Editors' Suggestion

Data-driven inference of hidden nodes in networks

Danh-Tai Hoang,[1,2] Junghyo Jo,[3,4,*] and Vipul Periwal[1,†]

[1]*Laboratory of Biological Modeling, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, Maryland 20892, USA*

[2]*Department of Natural Sciences, Quang Binh University, Dong Hoi, Quang Binh 510000, Vietnam*

[3]*Department of Statistics, Keimyung University, Daegu 42601, Korea*

[4]*School of Computational Sciences, Korea Institute for Advanced Study, Seoul 02455, Korea*

(Received 7 January 2019; published 10 April 2019)

The explosion of activity in finding interactions in complex systems is driven by availability of copious observations of complex natural systems. However, such systems, e.g., the human brain, are rarely completely observable. Interaction network inference must then contend with hidden variables affecting the behavior of the observed parts of the system. We present an effective approach for model inference with hidden variables. From configurations of observed variables, we identify the observed-to-observed, hidden-to-observed, observed-to-hidden, and hidden-to-hidden interactions, the configurations of hidden variables, and the number of hidden variables. We demonstrate the performance of our method by simulating a kinetic Ising model, and show that our method outperforms existing methods. Turning to real data, we infer the hidden nodes in a neuronal network in the salamander retina and a stock market network. We show that predictive modeling with hidden variables is significantly more accurate than that without hidden variables. Finally, an important hidden variable problem is to find the number of clusters in a dataset. We apply our method to classify MNIST handwritten digits. We find that there are about 60 clusters which are roughly equally distributed among the digits.

DOI: 10.1103/PhysRevE.99.042114

I. INTRODUCTION

To go from observations to predictive understanding is to go from stamp-collecting to science. Absent principled quantitative laws, biological and social systems can be generally described as networks of interacting nodes, with time-series data providing a window on the dynamics of the underlying system. In the present era of big data, the network reconstruction problem has attracted considerable interest in research areas ranging from neuroscience [1–4] and genomics [5–7] to finance [8–11]. A fundamental caveat is that such reconstructions always rely on partial observation of these complex networks. For example, it is hopeless to follow the simultaneous spiking activity of every neuron in the brain, the transcription of every gene in the genome, and every fluctuating factor in a financial system.

The problem of accounting for the unobserved constituents of any system is ill-posed without further information, simply because the number, the interactions, and the configurations of these hidden nodes must all be identified from the observed data and, *a priori*, one can make the former two as large and as complicated, respectively, as one pleases. To render the problem well-defined, one can first choose a theoretical model structure and then account for the unobserved nodes within this structure. Given the importance of this problem, much work has been devoted to it.

A simple approach is to maximize the likelihood of observed configurations after marginalizing unobserved configurations [12]. Another effective approach is the expectation maximization (EM) algorithm for hidden variables that contains two alternating steps, inferring all interactions of observed and hidden variables from configurations of observed variables and reconstructing the configurations of hidden variables consistent with these inferred interactions [13]. As one might expect, this algorithm is computationally impractical for even moderately large systems if the fraction of unobserved variables is significant. Furthermore, hidden variable configuration reconstruction accuracy is greatly dependent on interaction inference accuracy, a factor that becomes significant for limited datasets. Therefore, recent network reconstruction methods have considered alternative approaches, such as mean-field approximations [12,14] and replica methods [15,16]. However, the mean-field approximations work only for weak and dense interactions [14], while the replica methods allow the inference of strong and sparse interactions but impose the stringent assumption of independence between hidden variables [16]. In addition to noninteracting hidden variables, random interaction strengths and the thermodynamic limit are two prerequisites for exact inference using the replica methods [15].

We recently formulated an alternative approach [17,18] to network reconstruction for observed variables that is significantly more accurate inferencewise in the limit of sparse sampling and orders of magnitude faster computationwise than previous methods. Based on this foundation, we propose an alternative approach for network reconstruction including hidden variables, by replacing the inference step with our

*jojunghyo@kmu.ac.kr
†vipulp@mail.nih.gov

approach. This does not, by itself, address the crucial question of the number of unobserved variables, so we complete our proposal by formulating a simple quantitative test of model complexity to determine this number.

This paper is organized as follows: We briefly review our inference method and outline its extension to hidden variables, paying special attention to the determination of the number and interactions of hidden variables, among themselves and with observed variables. We then validate our method with simulated data from kinetic Ising models, showing the accurate determination of the number and interactions of hidden variables for a range of observed fractions of systems, going up to 40% hidden variables. Turning to real data, we apply our method to reconstruct a neural network from partially observed neuronal activities, and a stock-market network using data of opening and closing stock prices of 25 American companies. We validate our network reconstructions by reproducing observed neuronal activities by pinning just a few neuron configurations, and by exhibiting a profitable stock trading strategy based on our inferred network. Finally, we demonstrate that our approach is suited to unsupervised data clustering as well, given that cluster membership is a type of hidden variable. We estimate the number of hidden features that can explain the MNIST handwritten digit dataset. Complete source code with documentation is available [19].

II. METHOD

We explain our approach in the context of a concrete example for ease of understanding. Consider a stochastic dynamical system in which a vector of N binary (± 1) variables $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_N)$ evolves stochastically according to the conditional probability:

$$P(\sigma_i(t+1)|\boldsymbol{\sigma}(t)) = \frac{\exp[\sigma_i(t+1)H_i(t)]}{\exp[H_i(t)] + \exp[-H_i(t)]}, \quad (1)$$

for $i = 1, \cdots, N$, with time $t = 1, \ldots, L$. The local field $H_i(t) = \sum_j W_{ij}\sigma_j(t)$ represents the summed influence of the present state $\sigma_j(t)$ on the future state $\sigma_i(t+1)$ through the weight W_{ij}. This kinetic Ising model has a model expectation, $\langle \sigma_i(t+1)\rangle_{H_i(t)} = \tanh H_i(t)$. Generating $\boldsymbol{\sigma}(t)$ given W_{ij} is easy, but inferring W_{ij} given $\boldsymbol{\sigma}(t)$ is not trivial. Although numerous methods exist for the inverse problem [20–22], we recently proposed an alternative approach [17,18]. We give here a simplified intuitive account. The first step is the linear regression of $H_i = \sum_j W_{ij}\sigma_j$ between H_i and σ_j. Suppose we know $H_i(t)$ and $\sigma_j(t)$. The coefficient W_{ij} can then be obtained as usual for the multivariate linear regression:

$$W_{ij} = \sum_k \langle \delta H_i \delta\sigma_k\rangle [C^{-1}]_{kj}, \quad (2)$$

where $C_{jk} \equiv \langle \delta\sigma_j \delta\sigma_k\rangle$ is the covariance matrix for $\boldsymbol{\sigma}(t)$, with $\langle f\rangle \equiv L^{-1}\sum_{t=1}^{L} f(t)$ and $\delta f \equiv f - \langle f\rangle$. We derived the linear regression in Eq. (2) using the concept of free energy in statistical mechanics [17], so we call this method free-energy minimization (FEM). The second step is the update of the observable,

$$H_i^{\text{new}}(t) \leftarrow \frac{\sigma_i(t+1)}{\langle \sigma_i(t+1)\rangle_{H_i(t)}} H_i(t) = \sigma_i(t+1)\frac{H_i(t)}{\tanh H_i(t)}. \quad (3)$$

The multiplicative update of $H_i(t)$ corrects the magnitude and sign of $H_i(t)$ based on the ratio of observed $\sigma_i(t+1)$ and model expectation $\langle \sigma_i(t+1)\rangle_{H_i(t)}$, which is always larger than unity in absolute magnitude.

A critical aspect of Eq. (3) is that the limit $|H_i| \downarrow 0$ gives $H_i^{\text{new}}(t) \leftarrow \sigma_i(t+1)$, independent of H_i. In intuitive terms, a vanishing local field $H_i(t) = 0$ can lead to $\sigma_i(t+1) = \pm 1$ with equal probability. However, our algorithm forces $H_i^{\text{new}}(t) = \sigma_i(t+1)$, trying to explain the value of $\sigma_i(t+1)$ as a causal result from nonvanishing local field $H_i^{\text{new}}(t)$, instead of interpreting it as a random event. Therefore, the update in Eq. (3) avoids being entirely multiplicative for determining W_{ij}.

These two steps, $H_i(t) \to W_{ij}$ and $W_{ij} \to H_i(t)$, provide a powerful iterative method: (i) given $H_i(t) = \sum_j W_{ij}\sigma_j(t)$, update $H_i(t) \to H_i^{\text{new}}(t)$ using Eq. (3); (ii) obtain W_{ij}^{new} by replacing $H_i(t)$ with $H_i^{\text{new}}(t)$ in Eq. (2); and (iii) repeat these steps after updating $W_{ij}^{\text{new}} \to W_{ij}$. We continue this iteration until the discrepancy between data and model expectation $D_i(W) \equiv \sum_t [\sigma_i(t+1) - \langle \sigma_i(t+1)\rangle_{H_i(t)}]^2$ is minimized. Notice that the parameter update in Eqs. (2) and (3) is completely independent of the computation of D_i. This crucial feature allows the small sample size inference to avoid overfitting because the minimization of D_i is used only as a stopping criterion. This is notably distinct from many maximum likelihood methods that minimize a cost function and stop at a minimal cost configuration, even if this is a local minimum.

Now we propose to apply the FEM method to infer interactions from/to hidden variables. The system $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_N) = (\boldsymbol{\sigma}^v, \boldsymbol{\sigma}^h) = (\sigma_1^v, \cdots, \sigma_{N_v}^v, \sigma_1^h, \cdots, \sigma_{N_h}^h)$ has N_v observed (visible) and N_h hidden variables ($N = N_v + N_h$). As a variant of the EM algorithm, we first assign random configurations for hidden variables. We then infer interaction weights W_{ij} for observed-to-observed, hidden-to-observed, observed-to-hidden, and hidden-to-hidden variables with the FEM method. Given W_{ij}, we can update the configurations of hidden variables as follows. First, we define the likelihood for $\sigma_J^h(t) = \pm 1$:

$$\mathcal{L}_{J,t}^{\pm} = P\big(\sigma_J^h(t) = \pm 1|\boldsymbol{\sigma}(t-1)\big)\prod_{i=1}^{N} P(\sigma_i(t+1)|F_J^{\pm}[\boldsymbol{\sigma}(t)]), \quad (4)$$

where we use a flip operator $F_J^{\pm}[\boldsymbol{\sigma}] = (\sigma_1^v, \cdots, \sigma_J^h = \pm 1, \cdots, \sigma_{N_h}^h)$. Note that $\mathcal{L}_{J,L}^{\pm} = P(\sigma_J^h(L) = \pm 1|\boldsymbol{\sigma}(L-1))$. Then, we assign $\sigma_J^h(t) = \pm 1$ using the likelihood ratio $\mathcal{L}_{J,t}^{\pm}/(\mathcal{L}_{J,t}^{+} + \mathcal{L}_{J,t}^{-})$. The iterations between the parameter optimization (M step) and the variable update (E step) provide accurate inference of the interaction weights, W_{ij}, and the unknown configurations of hidden variables. Note, in particular, that the hidden variables are updated one by one from $t = 1$ to $t = L$ because the likelihood of each hidden variable depends on the state of other hidden variables.

We must now consider the problem of determining the number of hidden variables. A simple measure would be the same discrepancy between observation $\sigma_i^v(t+1)$ and model expectation $\langle \sigma_i^v(t+1)\rangle_{H_i(t)}$,

$$D_v \equiv \sum_{i=1}^{N_v} D_i(W), \quad (5)$$

but this is clearly not taking the hidden variables into account. However, extending the sum in Eq. (5) to include hidden variables is useless because the E step update is minimizing these additional terms explicitly. Nevertheless, as we know nothing about hidden states, the inference error of hidden states cannot be assumed to be smaller than the inference error of visible states because the hidden states are being inferred as a function of the visible states in the E step. Thus, motivated by standard error propagation, for an unbiased estimation we assume that both inference errors have the same scale, and define the scaled discrepancy of the entire system based on the observed part as

$$D \equiv D_v\left(1+\frac{N_h}{N_v}\right). \tag{6}$$

The scaled discrepancy can be interpreted as balancing the goodness of fit for observed variables, D_v, and hidden variable model complexity, $(1+N_h/N_v)$. The empirical discrepancy is similar in spirit to the Akaike information criterion [23] and Bayesian information criterion [24] with log-likelihood of observation $(-\log \mathcal{L}_v \sim D_v)$ and model degrees of freedom (N_v+N_h). Crucially, note here that the information criteria are formulated for complexity in terms of the number of model parameters. Hidden variables, however, add two types of complexity, both an increase in the number of couplings and an increase in the number of states underlying every visible state, which are themselves being determined by the E step.

Finally, our method can be summarized as the following set of steps:

For a range of numbers of hidden variables, in parallel and independently,

(i) Assign configurations of hidden variables at random;

(ii) Infer interaction weights W_{ij} including observed-to-observed, hidden-to-observed, observed-to-hidden, and hidden-to-hidden from the configurations of observed and hidden variables using FEM;

(iii) Flip the states $\sigma_j^h(t)$ of hidden variables using the likelihood ratio $\mathcal{L}_{J,t}^{\pm}/(\mathcal{L}_{J,t}^{+}+\mathcal{L}_{J,t}^{-})$ [see Eq. (4)];

(iv) Repeat steps (ii) and (iii) until the discrepancy of observed variables is minimized. The final values of W_{ij} and hidden states are the inferred coupling weights and configurations of hidden spins, respectively.

Pick the number of hidden variables that minimizes Eq. (6).

III. RESULTS

A. Kinetic Ising model

To demonstrate the performance of our method, we synthesized binary time series of $N=100$ spins by using the Sherrington-Kirkpatrick model [25]. The update of spin $\boldsymbol{\sigma}$ follows Eq. (1) with preset coupling strengths W_{ij} [Fig. 1(a)]. Our goal is to reconstruct all of W_{ij} from observations of a fraction of $\boldsymbol{\sigma}(t)$. Suppose that we only observe the time series of 60 spins with 40 spins hidden [Fig. 1(b)]. When we reconstructed the interactions W_{ij} between observed node i and observed node j using FEM, the reconstructed W_{ij}, based on the partial observations, showed a large error [Fig. 1(c)]. We introduced 40 hidden variables, and applied the EM algorithm outlined in the previous section. The reconstructed

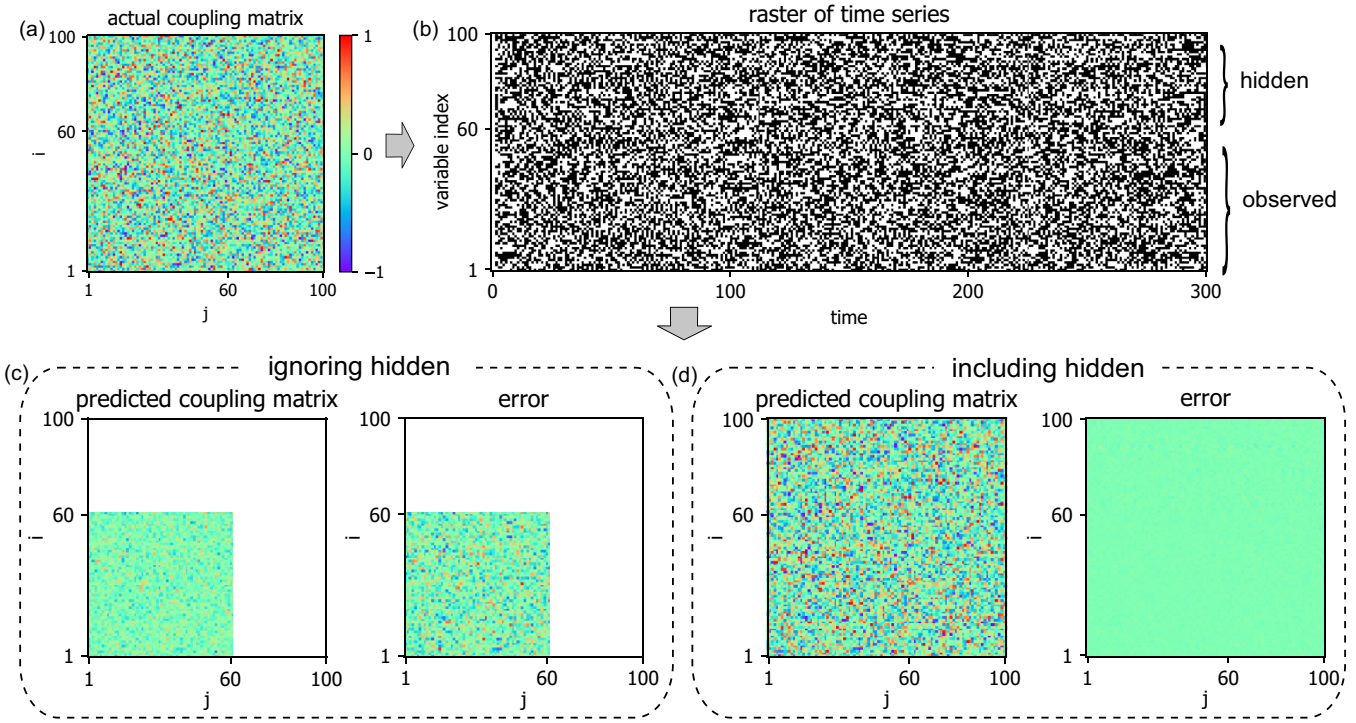

FIG. 1. Network reconstruction from partial observations. From the actual interaction weights (a), typical time series of 100 variables are generated according to the kinetic Ising model (b). Using the configuration of 60 observable variables, the interaction weights are recovered in two cases: ignoring (c) and including (d) the existence of hidden variables. Error represents $(W_{ij}^{\text{true}}-W_{ij})$ with the same scale of the color bar as in (a). Data length $L=40\,000$ is used.

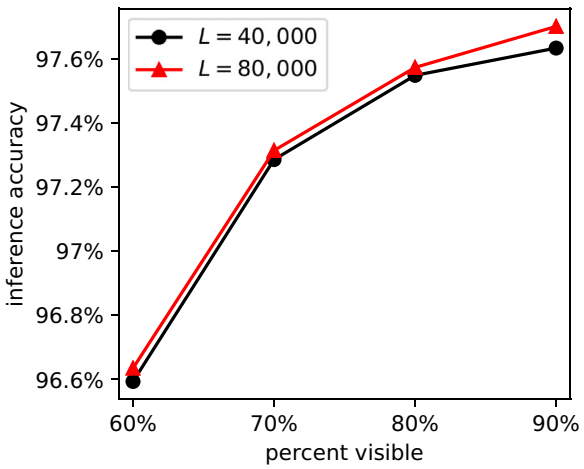

FIG. 2. Accuracy of network reconstruction and fraction of hidden variables. The inference accuracy is plotted as a function of the fraction of visible variables, N_v/N. Results for system size $N = 100$ and data lengths $L = 40\,000$ (black circles) or $L = 80\,000$ (red triangles) are shown.

W_{ij} was close to the true W_{ij} [Fig. 1(d)]. How well are the hidden variable configurations recovered? For the case of 40 hidden variables, the true configurations of the hidden variables were recovered with an accuracy of 96.6%. The reconstruction accuracy increased with fewer spins hidden (Fig. 2). For instance, when 90 spins were observed with 10 spins hidden, the accuracy was 97.6%.

The number of hidden variables is usually unknown in real-world problems. When we reconstructed W_{ij} with different numbers of hidden variables, the mean-square errors of observed-to-observed interaction strengths, $\mathrm{MSE} = N_v^{-2}\sum_{i,j\in\mathrm{obs}}(W_{ij} - W_{ij}^{\mathrm{true}})^2$, were minimal at the right number of hidden variables (Fig. 3, upper panel). The MSE is also inaccessible in real-world problems, but the minimum of D [Eq. (6)] captured the correct value of N_h (Fig. 3, lower panel, red lines).

To reconstruct W_{ij} from observed and hidden variables, we used FEM. For the M step, mean field methods such as naïve, Thouless-Anderson-Palmer, and exact mean-field methods (nMF, TAP, and eMF) and maximum likelihood estimation (MLE) can also be used. A brief review of these methods can be found in Ref. [17]. Given partial observations, mean field approaches were not successful in reconstructing W_{ij} (Fig. 4). For a small percentage of hidden variables (90 observable and 10 hidden), FEM and MLE showed a similar performance in the reconstruction of observed-to-observed and hidden-to-observed interactions. However, FEM outperformed MLE in reconstructing observed-to-hidden and hidden-to-hidden interactions [Figs. 4(a)–4(d)]. For a large percentage of hidden variables (60 observable and 40 hidden variables), FEM showed significantly better performance even for observed-to-observed and hidden-to-observed interactions [Figs. 4(e)–4(h)]. We quantified the reconstruction performance by measuring MSE between W_{ij} and W_{ij}^{true} [Figs. 4(i)–4(l)]. FEM showed more accurate reconstruction of W_{ij} with lower MSE in every case. More importantly, in addition to better performance, FEM took approximately 100 times less computation time than MLE due to its multiplicative update (see Refs. [17–19] for details).

B. Neuronal network

The analysis of real data brings out issues far more clearly than simulated validations. Therefore, we applied our method to infer hidden nodes and their contributions in a real neuronal network. We used the time series data of the 80 most active neurons from published multi-channel recordings of neuronal firing in the salamander retina [26]. Considering the existence of unobserved hidden variables, we modeled the evolution of neuronal activities by defining a local field, $H_i(t) = H_i^{\mathrm{ext}} + \sum_j W_{ij}\sigma_j(t)$, that determines the future activity of $\sigma_i(t+1)$.

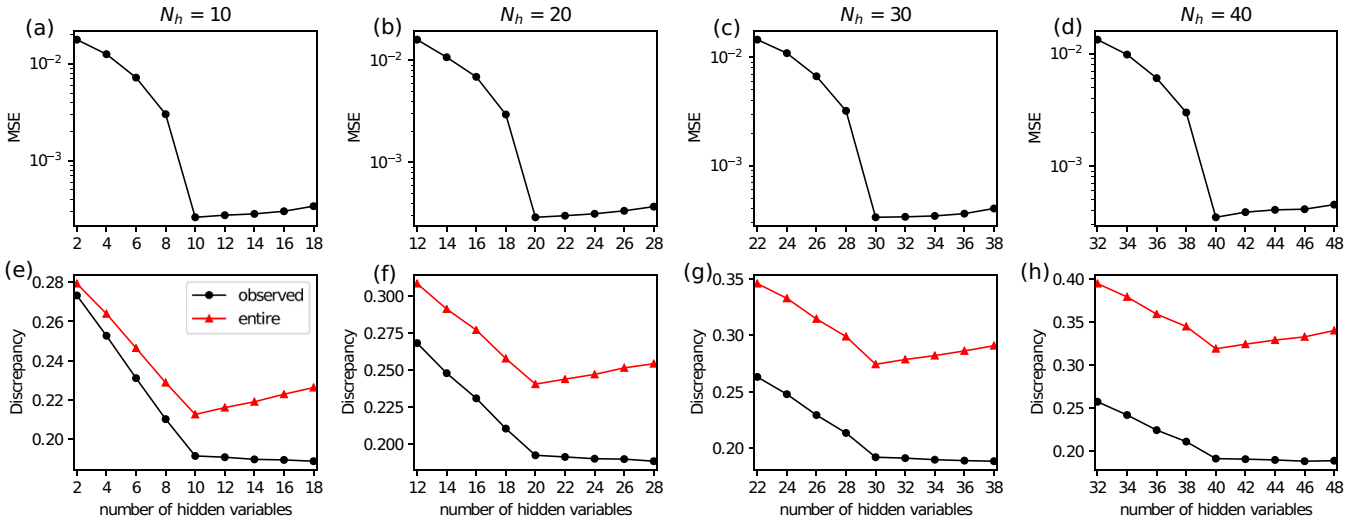

FIG. 3. Estimation of hidden degrees of freedom. Mean square errors of observed interaction strengths (upper) and discrepancy D_v of observed variables (lower, black circles) and discrepancy D of total (observed and hidden) variables (lower, red triangles) are shown with various assumed numbers N_h of hidden variables. The actual numbers of hidden variables are $N_h = 10$, 20, 30, and 40, from left to right. Results with system size $N = 100$ and data length $L = 40\,000$ are shown.

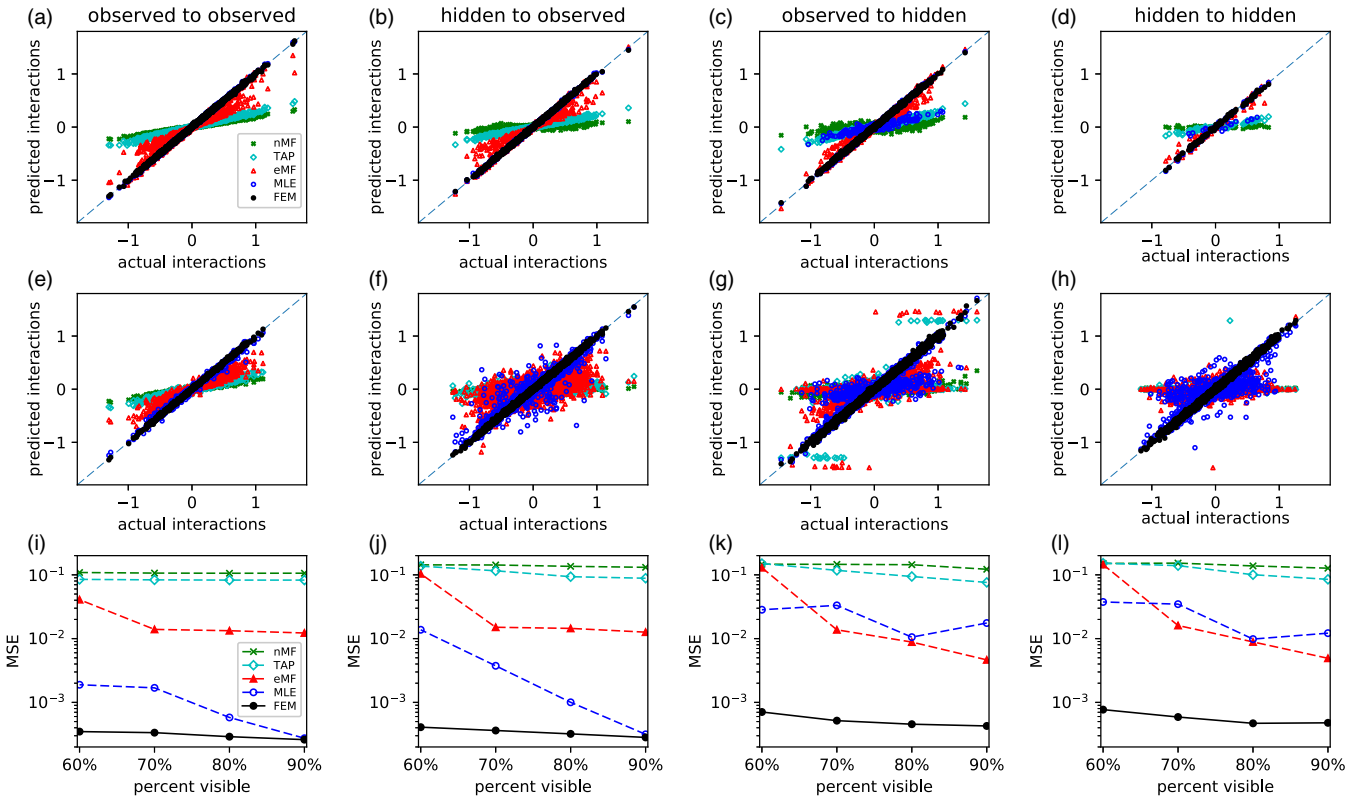

FIG. 4. Performance comparison between inference methods. Predicted interactions versus actual interactions, separately in observed-to-observed (a and e), hidden-to-observed (b and f), observed-to-hidden (c and g), and hidden-to-hidden (d and h), for two numbers of hidden variables $N_h = 10$ (first row) and 40 (second row). The mean square errors between predicted interactions and actual interactions are shown as a function of the fraction (N_v/N) of observed variables over total variables (i–l). We compared five inference methods: naïve mean-field (nMF), Thouless-Anderson-Palmer (TAP), exact mean-field (eMF), maximum likelihood estimation (MLE), and free-energy minimization (FEM). Results with system size $N = 100$ and data length $L = 40\,000$ are shown. For MLE, we used a learning rate $\alpha = 1$.

The external local field H_i^{ext} represents the bias of the ith neuron that sets its threshold [17]. For various numbers of hidden variables N_h, we computed H_i^{ext} and W_{ij}. The activities of observed neurons were explained better and D_v kept decreasing with a larger N_h of hidden variables [Fig. 5(b)]. However, once we considered the overall discrepancy D, an optimal number of hidden variables was $N_h^* = 4$. Thus, the inclusion of four hidden variables best explained the activities of observed neurons, taking model complexity into account. Given these four hidden variables, the connection weights W_{ij} were reconstructed as shown in Fig. 5(c).

Since W_{ij}^{true} is unknown for the neuronal network, we validated our reconstruction in two different ways. First, we selected some neurons as input neurons, and then based on the activities of these input neurons, we generated the activities of remaining neurons by using the reconstructed W_{ij}. Here we selected the input neurons based on having the strongest influence to other neurons by gauging $\sum_i |W_{ij}|$. Given varying numbers of input neurons, we could successfully reconstruct the actual activities of the remaining neurons [Fig. 5(d)]. As the number of input neurons increased, the reconstruction accuracy increased [Fig. 5(e)]. Moreover, the inference accuracy for including hidden variables was significantly better than that for ignoring hidden variables [Fig. 5(e)]. Second, given $\boldsymbol{\sigma}(t)$, we predicted $\boldsymbol{\sigma}(t+1)$, and then calculated the covariance $C_{ij} = \langle \delta\sigma_i(t+1)\delta\sigma_j(t)\rangle$. The reconstructed covariance was plotted to compare with the actual covariance from the observation. The prediction performance was significantly improved when the 4 hidden variables were included [Figs. 5(f) and 5(g)].

C. Stock network

Our method has a wide range of practical applications. As a demonstration, we reconstruct a stock market network with possible hidden nodes. We used stock price time series of 25 major companies in the S&P 500 index in five different sectors: technology (AAPL, GOOGL, MSFT, INTC, IBM), finance (BRK.B, JPM, WFC, BAC, C), health care (JNJ, PFE, UNH, MRK, AMGN), consumer discretionary (AMZN, WMT, HD, DIS, EBAY), and energy and industrial (XOM, CVX, GE, BA, MMM) [27]. We examined the price difference between daily opening and closing stock prices. Their fluctuations from January 2005 to July 2018 are shown in [Fig. 6(a)]. First, instead of considering the continuous price fluctuations, we defined a discretized measure of price changes. If the daily price increased at time t for the ith company (opening price $<$ closing price), then we defined $\sigma_i(t) = +1$. However, if the price decreased (opening price $>$ closing price), then $\sigma_i(t) = -1$. Finally, if the price was unchanged (opening price $=$ closing price), then we defined $\sigma_i(t) = \sigma_i(t-1)$.

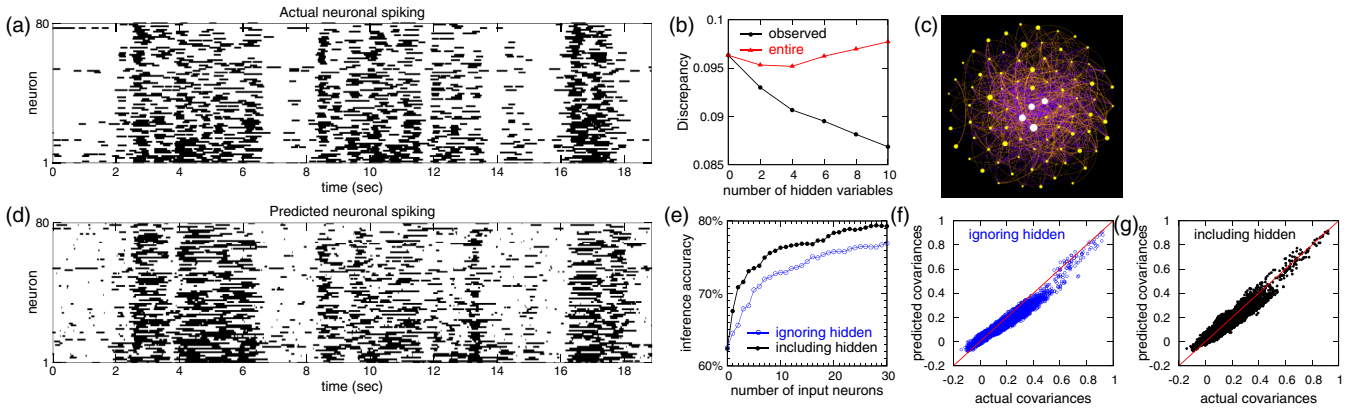

FIG. 5. Neural network reconstruction with hidden nodes. Activities of the 80 most active neurons are plotted with black dots representing active states and white dots representing silent states (a). Discrepancies between observed and expected neuronal activities (black circles) and discrepancies between entire neuronal activities and their expectations (red triangles) are shown as a functions of the number of hidden variables (b). A predicted neuronal network is visualized in which green nodes represent observed neurons, while white nodes represent hidden variables. The red and blue edges represent positive and negative couplings, respectively. Edge direction is clockwise. The node size scales with value of firing rate, and edge thickness scales with coupling weight (c). Given the reconstructed coupling strengths, external local fields, and configurations of hidden variables, the activities of 80 neurons are reconstructed (d). Inference accuracy of remaining neuronal activities are shown as a function of the number of input neurons for ignoring (blue) and including (black) the existence of hidden variables (e). Inferred covariances C_{ij} versus actual covariances C_{ij}^{true} when hidden variables are ignored (f) and included (g).

We applied our method to this discretized data and inferred external factors H_i^{ext} and interacting factors $\sum_j W_{ij}\sigma_j(t)$ that stochastically determine $\sigma_i(t+1)$. Since FEM works well even for small sample sizes [17], we divided the data into two-year periods to probe possible slower temporal changes in the interactions W_{ij} between stock prices. In particular, we show results from more recent data for 2014 to 2016 [Figs. 6(b)–6(d)] and 2016 to 2018 [Figs. 6(e) and 6(g)]. The discrepancy D_v between observed $\sigma_i^v(t+1)$ and model expectation $\langle\sigma_i^v(t+1)\rangle_{\text{model}}$ kept decreasing as expected when more hidden nodes were introduced [Figs. 6(b) and 6(e)]. However, the entire discrepancy D, considering the model complexity with hidden variables, showed a minimum at $N_h^* \approx 4$–5 hidden nodes. The inferred stock market network including interactions between observed and hidden nodes is visualized in Figs. 6(c) and 6(f). When we generated time series of stock prices using the reconstructed network, we found that the covariance of the generated sequences was consistent with the covariance of original sequences [Figs. 6(d) and 6(g)].

An accurate predictive network reconstruction should enable profitable trades. In particular, does our discrete reduction of the price data still contain enough information to be useful? First, we reconstructed the interactions between companies including the appropriate number of hidden nodes by using stock price data for the most recent T days: $\boldsymbol{\sigma}^v(t-T+1), \boldsymbol{\sigma}^v(t-T+2), \cdots, \boldsymbol{\sigma}^v(t)$. Then, we predicted the price change direction $\boldsymbol{\sigma}^v(t+1)$ for the next day. Our strategy was to buy the stock i that had the highest probability of increasing with a maximum $H_i(t)$, and to sell the stock j that had the highest probability of decreasing with a minimum $H_j(t)$ at the beginning of the day. This trading strategy is expected to have a maximum profit bounded by [close price(i) − open price(i)] + [open price(j) − close price(j)]. The trading simulation from 2008 to 2018 with a moving time window $T = 500$ days obtained 350% cumulative profit [Fig. 6(h)]. This profit was significantly higher than the profit of 50% using random trades, which is due to the secular rise of the entire stock index. Furthermore, the reconstructed network including hidden nodes showed a larger profit than the 250% profit from the reconstructed network ignoring the hidden nodes. Next, we refined the trading strategy by buying and selling the stock that has the highest probability of increasing and decreasing but only if its price has decreased and increased on the previous day. In particular, this may result in only buying or only selling on any specific day. This strategy produced the same cumulative profit in total, but it doubled the profit per transaction [Figs. 6(i) and 6(j)]. Finally, we confirmed that the optimal time window for the highest profit was about $T = 500$ days [Figs. 6(j)].

D. Classification of handwritten digits

Another potential application of our method, interpreting hidden states as labels, is for unsupervised classification. We demonstrate this idea with the MNIST data of handwritten digits [28]. The data has 60 000 digit samples of 28×28 pixel gray-scale (between 0 and 255) images obtained from 500 different individuals. Some of sample digits are shown in Fig. 7(a). Our goal is to classify the 60 000 images into distinct clusters where each cluster represents different digits as well as different writing styles without using true labels for unsupervised classification. We formulated the classification problem as follows. Different digits and writing style combinations are encoded in hidden variable states $\boldsymbol{\sigma}^h$. Then, one realization of $\boldsymbol{\sigma}^h(t)$ generates a digit image $\boldsymbol{\sigma}^v(t)$, where t is now being used to index the MNIST images. The feature has N_h degrees of freedom with $\sigma_J^h(t), J = 1, \cdots, N_h$. In particular, for simplicity, we adopted one-hot encoding by assigning only one nonzero element $\sigma_J^h(t) = 1$ among N_h elements of $\boldsymbol{\sigma}^h(t)$.

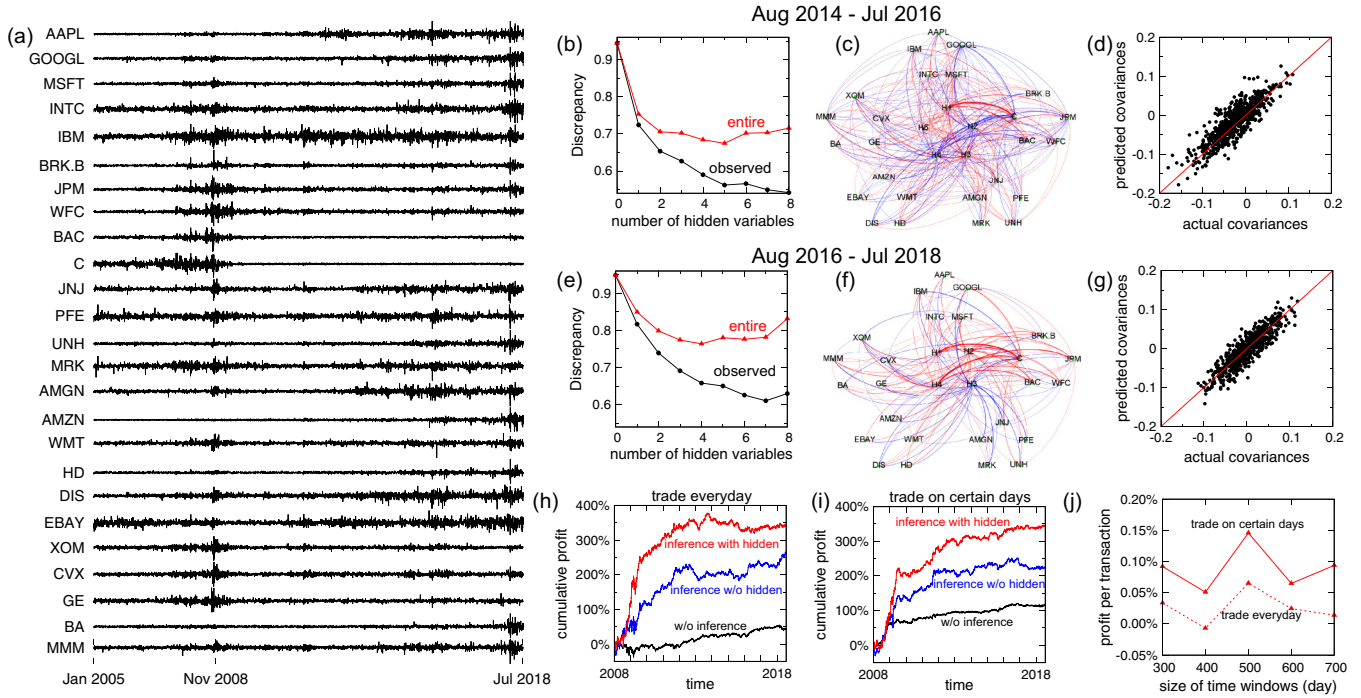

FIG. 6. Stock-market network reconstruction with hidden nodes. The time series of the difference between opening and closing prices of 25 American companies are shown from January 2005 to July 2018 (a). The network reconstruction considered for different periods: from August 2014 to July 2016 (b, c, d), and from August 2016 to July 2018 (e, f, g). Discrepancies between observed and expected configurations (black circles) and discrepancies between entire variables and their expectations (red triangles) are computed to estimate the necessary number of hidden variables (b, e). Reconstructed stock-market networks are visualized (c, f) in which the red and blue edges represent positive and negative couplings, respectively. Edge direction is clockwise, and edge thickness scales with coupling strengths. Inferred covariances C_{ij} versus actual covariances C_{ij}^{true} (d, g). Cumulative profits are shown as a function of the time period for trade on everyday (fully trade) (h) and on certain days (alternative trade) (i) with different trade strategies: random trades (black), strategic trades ignoring (blue), and including hidden variables (red). Profit per transaction versus time window size for the network reconstruction (j), for everyday trade (dashed line) and certain-day trade (solid line).

Then, the generated image has binary values of $\sigma_i^v(t) = 1$ (gray > 1) or $\sigma_i^v(t) = -1$ (otherwise) for the ith pixel, which is determined by the conditional probability,

$$P(\sigma_i^v(t) = \pm 1 | \boldsymbol{\sigma}^h(t)) = \frac{\exp[\pm H_i(t)]}{\exp[H_i(t)] + \exp[-H_i(t)]}, \quad (7)$$

where $H_i(t) \equiv \sum_J W_{iJ}\sigma_J^h(t)$ represents a local field acting on the ith pixel. Here, to reduce the computational cost, we ignored observed pixels i if more than 95% samples had the same value. The threshold 95% showed similar results as a more restrictive threshold of 99%. Thus, for this setup, our reconstruction method considers only hidden-to-observed interactions. Note that unlike the general setup in Eq. (1), this setup assumes no observed-to-observed, observed-to-hidden, and hidden-to-hidden interactions. Furthermore, here the label t is an image index, not time as in the previous examples. The hidden state $\boldsymbol{\sigma}^h(t)$ with label t affects the visible state $\sigma_i^v(t)$ with the *same* label t. Briefly summarizing the inference procedure, we (i) assign a random binary vector $\boldsymbol{\sigma}^h(t)$, in which only one element has nonzero value ($\sigma_J^h(t) = 1$); (ii) apply FEM to reconstruct the interaction strength W_{iJ} from hidden label J to observed pixel i; (iii) update the hidden states by assigning $\sigma_J^h(t) = 1$ for the label J that makes the likelihood of the observed pixels of sample t the highest and $\sigma_J^h(t) = 0$ for the other $N_h - 1$ elements; (iv) repeat steps (ii) and (iii) until the discrepancy D_v between $\sigma_i^v(t)$ and $\langle\sigma_i^v(t)\rangle_{H_i(t)} = \tanh H_i(t)$ saturates. Then, the one-hot hidden states $\boldsymbol{\sigma}^h$ represent distinct classes of MNIST images $\boldsymbol{\sigma}^v$.

We examined various possible numbers (10 to 100) of labels by varying the number N_h of hidden variables. As the hidden degrees of freedom N_h increased, the model generated images of $\sigma_i^v(t)$ closer to the originals. In other words, the discrepancy D_v kept decreasing as N_h increased [Fig. 7(b)]. However, once the model complexity was penalized with the overuse of the hidden degrees of freedom, an optimal degrees of freedom N_h^* was determined with a minimum overall discrepancy D. The estimate $N_h^* \approx 60$ means that the 60 000 MNIST images can be optimally clustered into about 60 classes of digits and writing styles. The mean images $1/N_c \sum_{t\in c} \sigma_i^v(t)$ corresponding to the 60 labels are shown in Fig. 7(c). Here N_c is the number of samples corresponding to label c. It is of particular interest that each digit was divided into approximately six classes, suggesting that, in the MNIST dataset, about six different writing styles exist for every digit. To confirm the robustness of this result, we repeated the analysis with only 20 000 of the MNIST images, and obtained a similar conclusion [dashed lines in Fig. 7(b)].

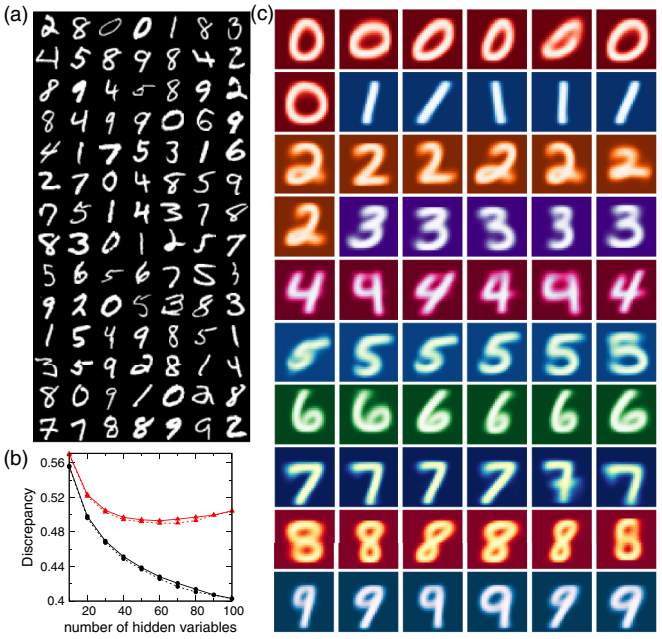

FIG. 7. Hidden degrees of freedom for the classification of handwritten digit images. (a) 98 image samples were randomly selected from the training set of the MNIST data. (b) Discrepancies between observed and expected configurations (black circles) and discrepancies between entire variables and their expectations (red triangles) are shown as a function of the number of hidden variables. For the clustering, we used 60 000 samples (solid lines) and 20 000 samples (dashed lines). (c) Mean images of each cluster were obtained from our inference method with 60 hidden variables. Difference colors are used just to distinguish different clusters.

IV. SUMMARY

Given partial observations of systems, complete network reconstruction is a longstanding problem in inference. In this paper, we propose an alternative iterative approach based on free-energy minimization (FEM) and expectation maximization. We demonstrated on simulated systems that our method can accurately estimate the actual number of hidden variables from partial observations. Furthermore, network reconstruction was successful in recovering not only observed-to-observed interactions but also those involving hidden variables (hidden-to-observed, observed-to-hidden, and hidden-to-hidden). Hidden-to-hidden interactions are challenging to reconstruct with mean-field methods [12,14]. We applied this method to reconstruct a real neuronal network and a stock market network with the inclusion of possible hidden variables. The reconstructed networks were then validated by reproducing real neuronal activities and by a profitable trade simulation, respectively. Finally, as another potential application to unsupervised pattern classification, we found hidden labels in hand-written digit data.

FEM is more effective for network reconstruction than MLE, because it separates the cost function evaluation from the independent multiplicative parameter update. This has two major benefits that are crucial for the application to hidden variable problems to succeed. The first is that the cost function can be used as a stopping criterion to avoid overfitting for small sample sizes, important when considering large numbers of possible hidden variables. The second is that the multiplicative update is computationally much more efficient (approximately 100 times faster than usual MLE-based network reconstruction methods), also critical for determining the configurations of hidden variables. The computational cost for the M and E steps depends on the number of observed variables (N_v), number of hidden variables (N_h), and data length (L). The computational cost of the M step is proportional to $N_h + N_v$ while the computational cost of the E step is proportional to $N_h \times L$. For the kinetic Ising model (Fig. 1), with $N_v = 60$, $N_h = 40$, and $L = 40\,000$, the computational cost of the M step is less than that of E step. However, for the stock data (Fig. 6), with $N_v = 25$, $N_h \sim 4$, and $L = 500$, the computational cost of the M step is larger. Since the algorithm reconstructs interactions strengths W_{ij} from the jth node to the ith node independently for the ith node, the network reconstruction can be easily parallelized, and therefore scaled to large system sizes.

In this paper, we focused on Markovian dynamics with constant linear interactions for clear demonstration. We focused only on inference, without addressing prediction issues, though we did provide evidence in the stock market case that there is predictive value in our inferred model. However, little modifications can cover more general problems. For example, FEM is applicable for inferring polynomial nonlinear interactions as well as linear ones [17]. In addition, to probe the time dependence of the interaction strengths, one may divide data into groups of data with different time windows. Then one can examine the temporal modulation of the interactions by inferring the interactions for different time periods. Finally, if one rearranges data, e.g., $\tilde{\boldsymbol{\sigma}}(t) = [\boldsymbol{\sigma}(t), \boldsymbol{\sigma}(t-1)]$, one can apply FEM to infer the influence of present and one-step past states. We expect, therefore, that FEM can be used for many applications of network inference.

ACKNOWLEDGMENTS

This work was supported by Intramural Research Program of the National Institutes of Health, NIDDK (D.-T.H.,V.P.), and by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. 2016R1D1A1B03932264) (J.J.).

[1] E. Schneidman, Michael J. Berry II, R. Segev, and W. Bialek, Nature **440**, 1007 (2006).

[2] D. A. Dombeck, A. N. Khabbaz, F. Collman, T. L. Adelman, and D. W. Tank, Neuron **56**, 43 (2007).

[3] J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer, Proc. Natl. Acad. Sci. USA **113**, E1074 (2016).

[4] D. Bernal-Casas, H. J. Lee, A. J. Weitz, and J. H. Lee, Neuron **93**, 522 (2017).
[5] T. R. Lezon, J. R. Banavar, M. Cieplak, A. Maritan, and N. V. Fedoroff, Proc. Natl. Acad. Sci. USA **103**, 19033 (2006).
[6] G. Hickman and C. Hodgman, J. Bioinformat. Comput. Biol. **7**, 1013 (2009).
[7] Z. Bar-Joseph, A. Gitter, and I. Simon, Nature Rev. Genet. **13**, 552 (2012).
[8] S. Pincus and R. E. Kalman, Proc. Nat. Acad. Sci. USA **101**, 13709 (2004).
[9] C. K. Tse, J. Liu, and F. C. Lau, J. Empir. Finance **17**, 659 (2010).
[10] B. M. Tabak, T. R. Serra, and D. O. Cajueiro, Physica A **389**, 3240 (2010).
[11] T. Bury, Physica A **392**, 1375 (2013).
[12] B. Dunn and Y. Roudi, Phys. Rev. E **87**, 022127 (2013).
[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, J. Roy. Stat. Soc. Ser. B (Methodological) **39**, 1 (1977).
[14] J. Tyrcha and J. Hertz, Math. Biosci. Engineer. **11**, 149 (2014).
[15] L. Bachschmid-Romano and M. Opper, J. Stat. Mech.: Theory Exp. (2014) P06013.
[16] C. Battistin, J. Hertz, J. Tyrcha, and Y. Roudi, J. Stat. Mech.: Theory Exp. (2015) P05021.
[17] D.-T. Hoang, J. Song, V. Periwal, and J. Jo, Phys. Rev. E **99**, 023311 (2019).
[18] D.-T. Hoang, J. Song, V. Periwal, and J. Jo, Network Inference in Stochastic Systems, https://nihcompmed.github.io/network-inference/.
[19] D.-T. Hoang, J. Jo, and V. Periwal, Network Inference with Hidden Variables, https://nihcompmed.github.io/hidden-variable/.
[20] Y. Roudi and J. Hertz, Phys. Rev. Lett. **106**, 048702 (2011).
[21] M. Mézard and J. Sakellariou, J. Stat. Mech.: Theory Exp. (2011) L07001.
[22] H.-L. Zeng, M. Alava, E. Aurell, J. Hertz, and Y. Roudi, Phys. Rev. Lett. **110**, 210601 (2013).
[23] H. Akaike, IEEE Trans. Auto. Control **19**, 716 (1974).
[24] G. Schwarz, Ann. Stat. **6**, 461 (1978).
[25] D. Sherrington and S. Kirkpatrick, Phys. Rev. Lett. **35**, 1792 (1975).
[26] G. Tkačik, O. Marre, D. Amodei, E. Schneidman, W. Bialek, and M. J. Berry, II, PLoS Comput. Biol. **10**, e1003408 (2014).
[27] Fusion Media Limited, Stock Quotes: SP 500, https://www.investing.com/equities/.
[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Proc. IEEE **86**, 2278 (1998).