Cross-validation model assessment for modular networks

Tatsuro Kawamoto[1] and Yoshiyuki Kabashima[1]

[1] *Department of Mathematical and Computing Science, Tokyo Institute of Technology, 4259-G5-22, Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa 226-8502, Japan*

(Dated: May 26, 2016)

Model assessment of the stochastic block model is a crucial step in identification of modular structures in networks. Although this has typically been done according to the principle that a parsimonious model with a large marginal likelihood or a short description length should be selected, another principle is that a model with a small prediction error should be selected. We show that the leave-one-out cross-validation estimate of the prediction error can be efficiently obtained using belief propagation for sparse networks. Furthermore, the relations among the objectives for model assessment enable us to determine the exact cause of overfitting.

Introduction — Mathematical tools for graph or network analysis have wide applicability in various disciplines of science. In fact, many datasets, e.g., biological, information, and social data, that represent interactions or relations among elements have been successfully studied as networks in machine learning, computer science, and statistical physics. In a broad sense, a major goal is to identify macroscopic structures, including temporal structures, hidden in the data. To this end, for example, degree sequences, community and core–periphery structures, and various centralities have been extensively studied. In this Letter, we focus on identifying modular structures including community structures, bipartite structure, and a type of core–periphery structure through graph clustering. Bayesian approaches using the stochastic block model [1] are a powerful tool for this task. Graph clustering consists of two steps: selecting the number of clusters and determining the cluster assignment of each vertex. These steps may be performed repeatedly. Some methods require the number of clusters as an input, whereas others determine it automatically. In a Bayesian framework, the former step is called model selection, and this is our major focus. Model selection and its assessment for modular networks have been discussed in several ways. A classical prescription is to optimize an objective function, e.g., modularity [2, 3] and the map equation [4], or to utilize the spectral method and count the number of eigenvalues outside of the spectral band [5, 6]. In the Bayesian framework, one principle is to select a model that maximizes the model's posterior probability [7–10] or the one with the minimum description length [11]. Finally, minimization of the prediction error is a well-accepted principle, and cross-validation estimates it adequately [12, 13]. Unfortunately, a straightforward implementation of cross-validation is computationally expensive when the number of samples for validation is large; for instance, for leave-one-out cross-validation (LOOCV), to assess $q \in \{2, \dots, q_{\max}\}$, one has to run the learning algorithm a total of $N(q_{\max}-1)$ times for very similar training sets. Nevertheless, we show that the LOOCV is an exception and can be applied without the need to perform learning N times by exploiting the fact that the cavity biases in belief propagation (BP) are exactly the ingredients of the LOOCV. Throughout this Letter, we consider undirected sparse networks, and we ignore multi-edges and self-loops for simplicity. We denote by E the set of edges in the network. We denote by N and L the total numbers of vertices and edges, respectively. All the detailed derivations of the results can be found in the Supplemental Material.

Stochastic block model — The hyperparameters that specify the standard stochastic block model are the number of clusters q, the fraction of the cluster size γ_σ, and the so-called affinity matrix $\omega_{\sigma\sigma'}$, which indicates the connection probabilities within and between clusters, where σ is the cluster label. Because the networks we consider are sparse, $\boldsymbol{\omega} = O(N^{-1})$. Assuming that the edges are generated independently and randomly on the basis of the affinity matrix, the probability that an adjacency matrix A is generated, i.e., the likelihood of the model, is expressed as

$$p(A, \boldsymbol{\sigma}|\boldsymbol{\gamma}, \boldsymbol{\omega}, q) = \prod_{i=1}^{N} \gamma_{\sigma_i} \prod_{i<j} \omega_{\sigma_i\sigma_j}^{A_{ij}} \left(1 - \omega_{\sigma_i\sigma_j}\right)^{1-A_{ij}}. \quad (1)$$

The cluster assignment of the vertices $\boldsymbol{\sigma}$ is the hidden variable, and one usually conducts hyperparameter learning with respect to $(\boldsymbol{\gamma}, \boldsymbol{\omega})$ and cluster inference using the expectation–maximization (EM) algorithm so that the marginal log-likelihood, or the negative free energy, $\log \sum_{\boldsymbol{\sigma}} p(A, \boldsymbol{\sigma}|\boldsymbol{\gamma}, \boldsymbol{\omega}, q)$, is maximized.

BP and the Bethe free energy — The EM algorithm for a block model inference requires computation of the marginal probability of cluster assignment ψ_σ^i for each vertex. Under the tree approximation, BP offers its estimate by calculating the cavity bias $\psi_\sigma^{i\to j}$, which is the marginal probability of vertex i without the marginalization of vertex j. (See Ref. [9] or the Supplemental Material for details.) Using the cavity biases, the negative marginal log-likelihood per vertex is estimated as

$$f_{\text{Bethe}} = -\frac{1}{N}\sum_i \log Z^i + \frac{1}{N}\sum_{(i,j)\in E} \log Z^{ij} - \frac{c}{2}, \quad (2)$$

where c is the average degree. Each term in the summations is

$$Z^i = \sum_{\sigma_i} \gamma_{\sigma_i} \mathrm{e}^{-h_{\sigma_i}} \prod_{k\in\partial i} \left(\sum_{\sigma_k} \psi^{k\to i}_{\sigma_k} \omega_{\sigma_k\sigma_i} \right), \tag{3}$$

$$Z^{ij} = \sum_{\sigma\sigma'} \omega_{\sigma\sigma'} \psi^{i\to j}_{\sigma} \psi^{j\to i}_{\sigma'} \qquad \text{for } (i,j)\in E, \tag{4}$$

where ∂i indicates the set of neighboring vertices of i, and $h_\sigma = \sum_{k=1}^{N} \sum_{\sigma_k} \psi^k_{\sigma_k} \omega_{\sigma_k\sigma}$. Note that undirected networks have the symmetry $\omega_{\sigma\sigma'} = \omega_{\sigma'\sigma}$. Although the result of BP is generally an approximation, it is exact when the network is a tree and is quite accurate when the network is sparse. The function f_{Bethe} is called the Bethe free energy, and the parsimonious model that minimizes it is expected to give the correct number of clusters q of the generative model, which corresponds to the maximum likelihood estimation of the hyperparameters. In [9], $-c\log N/2$ is added to (2) to make the Bethe free energy extensive [14]; in numerical experiments, we follow their convention.

Bayes prediction error — The need to evaluate the predictability of the learned model is another principle of model assessment. In the context of a network, we quantify how well our model predicts the existence or nonexistence of edges for "new data." In reality, however, we do not have the "new data" in a given dataset. Therefore, we consider the cross-validation estimate as a proxy of the prediction error; we select a model with a small value of the error. We denote by $A^{\setminus(i,j)}$ the adjacency matrix of a network in which A_{ij} is missing, i.e., in which it is unknown whether $A_{ij} = 0$ or $A_{ij} = 1$. The posterior predictive distribution $\hat{p}(A_{ij} = 1|A^{\setminus(i,j)})$ of the model in which vertices i and j are connected given dataset $A^{\setminus(i,j)}$, or the marginal likelihood of the learned model for the vertex pair, is

$$\hat{p}(A_{ij} = 1|A^{\setminus(i,j)}) = \sum_{\sigma_i,\sigma_j} \hat{p}(A_{ij} = 1|\sigma_i,\sigma_j) p(\sigma_i,\sigma_j|A^{\setminus(i,j)}). \tag{5}$$

The error should be small when the prediction of edge existence for every vertex pair is accurate. Thus, the probability distribution (5) is close, in the sense of the Kullback–Leibler (KL) divergence, to the actual distribution, which is given as the empirical distribution. Therefore, it is natural to employ, as a measure of the prediction error, the cross-entropy error function [15]

$$\begin{aligned} E_{\mathrm{Bayes}}(q) &= -\overline{\log p(A_{ij}|A^{\setminus(i,j)})} \\ &= -\frac{1}{L}\sum_{i<j}\Big[A_{ij}\log\hat{p}(A_{ij}=1|A^{\setminus(i,j)}) \\ &\quad + (1-A_{ij})\log\hat{p}(A_{ij}=0|A^{\setminus(i,j)})\Big], \end{aligned} \tag{6}$$

where $\overline{X} \equiv \sum_{i<j}\sum_{A_{ij}=\{0,1\}} P(A_{ij})X(A_{ij})/L$. Note that we chose the normalization so that E_{Bayes} is typically $O(1)$ in sparse networks. We refer to (6) as the Bayes prediction error, which corresponds to the LOOCV estimate of the *stochastic complexity* [16]. As long as the model we use is consistent with the data, the posterior predictive distribution is optimal as an element of the prediction error, because the intermediate dependence (σ_i,σ_j) is fully marginalized and gives the smallest error. In (5), $\hat{p}(A_{ij} = 1|\sigma_i,\sigma_j) = \omega_{\sigma_i\sigma_j}$ by model definition. An important observation is that, because the cavity bias $\psi^{i\to j}_{\sigma_i}$ represents the marginal probability without the information for vertex j, this is exactly what we need for prediction in the LOOCV, that is, $p(\sigma_i,\sigma_j|A^{\setminus(i,j)}) = \psi^{i\to j}_{\sigma_i}\psi^{j\to i}_{\sigma_j}$. On the other hand, we have $p(\sigma_i,\sigma_j|A_{ij}=1,A^{\setminus(i,j)}) = \psi^{i\to j}_{\sigma_i}\omega_{\sigma_i\sigma_j}\psi^{j\to i}_{\sigma_j}/Z^{ij}$ for the conditional probability, where Z^{ij} is defined in (4), and $p(A_{ij} = 1|A^{\setminus(i,j)}) = Z^{ij}$. Note that utilizing BP for the LOOCV itself is not totally new; this idea has been addressed in a different context in the literature, e.g., Ref. [17]. By using the fact that $L = O(N)$ and $\hat{p}(A_{ij} = 1|A^{\setminus(i,j)}) = O(N^{-1})$, $E_{\mathrm{Bayes}}(q)$ can be written as

$$E_{\mathrm{Bayes}}(q) = -\frac{1}{L}\sum_{(i,j)\in E}\log Z^{ij} + \text{const.} + O(N^{-1}). \tag{7}$$

Equation (7) indicates that the prediction with respect to the non-edges contributes only as a constant, so $E_{\mathrm{Bayes}}(q)$ essentially measures whether the existence of edges is correctly predicted in a sparse network. Using the relation $Z^{i\to j} = Z^i/Z^{ij}$, where $Z^{i\to j}$ is the normalization factor of $\psi^{i\to j}_\sigma$, we can also write (7) as

$$E_{\mathrm{Bayes}}(q) = -\frac{1}{2L}\sum_i\sum_{k\in\partial i}\left(\log Z^i - \log Z^{k\to i}\right), \tag{8}$$

where we ignored the constant and $O(N^{-1})$ factors.

Gibbs prediction error — Although the prediction error using the posterior predictive distribution is the best choice when the model we use is consistent with the data, this assumption is often invalid in practice. In that case, the Bayes prediction error may no longer be optimal for prediction. In (6), we employed $-\log p(A_{ij}|A^{\setminus(i,j)})$ as the error of a vertex pair. Instead, we can consider the log-likelihood of cluster assignment $E(A_{ij}|\sigma_i,\sigma_j) = -\log\hat{p}(A_{ij}|\sigma_i,\sigma_j)$ as a fundamental element and measure $\langle E\rangle$ as the prediction error of a vertex pair, where $\langle\cdots\rangle$ is the average over $p(\sigma_i,\sigma_j|A^{\setminus(i,j)})$; that is, the cluster assignment (σ_i,σ_j) is drawn from the posterior distribution, and the error of the vertex pair is measured with respect to this fixed assignment. Then, the corresponding cross-entropy error function is

$$\begin{aligned} E_{\mathrm{Gibbs}}(q) &= \overline{\langle E(A_{ij}|\sigma_i,\sigma_j)\rangle} \\ &= -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}\psi^{i\to j}_{\sigma_i}\psi^{j\to i}_{\sigma_j}\log\omega_{\sigma_i\sigma_j}. \end{aligned} \tag{9}$$

Again, we omitted the constant and $O(N^{-1})$ factors. We refer to (9) as the Gibbs prediction error. If the probability distribution of cluster assignment is highly peaked, $E_{\rm Gibbs}$ will be close to $E_{\rm Bayes}$, and $E_{\rm Gibbs}$ and $E_{\rm Bayes}$ will be very small if those assignments predict the actual network well. Alternatively, the *maximum a posteriori* (MAP) estimate of (9) is often measured for the Gibbs prediction error; hence, we refer to $E_{\rm MAP}(q)$ as the prediction error with $\psi_{\sigma_i}^{i\to j}$ replaced by $\delta_{\sigma_i,{\rm argmax}\{\psi_\sigma^{i\to j}\}}$ in (9).

Gibbs training error — Finally, we define the BP estimate of the training error. That is, not only do we use $A^{\setminus(i,j)}$, but we also use A_{ij} for cluster inference. This can be done by taking the average over $p(\sigma_i,\sigma_j|A)$ instead in (9). Thus, omitting the constant and $O(N^{-1})$ factors, we have

$$E_{\rm training}(q) = -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}\frac{\psi_{\sigma_i}^{i\to j}\omega_{\sigma_i\sigma_j}\psi_{\sigma_j}^{j\to i}}{Z^{ij}}\log\omega_{\sigma_i\sigma_j}, \tag{10}$$

which measures the goodness of fit of the assumed model to the given data. We refer to $E_{\rm training}$ as the Gibbs training error. Interestingly, as shown in the Supplemental Material, this quantity appears in the free energy (not the Bethe free energy) as a part of the internal energy.

Relations among the errors — By exploiting Bayes' rule, we have

$$\log\hat{p}(A_{ij}|A^{\setminus(i,j)}) = \log\hat{p}(A_{ij}|\sigma_i,\sigma_j,A^{\setminus(i,j)}) + \log\frac{p(\sigma_i,\sigma_j|A^{\setminus(i,j)})}{p(\sigma_i,\sigma_j|A)}. \tag{11}$$

Note here that the left-hand side does not depend on σ_i and σ_j. If we take the average with respect to $p(\sigma_i,\sigma_j|A^{\setminus(i,j)})$ on both sides,

$$\log\hat{p}(A_{ij}|A^{\setminus(i,j)}) = \left\langle\log\hat{p}(A_{ij}|\sigma_i,\sigma_j,A^{\setminus(i,j)})\right\rangle + D_{\rm KL}\left(p(\sigma_i,\sigma_j|A^{\setminus(i,j)})\,||\,p(\sigma_i,\sigma_j|A)\right), \tag{12}$$

where $D_{\rm KL}(p||q)$ is the KL divergence. Taking the sample average of the edges, we obtain

$$E_{\rm Bayes} = E_{\rm Gibbs} - \overline{D_{\rm KL}\left(p(\sigma_i,\sigma_j|A^{\setminus(i,j)})||p(\sigma_i,\sigma_j|A)\right)}. \tag{13}$$

If we take the average over $p(\sigma_i,\sigma_j|A)$ in (12) instead,

$$E_{\rm Bayes} = E_{\rm training} + \overline{D_{\rm KL}\left(p(\sigma_i,\sigma_j|A)||p(\sigma_i,\sigma_j|A^{\setminus(i,j)})\right)}. \tag{14}$$

Because the KL divergence is non-negative, $E_{\rm training} < E_{\rm Bayes} < E_{\rm Gibbs}$. Essentially the same relations as (13) and (14) were derived in the context of neural networks in a slightly different manner [16]. We can go even further. If the cluster assignment distributions with small q can be regarded as the coarse graining of that with a larger q, the information monotonicity [18, 19] of the KL divergence ensures that $E_{\rm Gibbs}$ always estimates a smaller number of clusters q than $E_{\rm Bayes}$ and $E_{\rm training}$. (See the Supplemental Material for the detailed argument.) When the inference of BP correctly predicts the edges, $E_{\rm MAP}$ is biased so that the error becomes small. Therefore, $E_{\rm MAP}$ tends to be smaller than $E_{\rm Gibbs}$. As we will observe later, $E_{\rm Gibbs}$ typically performs better than $E_{\rm Bayes}$ in practice. Equation (13) implies that detailed information about the difference in the cluster assignment distribution is often not relevant and simply causes overfitting. As shown in Fig. 1, when the network is truly generated by the stochastic block model, the Bethe free energy and all the prediction errors saturate at the planted value of q, as they should.

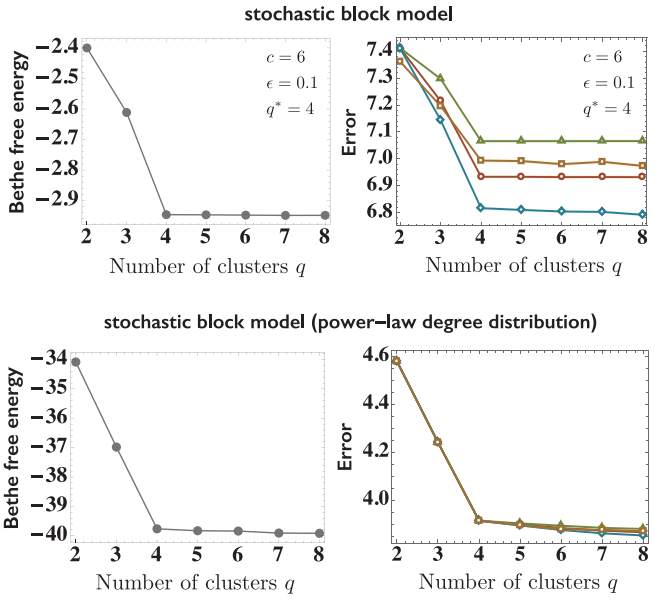

FIG. 1. (Color online) Bethe free energies (left) and prediction errors (right) of the standard stochastic block model (top) and the stochastic block model with a power-law degree distribution (bottom). Both models consist of four equal-size clusters, and $N = 10,000$ in total. For the standard stochastic block model, we set the average degree c to 6 and ϵ to 0.1, where $\epsilon = \omega_{\rm out}/\omega_{\rm in}$, and $\omega_{\sigma\sigma'} = \omega_{\rm in}$ for $\sigma = \sigma'$ and $\omega_{\rm out}$ otherwise. For the other model, which we generated as the LFR network, we set the average degree c to 9.58, mixing parameter μ to 0.01, and exponent of the degree distribution τ to -2, with a maximum degree $d_{\rm max}$ of 100. The four data in the right panel are the Bayes prediction errors $E_{\rm Bayes}$ (red circles), Gibbs prediction errors $E_{\rm Gibbs}$ (green triangles), Gibbs training errors $E_{\rm training}$ (blue diamonds), and MAP estimates $E_{\rm MAP}$ of $E_{\rm Gibbs}$ (yellow squares).

Degree-corrected stochastic block model — In practice, it is observed that the standard stochastic block model

is often not flexible enough to fit real-world data with heterogeneous degree distributions. In such a case, the assessment of the number of clusters q may not make any sense. Therefore, in addition, we conduct the analysis for the degree-corrected stochastic block model [20] in parallel. In the degree-corrected stochastic block model of a simple graph, the probability $p(A_{ij} = 1|\sigma_i, \sigma_j)$ that a pair of vertices is connected given their cluster assignments σ_i and σ_j is $\theta_i \omega_{\sigma_i,\sigma_j} \theta_j$ instead of $\omega_{\sigma_i,\sigma_j}$. With this replacement, we can obtain the corresponding Bethe free energy and error functions analogously. Figure 1 shows an equal-size stochastic block model with a power-law degree distribution, which we generated as an instance of the LFR network [21]. The mixing parameter μ is set to 0.1. The planted number of clusters is correctly estimated using both the Bethe free energy and the prediction and training errors in this case.

Real-world networks — Finally, and most importantly, model assessment using various error functions is applied to real-world networks. Unlike the case for synthetic networks, the selection of q is not very obvious for many networks because the error functions do not saturate clearly as q increases. This makes sense, because real-world networks may not have a clear simple modular structure; thus, they may not be perfectly fitted by either the standard or degree-corrected stochastic block models. In other words, by plotting the error functions, we can see how confident we can be about our model selection. Recall that each cross-validation estimate is given as an average error per edge, so we can also measure its standard errors. To select the parsimonious model, the "one-standard error" rule [22] is often used, in which the most parsimonious model whose error is no more than one standard error above the error of the best model is selected. To apply this empirical rule, we plotted the standard errors of the cross-validation estimates as shadows. For example, although the best model of the network of books about US politics (which we refer to as *political books*) is $q = 7$ for the standard stochastic block model, we choose $q = 5$ as the most parsimonious model. Although the estimated q from the Bethe free energy and prediction errors coincide in some cases, as far as we examined, the Bethe free energy does not saturate at a reasonable value of q, as already pointed out in [23]. The Bayes prediction error E_{Bayes} and Gibbs training error E_{training} perform similarly. In contrast, the Gibbs prediction error E_{Gibbs} shows good performance in the sense that its suggestion often coincides with the number of "ground-truth" communities of well-studied networks. Note that, even when we do not measure the Bethe free energy for model assessment, we still minimize the Bethe free energy in the cluster inference step.

Summary and Discussion — We derived cross-validation estimates for various types of errors in terms of the distribution obtained by BP. This approach is incomparably more efficient than a straightforward application of LOOCV and offers a reasonable model assessment. Moreover, we also showed the relations between the objectives for model assessment. This is quite important, because we can determine the exact cause of overfitting. The codes we used can be found at [25]. Although the generation of edges is highly correlated, the validity of the cross-validation is justified because we fit the data based on a stochastic block model, which assumes that every edge is generated independently and randomly. In addition, although one may expect that the LOOCV estimates the conditional prediction error because it uses very similar training sets, it reportedly estimates rather the expected prediction error [22]. Fitting with the stochastic block models is flexible, so the algorithm can infer not only the assortative structure, but also more complex structures. However, this is not always an advantage in practice. The flexibility also means that slightly different models may fit the data as well as the best model. Therefore, as a trade-off, model selection becomes more difficult. By restricting the structure that we can detect, it is possible to find a good balance of the cluster inference and model selection performance. We will address this problem in a future publication.

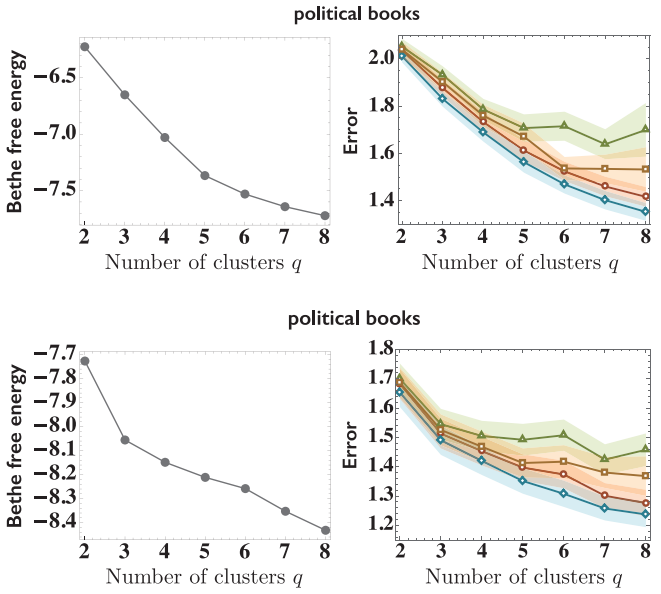

FIG. 2. (Color online) Bethe free energies (left) and prediction errors (right) of the network of books about US politics [24] as functions of the number of clusters q. They are plotted in the same manner as in Fig. 1. The standard errors are shown as shadows.

This work was supported by JSPS KAKENHI No. 26011023 (TK) and No. 25120013 (YK).

[1] P. W. Holland, K. B. Laskey, and S. Leinhardt, Soc. Networks **5**, 109 (1983).

[2] M. E. J. Newman and M. Girvan, Phys. Rev. E **69**, 026113 (2004).

[3] P. Zhang and C. Moore, Proc. Natl. Acad. Sci. U.S.A. **111**, 18144 (2014).

[4] M. Rosvall and C. Bergstrom, Proc. Natl. Acad. Sci. U.S.A. **105**, 1118 (2008).

[5] U. Luxburg, Statistics and Computing **17**, 395 (2007).

[6] F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang, Proc. Natl. Acad. Sci. U.S.A. **110**, 20935 (2013).

[7] K. Nowicki and T. A. B. Snijders, Journal of the American Statistical Association **96**, 1077 (2001).

[8] J. J. Daudin, F. Picard, and S. Robin, Statistics and Computing **18**, 173 (2008).

[9] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. E **84**, 066106 (2011).

[10] K. Hayashi, T. Konishi, and T. Kawamoto, arXiv preprint arXiv:1602.02256 (2016).

[11] T. P. Peixoto, Phys. Rev. X **5**, 011033 (2015).

[12] P. Hoff, in *Advances in Neural Information Processing Systems 20*, edited by J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis (Curran Associates, Inc., 2008), pp. 657–664.

[13] K. Chen and J. Lei, arXiv preprint arXiv:1411.1715 (2014).

[14] Note that Z^i and Z^{ij} are defined by the rescaled affinity matrix $c_{\sigma\sigma'} = N\omega_{\sigma\sigma'}$ in [9].

[15] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006), ISBN 0387310738.

[16] E. Levin, N. Tishby, and S. A. Solla, in *Proceedings of the Second Annual Workshop on Computational Learning Theory* (1989), COLT '89, pp. 245–260.

[17] M. Opper and O. Winther, Phys. Rev. Lett. **76**, 1964 (1996).

[18] I. Csiszár, Entropy **10**, 261 (2008).

[19] S.-i. Amari and A. Cichocki, Bulletin of the Polish Academy of Sciences: Technical Sciences **58**, 183 (2010).

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

[21] A. Lancichinetti and S. Fortunato, Phys. Rev. E **80**, 056117 (2009).

[22] T. J. Hastie, R. J. Tibshirani, and J. H. Friedman, *The elements of statistical learning : data mining, inference, and prediction*, Springer series in statistics (Springer, New York, 2009).

[23] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, Phys. Rev. Lett. **107**, 065701 (2011).

[24] M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **103**, 8577 (2006).

[25] `https://github.com/tatsuro-kawamoto/graphBIX`.

[26] X. Yan, C. Shalizi, J. E. Jensen, F. Krzakala, C. Moore, L. Zdeborová, P. Zhang, and Y. Zhu, Journal of Statistical Mechanics: Theory and Experiment **2014**, P05007 (2014).

[27] W. W. Zachary, Journal of Anthropological Research **33**, 452 (1977).

[28] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behavioral Ecology and Sociobiology **54**, 396 (2003).

[29] D. E. Knuth, *The Stanford GraphBase: a platform for combinatorial computing*, vol. 37 (Addison-Wesley Reading, 1993).

[30] M. Girvan and M. E. J. Newman, Proc. Natl. Acad. Sci. U.S.A. **99**, 7821 (2002).

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

[32] J. Duch and A. Arenas, Phys. Rev. E **72**, 027104 (2005).

[33] L. A. Adamic and N. Glance, in *Proceedings of the 3rd International Workshop on Link Discovery* (ACM, New York, NY, USA, 2005), LinkKDD '05, pp. 36–43.

Supplemental Material: Cross-validation model assessment for modular networks

BP INFERENCE OF CLUSTER ASSIGNMENT AND HYPERPARAMETER LEARNING FOR THE STOCHASTIC BLOCK MODEL

For self-containedness, we briefly summarize the EM algorithm with BP of a stochastic block model introduced in [9, 23]. The goal of cluster assignment inference is to evaluate the marginal probability of the cluster assignment ψ^i_σ for each vertex, provided that the hyperparameter set, $\boldsymbol{\gamma}$ and ω, is fixed at the estimated values. To this end, we iteratively compute the BP equation based on the likelihood $p(A, \boldsymbol{\sigma}|\boldsymbol{\gamma}, \omega, q)$; that is, for an edge $(i, j) \in E$,

$$\psi^{i\to j}_{\sigma_i} = \frac{1}{Z^{i\to j}} \gamma_{\sigma_i} \mathrm{e}^{-h_{\sigma_i}} \prod_{k\in\partial i\backslash j} \left(\sum_{\sigma_k} \psi^{k\to i}_{\sigma_k} \omega_{\sigma_k\sigma_i} \right), \tag{S1}$$

where $\partial i\backslash j$ indicates the set of neighboring vertices of i in the network except for j, and $\psi^{i\to j}_{\sigma_i}$ and $Z^{i\to j}$ are the marginal probability of vertex i without the marginalization from vertex j and its normalization factor as defined in the main text. The external field, $h_\sigma = \sum_{k=1}^N \sum_{\sigma_k} \psi^k_{\sigma_k} \omega_{\sigma_k\sigma}$, is due to the effect of non-edges $(i, k) \notin E$, where the full marginal $\psi^i_{\sigma_i}$ is

$$\psi^i_{\sigma_i} = \frac{1}{Z^i} \gamma_{\sigma_i} \mathrm{e}^{-h_{\sigma_i}} \prod_{k\in\partial i} \left(\sum_{\sigma_k} \psi^{k\to i}_{\sigma_k} \omega_{\sigma_k\sigma_i} \right). \tag{S2}$$

As defined in the main text, Z^i is the normalization factor of the full marginal. With these marginals in hand, we can update the estimate of the hyperparameters to $\hat{\boldsymbol{\gamma}}$ and $\hat{\omega}$ as

$$\hat{\gamma}_\sigma = \frac{1}{N} \sum_{i=1}^N \psi^i_\sigma, \tag{S3}$$

$$\hat{\omega}_{\sigma\sigma'} = \frac{1}{N^2 \gamma_\sigma \gamma_{\sigma'}} \sum_{(i,j)\in E} \frac{\omega_{\sigma\sigma'} (\psi^{i\to j}_\sigma \psi^{j\to i}_{\sigma'} + \psi^{j\to i}_\sigma \psi^{i\to j}_{\sigma'})}{Z^{ij}}. \tag{S4}$$

BP INFERENCE OF CLUSTER ASSIGNMENT AND HYPERPARAMETER LEARNING FOR THE DEGREE-CORRECTED STOCHASTIC BLOCK MODEL

Because the degree distribution of the standard stochastic block model is always the Poisson distribution, it is sometimes not flexible enough to fit the data. To overcome this issue, the degree-corrected stochastic block model was proposed [20]. Although the EM algorithm with BP updates was discussed in [26], we write it in a form similar to those for the standard stochastic block model in the previous section. The likelihood of the degree-corrected stochastic block model is

$$p(A, \boldsymbol{\sigma} \mid \boldsymbol{\gamma}, \boldsymbol{\omega}, \boldsymbol{\theta}) = \prod_i \gamma_{\sigma_i} \prod_{i<j} \frac{(\theta_i \omega_{\sigma_i\sigma_j} \theta_j)^{A_{ij}}}{A_{ij}!} \mathrm{e}^{-\theta_i \omega_{\sigma_i\sigma_j} \theta_j}, \tag{S5}$$

where θ_i is an arbitrary hyperparameter for degree correction. By grouping vertices into clusters, the likelihood can be rewritten as

$$\left(\prod_{i<j} \frac{1}{A_{ij}!} \right) \prod_i \gamma_{\sigma_i} \prod_i \theta_i^{d_i} \prod_{\sigma\sigma'} \omega_{\sigma\sigma'}^{\frac{1}{2} m_{\sigma\sigma'}} \mathrm{e}^{-\frac{1}{2}\omega_{\sigma\sigma'} \left(\sum_i \delta_{\sigma\sigma_i} \theta_i\right) \left(\sum_j \delta_{\sigma'\sigma_j} \theta_j\right)}, \tag{S6}$$

where $m_{\sigma\sigma'} = \sum_{ij} A_{ij} \delta_{\sigma\sigma_i} \delta_{\sigma'\sigma_j}$ is the number of edges between clusters σ and σ' if $\sigma \neq \sigma'$, and it is doubly counted if $\sigma = \sigma'$. By assuming that A_{ij} is either zero or one for any vertex pair, we neglect the first product. As a normalization constraint of $\boldsymbol{\theta}$, $\sum_i \delta_{\sigma\sigma_i} \theta_i = n_\sigma$ is usually imposed, where n_σ is the number of vertices within cluster σ. Then, the log-likelihood reads

$$\log p(A, \boldsymbol{\sigma} \mid \boldsymbol{\gamma}, \boldsymbol{\omega}, \boldsymbol{\theta}) = \sum_i (\log \gamma_{\sigma_i} + d_i \log \theta_i) + \frac{1}{2} \sum_{\sigma\sigma'} (m_{\sigma\sigma'} \log \omega_{\sigma\sigma'} - \omega_{\sigma\sigma'} n_\sigma n_{\sigma'}) + \text{const.} \tag{S7}$$

The BP equation corresponding to (S5) is

$$\psi^{i\to j}_{\sigma_i} = \frac{1}{Z^{i\to j}}\gamma_{\sigma_i}\prod_{k\backslash j}\left[\sum_{\sigma_k}\psi^{k\to i}_{\sigma_k}(\theta_i\omega_{\sigma_i\sigma_j}\theta_j)^{A_{ij}}\,\mathrm{e}^{-\theta_i\omega_{\sigma_i\sigma_j}\theta_j}\right]. \tag{S8}$$

The expansion that ignores the $O(N^{-1})$ factors yields, analogously to (S2),

$$\psi^{i}_{\sigma_i} = \frac{1}{Z^{i}}\gamma_{\sigma_i}\mathrm{e}^{-\theta_i h_{\sigma_i}}\prod_{k\in\partial i}\left[\sum_{\sigma_k}\psi^{k\to i}_{\sigma_k}\theta_k\omega_{\sigma_k\sigma_i}\theta_i\right], \tag{S9}$$

$$h_\sigma = \sum_k \theta_k \sum_{\sigma_k}\psi^{k}_{\sigma_k}\omega_{\sigma_k\sigma}, \tag{S10}$$

and the BP equation (S8) for $(i,j)\in E$ is approximated as

$$\psi^{i\to j}_{\sigma_i} = \frac{1}{Z^{i\to j}}\gamma_{\sigma_i}\mathrm{e}^{-\theta_i h_{\sigma_i}}\prod_{k\in\partial i\backslash j}\left[\sum_{\sigma_k}\psi^{k\to i}_{\sigma_k}\theta_k\omega_{\sigma_k\sigma_i}\theta_i\right]. \tag{S11}$$

By setting $\theta_i = 1$ for every vertex, (S1) is recovered. According to the saddle-point conditions of (S7), the hyperparameters γ_σ and $\omega_{\sigma\sigma'}$ should be updated as (S3) and (S4). For θ_i of vertex i with degree d_i,

$$\hat{\theta}_i = \frac{d_i}{d_\sigma}, \qquad d_\sigma = \frac{1}{n_\sigma}\sum_i \delta_{\sigma\sigma_i}d_i. \tag{S12}$$

The cluster assignment σ_i is determined as that with the maximum value in ψ^i_σ. The Bethe free energy can also be written analogously to that in the standard stochastic block model.

$$f_{\mathrm{Bethe}} = -\frac{1}{N}\sum_i \log Z^i + \frac{1}{N}\sum_{(i,j)\in E}\log Z^{ij} + \frac{1}{N}\sum_{(i,j)\notin E}\log\tilde{Z}^{ij}. \tag{S13}$$

The first and second terms are

$$Z^i = \sum_\sigma \gamma_{\sigma_i}\mathrm{e}^{-\theta_i h_{\sigma_i}}\prod_{k\in\partial i}\left[\sum_{\sigma_k}\psi^{k\to i}_{\sigma_k}\theta_k\omega_{\sigma_k\sigma_i}\theta_i\right], \tag{S14}$$

$$Z^{ij} = \sum_{\sigma\sigma'}\psi^{i\to j}_\sigma\theta_i\omega_{\sigma\sigma'}\theta_j\psi^{j\to i}_{\sigma'}. \tag{S15}$$

Note again that we have the symmetry $\omega_{\sigma\sigma'} = \omega_{\sigma'\sigma}$. The non-edge part of (S13) is

$$\begin{aligned}\sum_{(i,j)\notin E}\log\tilde{Z}^{ij} &= \sum_{(i,j)\notin E}\log\sum_{\sigma\sigma'}\psi^{i\to j}_\sigma\left(1-\theta_i\omega_{\sigma\sigma'}\theta_j\right)\psi^{j\to i}_{\sigma'} \\ &\approx -\frac{1}{2}\sum_{\sigma\sigma'}\left(\sum_i\psi^i_\sigma\theta_i\right)\omega_{\sigma\sigma'}\left(\sum_j\theta_j\psi^j_{\sigma'}\right) \\ &= L.\end{aligned} \tag{S16}$$

DERIVATION OF THE ERROR FUNCTIONS

In this section, we explain the derivation of the Bayes prediction error $E_{\mathrm{Bayes}}(q)$, Gibbs prediction error $E_{\mathrm{Gibbs}}(q)$, and Gibbs training error $E_{\mathrm{training}}(q)$. Because $L = O(N)$ and $\hat{p}(A_{ij}=1|A^{\backslash(i,j)}) = O(N^{-1})$, the Bayes prediction

error $E_{\rm Bayes}(q)$ is

$$\begin{aligned}
E_{\rm Bayes}(q) &= -\frac{1}{L}\sum_{i<j}\left[A_{ij}\log\hat{p}(A_{ij}=1|A^{\setminus(i,j)}) + (1-A_{ij})\log\left(1-\hat{p}(A_{ij}=1|A^{\setminus(i,j)})\right)\right] \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\log\hat{p}(A_{ij}=1|A^{\setminus(i,j)}) + \frac{1}{L}\sum_{i<j}\hat{p}(A_{ij}=1|A^{\setminus(i,j)}) + O(N^{-1}) \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\log\sum_{\sigma_i,\sigma_j}\psi^{i\to j}_{\sigma_i}\omega_{\sigma_i\sigma_j}\psi^{j\to i}_{\sigma_j} + \text{const.} + O(N^{-1}) \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\log Z^{ij} + \text{const.} + O(N^{-1}),
\end{aligned} \tag{S17}$$

where we used

$$\frac{1}{L}\sum_{i<j}\hat{p}(A_{ij}=1|A^{\setminus(i,j)}) = \frac{1}{L}\sum_{i<j}Z^{ij} = \text{const.} \tag{S18}$$

Precisely speaking, the quantity above is a function of the hyperparameters, because $\sum_{i<j}Z^{ij}$ is the expectation of the total number of edges. However, we can assume that its dependence is negligible because the total number of edges is a macroscopic quantity. Analogously to $E_{\rm Bayes}(q)$, the Gibbs prediction error $E_{\rm Gibbs}(q)$ is

$$\begin{aligned}
E_{\rm Gibbs}(q) &= -\frac{1}{L}\sum_{i<j}\Bigg[A_{ij}\sum_{\sigma_i,\sigma_j}p(\sigma_i,\sigma_j|A^{\setminus(i,j)})\log\hat{p}(A_{ij}=1|\sigma_i,\sigma_j) \\
&\qquad + (1-A_{ij})\sum_{\sigma_i,\sigma_j}p(\sigma_i,\sigma_j|A^{\setminus(i,j)})\log\hat{p}(A_{ij}=0|\sigma_i,\sigma_j)\Bigg] \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}p(\sigma_i,\sigma_j|A^{\setminus(i,j)})\log\hat{p}(A_{ij}=1|\sigma_i,\sigma_j) + \frac{1}{L}\sum_{i<j}Z^{ij} + O(N^{-1}) \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}\psi^{i\to j}_{\sigma_i}\psi^{j\to i}_{\sigma_j}\log\omega_{\sigma_i\sigma_j} + \text{const.} + O(N^{-1}).
\end{aligned} \tag{S19}$$

Finally, the training error $E_{\rm training}(q)$ is

$$\begin{aligned}
E_{\rm training} &= -\frac{1}{L}\sum_{i<j}\Bigg[A_{ij}\sum_{\sigma_i,\sigma_j}p(\sigma_i,\sigma_j|A_{ij}=1,A^{\setminus(i,j)})\log\hat{p}(A_{ij}=1|\sigma_i,\sigma_j) \\
&\qquad + (1-A_{ij})\sum_{\sigma_i,\sigma_j}p(\sigma_i,\sigma_j|A_{ij}=0,A^{\setminus(i,j)})\log\hat{p}(A_{ij}=0|\sigma_i,\sigma_j)\Bigg] \\
&= -\frac{1}{L}\sum_{i<j}\Bigg[A_{ij}\sum_{\sigma_i,\sigma_j}\frac{\psi^{i\to j}_{\sigma_i}\omega_{\sigma_i\sigma_j}\psi^{j\to i}_{\sigma_j}}{Z^{ij}}\log\omega_{\sigma_i\sigma_j} \\
&\qquad + (1-A_{ij})\sum_{\sigma_i,\sigma_j}\frac{\psi^{i\to j}_{\sigma_i}\left(1-\omega_{\sigma_i\sigma_j}\right)\psi^{j\to i}_{\sigma_j}}{1-Z^{ij}}\log(1-\omega_{\sigma_i\sigma_j})\Bigg] \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}\frac{\psi^{i\to j}_{\sigma_i}\omega_{\sigma_i\sigma_j}\psi^{j\to i}_{\sigma_j}}{Z^{ij}}\log\omega_{\sigma_i\sigma_j} + \frac{1}{L}\sum_{i<j}\frac{Z^{ij}}{1-Z^{ij}} + O(N^{-1}) \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}\frac{\psi^{i\to j}_{\sigma_i}\omega_{\sigma_i\sigma_j}\psi^{j\to i}_{\sigma_j}}{Z^{ij}}\log\omega_{\sigma_i\sigma_j} + \frac{1}{L}\sum_{i<j}Z^{ij} + O(N^{-1}) \\
&= -\frac{1}{L}\sum_{(i,j)\in E}\sum_{\sigma_i,\sigma_j}\frac{\psi^{i\to j}_{\sigma_i}\omega_{\sigma_i\sigma_j}\psi^{j\to i}_{\sigma_j}}{Z^{ij}}\log\omega_{\sigma_i\sigma_j} + \text{const.} + O(N^{-1}).
\end{aligned} \tag{S20}$$

This training error can be interpreted as a part of the internal energy, as follows. Let θ be the set of hyperparameters. Because

$$\sum_{\boldsymbol{\sigma}'} p(A, \boldsymbol{\sigma}'|\theta) = p(A|\theta) = \frac{p(A, \boldsymbol{\sigma}|\theta)}{p(\boldsymbol{\sigma}|A, \theta)} \tag{S21}$$

holds for an arbitrary $\boldsymbol{\sigma}$, we can decompose the free energy (not the Bethe free energy) $-\log \sum_{\boldsymbol{\sigma}'} p(A, \boldsymbol{\sigma}'|\boldsymbol{\gamma}, \omega, q)/N$ as

$$f = -\frac{1}{N} \log \frac{p(A, \boldsymbol{\sigma}|\theta)}{p(\boldsymbol{\sigma}|A, \theta)}. \tag{S22}$$

Taking the average over a probability distribution $q(\boldsymbol{\sigma})$ on both sides, we obtain the following variational expression.

$$\begin{aligned} f &= -\frac{1}{N} \sum_{\boldsymbol{\sigma}} q(\boldsymbol{\sigma}) \log \frac{p(A, \boldsymbol{\sigma}|\theta)}{q(\boldsymbol{\sigma})} \frac{q(\boldsymbol{\sigma})}{p(\boldsymbol{\sigma}|A, \theta)} \\ &= \frac{1}{N} \left[-\sum_{\boldsymbol{\sigma}} q(\boldsymbol{\sigma}) \log p(A, \boldsymbol{\sigma}|\theta) + \sum_{\boldsymbol{\sigma}} q(\boldsymbol{\sigma}) \log q(\boldsymbol{\sigma}) - D_{\mathrm{KL}} \left(q(\boldsymbol{\sigma}) || p(\boldsymbol{\sigma}|A, \theta) \right) \right]. \end{aligned} \tag{S23}$$

When $q(\boldsymbol{\sigma})$ is the correct marginal, $p(\boldsymbol{\sigma}|A, \theta)$, the KL divergence disappears. We can interpret the first and second terms as corresponding to the internal energy and negative entropy, respectively, and (S23) as the thermodynamic relation of the free energy. Substituting the specific form of $\log p(A, \boldsymbol{\sigma}|\theta)$, we have

$$\begin{aligned} u = -\frac{1}{N} \sum_{\boldsymbol{\sigma}} q(\boldsymbol{\sigma}) \log p(A, \boldsymbol{\sigma}|\theta) &= -\frac{1}{N} \sum_{\boldsymbol{\sigma}} q(\boldsymbol{\sigma}) \left[\sum_i \log \gamma_{\sigma_i} + \sum_{i<j} \left(A_{ij} \log \omega_{\sigma_i \sigma_j} + (1 - A_{ij}) \log(1 - \omega_{\sigma_i \sigma_j}) \right) \right] \\ &= \frac{1}{N} \sum_i \sum_\sigma q^i_\sigma \log \gamma_\sigma + \frac{1}{N} \sum_{(i,j) \in E} \sum_{\sigma\sigma'} q^{ij}_{\sigma\sigma'} \log \omega_{\sigma\sigma'} + \frac{c}{2} + O(N^{-1}), \end{aligned} \tag{S24}$$

where $q^i_\sigma = \langle \delta_{\sigma\sigma_i} \rangle_{\boldsymbol{\sigma}}$ and $q^{ij}_{\sigma\sigma'} = \left\langle \delta_{\sigma,\sigma_i} \delta_{\sigma'\sigma_j} \right\rangle_{\boldsymbol{\sigma}}$ are the marginalized probabilities, where $\langle \cdots \rangle_{\boldsymbol{\sigma}}$ is the average over $q(\boldsymbol{\sigma})$. The second term with $q^{ij}_{\sigma\sigma'}$ estimated by the Bethe approximation is the essential factor of the Gibbs training error (S20). Sometimes, the fractions of the size of clusters $\boldsymbol{\gamma}$ are not included in the log-likelihood. In that case, up to a constant factor, the Gibbs training error is exactly the internal energy.

INFORMATION MONOTONICITY AND THE RELATIONS AMONG THE ERRORS

Recall that the Bayes prediction error E_{Bayes}, Gibbs prediction error E_{Gibbs}, and Gibbs training error E_{training} are related via (13) and (14) in the main text. We select the number of clusters q as the point at which the error function saturates (i.e., stops decreasing) with increasing q. For a smaller q to be selected by E_{Bayes} than by E_{Gibbs}, the gap between them, $\overline{D_{\mathrm{KL}} \left(p(\sigma_i, \sigma_j | A^{\setminus (i,j)}) || p(\sigma_i, \sigma_j | A) \right)}$, must decrease (see Fig. S1). In this section, we explain the information monotonicity of the KL divergence and when it is applicable in the present context. Let us consider sets of variables $\boldsymbol{A} = \{A_1, \ldots, A_m\}$ and $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ $(n > m)$. We define the probability distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ as refinements of the probability distributions $\bar{p}(\boldsymbol{A})$ and $\bar{q}(\boldsymbol{A})$, respectively, if there exists a family of sets $\{\boldsymbol{x}^\mu\}_{\mu=1}^m$ that is a partition of $\boldsymbol{x}$, i.e., $\boldsymbol{x}^\mu \cap \boldsymbol{x}^{\mu'} = \emptyset$ for $\mu \neq \mu'$ and $\cup_\mu \boldsymbol{x}^\mu = \boldsymbol{x}$, that satisfies $p(\boldsymbol{x}^\mu) = \bar{p}(A_\mu)$ and $q(\boldsymbol{x}^\mu) = \bar{q}(A_\mu)$ for any μ. In other words, $\boldsymbol{A}$ can be regarded as the coarse graining of $\boldsymbol{x}$. An example is given in Fig. S2. Note, however, that if $\boldsymbol{A}$ is actually constructed as the coarse graining of $\boldsymbol{x}$, the above condition trivially holds for $\bar{p} = p$ and $\bar{q} = q$. In general, a family that satisfies the above condition may not exist; even if it exists, it may not be unique. The *information monotonicity* of the KL divergence states that, for $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$, which are the refinements of $\bar{p}(\boldsymbol{A})$ and $\bar{q}(\boldsymbol{A})$, respectively,

$$D_{\mathrm{KL}}(p||q) \geq D_{\mathrm{KL}}(\bar{p}||\bar{q}), \tag{S25}$$

which is natural, because the difference between distributions is more visible at finer resolution. Equation (S25) is deduced by the convexity of the KL divergence. First, we can rewrite the right-hand side of (S25) in terms of p and

FIG. S1. Schematic picture of the shapes of the error functions. As long as the gap between the Bayes prediction error (bottom) and the Gibbs prediction error (top) is nondecreasing, the former does not saturate earlier than the latter.

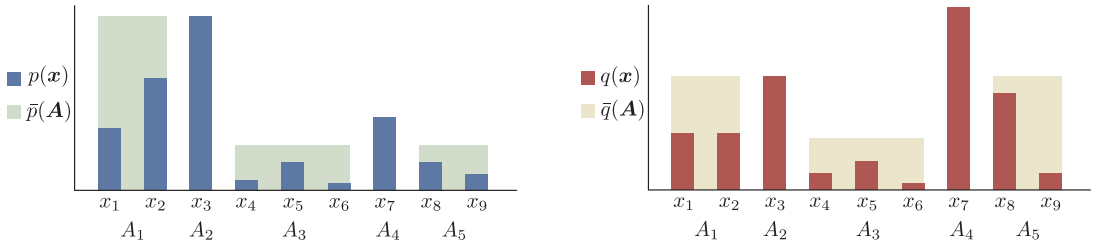

FIG. S2. Example of refinement of probability distributions. We can regard $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ as the refinements of $\bar{p}(\boldsymbol{A})$ and $\bar{q}(\boldsymbol{A})$, respectively, with $\boldsymbol{x}^1 = \{x_1, x_2\}$, $\boldsymbol{x}^2 = \{x_3\}$, $\boldsymbol{x}^3 = \{x_4, x_5, x_6\}$, $\boldsymbol{x}^4 = \{x_7\}$, and $\boldsymbol{x}^5 = \{x_8, x_9\}$ as a possible correspondence. Note that the correspondence is not unique. If we refer only to $q(\boldsymbol{x})$ and $\bar{q}(\boldsymbol{A})$, the assignment of $\{x_1, x_2\}$, $\{x_3\}$, and $\{x_8, x_9\}$ is exchangeable within A_1, A_2, and A_5. However, $\bar{p}(A_5)$ does not coincide with $p(\{x_1, x_2\})$ or $p(x_3)$; therefore, only A_1 and A_2 are exchangeable between $\{x_1, x_2\}$ and $\{x_3\}$. The same goes for $\{x_4, x_5, x_6\}$ and $\{x_8, x_9\}$.

q as

$$D_{\mathrm{KL}}(\bar{p}||\bar{q}) = \sum_{\mu=1}^{m} p(\boldsymbol{x}^\mu) \log \frac{p(\boldsymbol{x}^\mu)}{q(\boldsymbol{x}^\mu)}; \tag{S26}$$

thus, if

$$\sum_{x \in x^\mu} p(x) \log \frac{p(x)}{q(x)} \geq p(\boldsymbol{x}^\mu) \log \frac{p(\boldsymbol{x}^\mu)}{q(\boldsymbol{x}^\mu)} \tag{S27}$$

holds for an arbitrary μ, then (S25) holds. We split $\boldsymbol{x}^\mu$ into $x_1 \in \boldsymbol{x}^\mu$ and $\boldsymbol{x}^\mu \backslash x_1$ and denote the corresponding probabilities as $p_1 := p(x_1)$, $q_1 := q(x_1)$, $p_1^c := p(\boldsymbol{x}^\mu \backslash x_1)$, and $q_1^c := q(\boldsymbol{x}^\mu \backslash x_1)$. The right-hand side of (S27) is then

$$\begin{aligned}(p_1 + p_1^c) \log \left(\frac{p_1 + p_1^c}{q_1 + q_1^c} \right) &= (p_1 + p_1^c) \log \left(\frac{q_1}{q_1 + q_1^c} \frac{p_1}{q_1} + \frac{q_1^c}{q_1 + q_1^c} \frac{p_1^c}{q_1^c} \right) \\ &\geq q_1 \log \frac{p_1}{q_1} + q_1^c \log \frac{p_1^c}{q_1^c},\end{aligned} \tag{S28}$$

where we used the convexity of the logarithmic function. By repeating the same argument for the second term of (S28), we obtain (S27). Although the KL divergence is our focus, the information monotonicity holds more generally, e.g., for f-divergence [19]. We now use the information monotonicity to estimate the error functions. In the present context, the sets of variables $\boldsymbol{A}$ and $\boldsymbol{x}$ correspond to the cluster assignments of different q's, (σ_i, σ_j) with q and (σ_i', σ_j') with q' $(q' > q)$, for a vertex pair i and j. Because the labeling of the clusters is common to all vertices, we require that the refinement condition is satisfied with the common family of sets for every vertex pair. Under this condition, the KL divergence is nondecreasing as a function of q, which means E_{Bayes} does not saturate earlier than E_{Gibbs}. Similarly, E_{training} does not saturate earlier than E_{Bayes}. Although the refinement condition we required above is too strict to be satisfied exactly in numerical calculation, it is what we naturally expect when the algorithm detects hierarchical structure or the same structure with excess numbers of clusters. Moreover, the argument above is only a sufficient condition. Therefore, we naturally expect that E_{Gibbs} suggests a smaller number of clusters than E_{Bayes} and E_{training} quite commonly in practice. Note also that, if we use a different criterion for the selection of q, e.g., variation of the slope of the error function, the above conclusion can be violated.

DETAILS OF THE ALGORITHM AND FURTHER RESULTS FOR REAL-WORLD NETWORKS

The EM algorithm with BP for cluster inference and hyperparameter learning is based on [9]. When there are multiple local minima in the Bethe free energy, the resulting fixed point varies depending on the initial condition. For the initial values of the hyperparameters $\boldsymbol{\gamma}$ and ω, we examined (i) the values estimated by the spectral method using the normalized Laplacian and k-means algorithm, (ii) equal-size assortative clusters that have equally large values for the diagonal elements of the affinity matrix, and (iii) equal-size clusters with a randomly polarized affinity matrix in which only one element has a large value. We tried these three initial states multiple times and selected the one with the minimum error and Bethe free energy. Note that, although the eigenvectors of the normalized Laplacian might be localized, because the resulting partition obtained by the Bayesian framework is not necessarily close to that of the spectral method, we do not regard the emergence of such eigenvectors as a deterioration. More results on the real-world networks are presented in Fig. S4. The data labels are the same as in the main text. Overall, the Bethe free energy f_{Bethe}, Bayes prediction error E_{Bayes}, and Gibbs training error E_{training} do not exhibit good saturation. In contrast, the Gibbs prediction error E_{Gibbs} often suggests a reasonable number of clusters. The MAP estimate of the Gibbs prediction error E_{MAP} also behaves well in many cases; it tends to suggest a slightly larger q.

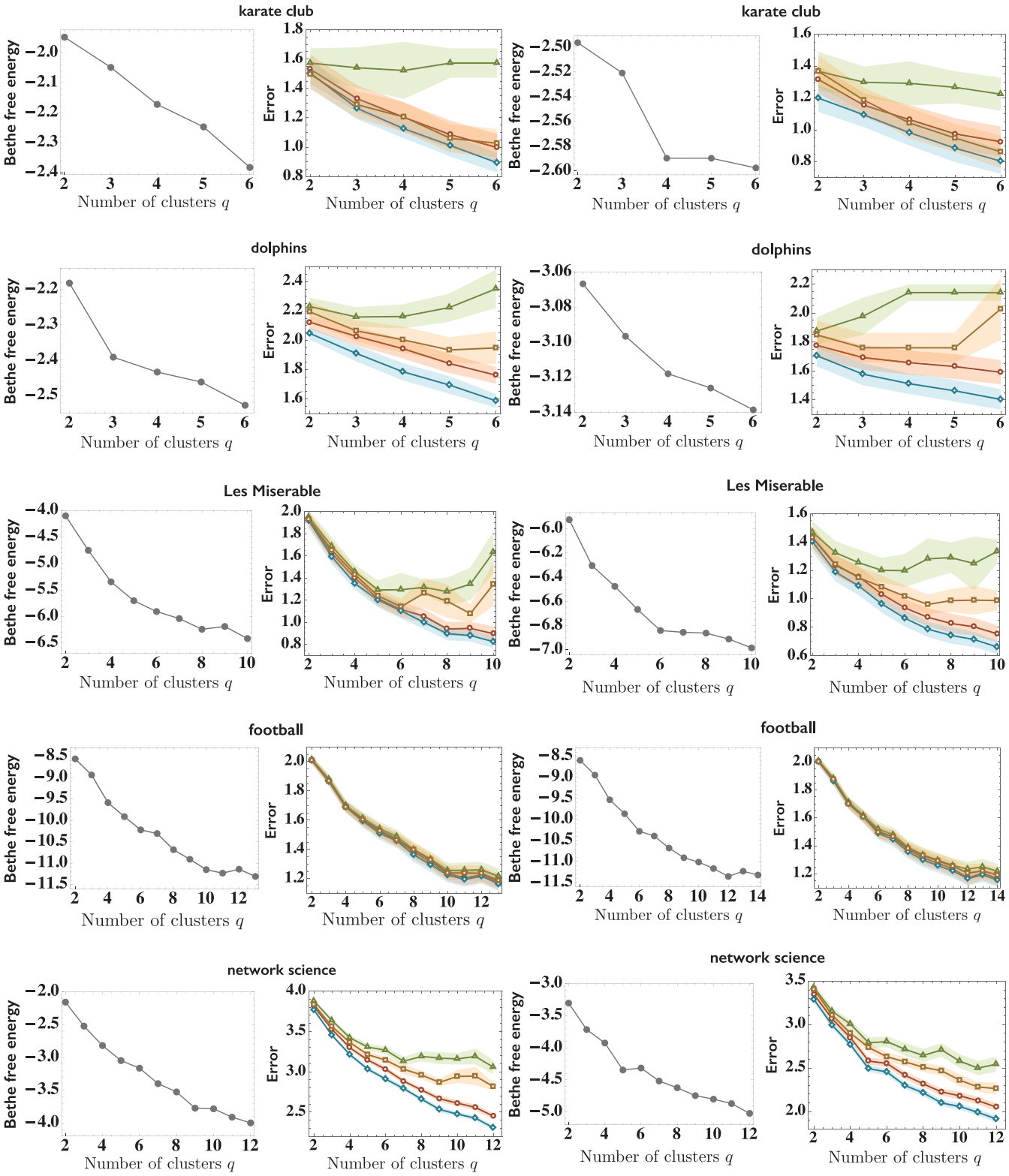

FIG. S3. Bethe free energies and various errors of real-world networks: Zachary's karate club network (*karate club*) [27], social network of frequent associations between dolphins (*dolphins*) [28], coappearance network of characters in the novel *Les Miserables* (*Les Miserables*) [29], network of American football games (*football*) [30], and coauthorship network of scientists working on network theory and experiment (*network science*) [31]. All networks are converted to undirected, unweighted, simple graphs, and only the largest connected components are analyzed. Left two columns show the Bethe free energy and the errors for the standard stochastic block model. Right two columns show the same plot for the degree-corrected stochastic block model. As in the main text, the standard errors of the error functions are plotted as shadows.

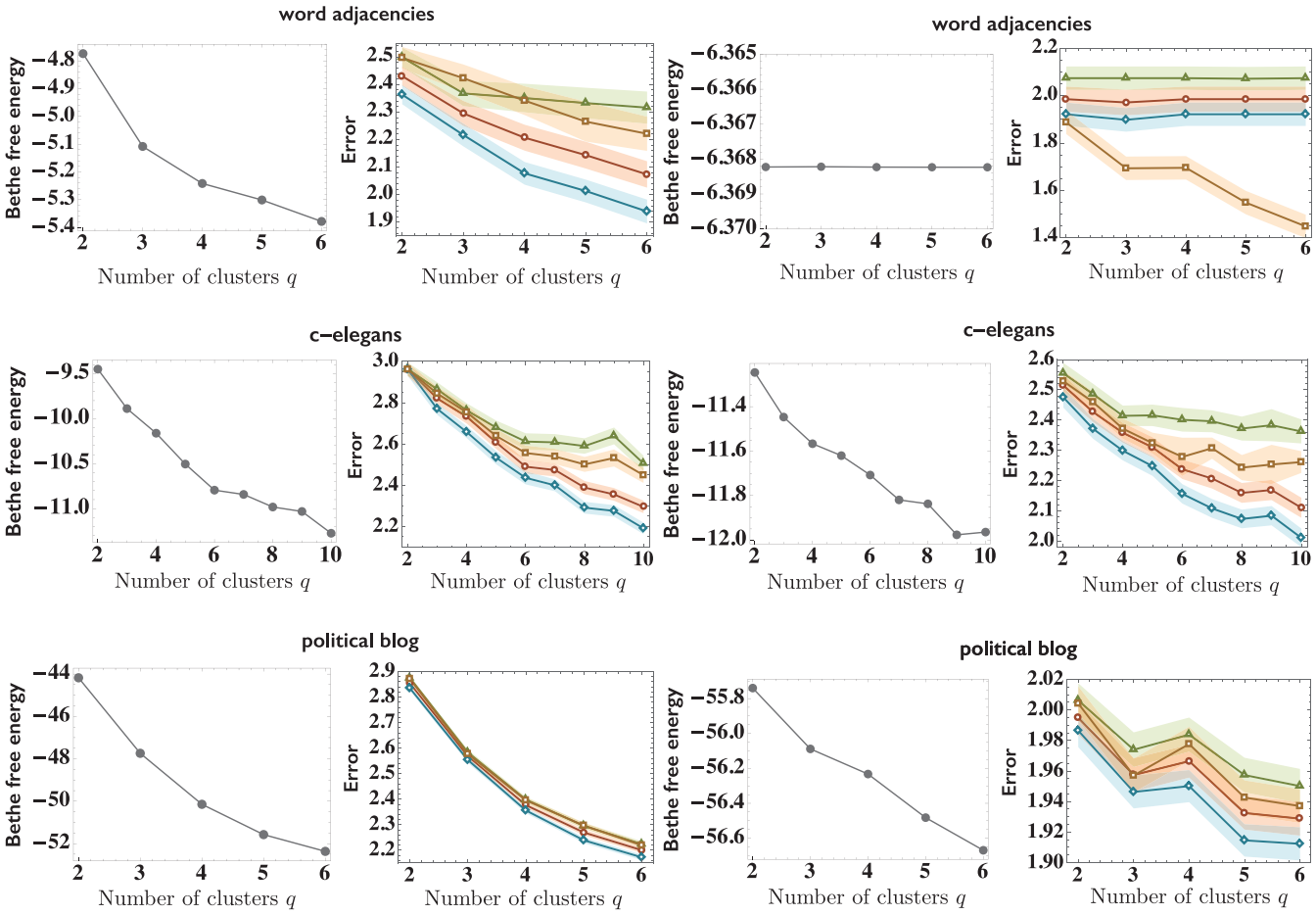

FIG. S4. More examples of real-world networks: adjacency network of common adjectives and nouns in a novel (*word adjacencies*) [31], metabolic network of *C. elegans* (*c-elegans*) [32], and network of hyperlinks between blogs on US politics (*political blog*) [33]. The panels are placed in the same order as in Fig. S4. The word adjacencies network is known to have a bipartite structure. Again, the networks are converted to undirected, unweighted, simple graphs. The c-elegans network and political blog network have hub structures, which are inconsistent with the standard stochastic block model; this inconsistency can be observed clearly for the political blog network. Although the Gibbs prediction error E_{Gibbs} often saturates at a value slightly larger than the number of ground-truth communities, its performance is much better than that of the Bethe free energy and other error functions, as is the case for other real-world networks.