Layer reconstruction and missing link prediction of a multilayer network with maximum *a posteriori* estimation

Junyao Kuang* and Caterina Scoglio
Department of Electrical and Computer Engineering, Kansas State University, Manhattan, Kansas 66506, USA

(Received 7 January 2021; revised 22 May 2021; accepted 16 July 2021; published 2 August 2021)

From social networks to biological networks, different types of interactions among the same set of nodes characterize distinct layers, which are termed multilayer networks. Within a multilayer network, some layers, confirmed through different experiments, could be structurally similar and interdependent. In this paper, we propose a maximum *a posteriori*-based method to study and reconstruct the structure of a target layer in a multilayer network. Nodes within the target layer are characterized by vectors, which are employed to compute edge weights. Further, to detect structurally similar layers, we propose a method for comparing networks based on the eigenvector centrality. Using similar layers, we obtain the parameters of the conjugate prior. With this maximum *a posteriori* algorithm, we can reconstruct the target layer and predict missing links. We test the method on two real multilayer networks, and the results show that the maximum *a posteriori* estimation is promising in reconstructing the target layer even when a large number of links is missing.

DOI: 10.1103/PhysRevE.104.024301

I. INTRODUCTION

Network science has been widely used in different areas, such as information diffusion, infectious disease spread, and gene coexpression analysis. Through network analysis, one can study the relations among nodes and the robustness of a system [1,2]. For example, epidemiologists can predict the number of people infected by COVID-19 through epidemic analysis. Then, they can provide advice to policymakers at an early stage to curb the spreading of the disease [3]. By constructing coexpression networks, biologists can discover crucial genes (nodes) by simply choosing genes with high degree centralities, closeness centralities, or eigenvector centralities. In a typical protein network, five to seven layers are considered to represent different types of molecular interactions [4,5], including proteolysis, genetic interaction, coexpression, etc. Such multilayer networks are obtained through biological experiments, which could be expensive and time-consuming. A layer can be particularly important but also incomplete, with many missing links. A critical research goal is to reconstruct this important layer, called the target layer, without performing the expensive experiments, but by exploiting all information embedded in the other layers. In other words, we can estimate the target layer through existing layers [4,6–9]. After constructing the target layer, researchers can devote restricted resources to the detection of edges with high probabilities.

Various methods have been proposed to reconstruct networks and predict missing links, and most of them are based on generative models [4,6,7,10–24]. A generative model reconstructs the network topology by fitting a stochastic network model [7,13,14,25], and uses a maximum-likelihood estimation (MLE) algorithm to find the optimal parameters that can describe the network. In Refs. [13,14], the authors presented a degree-correlated stochastic block model to reconstruct a single layer network. Edges are computed through the tensor product of node vectors, and the entries of the node vectors are the degrees of the nodes in each community. Authors of Ref. [7] extended the single-layer stochastic block model to multilayer networks and used it to predict missing links and detect overlapping communities. The authors tried all the layer combinations to find the layers that can improve the maximum likelihood. However, when there are many layers, it is burdensome to try all the layer combinations to find the interdependent layers. The authors validated the algorithm through two real multilayer networks by hiding 20% of links and nonlinks.

In multilayer networks, there could be structurally similar layers. Therefore, it is possible to take advantage of similar layers to help reconstruct the target layer. We propose comparing the target layer with the remaining layers if the target layer is partially known. In the literature, multiple methods have been proposed to compare networks [26–30]. The authors in Refs. [27,28] present a method called DeltaCon. The DeltaCon method compares the affinity scores of every pair of nodes in two networks. The method is very sensitive to changes in the number of edges, and the removal of edges results in a significant change in the distance. References [29,30] review and compare some network-comparing methods, including vertex-edge overlapping, vertex-edge vector similarity, and the SimHash algorithm. The vertex-edge overlapping method applies the rule that two graphs are similar if they share many vertices and edges. According to the analysis in Ref. [29], the drawbacks of this method are that it is not sensitive to changes in high-quality vertices, topology, and properties of networks. The vertex-edge vector similarity method compares the node-edge weight vectors of two

*kuang@ksu.edu

networks. The drawback of this method is that it is not sensitive to changes in the topology and other properties of networks. To take advantage of the features of networks, the SimHash algorithm is introduced to compare networks. The PageRank [31] together with edges are used as network features in SimHash algorithm to compare web page networks.

In this paper, we propose a *maximum a posteriori* (MAP)-based method for target layer reconstruction as well as for link prediction. The MLE algorithm and entropy-related approaches must depend on the known information of the target layer. Consequently, the reconstruction is significantly affected by the available information of the target layer. In the MAP algorithm, the layers that are similar to the target layer will be considered to compute the parameters of the conjugate prior. Experimental results show that the MAP algorithm provides more consistent results than the MLE method. The first contribution of this paper is to discover an incomplete target layer by computing its edges through a dot product of node vectors. The optimal entries of node vectors are obtained by maximizing the posterior probabilities of the stochastic model. In our experiments, we find that the model accuracy can be improved if we increase the dimension of node vectors, but the return is diminishing for large vector dimensions. Another contribution is that we introduce the eigenvector centrality-based SimHash algorithm to detect structurally similar layers (interdependent layers). The eigenvector centralities of nodes are extracted as network features, which allow us to recover the structure of networks, as shown in the experimental results. In this work, we assume that the number of edges between any pair of nodes follows Poisson distribution [7,13]. Hence, the gamma distribution will be the conjugate prior for the Poisson distribution [32]. We compute the parameters of the conjugate prior through the adjacency matrices of similar layers, and the contributions of the similar layers are weighted by their similarities. The number of edges between each pair of nodes is calculated as the dot product of the node vectors. In our experiments, we show that similar layers are critical in improving the robustness of link predictions.

The paper is organized as follows. In Sec. II, we first introduce the MAP method on target layer reconstruction. Next, we propose the eigenvector centrality-based SimHash algorithm to find structurally similar layers. Then, we propose the process for identifying parameters of the conjugate prior under different circumstances. In Sec. III, we first evaluate the eigenvector centrality-based SimHash algorithm on two real multilayer networks. Then, we evaluate the MAP algorithm-based target layer reconstruction on the two real networks and compare the differences between the MLE algorithm and the MAP algorithm. We conclude the paper in Sec. IV.

II. LAYER RECONSTRUCTION IN MULTILAYER NETWORKS

A. Maximum *a posteriori*-based stochastic model

In this section, we define the stochastic model for both directed and undirected multilayer networks. The adjacency matrix of the target layer is denoted by A. The goal of this reconstruction is to estimate A, given a partial knowledge of the target layer and of other layers in the multilayer network. To reconstruct the target layer, a set of parameters is needed to describe the model, which we denote as θ. Based on Bayes' theorem, the posterior probability of θ is

$$P(\theta \mid A) = \frac{P(A \mid \theta)P(\theta)}{P(A)}, \tag{1}$$

where $P(\theta \mid A)$ is the posterior probability of θ, $P(A \mid \theta)$ is the likelihood of A under θ, $P(\theta)$ is the prior probability of θ and $P(A)$ is the marginal likelihood that contains all the information of the network. Since $P(A)$ is a constant, $P(\theta \mid A)$ is proportional to the product of $P(A \mid \theta)$ and $P(\theta)$. Therefore, we have

$$P(\theta \mid A) \propto P(A \mid \theta)P(\theta). \tag{2}$$

For any pair of nodes in the network, we use E_{ij} to denote the expected number of links (which could be fractional) between node i and node j. In unweighted networks, the entries of the adjacency matrix are denoted by 0 or 1. Here, since the entries E_{ij} are real numbers, we can interpret network A as a weighted network.

Before we substitute any parameters into Eq. (2), we make the following assumptions. The links in the target layer are independent and identically distributed. In other words, the number of edges between node i and node j does not affect the relation between node i and node k. Further, we assume the number of links between any pair of nodes is extracted from a Poisson distribution, i.e., $P(A_{ij} \mid E_{ij}) = \frac{e^{-E_{ij}}(E_{ij})^{A_{ij}}}{A_{ij}!}$. We can rewrite Eq. (2) after substituting E_{ij} and A_{ij} as

$$P(\theta \mid A) \propto \prod_{i,j} \frac{e^{-E_{ij}}(E_{ij})^{A_{ij}}}{A_{ij}!} P(E_{ij}). \tag{3}$$

In the MLE algorithm, the prior probability $P(E_{ij})$ can be neglected since it is a constant. In the MAP algorithm, we need to specify the prior distribution of $P(E_{ij})$. The conjugate prior distribution for the Poisson distribution is the gamma distribution

$$\begin{aligned} P(E_{ij}) &= \frac{\beta_{ij}^{\alpha_{ij}}}{\Gamma(\alpha_{ij})} E_{ij}^{\alpha_{ij}-1} e^{-\beta_{ij}E_{ij}} \\ &\propto E_{ij}^{\alpha_{ij}-1} e^{-\beta_{ij}E_{ij}}, \end{aligned} \tag{4}$$

where α_{ij}, β_{ij}, and $\Gamma(\alpha_{ij})$ are the shape parameter, the scale parameter, and the gamma function of the gamma distribution, respectively. In Sec. II C, we introduce a procedure to determine α_{ij} and β_{ij} through the layers with high similarities. Substituting the conjugate gamma distribution into Eq. (3), we have

$$P(\theta \mid A) \propto \prod_{i,j} e^{-(\beta+1)E_{ij}} (E_{ij})^{A_{ij}+\alpha-1}. \tag{5}$$

Note that we have left out constant terms.

The problem now has been simplified to finding the parameters E_{ij} that can maximize the posterior probability. However, an expression for E_{ij} has not been specified. In this work, we compute the links through node vectors. The nodes in the target layer are represented by vectors. The expected

number of links E_{ij} can be computed by

$$E_{ij} = \sum_{z=1}^{K} s_{iz} t_{jz}, \tag{6}$$

where s_{iz} and t_{jz} are, respectively, the zth entry of node i's vector and node j's vector. Here, we use s and t to denote source and target nodes. K is the dimension of the vector. Some MLE related works use tensor factorization to decompose the links, and the dimension of the tensor is interpreted as the number of overlapping communities. As a result, there should be an optimal number of communities that can maximize the estimation accuracy. However, this cannot be rigorously achieved and the tensor factorization can be simplified to the dot product according to our analysis in Appendix A.

Equation (5) is still intractable after substituting E_{ij} with Eq. (6). We take the logarithmic form of Eq. (5), which gives

$$\begin{aligned} L(\theta \mid A) &= \sum_{i,j} [(A_{ij} + \alpha_{ij} - 1) \log E_{ij} - (\beta_{ij} + 1) E_{ij}] \\ &= \sum_{i,j} \Bigg[(A_{ij} + \alpha_{ij} - 1) \log \sum_{z}^{K} s_{iz} t_{jz} \\ &\quad - (\beta_{ij} + 1) \sum_{z}^{K} s_{iz} t_{jz} \Bigg], \end{aligned} \tag{7}$$

where $L(\theta \mid A)$ is the log posterior.

To find the maximized posterior for Eq. (7), we apply the Jensen's inequality $\log \bar{x} \geqslant \overline{\log x}$, which gives

$$\begin{aligned} \log \sum_{z}^{K} s_{iz} t_{jz} &= \log \sum_{z}^{K} q_{ijz} \frac{s_{iz} t_{jz}}{q_{ijz}} \\ &\geqslant \sum_{z}^{K} q_{ijz} \log \frac{s_{iz} t_{jz}}{q_{ijz}} \\ &= \sum_{z}^{K} q_{ijz} (\log s_{iz} t_{jz} - \log q_{ijz}). \end{aligned} \tag{8}$$

The equality is satisfied when

$$q_{ijz} = \frac{s_{iz} t_{jz}}{\sum_z s_{iz} t_{jz}}. \tag{9}$$

After substituting Eqs. (8) and (9) into Eq. (7), Eq. (7) can be simplified to

$$\begin{aligned} L(\theta \mid D) = \sum_{i,j,z} [&(A_{ij} + \alpha_{ij} - 1) q_{ijz} \log s_{iz} t_{jz} \\ &- (\beta_{ij} + 1) s_{iz} t_{jz}]. \end{aligned} \tag{10}$$

Taking the derivative of Eq. (10) and equating to zero, we obtain the values of s_{iz} and t_{iz}, the optimal node vectors that maximize the posterior probability:

$$s_{iz} = \frac{\sum_j (A_{ij} + \alpha_{ij} - 1) q_{ijz}}{\sum_j (\beta_{ij} + 1) t_{jz}}, \tag{11}$$

$$t_{jz} = \frac{\sum_i (A_{ij} + \alpha_{ij} - 1) q_{ijz}}{\sum_i (\beta_{ij} + 1) s_{iz}}. \tag{12}$$

The procedure to obtain the optimized s_{iz} and t_{jz} is to assign random initial values for s_{iz} and t_{jz}, then update Eqs. (9), (11), and (12) iteratively until Eq. (7) converges. However, before applying the above iteration, we need to identify α_{ij} and β_{ij}, which are introduced in Secs. II B and II C.

B. Similarity and layer comparison

In Sec. II A, we have detailed the procedure to compute the optimal node vectors by maximizing the posterior probability. The parameters α_{ij} and β_{ij} for the prior distribution are required to perform the posterior probability maximization. We propose to compute α_{ij} and β_{ij} through the layers of the multilayer network that are similar to the target layer. Keep in mind that there are missing links in the target layer, and the percentage of missing links is not known at all. Therefore, the primary factor for an effective network-comparing method is that the method must not be significantly affected by the percentage of missing links.

Networks can be characterized by multiple types of centralities, such as, degree centrality, eigenvector centrality, closeness centrality. The degree centrality measures the importance of a node by capturing the number of links the node has, while the eigenvector centrality can be regarded as an extension of degree centrality in which node's importance is also affected by its neighbors' importance. The closeness centrality of a node is the average length of shortest paths between the node and all other nodes. One or more of the centralities can describe the features of a network. To compare the target layer with the other layers, we can compare the features of the layers. For our purpose, the eigenvector centrality is selected as the network feature and is used in the SimHash algorithm to compute similarities.

The SimHash algorithm works as follows [29,30]. The feature of a network can be expressed as a set of token-value pairs $\{(v_i : w_i)\}$, where v_i is a node and w_i is its measure under the feature, for example its eigenvector centrality. Note that the target layer and the layer to be compared have the same set of tokens. In cryptography, any messages can be encrypted to a unique binary number (digest). Similarly, we can represent each token with a unique binary number with ϕ bits ($2^\phi >$ the number of v_i). For each binary number (digest), we map every 1 to w_i and 0 to $-w_i$. Thus, each token is mapped to a weighted digest with ϕ digits. To obtain the weighted digest of the network, we sum up all the weighted digests of the tokens. Note that there is no carry in the summation.

To measure the similarities between the target layer and the other layers, we can compare the digests of the layers. A simple way to compare the digests is to convert the weighted digests to binary digests. The binary digest of a network can be obtained by setting positive digits to 1 and negative digits to 0. The similarity between the target layer m and any other layer r can be measured by

$$\mu_{m,r} = 1 - \frac{\text{Hamming}\left(H_m^d,\ H_r^d\right)}{\phi}, \tag{13}$$

where $\mu_{m,r}$ is the similarity between layer m and r, H_m^d is the binary digest of the target layer, and H_r^d is the binary digest of the layer to be compared, respectively.

An alternative way to obtain the similarities is by computing the Pearson correlation coefficient of the weighted digests.

The estimation results based on the Pearson correlation are shown in Appendix B. In this work, we measure the similarity between the target layer and the other layers through the binary-based digests. The influence of the bit number is discussed in Appendix C.

C. Identify the parameters of the gamma distribution prior

Finally, we introduce the procedure to determine the parameters α_{ij} and β_{ij} of the conjugate prior. We discuss this problem in two cases.

In the first case, we assume the structure of the target layer is partially known, i.e., the entries of the adjacency matrix are partially known. In this case, we apply the layer comparison method introduced above, and the L' layers with highest similarities are considered to identify the parameters of conjugate priors. The parameters of the gamma distribution prior are computed as

$$\alpha_{ij} = \max\left(\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r,\ 1\right),$$
$$\beta_{ij} = \max\left(\sum_{r=1}^{L'} \mu_{m,r},\ \frac{\sum_{r=1}^{L'} \mu_{m,r}}{\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r}\right), \tag{14}$$

where m denotes the target layer, $\mu_{m,r}$ is the similarity between the target layer m and any layer r. A_{ij}^r is the adjacency matrix of layer r in L'. In Eqs. (11) and (12), since A_{ij} could be zero, if α_{ij} is less than one, we obtain a negative s_{ij}. Thus, we need to limit the range of α_{ij}. If $\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r \in (0, 1)$, then we set $\alpha_{ij} = 1$, and set β_{ij} to $\frac{\sum_{r=1}^{L'} \mu_{m,r}}{\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r}$ to maintain the means of the gamma distribution prior unchanged. If $\sum_{r=1}^{L'} \mu_{m,r} A_{ij}^r = 0$, then we will set $\alpha_{ij} = 1$, and set β_{ij} equal to a large number to ensure the MAP algorithm converges.

A special case is the structure of the target layer is not known at all. In this case, the comparison between the target layer and the other layers is not feasible. In this case, an alternative and heuristic way to compute the parameters of conjugate prior can be based on the functionally similar layers. If available, then we can use additional published networks and data as a new multilayer network, in which layers are functionally similar. Thereafter, we can apply the proposed MAP algorithm to compute the parameters of conjugate prior through this new multilayer network. The similarities between the target layer and the functionally layers cannot be determined through any network comparison algorithm. Thus, we assume the similarities are all ones. We assign the parameters of the gamma distribution as

$$\alpha_{ij} = \sum_{r=1}^{L'} A_{ij}^r,$$
$$\beta_{ij} = L'. \tag{15}$$

Similarly, if $\sum_{r=1}^{L} A_{ij}^r < 1$, then we will set $\alpha_{ij} = 1$, and set β_{ij} to a large number to make the MAP algorithm converge.

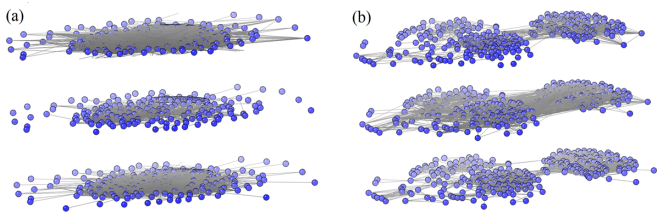

FIG. 1. Three layers of the FAO (a) network and HVR (b) network. Nodes in different layers share the same plane coordinates.

In this scenario, we do not have any structural information regarding the target layer. If we set $A_{ij} = 0$ in Eqs. (11) and (12), then this is equivalent to assuming zero presences of all the edges in the target layer. To avoid this issue, we take the entries of A_{ij} as the ratio of α_{ij} and β_{ij}, i.e., $A_{ij} = \alpha_{ij}/\beta_{ij}$. The entries are assigned as the average presences over the other layers.

III. EXPERIMENTAL VALIDATION

In this section, we evaluate the method introduced in Sec. II. First, the SimHash algorithm is applied on two real networks, in which different percentages of links are removed uniformly at random. Second, the effectiveness of the MAP algorithm for the two real multilayer networks is evaluated, and the MAP algorithm is compared to the MLE algorithm.

The first real network we use to evaluate our proposed MAP algorithm is the Food and Agriculture Organization (FAO) trade network [33]. The FAO multilayer network is composed of 364 layers, and each layer represents a product trading among 214 countries. A link is detected between two nodes in a layer if there is trading of the corresponding product between the two countries. We show three layers of the FAO network in Fig. 1 through the KiNG software [34]. Since the layers in the FAO network are not ordered in any particular way, in the experiments, we only perform the evaluation by assuming that the target layer is one of the the first nine layers of the FAO network. Note that all the 363 remaining layers are always used in our experiments for detecting the similar layers.

The second real network (HVR network) we use to evaluate the proposed MAP algorithm has nine layers and 307 nodes [35], which represent malaria parasite genes. The nine layers correspond to nine highly variable regions on the genes themselves. An edge is detected if two genes share an exact match of significant length in the highly variable region. We show three layers of the HVR network in Fig. 1.

A. Numerical results on layer comparison

The evaluation of the SimHash algorithm is performed on both the FAO network and the HVR network. The binary digest of each layer is obtained through the SimHash algorithm based on the eigenvetor centrality of the nodes. The similarity of any two layers can be computed through Eq. (13). In Fig. 2, each layer is, respectively, set as the target layer, and the similarities between the target layer and all the other layers are shown in interquartile ranges (IQR). For the FAO network, we only show the results of the first nine layers as

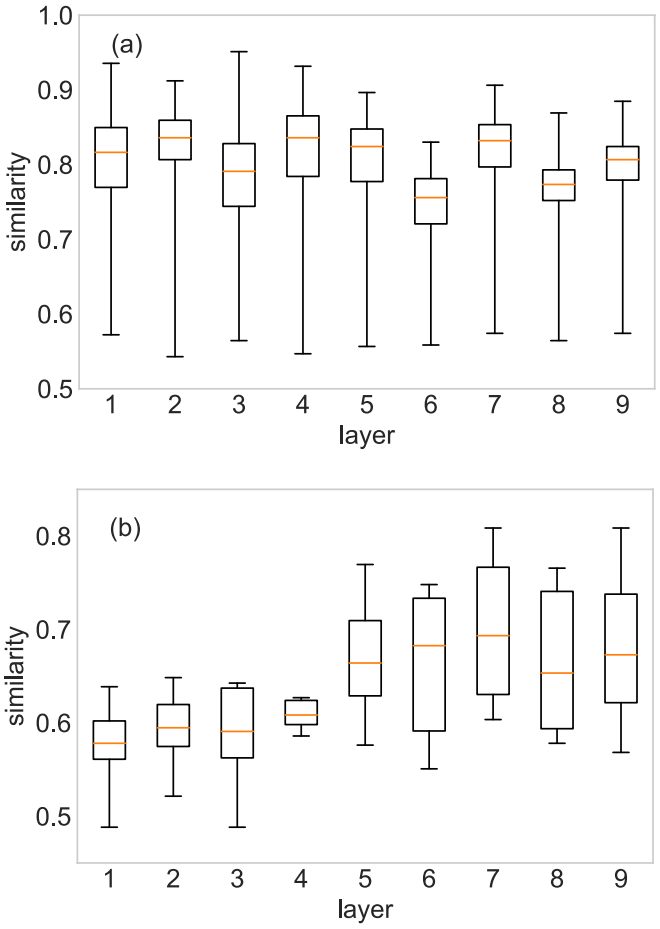

FIG. 2. Layer comparison of the FAO network (a) and HVR network (b). (a) shows the similarities between the target layer and all the other 363 layers in the FAO network. (b) shows the similarities between the target layer and the other eight layers in the HVR network. Note that each similarity value is averaged over 10 runs.

target layers, and each IQR bar contains 363 similarity values. The similarities obtained through the SimHash algorithm are between 0 and 1. A zero similarity means the two layers have totally opposite eigenvector centrality distribution, while for 0.5 similarity the two layers are independent. A similarity approaching one means the two layers are similar. In the FAO network, the similarities are between 0.5 and 1, which indicates there are no layers with totally opposite eigenvector centrality distribution. In the HVR network, most of the layers are independent, since the similarities are all less than 0.8, except layers 7 and 9.

The network-comparing method must be able to discover structurally similar layers even if there are missing links in the network. To this end, we uniformly at random remove 20%, 40%, 60%, and 80% of edges from the target layer to generate incomplete networks and compare the incomplete networks with the target network. In Fig. 3(a), each layer of the FAO network (first nine layers) and HVR network is set as target network, respectively, and we randomly remove 20%, 40%, 60%, and 80% of edges from the target layer. The similarities between the incomplete networks with removed edges and the target layer are computed. From the top panels, we observe that the similarities are greater than 0.8 even with 80% of edges randomly removed.

There are always differences between the target layer and similar layers. In the second experiment, we show that missing links in the target layers do not affect the similarity significantly between the target layer and its similar layers. For each of the target layers, we choose five most similar layers from all the other layers, we then randomly remove 20%, 40%, 60%, and 80% of edges from the target layer to generate incomplete networks. The incomplete networks are compared to the five similar layers, and we can obtain five similarities for different removal percentages. In the bottom panels, we show the average of the five similarities. The results for the first nine layers of the FAO network are shown in Fig. 3(c). We observe that the similarities decrease slightly even when 80% of edges are randomly removed. For the HVR network, where the layers are heterogeneous, though we randomly remove different percentages of links, the similarities are still maintained at low levels.

B. Validation of layer reconstruction and link prediction

The MAP method we have introduced in this work has the goal of layer reconstruction and missing link estimation. The similarity between the target layer and other layers can be obtained through the SimHash algorithm as introduced above. In this part, we show the effectiveness of the MAP algorithm, and compare the differences between the MAP algorithm and the MLE algorithm performance. Other methods such as entropy-based approaches, are equivalent to the MLE algorithm and are discussed in Appendix D.

Here, we use the receiver-operator characteristic (ROC) curve and the area under the curve (AUC) to evaluate the effectiveness of our method. The model is perfect when the AUC is approaching one, and 0.5 means the model guesses the edge weights randomly.

a. The dimension of node vectors. The dimension of node vectors is a critical factor for the MAP algorithm. In experiments, we find that there is no such number of communities [36–40] that can maximize the estimation accuracy. For both the FAO network and HVR network, we remove 40% of edges from the target layer to generate incomplete networks. The five layers with highest SimHash similarities are used to compute the parameters of the conjugate prior. The target layer is reconstructed through both the MLE algorithm and the MAP algorithm. In Fig. 4, we can observe an increased estimation accuracy with respect to the increment of the number of dimensions, though the gain diminishes for dimensions greater than 40. In the following experiments, we use a dimension of 50 which balances running time and estimation accuracy. Apart from the dimension of node vectors, the estimation accuracy is also affected by the number of similar layers. The influence of the number of similar layers on the estimation accuracy is analyzed in Appendix E.

b. Comparison between the MLE algorithm and the MAP algorithm. Both the MLE algorithm and the MAP algorithm have their own benefits and disadvantages. The major difference between the MAP and MLE methods is that the MAP method can incorporate prior information (other similar layers), while the MLE relies on the available information of

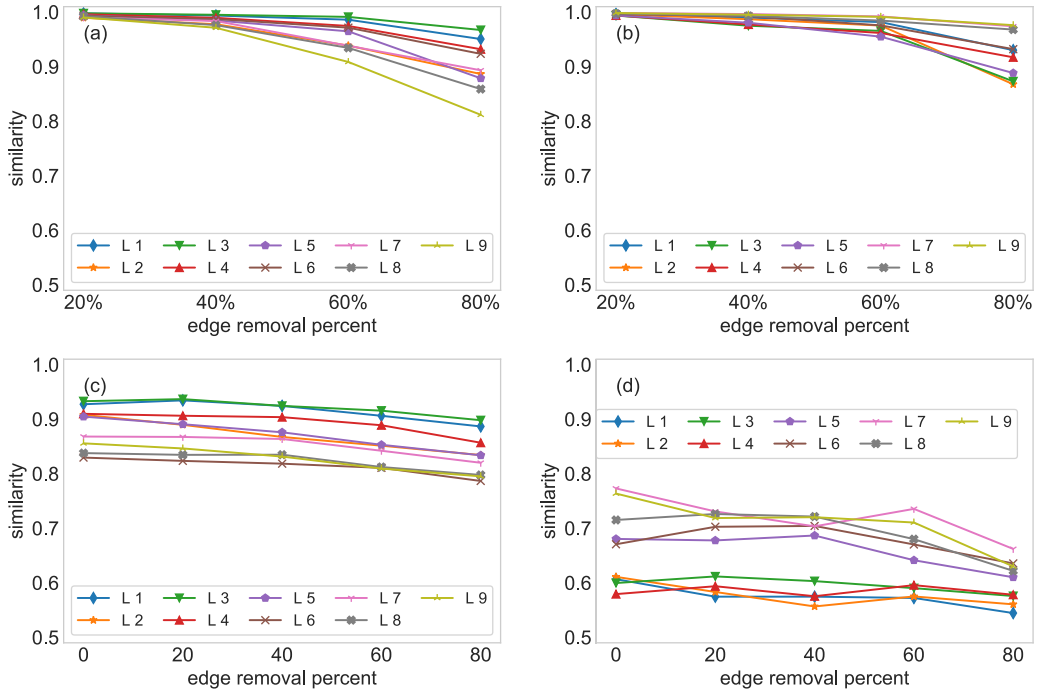

FIG. 3. Eigenvector centrality-based SimHash algorithm. Panel (a) shows the comparison between the first nine layers of the FAO network and their incomplete counterparts, where 20%, 40%, 60%, and 80% edges are removed uniformly at random from the target layer. Panel (b) shows the same experiment on the HVR network. Panel (c) compares the reduced networks (the first nine layers of the FAO network) with the five most similar layers. The mean of the similarities is shown in the panel. Similarly, 20%, 40%, 60%, and 80% edges are removed uniformly at random from the target layer. Panel (d) shows the same experiment on the HVR network. Note that each similarity value is averaged over 10 runs.

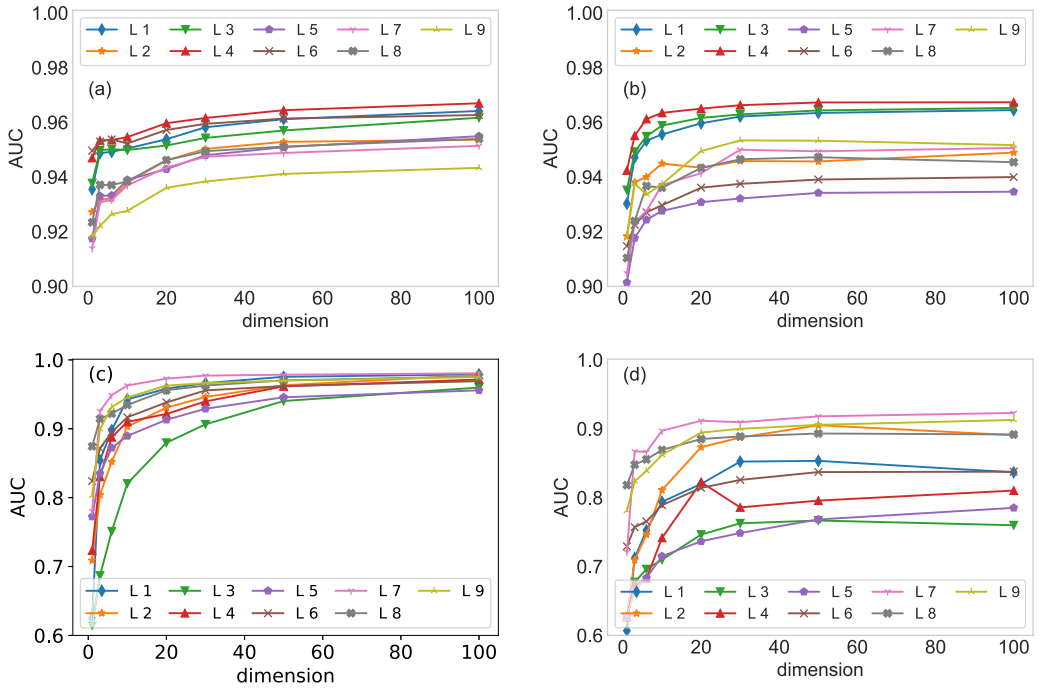

FIG. 4. AUC vs. the dimension of node vectors. Panel (a) shows the MLE method on the first nine layers of the FAO network, where each layer is set as the target layer, respectively. Panel (b) shows the results of MAP method when the five layers with highest similarities are adopted to compute the parameters of the conjugate prior. Panel (c) shows the results of the MLE method on the HVR network. Panel (d) is the results of MAP method on the HVR network. Note that the results are averaged over 10 runs of cross-validations.

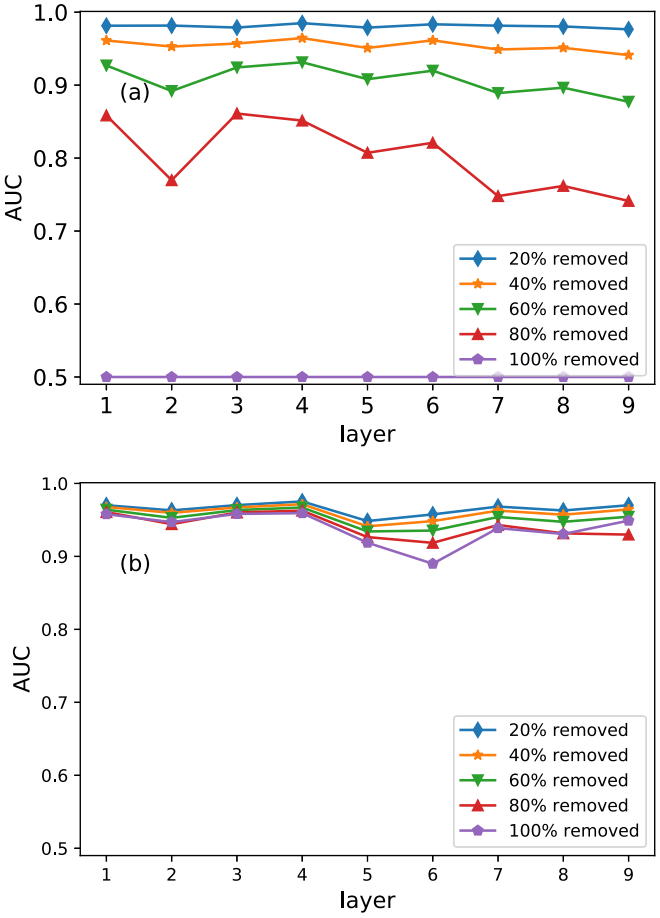

FIG. 5. Comparison of the MLE and MAP methods on the FAO network. Panel (a) shows results of the MLE method. Panel (b) shows the results of the MAP method. Results are averaged over 10 runs of cross-validations.

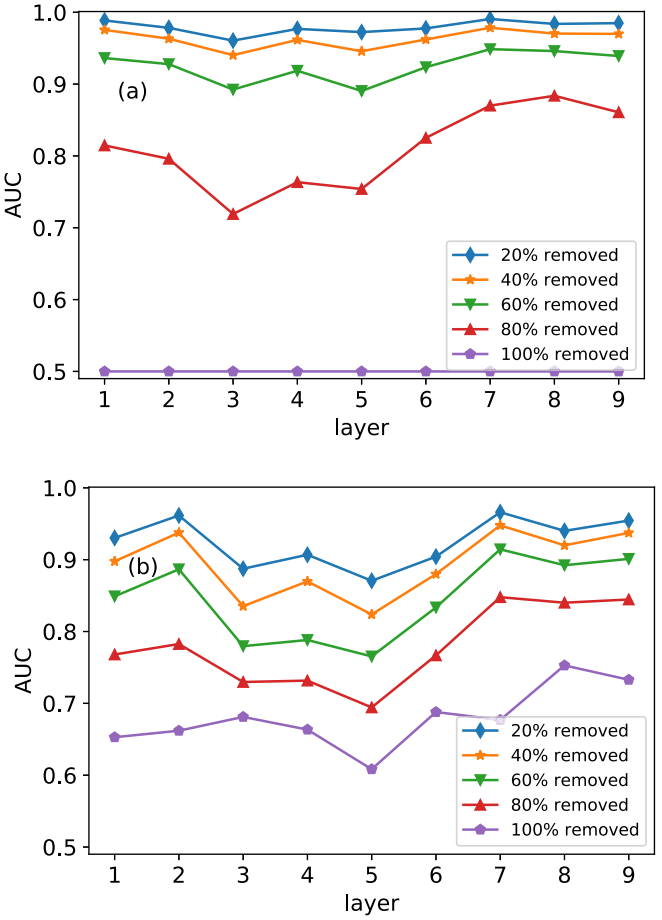

FIG. 6. Comparison of the MLE and MAP methods on the HVR network. Panel (a) shows results of the MLE method. Panel (b) shows the results of the MAP method. Results are averaged over 10 runs of cross-validations.

the target layer solely. The comparison between the MLE algorithm and the MAP algorithm is performed on both the FAO network (first nine layers) and the HVR network. The two algorithms are implemented on incomplete layers, which are generated by randomly removing 20%, 40%, 60%, 80%, and 100% of edges from the two networks. In Figs. 5(a) and 6(a), we can see that the estimation based on the MLE algorithm is significantly affected by the missing links. However, the robustness of the estimation is greatly improved after we adopt structurally similar layers to reconstruct the target layer, as shown in Fig. 5(b). On the contrary, the estimation accuracy deteriorates if we adopt layers with heterogeneous structures to reconstruct the target layer, which is shown in Fig. 6(b).

Intuitively, the MLE method reconstructs the target layer through the known information of the target layer itself. As a result, the estimation accuracy is related to the available information, i.e., the percentage of known links. In the MAP algorithm, the estimation is not only affected by the known information of the target layer but also the similar layers. In real applications, if the percent of missing links is less than 20%, it is recommended to use the MLE algorithm, since it provides more accurate results than the MAP algorithm. Conversely, the MAP algorithm is the better choice if researchers are unaware of the percentage of missing links.

IV. CONCLUSION AND FUTURE WORKS

In this paper, we present a MAP estimation-based algorithm for target layer reconstruction in multilayer networks. In multilayer networks, some layers are structurally similar; thus, we can take advantage of the similar layers to reconstruct the target layer. In Sec. II, we first derive the *maximum a posteriori* estimation for target layer reconstruction in multilayer network. Second, the eigenvector centrality-based SimHash algorithm is introduced to detect structurally similar layers. The SimHash algorithm compares network features, thus it is not affected by the missing links. Third, we introduce two scenarios to obtain the parameters of the gamma conjugate prior. In the first case, where the target layer is partially known, the SimHash algorithm is adopted to detect structurally similar layers. In the second case, where the target layer is not known at all, functionally similar layers is used to compute the parameters of the conjugate prior. In Sec. III, we first show that the eigenvector centrality-based SimHash algorithm is able to return consistent similarity levels for different percentages of missing links. Then, we show that the estimation accuracy can be improved by increasing the number of dimensions of node vectors, and the gain is diminishing for dimensions greater than 40 for the two networks. We find that with a great number

of similar layers, we can obtain more consistent estimation results. Finally, the MLE method and MAP method are compared on two real networks. The experimental results suggest that if there are less than 20% of missing links, the MLE method has better performance. However, if the percentage of missing links is 40% or more, the MAP method returns results that are more consistent.

However, there are still some limitations to the MAP method we present. The first limitation is that the missing links in the target layer are required to be removed uniformly at random. The similarities are obtained through network feature comparison, which means the structure of the target layer needs to be maintained. Targeted removal of links will change the structure of the network. The second limitation concerns the unknown relation between the vector dimension and network size. In our numerical experiments, we test multiple dimensions and adopt a dimension of 50, which balances running time and estimation accuracy.

Recent results in Refs. [41–44] present some methods to estimate the eigenvector centralities based on nodal data without requiring the network structure. Recall that SimHash algorithm is based on the eigenvector centrality obtained from the target layer. Therefore, estimating eigenvector centrality without constructing network is a good alternative for future work.

The MAP algorithm we present to reconstruct a target layer in a multilayer network shows promising results when we can identify the similarities between the target layer and other layers. The experimental results show that the estimations of our MAP method are less likely to be affected by missing links, which is not known in real applications. Therefore, our MAP method can be used to direct experiments, especially when there is no information about the target layer.

ACKNOWLEDGMENT

This work has been supported by the National Institutes of Health under Grant No. R01AI140760.

APPENDIX A: TENSOR FACTORIZATION

In Refs. [7,13,14], nodes are factorized by membership vectors. The dimension of membership vectors is interpreted as the number of overlapping communities. In addition, each edge is computed through the tensor product of the membership vectors. The number of communities is obtained by maximizing the likelihood. Vectorization is also used in some natural language processing algorithms [45–49] in which words are embedded as vectors to preserve their relations to contexts. However, the dimension of word vectors does not have any semantic meaning, rather the choice of the vector dimension is first affected by the data set size. According to Refs. [46,49], larger dimensions can improve model accuracy, but the gain diminishes for vectors larger than 200 dimensions. The choice of dimension is also related to the available resources, and it is better to reduce the dimension as long as the choice does not affect the estimation accuracy substantially. In fact, the tensor factorization can be simplified to dot product, we prove this point as follows.

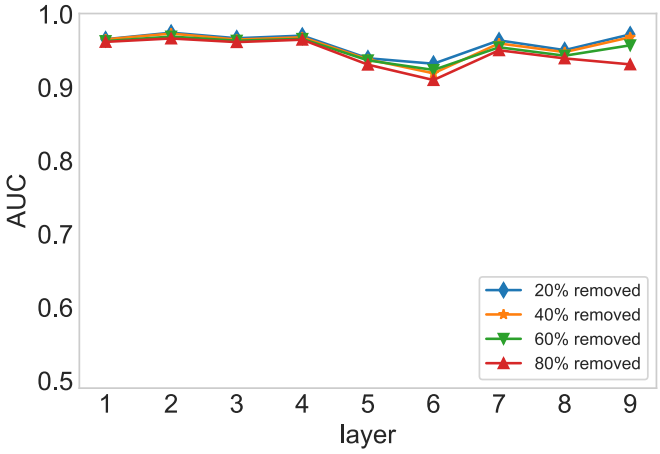

FIG. 7. The estimation results based on the Pearson correlation of the weighted digest.

We assume the number of edges between any two nodes is the tensor product of two node vectors and the control matrix, i.e., $E_{ij} = \sum s_{ik} t_{jl} w_{kl}$. w_{kl} (k is not equal to l) is the parameter that controls the edges from any source node i (in community k) to any target node j (in community l). Then, we have

$$\begin{aligned} E_{ij} &= \sum s_{ik} t_{jl} w_{kl} \\ &= s_{ik} t_{jk} w_{kk} + s_{ik} t_{jl} w_{kl} + s_{il} t_{jl} w_{ll} \\ &\quad + \left(\sum s_{im} t_{jn} w_{mn} - s_{ik} t_{jk} w_{kk} - s_{ik} t_{jl} w_{kl} - s_{il} t_{jl} w_{ll} \right). \end{aligned} \tag{A1}$$

In Eq. (A1), we denote the sum of the first three terms as E_{ij}^{kl}. Then, we can incorporate w_{kk} into s_{ik}, and denote it as $s'_{ik} = s_{ik} w_{kk}$. Similarly, $s'_{il} = s_{il} w_{ll}$. Therefore, we have

$$\begin{aligned} E_{ij}^{kl} &= s'_{ik} t_{jk} + s'_{ik} t_{jl} \frac{w_{kl}}{w_{kk}} + s'_{il} t_{jl} \\ &= s'_{ik} t_{jk} + \left(s'_{ik} \frac{w_{kl}}{w_{kk}} + s'_{il} \right) t_{jl}. \end{aligned} \tag{A2}$$

Since nodes between different communities are loosely connected, we have $w_{kl} < w_{kk}$, and we assume $\epsilon = w_{kl}/w_{kk}$. Then, Eq. (A2) can be written as

$$\begin{aligned} E_{ij}^{kl} &= s'_{ik} t_{jk} + (\epsilon s'_{ik} + s'_{il}) t_{jl} \\ &= s'_{ik} t_{jk} + s''_{il} t_{jl}, \end{aligned} \tag{A3}$$

where $s''_{il} = \epsilon s'_{ik} + s'_{il}$. For every intercommunity edge originating from node i in community k, we can incorporate it into community l by incrementing ϵs_{ik} to s'_{il}. Edges can be factorized through the dot product of node vectors. Therefore, the tensor factorization is equivalent to the dot product.

APPENDIX B: ESTIMATION BASED ON THE PEARSON CORRELATION

With the SimHash algorithm, the weighted digest of each layer can be obtained. The similarity between any two layers can be measured by computing the Pearson correlation of the weighted digests. In Fig. 7, we show the estimation results based on the Pearson correlation coefficient. We observe that

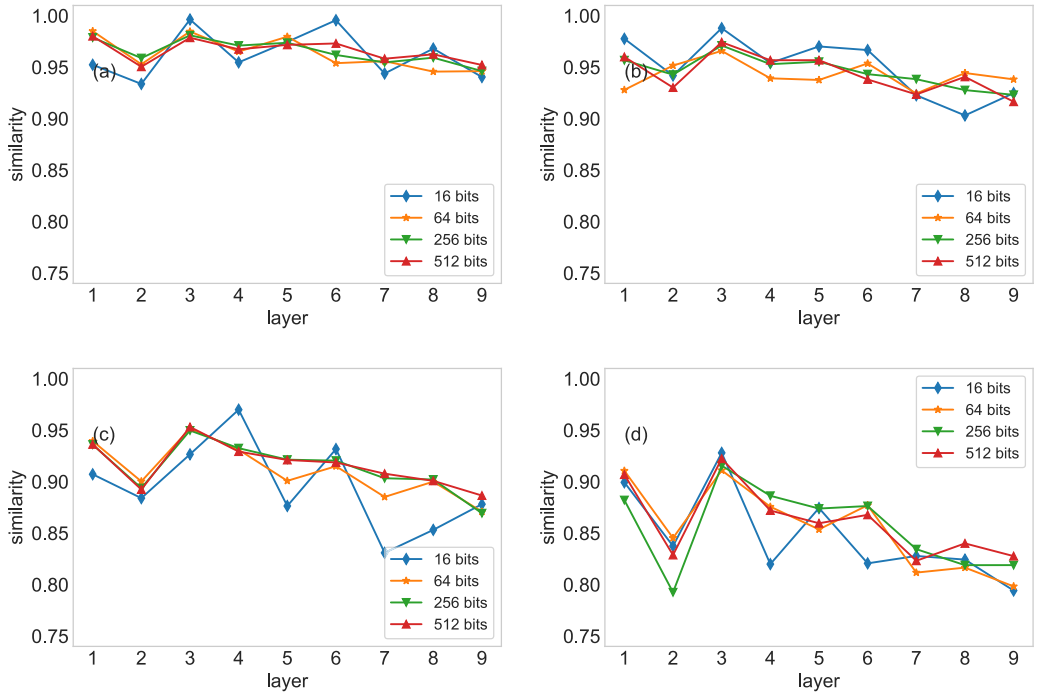

FIG. 8. Comparison of varying the number of digits (based on the Hamming distance). The four panels show the results based on 20% (a), 40% (b), 60% (c), and 80% (d) of edges randomly removed.

the estimation is as consistent as the results based on the Hamming distance. Therefore, the correlation can be an alternative for the Hamming distance.

APPENDIX C: THE CHOICE OF THE NUMBER OF BITS

The choice of ϕ affects the resolution and the consistency of the similarity. If the number of tokens is large, then it is recommended to employ large ϕ. In our work, the sizes of the two networks are not too large, so we adopt $\phi = 512$ bits. In Fig. 8, we validate this choice on the FAO network. We set each of the first nine layers of the FAO network as the target network (original network), and randomly remove 20%, 40%, 60%, and 80% of edges from the target network to generate incomplete networks. We then compare the incomplete networks with the original network. In the experiments, we adopt $\phi = 16,\ 64,\ 256$, and 512 digits, respectively. We can observe that the similarities obtained with 256 digits are close to that obtained with 512 digits, while the similarities obtained with 16 and 64 digits vary remarkably.

APPENDIX D: COMPARISON BETWEEN THE MLE ALGORITHM AND ENTROPY-BASED APPROACHES

In Sec. III B, we compared the MAP algorithm with the MLE algorithm. In fact, the entropy-based approaches are equivalent to the MLE algorithm. In the following, we prove that the MLE algorithm is equivalent to the entropy-based approaches.

Given a network G, we assume the number of nodes is fixed, and the edge weights are variables. Thus, we can define the entropy of a network in terms of edge weights as

$$H(G) = -\sum_G \sum_u p^U(u) \log p^U(u) = \sum_G \sum_u p^U(u) \log \frac{1}{p^U(u)}, \tag{D1}$$

where U is any edge in the network G, u is the weight of edge U. $p^U(u)$ is the precise probability of edge U with weight u. We assume for any edge U, it can take n_U values. If the n_U values follow uniform distribution, then we have $p^U(u) = \frac{1}{n_U}$. In this case, the network has the largest entropy, which means the network is totally uncertain. However, for any edge U, if there is a value u_s with $p^U(u_s) = 1$, the entropy of the network will be zero, which indicates all the edge weights in the network are known.

In our problem, the goal is to find a model to reconstruct the target layer, thus, we need a set of parameters to describe the model. If the model is close to the true network, then the entropy of the model is minimized. Hence, the problem is transformed to finding the parameters of a model with minimum entropy, i.e., reducing uncertainty.

Equation (D1) can be written in expectation form

$$H(p) = \sum_G E_{U\sim p^U(u)}\left(\log \frac{1}{p^U(u)}\right). \tag{D2}$$

Consider the Kullback-Leibler divergence (KL divergence). We assume θ is the parameter set that can describe the model, the probability of edge U with weight u is $q^U(u;\theta)$.

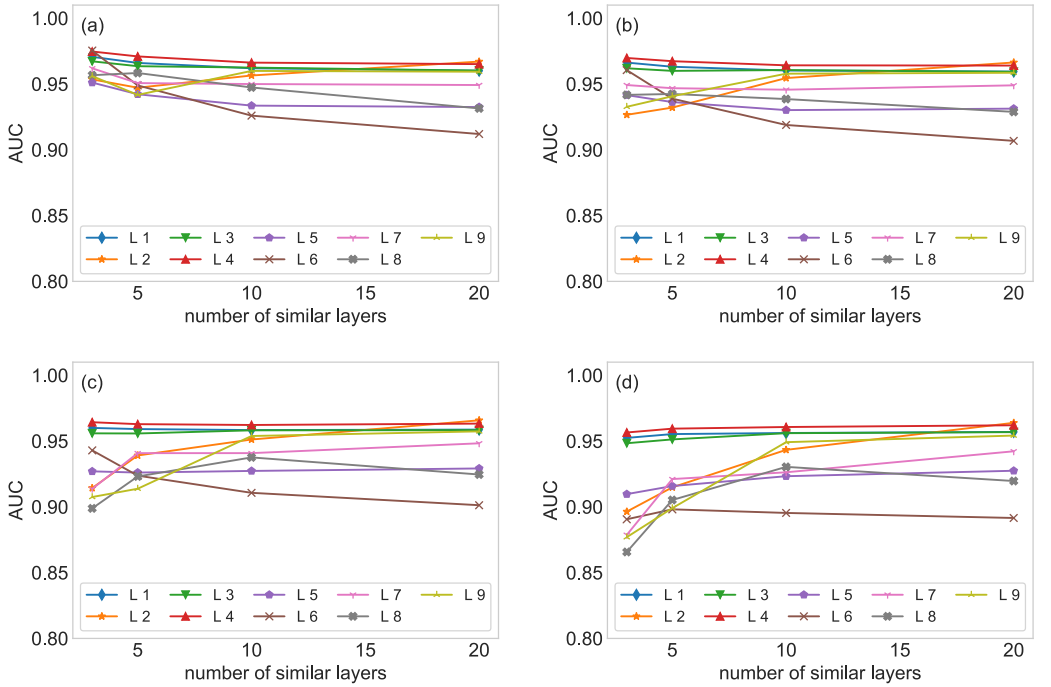

FIG. 9. The number of similar layers on the estimation results. 20% (a), 40% (b), 60% (c), and 80% (d) of edges are randomly removed from the target layer. Different numbers of similar layers are used to compute the parameters of the conjugate prior. The experiments are conducted on the first nine layers of the FAO network. Each AUC is averaged over 10 runs of cross-validations.

Thus, the relative entropy is

$$\begin{aligned} D_{kl}(p||q) &= \sum_G \sum_u p^U(u) \log \frac{p^U(u)}{q^U(u;\theta)} \\ &= \sum_G E_{U\sim p^U(u)} \left(\log \frac{p^U(u)}{q^U(u;\theta)} \right) \\ &= \sum_G E_{U\sim p^U(u)} \left(\log \frac{1}{q^U(u;\theta)} \right) \\ &\quad - \sum_G E_{U\sim p^U(u)} \left(\log \frac{1}{p^U(u)} \right). \end{aligned} \tag{D3}$$

The second term in the right-hand side is Eq. (D2) and the first term in the right-hand side is the cross entropy, which is

$$H(p,q) = \sum_G E_{U\sim p^U(u)} \log \frac{1}{q^U(u;\theta)}. \tag{D4}$$

Recall that our problem is to reconstruct the target layer and Eq. (D2) is the entropy of the target layer. Equation (D2) is determined by the data (network) solely. Thus, the problem can be simplified to minimizing the cross entropy, i.e., minimizing Eq. (D4).

Then, we consider the MLE method. The MLE algorithm is to find the parameter set θ that most likely fit a given set of data (network) D. We skip some intermediate steps and use the logarithm form directly as follows:

$$\begin{aligned} \theta &= \arg\max_\theta P(G \mid \theta) \\ &= \arg\max_\theta \sum_G \sum_u \log p^U(u;\theta). \end{aligned} \tag{D5}$$

$p^U(u;\theta)$ is the model with parameter set θ to describe the true data D. We assume that edge U can take n_U values and the n_U values follow the uniform distribution, then we have $p^U_{\mathrm{MLE}}(u) = \frac{1}{n_U}$. Introducing p^U_{MLE} to Eq. (D5) does not change the results, we have

$$\begin{aligned} \theta &= \arg\max_\theta \sum_G \sum_u \frac{1}{n_U} \log p^U(u;\theta) \\ &= \arg\max_\theta \sum_G E_{u\sim p^U_{\mathrm{MLE}}(u)} [\log p^U(u;\theta)]. \end{aligned} \tag{D6}$$

Our goal is still to find a set of parameters that can reconstruct the layer. Then, we can generalize this problem by replacing $p^U_{\mathrm{MLE}}(u)$ with $p^U(u)$, and replacing $p^U(u;\theta)$ with $q^U(u;\theta)$. The replacements can be regarded as taking real data (network) into Eq. (D7). We have

$$\begin{aligned} \theta &= \arg\max_\theta \sum_G E_{u\sim p^U(u)} \log q^U(u;\theta) \\ &= \arg\max_\theta \sum_G -E_{u\sim p^U(u)} \log q^U(u;\theta) \\ &= \arg\max_\theta \sum_G E_{u\sim p^U(u)} \log \frac{1}{q^U(u;\theta)}. \end{aligned} \tag{D7}$$

Thus, the MLE algorithm is equivalent to minimizing the cross entropy.

APPENDIX E: THE NUMBER OF SIMILAR LAYERS ON THE ESTIMATION ACCURACY

If more similar layers are employed to compute the parameters of the conjugate prior, then the influence of similar layers on the reconstruction will be promoted. Consequently,

the influence of the known part of the target layer will be down weighted. On the contrary, if we trust the known part of the target layer, we can employ less similar layers to compute the parameters of the conjugate prior. The experimental results are shown in Fig. 9, top 3, 5, 10, and 20 similar layers are adopted to compute the parameters of the conjugate prior. We observe that the estimation based on 10 and 20 similar layers are robust with respect to the missing links, while the estimation based on three and five similar layers are influenced by the missing links significantly.

[1] Q. Yang, D. Gruenbacher, J. L. Heier Stamm, G. L. Brase, S. A. DeLoach, D. E. Amrine, and C. Scoglio, Physica A **526**, 120856 (2019).

[2] Q. Yang, D. Gruenbacher, J. L. Heier Stamm, G. L. Brase, S. A. DeLoach, D. E. Amrine, and C. Scoglio, PLoS ONE **15**, e0240819 (2020).

[3] D. M. Gysi, I. Valle, M. Zitnik, A. Ameli, X. Gan, O. Varol, S. D. Ghiassian, J. J. Patten, R. A. Davey, J. Loscalzo, and A.-L. Barabási, Proc. Natl. Acad. Sci. U.S.A. **118**, e2025581118 (2021).

[4] S. Boccaletti, G. Bianconi, R. Criado, CI. del Genio, J. Gómez-Gardeñes, and M. Romance, Phys. Rep. **544**, 1 (2014).

[5] M. Kivela, A. Arenas, M. Barthelemy, JP. Gleeson, Y. Moreno, and M. A. Porter, J. Complex Netw. **2**, 203 (2014)

[6] R. Guimera and M. Sales-Pardo, Proc. Natl. Acad. Sci. USA **106**, 22073 (2009).

[7] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, Phys. Rev. E **95**, 042317 (2017).

[8] S. van Dam, U. Vosa, A. van der Graaf, L. Franke, and J. P. de Magalhaes, Brief. Bioinform. **19**, 575 (2018).

[9] F. Liesecke, J. O. De Craene, S. Besseau, V. Courdavault, M. Clastre, V. Vergès, N. Papon, N. Giglioli-Guivarc'h, G. Glévarec, O. Pichon *et al.*, Sci Rep. **9**, 14431 (2019).

[10] B. Alfredo, I. Alessandro, and M. A. Paola, J. R. Soc. Interface **16**, 20180844 (2019).

[11] X. Han, Z. Shen, W. X. Wang, and Z. Di, Phys. Rev. Lett. **114**, 028701 (2015).

[12] B. Prasse and P. Van Mieghem, arXiv:1807.08630.

[13] B. Karrer and M. E. Newman, Phys. Rev. E **83**, 016107 (2011)

[14] B. Ball, B. Karrer, and M. E. Newman, Phys. Rev. E **84**, 036103 (2011).

[15] M. E. Newman, Nat. Phys. **14**, 542 (2018).

[16] T. P. Peixoto, Phys. Rev. Lett. **123**, 128301 (2019).

[17] T. P. Peixoto, Phys. Rev. X **8**, 041011 (2018).

[18] M. E. Newman and E. A. Leicht, Proc. Natl. Acad. Sci. USA **104**, 9564 (2007).

[19] M. E. Newman, Phys. Rev. E **98**, 062321 (2018).

[20] P. Van Mieghem and Q. Liu, Phys. Rev. E **100**, 022317 (2019).

[21] T. P. Peixoto, Bayesian stochastic blockmodeling, in *Advances in Network Clustering and Blockmodeling*, edited by P. Doreian, V. Batagelj, and A. Ferligoj, Vol. 23 (Wiley, New York, 2019), p. 289.

[22] F. Parisi, G. Caldarelli, and T. Squartini, Appl. Netw. Sci. **3**, 17 (2018).

[23] L. Yin, H. Zheng, T. Bian, and Y. Deng, Physica A **482**, 699 (2017).

[24] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, Physica A **553**, 124289 (2020).

[25] T. P. Peixoto, Phys. Rev. E **97**, 012306 (2018).

[26] E. Costenbader and T. W. Valente, Soc. Netw. **25**, 283 (2003).

[27] D. Koutra, N. Shah, J. T. Vogelstein, B. Gallagher, and C. Faloutsos, ACM Trans. Knowl. Discov. Data **10**, 1 (2016).

[28] M. Tantardini, F. Ieva, L. Tajoli, and C. Piccardi, Sci. Rep. **9**, 17557 (2019).

[29] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina, J. Internet Serv. Appl. **1**, 19 (2010).

[30] M. Charikar, in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing* (ACM, New York, 2002), pp. 380–388.

[31] L. Page, S. Brin, R. Motwani, and T. Winograd, The PageRank citation ranking: Bringing order to the web, Technical Report No. 1999-66 (Stanford InfoLab, 1999), http://ilpubs.stanford.edu:8090/422/.

[32] M. C. K. Wu, F. Deniz, R. J. Prenger, and J. L. Gallant, arXiv:1811.01043.

[33] M. De Domenico, V. Nicosia, A. Arenas, and V. Latora, Nature Comms. **6**, 6864 (2015).

[34] V. B. Chen, I. W. Davis, and D. C. Richardson, Protein Sci. **18**, 2403 (2009).

[35] D. B. Larremore, A. Clauset, and C. O. Buckee, PLoS Comput. Biol. **9**, e1003268 (2013).

[36] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, J. Stat. Mech. (2008) P10008.

[37] S. Fortunato and D. Hric, Phys. Rep. **659**, 1 (2016).

[38] M. E. J. Newman, Phys. Rev. E **74**, 036104 (2006).

[39] M. Girvan and M. E. Newman, Proc. Natl. Acad. Sci. USA **99**, 7821 (2002).

[40] A. Clauset, C. Moore, and M. E. Newman, Nature (London) **453**, 98 (2008).

[41] T. M. Roddenberry and S. Segarra, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Barcelona, Spain, 2020).

[42] T. M. Roddenberry and S. Segarra, in *Proceedings of the 2020 54th Asilomar Conference on Signals, Systems, and Computers* (IEEE, Piscataway, NJ, 2020), pp. 465–469.

[43] N. Ruggeri and C. De Bacco, Appl. Netw Sci. **5**, 81 (2020).

[44] N. Ruggeri and C. De Bacco, in *Proceedings of the International Conference on Complex Networks and Their Applications* (Springer, Cham, 2019).

[45] J. Devlin, M. Chang, K. Lee, and K. Toutanova, arXiv:1810.04805.

[46] J. Pennington, R. Socher, and C. D. Manning, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014).

[47] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, arXiv:1802.05365.

[48] R. Xin, arXiv:1411.2738.

[49] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, in *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS'13)* , Vol. 2 (Curran Associates Inc., Red Hook, New York, 2013), pp. 3111–3119.