Bayesian Modeling of Random Walker for Community Detection in Networks

Takafumi J. Suzuki
Communication Technology Laboratory, Research & Technology Group, Fuji Xerox Co., Ltd.
6-1 Minatomirai, Nishi-ku, Yokohama, Kanagawa, 220-8668, Japan
(Dated: October 28, 2019)

We propose a generative model to detect globally optimal community structures in networks by utilizing random walks. Sophisticated parameter optimization algorithms are developed based on the Markov chain Monte Carlo methods to overcome limitations of the EM algorithm, which has been used in previous works but is sometimes trapped in local optima depending on initial conditions. We apply the algorithms to synthetic and real-world networks to examine their performance in terms of precision and robustness of detected communities. It is found that the Gibbs samplers outperform the previous approaches especially in detecting overlapping communities. The Markovian dynamics of random walkers is crucial to robustly detect the optimal community structures.

I. INTRODUCTION

Network structures are ubiquitously found in a wide variety of fields in science such as biology, sociology, and computer science [1]. Nodes and links in networks represent fundamental elements and their interactions in the target systems. These constituents are often spontaneously organizing into structures called communities, i.e. groups of nodes which are densely connected to each other but sparsely connected to the rest of the network. In the last few decades, a large number of methods have been developed to detect communities because uncovering the underlying community structures is an essential step to clarify both microscopic and macroscopic functionalities of the networks [2–8].

Among the various approaches for community detection, stochastic modeling of generative processes in networks has been recognized as a promising technique, and formed the active field of research [6]. The stochastic block model and its variants are prominent examples of such models [9–12]. These models are based on an idea that links in the networks are generated with probability determined by community assignments of their endpoint nodes; the nodes belonging to the same community are more likely to be connected than those in different communities. This simple idea can be flexibly extended to incorporate various aspects of the networks such as heterogeneity in the degrees of node [11] and their mixed memberships [13]. Another successful approach for detecting communities is to utilize random walkers on networks. Random walks are stochastic processes of the agents who randomly select their paths and travel around the networks. The random walks have formed the basis of modern community-detection methods such as Walktrap [14] and Infomap [15, 16] because they offer intuitive and useful pictures of elementary dynamics of the networks.

Recent research has shown that fundamental characteristics of real-world networks can be reasonably explained by communities of links rather than nodes [6]. In particular, overlapping of node communities is understood as a natural consequence of link communities [17–18]. The key point is that most of the links in typical networks have unique properties to characterize relationships between the nodes. For instance, links in social networks have clear meanings such as kinship or coworker relationships, while each individual plays multiple roles in the concerned networks. These observations have promoted the development of methods for detecting link communities [6, 17–22].

Modular decomposition of Markov chain (MDMC) is a community detection method proposed in Ref. [23] by combining the aforementioned insights into community structures. MDMC shares common ideas with previous methods for link communities in that generative processes of links are modeled with underlying community structures. The important difference is that random walks are incorporated in the generative processes in MDMC, while only the local pairs of nodes are considered in the previous methods. Since the random walks are useful to capture global structures of the networks, they allow us to find optimal community structures without being trapped in meta-stable local ones. In return for stable and high-quality results in MDMC, however, it is necessary to solve coupled master equations, which require sophisticated schemes to optimize model parameters. Unfortunately, the EM algorithm, which was utilized in the original formulation [23, 24], does not always find the optimal structures because it is sensitive to initial values of the parameters. Hence, elaboration of the optimization algorithm is highly desired to make full use of the powerful modeling of MDMC.

In this paper, we develop efficient parameter optimization algorithms for MDMC based on the variational Bayesian and Markov chain Monte Carlo methods to enhance the performance of the model. The developed algorithms are applied to both synthetic and real-world networks, and evaluated in terms of the normalized mutual information and the number of detected communities. It is shown that Gibbs samplers enable us to robustly detect optimal community structures in a wide variety of networks compared with previous deterministic methods.
We further analyze intermediate optimization processes of the community assignments and show that the Markovian dynamics of random walkers is essential for finding the globally optimal solutions.

This paper is organized as follows. In Sec. II, we describe guiding principles of MDMC, and formulate generative processes of links with random walkers on networks. Gibb samplers and variational Bayesian methods for inferring the model parameters are developed based on the Bayesian formulation of MDMC. The performance of the developed optimization algorithms is evaluated in Sec. III with synthetic and real-world networks. We will also show optimization processes of community assignments and model parameters to understand the roles of the random walkers in the inference steps. Concluding remarks are stated in Sec. IV.

II. METHODOLOGY

A. Bayesian Formulation of MDMC

The key idea of modular decomposition of Markov chain (MDMC) is to introduce a random walker on a network and trace the Markovian dynamics of the agent in terms of the probability distributions. With \( K \) latent communities behind the network, the probability for the agent to be found at node \( n \) at time \( t \) is denoted by \( p(t)(n|k) \) provided that he is in community \( k \).

Suppose that the agent is observed to be traveling on a link \( d \), which connects nodes \( i \) and \( j \) in the network. This information can be encoded into an \( N \)-dimensional vector \( \tau_{(t)} \) as

\[
\tau_{(t)} = \begin{cases} 
1 & n \in \{i, j\} \\
0 & \text{otherwise}
\end{cases}.
\] (1)

Hereafter, the colon is used as a placeholder of the corresponding arguments. We repeat this procedure for every observable link \( d = 1, 2, \cdots, D \) in the network to get observed data \( \tau^{(t)} \) on the positions of the agent. In order to identify community assignments of the links, we introduce a \( K \)-dimensional one-hot vector \( z_{(t)}^d \) for each link \( d \): When the link \( d \) belongs to community \( k \), the \( k \)-th component of the latent variable is set as \( z_{dk}^{(t)} = 1 \). Mixtures of the link communities are described by an additional \( K \)-dimensional parameter \( \pi_{(t)} \), which satisfies the condition \( \sum_{k=1}^{K} \pi_{dk}^{(t)} = 1 \). When the link \( d \) connecting nodes \( m \) and \( n \) is assigned to, let’s say, community \( k \), the probability for the link to be generated is assumed to be proportional to \( p(t)(n|k)p(t)(m|k) \). The probability of observing the data \( \tau_{(t)}^d \) is obtained by summing the above probabilities over all the possible communities.

In the Bayesian formulation, MDMC consists of the following generating processes at time \( t \).

1. A probability distribution \( p(t) \) is sampled from the Dirichlet prior distribution;

\[
P(p(t)|\alpha(t)) = \prod_{k=1}^{K} \text{Dir}(p(t)(|k)|\alpha(t)(|k)),
\] (2)

where \( \text{Dir}(x|\beta) \) denotes the Dirichlet distribution

\[
\text{Dir}(x|\beta) = \frac{\Gamma(\sum_{n=1}^{N} \beta_n)}{\prod_{n=1}^{N} \Gamma(\beta_n)} \prod_{n=1}^{N} (x_n)^{\beta_n-1},
\] (3)

with the gamma function \( \Gamma(x) \). The parameter of the distribution (2) is defined by

\[
\alpha(t)(n|k) := \alpha_k^{(t)} \sum_{m=1}^{N} T_{nm} p^{(t-1)}(m|k) + 1,
\] (4)

with the transition matrix \( T_{nm} \) \( (n, m = 1, 2, \cdots, N) \), which satisfies \( \sum_{n=1}^{N} T_{nm} = 1 \). We note that the probability distribution at time \( t \) depends only on the latest one \( p^{(t-1)} \). This assumption leads to Markovian dynamics of random walkers. It is shown later that the maximum a posteriori (MAP) estimation of the parameter \( p^{(t)} \) results in master equations for a network augmented by the observed data \( \tau^{(t)} \) [see Eq. (36) in Sec. II C].

2. For \( d = 1, 2, \cdots, D \), a \( K \)-dimensional mixture parameter \( \pi_d^{(t)} \) is independently sampled from the Dirichlet distribution with a prior parameter \( \eta^{(t)} \) of the same dimension;

\[
P(\pi_d^{(t)}|\eta^{(t)}) = \text{Dir}(\pi_d^{(t)}|\eta^{(t)}).
\] (5)

3. A latent variable \( z_{d}^{(t)} \) for community assignments of the link \( d \) is sampled from the categorical distribution with the mixture parameter \( \pi_d^{(t)} \);

\[
P(z_d^{(t)}|\pi_d^{(t)}) = \prod_{k=1}^{K} \left( \pi_{dk}^{(t)} \right)^{z_{dk}^{(t)}}.
\] (6)
Given that the link \( d \) belongs to community \( k \), the observed data \( r_{dk}^{(t)} \) obeys the multinomial distribution:

\[
P(r_{dk}^{(t)} | z_{dk}^{(t)} = 1, \rho^{(t)}) = \text{Mult}(r_{dk}^{(t)} | \rho^{(t)} : | k \rangle), \tag{7}
\]

with

\[
\text{Mult} (x|p) = \frac{\sum_{n=1}^{N} x_n!}{\prod_{n=1}^{N} x_n!} \prod_{n=1}^{N} (p_n)^{x_n}. \tag{8}
\]

The generative model composed of Eq. (2), (5), (6), and (7) is graphically represented in Fig. 1 [25]. The grayed circles describe the observed data, while the open ones denote the random variables and the parameters to be determined in subsequent inference steps. The rectangle plates represent that the stochastic variables inside are independently sampled for \( D \) times. Table I lists the notations of the variables and the parameters used in MDMC.

### Table I. Notations of parameters and variables in MDMC

| Notation | Description | Dimension |
|----------|-------------|-----------|
| \( N \) | Number of nodes | 1 |
| \( D \) | Number of links | 1 |
| \( K \) | Number of communities | 1 |
| \( T_{\text{step}} \) | Number of Markovian time steps | 1 |
| \( S \) | Number of Monte Carlo sampling | 1 |
| \( T_{\text{iter}} \) | Number of iterations in deterministic algorithms | 1 |
| \( t \) | Time of Markov chain | 1 |
| \( \alpha_k^{(t)} \) | Parameters of Dirichlet distribution for \( p^{(t)} : | k \rangle \) | \( K \) |
| \( \eta_k \) | Parameters of Dirichlet distribution for \( \pi^{(t)} \) | \( K \) |
| \( T_{\text{obs}} \) | Transition matrix | \( N \times N \) |
| \( r_{nd} \) | Observed data | \( N \times D \) |
| \( z_{dk} \) | Latent variables for community assignments | \( D \times K \) |
| \( p^{(t)}(n|k) \) | Probability distributions of random walkers | \( N \times K \) |
| \( \pi_{dk} \) | Mixture parameters | \( D \times K \) |

4. With the joint probability distribution

\[
P(\tau^{(t)} , z^{(t)} , \rho^{(t)} , \pi^{(t)} | \alpha^{(t)} , \eta^{(t)})
\]

with the joint probability distribution

\[
P(\tau^{(t)} , z^{(t)} , \rho^{(t)} , \pi^{(t)} | \alpha^{(t)} , \eta^{(t)})
\]

\[
\approx \prod_{t=1}^{T_{\text{step}}} P(\tau^{(t)} | \alpha^{(t)} , \eta^{(t)}) \tag{11}
\]

where the number of the whole time steps is denoted by \( T_{\text{step}} \). However, direct optimization of the total likelihood function (11) is challenging because it requires us to simultaneously determine the parameters at the whole time steps. In our formulation, we sidestep this issue by approximating the total likelihood function (11) as the product of the likelihood function (9) with parameters separately optimized for each time slice \( t \):

\[
P(\tau^{(t)}_{t=1,2,\cdots,T_{\text{step}}}) \approx \prod_{t=1}^{T_{\text{step}}} P(\tau^{(t)} | \hat{\alpha}^{(t)} , \hat{\eta}^{(t)}) \tag{12}
\]

Here, we determine the parameters \( \hat{\alpha}^{(t)} \) and \( \hat{\eta}^{(t)} \) by using the estimate of the probability distribution \( \hat{p}^{(t-1)} \) obtained at the previous time step. Thus, we cut the dependency chain in Fig. 1 and the problem boils down to the sequential optimization of Eq. (9) with given \( \hat{p}^{(t-1)} \). This strategy is similar to the one utilized in the topic tracking model [20]. In following discussions, we omit the hat of the estimated parameters because they may cause no confusion.
C. Inference Methods of MDMC Parameters

In this subsection, we develop stochastic [Gibbs and collapsed Gibbs sampling] and deterministic [variational Bayesian and EM algorithms] methods to determine the stochastic variables $p(t)$, $\pi(t)$, and $z(t)$ in MDMC. At the same time, we will derive update equations of the hyperparameters $\alpha_k^{(t)}$ and $\eta_k^{(t)}$ by approximately maximizing the likelihood function [9].

1. Gibbs Sampling

We will first develop the Gibbs sampling algorithm, where the stochastic variables $p(t)$, $\pi(t)$, and $z(t)$ are numerically sampled from their posterior distributions. The probability distribution of $p(t)$ on the condition that the other parameters are fixed is calculated as

$$P(p(t); |k), \pi(t), \alpha(t), \eta(t)) = \text{Dir}(\pi_d(t); |k) + (\tau z)^{(t)}),$$

with $(\tau z)^{(t)} = \sum_{d=1}^{D} \gamma_d z_{dk}$. Hereafter, the backslash symbol $\backslash$ is used to denote a set of the components other than $x$. Similarly, the probability distribution of the mixture parameter $\pi_d^{(t)}$ can be obtained as

$$P(\pi_d^{(t)}; r(t), z(t), p(t), \pi(t), \alpha(t), \eta(t)) = \text{Dir}(\pi_d^{(t)}; |k),$$

for each $d = 1, 2, \ldots, D$. Given the current samples of $p(t)$ and $\pi(t)$, the latent one-hot vector $z_{dk}$ obeys the following probability distribution function:

$$P(z_{dk} = 1; \pi_d(t), z(t), p(t), \pi(t), \alpha(t), \eta(t)) = \frac{\pi_{dk}^{(t)} \text{Mult}(\tau_{dk}^{(t)}; |k)}{\sum_{k=1}^{K} \pi_k^{(t)} \text{Mult}(\tau_{dk}^{(t)}; |k)}.$$

The detailed derivation of Eqs. (13), (14), and (15) are given in Appendices.

After taking $S$ samples of the probability distribution, we estimate the expectation value of $p(t)(n|k)$ for each time slice as

$$p(t)(n|k) \approx \frac{1}{S} \sum_{s=1}^{S} p_{ss}^{(t)}(n|k).$$

Hereafter, $X_s$ denotes each realization of the random variable $X$ at sample $s$.

In this paper, we update the hyperparameters $\alpha_k^{(t)}$ and $\eta_k^{(t)}$ at the ends of each Markov step by approximately maximizing the likelihood function with Newton’s method and Minka’s fixed-point iteration [27], respectively. First, the parameter $\alpha_k^{(t)}$ is updated as

$$\alpha_k^{(t+1)} = \alpha_k^{(t)} - \frac{F_k(\alpha_k^{(t)})}{F_k'(\alpha_k^{(t)})},$$

with the logarithmic derivative of the likelihood function

$$F_k(\alpha_k^{(t)}) = \frac{d}{d\alpha_k^{(t)}} \ln P(\pi^{(t)}; z^{(t)}, \alpha(t), \eta(t))$$

$$= \sum_{n=1}^{N} \sum_{t=1}^{T} \left( Tp(t-1) \right)_{nk} \alpha(t)(n|k) + l - 1$$

$$- \sum_{t=1}^{T} \left( \sum_{n=1}^{N} \alpha(t)(n|k) \right) + N + l - 1.$$  (18)

The derivation of Eq. (18) is given in Appendix. The updated parameters $\alpha_k^{(t+1)}$ and the estimate (16) of the probability distribution are used to compute $\alpha(t+1)(n|k)$ for the prior probability distribution of $p(t+1)$ at the next time [see Eq. (14) for the definition of $\alpha(t)(n|k)$]. Second, the parameter $\eta_k^{(t+1)}$ for the next time step is obtained by using Minka’s fixed-point iteration as

$$\eta_k^{(t+1)} = \frac{Z_k^{(t)}}{D} \sum_{k=1}^{K} \eta_k^{(t)}$$

with $Z_k^{(t)} = \sum_{d=1}^{D} z_{dk}$. We note that the sum of the parameter $\eta_k^{(t)}$ is conserved during the Markov time steps because of the relation $\sum_{k=1}^{K} \eta_k^{(t)} = D$.

To summarize, pseudocode of the Gibbs sampling is shown in Algorithm 1. As is discussed below, we can elaborate the Gibbs sampling algorithm by analytically integrating the intermediate variables $p(t)$ and $\pi(t)$. Hence, this algorithm is primarily used to check consistency of results obtained by the collapsed Gibbs sampling.

Algorithm 1 Gibbs Sampling

1. Initialize $p^{(0)}$, $\pi^{(0)}$, and $\eta^{(0)}$
2. for $t = 1, 2, \ldots, T_{	ext{step}}$
3. Initialize $z^{(t)}$, $p^{(t)}$, and $\pi^{(t)}$
4. for $s = 1, 2, \ldots, S$
5. for $k = 1, 2, \ldots, K$
6. Sample $p^{(t)}(; |k)$ [Eq. (13)]
7. end for
8. for $d = 1, 2, \ldots, D$
9. Sample $\pi_d^{(t)}$ [Eq. (14)]
10. Sample $z_{dk}^{(t)}$ [Eq. (15)]
11. end for
12. end for
13. Estimate $p^{(t)}$ [Eq. (16)]
14. Compute $\alpha_{k}^{(t+1)}$ [Eq. (17)]
15. Compute $\eta_k^{(t+1)}$ [Eq. (19)]
16. end for
2. Collapsed Gibbs Sampling

It is possible to analytically integrate out the stochastic variables \(\pi^{(t)}\) and \(p^{(t)}\) in the likelihood function to get the marginalized distribution. This property enables us to efficiently sample the latent variable \(z^{(t)}\) compared with the Gibbs sampling algorithm.

The marginalized distribution is given by

\[
P(z^{(t)} = \alpha^{(t)}, \eta^{(t)})
\]

In summary, pseudocode of the collapsed Gibbs sampling algorithm is shown in Algorithm 2.

3. Variational Bayesian Approach

We can derive the lower bound of the log likelihood by using Jensen’s inequality as

\[
\ln P(\tau^{(t)}|\alpha^{(t)}, \eta^{(t)}) \geq F[q(z^{(t)}, p^{(t)}, \pi^{(t)})]
\]

which should be maximized with a distribution \(q(z^{(t)}, p^{(t)}, \pi^{(t)})\). We assume that an optimal form of the distribution can be decoupled as

\[
q(z^{(t)}, p^{(t)}, \pi^{(t)}) = q^z(z^{(t)})q^p(p^{(t)})q^\pi(\pi^{(t)}),
\]

with

\[
q^z(z^{(t)}) = \prod_{d=1}^{D} \prod_{k=1}^{K} q^z_{dk}(z^{(t)}_{dk}),
\]

\[
q^p(p^{(t)}) = \prod_{k=1}^{K} q^p_k(p^{(t)}(:|k)),
\]

\[
q^\pi(\pi^{(t)}) = \prod_{d=1}^{D} q^\pi_d(\pi^{(t)}_d).
\]

With the aid of the joint probability distribution \(P(\tau^{(t)}|\alpha^{(t)}, \eta^{(t)})\) and the mean-field approximation, the variational
lower bound [23] can be computed as

\[
F[q(z^{(t)}, p^{(t)}, \pi^{(t)})] = \int dp^{(t)}d\pi^{(t)} \sum_{z^{(t)}} q^z(z^{(t)}) q^p(p^{(t)}) q^\pi(\pi^{(t)}) \times \ln P(\pi^{(t)}|z^{(t)}, p^{(t)}) P(z^{(t)}|\pi^{(t)}) - D_{KL}(q^{p}|P(p^{(t)}|\alpha^{(t)})) - D_{KL}(q^{\pi}|P(\pi^{(t)}|\eta^{(t)})) - \sum_{z^{(t)}} q^z(z^{(t)}) \ln q^z(z^{(t)}),
\]

with the Kullback-Leibler divergence

\[
D_{KL}(Q||P) = \int dx Q(x) \ln \frac{Q(x)}{P(x)}.
\]

The optimal functional form of \(q^z, q^p,\) and \(q^\pi\) can be obtained by functionally differentiating the variational lower bound [23] with respect to them as

\[
q^z_{dk}(z_{dk}) = 1 \propto \frac{\exp \left[ \sum_{n=1}^{N} t_{nd}^{(t)} \Psi(\xi_{pnk}) \right]}{\exp \left[ \sum_{n=1}^{N} t_{nd}^{(t)} \Psi(\sum_{n'=1}^{N} \xi_{pn'k}) \right]} \times \frac{\Psi(\xi_{\tau dk})}{\Psi(\sum_{k'=1}^{K} \tau_{k'dk})},
\]

(30)

\[
q^p(\pi^{(t)}) = \text{Dir}(\pi_d^{(t)}|\xi_d^{(t)}),
\]

(31)

\[
q^\pi_k(p^{(t)}; |k|) = \text{Dir}(p^{(t)}; |k|) \xi_{pk}^{(t)},
\]

(32)

where \(\Psi(x)\) denotes the digamma function and the parameters are defined by

\[
\xi_{\tau dk} = \alpha_k^{(t)} + \gamma_{dk},
\]

(33)

\[
\xi_{pnk}^{(t)} = \alpha_k^{(t)}(n|k) + \sum_{d=1}^{D} \tau_{nd}^{(t)} \gamma_{dk},
\]

(34)

with the responsibility \(\gamma_{dk} = q^z_{dk}(z_{dk}) = 1\). The parameters \(\gamma_{dk}, \xi_{\tau dk},\) and \(\xi_{pnk}^{(t)}\) in the variational Bayesian approach are self-consistently determined by solving the coupled equations [30], [33], and [34] at each time step. Finally, we estimate \(p^{(t)}\) by using its expectation value as

\[
p^{(t)}(n|k) = \frac{\alpha_k^{(t)}(n|k) + \sum_{d=1}^{D} \tau_{nd}^{(t)} \gamma_{dk}}{\sum_{n=1}^{N} \left[ \alpha_k^{(t)}(n|k) + \sum_{d=1}^{D} \tau_{nd}^{(t)} \gamma_{dk} \right]},
\]

(35)

at the end of each Markov step. The optimization scheme based on the variational Bayesian approach is summarized in Algorithm 3. In this paper, we repeat the substitution of the parameters for Eqs. (30), (33), and (34) \(T_{iter}\) times on each Markov time step to achieve the self-consistent solutions.

### 4. EM Algorithm

The original MDMC model developed by Okamoto and Qiu utilized the EM algorithm to infer the model parameters [23, 24]. In the following, we derive the EM algorithm based on the results of the variational Bayesian approach, and discuss differences between the original and current formulations.

We will first determine the probability \(p^{(t)}(n|k)\) with the MAP estimation. Since \(p^{(t)}(n|k)\) obeys the Dirichlet distribution [22], the MAP estimate of the parameter is given by the mode of the posterior Dirichlet distribution as

\[
p^{(t)}(n|k) = \frac{1}{N} \left[ \frac{\alpha_k^{(t)}(n|k)}{\sum_{m=1}^{N} t_{nm}^{(t-1)}(m|k)} + \sum_{d=1}^{D} \tau_{nd}^{(t)}(m|k) \right].
\]

(36)

Here, we have used the definition of \(\alpha_k^{(t)}(n|k)\) [Eq. (4)] and introduced the normalization factor \(N_k = \sum_{k=1}^{K} \alpha_k^{(t)}(m|k) + \sum_{d=1}^{D} \tau_{nd}^{(t)}(m|k)\). At this stage, the designing principles of the prior parameter \(\alpha_k^{(t)}(n|k)\) can be clarified. The first term in the right hand side of Eq. (36) corresponds to the master equation with the original network structure, while the second one describes the modification via \(D\) times observation. Hence, Eq. (36) describes the Markovian dynamics of random walkers on a network augmented by the explicitly observed data \(\tau^{(t)}\). The hyperparameter \(\alpha_k^{(t)}\) controls the ratio of these contributions: When we take the infinite limit of the parameter as \(\alpha_k^{(t)} \to \infty\), Eq. (36) is reduced to the original master equation, where random walkers are independently obeying the same equation for each community. On the other hand, the contributions from the random walkers are washed away in the vanishing limit of \(\alpha_k^{(t)}\).

The current modeling of MDMC is slightly different from the original one [23, 24] in the presence of the prior distribution of \(\pi_{dk}^{(t)}\). Since the variance of the Dirichlet
distribution shrinks in large parameter regimes, the original model can be recovered by taking the infinite \( \eta \) limit of ours. Noting that the digamma function is asymptotically identical to the logarithm function as \( \Psi(x) \sim \ln(x) \) for \( x \gg 1 \), we can transform the result (30) of the variational Bayesian approach to

\[
\gamma_{dk}^{(t)} = \frac{\eta_k^{(t)} \text{Mult} \left( \tau_d^{(t)} | p_d^{(t)}(\cdot | k) \right)}{\sum_{k=1}^{K} \eta_k^{(t)} \text{Mult} \left( \tau_d^{(t)} | p_d^{(t)}(\cdot | k) \right)}. \tag{37}
\]

The estimate of the probability distribution (36), the responsibility (37), and the update equation (39) of \( \eta_k^{(t)} \) can form a closed set of self-consistent equations, which are equivalent to those proposed in the original MDMC [23, 24]. We note that there remains some differences in the policies of updating the parameters \( \alpha_k^{(t)} \) and \( \eta_k^{(t)} \). (1) The parameters \( \alpha_k^{(t)} \) are not updated in the original MDMC. (2) The parameters \( \eta_k^{(t)} \) are updated at the end of each time step in the current formulation, while they are updated at the end of each EM step in the original MDMC. In the following analysis, we use our new procedures described in Algorithm 4 in order to compare the algorithm with other solvers.

Algorithm 4. EM Algorithm

1: Initialize \( p_0^{(0)}, \alpha_k^{(1)} \), and \( \eta_k^{(1)} \)
2: for \( t = 1, 2, \ldots, T_{\text{step}} \) do
3: for \( \text{iter} = 1, 2, \ldots, T_{\text{iter}} \) do
4: for \( d = 1, 2, \ldots, D \) do
5: Update responsibility \( \gamma_d^{(t)} \) [Eq. (37)]
6: end for
7: for \( k = 1, 2, \ldots, K \) do
8: Update \( p_d^{(t)}(\cdot | k) \) [Eq. (36)]
9: end for
10: end for
11: Estimate \( p_d^{(L)} \) [Eq. (36)]
12: Compute \( \alpha_k^{(L+1)} \) [Eq. (37)]
13: Compute \( \eta_k^{(L+1)} \) [Eq. (39)]
14: end for

III. EVALUATION OF PARAMETER INFERENCE METHODS

In this section, we apply the optimization algorithms developed in the previous section to various testbed networks to qualitatively evaluate their performance. We use a normalized mutual information (NMI), which measures the similarity between two different partitions of community assignments, namely our prediction and the ground truth. As has been discussed in the previous section, modular decomposition of Markov chain (MDMC) has been designed to capture global structures of networks by combining stochastic modeling of network generation processes and Markovian dynamics of random walkers. This interpretation allows us to further analyze the detected communities by visualizing and diagnosing optimization processes.

A. Node Clustering

While MDMC assigns communities to every link in networks, most of the existing testbed networks are equipped with ground-truth node communities. Hence, we need an additional policy to determine the membership of the nodes within our framework. In this paper, we will define the joint probability distribution of the random walker with respect to node \( n \) and community \( k \) as \( p_d^{(t)}(n, k) = p_d^{(t)}(n|k)p_d^{(t)}(k) \) with \( p_d^{(t)}(k) = \eta_k^{(t)}/\sum_{k=1}^{K} \eta_k^{(t)} \). Then, the most probable community can be assigned to each node \( n \) by taking the largest component of \( p_d^{(t)}(k|n) \). As a metric for comparison, we use an NMI which has been extended to quantify similarities of two different partitions with overlapping communities 28, 29.

B. Experiment with LFR Benchmark

We use LFR benchmark 30 in order to evaluate the performance of MDMC solvers. The LFR benchmark has been widely used to evaluate community detection algorithms because it allows us to generate various types of realistic networks including those with overlapping communities [4, 19, 20]. We will compare MDMC with a stochastic model, which was proposed by Ball et al. in Ref. 4 and has been considered as one of the best models to detect overlapping communities.

We have generated undirected and unweighted networks with \( N = 1000 \) nodes for four different ratios \([o_m/N = 0, 0.1, 0.2, \text{and } 0.3]\) of nodes on which \( o_m = 2 \) different communities overlap. The minus exponents of the degree and the community size distributions are set as \( t_1 = 2 \) and \( t_2 = 1 \), respectively. The average degree is \( k_{\text{ave}} = 15 \) with an upper bound \( k_{\text{max}} = 50 \). The size of the generated communities ranges from \( c_{\text{min}} = 20 \) to \( c_{\text{max}} = 100 \). For each overlapping-node ratio, the mixing parameter \( \mu \), which controls the number of edges between different communities, is varied from \( \mu = 0.1 \) to 0.7 with spacing 0.1 to examine effects of the mixtures on the performance. We generated 10 networks for each parameter set to evaluate the mean and the variance of NMI values. In order to set initial values of MDMC parameters, we read the number of ground truth community \( K \) and that of edges \( D \) for each network and set the values as \( \alpha_k^{(1)}/D = 0.1 \) and \( \eta_k^{(1)} = 1 \) for \( k = 1, 2, \ldots, K \).

We show the results of the extended NMI for LFR benchmark in Fig. 2 where EM, VB, GS, and CGS stand for the EM algorithm, the variational Bayesian approach, the Gibbs sampling, and the collapsed Gibbs sampling, respectively. In wide parameter regimes, the Gibbs samplers outperform the variational Bayesian approach and
FIG. 2. (Color online) The results of the extended NMI for LFR benchmark with various values of the mixing parameter $\mu$ and the number of the overlapping nodes. The networks are generated 10 times for each parameter set with $N = 1000$, $o_m = 2$, $t_1 = 2$, $t_2 = 1$, $k_{\text{ave}} = 15$, $k_{\text{max}} = 50$, $c_{\text{min}} = 20$, and $c_{\text{max}} = 100$. MDMC is performed with $T_{\text{step}} = 50$, $\alpha_k^{(1)} = 1, 2, \ldots, K/D = 1$, and $\eta^{(1)}_{k=1,2,\ldots,K} = 1$, where $D$ and $K$ are read from each generated network. The number of the Monte Carlo sampling is set as $S = 100$ with the burn in period $S_{\text{burn}} = 100$ in the Gibbs samplers, while the number of the iteration is set as $T_{\text{iter}} = 1000$ and 100 for the EM algorithm and the variational Bayesian approach, respectively.

EM algorithms in terms of the NMI values. In particular, the results of the deterministic methods deteriorate as the number of overlapping nodes increases, while the Gibbs samplers keep their qualities ($0.8 \lesssim \text{NMI}$) even in highly overlapping $o_n/N = 0.3$ and mixed $\mu \lesssim 0.4$ situations. The Gibbs and collapsed Gibbs sampling algorithms give the same values within statistical errors in every situation, which indicates that the number $S$ of Monte Carlo sampling and the additional parameter $S_{\text{burn}}$ denoting the burn-in period are large enough to achieve the best performance of the model. The results of Ball’s model [6], which is denoted by BKN, is comparable or slightly inferior to those from MDMC with EM-algorithm solver. There may be two possible reasons to explain this behavior: representability of these models and performance of their solvers. MDMC is designed to incorporate global structures of networks by introducing random walkers, while Ball’s model essentially focuses on local link structures around each node. Another possible reason is that Ball’s model exploits a simple EM algorithm, which is known to suffer from trapping of locally optimized states. This leads to a natural concern that the EM algorithm might have not utilized the maximum potential of Ball’s model. In order to fully answer this question, it is necessary to systematically elaborate Ball’s model, which is out of scope of this paper but deserves
another research.

C. Real-world Network

It is meaningful to apply and examine our methods with real-world networks because they may have features which are not captured in synthetic ones. Among the widely studied real-world networks, we choose American college football network [2] to test the performance of our model. The nodes and links in the dataset represent football teams and games played by two of them, respectively. Every node in the network belongs to one of 12 communities called “conferences”, and games took place more often between teams in the same conference than those in different conferences. Football teams in “Independents” conference are exceptional because they do not have particular preferences for their opponents. Since the generative processes of the network imply typical characteristics of node communities, link clustering schemes are known to suffer from severe challenges for detecting the ground truth communities [22]. This observation raises a concern that MDMC, which models community assignments of links rather than nodes, may not detect the true communities.

We have applied MDMC solvers to American college football, and summarized the mean (standard deviation) of the extended NMI in the second column in Table I. We have also analyzed the number of communities which are detected as main one of at least one node in the network. In order to interpret the result that the number of communities detected by MDMC is less than that of the ground truth communities, we identify the communities missed by our detection algorithms. We establish the correspondence between the detected and the ground-truth communities by associating each detected community with the most similar ground-truth community in terms of Jaccard index. The detected community is considered as “undefined” when the maximum value of the Jaccard index is less than 0.5. With these criteria, we have found that “Independents” and “Sun Belt” communities are missing in most cases. In later analysis, we use the collapsed Gibbs sampling and detail the missing communities by directly visualizing optimization processes of the community assignments.

Fig. 3 shows typical results of the main community assignments at different time slices \( t = 1, 3, \) and 50. The colors of the nodes represent the main communities detected by our model. Football teams belonging to “Independents” and “Sun Belt” conferences are shown as triangle and square nodes, respectively. Since “Independents” conference do not have particular community structures, the automatic drop of this community during optimization processes is favorable property of our model. On the other hand, “Sun Belt” nodes partially looks washed away during the Markov chain processes [Fig. 3(c)]. Considering that “Sun Belt” is the smallest community aside from “Independents”, we infer that the resolution-limit problem, which are recognized in modularity optimization methods [31], also matters in the MDMC approach. In fact, small communities observed in early times of the optimization are prone to be merged into larger ones as the random walkers travel the network.

Further insights on the selection of optimal communities can be gained from optimization processes of the model parameters. The left, center, and right panels of Fig. 4 show values of the hyperparameters \( \alpha_k^{(t)}/D \) and \( \eta_k^{(t)} \), and the log likelihood \( \ln P^{(t)}(k|n) \) at different time slices \( t = 1, 3, \) and 50. The colors of the nodes represent the main communities detected by our model. Football teams belonging to “Independents” and “Sun Belt” conferences are shown as triangle and square nodes, respectively. Since “Independents” conference do not have particular community structures, the automatic drop of this community during optimization processes is favorable property of our model. On the other hand, “Sun Belt” nodes partially looks washed away during the Markov chain processes [Fig. 3(c)]. Considering that “Sun Belt” is the smallest community aside from “Independents”, we infer that the resolution-limit problem, which are recognized in modularity optimization methods [31], also matters in the MDMC approach. In fact, small communities observed in early times of the optimization are prone to be merged into larger ones as the random walkers travel the network.

Further insights on the selection of optimal communities can be gained from optimization processes of the model parameters. The left, center, and right panels of Fig. 4 show values of the hyperparameters \( \alpha_k^{(t)}/D \) and \( \eta_k^{(t)} \), and the log likelihood \( \ln P^{(t)}(k|n) \) at different time slices \( t = 1, 3, \) and 50. The colors of the nodes represent the main communities detected by our model. Football teams belonging to “Independents” and “Sun Belt” conferences are shown as triangle and square nodes, respectively. Since “Independents” conference do not have particular community structures, the automatic drop of this community during optimization processes is favorable property of our model. On the other hand, “Sun Belt” nodes partially looks washed away during the Markov chain processes [Fig. 3(c)]. Considering that “Sun Belt” is the smallest community aside from “Independents”, we infer that the resolution-limit problem, which are recognized in modularity optimization methods [31], also matters in the MDMC approach. In fact, small communities observed in early times of the optimization are prone to be merged into larger ones as the random walkers travel the network.

TABLE II. The statistics of NMI values and the counts of the number of detected main communities out of 100 trials for American college football network. MDMC is performed with \( T_{\text{step}} = 50, \alpha_k^{(t)}/D = 0.1, \eta_k^{(t)} = 1, D = 613, \) and \( K = 12. \) The number of the Monte Carlo sampling and the burn in period is set as \( S_{\text{fin}} = 1000 \) and \( S_{\text{burn}} = 200, \) respectively in the Gibbs samplers. The number of the iteration is set as \( T_{\text{iter}} = 1000 (100) \) for the EM algorithm (variational Bayesian approach).

| Method | NMI       | \( K_{\text{fin}} = 9 \) | 10 | 11 | 12 |
|--------|-----------|--------------------------|----|----|----|
| EM     | 0.869 (0.018) | 37 | 53 | 10 | 0  |
| VB     | 0.871 (0.016) | 37 | 63 | 0  | 0  |
| GS     | 0.875 (0.004) | 0  | 92 | 8  | 0  |
| CGS    | 0.875 (0.003) | 0  | 96 | 4  | 0  |
| BKN    | 0.897 (0.021) | 0  | 0  | 20 | 80 |
dynamics occurs when the random walkers travel across different communities. This induces global organization of communities as is observed in Fig. 4(b) for $5 \lesssim t \lesssim 30$. The parameters are finally converged into the stationary values in accordance with the basic principle of Markov chains [32]. These pictures also explain the behavior of the log likelihood in Fig. 4(c), where the error bars represent the mean and the standard deviation of $\ln P$ estimated by Monte Carlo sampling for each time step. We note that the log likelihood is not monotonically increasing in time because of the approximation described in Sec. II B. Developing efficient algorithms to optimize the whole Markov chain of MDMC is one of the important remaining issues.

IV. DISCUSSIONS AND CONCLUSIONS

In this paper, we have improved modular decomposition of Markov chain (MDMC), which is a stochastic model proposed in Ref. [23] and [24] for detecting community structures in networks, by developing various optimization algorithms based on variational Bayesian and Monte Carlo sampling approaches. We have applied MDMC to LFR benchmark [30], and found that the Gibbs sampling algorithms outperform the EM algorithm, which was used in the original paper [23, 24], in wide parameter regimes. In particular, overlapping communities are more accurately detected by the Gibbs samplers, which enhance the effectiveness of MDMC in elucidating these challenging community structures.

We have also examined the performance of MDMC with American college football network [2]. Real-world networks often possess properties which are absent in synthetic ones. For instance, ground-truth communities are not necessarily consistent with given network structures as is the case with “Independents” conference in American college football network. We have found that “Independents” community is automatically dropped during optimization processes of MDMC, while

FIG. 3. (Color online) Typical results of community assignments in American college football network at different time slices $t = 1, 3, 50$. The colors of the nodes represent their main communities. The triangle and square nodes represent teams in “Independents” and “Sun Belt”, respectively. MDMC is performed with $T_{\text{step}} = 50$, $\alpha_k^{(1)} = 1, 2, \ldots, K/D = 0.1$, $\eta_k^{(1)} = 1, 2, \ldots, K = 1$, and $K = 12$. The collapsed Gibbs sampling is used as an MDMC solver with $S_{\text{burn}} = 200$ and $S = 1000$.

FIG. 4. (Color online) The optimization processes of the hyperparameters (a) $\alpha_k^{(t)}$ and (b) $\eta_k^{(t)}$ and (c) the log likelihood $\ln P$ for American college football network with the same run used in Fig. 3. The labels of the colored curves are determined with the Jaccard index, and are common in (a) and (b).
Another promising stochastic model developed by Ball et al. [2] fails to exclude this irrelevant community. The optimal community structures were more often detected by the Gibbs samplers, while deterministic methods, i.e. EM algorithm and the variational Bayesian approach, are too sensitive to initial conditions to robustly find them. We have further detailed the optimization processes of MDMC parameters, and clarified that the automatic selection of the optimal communities is a consequence of global dynamics of random walkers on the network.

Although the Gibbs samplers developed in this paper allow us to exploit the full potential of MDMC to obtain accurate and stable results, they also clarify some limitations of the model. First, the resolution-limit problem looks inevitable in the current formulation because small communities tend to be aggregated into larger ones as the random walkers travel through a network. In order to detect locally-stable but globally-unstable community structures, we need to develop some schemes to reliably extract meta-stable structures in the optimization processes. Second, the likelihood function is not guaranteed to monotonically increase in the Markov steps because it is separately optimized at different time slices [see Sec. II B for detail of the approximation]. As a consequence, it is impossible at the present stage to simultaneously satisfy the two guiding principles of MDMC, i.e. the maximization of the likelihood function and the convergence to the steady-state distribution in the Markov chain. It seems that reliable results are already achieved in a practical point of view, but efficient optimization methods unifying these guiding principles are desirable to assess the validity of this approximation.

It is shown, in this paper, that stochastic modeling of network generation processes are quite efficient to detect community structures in a wide variety of networks. This is reminiscent of the situation in topic modeling, where latent Dirichlet allocation (LDA) model [33] has ignited subsequent works on document clustering methods with additional structures [26, 34]. Among the fundamental characteristics of real-world networks, the existence of underlying hierarchical structures is essential in many cases to understand intrinsic nature of the networks [35]. Another promising direction is to utilize additional meta-information of nodes and links for community detection. We leave these promising directions for future works.

ACKNOWLEDGMENTS

The author would like to thank Xule Qiu, Seiya Inagai, Hiroshi Okamoto, Hiroshi Unemoto, and Takeshi Onishi for useful discussions and comments. In order to evaluate the proposed methods, an open software available at https://sites.google.com/site/santofortunato/inthepress2 was used to generate synthetic networks. In addition, an extended normalized mutual information was computed with another open software available at https://github.com/aaronmcdaid/Overlapping-NMI.

Appendix A: Derivation of Gibbs Sampling

The probability of the latent variable \(z_{dk}^{(t)}\) can be computed as

\[
P(z_{dk}^{(t)} = 1|\tau^{(t)}, z_{d}^{(t)}, p^{(t)}, \pi^{(t)}, \alpha^{(t)}, \eta^{(t)})
\]

\[
= P(z_{dk}^{(t)} = 1|\pi^{(t)}_{d}) P(\tau^{(t)}_{d} p^{(t)}_{d}, \pi^{(t)}_{d})
\]

\[
= P(\tau^{(t)}_{d}, z_{dk}^{(t)} = 1|p^{(t)}_{d}, \pi^{(t)}_{d})
\]

\[
\propto P(\tau^{(t)}_{d}, z_{dk}^{(t)} = 1|p^{(t)}_{d}, \pi^{(t)}_{d})
\]

\[
\propto P(\tau^{(t)}_{d}, |z_{dk}^{(t)} = 1, p^{(t)}_{d}) P(z_{dk}^{(t)} = 1|\pi^{(t)}_{d})
\]

\[
\propto \text{Mult}(\tau^{(t)}_{d} | p^{(t)}_{d}; | k) \pi^{(t)}_{d}. \quad (A1)
\]

Here, we have ignored the denominator of the third line because it is independent of \(z_{dk}^{(t)}\). In the last line, we have used Eqs. (6) and (7). The equation above is equivalent to Eq. (15) up to a normalization factor.

The posterior probability of \(p^{(t)}_{d} | k\) is computed as

\[
P(p^{(t)}_{d} | k) \propto P(\tau^{(t)}_{d}, p^{(t)}_{d} | k), \alpha^{(t)}_{d}, \eta^{(t)}
\]

\[
\propto P(\tau^{(t)}_{d}, p^{(t)}_{d} | k) z_{dk}^{(t)} \alpha^{(t)}_{d}, \eta^{(t)}
\]

\[
\times \text{Dir}(p^{(t)}_{d} | k) \alpha^{(t)}_{d}, \eta^{(t)}
\]

\[
\propto \prod_{d=1}^{D} \prod_{n=1}^{N} \left[ p^{(t)}_{d} (n|k) \right]^{\alpha^{(t)}_{n} (n|k)+\eta(z_{n,k}^{(t)})-1} \quad (A2)
\]

In the third line, we have used Eqs. (2) and (7), and left the factor relevant to \(p^{(t)}_{d} | k\) in the subsequent line. Noting that the last line is proportional to the Dirichlet distribution with parameter \(\alpha^{(t)}_{n} (n|k)+\eta(z_{n,k}^{(t)})\), we arrive at the expression [13].

As is seen from the graphical model [Fig. 1], the probability distribution of \(z_{d}^{(t)}\) depends only on the parameter \(\eta^{(t)}\) of the prior distribution and the latent variables \(z_{d}^{(t)}\). Since the Dirichlet distribution is a conjugate prior of the multinomial distribution, the resulting probability distribution is the Dirichlet distribution with a parameter \(\eta^{(t)} + z_{d}^{(t)}\). This is nothing but the probability distribution [14].
Appendix B: Derivation of Collapsed Gibbs Sampling

The marginalized joint probability distribution is rewritten as

\[ P(\tau^{(t)}, z^{(t)}|\alpha^{(t)}, \eta^{(t)}) = P(\tau^{(t)}|z^{(t)}, \alpha^{(t)}, \eta^{(t)})P(z^{(t)}|\alpha^{(t)}, \eta^{(t)}), \tag{B1} \]

with the multiplication rule. In the following, we will separately compute the two factors in the last line of the above equation.

The first factor in Eq. (B1) is computed as

\[
P(\tau^{(t)}|z^{(t)}, \alpha^{(t)}, \eta^{(t)}) = \frac{\prod_{k=1}^{K} \Gamma\left(\sum_{n=1}^{N} \alpha(n|k)\right)}{\prod_{n=1}^{N} \Gamma(\alpha(n|k))} \times \text{Dir}(p^{(t)}|\alpha^{(t)}),
\]

The last line of Eq. (B2) is obtained with the identity

\[
\int d\phi \prod_{n=1}^{N} \phi_{n}^{\delta_{n}-1} = \prod_{n=1}^{N} \frac{\Gamma(\delta_{n})}{\Gamma\left(\sum_{n=1}^{N} \delta_{n}\right)}, \tag{B3}
\]

where \( \int d\phi \) denotes the integral over the probability distribution \( \phi_{n=1,2,\ldots,N} \). The second factor in Eq. (B1) is computed in a similar way as

\[
P(z^{(t)}|\alpha^{(t)}, \eta^{(t)}) = \prod_{d=1}^{D} \int d\pi^{(t)}_{d} \prod_{k=1}^{K} \left(\pi^{(t)}_{dk}\right) \frac{\Gamma\left(\sum_{k=1}^{K} \eta_{k}\right)}{\Gamma\left(\eta_{k}\right)} \Gamma\left(\frac{\eta_{k} + z_{dk}}{2}\right) \text{Dir}\left(\pi^{(t)}_{dk}|\eta^{(t)}\right).
\]

Multiplying the first (B2) and the second (B4) factors, we have obtained the marginalized distribution (20) described in the main text.

The probability distribution of the latent variable \( z^{(t)} \) is calculated as

\[
P(z^{(t)} = 1|\tau^{(t)}, z_{\text{cup}}, \alpha^{(t)}, \eta^{(t)}), \tag{B5}
\]

up to a constant factor in terms of \( z^{(t)} \). Most of the contributions in the numerator and denominator of Eq. (B5) are canceled out because they are nothing but the marginalized likelihood function (20) with and without the observed data \( \tau^{(t)} \), respectively. With the identity \( \Gamma(x+1) = x\Gamma(x) \) and \( \tau^{(t)} \in \{0,1\} \), we have proved the distribution (21) for the collapsed Gibbs sampling.

By differentiating the explicit expression of the likelihood function (20), we can derive the update equation (18) for the parameter \( \alpha^{(t)} \) with the relation of the digamma function \( \Psi(x) : \Psi(x+n) - \Psi(x) = \sum_{l=1}^{n} \frac{1}{x+1-l} \) for \( n \in \mathbb{Z}^{+} \).
gence, AAAI’15 (AAAI Press, 2015) pp. 130–136.

[22] D. He, D. Jin, Z. Chen, and W. Zhang, Sci. Rep. 5, 8638 (2015).

[23] H. Okamoto and X. Qiu, in The 7th International Conference on Complex Networks and Their Applications (Cambridge, UK, 2018) p. 59.

[24] H. Okamoto and X. le Qiu (2019) arXiv:1909.07066.

[25] C. M. Bishop, Pattern Recognition and Machine Learning (springer, 2006).

[26] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda, in Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI’09 (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2009) pp. 1427–1432.

[27] T. P. Minka, “Estimating a Dirichlet distribution,” https://tminka.github.io/papers/dirichlet/minka-dirichlet.pdf (2000).

[28] A. Lancichinetti, S. Fortunato, and J. Kertész, New J. Phys. 11, 033015 (2009)

[29] A. F. McDaid, D. Greene, and N. Hurley, preprint (2011) arXiv:1110.2515

[30] A. Lancichinetti, S. Fortunato, and F. Radicchi, Phys. Rev. E 78, 046110 (2008).

[31] S. Fortunato and M. Barthelemy, Proc. Natl. Acad. Sci. 104, 36 (2007).

[32] We note that the convergence of the distributions is not guaranteed in general because of the stochastic optimization of the parameters.

[33] D. M. Blei, A. Y. Ng, and M. I. Jordan, J. Mach. Learn. Res. 3, 993 (2003).

[34] D. M. Blei and J. D. Lafferty, in Proceedings of the 23rd International Conference on Machine Learning ICML ’06 (ACM, New York, NY, USA, 2006) pp. 113–120.

[35] E. Ravasz and A.-L. Barabási, Phys. Rev. E 67, 026112 (2003).