Belief Propagation for Subgraph Detection with Imperfect Side-information

Arun Kadavankandy, Konstantin Avrachenkov, Laura Cottatellucci, Rajesh Sundaresan

Belief Propagation for Subgraph Detection with Imperfect Side-information

Arun Kadavankandy, Konstantin Avrachenkov, Laura Cottatellucci and Rajesh Sundaresan

***Abstract*— We propose a local message passing algorithm based on Belief Propagation (BP) to detect a small hidden Erdős-Rényi (ER) subgraph embedded in a larger sparse ER random graph in the presence of side-information. We consider side-information in the form of revealed subgraph nodes called cues, some of which may be erroneous. Namely, the revealed nodes may not all belong to the subgraph, and it is not known to the algorithm a priori which cues are correct and which are incorrect. We show that asymptotically as the graph size tends to infinity, the expected fraction of misclassified nodes approaches zero for any positive value of a parameter λ, which represents the effective Signal-to-Noise Ratio of the detection problem. Previous works on subgraph detection using BP without side-information showed that BP fails to recover the subgraph when $\lambda < 1/e$. Our results thus demonstrate the substantial gains in having even a small amount of side-information.**

***Index Terms*— Subgraph detection, Erdos-Renyi, Belief Propagation**

I. Introduction and Related Literature

Hidden community detection in a graph is an important problem with many applications in Information Theory, Signal Processing and Machine Learning. A hidden community could be in the form of a group of densely linked nodes in a large sparser network. Many real-world problems such as fraud detection in auction networks [1] and webgraphs [2] can be formulated as dense subgraph detection problems. The interested reader is referred to [3] for an expository survey.

In this work, we follow a model-based approach, where we model the hidden community as a small Erdős-Rényi (ER) graph of size K with a large edge probability embedded in a larger ER graph of size n with a smaller edge probability. Let p be the edge probability among the subgraph nodes and q be the edge probability outside, such that $p > q$. Henceforth, we denote this random graph by $G(K, n, p, q)$. This model was proposed in [4] to study anomalous transactions in a computer network. It is a special case of the Stochastic Block Model (SBM) extensively studied recently in community detection and also in Information Theory [5]–[7].

The hidden subgraph problem as well as its variations such as hidden clique detection has been considered in several recent works, e.g. [8]–[14]. In [12] the author analyzed the setting where $p = a/n, q = b/n$ and $K = \kappa n$ where a, b, κ are constants independent of n. This is an example of a diluted random graph, i.e., a graph where the average degree is a constant irrespective of the graph size. It was shown in [12] that a Maximum Likelihood (ML) detector achieves vanishing asymptotic probability of error for any positive λ, defined as a function of the graph parameters as below:

$$\lambda = \frac{K^2(p-q)^2}{(n-K)q}. \tag{1}$$

This work was partly funded by the French Government (National Research Agency, ANR) through the "Investments for the Future" Program reference #ANR-11-LABX-0031-01 and Indo-French Centre for Promotion of Advanced Research (IFCPAR/CEFIPRA) Grant No.5100-IT1 "Monte Carlo and Learning Schemes for Network Analytics."

A. Kadavankandy is with Inria, Sophia Antipolis and University of Nice, arun.kadavankandy@inria.fr. K. Avrachenkov is with Inria, Sophia Antipolis, k.avrachenkov@inria.fr. L. Cottatellucci is with Eurecom, France, laura.cottatellucci@eurecom.fr. R.Sundaresan is with the ECE Department and Robert Bosch Centre for Cyber Physical Systems, Indian Institute of Science, Bangalore, India, rajeshs@ece.iisc.ernet.in

This parameter can be interpreted as an effective Signal-to-Noise Ratio (SNR). In [12] the author also analyzed the performance of subgraph detection using Belief Propagation (BP) and showed that BP recovers the subgraph with vanishing error only if $\lambda > 1/e$ and fails otherwise[1]. In [14] the authors analyzed BP in a more general parameter setting where $K = o(n)$ and also proved that BP fails when $\lambda < 1/e$, whereas ML detection succeeds when $\lambda = \Omega(K/n \log(n/K))$. It can be concluded that BP, which is near-linear in computational complexity on diluted graphs, is strictly inferior to ML detection, which has exponential complexity with respect to (w.r.t.) n, for the problem of subgraph detection. The above works on subgraph detection on $G(K, n, p, q)$ do not take into account any form of side-information.

In this work, we study the influence of side-information on the detectability threshold of BP for subgraph detection when this side-information may be erroneous. The side-information we consider is in the form of cues, i.e., we are revealed a few nodes that belong to the hidden community. BP with side-information for community partitioning for two-block and general k-block SBM has been studied in [15], [16]. BP with exact cues for subgraph detection was studied recently in [17]. However, this algorithm does not take into account the possibility that the cues may be erroneous, i.e., some of the cued nodes may be erroneously assigned to the subgraph.

Our Contributions: We develop a BP based subgraph detection algorithm that uses imperfect side-information. The accuracy of cues is quantified by a parameter β, which we define as the expected fraction of correct cues. We then analyse the error performance of this algorithm on $G(K, n, p, q)$ with $p = a/n, q = b/n$, $K = \kappa n$ and derive the asymptotic distributions of the BP messages in the large degree regime where $a, b \to \infty$. Using these distributions, we derive an expression for the asymptotic misclassification rate of the algorithm and show that in the limit when $K/n = \kappa \to 0$, the error rate tends to zero for any $\lambda > 0$ as long as $\beta > 0$. A similar result was shown in [17] for exact cues. Thus, we show that the presence of side-information removes the detectability threshold found in BP without side-information and hence any small amount of side-information greatly improves the performance of the algorithm.

Notation: We denote the cardinality of a set S by $|S|$ and for two sets A, B $A \Delta B$ denotes the set difference given by $(A \cup B) \cap (A \cap B)^c$, where A^c denotes the complement of set A. We denote by Poi(λ) the Poisson distribution with mean λ and by $\mathcal{N}(\mu, \sigma^2)$ the Gaussian distribution with mean μ and variance σ^2. Also, $X \sim f(x)$ means that the random variable (rv) X has distribution $f(x)$. Lastly, $\mathbb{P}(E)$ denotes the probability of an event E and $\mathbb{E}$ denotes the expectation operator.

The rest of the paper is organised as follows. In Section II we introduce the system model. In Section III we present our algorithm and a sketch of its derivation. In Section IV we analyse the asymptotic error rate of the algorithm, in Section V we provide some simulation results, and conclude the paper in Section VI.

[1]Here and in the rest of the paper e represents the base of the natural logarithm

II. System Model

Consider a realization $G = (V, E)$ of $G(K, n, p, q)$ where $V = \{1, 2, \ldots, n\}$ is the set of nodes and $E \subset V \times V$ is the set of edges. Community $S \subset V$ is chosen from V uniformly from all subsets of size K. Then edges are added between any pair of nodes i, j independently of all other pairs such that $\mathbb{P}((i,j) \in E | i, j \in S) = p$ and $\mathbb{P}((i,j) \in E | i \in S, j \notin S) = \mathbb{P}((i,j) \in E | i \notin S, j \notin S) = q$, with $p > q$. In other words, edges between subgraph nodes appear with probability p, and all other edges appear with probability q. We can denote subgraph membership using a function $\sigma : V \to \{0,1\}^n$ such that $\sigma_i = 1$ if $i \in S$ and $\sigma_i = 0$ if $i \notin S$. A subset of nodes is labelled as cues. We use a function $c : V \to \{0,1\}^n$ to denote cue membership, i.e., $c_i = 1$, if i is a cue and $c_i = 0$ otherwise. To describe the side-information, we define two parameters α and β as follows:

$$\alpha = \frac{\mathbb{E}(\sum_{i=1}^{n} \chi_{\{c_i=1\}})}{K} = \frac{n\mathbb{P}(c_i = 1)}{K}, \tag{2}$$

where χ is the indicator function, and

$$\beta = \mathbb{P}(\sigma_i = 1 | c_i = 1), \tag{3}$$

for any node i. Thus, α is the expected fraction of cues w.r.t. the subgraph size and β represents the quality of cues, i.e., how likely a node is to be in a subgraph if it is a cue. We then choose cued nodes such that $\mathbb{P}(c_i = 1 | \sigma_i = 1) = \alpha\beta$ and $\mathbb{P}(c_i = 1 | \sigma_i = 0) = \frac{K\alpha(1-\beta)}{n-K}$ so that (2) and (3) hold.

III. Belief Propagation Algorithm

In this section, we present our algorithm and an outline of its derivation. The optimal algorithm that minimises the average probability of node misclassification, and hence the expected number of misclassified nodes, is the MAP (Maximum a Posteriori Probability) detector [13]. For every node i, this algorithm computes the likelihood function

$$R_i = \log\left(\frac{\mathbb{P}(G, C | \sigma_i = 1)}{\mathbb{P}(G, C | \sigma_i = 0)}\right)$$

and it declares a node as belonging to the subgraph if $R_i > \log\left(\frac{\mathbb{P}(\sigma_i=0)}{\mathbb{P}(\sigma_i=1)}\right)$, where G denotes the graph realization and C is the cue information in the form of the function c. This algorithm would need the knowledge of the entire graph and moreover has exponential complexity for calculating the likelihood function w.r.t. σ_i. Therefore we focus on Belief Propagation, which approximates R_i using the observation of only local neighbourhoods of nodes. In a graph G, we define a t-hop neighbourhood G_i^t of a node i as the set of nodes that can be reached from i by traversing at most t links. Then the set of neighbours of i, which we denote δi, is equal to G_i^1. BP is a recursive algorithm that uses G_i^t and the cue information contained therein, denoted by C_i^t, to compute the following likelihood function at each node locally by aggregating messages sent by its neighbours:

$$R_i^t = \log\left(\frac{\mathbb{P}(G_i^t, c_i, C_i^t | \sigma_i = 1)}{\mathbb{P}(G_i^t, c_i, C_i^t | \sigma_i = 0)}\right). \tag{4}$$

To derive the BP recursions, we exploit the fact that $G(K, n, p, q)$ in a small neighbourhood of a node resembles a tree, i.e., it is locally tree-like. This was formalized in [14, Lemma 15] by means of a coupling formulation between G_i^t and a specially constructed Galton-Watson (G-W) tree T_i^t with Poisson degrees rooted at i. We denote by τ_i the subgraph label of node i on the tree, whereas σ_i is the label on the original graph.

Lemma 1 *[14, Lemma 15] For $G(K, n, p, q)$ with $p = a/n$ and $q = b/n$, if $t = o(log(n))$ there exists a coupling such that $(G_i^t, \sigma^t, C_i^t) = (T_i^t, \tau^t, C_i^t)$ with probability $1 - n^{-1+o(1)}$, where τ^t represents the labels on the tree T_i^t and σ^t denotes the labels on G_i^t.*

The proof proceeds by showing that the probability of having a cycle in G_i^t tends to zero and that the degrees of nodes, which are Bernoulli rvs, converge in total-variation distance to Poisson rvs as $n \to \infty$.

We describe briefly the construction of the tree T_i^t to aid in the understanding of the derivation of our algorithm. The label τ_i of node i is 1 with probability K/n and zero with probability $1 - K/n$. Node i has N_i children, where N_i is distributed as a Poisson rv with mean $d_1 = Kp + (n - K)q = \kappa a + (1 - \kappa)b$ if $\tau_i = 1$ or mean $d_0 = nq = \kappa b$ if $\tau_i = 0$. If $\tau_i = 1$, each child j of node i is assigned a label τ_j such that $\tau_j = 1$ with probability Kp/d_1 and 0 with probability $1 - Kp/d_1$. If on the other hand $\tau_i = 0$, $\tau_j = 1$ with probability Kq/d_0 and $\tau_j = 0$ with probability $1 - Kq/d_0$. Therefore, the number of children of i with label 1 is Poisson distributed with mean Kp if $\tau_i = 1$ and mean Kq if $\tau_i = 0$. Similarly, the number of children with label 0 is Poisson distributed with mean $(n - K)q$. At any level of the tree, a node j is labelled a cue such that

$$\begin{aligned} \mathbb{P}(c_j = 1 | \tau_j = 1) &= \alpha\beta \\ \mathbb{P}(c_j = 1 | \tau_j = 0) &= K\alpha(1 - \beta)/(n - K) \end{aligned} \tag{5}$$

This construction then continues up to depth t. An interesting consequence of the tree coupling is the fact that given the label at node i, the subtrees T_j^{t-1} rooted at the children j are jointly independent. Therefore the likelihood ratios at the subtrees can be computed independently and then transmitted up to node i where they are combined. The tree coupling along with likelihood computation on trees is used by BP to simplify the methodology for calculating the likelihood function R_i^t. Algorithm 1 is the resulting BP algorithm.

Algorithm 1 BP with imperfect cues

1: Initialize: Set $R_{i\to j}^0$ to 0, for all $(i,j) \in E$. Let $t_f < \frac{\log(n)}{\log(np)} + 1$. Set $t = 0$.

2: For all directed pairs $(j,i) \in E$,

$$R_{j\to i}^{t+1} = -K(p - q) + h_j + \sum_{\substack{l \in \delta j, \\ l \neq i}} \log\left(\frac{\exp(R_{l\to j}^t - \nu)(p/q) + 1}{\exp(R_{l\to j}^t - \nu) + 1}\right), \tag{6}$$

where $\nu = \log(\frac{n-K}{K})$ and h_i is defined in (9).

3: Increment t. If $t < t_f - 1$ go back to 2, else go to 4

4: Compute $R_i^{t_f}$ for every $u \in V \backslash C$ as follows:

$$R_i^{t+1} = -K(p - q) + h_i + \sum_{l \in \delta i} \log\left(\frac{\exp(R_{l\to i}^t - \nu)(p/q) + 1}{\exp(R_{l\to i}^t - \nu) + 1}\right) \tag{7}$$

5: Output $\widehat{S}$ as K set of nodes in V with the largest values of $R_u^{t_f}$.

Here we provide a brief sketch of its derivation. Recall the definition of R_i^t given in (4). Assuming that up to depth t the neighbourhood G_i^t has no cycles, i.e., the tree-coupling

holds, we can split the log-likelihood ratio in three parts as follows.

$$R_i^t = \log\left(\frac{\mathbb{P}(G_i^t, C_i^t|\sigma_i=1)}{\mathbb{P}(G_i^t, C_i^t|\sigma_i=0)}\right) + \log\left(\frac{\mathbb{P}(c_i|\sigma_i=1)}{\mathbb{P}(c_i|\sigma_i=0)}\right)$$
$$= \sum_{j\in\delta i}\log\left(\frac{\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_i=1)}{\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_i=0)}\right) +$$
$$\log\left(\frac{\mathbb{P}(N_i|\sigma_i=1)}{\mathbb{P}(N_i|\sigma_i=0)}\right) + h_i. \quad (8)$$

Here we used the independence property of $|\delta i| = N_i$ and the subtrees G_j^{t-1}. The second term in (8) captures the randomness of $|\delta i|$. Also $h_i = \log\left(\frac{\mathbb{P}(c_i|\sigma_i=1)}{\mathbb{P}(c_i|\sigma_i=0)}\right)$ is given by

$$h_i = \begin{cases} \log\left(\frac{\beta(n-K)}{(1-\beta)K}\right) & \text{if } i \in C \\ \log\left(\frac{(1-\alpha\beta)(n-K)}{(n-K-\alpha K+\alpha K\beta)}\right) & \text{otherwise.} \end{cases} \quad (9)$$

This above term captures our faith in the cues; notice that when $\beta = 1$, $h_i = \infty$ if i is a cue. In this case the cues are exact and hence Algorithm 1 becomes the same as the BP algorithm for exact cues from [17]. Observe that we can expand $\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_i=1)$ as below

$$\begin{aligned}
&\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_i=1) \\
&= \mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j, \sigma_j=1|\sigma_i=1) + \\
&\quad \mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j, \sigma_j=0|\sigma_i=1) \\
&= \frac{Kp}{d_1}\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_j=1) + \\
&\quad \frac{(n-K)q}{d_1}\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_j=0)
\end{aligned}$$

where we used the fact that $\mathbb{P}(\sigma_j=1|\sigma_i=1) = Kp/d_1$ and $\mathbb{P}(\sigma_j=0|\sigma_i=1) = (n-K)q/d_1$. Similarly we can expand $\mathbb{P}(G_j^{t-1}\backslash\{u\}, C_j^{t-1}, c_j|\sigma_i=0)$ and defining $R_{j\to i}^{t-1} = \log\left(\frac{\mathbb{P}(G_j^{t-1}\backslash\{u\}, c_j, C_j^{t-1}|\sigma_j=1)}{\mathbb{P}(G_j^{t-1}\backslash\{u\}, c_j, C_j^{t-1}|\sigma_j=0)}\right)$ we obtain the recursion (7) to update the beliefs R_i^t. The recursion to update $R_{j\to i}^t$ can be obtained by repeating the same steps. It can be verified that the computational complexity of this algorithm is $O(t_f|E|)$.

IV. Asymptotic Error Analysis

In this section we analyse the distributions of the messages R_i^t of a node i in the limit where the graph size $n\to\infty$. In our analysis we assume that $p = a/n, q = b/n$ and $K = \kappa n$. We derive the conditional distributions of the messages R_i^t for a finite t given $\sigma_i = 0$ and $\sigma_i = 1$. In this limit the tree assumption holds by Lemma 1. Let F_i^t be such that $R_i^t = F_i^t + h_i$. Then, it can be seen from (7) that F_i^t satisfies the following recursion

$$F_i^{t+1} = -K(p-q) + \sum_{l\in\delta i} f(F_l^t + h_l), \quad (10)$$

where $f(x) = \log\left(\frac{e^{(x-\nu)}(a/b)+1}{e^{(x-\nu)}+1}\right)$. Let Ψ_0^t, Ψ_1^t be the rvs that have the asymptotic conditional distribution of F_i^t given $\sigma_i = 0$ and $\sigma_i = 1$ respectively. Then, by studying the recursion (10) on the G-W tree T_i^t we can conclude that Ψ_0^t, Ψ_1^t satisfy the following recursive distributional equations

$$\begin{aligned}
\Psi_0^{(t+1)} &\stackrel{\mathrm{D}}{=} -\kappa(a-b) + \sum_{i=0}^{L_{01c}} f(\Psi_{1i}^t + B_c) + \\
&\quad \sum_{i=0}^{L_{01n}} f(\Psi_{1i}^t + B_n) + \sum_{i=0}^{L_{00c}} f(\Psi_{0i}^t + B_c) + \\
&\quad \sum_{i=0}^{L_{00n}} f(\Psi_{0i}^t + B_n), \\
\Psi_1^{(t+1)} &\stackrel{\mathrm{D}}{=} -\kappa(a-b) + \sum_{i=0}^{L_{11c}} f(\Psi_{1i}^t + B_c) + \\
&\quad \sum_{i=0}^{L_{11n}} f(\Psi_{1i}^t + B_n) + \sum_{i=0}^{L_{10c}} f(\Psi_{0i}^t + B_c) + \\
&\quad \sum_{i=0}^{L_{10n}} f(\Psi_{0i}^t + B_n),
\end{aligned}$$

where $\stackrel{\mathrm{D}}{=}$ represents equality in distribution, and the random sums are such that $L_{01c} \sim \mathrm{Poi}(\kappa b\alpha\beta)$ is the rv that represents the number of cued children with label 1 of a node with label 0, $L_{01n} \sim \mathrm{Poi}(\kappa b(1-\alpha\beta))$ is number of its uncued children with label 1, $L_{00c} \sim \mathrm{Poi}(\kappa b\alpha(1-\beta))$ is number of cued children with label 0 of a node with label 0 and $L_{00n} \sim \mathrm{Poi}(b(1-\kappa-\kappa\alpha(1-\beta))$ is the number of its uncued children with label 0. Similarly for a node with label 1, $L_{11c} \sim \mathrm{Poi}(\kappa a\alpha\beta)$ represents the number of its cued children with label 1 , $L_{11n} \sim \mathrm{Poi}(\kappa a(1-\alpha\beta))$ represents its uncued children with label 1, $L_{10c} \sim \mathrm{Poi}(\kappa b\alpha(1-\beta))$ represents the number of its cued children with label 0 and finally $L_{10n} \sim \mathrm{Poi}(b(1-\kappa-\kappa\alpha(1-\beta)))$ is the number of its children with label 0 that are not cues. Here we used the tree construction and (5) to derive the means of the Poisson rvs. In addition, $B_c = h_i$ when $c_i = 1$ and $B_n = h_i$ when $c_i = 0$ as given in (9) and $\Psi_{0,i}^t$ and $\Psi_{1,i}^t$ are iid rvs with the same distribution as Ψ_0^t and Ψ_1^t respectively. Based on the above recursions we can analyse the distributions of Ψ_0^t, Ψ_1^t in the large degree regime when $a, b\to\infty$. In the following proposition, we present the asymptotic distributions in this regime.

Proposition 1 *Consider the distribution of BP messages R_i^t when $n\to\infty$. In the high degree regime where $b = qn$ tends to ∞ such that $\lambda = \frac{K^2(p-q)^2}{(n-K)q} = \frac{\kappa^2(a-b)^2}{(1-\kappa)b}$ and $\kappa = K/n$ are held fixed, for any $t > 0$, the random variables Ψ_0^t and Ψ_1^t converge in distribution to Γ_0^t and Γ_1^t, which have Gaussian distributions given as follows*

$$\Gamma_0^t \sim \mathcal{N}(-\mu^{(t)}/2, \mu^{(t)})$$
$$\Gamma_1^t \sim \mathcal{N}(\mu^{(t)}/2, \mu^{(t)}),$$

where $\mu^{(t)}$ satisfies the following recursion with $\mu^{(0)} = 0$

$$\mu^{(t+1)} = \alpha\beta^2\lambda\mathbb{E}\left(\frac{(1-\kappa)/\kappa}{\beta + (1-\beta)e^{(-\mu^{(t)}/2-\sqrt{\mu^{(t)}}Z)}}\right) + (1-\alpha\beta)^2\lambda$$
$$\mathbb{E}\left(\frac{(1-\kappa)}{\kappa(1-\alpha\beta) + (1-\kappa-\alpha\kappa+\alpha\kappa\beta)e^{(-\mu^{(t)}/2-\sqrt{\mu^{(t)}}Z)}}\right), \quad (11)$$

with λ as defined in (1) and the expectation is w.r.t. $Z \sim \mathcal{N}(0,1)$. The messages R_u^t for any node u given $\{\sigma_u = i\}$ for $i \in \{0,1\}$ are asymptotically distributed as $\Gamma_i^t + B_c$ if u is a cue, and $\Gamma_i^t + B_n$ if u is not a cue.

The proof can be found in the technical report [18]. Next, we present our main result on the asymptotic error rate of Algorithm 1.

Theorem 1 *For any $\lambda > 0, \alpha > 0, \beta > 0$,*

$$\lim_{b\to\infty}\lim_{n\to\infty}\frac{\mathbb{E}(|\hat{S}\Delta S|)}{K} \leq 2\Bigg(\alpha\sqrt{\beta(1-\beta)} + \sqrt{(1-\alpha\beta)(\frac{1-\kappa}{\kappa}-\alpha(1-\beta))}\Bigg)e^{-\frac{\lambda\alpha\beta^2(1-\kappa)}{8\kappa}}.$$

Consequently,

$$\lim_{\kappa\to 0}\lim_{b\to\infty}\lim_{n\to\infty}\frac{\mathbb{E}(|S\Delta\widehat{S}|)}{K} = 0.$$

Sketch of the Proof: We only provide a sketch of the proof; the missing details can be found in the technical report [18]. Let $0 < \alpha, \beta < 1$. Using the conditional asymptotic distributions from Proposition 1, we can derive the expected error rate of the following local MAP detector, which uses the local likelihood functions R_i^t. Its output $\widehat{S}_0$ is given as

$$\widehat{S}_0 = \{i : R_i^t > \nu\}, \tag{12}$$

where $\nu = \log\left(\frac{\mathbb{P}(\sigma_i=0)}{\mathbb{P}(\sigma_i=1)}\right) = \log\left(\frac{1-\kappa}{\kappa}\right)$.[2] Note that, in contrast, Algorithm 1 outputs $\widehat{S}$, which is the set of K nodes with K largest values of R_i^t. However the expected number of misclassified nodes $\mathbb{E}(|\widehat{S}\Delta S|)$ can be related to $\mathbb{E}(|\widehat{S}_0\Delta S|)$ as follows

$$\mathbb{E}(|S\Delta\widehat{S}|) \leq 2\mathbb{E}(|S\Delta\widehat{S}_0|). \tag{13}$$

In what follows, we give a bound for $\mathbb{E}(|S\Delta\widehat{S}_0|)$. By analyzing the recursion (11) we can show that for any $t > 0$,

$$\alpha\beta^2\lambda\frac{1-\kappa}{\kappa} \leq \mu^{(t)} \leq \lambda\frac{(1-\kappa)}{\kappa}. \tag{14}$$

Note that the lower bound above is useful only when α and β are non-zero, hence why it is important to have non-trivial side-information. Let $\Gamma_{i,j}^t$ be the rv that has the conditional asymptotic distribution, as $n \to \infty, b \to \infty$, of R_u^t given $\{\sigma_u = i, c_u = j\}$ for $i,j \in \{0,1\}$ for any node u. Then from Proposition 1 we have

$$\begin{aligned}
\Gamma_{0,0}^t &\sim \mathcal{N}\left(-\mu^{(t)}/2 + B_n, \mu^{(t)}\right)\\
\Gamma_{0,1}^t &\sim \mathcal{N}\left(-\mu^{(t)}/2 + B_c, \mu^{(t)}\right)\\
\Gamma_{1,0} &\sim \mathcal{N}\left(\mu^{(t)}/2 + B_n, \mu^{(t)}\right)\\
\Gamma_{1,1}^t &\sim \mathcal{N}\left(\mu^{(t)}/2 + B_c, \mu^{(t)}\right).
\end{aligned}$$

[2]Although the subgraph is uniformly sampled from sets of size K, $\mathbb{P}(\sigma_i = 1) = K/n = 1 - \mathbb{P}(\sigma_i = 0)$ exactly in the limit as $n \to \infty$ by the coupling in Lemma 1.

Let $p_e(u)$ be the average probability of error of the detector in (12) for any node u. It can be written as

$$\begin{aligned}
p_e(u) = {} & p_e(u|\sigma_u = 0, c_u = 0)\mathbb{P}(\sigma_u = 0, c_u = 0) +\\
& p_e(u|\sigma_u = 0, c_u = 1)\mathbb{P}(\sigma_u = 0, c_u = 1)\\
& + p_e(u|\sigma_u = 1, c_u = 0)\mathbb{P}(\sigma_u = 1, c_u = 0)\\
& + p_e(u|\sigma_u = 1, c_u = 1)\mathbb{P}(\sigma_u = 1, c_u = 1),
\end{aligned}$$

where $p_e(u|\sigma_u = 0, c_u = 0)$ denotes the probability that a non-subgraph node u is classified by the algorithm as a subgraph node if it is not a cue, and similarly for the other terms. Consequently,

$$\begin{aligned}
\lim_{b\to\infty}\lim_{n\to\infty} p_e(u) = {} & \mathbb{P}(\Gamma_{0,0}^t > \nu)\pi_{0,0} + \mathbb{P}(\Gamma_{0,1}^t > \nu)\pi_{0,1}\\
& + \mathbb{P}(\Gamma_{1,0}^t < \nu)\pi_{1,0} + \mathbb{P}(\Gamma_{1,1}^t < \nu)\pi_{1,1},
\end{aligned}$$

where $\pi_{i,j} = \mathbb{P}(\sigma_u = i, c_u = j)$ for $i,j \in \{0,1\}$. These probabilities can be computed using (5). Since $p_e(u)$ is the same for any node u, henceforth we denote it as p_e. If $Q(x)$ denotes $\mathbb{P}(Z > x)$ for $Z \sim \mathcal{N}(0,1)$, the expected fraction of mislabelled nodes can be computed as

$$\begin{aligned}
&\lim_{b\to\infty}\lim_{n\to\infty}\frac{\mathbb{E}(|\widehat{S}_0\Delta S|)}{K}\\
&= \lim_{b\to\infty}\lim_{n\to\infty}\frac{np_e}{K}\\
&= Q\left(\frac{\frac{\mu^{(t)}}{2} - \log\left(\frac{(1-\alpha\beta)\kappa}{(1-\kappa-\alpha\kappa+\alpha\kappa\beta)}\right)}{\sqrt{\mu^{(t)}}}\right)\left(\frac{1-\kappa}{\kappa} - \alpha(1-\beta)\right) + Q\left(\frac{\frac{\mu^{(t)}}{2} - \log(\frac{\beta}{1-\beta})}{\sqrt{\mu^{(t)}}}\right)\alpha(1-\beta)\\
&\quad + Q\left(\frac{\frac{\mu^{(t)}}{2} - \log\left(\frac{1-\kappa-\alpha\kappa+\alpha\kappa\beta)}{(1-\alpha\beta)\kappa}\right)}{\sqrt{\mu^{(t)}}}\right)(1-\alpha\beta)\\
&\quad + Q\left(\frac{\frac{\mu^{(t)}}{2} + \log(\frac{\beta}{1-\beta})}{\sqrt{\mu^{(t)}}}\right)\alpha\beta,
\end{aligned}$$

which can be upper bounded using the Chernoff bound[3] and the upper bound in (14). Then, by (13), the result follows. □

In short, Theorem 1 states that the expected fraction of misclassified subgraph nodes tends to zero as $K/n \to 0$ for any $\lambda > 0$ as long as $\alpha > 0$ and $\beta > 0$, i.e., if there exists non-trivial side-information. This form of recovery is called weak recovery [14]. In [12], [14] it is shown that BP without side-information achieves weak-recovery when $\lambda > 1/e$ and fails to achieve weak recovery when $\lambda < 1/e$ whereas global ML detection succeeds for all $\lambda > 0$. Thus we have shown that side-information enables us to remove the threshold phenomenon that exists in BP for subgraph detection without side-information.

We also compare our algorithm with a naive algorithm defined as follows. Pick nodes with K largest value of $d_i(C)$ defined as $d_i(C) = |\{(i,j) : j \in C\}|$. The question to ask is: How does this method fare against BP? We can determine the distribution of $d_i(C)$ in the limit as $n \to \infty$ using the coupling formulation in Lemma 1 and it is given as:

$$d_i(C) \sim \begin{cases}\text{Poi}(\kappa\alpha b(1 + (\rho-1)\beta)) & \text{if } i \in S\\ \text{Poi}(\kappa\alpha b) & \text{if } i \notin S,\end{cases} \tag{15}$$

[3]$Q(x) \leq \exp(-x^2/2)$.

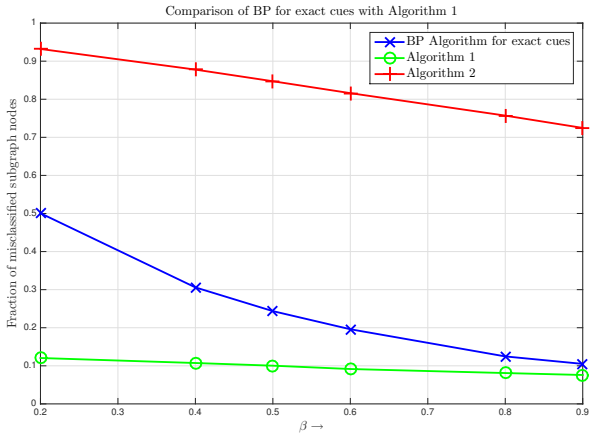

Fig. 1. Comparison of Algorithms 1 and BP for exact cues

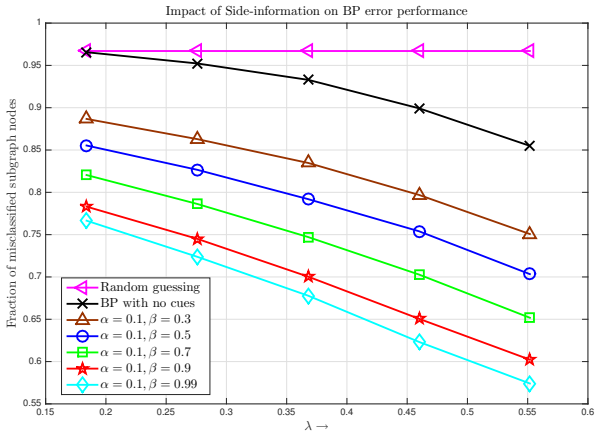

Fig. 2. Comparison of BP for subgraph detection for different amounts of side-information

where $\rho := a/b$.
Clearly, for any constant λ we have that $\rho \to 1$ as $b \to \infty$ and thus, this method of detection will fail to correctly classify the subgraph nodes for any constant λ, and is thus inferior to the BP algorithm. In the next section, we compare the two methods on simulated graphs.

V. Numerical Experiments

In our first experiment, we compare Algorithm 1 against BP algorithm for exact cues given in [17], in the presence of inaccurate cues. We also compare our algorithm with the simple algorithm detailed in (15), denoted as Algorithm 2 in the figure. The numerical results we obtain demonstrate that BP for exact cues is not robust with respect to erroneous cues, and hence there is a need for adopting an algorithm that allows for inexact or erroneous cues. We simulate $G(K, n, p, q)$ with $n = 10^4, K = 200, p = 0.05$, and $q = 0.0046$. We fix $\alpha = 0.1$ and compute the error metric $\sum_{i \in S} \chi_{\{\hat{\sigma_i}=0\}}/K$, i.e., the fraction of wrongly classified subgraph nodes. In Figure 1 we plot this metric against β.

In the second set of experiments we study the impact of side-information on BP performance. To this end we compare the performance of BP algorithm without side-information given in [12] to our algorithm. We simulate a graph of size $10^4, q = 0.0140$ and $K = 330$ for different values of λ by varying p. In Figure 2 we plot the metric $\sum_{i \in S} \chi_{\{\hat{\sigma_i}=0\}}/K$ against λ for different values of β, with $\alpha = 0.1$. For comparison we plot the error rate of random guessing. The results demonstrate that having side-information provides significant improvements over standard BP.

VI. Conclusions

In this work we investigated how a subgraph detection algorithm using cues based on Belief Propagation can be modified to take into account imperfect side-information, i.e., the fact that some of the cues can be wrong. We presented a new algorithm based on Belief Propagation that uses imperfect side-information and showed that it achieves weak recovery whenever $\lambda > 0$, in the limiting regime where $a, b \to \infty$, thus showing that BP with side-information does not exhibit the threshold phenomenon found in BP without side-information.

References

[1] D. H. Chau, S. Pandit, and C. Faloutsos, "Detecting fraudulent personalities in networks of online auctioneers," in *PKDD*. Springer, 2006, pp. 103–114.

[2] A. Beutel, W. Xu, V. Guruswami, C. Palow, and C. Faloutsos, "Copycatch: stopping group attacks by spotting lockstep behavior in social networks," in *Proceedings of the 22nd WWW*. ACM, 2013, pp. 119–130.

[3] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[4] T. Mifflin, C. Boner, G. Godfrey, and J. Skokan, "A random graph model for terrorist transactions," in *2004 IEEE Aerosp. Conf. Proc.*, vol. 5. IEEE, 2004, pp. 3258–3264.

[5] K. Rohe, S. Chatterjee, and B. Yu, "Spectral clustering and the high-dimensional stochastic blockmodel," *Ann. Stat.*, pp. 1878–1915, 2011.

[6] E. Abbe and C. Sandon, "Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery," in *FOCS, 2015*. IEEE, 2015, pp. 670–688.

[7] Y. Deshpande, E. Abbe, and A. Montanari, "Asymptotic mutual information for the binary stochastic block model," in *ISIT 2016*. IEEE, 2016, pp. 185–189.

[8] Y. Deshpande and A. Montanari, "Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time," *Foundations of Computational Mathematics*, vol. 15, no. 4, pp. 1069–1128, 2015.

[9] ——, "Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems." in *COLT*, 2015, pp. 523–562.

[10] M. Jerrum, "Large cliques elude the metropolis process," *Random Structures & Algorithms*, vol. 3, no. 4, pp. 347–359, 1992.

[11] N. Alon, M. Krivelevich, and B. Sudakov, "Finding a large hidden clique in a random graph," *Random Structures and Algorithms*, vol. 13, no. 3-4, pp. 457–466, 1998.

[12] A. Montanari, "Finding one community in a sparse graph," *Journal of Statistical Physics*, vol. 161, no. 2, pp. 273–299, 2015.

[13] B. Hajek, Y. Wu, and J. Xu, "Information limits for recovering a hidden community," in *Information Theory (ISIT), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1894–1898.

[14] ——, "Recovering a Hidden Community Beyond the Spectral Limit in $O(|E|\log^*|V|)$ Time," *arXiv Prepr. arXiv1510.02786*, 2015.

[15] E. Mossel and J. Xu, "Local Algorithms for Block Models with Side Information," in *ITCS '16*. New York, New York, USA: ACM Press, jan 2016, pp. 71–80.

[16] T. T. Cai, T. Liang, and A. Rakhlin, "Inference via message passing on partially labeled stochastic block models," *arXiv preprint arXiv:1603.06923*, 2016.

[17] A. Kadavankandy, K. Avrachenkov, L. Cottatellucci, and R. Sundaresan, "Subgraph detection with cues using belief propagation," *arXiv preprint arXiv:1611.04847*, 2016.

[18] ——, "Belief Propagation for Subgraph Detection with Imperfect Side-information," Eurecom, France, Research Report RR-17-330, Jan 2017.