Consistency Thresholds for the Planted Bisection Model

[Extended Abstract]

Elchanan Mossel†
University of Pennsylvania and University of California, Berkeley
400 Jon M. Huntsman Hall
3730 Walnut Street
Philadelphia, PA 19104-6340
mossel@wharton.upenn.edu

Joe Neeman‡
University of Texas, Austin
Mathematics Dept, RLM 8.100
2515 Speedway Stop C1200
Austin, Texas 78712-1202
joeneeman@gmail.com

Allan Sly‡
University of California, Berkeley and the Australian National University
367 Evans Hall
Berkeley, CA 94720
sly@stat.berkeley.edu

ABSTRACT

The planted bisection model is a random graph model in which the nodes are divided into two equal-sized communities and then edges are added randomly in a way that depends on the community membership. We establish necessary and sufficient conditions for the asymptotic recoverability of the planted bisection in this model. When the bisection is asymptotically recoverable, we give an efficient algorithm that successfully recovers it. We also show that the planted bisection is recoverable asymptotically if and only if with high probability every node belongs to the same community as the majority of its neighbors.

Our algorithm for finding the planted bisection runs in time almost linear in the number of edges. It has three stages: spectral clustering to compute an initial guess, a “replica” stage to get almost every vertex correct, and then some simple local moves to finish the job. An independent work by Abbe, Bandeira, and Hall establishes similar results but only in the sparse case where \( p_n, q_n = O(\log n/n) \).

Categories and Subject Descriptors

[Theory of computation]: Random network models

: [Mathematics of computing]: Maximum likelihood estimation

‡A full version of this paper is available at arxiv.org/abs/1407.1591
†Supported by NSF grants DMS 1106999 and CCF 1320105, by DOD ONR grant N00014-14-1-0823, and by grant 328025 from the Simons Foundation.
‡Supported by grant 197982 from the Simons Foundation, and by DTRA grant HDTRA1-13-1-0024.
§Supported by an Alfred Sloan Fellowship and NSF grants DMS-1208338 and DMS-1352013.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

STOC’15, June 14–17, 2015, Portland, Oregon, USA.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3536-2/15/06 ...$15.00.
http://dx.doi.org/10.1145/2746539.2746603.

1. INTRODUCTION

The “planted bisection model” is a random graph model with \( 2n \) vertices that are divided into two classes with \( n \) vertices each. Edges within the classes are added to the graph independently with probability \( p_n \), while edges between the classes are added with probability \( q_n \). Following Bui et al, [5] who studied a very similar model, Dyer and Frieze [9] introduced the planted bisection model in order to study the average-case complexity of the Min-Bisection problem, which asks for a bisection of a graph that cuts the smallest possible number of edges. This problem is known to be NP-complete in the worst case [13], but on a random graph model with a “planted” small bisection one might hope that it is usually easy. Indeed, Dyer and Frieze showed that if \( p_n = p > q = q_n \) are fixed as \( n \to \infty \) then with high probability the bisection that separates the two classes is the minimum bisection, and it can be found in expected \( O(n^3) \) time.

These models were introduced slightly earlier in the statistics literature [11] (under the name “stochastic block model”) in order to study the problem of community detection in random graphs. Here, the two parts of the bisection are interpreted as latent “communities” in a network, and the goal is to identify them from the observed graph structure. If \( p_n > q_n \), the maximum a posteriori estimate of the true communities is exactly the same as the minimum bisection (see the discussion leading to Lemma 4.1), and so the community detection problem on a stochastic block model is exactly the same as the Min-Bisection problem on a planted bisection model; hence, we will use the statistical and computer science terminologies interchangeably. We note, however, the statistics literature is slightly more general, in the sense that it often allows \( q_n > p_n \), and sometimes relaxes the problem by allowing the detected communities to contain some errors.

Later work improved the algorithmic guarantees in two ways: by improving the running time, and by extending
the range of \( p_n \) and \( q_n \) for which the planted bisection can be recovered. (Various results also generalized the problem by allowing more than two labels, but we will ignore this generalization here.) For example, Jerrum and Sorkin [12] required \( p_n - q_n = \Omega(n^{-1/6+\epsilon}) \), while Condon and Karp improved this to \( p_n - q_n = \Omega(n^{-1/2+\epsilon}) \). McSherry [19] made a big step by showing that if
\[
\frac{p_n - q_n}{q_n} \geq C \sqrt{n \log n},
\]
for a large enough constant \( C \) then spectral methods can exactly recover the labels. This was significant because it allows \( p_n \) and \( q_n \) to be as small as \( O(n^{-1} \log n) \), which is order-wise the smallest possible since it also corresponds to the connectivity threshold. A similar result for a slightly different random graph model had been claimed earlier by Boppana [4], but the proof was incomplete. Carson and Impagliazzo [6] showed that with slightly worse poly-logarithmic factors, a simple hill-climbing algorithm also works. Analogous results were later obtained by Bickel and Chen [3] using an algorithm based on modularity maximization (for which no efficient algorithm is known).

It is instructive to keep the example \( p_n = 1/2, q_n = 1/2 - r_n \) in mind. In this case McSherry’s condition is the same as requiring that \( r_n \geq C' \sqrt{n \log n} \). On the other hand, Carson and Impagliazzo [6] pointed out that if \( r_n \leq c \sqrt{n \log n} \) for some small constant \( c \), then the minimum bisection no longer coincides with the planted bisection. From a statistical point of view, this means that the true communities can no longer be reconstructed perfectly. Our main contribution is to close the gap between McSherry’s sufficient condition and Carson-Impagliazzo’s necessary condition. In the above case, for example, we show that the critical constant is \( C = c = 1 \).

2. DEFINITIONS AND RESULTS

**Definition 2.1 (Planted Bisection Model).** Given \( n \in \mathbb{N} \) and \( p, q \in [0, 1] \), we define the random \( 2n \)-node labelled graph \( (G, \sigma) \sim G(2n, p, q) \) as follows: first, choose a balanced labelling \( \sigma \) uniformly at random from \( \{ \tau \in \{1, -1\}^{V(G)} : \sum_{u} \tau_u = 0 \} \). Then, for every distinct pair \( u, v \in V(G) \) independently, add an edge between \( u \) and \( v \) with probability \( p \) if \( \sigma_u = \sigma_v \), and with probability \( q \) if \( \sigma_u \neq \sigma_v \).

The oldest and most fundamental question about stochastic block models is the label reconstruction problem: if we were given the graph \( G \) but not the labelling \( \sigma \), could we reconstruct \( \sigma \) (up to its sign) from \( G \)? This problem is usually framed in asymptotic regime, where the number of nodes \( n \to \infty \), and \( p \) and \( q \) are allowed to depend on \( n \).

**Definition 2.2 (Strong Consistency).** Given two sequences \( p_n \) and \( q_n \) in \([0, 1]\), and given a map \( A \) from graphs to vertex labelings, we say that \( A \) is consistent if
\[
\Pr(A(G_n) = \sigma_n) \text{ or } A(G_n) = -\sigma_n \to 1,
\]
where the probability is taken with respect to \( (G_n, \sigma_n) \sim G(2n, p_n, q_n) \).

Depending on the application, it may also make sense to ask for a labelling which is almost completely accurate, in the sense that it correctly labels all but a vanishingly small fraction of nodes. Amini et al. [2] suggested the term “weak consistency” for this notion.

**Definition 2.3 (Weak Consistency).** Given any \( \sigma, \tau \in \{1, -1\}^{2n} \), define
\[
\Delta(\sigma, \tau) = 1 - \frac{1}{2n} \sum_{i=1}^{2n} \sigma_i \tau_i.
\]
Given sequences \( p_n \) and \( q_n \) in \([0, 1]\), and given a map \( A \) from graphs to vertex labelings, we say that \( A \) is weakly consistent if
\[
\Delta(\sigma_n, A(G_n)) \to 0,
\]
where the probability is taken with respect to \( (G_n, \sigma_n) \sim G(2n, p_n, q_n) \).

Our main result is a characterization of the sequences \( p_n \) and \( q_n \) for which consistent or weakly consistent estimators exist. Note that the characterization of weak consistency was obtained previously by Yun and Proutiere [26], but we include it here for completeness.

**Definition 2.4.** Given \( m, n, p, q \), let
\[
X \sim \text{Binom}(m, \max\{p, q\}) \quad Y \sim \text{Binom}(n, \min\{p, q\})
\]
be independent. We define
\[
P(m, n, p, q) = \Pr(Y \geq X).
\]
When \( m = n \), we will abbreviate by \( P(n, p, q) = P(n, n, p, q) \).

**Theorem 2.5 (Characterization of Consistency).** Consider sequences \( p_n \) and \( q_n \) in \([0, 1]\). There is a strongly consistent estimator for \( G(2n, p_n, q_n) \) if and only if
\[
P(n, p_n, q_n) = o(n^{-1}).
\]
There is a weakly consistent estimator for \( G(2n, p_n, q_n) \) if and only if
\[
P(n, p_n, q_n) \to 0.
\]
In order to provide some intuition for Definition 2.4 and its appearance in our characterization, we note the following graph-theoretic interpretation of \( P(n, p, q) \).

**Definition 2.6.** Given a labelled graph \( (G, \sigma) \) and a node \( v \in V(G) \), we say that \( v \) has a majority of size \( k \) if either
\[
p > q \text{ and } \#\{u \sim v : \sigma_u = \sigma_v\} \geq \#\{u \sim v : \sigma_u \neq \sigma_v\} + k
\]
or
\[
p < q \text{ and } \#\{u \sim v : \sigma_u \neq \sigma_v\} \geq \#\{u \sim v : \sigma_u = \sigma_v\} + k.
\]
We say that \( v \) has a majority if it has a majority of size one. If \( v \) does not have a majority, we say that it has a minority.

**Proposition 2.7.** Fix sequences \( p_n \) and \( q_n \) in \([0, 1]\) and let \( (G_n, \sigma_n) \sim G(n, p_n, q_n) \). Then
\[
\bullet P(n, p_n, q_n) = o(n^{-1}) \text{ if and only if a.a.s. } \forall v \in V(G_n) \text{ has a majority; and}
\]
\[
\bullet P(n, p_n, q_n) \to 0 \text{ if and only if a.a.s. at most } o(n) \text{ nodes in } V(G_n) \text{ fail to have a majority.}
\]
Proposition 2.7 suggests some intuition for Theorem 2.5: namely, that a node can be labelled correctly if and only if it has a majority. In fact, having a majority is necessary for correct labelling (and we will use this to prove one direction of Theorem 2.5); however, it is not sufficient. For example, there are regimes in which 51% of nodes have majorities, but only 99% of them can be correctly labelled (see [21]).

We note that Theorem 2.5 has certain parallels with local-to-global threshold phenomena in random graphs. For example, Erdős and Rényi showed [10] that for \( G(n,p) \), if \( p \) is large enough so that with high probability every node has a neighbor then the graph is connected with high probability. On the other hand, every node having a neighbor is clearly necessary for the graph to be connected. An analogous story holds for the existence of Hamiltonian cycles: Komlós and Szemerédi [14] showed that \( G(n,p) \) has a Hamiltonian cycle with high probability if and only if with high probability every node has degree at least two.

These results on connectedness and Hamiltonicity have a feature in common: in both cases, an obviously necessary local condition turns out to also be sufficient (on random graphs) for a global condition. One can interpret Theorem 2.5 similarly: the minimum bisection in \( G(n,p_n,q_n) \) equals the planted bisection with high probability if and only if with high probability every node has more neighbors of its own label than those of the other label.

### 2.1 The algorithm

In order to prove the positive direction of Theorem 2.5, we provide an algorithm that recovers the planted bisection with high probability whenever \( P(n,p_n,q_n) = o(n^{-1}) \). Moreover, this algorithm runs in time \( \tilde{O}(n^2(p_n + q_n)) \), where \( \tilde{O} \) hides polylogarithmic factors. That is, it runs in time that is almost linear in the number of edges. In addition, we remark that the algorithm does not need to know \( p_n \) and \( q_n \). For simplicity, we assume that we know whether \( p_n > q_n \) or vice versa, but this can be checked easily from the data (for example, by checking the sign of the second-largest-in-absolute-value eigenvalue of the adjacency matrix; see Section 3.1).

Our algorithm comes in three steps, each of which is based on an idea that has already appeared in the literature. Our first step is a spectral algorithm, along the lines of those developed by Boppana [4], McSherry [19], and Coja-Oghlan [7]. Yun and Prouix recently made some improvements to (a special case of) Coja-Oghlan’s work, showing that a spectral algorithm can find a bisection with \( o(n) \) errors if \( n\sqrt{p_n - q_n} \to \infty \) (this is substantially weaker than McSherry’s condition, which would require converging to infinity with a rate of at least \( \log n \)).

The second stage of our algorithm is to apply a “replica trick.” We hold out a small subset \( U \) of vertices and run a spectral algorithm on the subgraph induced by \( V \setminus U \). Then we partition \( U \) by examining the edges between \( U \) and \( V \setminus U \). By repeating the process for many subsets \( U \), we dramatically reduce the number of errors made by the spectral algorithm. More importantly, we get extra information about the structure of the errors; for example, we can show that the set of incorrectly-labelled vertices is very poorly connected. Similar ideas are used by Condon and Karp [8], who used successive augmentation to build an initial guess on a subset of vertices, and then used that guess to correctly classify the remaining vertices. The authors [20] also used a similar idea in the \( p_n, q_n = O(n^{-1}) \) regime, with a more complicated replica trick based on belief propagation.

The third step of our algorithm is a hill-climbing algorithm, or a sequence of local improvements. We simply re-label vertices so that they agree with the majority of their neighbors. An iterative version of this procedure was considered in [6], and a randomized version (based on simulated annealing) was studied by Jerrum and Sorkin [12]. Our version has better performance guarantees because we begin our hill-climbing just below the summit: as we will show, we need to re-label only a tiny fraction of the vertices and each of those will be re-labelled only once.

As noted above, none of the ingredients in our algorithm are novel on their own. However, the way that we combine them is new. For example, McSherry [19] used a spectral algorithm with a “clean-up” stage, but his clean-up stage was different from our second and third stages.

### 2.2 Formulas in terms of \( p_n \) and \( q_n \)

Although Theorem 2.5 is not particularly explicit in terms of \( p_n \) and \( q_n \), one can obtain various explicit characterizations in particular regimes (for example, in order to better compare our results with the literature). The computations leading to these estimates are in the full version of the paper. When the degrees are logarithmic we obtain:

**Theorem 2.8.** Suppose that \( p_n = a_n \log n/n \) and \( q_n = b_n \log n/n \), where \( a_n \) and \( b_n \) are \( \Theta(1) \). Then there exists a strongly consistent estimator if and only if

\[
(a_n + b_n - 2\sqrt{a_n b_n} - 1) \log n + \frac{1}{2} \log \log n \to \infty.
\]

In a sufficiently dense regime, we are also able to obtain a characterization of strong consistency:

**Theorem 2.9.** Suppose that \( p_n = \omega(\frac{\log n}{n}) \) and \( q_n = \omega(\frac{\log n}{n}) \) and that \( \lim \sup \max(p_n, q_n) \leq 0.9 \). Let \( \sigma_n = \sqrt{p_n(1 - p_n) + q_n(1 - q_n)} \). Then there exists a strongly consistent estimator if and only if

\[
\sqrt{n\sigma_n} \exp\left(-\frac{(p_n - q_n)^2}{2\sigma_n^2}\right) \to 0.
\]

In the intermediate regime between Theorems 2.8 and 2.9, we were not able to obtain a simple characterization. However, one can fairly easily derive accurate estimates. In particular,

\[
\lim \inf \frac{n(p_n - q_n)^2}{2\sigma_n^2 \log n} > 1
\]

is sufficient, and

\[
\lim \inf \frac{n(p_n - q_n)^2}{2\sigma_n^2 \log n} \geq 1
\]

is necessary, for the existence of a strongly consistent estimator.

The formula for weak consistency is rather simpler:

**Theorem 2.10.** There exists a strongly consistent estimator if and only if

\[
\frac{n(p_n - q_n)}{p_n + q_n} \to \infty.
\]

To prove Theorem 2.10, one only needs to show that the condition above is equivalent to \( P(n, p_n, q_n) \to 0 \). One direction follows from Chebyshev’s inequality, while the other follows from the central limit theorem.
2.3 Parallel independent work

Abbe et al. [1] independently studied the same problem in the logarithmic sparsity regime. They consider \( p_n = (a \log n)/n \) and \( q_n = (b \log n)/n \) for constants \( a \) and \( b \); they show that \((a + b) - 2\sqrt{ab} > 1\) is sufficient for strong consistency and that \((a + b) - 2\sqrt{ab} \geq 1\) is necessary. Note that these are implied by (1), which is more precise. Abbe et al. also consider a semidefinite programming algorithm for recovering the labels; they show that it performs well under slightly stronger assumptions.

2.4 Other related work, and an open problem

Consistency is not the only interesting notion that one can study on the planted partition model. Earlier work by the authors [21, 22] and by Massoulié [18] considered a much weaker notion of recovery: they only asked whether one could find a labelling that was positively correlated with the true labels.

There are also model-free notions of consistency. Kumar and Kannan [15] considered a deterministic spatial clustering problem and showed that if every point is substantially closer to the center of its own cluster than it is to the center of the other cluster then one can exactly reconstruct the clusters. This is in much the same spirit as Theorem 2.5.

There are also model-free notions of consistency. Kumar and Kannan [15] considered a deterministic spatial clustering problem and showed that if every point is substantially closer to the center of its own cluster than it is to the center of the other cluster then one can exactly reconstruct the clusters. This is in much the same spirit as Theorem 2.5.

Mirrored, Makarychev, Makarychev, and Vijayaraghavan [16, 17] proposed semi-random models for planted bisections. These models allow for adversarial noise, and also allow edge distributions that are not independent, but only invariant under permutations. They then give approximation algorithms for MIN-BISECTION, which they prove to work under expansion conditions that hold with high probability for their semi-random model.

We ask whether the techniques developed here could improve the results obtained by Makarychev et al. For example, exact recovery under adversarial noise is clearly impossible, but if the adversary is restricted to adding \( o(n) \) edges, then maybe one can guarantee almost exact recovery.

3. THE ALGORITHM, AND THE PROOF OF CONSISTENCY

The rough idea behind our strongly consistent labelling algorithm is to first run a weakly consistent algorithm and then try to improve it. The natural way to improve an almost-accurate labelling \( \tau \) is to search for nodes \( u \) that have a minority with respect to \( \tau \) and flip their signs. In fact, if the errors in \( \tau \) were independent of the neighbors of \( u \) then this would work quite well: assuming that \( u \) has a decently large majority (which it will, for most \( u \), by Lemma 3.7), then having a labelling \( \tau \) with few errors is like observing each neighbor of \( u \) with a tiny amount of noise. This tiny amount of noise is very unlikely to flip \( u \)'s neighborhood from a majority to a minority. Therefore, choosing \( u \)'s sign to give it a majority is a reasonable approach.

There are two important problems with the argument outlined in the previous paragraph: it requires the errors in \( \tau \) to be independent of \( u \)'s neighbors, and it is only guaranteed to work for those \( u \) that have a sizeable majority (i.e., almost, but not quite, all the nodes in \( G \)). Nevertheless, this procedure is a good starting point and it motivates Algorithm 1. By removing \( u \) from the graph before looking for the almost-accurate labelling \( \tau \), we ensure the required independence properties (as a result, note that we will be dealing with multiple labellings \( \tau \), depending on which nodes we removed before running our almost-accurate labelling algorithm). And although the final labelling we obtain is not guaranteed to be entirely correct, we show that it has very few (i.e., at most \( n^\epsilon \)) errors whereas the initial labelling might have had up to \( o(n) \) errors.

In order to finally produce the correct labelling, we essentially iterate the previous idea: we flip the label of every node that has a minority. We analyze this procedure by noting that in the first step of the algorithm, the errors were confined to a very particular set of nodes (namely, those without a very strong majority). We show that this set of nodes is small and poorly connected, which means that every node in the graph is guaranteed to only have a few neighbors in this bad set. In particular, even nodes with relatively weak majorities cannot be flipped by labelling errors in the bad set. We analyze this procedure in Section 3.3.

3.1 The initial guess

As stated in the introduction, that there exist algorithms for a.a.s. correctly labelling all but \( o(n) \) nodes. Assuming that \( p_n + q_n = \Omega(n^{-1} \log n) \), such an algorithm is easy to describe, and we include it for completeness; indeed, the algorithm we give is essentially folklore, although a nice treatment is given in [23].

Note that the conditional expectation of the adjacency matrix given the labels is \( \frac{p_n + q_n}{2} \mathbf{1} \mathbf{1}^T + \frac{p_n - q_n}{2} \sigma \tau^T \), where \( \sigma \in \{\pm 1\}^{2n} \) is the true vector of class labels. Now, let \( A \) be the adjacency matrix of \( G \). Then \( \tau \) is the second eigenvector of \( E[A \mid \sigma] \), and its eigenvalue is \( \frac{p_n + q_n}{2} \). In particular, if we had access to \( E[A \mid \sigma] \) then we could recover the labels exactly, simply by looking at its second eigenvector. Instead, we have access only to \( A \). However, if \( A \) and \( E[A \mid \sigma] \) are close then we can recover the labels by rounding the second eigenvector of \( A \).

Conditioned on \( \sigma \), \( A - E[A \mid \sigma] \) is a symmetric matrix whose upper diagonal consists of independent entries, and so we can use results from random matrix theory (e.g. [25, 24]) to bound its norm:

\[
\text{Theorem 3.1. If } p_n + q_n = \Omega(n^{-1} \log n) \text{ then there is a constant } C \text{ such that }
\| A - E[A \mid \sigma] \| \leq C \sqrt{n} \bigl( p_n + q_n \bigr)
\]
\( \text{a.a.s. as } n \to \infty. \)

Assuming Theorem 3.1, note that if
\[
|p_n - q_n| / \sqrt{n} (p_n + q_n) \to \infty
\]
then \( \| A - E[A \mid \sigma] \| \) is order-wise smaller than the second eigenvalue of \( A \). By the Davis-Kahan theorem, it is possible to recover \( x \) up to an error of size \( o(1)\| x \| \). In other words, we can recover the labels of all but \( o(n) \) vertices.

3.2 The replica step

Let BBPartition be an algorithm that is guaranteed to label all but \( o(n) \) nodes correctly; we will use it as a black box. Note that we may assume that BBPartition produces an exactly balanced labelling. If not, then if its output has more \( + \) labels than \(-\) labels, say, we can randomly choose some \(+\)-labelled vertices and relabel them. The new labelling is balanced, and it is still guaranteed to have at most \( o(n) \) mistakes.
We define \( V_e \) to be a set of “bad” nodes that our first step is not required to label correctly.

**Definition 3.2.** Let \( V_e \) be the elements of \( V \) that have a majority of size less than \( \sqrt{(p_u + q_u)n \log n} \) or that have more than \( 100(p_u + q_u) \) neighbors.

**Proposition 3.3.** For every \( \epsilon > 0 \), Algorithm 1 a.a.s. correctly labels every node in \( V \setminus V_e \).

**Algorithm 1: Algorithm for initial accuracy boost**

<table>
<thead>
<tr>
<th>input : graph ( G ), parameter ( \epsilon &gt; 0 )</th>
<th>output: a partition ( W_+, W_- ) of ( V(G) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( W_+ ) ← ( \emptyset ); ( W_- ) ← ( \emptyset );</td>
<td>2. ( U, V ) ← ( \text{BBPartition}(G) );</td>
</tr>
<tr>
<td>3. choose ( m ) so ( (1 - 1/m)\epsilon - 32m^{-1/2} \geq \epsilon/2 );</td>
<td>for ( i ) ← 1 to ( m ) do</td>
</tr>
<tr>
<td>4. partition ( V(G) ) randomly into ( U_1, \ldots, U_m );</td>
<td>if ( U_i \cup W_+ \geq n/2 ) then</td>
</tr>
<tr>
<td>5. ( U_i, U_j ) ← ( \text{BBPartition}(G \setminus U_i) );</td>
<td>swap ( U_i, U_j );</td>
</tr>
<tr>
<td>for ( v ) ∈ ( U ) do</td>
<td>end</td>
</tr>
<tr>
<td>if ( p &gt; q ) and</td>
<td>for ( v ) ∈ ( U_i ) do</td>
</tr>
<tr>
<td># { { u \in U_i + : u \sim v } &gt; # { u \in U_i - : u \sim v } }</td>
<td>if ( p &gt; q ) and</td>
</tr>
<tr>
<td>then</td>
<td># { { u \in U_i + : u \sim v } &lt; # { u \in U_i - : u \sim v } }</td>
</tr>
<tr>
<td>( W_+ ) ← ( W_+ \cup { v } );</td>
<td>then</td>
</tr>
<tr>
<td>else if ( p &lt; q ) and</td>
<td>( W_- ) ← ( W_- \cup { v } );</td>
</tr>
<tr>
<td># { { u \in U_i + : u \sim v } &lt; # { u \in U_i - : u \sim v } }</td>
<td>else</td>
</tr>
<tr>
<td>then</td>
<td>( W_+ ) ← ( W_+ \cup { v } );</td>
</tr>
<tr>
<td>( W_- ) ← ( W_- \cup { v } );</td>
<td>end</td>
</tr>
</tbody>
</table>

The meat of Proposition 3.3’s proof is the following: suppose that \( v \in U_i \) has a strong majority, in the sense that it belongs to \( V \setminus V_e \). Since \( U_i \) is a uniformly random set, it’s very likely that even after removing \( U_i \) from \( G \), \( v \) still has a strong majority among the remaining vertices. Since \( \text{BBPartition}(G \setminus U_i) \) depends only on edges within \( G \setminus U_i \), the edges from \( v \) to \( G \setminus U_i \) are independent of the labelling produced by \( \text{BBPartition}(G \setminus U_i) \). In particular, the mistakes among \( v \)'s neighbors are rare, independent, and unbiased. The chance that these small unbiased mistakes will add up to overturn \( v \)'s large majority is therefore very small, and so we will label \( v \) correctly with high probability. The definition of \( V_e \) was designed so that we can then take a union bound over all \( v \in V_e \). The full proof is contained in the full version of the paper.

### 3.3 The hill-climb step

After running Algorithm 1, we are left with a graph in which only nodes belonging to \( V_e \) could possibly be mislabelled. Fortunately, very few nodes belong to \( V_e \), and those that do are poorly connected to the rest of the graph. This is the content of the next three propositions, whose full proofs are contained in the full version.

**Proposition 3.4.** For every \( \delta > 0 \) there exists an \( \epsilon > 0 \) such that if \( P(n, p_u, q_u) = o(n^{-1}) \) then \( |V_e| \leq n^\delta \) a.a.s.

**Proposition 3.5.** Suppose that \( n(p_u + q_u) \leq n^{1/4} \). For sufficiently small \( \epsilon \), a.a.s. no node has two or more neighbors in \( V_e \).

**Proposition 3.6.** Suppose that \( n(p_u + q_u) \leq n^{1/4} \). For sufficiently small \( \epsilon \), a.a.s. no two nodes in \( V_e \) are adjacent.

The main technical tool in the proof of the preceding propositions is the following lemma, whose proof is in the full version of the paper. Its primary use is for deducing statements like the following: if the parameters are such that a.a.s. every vertex has a majority, then the parameters are such that for any constant \( C \), a.a.s. every vertex has a majority of at least \( C \).

**Lemma 3.7.** Take \( X \) ~ Binom(\( n, p \)) and \( Y \) ~ Binom(\( n, q \)). There are constants \( C, c > 0 \) such that if \( (n(p + q) \geq 64 log n \) and \( |\ell| \leq \sqrt{n(p + q) log n} \) then

\[
Pr(X \geq X) \exp \left( C \sqrt{\frac{\log n}{n(p + q)}} \right) \leq Pr(X \geq X - \ell) \leq Pr(Y \geq X) \exp \left( C \sqrt{\frac{\log n}{n(p + q)}} \right) + 2n^{-2}.
\]

Assuming Lemma 3.7, we will briefly sketch the proofs of Propositions 3.4 through 3.6. Proposition 3.4 follows by setting \( \ell = \epsilon \sqrt{(p_u + q_u)n \log n} \) in Lemma 3.7: this shows that for a fixed \( \epsilon \), \( Pr(v \in V_e) = o(n^{-1 + O(\epsilon)}) \). Hence, \( E[|V_e|] \leq n^{O(\epsilon)} \), which implies Proposition 3.4.

To motivate the proofs of Propositions 3.5 and 3.6, fix a node \( u \) and let us first assume (false) that \( \{ v \sim u \} \) and \( \{ v \in V_e \} \) are independent. Then \( u \) has at most \( O(n^{1/4}) \) neighbors and each of them (by Proposition 3.4) belongs to \( V_e \) with probability at most \( n^{-3-1} \). To prove Proposition 3.6, note that the probability of \( u \) having two neighbors that both belong to \( V_e \) is \( O((n^{1/4+2(3-1)})^{-1}) = o(n^{-1}) \) if \( \delta \) is small enough; then a union bound implies Proposition 3.6. To prove Proposition 3.6, note that the probability of \( u \in V_e \) and a neighbor of \( u \) in \( V_e \) is also \( O(n^{1/4+2(3-1)}) = o(n^{-1}) \). In order to make the previous argument valid, we need to study probabilities of the form \( Pr(v \in V_e | v \sim u) \) instead of blindly assuming independence. But such a probabilities can be controlled by Lemma 3.7 (for example, to get an upper bound on the probability above, take \( \ell = \epsilon \sqrt{(p_u + q_u)n \log n} - 1 \) in Lemma 3.7).
Propositions 3.4 through 3.6 imply that one can recover the true partition by simple local improvements: whenever the predicted label of a node differs from the majority label of its neighbors, relabel it to agree with the majority of its neighbors (assuming $p > q$, that is; otherwise, make each node agree with the minority of its neighbors). Algorithms that make use of this kind of local improvement are known as hill-climbing algorithms, and were applied to the planted partition model in [6]. Due to the strong structure present in $V_*$, our hill-climbing algorithm takes an even simpler form: we can make all the local improvements in one step.

**Algorithm 2:** Algorithm for final labelling

1. **input:** graph $G$, an initial partition $U_+, U_-$ of $V(G)$
2. **output:** a partition $W_+, W_-$ of $V(G)$
   
   $W_+ \leftarrow \{ v \in V(G) : v$ has more neighbors in $U_+$ than in $U_- \}$;
   $W_- \leftarrow V(G) \setminus W_+$

### Proposition 3.8
Suppose that we initialize Algorithm 2 with a partition whose errors are restricted to $V_*$, and suppose that $P(n, p_*, q_*) = o(n^{-1})$. Then a.a.s., Algorithm 2 returns the true partition.

**Proof of Proposition 3.8.** We consider three cases: the dense regime $np \geq n^{1/4}$, the intermediate regime $100 \log n \leq np \leq n^{1/4}$, and the sparse regime $\frac{1}{2} \log n \leq np \leq 100 \log n$.

In the dense regime $np \geq n^{1/4}$, note that by Lemma 3.7, a.a.s. every node has a majority of $\Omega((np/\log n)^{1/3}) \geq \Omega(n^{1/8})$.

On the other hand, if $\epsilon$ is sufficiently small then (by Proposition 3.4) $|V_\epsilon| \leq n^{1/10}$, which implies that every node in $V_\epsilon$ will have most of its neighbors in $U_\epsilon$.

In the intermediate regime $100 \log n \leq np \leq n^{1/4}$, Lemma 3.7 implies that a.a.s. every node has a majority of size three. On the other hand, Proposition 3.5 implies that a.a.s. every node has at most two neighbors in $V_\epsilon$, which again implies that every node in $V_\epsilon$ will have most of its neighbors in $U_\epsilon$.

The sparse regime $\frac{1}{2} \log n \leq np \leq 100 \log n$ is only slightly more complicated. Let $V'$ be the set of nodes with a majority of less than three. By the same argument as the last paragraph, all nodes outside $V'$ are correctly labelled by the algorithm. On the other hand, Proposition 3.6 shows that nodes in $V'$ are also correctly labelled, since none of them have any neighbors in $V_\epsilon$ (recalling that $V' \subset V_\epsilon$).

### 4. NECESSARY CONDITIONS FOR CONSISTENCY

In this section, we prove that $P(n, p_*, q_*) = o(n^{-1})$ is necessary for strong consistency. (In the full version of this paper, we also show that $P(n, p_*, q_*) = o(1)$ is necessary for weak consistency.) This is easier than proving sufficiency, and it makes rigorous our earlier statements about failing to label nodes without majorities.

By standard statistical arguments, if we are asked to produce a configuration $\hat{\sigma}$ from the graph $G$, then the algorithm with the highest probability of success is the maximum a posteriori estimator, $\hat{\sigma}$, which is defined to be the $\tau \in \{-1, 1\}^{V(G)}$ satisfying $\sum \tau_u = 0$ that maximizes $Pr(G | \tau = \tau)$. If there are multiple maxima, then we choose among them arbitrarily. In order to prove that $P(n, p_*, q_*) = o(n^{-1})$ is necessary for strong consistency, we relate the success probability of $\hat{\sigma}$ to the existence of nodes with minorities. Note that we say $v$ has a majority with respect to $\tau$ if (assuming $p > q$) $\tau$ gives the same label to $v$ as it does to most of $v$'s neighbors.

**Lemma 4.1.** If there is a unique maximal $\theta$ such that $v$ has a majority with respect to $\theta$, then there cannot be both a $+$-labelled node with a majority and a $-$-labelled node with a minority.

**Proof.** For convenience, we will assume that $p > q$. The same proof works for $p < q$, but one needs to remember that the definition of “majority” and “minority” swap in that case (Definition 2.6).

The probability of $G$ conditioned on the labelling $\tau$ may be written explicitly: if $A_\tau$ is the set of unordered pairs $u \neq v$ with $\tau_u = \tau_v$ and $B_\tau$ is the set of unordered pairs $u \neq v$ with $\tau_u \neq \tau_v$, then

$$
Pr(G | \tau = \tau) = (1 - p)^{|A_\tau|} (1 - q)^{|B_\tau|} \left( \frac{p}{1 - p} \right)^{|E(G) \cap A_\tau|} \left( \frac{q}{1 - q} \right)^{|E(G) \cap B_\tau|}.
$$

Consider a labelling $\tau$. Suppose that there exist nodes $u$ and $v$ with $\tau_u = +$ and $\tau_v = -$, and such that both $u$ and $v$ have minorities with respect to $\tau$. By swapping the labels of $u$ and $v$, $|A_\tau|$ and $|B_\tau|$ are unchanged, while $|E(G) \cap A_\tau|$ increases and $|E(G) \cap B_\tau|$ decreases. Since $p > q$, this increases $Pr(G | \sigma = \tau)$ and so the original labelling $\tau$ cannot have been the unique maximizer of $Pr(G | \sigma = \tau)$.

In order to argue that $P(n, p_*, q_*) = o(n^{-1})$ is necessary for strong consistency, we need to show that if $P(n, p_*, q_*)$ is not $o(n^{-1})$ then $(G, \sigma) \sim G(2n, p_*, q_*)$ has a non-vanishing chance of containing nodes of both labels with minorities.

Suppose that $P(n, p_*, q_*)$ is not $o(n^{-1})$. By Proposition 2.7, there is some $\epsilon > 0$ such that for infinitely many $n$, $Pr(\exists u : u$ has a minority $) \geq \epsilon$. Since $+$-labelled nodes and $-$-labelled nodes are symmetric, there are infinitely many $n$ such that

$$
Pr(\exists u : \sigma_u = + \text{ and } u \text{ has a minority}) \geq \epsilon/2
$$

$$
Pr(\exists v : \sigma_v = - \text{ and } v \text{ has a minority}) \geq \epsilon/2.
$$

Now, the two events above are non-negatively correlated because both of them are monotonic events with the same directions: both are monotonic increasing in the edges between $+$-labelled and $-$-labelled nodes and monotonic decreasing in the other edges. Hence, there are infinitely many $n$ for which

$$
Pr(\exists u, v : \sigma_u = +, \sigma_v = - \text{ and } u \text{ and } v \text{ have minorities}) \geq \epsilon^2/4.
$$

### 5. REFERENCES


