follows e.g. from [16, Corollary 2.3] together with a union bound).

▶ **Lemma 2.** *For every $\varepsilon > 0$ there exists a positive constant C such that for $G \sim G_{n,p}$ with $p \geq \frac{C\log(n)}{n}$ with probability at least $1 - 2ne^{-\frac{\varepsilon^2 d}{3}}$, it holds for every vertex $v \in V$ that*

$$(1-\varepsilon)d \leq d(v) \leq (1+\varepsilon)d,$$

where $d = pn$.

The next lemma gives a lower bound on the number of vertices that do not have a neighbour in a large subset.

▶ **Lemma 3.** *Let $M > 0$, $\varepsilon > 0$, $\log(n)/n \leq p \leq n^{-1+\varepsilon}$ and $G = (V,E) \sim G_{n,p}$. Let $S \subset V$ be a subset of vertices of size $|S| = \Omega\left(\frac{n(\log(d))^2}{d}\right)$, where $d = pn$. Then whp all but at most nd^{-M} vertices $v \in V \setminus S$ have at least one neighbour in S.*

Proof. As each vertex $v \in V \setminus S$ has $|S|$ opportunities to have an edge into S we have that

$$\Pr\left[e(v,S) = 0\right] = (1-p)^{|S|} \leq e^{-\Omega((\log(d))^2)} = d^{-\Omega(\log(d))}.$$

Thus, letting X be the random variable which counts the number of vertices in $V \setminus S$ with no neighbour on S and writing X as a sum of indicator random variables, we can conclude that

$$\mathbb{E}[X] \leq nd^{-\Omega(\log(d))}.$$

Setting $t = nd^{-M} - \mathbb{E}[X] = \omega(\log(n))$, the proof now follows by Chernoff bounds. ◀

In the following we argue that there are not many vertices which are in short cycles.

▶ **Lemma 4.** *Let k be a positive integer and let $G \sim G_{n,p}$ with $p = \omega(1/n)$. Then whp the number of cycles of length at most k is less than d^{k+1}, where $d = pn$. Therefore, at most kd^{k+1} vertices are contained in such cycles.*

Proof. For $1 \leq i \leq k$ let X_i be a random variable counting the number of cycles of length i in G. Then the expectation of X_i is given by

$$\mathbb{E}[X_i] = \binom{n}{i}\frac{(i-1)!}{2}p^i \leq d^i \leq d^k,$$

where the last inequality follows as $d \geq 1$. Thus, letting $X := \sum_{i=1}^{k} X_i$ be the random variable counting the number of cycle of length at most k, we can conclude by linearity of expectation that

$$\mathbb{E}[X] \leq kd^k.$$

Using Markov's inequality we get

$$\Pr\left[X \geq d^{k+1}\right] \leq \frac{k}{d} = o(1)$$

which finishes the proof, as each cycle contains at most k vertices. ◀

3 Notation and terminology for the Max-LPA process

Let us assume that the Max-LPA process runs on a random graph $G \sim G_{n,p}$ with $C\log(n)/n \leq p \leq n^{-1+\varepsilon}$, for some suitable constants $C > 0$ and $\varepsilon > 0$. We first introduce some notation. We define $\mathcal{L}$ to be the set of labels used in the Max-LPA process and for any $t \geq 0$ we denote by $\ell_t(v)$ the label of a vertex $v \in V$ after the t-th round of the Max-LPA process. Recall that we assume without loss of generality that all labels $\ell_0(v)$ are distinct. For any label $\ell \in \mathcal{L}$ we denote by v_ℓ the (by our assumption unique) vertex from which label ℓ originated from, i.e. $\ell_0(v_\ell) = \ell$. Moreover, we call a label ℓ *extinct* in a round $t > 0$ if there exists no vertex $v \in V$ with $\ell_t(v) = \ell$.

During the Max-LPA process, we say that a vertex v *propagates* its label onto $u \in N(v)$ in the t-th round of the process if $\ell_{t-1}(v) = \ell_t(u)$ and $\ell_{t-1}(v) \neq \ell_{t-1}(u)$. If $\ell_i(u) \neq \ell_{t-1}(v)$ for all $i \leq t-1$ (i.e. if u never held the label $\ell_{t-1}(v)$ so far), we say that v *forward-propagates* the label $\ell_{t-1}(v)$ onto u in round t. Otherwise, we say that v *back-propagates* the label $\ell_{t-1}(v)$ onto u in round t. In abuse of notation, we will also call a label ℓ to be *forward-propagating* and *backward-propagating* from or onto a vertex v.

As we saw in the introduction, in the absence of (short) cycles, back-propagation is essentially the only way that a vertex which has a neighbour holding the same label can change its current label. Thus, in order to create stable configurations, we need to understand such back-propagation of labels properly. The following definition will help us do so.

▶ **Definition 5.** *Let $v \in V$, $t \geq 0$ and assume that ℓ is a label which v holds for the first time in round t, i.e. $\ell_t(v) = \ell$ and $\ell_i(v) \neq \ell$ for all $i < t$. Then we define the ℓ-*propagation set *of v to be the vertex v together with all vertices w for which there exists a path $v = v_0, \ldots, v_k = w$ such that for all $0 \leq i \leq k$ the vertex v_i receives label ℓ in round $t+i$ from v_{i-1} and did not hold it in any round $j < t+i$.*

Recall from Section 1.2 that two vertices u and v that are connected by an edge and that hold the same label in some round t can only change their label if they are in a short cycle or if a label is back-propagated. In the following definition we only capture the latter (as we will argue in Section 4 that we do not have to consider vertices that are in short cycles).

▶ **Definition 6.** *Let $t \geq 0$ and assume that the Max-LPA process has run for t rounds. For any $v \in V$ we denote by $L_v^{(t)} := \{\ell \in \mathcal{L} : \exists i \leq t \text{ such that } \ell_i(u) = \ell\}$ the set of labels v held in the first t rounds of the process. We then call an edge $\{u, v\} \in E$* stable *after round t if*

- **(i)** *$\ell_t(v) = \ell_t(u)$,*
- **(ii)** *for all forward-propagating labels $\ell \in L_v^{(t)} \setminus \{\ell_t(v)\}$ of v we have that no vertex in the ℓ-propagation set of v holds the label ℓ after round t, and*
- **(iii)** *for all forward-propagating labels $\ell \in L_u^{(t)} \setminus \{\ell_t(u)\}$ of u we have that no vertex in the ℓ-propagation set of u holds the label ℓ after round t.*

A vertex $v \in V$ is then called stable *after round t if it belongs to a stable edge after the t-th round. All other vertices are called* unstable *after round t.*

Moreover, we call an unstable vertex vulnerable *in round $t+1$ if the labels of all vertices in $N(v) \cup \{v\}$ are pairwise different after round t.*

Note that being stable a priori does not mean that a vertex will never change its label again. However, we can show:

▶ **Lemma 7.** *Let t_0 and t be positive integers with $t < t_0$ and let $u, v \in V$ be adjacent vertices. If $\{u, v\}$ is a stable edge at time t and neither u nor v belong to a cycle of length $2t_0$ or less then both u and v will keep their label (and thus remain stable vertices) until round t_0.*

Proof. Let $t \leq i < t_0$ and let us assume that $\{u, v\}$ is a stable edge in round i. We claim that then the same is true in round $i + 1$ as well. Indeed, as u and v are in a stable edge and not contained in cycles of length less than $2t_0$, we have that any label $\ell \in L_u^{(t)} \setminus \ell_t(u)$ appears at most once in the neighbourhood of u namely on the unique vertex which pushed the label ℓ onto u. As the same is true for v as well, no label ℓ can be back-propagated onto u or v.

Hence, the only way for u or v to change their label is if they see a label ℓ in their neighbourhood at least twice. As these two neighbours cannot be connected (else u or v would be in a triangle), this can only happen if there exist two different paths from u or v to the vertex v_ℓ from which the label ℓ originated from. Hence, the appearance of the label ℓ in two neighbours of u (or v) in round $t \leq i < t_0$ implies the existence of a cycle of length at most $2i + 2 \leq 2t_0$ which contains u (or v). As this is a contradiction, we have that u and v will not change their label in the i-th round. Thus $\ell_{i+1}(u) = \ell_{i+1}(v)$. As furthermore, again since u and v are not contained in cycles of length less than $2t_0$, no label $\ell \in L_u^{(t)} \setminus \ell_t(u)$ (or $\ell \in L_v^{(t)} \setminus \ell_t(v)$) can reappear in the ℓ-propagation set of u (or v) through a path from v_ℓ not through u (or v), the edge $\{u, v\}$ remains stable as desired. The statement of the lemma now follows by a simple induction. ◀

4 The first two rounds of the process

In this section we carefully analyse the first two rounds of the process and show that after two rounds there are at most $\tilde{O}(n/d)$ unstable vertices left. As explained in the introduction, we are mainly interested in the behaviour of the highest labels, hence we define the following.

▶ **Definition 8.** *For a label $\ell \in \mathcal{L}$, let the rank of ℓ be $\mathrm{rk}(\ell) := |\{\ell' \in \mathcal{L} \mid \ell' \leq \ell\}|$. In particular, the smallest label has rank 1, and the largest label has rank n. Then we define*

$$L_X = \left\{ \ell \in \mathcal{L} \colon \mathrm{rk}(\ell) \geq n \left(1 - \frac{(\log(d))^2}{d^3} \right) \right\},$$
$$L_Y = \left\{ \ell \in \mathcal{L} \colon n \left(1 - \frac{(\log(d))^2}{d^2} - \frac{(\log(d))^2}{d^3} \right) \leq \mathrm{rk}(\ell) < n \left(1 - \frac{(\log(d))^2}{d^3} \right) \right\},$$
$$L_Z = \left\{ \ell \in \mathcal{L} \colon n \left(1 - \frac{(\log(d))^2}{d} \right) \leq \mathrm{rk}(\ell) < n \left(1 - \frac{(\log(d))^2}{d^2} - \frac{(\log(d))^2}{d^3} \right) \right\}.$$

Additionally, we denote by $X^{(t)}$, $Y^{(t)}$ and $Z^{(t)}$ the sets of vertices holding labels in L_X, L_Y and L_Z after round $t \geq 0$, respectively.

We are particularly interested in vertices that propagate their label to at least one neighbour in the first round but also lose their label in this round. Those are vertices which initially engage in back-propagation. As this can cause a series of troubles, we call those vertices and their labels "bad". More precisely, we define the following:

▶ **Definition 9.** *A label ℓ is called* good *if $\ell_1(v_\ell) = \ell$ and v_ℓ forward-propagates label ℓ in the first round to at least one of its neighbours. Otherwise, we call ℓ* bad.

As not only vertices which initially hold a bad label can cause troubles, but also all vertices which (potentially) hold such a bad label in later rounds, we also consider the 2-neighbourhood of such vertices.

▶ **Definition 10.** *We denote by*

$$X^{(2)}_{\text{bad}} := \bigcup_{\substack{\ell \in L_X \\ \ell \text{ is bad}}} N^2(v_\ell)$$

the set of all vertices that are in the 2-neighbourhood of a vertex $v \in X^{(0)}$ initially holding a bad label. Similarly, we define

$$Y^{(2)}_{\text{bad}} := \bigcup_{\substack{\ell \in L_Y \\ \ell \text{ is bad}}} N^2(v_\ell).$$

Additionally, for any bad label $\ell \in L_X$ or $\ell \in L_Y$ we call the 2-neighbourhood $N^2(v_\ell)$ an X_{bad}-set *and a* Y_{bad}-set*, respectively.*

In the later rounds the following situation can also arise. Whenever we have a vulnerable vertex, it can propagate its label and get a new label in the same round. This again results in vertices which engage in back-propagation. The following rather general definition will later allow us to capture all such situations. The definition may remain a bit mysterious at first glance, but we will show later why it makes sense (see Section 5).

▶ **Definition 11.** *We define two sets set of vertices, called A-*nodes *and B-*nodes*, as follows:*

$$A := \bigcup_{v \in X^{(0)}} \bigcup_{w \in N(v) \cap Z^{(0)}} (N(w) \setminus \{v\})$$

$$B := \bigcup_{v \in Y^{(0)}} \bigcup_{w \in N(v) \cap Z^{(0)}} (N(w) \setminus \{v\})$$

With the above definitions at hand, we can state our first main lemma, which summarises the behaviour of the algorithm in the first two rounds.

▶ **Lemma 12.** *For every $M > 0$ there exist $C > 0$ and $\varepsilon > 0$ such that the following holds. Let $\frac{C \log n}{n} \leq p \leq n^{-1+\varepsilon}$ and assume that we run the `Max-LPA` process on a graph $G \sim G_{n,p}$. Then with high probability there exists a set $D \subseteq V$ of size at most nd^{-M} such that the following statements hold:*

(ACYC) *All cycles in $G[V \setminus D]$ have length larger than* 200.

(NBZ) *Every vertex in $V \setminus D$ has at least one neighbour in $Z^{(0)}$. Moreover, all vertices in $V \setminus D$ which forward-propagate in round 1 hold a label from L_X, L_Y or L_Z.*

(NBY) *Every vertex in $V \setminus D$ has at least one neighbour in $Y^{(1)}$. Moreover, all vertices in $V \setminus D$ which forward-propagate in round 2 hold a label from L_X or L_Y.*

(NBX) *Every vertex in $V \setminus D$ has at least one neighbour in $X^{(2)}$. Moreover, all vertices in $V \setminus D$ which forward-propagate in round 3 hold a label from L_X.*

(UNST-TYPES2) *After the second round, every unstable vertex in $V \setminus N(D)$ is in $Z^{(0)}$, A, B, $X^{(2)}_{\text{bad}}$ or $Y^{(2)}_{\text{bad}}$.*

(UNST2X) *There are* $\tilde{O}(n/d^3)$ *vertices in* $X^{(2)}_{\text{bad}}$.

(UNST2Y) *There are* $\tilde{O}(n/d)$ *vertices in* $Y^{(2)}_{\text{bad}}$.

(UNST2Z) *There are* $\tilde{O}(n/d)$ *vertices in* $Z^{(0)}$.

(UNST2A) *There are* $\tilde{O}(n/d^2)$ *vertices in* A.

(UNST2B) *There are* $\tilde{O}(n/d)$ *vertices in* B.

(UNST2) *There are* $\tilde{O}(n/d)$ *unstable vertices after the second round.*

As the proof of Lemma 12 is rather long, we only give a short overview. As a first step one can argue that the labels from L_Y and L_X propagate their labels to a linear fraction of their 1-Neighbourhood and 2-Neighbourhood, respectively. I.e. one can show that both $Y^{(1)}$ and $X^{(2)}$ are of size $\Theta(n(\log(d))^2/d)$. Thus, by Lemma 3 and Lemma 4, we can define D as the 5-Neighbourhood of all vertices which do not have a neighbour in $Z^{(0)}$ or $Y^{(1)}$ or $X^{(2)}$ and which are contained in cycles of length 200 or less, and the first four items of Lemma 12 then follow easily.

The proof of (UNST-TYPES2) is a rather involved case analysis. In a nutshell, one can show that all vertices $v \in V \setminus N(D)$ which are not contained in $Z^{(0)}$, A, B, $X^{(2)}_{\text{bad}}$ or $Y^{(2)}_{\text{bad}}$ are connected through a stable path of length at most two to a vertex v_ℓ for some good label ℓ.

To show items (UNST2X) to (UNST2B) one can calculate the number of bad labels in L_X and L_Y using standard probabilistic tools (as e.g. Chernoff bounds). The sizes of these five sets then simply follow from their definition and Lemma 2. Last, note that (UNST2) is just a summary of the other statements of Lemma 12.

5 The next rounds of the Max-LPA process

The main goal of this section is to show the following proposition.

▶ **Proposition 13.** *After at most* 100 *rounds, whp the number of unstable vertices is* $\tilde{O}(n/d^4)$.

In the remainder of the paper, we will mostly neglect vertices which are outside the 101-neighbourhood of D. Since $|N^{101}(D)| = o(n/d^4)$ (if we choose the constant M in Lemma 12 large enough) the above proposition also follows if we only show that a $1-\tilde{O}(n/d^4)$ fraction of vertices in $V \setminus N^{101}(D)$ is stable after round 100. The main advantage of neglecting these vertices is that all vertices we consider in this section, and their complete 100-neighbourhood, are not contained in cycles of length less than 200. Thus, all structures we analyse (such as X_{bad}-sets, Y_{bad}-sets and ℓ-propagation sets, label classes) are actually trees (and in the following, we hence also refer to them as e.g. ℓ-propagation trees instead of ℓ-propagation sets). To ease notation, we henceforth denote for any set $S \subseteq V$ by $\hat{S} := S \setminus N^{101}(D)$ the set S without the 100-neighbourhood of D.

As the proof of the above proposition is quite involved, we only provide a detailed overview.

By Lemma 12, we know that after the second round a $1 - \tilde{O}(1/d)$ fraction of the vertices in $\hat{V}$ is already stable. In order to argue that the number of unstable vertices drops even further over the next few rounds, we cover all unstable vertices in $\hat{V}$ by the five classes $\hat{X}_{\text{bad}}$, $\hat{Y}_{\text{bad}}$, $\hat{Z}^{(0)}$, $\hat{A}$ and $\hat{B}$, depending on the mechanism that keep them unstable. Note that the definitions of $\hat{X}_{\text{bad}}$, $\hat{Y}_{\text{bad}}$, $\hat{Z}^{(0)}$, $\hat{A}$ and $\hat{B}$ are a bit over-pessimistic: vertices may occur in several classes, or several times in the same class, and all classes may also contain some stable vertices. However, our definitions allow us to show that the vertices in each class behave as if they were randomly distributed. In particular, for a given vertex, say, of type $\hat{X}_{\mathrm{bad}}$, it is easy to count the number of neighbours in other $\hat{X}_{\mathrm{bad}}$-structures, in $\hat{Y}_{\mathrm{bad}}$-structures, and so on.

The weighted meta-graph in Figure 1 summarizes the situation after round 2. Each vertex in the meta-graph corresponds to one (or two) of the five classes of unstable vertices (we depict $\hat{Z}^{(0)}$ and $\hat{B}$ as one vertex in all our figures because they evolve identically). The weights on the edges in the meta-graph are guided by the following idea. Assume that a vertex is, say, in an $\hat{X}_{\mathrm{bad}}$-structure, and assume pessimistically that its label can take over arbitrary parts of this structure. Then the weights of the arrows going out from $\hat{X}_{\mathrm{bad}}$ are (an upper bound for) the expected number of structures of other types that the label sees (and thus, can potentially take over). If we pessimistically assume that it also takes over these new structures, then the weights of the outgoing edges in the meta-graph also gives a bound on the expected number of structures that the label can see form there, and so on. Thus if for each walk in the meta-graph we multiply the weights along a walk, and then sum these products over all walks in the meta-graph, then we obtain an a bound for the expected number of structures that the label can take over.

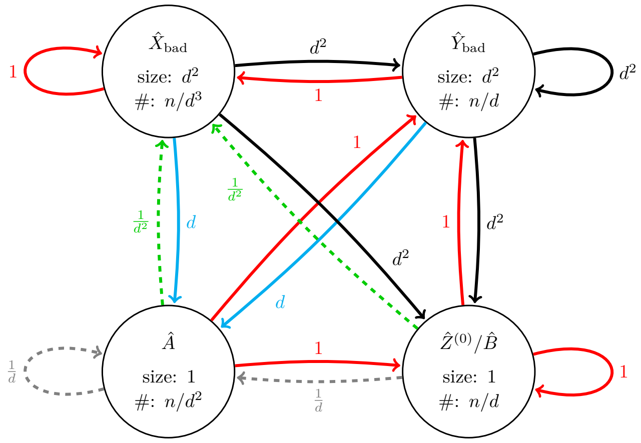

Figure 1 Meta-graph after round 2. The first value in a node S indicates the size of a structure S, the second value indicates number of vertices (counted with multiplicity) in structures of type S. The weight x of a meta-edge from node S to node T indicates that from a random S-structure S_0 there are in expectation at most x edges to T-structures (not counting edges within S_0 if $S = T$). The same bound is also valid if S_0 is not random among all S-structures, but rather a random neighbour of a T'-structure, for any type T' that appears in the meta-graph. All values are upper bounds and suppress any polylog(d) factors.

An important intermediate goal is to show that the graph $G' = (V', E')$ induced by unstable vertices from $\hat{V}$ is scattered, i.e. that most vertices are in small components. To this end, we would like to have that for any walk in the meta-graph, the weights of the edges multiply up to something of size $o(1)$. Then, if we start with a random vertex in V', all paths in G' containing this vertex are short, since its unstable structure is only connected to few other meta-vertices. Hence, the vertex has small probability to be in a large component. Unfortunately, after round 2 the meta-graph is still much too dense for our purpose.

Analysing the `Max-LPA` process further, we show that in the third round the number of unstable $\hat{Z}^{(0)}$ and $\hat{B}$ structures decreases by a factor of $\tilde{O}(1/d)$. For the $\hat{Y}_{\mathrm{bad}}$ structures, something even more dramatic happens: most of them scatter internally by round 5, in the sense that if we choose a random vertex in a random $\hat{Y}_{\text{bad}}$ structure, then the expected size of the component of this vertex within the $\hat{Y}_{\text{bad}}$ structure shrinks to $\tilde{O}(1)$. The analysis after round 2 becomes more tricky, since the structures lose their independence. For example, although the number of $\hat{Z}^{(0)}/\hat{B}$-structures decreases, the weight of the arrow $\hat{A} \to \hat{Z}^{(0)}/\hat{B}$ does not decrease because neighbours of $\hat{A}$-vertices are biased towards remaining unstable. However, we can prove that some weights decrease as one would expect for random edges, namely for the edges between $\hat{Y}_{\text{bad}}$ and $\hat{Z}^{(0)}/\hat{B}$, and the edges that go out of $\hat{Y}_{\text{bad}}$. The meta-graph after round 5 can be found in Figure 2.

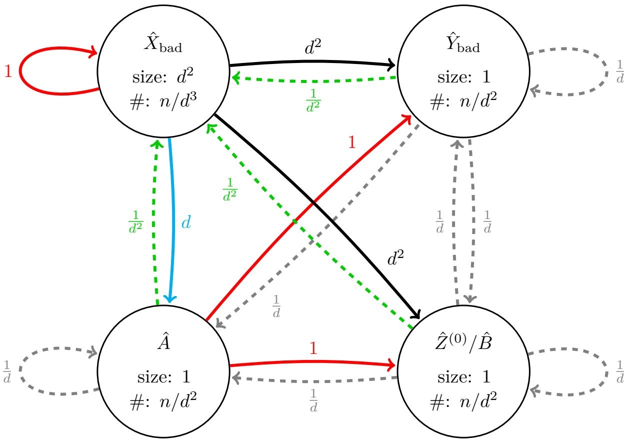

Figure 2 Meta-graph after round 5. The first value in a node S indicates, given a random vertex v in a random structure S_0 of type S, the size of the connected component of v induced by unstable vertices in S_0. The weight of the edge from S to T indicates how many edges into T-structures there are in expectation from this connected component. Again, all bounds remain valid if we choose v as a random vertex as seen from structures of some type T'. The second value in a node S indicates the number of unstable vertices in structures of type S. All values are upper bounds and suppress any polylog(d) factors. Compared to round 2, only a $\tilde{O}(1/d)$-fraction of the nodes in $\hat{Y}_{\text{bad}}$ and in $\hat{Z}^{(0)}/\hat{B}$ remain unstable. This decreases the weights from Y_{bad} and Z/B into Y_{bad} and Z/B by a factor of $1/d$, but not the weights from $\hat{X}_{\text{bad}}$ or $\hat{A}$ into these sets, because the target nodes of these edges are biased towards remaining unstable. Moreover, the remaining $\hat{Y}_{\text{bad}}$-structures are scattered into connected components of expected size $\tilde{O}(1)$, which decreases the weights of all outgoing edges from $\hat{Y}_{\text{bad}}$ by $1/d^2$. There are no cycles in which the weights multiply to strictly more than one, and every cycle with a product of one is a combination of the cycles $\hat{X}_{\text{bad}} \to \hat{X}_{\text{bad}}$, $\hat{X}_{\text{bad}} \leftrightarrow \hat{Y}_{\text{bad}}$ and $\hat{X}_{\text{bad}} \leftrightarrow \hat{Z}^{(0)}/\hat{B}$.

After round 5, there are still some cycles for which the weights do not multiply to $o(1)$, in particular the loop at $\hat{X}_{\text{bad}}$ and the cycles $\hat{X}_{\text{bad}} \leftrightarrow \hat{Z}^{(0)}/\hat{B}$ and $\hat{X}_{\text{bad}} \leftrightarrow \hat{Y}_{\text{bad}}$. To deal with those, we use the fact that a typical leaf v of an $\hat{X}_{\text{bad}}$-structure sees neighbours in $\hat{Y}_{\text{bad}}$ and in $\hat{Z}^{(0)}/\hat{B}$ after round 2, but these do not see further unstable neighbours in round 3 and 4 with probability $1 - \tilde{O}(1/d)$. In this case, we show that the leaf v stabilizes by round 20, and we may decrease the weights from $\hat{X}_{\text{bad}}$ to $\hat{Y}_{\text{bad}}$ and to $\hat{Z}^{(0)}/\hat{B}$ accordingly (Figure 3). In the remaining meta-graph, all walks accumulate decreasing factors except for the loop for $\hat{X}_{\text{bad}}$. However, since there are only $\tilde{O}(n/d^4)$ vertices in $\hat{X}_{\text{bad}}$ structures, we can show that even the union of the connected components in G' of all these structures has size $\tilde{O}(n/d^4)$. For all vertices outside of this union, a long path in G' from such a vertex induces a long walk in the meta-graph, which is unlikely. Thus almost all vertices are either among the $\tilde{O}(n/d^4)$ vertices of $\hat{X}_{\text{bad}}$ components, or are in components of diameter less than K, for a suitable constant K. For the latter one, we show that they stabilise after K further rounds if the vertices are not contained in any cycles of length at most $2K$.

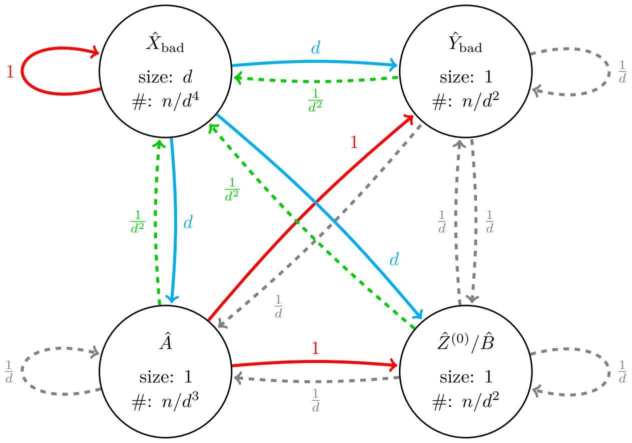

Figure 3 Meta-graph after round 20. Values have the same meaning as in Figure 2. Compared to Figure 2, the X_{bad}-structures have decreased by a $1/d$ factor in component size and in the number of unstable vertices, and the number of outgoing edges from $\hat{X}_{\text{bad}}$ to $\hat{Y}_{\text{bad}}$ and to $\hat{Z}^{(0)}/\hat{B}$ has decreased by a factor of $1/d$. The only remaining cycle in which the labels multiply to one is the loop at $\hat{X}_{\text{bad}}$.

6 Finishing the proof

For the last part of the proof, we fix a label, and show that this label cannot take over the complete graph. We only give the proof under the following simplifying assumption, and defer the full proof to the journal version. More precisely, we will assume that the label is $\ell_{\max}$, the maximum label, and after round 100 we change the labels of all unstable vertices and all vertices whose label appears in a cycle of length at most 200 to $\ell_{\max}$. This gives the label class of $\ell_{\max}$ a considerable boost after round 100, but it also simplifies the setting. In particular, since every other vertex v is stable and not in a cycles of length at most 200, the definition of stable implies that all neighbours of v have either the label of v, the label $\ell_{\max}$, or other mutually distinct labels. Moreover, after round 100 they have at least one neighbour of the same label, so they can only change their label to $\ell_{\max}$. This remains true inductively, since if a vertex v loses its neighbours of the same label, then those neighbours change their label to $\ell_{\max}$, and thus v also changes its label to $\ell_{\max}$. Thus the only possible change in the remaining graph is that vertices change their label to $\ell_{\max}$.

Let us first estimate the number of vertices that have or receive label $\ell_{\max}$ after round 100. There are at most $\tilde{O}(n/d^4)$ unstable vertices by Lemma 13. Moreover, at this point no label class has swallowed more than its 100-neighbourhood, which has size $O(d^{100}) = \tilde{O}(n/d^4)$ whp (Lemma 2). Consider some $\delta > 0$. By choosing $\varepsilon = \varepsilon(\delta)$ sufficiently small, if follows from Lemma 4 that the number of vertices that are contained in cycles of length at most 200 is $O(n^\delta)$. Since each label class has at most size $O(d^{100})$, the number of vertices with labels that appear in such cycles is $O(d^{100}n^\delta) = \tilde{O}(n/d^4)$, if we choose δ sufficiently small. Thus after round 100 the label class of $\ell_{\max}$ has size $\tilde{O}(n/d^4)$. In the following, we will show that with high probability, the structure of $G_{n,p}$ is such that no set of this size can take over all the stable vertices in the graph.

First we argue that in order to take over a certain set of stable vertices S from one of the stable trees, there needs to be a certain number of edges going from S to the vertices holding label $\ell_{\max}$. Let $T \subseteq V$ be the set of vertices that initially (i.e., after the relabelings in round 100) have label $\ell_{\max}$, and denote for each $\ell \neq \ell_{\max}$ by V_ℓ the set of vertices with label ℓ at this time. Now fix some later point in time t. Let $T' \supseteq T$ be the set of vertices with label $\ell_{\max}$ at round t, and for each $\ell \neq \ell_{\max}$, let $S_\ell = V_\ell \cap T'$ be the set of vertices with label ℓ that have been taken over by $\ell_{\max}$ by round t. Then we claim that

$$e(T' \setminus T, T) + e(T' \setminus T) \geq |T' \setminus T| + \sum_{\ell \in \mathcal{L}, \ell \neq \ell_{\max}} (e(S_\ell, V_\ell \setminus S_\ell) + e(S_\ell)). \qquad (1)$$

To prove (1), let us assume that $v_1, v_2, \ldots, v_k$ is the order in which the vertices of $T' \setminus T$ acquire the label $\ell_{\max}$, where we break ties arbitrarily. For an index $i \leq k$, let ℓ_i be the label of v_i and let $T_i := T \cup \{v_1, \ldots, v_{i-1}\}$. Note that

$$e(v_i, T') = e(v_i, T_i) + e(v_i, T' \setminus T_i).$$

Moreover, when v_i changes its label then all vertices in $V_{\ell_i} \setminus T_i$ still have label ℓ_i. Hence, v_i can only change its label if $e(v_i, T_i) \geq e(v_i, V_{\ell_i} \setminus T_i) + 1$, where the "+1" comes from the fact that the vertex v considers its own label as well when taking the majority. Hence,

$$\begin{aligned} e(v_i, T') &= e(v_i, T_i) + e(v_i, T' \setminus T_i) \\ &\geq e(v_i, V_{\ell_i} \setminus T_i) + 1 + e(v_i, T' \setminus T_i) \\ &= 1 + e(v_i, V_{\ell_i}) - e(v_i, \{v_1, \ldots, v_{i-1}\} \cap V_{\ell_i}) + e(v_i, T' \setminus T_i). \end{aligned}$$

Now we sum both sides over all $1 \leq i \leq k$. Note that summing over $e(v_i, T')$ yields $e(T' \setminus T, T) + 2e(T' \setminus T)$ since edges in $T' \setminus T$ are counted twice. Likewise, summing over $e(v_i, V_{\ell_i})$ yields $\sum_{\ell \neq \ell_{\max}} (e(S_\ell, V_\ell \setminus S_\ell) + 2e(S_\ell))$, and summing over $e(v_i, \{v_1, \ldots, v_{i-1}\} \cap V_{\ell_i})$ yields $\sum_{\ell \neq \ell_{\max}} e(S_\ell)$. Finally, summing over $e(v_i, T' \setminus T_i)$ yields $e(T' \setminus T)$. Thus, summing and canceling $e(T' \setminus T)$ yields

$$e(T' \setminus T, T) + e(T' \setminus T) \geq k + \sum_{\ell \in \mathcal{L}, \ell \neq \ell_{\max}} (e(S_\ell, V_\ell \setminus S_\ell) + e(S_\ell)),$$

which implies (1) as $k = |T' \setminus T|$.

Note that the term $e(T' \setminus T, T) + e(T' \setminus T)$ on the left hand side counts the number of edges by which the label class of $\ell_{\max}$ increases when it grows from T to T', while the term $e(S_\ell, V_\ell \setminus S_\ell) + e(S_\ell)$ counts the number of edges within V_ℓ which have at least one endpoint in S_ℓ. So basically (1) says that in order to recruit k vertices, the label class is "charged" at least $k + \sum_{\ell \neq \ell_{\max}} e(S_\ell, V_\ell \setminus S_\ell) + e(S_\ell)$ edges. It is easy to check that the minimal ratio of charged edges per recruited vertex is attained if $S_\ell = V_\ell$ is of size 2 for all $\ell \neq \ell_{\max}$, in which case the ratio is $3/2$ (three edges for two vertices).

Hence, in order to take over a set S of size k we need at least $3k/2$ edges in $S \cup V_{\ell_{\max}}$. However, a sparse $G_{n,p}$ does not have sets of this density of order $\Theta(n/d^3)$, as the following lemma shows.

▶ **Lemma 14** (Lemma 4.2 in [11]). *Consider $G_{n,p}$ with $1 \ll d = np = o(n)$. Let $\beta = \frac{3}{2} - \frac{1}{2\log(d)}$, and set $s = \frac{n}{3d^3}$. Then with high probability no set of s vertices spans at least βs edges.*

We may thus conclude the proof by contradiction as follows. Assume that $\ell_{\max}$ would take over the graph. After round 100, it has size $s_0 = \tilde{O}(n/d^4)$, so at some later point the label class must have size $s = n/(2d^3)$. At this point, the number of edges in the label class is at least $\frac{3}{2}(s - s_0) = (1 - \tilde{O}(1/d))\frac{3}{2}s \geq \beta s$, where β is as in Lemma 14. This is a contradiction to Lemma 14. Hence, the assumption must be wrong and $\ell_{\max}$ cannot take over the graph. In fact, the proof shows that the label class of $\ell_{\max}$ cannot grow to any size larger than $O(n/d^3)$.

7 Conclusion

We have shown that Max-LPA does not reach consensus on $G_{n,p}$ if $p = O(n^{-1+\varepsilon})$. Consequently it fails to identify communities in planted network models. This disproves a conjecture by Kothapalli, Pemmarajum, and Sardeshmukh. Our result is obtained by combining a careful local analysis of the process with suitable global properties of the network.

For the Max-LPA process, it is natural to assume that there is some threshold α such that for any small $\delta > 0$ we have that for $p = \Omega(n^{-\alpha+\delta})$ the Max-LPA process reaches consensus on $G_{n,p}$ with high probability, while for $p = O(n^{-\alpha-\delta})$ it does not reach consensus with high probability. Assuming such an α exists, it follows from our result that $\alpha \leq 1 - \varepsilon$; from [17] we know that $\alpha \geq 1/4$.

We conducted some experiments, which seem to suggest $\alpha = 4/5$. Our experimental data was obtained by doing a binary search where in every step we ran the algorithm on 32 independent $G_{n,p}$. If the majority of the runs converged with a unique label (modulo isolated vertices) then we decreased the value of p for the following run, otherwise we increased it. We stopped when the change in the probability was small enough. To visualize the data we plot it on a log-log-scale, with basis 2 for the log. In this setting the exponent becomes a linear factor. We computed a linear regression of the log-log-data (i.e., the line which minimizes the sum of the square-distances of the log-log data points), and obtained the line $\log_2(d) = 0.19964 \log_2(n) + 0.4652$. Since the leading constant is very close to 0.2, this suggests that the correct threshold might be $d = \Theta(n^{1/5})$ and thus $p = \Theta(n^{-4/5})$.

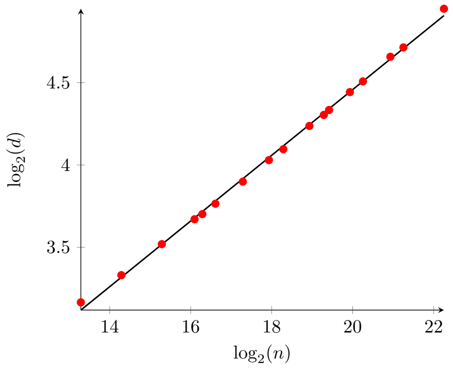

Figure 4 Our experimental data with a linear regression. We plot an experimental evaluation of the threshold d such that the Max-LPA converges in 50% of the cases with a unique label, on a log-log scale. The experimental data is extremely well described by a line with slope ≈ 0.2, suggesting that the threshold satisfies $d = \Theta(n^{1/5})$.

References

1 Michael J Barber and John W Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2):026129, 2009.

2 Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, Riccardo Silvestri, and Luca Trevisan. Simple dynamics for plurality consensus. *Distributed Computing*, 30(4):293–306, 2017.

3 Luca Becchetti, Andrea Clementi, Emanuele Natale, Francesco Pasquale, and Luca Trevisan. Stabilizing consensus with many opinions. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 620–635. SIAM, 2016.

4 Punam Bedi and Chhavi Sharma. Community detection in social networks. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(3):115–135, 2016.

5 Petra Berenbrink, Andrea Clementi, Robert Elsässer, Peter Kling, Frederik Mallmann-Trenn, and Emanuele Natale. Ignore or comply? on breaking symmetry in consensus. *arXiv preprint*, 2017. `arXiv:1702.04921`.

6 Andrea Clementi, Miriam Di Ianni, Giorgio Gambosi, Emanuele Natale, and Riccardo Silvestri. Distributed Community Detection in Dynamic Graphs. *Theor. Comput. Sci.*, 584(C):19–41, June 2015. `doi:10.1016/j.tcs.2014.11.026`.

7 Colin Cooper, Tomasz Radzik, Nicolás Rivera, and Takeharu Shiraga. Fast plurality consensus in regular expanders. *arXiv preprint*, 2016. `arXiv:1605.08403`.

8 Gennaro Cordasco and Luisa Gargano. Label propagation algorithm: a semi-synchronous approach. *International Journal of Social Network Mining*, 1(1):3–26, 2012.

9 Emilio Cruciani, Emanuele Natale, and Giacomo Scornavacca. On the Metastability of Quadratic Majority Dynamics on Clustered Graphs and its Biological Implications. *CoRR*, abs/1805.01406, 2018. `arXiv:1805.01406`.

10 Robert Elsässer, Tom Friedetzky, Dominik Kaaser, Frederik Mallmann-Trenn, and Horst Trinker. Efficient k-party voting with two choices. *ArXiv e-prints*, 2016.

11 Uriel Feige, Michael Krivelevich, and Daniel Reichman. Contagious sets in random graphs. *The Annals of Applied Probability*, 27(5):2675–2697, 2017.

12 Mohsen Ghaffari and Johannes Lengler. Nearly-tight analysis for 2-choice and 3-majority consensus dynamics. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pages 305–313. ACM, 2018.

13 Mohsen Ghaffari and Merav Parter. A polylogarithmic gossip algorithm for plurality consensus. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 117–126. ACM, 2016.

14 Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.

15 Steve Harenberg, Gonzalo Bello, L Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439, 2014.

16 S. Janson, T. Łuczak, and A. Rucinski. *Random graphs*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

17 Kishore Kothapalli, Sriram V. Pemmaraju, and Vivek Sardeshmukh. On the Analysis of a Label Propagation Algorithm for Community Detection. In *Distributed Computing and Networking*, pages 255–269, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

18 Ian XY Leung, Pan Hui, Pietro Lio, and Jon Crowcroft. Towards real-time community detection in large networks. *Physical Review E*, 79(6):066107, 2009.

19 Elchanan Mossel and Omer Tamuz. Opinion exchange dynamics. *Probability Surveys*, 14:155–204, 2017.

20 Svatopluk Poljak and Miroslav Sŭra. On periodical behaviour in societies with symmetric influences. *Combinatorica*, 3(1):119–121, 1983.

21 Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

22 Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

23 Zhao Yang, René Algesheimer, and Claudio J Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750, 2016.