Toward link predictability of complex networks

Linyuan Lü[a,1,2], Liming Pan[a,b,1], Tao Zhou[b,c,1], Yi-Cheng Zhang[a,d,2], and H. Eugene Stanley[a,e,2]

[a]Alibaba Research Center for Complexity Sciences, Alibaba Business College, Hangzhou Normal University, Hangzhou 311121, China; [b]CompleX Lab, Web Sciences Center and [c]Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, China; [d]Department of Physics, University of Fribourg, Fribourg CH-1700, Switzerland; and [e]Department of Physics and Center for Polymer Studies, Boston University, Boston, MA 02215

Contributed by H. Eugene Stanley, December 31, 2014 (sent for review March 10, 2014; reviewed by Giorgio Parisi and Dashun Wang)

The organization of real networks usually embodies both regularities and irregularities, and, in principle, the former can be modeled. The extent to which the formation of a network can be explained coincides with our ability to predict missing links. To understand network organization, we should be able to estimate link predictability. We assume that the regularity of a network is reflected in the consistency of structural features before and after a random removal of a small set of links. Based on the perturbation of the adjacency matrix, we propose a universal structural consistency index that is free of prior knowledge of network organization. Extensive experiments on disparate real-world networks demonstrate that (*i*) structural consistency is a good estimation of link predictability and (*ii*) a derivative algorithm outperforms state-of-the-art link prediction methods in both accuracy and robustness. This analysis has further applications in evaluating link prediction algorithms and monitoring sudden changes in evolving network mechanisms. It will provide unique fundamental insights into the above-mentioned academic research fields, and will foster the development of advanced information filtering technologies of interest to information technology practitioners.

link prediction | complex networks | structural perturbation | predictability

Understanding the organization of real networks is a long-standing challenge in many branches of science (1). Although some mechanisms have already been accepted as primary driving forces in network organization, including homophily (2–4), triadic closure (5–7), preferential attachment (8–10), reciprocity (11), and social balance (12), one or two of these mechanisms cannot provide a complete explanation; i.e., link formation in real-world networks is usually driven by both regular and irregular factors, and only the former can be explained using mechanistic models. This intrinsic network complexity presents us with the question of how to estimate what portions of a real network can be categorized as regular, in other words, to what extent the link formation in network is explicable.

This question brings to mind the link prediction problem in which the set of observed links in a network is used to estimate the likelihood that a nonobserved link exists (13). The extent to which the network formation is explicable coincides with our capacity to predict missing links (14, 15). On the one hand, an effective link prediction algorithm provides strong evidence of the corresponding mechanism(s) of network organization, e.g., effectiveness of common-neighborhood-based methods indicates the significance of triadic closure (16, 17). On the other hand, a better understanding of network organization should be transferable to a good link prediction algorithm, e.g., the prior assumption of hierarchical organization of networks can be directly applied to the design of a prediction algorithm (18). In this sense, the precision of a link prediction algorithm tells us the extent to which the link formation in network can be explained by this algorithm. However, different algorithms provide different precisions in same network (see Table 1, the precisions of seven link prediction (LP) methods on 10 networks) and thus the precision only reflects the link predictability associated with a specific algorithm, not the intrinsic feature of the network itself.

Predictability is usually defined as the possible maximum precision of a prediction algorithm (19). However, this kind of definition is not suitable for link prediction since a real network's link predictability under such definition should be 1 because their nonobserved links are almost always distinguishable (see *Materials and Methods*). In this paper, link predictability indeed characterizes the inherent difficulty of prediction that does not depend on specific algorithms, and our fundamental hypothesis is that missing links are difficult to predict if their addition causes huge structural changes, and thus network is highly predictable if the removal or addition of a set of randomly selected links does not significantly change the network's structural features. Accordingly, we propose a so-called "structural consistency" index that is based on the first-order matrix perturbation, which can reflect the inherent link predictability of a network and does not require any prior knowledge of the network's organization. We also propose a structural perturbation method for link prediction that is more accurate and robust than the state-of-the-art methods.

Structural Consistency

Consider a simple undirected network $G(V,E)$ where V is the set of nodes and E is the set of links. The given network can be represented by an $N\times N(N=|V|)$ adjacency matrix A, where the element $A_{ij}=1$ if nodes i and j are connected and $A_{ij}=0$ otherwise. We randomly select a fraction p^H of the links to constitute a perturbation set ΔE, while the rest of the links $E-\Delta E$ constitute the set E^R. Denote by A^R and ΔA the corresponding adjacency matrices; obviously, $A=A^R+\Delta A$. Since A^R is real symmetric, it can be diagonalized as

Significance

Quantifying a network's link predictability allows us to (*i*) evaluate predictive algorithms associated with the network, (*ii*) estimate the extent to which the organization of the network is explicable, and (*iii*) monitor sudden mechanistic changes during the network's evolution. The hypothesis of this paper is that a group of links is predictable if removing them has only a small effect on the network's structural features. We introduce a quantitative index for measuring link predictability and an algorithm that outperforms state-of-the-art link prediction methods in both accuracy and universality. This study provides fundamental insights into important scientific problems and will aid in the development of information filtering technologies.

Author contributions: L.L., L.P., T.Z., Y.-C.Z., and H.E.S. designed research; L.L., L.P., Y.-C.Z., and H.E.S. performed research; L.L., L.P., T.Z., and Y.-C.Z. analyzed data; and L.L., T.Z., Y.-C.Z., and H.E.S. wrote the paper.

Reviewers: G.P., University of Rome; and D.W., IBM TJ Watson Research Center.

The authors declare no conflict of interest.

Freely available online through the PNAS open access option.

[1]L.L., L.P., and T.Z. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: hes@bu.edu, linyuan.lv@hznu.edu.cn, or yi-cheng.zhang@unifr.ch.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1424644112/-/DCSupplemental.

$$A^R = \sum_{k=1}^{N} \lambda_k x_k x_k^T, \quad [1]$$

where λ_k and x_k are the eigenvalue and the corresponding orthogonal and normalized eigenvector for A^R, respectively.

We consider the set ΔE as a perturbation to the network A^R and construct the perturbed matrix via first-order approximation that allows the eigenvalues to change but fixes the eigenvectors. We first consider the nondegenerated case without any repeated eigenvalues (see *SI Appendix, Case of Degenerate Eigenvalues*, for the case with degenerate eigenvalues). After perturbation, the eigenvalue λ_k is corrected to be $\lambda_k + \Delta\lambda_k$ and its corresponding eigenvector is corrected to be $x_k + \Delta x_k$. Left-multiplying the eigenfunction

$$(A^R + \Delta A)(x_k + \Delta x_k) = (\lambda_k + \Delta\lambda_k)(x_k + \Delta x_k) \quad [2]$$

by x_k^T and neglecting second-order terms $x_k^T \Delta A \Delta x_k$ and $\Delta\lambda_k x_k^T \Delta x_k$, we obtain

$$\Delta\lambda_k \approx \frac{x_k^T \Delta A x_k}{x_k^T x_k}. \quad [3]$$

This formula is reminiscent of the expectation value of the first-order perturbation Hamiltonian in quantum mechanics. Using the perturbed eigenvalues while keeping eigenvectors unchanged, the perturbed matrix can be obtained,

$$\tilde{A} = \sum_{k=1}^{N} (\lambda_k + \Delta\lambda_k) x_k x_k^T, \quad [4]$$

which can be considered as the linear approximation of the given network A if the expansion is based on A^R.

The eigenvectors can well reflect network structural features (20). If the perturbation does not significantly change the structural features, the eigenvectors of the observed matrix A^R (i.e., x_k) and those of the matrix $A^R + \Delta A$ (i.e., $x_k + \Delta x_k$) should be almost the same. If so, according to Eq. **4**, $\tilde{A}$ should be very close to $A^R + \Delta A$. Therefore, given a network A, we first randomly remove a group of randomly selected links ΔE, and then we perturb the remaining part A^R by ΔE to obtain the perturbed matrix $\tilde{A}$ via Eq. **4**. If the network is highly regular, the random removal ΔE will not sharply change the structure features, and thus A and $\tilde{A}$ should be close to each other. To measure this quantitatively, we rank all of the links in set $U - E^R$ in descending order according to their values in $\tilde{A}$, where U is the universal set of links. We denote E^L the set of top-L ranked links, where $L = |\Delta E|$, namely, the number of links in the perturbation set. Then the links in E^R together with the links in E^L construct the perturbed network, which is usually different from $E^R + \Delta E$. The structural consistency σ_c is defined as the fraction of common links between ΔE and E^L, as

$$\sigma_c = \frac{|E^L \cap \Delta E|}{|\Delta E|}. \quad [5]$$

Fig. 1 shows how to calculate the structural consistency of a simple network, with a summary of detailed procedure presented in *SI Appendix, Six Steps to Calculate* σ_c.

Structural Perturbation Method

The perturbation method used to determine the structural consistency can be applied to predict missing links. Link prediction aims at estimating the existence likelihood of nonobserved links based on the observed topology (13). The simplest framework of link prediction is similarity-based algorithms (16) in which each pair of nodes, x and y, is assigned a similarity score s_{xy}. All nonobserved links are ranked according to their scores, with an assumption that links with higher scores have higher existence likelihoods (see mathematical description of LP problem as well as the accuracy metrics in *SI Appendix, Link Prediction Problem*). Under this framework, the entries of $\tilde{A}$ can be considered as the similarity scores assigned to links. For example, in Fig. 1, if we want to predict one missing link of given network A by using the structural perturbation method (SPM), we will rank all of the nonobserved links (i.e., the links corresponding to 0 in matrix A) according to their scores in $\tilde{A}$; then the top one is the link (3,8). The feasibility of SPM is based on the strong correlation between independent perturbations (see *SI Appendix*, Table S1), which indicates that the missing links, which are considered as unknown information, can be recovered by perturbing the network with another set of known links (i.e., ΔE).

Consider an undirected network $G(V,E)$: To test the algorithm's accuracy, the set of links, E, is randomly divided into two parts: (i) a training set E^T, which is treated as known information, and (ii) a probe set (i.e., validation subset) E^P, which is used for testing and can be considered as missing links. No information in the probe set is allowed to be used for prediction. Obviously, $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$. Our task is to uncover the links in the probe set based on the information in the training set.

Notice that, in this task, the training set E^T plays a similar role to the observed network A, and to obtain the perturbed matrix $\tilde{A}$, we randomly select a fraction p^H of links from E^T as perturbation set ΔE. Then, by perturbing $E^T - \Delta E$ with ΔE, we obtain $\tilde{A}$ through Eq. **4**. The final average prediction matrix $\langle\tilde{A}\rangle$ is obtained by averaging over 10 independent selections of ΔE. By ranking all of the nonobserved links (i.e., links in $U - E^T$) in

Table 1. Link prediction accuracy measured by precision on the 10 real networks

Precision	Jazz	Metabolic	Neural	USAir	Food web	Hamster	NetSci	Yeast	Email	Router
SPM	**0.677**	**0.354**	**0.168**	0.451	**0.561**	**0.469**	0.334	**0.166**	**0.158**	**0.357**
CN	0.506	0.137	0.095	0.374	0.073	0.061	0.329	0.109	0.149	0.027
AA	0.525	0.190	0.105	0.394	0.075	0.061	0.334	0.121	0.150	0.026
RA	0.541	0.267	0.104	**0.455**	0.076	0.054	**0.541**	0.090	0.148	0.027
Katz	0.546	0.147	0.107	0.379	0.181	0.108	0.370	0.061	0.149	0.120
HSM	0.326	0.100	0.073	0.216	0.249	0.202	0.303	0.081	0.134	0.309
SBM	0.410	0.197	0.143	0.335	0.460	0.275	0.177	0.122	0.094	0.176

We compare our method, SPM, to six well-known methods presented in *Materials and Methods*. For each real network, 10% of its links will be randomly selected to constitute the probe set, and the rest of the links constitute the training set. Prediction accuracy is measured by precision. We set $p^H = 0.1$ for SPM. For the parameter-dependent Katz index, the present results correspond to the optimal parameter subject to the highest precision. The highest value for each network is in boldface.

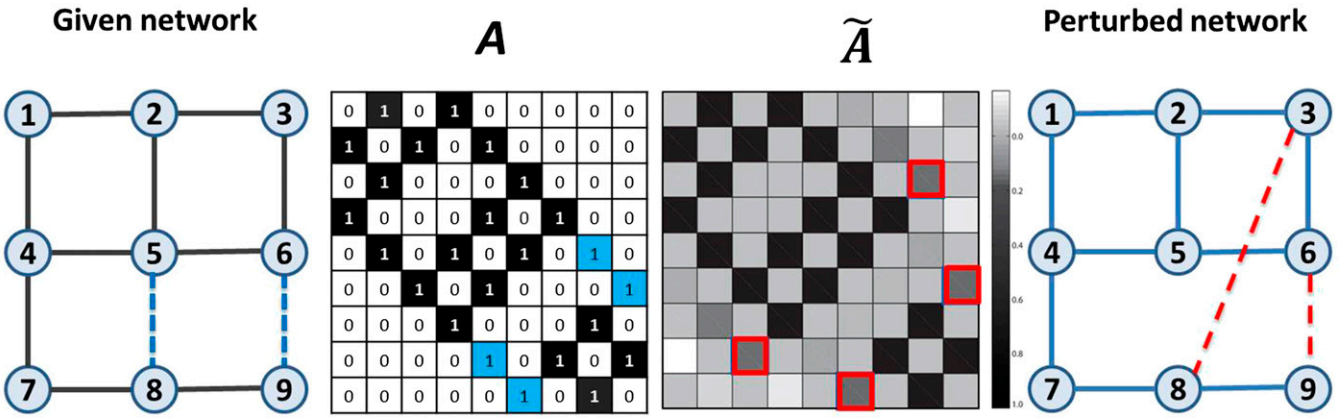

Fig. 1. An illustration of how to calculate the structural consistency. In the first plot, the blue dashed links constitute the perturbation set $\Delta E = \{(5,8),(6,9)\}$ (corresponding to ΔA), while the solid links constitute the set E^R (corresponding to A^R). The second plot shows the adjacency matrix A of the given network, where the number in each square is the corresponding value of the matrix element. The black and blue squares represent the links in E^R and ΔE, respectively. To calculate the consistency, we perturb A^R with ΔA. The perturbed matrix $\tilde{A}$ is shown in the third plot, from which we derive the perturbed network in the fourth plot, where the red dashed lines are outcome links selected by ranking all links in $U - E^R$ in descending order according to their corresponding values in $\tilde{A}$. Since there are two links in ΔE, then $L = 2$, and the set $E^L = \{(3,8),(6,9)\}$. In this case, only one of the two blue links is recovered by perturbation; then we have $\sigma_c = 0.5$.

decreasing order according to their scores given by $\langle \tilde{A} \rangle$, we select the top-$|E^P|$ links and see how many of them are in the probe set. This ratio is called "precision," which is used to quantify the performance of the algorithm (21). A summary of detailed procedures can be found in *SI Appendix, Five Steps to Calculate Prediction Accuracy of SPM*.

We compare the structural perturbation method with six widely applied link prediction algorithms, including four similarity-based indices: the common neighbors (CN) index (16), the Adamic-Adar (AA) index (22), the resource allocation (RA) index (17, 23), and the Katz index (24). We also use two likelihood methods: the hierarchical structure model (HSM) (18) and the stochastic block model (SBM) (25). See *Materials and Methods* for the six baseline algorithms. Table 1 shows the prediction accuracy of the 10 real-world networks (see *Materials and Methods* and *SI Appendix*, Table S2, for the description and basic statistics of the data), measured by precision [see *SI Appendix*, Table S3 for the results measured by another metric called AUC: the area under the receiver operating characteristic curve (26); see the definition in *SI Appendix, Link Prediction Problem*, Eq. 5]. The highest value for each network (in each column) is in boldface. Overall, SPM outperforms all other baseline algorithms including such state-of-the-art methods as the RA index, HSM, and SBM. In addition, SPM is the most robust method for disparate networks; i.e., although, in a few cases, its performance is not the best, it is always very good. In contrast, all six baseline algorithms give very poor predictions for some networks. In addition to the effectiveness of SPM, we can efficiently obtain an approximate result by sampling large-scale networks (see discussion in *SI Appendix, Applying to Large Networks*).

Notice that the random division of E^T and E^P is relevant to the prediction of missing parts of networks, such as protein–protein interaction networks where the known interactions are even fewer than unknown interactions (27). In addition to the prediction of missing links in static networks, LP algorithms can also predict future links in evolving networks, such as friendship recommendations in online social networks. In such issues, to evaluate the algorithmic performance, observed links should be divided according to their birth times: Elder (90%) and younger (10%) links constitute E^T and E^P, respectively. We have also tested LP algorithms in three real evolving networks (see *Materials and Methods* and *SI Appendix*, Table S2); as shown in Table 2, SPM still performs the best.

Table 2. The precision of link prediction on three real-world temporal networks

Networks	CN	AA	RA	Katz	HSM	SPM
Arxiv	0.021	0.022	0.026	0.033	0.020	**0.085**
Facebook	0.021	0.024	0.041	0.022	0.007	**0.051**
Enron	0.032	**0.033**	0.027	**0.033**	0.008	**0.033**

Each network is of size $N = 4{,}000$, that sampled from the original networks by using the random-walk method (see *SI Appendix*). The best-performed entries are emphasized in bold. We set $p^H = 0.1$ for SPM and for the parameter-dependent Katz index; the present results are obtained under the optimal parameter subject to the highest precision. The results of SBM are not included due to the high computational complexity.

Link Predictability

We first consider the structural consistency of modeled networks and show the validity of σ_c as an index for link predictability. In the Erdös–Rényi (ER) network (28), each pair of nodes is connected with probability p. If p is finite and the network size N goes to infinity, the spectral density adjacency matrices in ER networks obey the Wigner semicircle law and the eigenvectors are distributed isotropically at random (29, 30). The first-order perturbation of the eigenvalues is thus also random, leading to low structural consistency values. Given an ER network $G(N,p)$, we randomly select a fraction $p^H = 0.1$ of the links (we have tested that σ_c is not sensitive to the specific value of p^H; see *SI Appendix*, Fig. S3), and determine the average structural consistency $\langle \sigma_c \rangle$ as a function of N for different p. Fig. 2*A* shows how the structural consistency decreases with the network size in a power-law-like relationship and tends to the random chance $p \cdot p^H / (1 - p + p \cdot p^H)$ in the thermodynamical limit, supporting the intuition that fully random networks are unpredictable, which is also in accordance with the previous

Table 3. Pearson correlation coefficients (CC) between precision and structural consistency on the 10 real networks

	CN	AA	RA	Katz	HSM	SBM	SPM
CC	0.493	0.476	0.495	0.698	0.870	0.819	0.938

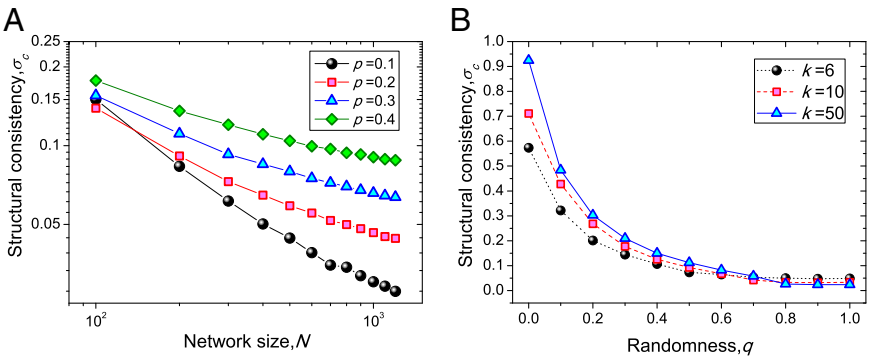

Fig. 2. Structural consistency of modeled networks: (*A*) ER networks with different sizes N and connecting probabilities p; (*B*) WS networks with N = 1,000, and different average degrees k and rewiring probabilities q. Each data point is averaged over 100 independent runs.

report about the very low link prediction accuracy on ER networks (31).

We next consider the Watts–Strogatz (WS) networks (32) with controllable randomness. A WS network $G(N,k,q)$ starts from a ring of N nodes where each node connects to its k nearest neighbors. With a probability q, each link is replaced by another link that joins two randomly chosen nodes. When $q=0$, the network is a deterministic ring, and when $q=1$, it is fully random. Fig. 2*B* shows how σ_c decreases with the rewiring probability q, indicating once again that higher irregularities (i.e., randomness) will result in lower predictability.

The above experimental results on modeled networks affirm the rationale behind the proposed index σ_c. We next turn our attention to real-world networks, whose predictabilities cannot be controlled as WS networks. We thus compare the structural consistency with the prediction accuracy from representative link prediction algorithms. Table 3 shows how the prediction precision is positively correlated with structural consistency σ_c for all six baseline algorithms, indicating that σ_c can, to some extent, reflect the link predictability of real networks. In addition, the precision values of algorithms that account for the global organization principles are approximately linearly correlated with the structural consistency (as indicated by >0.8 Pearson correlation coefficients in Table 3; see also Fig. 3), suggesting that the structural consistency is indeed a good indicator of whether the network is organized by some perceptible regulations. The results also show us that the missing links in networks with higher structural consistencies are easier to dig out using link prediction algorithms.

Discussion

Throughout history, human beings, from ancient prophets to modern scientists, have attempted to make predictions. The recent development of theoretical tools and the expanding availability of massive databases have allowed scientists to predict behaviors and trends, chart emergent events, and locate missing elements of a system (33–35). In this paper, we treat predictability as an inherent measurement of the regularities in the organization of a networked system, and our hypothesis is that a missing part is predictable only when it does not significantly change the structural features of the observable part. If it does, it cannot be revealed through observation. The perspective of this hypothesis is that high predictability indicates some perceptible regulative principles in the organization of the network. Putting aside any a priori hypotheses of what this regulative principle might be, we directly measure the structural consistency of a network by perturbing its adjacency matrix and observing the change of eigenvalues provided the fixed eigenvectors, similar to the well-known first-order perturbation in quantum mechanics. Although we cannot determine the maximum precision of any given link prediction algorithm, the structural consistency σ_c is a good indicator of the inherent predictability of both modeled networks and real-world networks. Surprisingly, by directly applying the first-order matrix perturbation method, we achieve more-accurate link predictions than some gracefully designed methods such as HSM (18) and SBM (25). In particular, the performance of SPM for disparate networks is very robust; i.e., it is either the best or very close to the best. In contrast, other algorithms can largely fail.

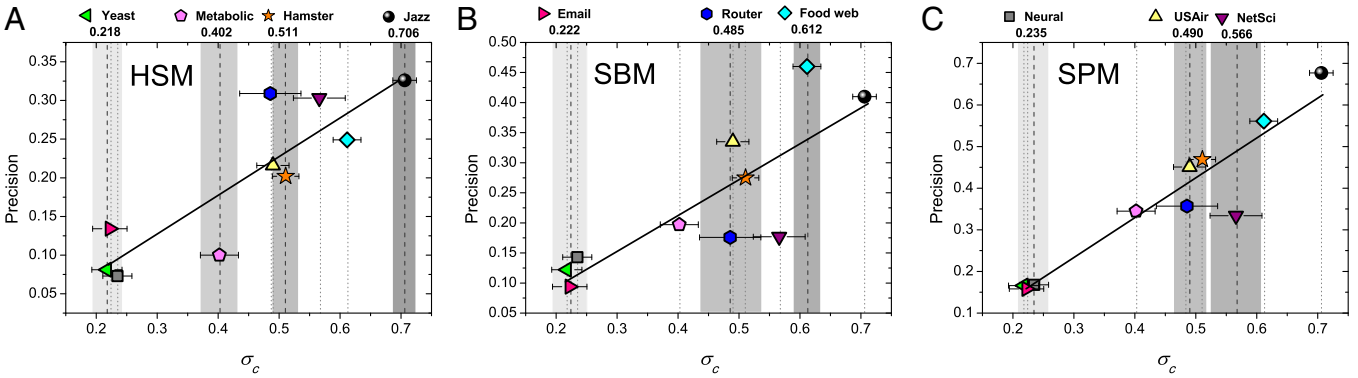

Fig. 3. Scatter plot between structural consistency and precision of SPM, SBM, and HSM. The numbers on the top of the panels are the structural consistencies of the corresponding networks obtained at $p^H=0.1$, averaged over 10 independent runs. The error bars represent the SDs of σ_c. The shadows in the background emphasize the structural consistencies of the corresponding networks. The higher σ_c is, the darker the corresponding shadow region is. The solid lines are the linear fittings with slopes being equal to 0.957, 0.598, and 0.495 for (*C*) SPM, (*B*) SBM, and (*A*) HSM, respectively.

For example, CN, AA, RA, and Katz indices cannot adequately manage the food web, and the HSM poorly predicts missing links in a metabolic network, which is not organized in a hierarchical way. The SPM, on the other hand, does not make any a priori assumptions about any specific organizing principles of a network, and thus its predictions are consistently more robust.

Potential applications of this work are wide and can take both theoretical and practical forms. Using structural consistency values, we can determine whether a poor prediction was caused by an inappropriate algorithm or was due to the intrinsic unpredictability of the network, and then estimate how large a space is needed to improve the algorithm. For example, the CN index does not perform well for either neural or food webs. Because the structural consistency of a food web is much larger than that of a neural web, we can infer that CN is not appropriate for a food web, while the low precision of a neural web may result from its own low predictability. Indeed, as shown in Fig. 3, the networks largely below the fitting line are those where the corresponding algorithm is not suitable to be applied. For an evolving network, the structural consistency can give a temporal estimation of whether the network becomes more elusive or not, as well as monitor the sudden changes in the evolving mechanisms (see *SI Appendix, Monitor the Sudden Changes of Evolving Networks with σ_c*, for numerical experiments). In addition, the structural perturbation method, as a straightforward extension of structural consistency, can be directly applied to determining the missing links in real-world networks. This work should be of interest to academic researchers in a variety of fields, to information technology practitioners, and to business practitioners.

Materials and Methods

Maximum Precision of Link Prediction. If we define link predictability as the maximum precision of any link prediction algorithm, then a network is of nearly zero predictability if all nonobserved links are completely the same (e.g., a star network). For example, a vertex-transitive (20) network is of zero predictability since all of the nodes in the observed structure are identical and thus missing links are also indistinguishable from nonexistent links. For a vertex-transitive network, given any of its two nodes u and v, there is some automorphism f such that $f(u)=v$. This extremely rigid definition from automorphism-based symmetry makes virtually all real-world networks have a predictability very close to 1, since the missing links can be distinguished from nonexistent links. Because it is approximately free of graph automorphisms, link predictability approaches 1 even in ER networks (36). Thus, this rigid approach does not allow us to obtain any a useful estimation of link predictability.

Data Description. We consider the following 10 real-world networks drawn from disparate fields: (*i*) Jazz (37), a collaboration network of jazz musicians consists of 198 nodes and 2742 interactions; (*ii*) Metabolic (38), the metabolic network of *Caenorhabditis elegans*; (*iii*) Neural (32), the neural network of *C. elegans* (the original network is directed and weighted; here we treat it as a simple network by ignoring the directions and weights); (*iv*) US Air (39), the US Air transportation network; (*v*) Food web (40), the food web in Florida Bay during wet season; (*vi*) Hamster (41), a friendship network of users on the website hamsterster.com; (*vii*) NetSci (42), a coauthorship network of scientists working on network theory and experiment; (*viii*) Yeast (43), a protein–protein interaction network in budding yeast; (*ix*) Email (44), a network of email interchanges between members of the University Rovira I Virgili; (*x*) Router (45), a symmetrized snapshot of the structure of the Internet at the level of autonomous systems; (*xi*) Arxiv (46), a scientific collaboration network from the arXiv's High Energy Physics C Theory (hep-th) section; (*xii*) Facebook (47), a network of a small group of Facebook users; and (*xiii*) Enron (48), an email communication network from employees of Enron between 1999 and 2003. The more detailed information and statistical features of these networks can be found in *SI Appendix, Statistical Features of Experimental Networks*.

Baseline Algorithms for Link Prediction. The link prediction problem has been a long-standing challenge in modern information science (13, 49). Its main goal is to estimate the existence likelihood of nonobserved links based on the known topology and node attributes. Link prediction has already found wide applications in interdisciplinary fields, including uncovering missing parts of social and biological networks (50–52) and recommending friends and products in online social networks and e-commerce web sites (53–55).

For comparison, we introduce four benchmark similarity indices (13). The simplest is the CN index (16) in which two nodes, x and y, have a higher connecting probability if they have more common neighbors. Two improved indices based on CN are the AA index (22) and the RA index (17, 23), both of which assign less-connected neighbors more weight. The mathematical expressions are

$$s_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)|, \quad [6]$$

$$s_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|}, \quad [7]$$

$$s_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|}, \quad [8]$$

where $\Gamma(x)$ denotes the set of neighbors of node x.

Unlike the above three local similarity indices, the Katz index (24) uses global topological information by summing over the collection of paths with exponential damping according to path lengths, i.e.,

$$s_{xy}^{Katz} = \alpha A_{xy} + \alpha^2 A_{xy}^2 + \alpha^3 A_{xy}^3 + \cdots, \quad [9]$$

which can be rewritten in the compact form, when $|\alpha| < 1/\lambda_{max}$, as

$$S = (I - \alpha A)^{-1} - I, \quad [10]$$

where I is the identity matrix, A is the adjacency matrix, and λ_{max} is the largest eigenvalue of A. In our experiments, we tune the parameter α to optimize the performance of the Katz index. Notice that, since α cannot be exactly zero, the Katz index cannot degenerate to the CN index. Even when α is very close to zero, the performance of the Katz index can be different from the CN index, because, under the CN index, many nonobserved links are scored the same and thus ranked in a random way (see analysis on this so-called degeneracy phenomenon in refs. 17 and 31); therefore the very slight differences contributed by the latter items in Eq. **9** may result in considerable changes in the order of nonobserved links associated with the same number of common neighbors.

We also consider two probability methods. The HSM (18) method assumes that many real-world networks are hierarchically organized and thus nodes can be divided into groups and further divided into subgroups. The SBM (25) approach is one of the most general network models. Nodes are partitioned into groups and the connecting probability of any two nodes is only determined by the groups they belong to.

ACKNOWLEDGMENTS. We thank G. D'Agostino for helpful discussion. This work was partially supported by the National Natural Science Foundation of China (Grants 11222543, 11075031, 11205042, and 61433014), NESS Project, and CCF-Tencent Open Research Fund. L.L. acknowledges the research start-up fund of Hangzhou Normal University under Grant PE13002004039 and the EU FP7 Grant 611272 (project GROWTHCOM). T.Z. acknowledges the Program for New Century Excellent Talents in University under Grant NCET-11-0070. The work at Boston University was supported by US National Science Foundation Grants 1125290 and 0855453.

1. Barabási A-L (2009) Scale-free networks: A decade and beyond. *Science* 325(5939): 412–413.
2. McPherson M, Smith-Lovin L, Cook JM (2001) Birds of a feather: Homophily in social networks. *Annu Rev Sociol* 27:415–444.
3. Currarini S, Jackson MO, Pin P (2010) Identifying the roles of race-based choice and chance in high school friendship network formation. *Proc Natl Acad Sci USA* 107(11): 4857–4861.
4. Lewis K, Gonzalez M, Kaufman J (2012) Social selection and peer influence in an online social network. *Proc Natl Acad Sci USA* 109(1):68–72.
5. Szabo G, Alava M, Kertész J (2004) Clustering in complex networks. *Complex Networks*, Lecture Notes in Physics, eds Ben-Naim E, Frauenfelder H, Toroczkai Z (Springer, New York), Vol 650, pp 139–162.
6. Kossinets G, Watts DJ (2006) Empirical analysis of an evolving social network. *Science* 311(5757):88–90.

7. Yin D, Hong L, Xiong X, Davison BD (2011) Link formation analysis in microblog. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM Press, New York), pp 1234–1236.
8. Barabási A-L, Albert R (1999) Emergence of scaling in random networks. *Science* 286(5439):509–512.
9. Jeong H, Néda Z, Barabási A-L (2003) Measuring preferential attachment in evolving networks. *Europhys Lett* 61(4):567.
10. Leskovec J, Backstrom L, Kumar R, Tomkins A (2008) Microscopic evolution of social networks. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM Press, New York), pp 462–470.
11. Garlaschelli D, Loffredo MI (2004) Patterns of link reciprocity in directed networks. *Phys Rev Lett* 93(26 Pt 1):268701.
12. Marvel SA, Strogatz SH, Kleinberg JM (2009) Energy landscape of social balance. *Phys Rev Lett* 103(19):198701.
13. Lü L, Zhou T (2011) Link prediction in complex networks: A survey. *Physica A* 390(6): 1150–1170.
14. Wang WQ, Zhang QM, Zhou T (2012) Evaluating network models: A likelihood analysis. *Europhys Lett* 98(2):28004.
15. Zhang QM, Lü L, Wang WQ, Zhu YX, Zhou T (2013) Potential theory for directed networks. *PLoS ONE* 8(2):e55437.
16. Liben-Nowell D, Kleinberg J (2007) The link prediction problem for social networks. *J Am Soc Inf Sci Technol* 58(7):1019–1031.
17. Zhou T, Lü L, Zhang YC (2009) Predicting missing links via local information. *Eur Phys J B* 71(4):623–630.
18. Clauset A, Moore C, Newman MEJ (2008) Hierarchical structure and the prediction of missing links in networks. *Nature* 453(7191):98–101.
19. Song C, Qu Z, Blumm N, Barabási A-L (2010) Limits of predictability in human mobility. *Science* 327(5968):1018–1021.
20. Godsil C, Royle G (2001) *Algebraic Graph Theory* (Springer, New York).
21. Herlocker JL, Konstann JA, Terveen K, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22(1):5–53.
22. Adamic LA, Adar E (2003) Friends and neighbors on the web. *Soc Networks* 25(3): 211–230.
23. Ou Q, Jin YD, Zhou T, Wang BH, Yin BQ (2007) Power-law strength-degree correlation from resource-allocation dynamics on weighted networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 75(2 Pt 1):021102.
24. Katz L (1953) A new status index derived from sociometric analysis. *Psychometrika* 18(1):39–43.
25. Guimerà R, Sales-Pardo M (2009) Missing and spurious interactions and the reconstruction of complex networks. *Proc Natl Acad Sci USA* 106(52):22073–22078.
26. Hanley JA, McNeil BJ (1983) A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 148(3):839–843.
27. Amaral LAN (2008) A truer measure of our ignorance. *Proc Natl Acad Sci USA* 105(19): 6795–6796.
28. Erdös P, Rényi A (1959) On random graphs. *Pub Math Debrecen* 6:290–297.
29. Chung F, Lu L, Vu V (2003) Spectra of random graphs with given expected degrees. *Proc Natl Acad Sci USA* 100(11):6313–6318.
30. Farkas IJ, Derényi I, Barabási A-L, Vicsek T (2001) Spectra of "real-world" graphs: Beyond the semicircle law. *Phys Rev E Stat Nonlin Soft Matter Phys* 64(2 Pt 2):026704.
31. Lü L, Jin CH, Zhou T (2009) Similarity index based on local paths for link prediction of complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys* 80(4 Pt 2):046122.
32. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393(6684):440–442.
33. Ginsberg J, et al. (2009) Detecting influenza epidemics using search engine query data. *Nature* 457(7232):1012–1014.
34. Vespignani A (2009) Predicting the behavior of techno-social systems. *Science* 325(5939):425–428.
35. Goel S, Hofman JM, Lahaie S, Pennock DM, Watts DJ (2010) Predicting consumer behavior with Web search. *Proc Natl Acad Sci USA* 107(41):17486–17490.
36. Bollobás B (2001) *Random Graphs* (Cambridge Univ Press, Cambridge, UK), 2nd Ed.
37. Gleiser P, Danon L (2003) Community structure in Jazz. *Adv Complex Syst* 6(4): 565–573.
38. Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. *Phys Rev E Stat Nonlin Soft Matter Phys* 72(2 Pt 2):027104.
39. Batagelj V, Mrvar A (2006) Pajek datasets. Available at vlado.fmf.uni-lj.si/pub/networks/data/. Accessed May 19, 2013.
40. Ulanowicz RE, Bondavalli C, Egnotovich MS (1998) *Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 97: The Florida Bay Ecosystem* (Chesapeake Biol Lab, Solomons, MD), Tech Rep CBL 98-123.
41. Kunegis J (2013) Hamsterster friendships unique network dataset – KONECT. Available at konect.uni-koblenz.de/networks/petster-friendships-hamster. Accessed June 1, 2013.
42. Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys Rev E Stat Nonlin Soft Matter Phys* 74(3 Pt 2):036104.
43. Bu D, et al. (2003) Topological structure analysis of the protein–protein interaction network in budding yeast. *Nucleic Acids Res* 31(9):2443–2450.
44. Guimerà R, Danon L, Díaz-Guilera A, Giralt F, Arenas A (2003) Self-similar community structure in a network of human interactions. *Phys Rev E Stat Nonlin Soft Matter Phys* 68(6 Pt 2):065103.
45. Spring N, Mahajan R, Wetherall D, Anderson T (2004) Measuring ISP topologies with Rocketfuel. *IEEE/ACM Trans Networking* 12(1):2–16.
46. Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. *ACM Trans Knowledge Discovery Data* 1(1):2.
47. Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in Facebook. *Proceedings of the 2nd Workshop on Online Social Networks* (ACM Press, New York), pp 37–42.
48. Klimt B, Yang Y (2004) The Enron corpus: A new dataset for email classification research. *Proceedings of the European Conference on Machine Learning* (Springer, New York), pp 217–226.
49. Getoor L, Diehl CP (2005) Link mining: A survey. *ACM SIGKDD Explor Newsl* 7(2):3–12.
50. Mamitsuka H (2012) Mining from protein–protein interactions, *Wiley Interdiscip Rev: Data Min Knowl Discov* 2(5):400–410.
51. Cannistraci CV, Alanis-Lobato G, Ravasi T (2013) From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Sci Rep* 3:1613.
52. Barzel B, Barabási A-L (2013) Network link prediction by global silencing of indirect correlations. *Nat Biotechnol* 31(8):720–725.
53. Schifanella R, Barrat A, Cattuto C, Markines B, Menczer F (2010) Folks in folksonomies: Social link prediction from shared metadata. *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining* (ACM Press, New York), pp 271–280.
54. Aiello LM, et al. (2012) Friendship prediction and homophily in social media. *ACM Trans Web* 6(2):9.
55. Lü L, et al. (2012) Recommender systems. *Phys Rep* 519(1):1–49.

Supporting Information (SI) for "Toward link predictability of complex networks"

by Linyuan Lü, Liming Pan, Tao Zhou, Yi-Cheng Zhang, and H. Eugene Stanley

I. CASE OF DEGENERATE EIGENVALUES

If the adjacency matrix contains degenerate eigenvalues, we must modify the approach using non-degenerate eigenvalues. We denote the eigenvalues as λ_{ki}, where the index k runs over different eigenvalues and the index i runs over M associated eigenvectors of the same eigenvalue. Note that there is no unique way of choosing a basis for the eigenvectors of the unperturbed network since any linear combination of the eigenvectors belonging to the same eigenvalue is still an eigenvector. Repeated eigenvalues have been shown to be related to the symmetric motifs and graph automorphisms in networks [1]. After a perturbation is added to the network, the symmetry of the nodes will be lifted either partly or completely, so the degenerate eigenvalues must be chosen such that they can be transformed continuously into the perturbed non-degenerate eigenvalues. If we define the chosen eigenvectors to be $\bar{x}_{ki} = \sum_{j=1}^{M} \beta_{kj} x_{kj}$, the eigenfunction becomes

$$\left(A^R + \Delta A\right) \bar{x}_{ki} = \left(\lambda_{ki} + \Delta\bar{\lambda}_{ki}\right) \bar{x}_{ki}, \tag{1}$$

giving us

$$\Delta\bar{\lambda}_{ki} \sum_{j=1}^{M} \beta_{kj} x_{kj} = \sum_{j=1}^{M} \beta_{kj} \Delta A x_{kj}. \tag{2}$$

For any $n = 1 \cdots M$, left multiplying Eq. (2) by x_{kn}^T, we obtain

$$\Delta\bar{\lambda}_{ki} \beta_{kn} = \sum_{j=1}^{M} \beta_{kj} x_{kn}^T \Delta A x_{kj}. \tag{3}$$

Written in matrix form, Eq. (3) becomes

$$W B_k = \Delta\bar{\lambda}_k B_k, \tag{4}$$

where W is an $M \times M$ matrix and is defined by $W_{nj} = x_{kn}^T \Delta A x_{kj}$ and B_k is the column vector of β_{kj}. After obtaining $\Delta\bar{\lambda}_k$ and B_k from the eigenfunction (4), the corrected eigenvectors as well as the first-order corrections to the corresponding eigenvalues can be determined simultaneously. Then the structural consistency can be calculated in the same way as in the non-degenerate eigenvalues case. Specifically, to obtain the perturbed adjacency matrix $\tilde{A}$ we simply replace x_k and $\Delta\lambda_k$ in Eq. 4 with $\bar{x}_k$ and $\Delta\bar{\lambda}_k$, respectively.

II. SIX STEPS TO CALCULATE σ_c

Given a network G, we calculate the structural consistency σ_c via the following procedure:

Step 1: Given network A, we randomly select a fraction of links to constitute a perturbation set $\Delta E(\Delta A)$, while the rest is $E^R(A^R)$, obviously $A = A^R + \Delta A$.

Step 2: We calculate the eigenvalues λ_k and their corresponding eigenvectors (x_k) of A^R.

Step 3: We use equation (3) to calculate $\Delta\lambda$.

Step 4: We use equation (4) to calculate the perturbed matrix $\tilde{A}$.

Step 5: We rank the non-observed links according to their scores given by $\tilde{A}$. For a non-observed link (i, j), its score is the value of $\tilde{A}_{ij}$.

Step 6: We select the top-L non-observed links. Here L is the number of links in set ΔE. And we see how many of them are also in the perturbation set ΔE. This ratio is σ_c.

For example, a true network with 20 nodes has 100 links, and we select 10 links as ΔE, then there are $20\times19/2-90 = 100$ non-observed links. By using our method, we find among these 10 links in ΔE, there are 6 links that are ranked within the top-10 places according to their scores in $\tilde{A}$. Then $\sigma_c = 6/10 = 0.6$.

III. LINK PREDICTION PROBLEM

The purpose of link prediction (LP) is to estimate the existence likelihood of all non-observable links based on known network topology and node attributes (assuming this information is available). We consider an undirected network $G(V, E)$ in which V is the set of nodes and E is the set of links. Multiple links and self connections are not allowed. Denote by U the universal set containing all $|V|(|V|-1)/2$ possible links, where $|V|$ denotes the number of elements in set V. Then, the set of non-existent links is $U - E$. We assume that missing links exist (or will exist in the future) in the set $U - E$, and the task of link prediction is to locate them.

Because we do not know which links in a system are missing or will appear in the future, we test the algorithm's accuracy by randomly dividing the observed links E (in the original network) into two sets, (i) a training set E^T made up of known information, and (ii) a probe set E^P used for testing and from which no information is allowed for use in prediction. Clearly, $E^T \bigcup E^P = E$ and $E^T \bigcap E^P = \emptyset$. In principle, a link prediction algorithm provides an ordered list of all non-observed links (i.e., $U - E^T$) or equivalently gives each non-observed link, say $(x, y) \in U - E^T$, a score s_{xy} to quantify its existence likelihood. Two standard metrics are used to quantify the accuracy of prediction algorithms: *the area under the receiver operating characteristic curve* (AUC) [2] and *the precision* [3]. The AUC evaluates the algorithm's performance according to the entire list and the precision focuses only on the L links with the top ranks or the highest scores. The following is a detailed description of these two metrics.

(i) AUC.— Given the ranking of the non-observed links, the AUC value is the probability that a randomly chosen missing link (i.e., a link in E^P) has a higher score than a randomly chosen nonexistent link (i.e., a link in $U - E$). In the algorithmic implementation, we usually calculate the score of each non-observed link instead of the ordered list since the latter task is more time-consuming. The computational complexity for an ordered list of nonexistent links in a sparse network is $\mathcal{O}(|V|^2 \log |V|^2)$. Since the number of nodes $|V|$ can be very large, it is very time-consuming to obtain the exact value of AUC which requires $|E^P| \cdot [V(V-1)/2 - |E|]$ pairs of comparison. Instead of the exact value, to estimate the AUC value with very good accuracy does not need to know the ordered list. At each time step we randomly pick a missing link and a nonexistent link and compare their scores. If among n independent comparisons there are n' times the missing link that have a higher score and n'' times that have the same score, the AUC value is

$$\texttt{AUC} = \frac{n' + 0.5n''}{n}. \tag{5}$$

If all the scores are generated from an independent and identical distribution, the AUC value will be approximately 0.5. Thus the degree to which the value exceeds 0.5 indicates how much better the algorithm performs than pure chance.

(ii) Precision.— Given the ranking of the non-observed links, the precision is defined as the ratio of relevant items selected to the number of items selected. That is to say, if we take the top-L links as the predicted ones, among which L_r links are right (i.e., there are L_r links in the probe set E^P), then the precision equals L_r/L. Thus a higher precision value means a higher prediction accuracy.

Figure S1 shows an example of how to calculate the AUC and the precision. In this simple network there are five nodes, seven existent links, and three nonexistent links ($(1, 2)$, $(1, 4)$ and $(3, 4)$). To test the algorithm's accuracy, we select several existent links as probe links. For example, we can pick $(1, 3)$ and $(4, 5)$ as probe links (dashed lines in the right plot). Then algorithms can only use the information contained in the training network (presented by solid lines in the right plot). If an algorithm assigns scores of all non-observed links as $s_{12} = 0.4$, $s_{13} = 0.5$, $s_{14} = 0.6$, $s_{34} = 0.5$ and $s_{45} = 0.6$. To calculate the AUC, we compare the scores of a probe link and a nonexistent link. There are in total six pairs: $s_{13} > s_{12}$, $s_{13} < s_{14}$, $s_{13} = s_{34}$, $s_{45} > s_{12}$, $s_{45} = s_{14}$ and $s_{45} > s_{34}$. Thus the AUC value equals $(3 \times 1 + 2 \times 0.5)/6 \approx 0.67$. For the precision, if $L = 2$, the predicted links are $(1, 4)$ and $(4, 5)$. Clearly the former is wrong and the latter is right, and thus the precision equals 0.5.

In this paper we make predictions based solely on the known topology of the network (i.e., the information contained in training set). In real applications, generally the reliability of prediction is not revealed by the LP algorithm itself, and predictions are sometimes even quite distinct for different algorithms. σ_c gives a basic understanding of how predictable a network is. Intuitively speaking, σ_c is a metric of how the observed links and missing links are linearly consist.

IV. CORRELATION BETWEEN INDEPENDENT PERTURBATIONS

The correlation between the first-order corrections and the eigenvalues between independent perturbations is the foundation of the SPM. For the first-order perturbation, each edge acts independently of the correction of the eigen-

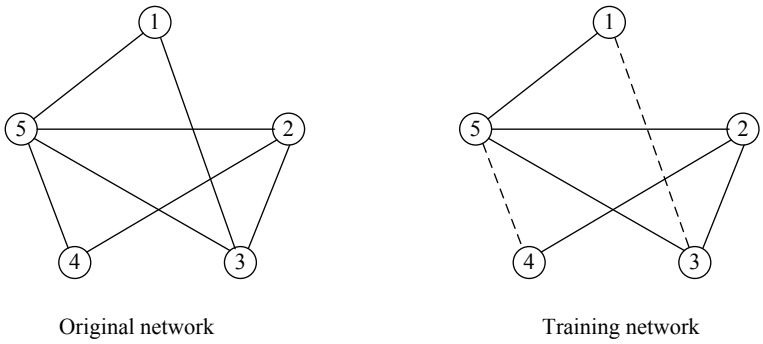

FIG. S1. An illustration about the calculation of AUC and Precision.

values. That is to say, two independent sets of the hidden edges ΔA_1 and ΔA_2 change the eigenvalues as

$$\begin{aligned}\Delta\lambda_k &= \frac{x_k^T \Delta A x_k}{x_k^T x_k} = \frac{x_k^T \left(\Delta A_1 + \Delta A_2\right) x_k}{x_k^T x_k} \\ &= \frac{x_k^T \Delta A_1 x_k}{x_k^T x_k} + \frac{x_k^T \Delta A_2 x_k}{x_k^T x_k} = \Delta\lambda_{k1} + \Delta\lambda_{k2}.\end{aligned} \tag{6}$$

The two sets are independent because they have no edges in common. We calculate the average Pearson correlation coefficient of $\Delta\lambda$ between independent perturbations, as shown in Table S1. $\Delta\lambda$ for independent perturbations are strongly correlated, implying that missing links can be predicted by perturbing the observed network. Generally, the larger the correlation r is, the higher precision SPM gives (see Table 1).

TABLE S1. Pearson correlation coefficient of $\Delta\lambda$ between independent perturbations. For a given network, we firstly remove two independent sets of edges, named ΔA_1 and ΔA_2, from the network. Then we perturb the rest network by ΔA_1 and ΔA_2, respectively, to obtain two group of the corrected eigenvalues.

Nets	**Jazz**	**Metabolic**	**Neural**	**USAir**	**Food web**	**Hamster**	**NetSci**	**Yeast**	**Email**	**Router**
r	0.920	0.778	0.758	0.837	0.882	0.853	0.592	0.680	0.663	0.781

V. FIVE STEPS TO CALCULATE PREDICTION ACCURACY OF SPM

For a given network to calculate the link prediction accuracy of SPM method, we have five steps:

Step 1: We first divide the true network A into training set E^T and probe set E^P, obviously, $A = A^T + A^P$.

Step 2: We further divide E^T into E^R and ΔE.

Step 3: We use ΔE to perturb E^R, and calculate $\tilde{A}$ following the procedures (i.e., step 2-4) for calculating σ_c.

Step 4: Repeat step 2 and 3 for ten times, namely we independently divide E^T into E^R and ΔE for ten times, then we obtain ten $\tilde{A}$ matrixes. Averaging the scores of ten $\tilde{A}$, we obtain the final score matrix $\langle\tilde{A}\rangle$ where $\langle\tilde{A}\rangle_{ij}$ is the score of link (i, j).

Step 5: Ranking all the non-observed links (i.e., links in $U - E^T$) in decreasing order according to their scores given by $\langle\tilde{A}\rangle$, we select $|E^P|$ links on the top places and see how many of them are in the probe set. This ratio is *precision*.

Repeat step 1-5 n times, we obtain an average precision. In this paper, we set $n = 100$.

VI. STATISTICAL FEATURES OF EXPERIMENTAL NETWORKS

We consider networks from disparate areas, including social, biological, and technological networks. The networks used in the experiment are described as follows and the basic statistical features are shown in Table S2. Directed links are treated as undirected, multiple links are treated as a single unweighted link and self loops are removed. Note that for the very large networks we consider the sampled subnetworks. The detailed sampling method is introduced in section VIII, which provides us with a useful tool for addressing large-scale networks.

(i) Jazz [4]: A collaboration network of jazz musicians consists of 198 nodes and 2742 interactions.
(ii) Metabolic [5]: A metabolic network of C.elegans.
(iii) Neural [6]: The neural network of C.elegans. The original network is directed and weighted; here we treat it as simple graph by simply ignore the directions and weights.
(iv) USAir [7]: The US Air transportation system.
(v) Food web [8]: A food web in Florida Bay during the rainy season.
(vi) Hamster [9]: A friendship network of users of the website hamsterster.com.
(vii) NetSci [10]: A coauthorship network of scientists working on network theory and experiment.
(viii) Yeast [11]: A protein-protein interaction network in budding yeast.
(ix) Email [12]: A network of e-mail interchanges between members of the University Rovira i Virgili (Tarragona).
(x) Router [13]: A symmetrized snapshot of the structure of the Internet at the level of autonomous systems.
(xi) Arxiv [14]: A collaboration graph of authors of scientific papers from the arXiv's High Energy Physics C Theory (hep-th) section. An edge between two authors represents a common publication. Timestamps denote the date of a publication.
(xii) Facebook [15]: A directed network of a small subset of posts to other user's wall on Facebook. The nodes of the network are Facebook users, and each directed edge represents one post, linking the users writing a post to the users whose wall the post is written on. Since users may write multiple posts on a wall, the network allows multiple edges connecting a single node pair. Since users may write on their own wall, the network contains loops.
(xiii) Enron [16]: This network consists of 1,148,072 emails sent between employees of Enron between 1999 and 2003. Nodes in the network are individual employees and edges are individual emails. It is possible to send an email to oneself, and thus the original network contains loops.

TABLE S2. The basic topological features of thirteen real networks. $|V|$ and $|E|$ are the number of nodes and links. C is the clustering coefficient [6] and r the assortative coefficient [17]. $\langle k\rangle$ is the average degree, $\langle d\rangle$ is the average shortest distance, and H is the degree heterogeneity, as $H = \langle k^2\rangle/\langle k\rangle^2$. Note that for the sampled subnetwork, $|V|$ of the original network are shown in the bracket.

Networks	$\|V\|$	$\|E\|$	C	r	$\langle k\rangle$	$\langle d\rangle$	H
Jazz	198	2742	0.618	0.020	27.697	2.235	1.395
Metabolic	453	2025	0.647	-0.226	8.940	2.664	4.485
Neural	297	2148	0.292	-0.163	14.465	2.455	1.801
USAir	332	2126	0.625	-0.208	12.807	2.738	3.464
Food web	128	2075	0.335	-0.112	32.422	1.776	1.237
Hamster	300(1858)	2503	0.201	-0.082	16.687	2.585	1.955
NetSci	300(1589)	707	0.727	-0.081	4.713	5.331	1.723
Yeast	300(2361)	830	0.122	-0.014	5.533	3.703	1.571
Email	300(1133)	1268	0.266	0.074	8.453	3.143	1.489
Router	300(5022)	530	0.039	-0.219	3.533	4.458	3.014
Arxiv	4000(22908)	612485	0.587	-0.102	306.243	2.018	2.074
Facebook	4000(56952)	51173	0.248	0.040	25.587	3.312	2.672
Enron	4000(87273)	77090	0.283	-0.151	38.545	2.834	3.507

VII. LINK PREDICTION ON REAL NETWORKS

We compare our method, structural perturbation method (SPM), to several well-known methods, including three local algorithms based on the number of common neighbors between pairs of nodes (CN, AA and RA), a path-dependent global method (Katz), and the approaches of Clauset et al. (HSM) and Guimerà et al. (SBM). For the definition of each algorithm see the **Materials and Methods**. For each real network a fraction of its links E^p will be removed to constitute the probe set which, in our experiments, always contain 10% of links in E. The rest of the links constitute the training set E^T used to generate an observable network. Using the LP methods we then calculate the existence likelihood of each node pair not connected in the observed network, and rank the node pairs in order of decreasing existence likelihood. Prediction accuracy is obtained by precision and AUC respectively, see the definition of these two evaluation metrics in **Materials and Methods**. We set $p^H = 0.1$ for SPM and $L = |0.1E|$ to calculate the precision. For the parameter-dependent Katz index, the present results are obtained under the optimal parameter α subject to the highest precision.

TABLE S3. Link prediction accuracy measured by AUC on the ten real networks.

AUC	Jazz	Metabolic	Neural	USAir	Food web	Hamster	NetSci	Yeast	Email	Router
SPM	**0.976**	0.922	0.885	0.932	**0.950**	**0.915**	0.918	**0.818**	0.827	0.786
CN	0.955	0.921	0.847	0.935	0.610	0.776	0.942	0.720	0.813	0.621
AA	0.962	0.953	0.863	0.946	0.611	0.777	0.942	0.721	0.813	0.621
RA	0.971	**0.958**	0.867	0.952	0.614	0.780	**0.947**	0.721	0.816	0.621
Katz	0.962	0.917	0.857	0.943	0.725	0.833	0.938	0.751	**0.857**	0.614
HSM	0.881	0.852	0.810	0.896	0.809	0.828	0.901	0.674	0.767	0.709
SBM	0.935	0.908	**0.889**	**0.954**	0.909	0.883	0.906	0.770	0.814	**0.920**

VIII. APPLYING TO LARGE NETWORKS

To apply our method to large networks, we use random walk sampling [18] to obtain a subnetwork. We do this by randomly picking a starting node and then using the random walk to select the subnetwork. At each time step there is a probability $c = 0.15$ that the random walk will jump back to the staring node. This procedure is continued until the desired number of nodes are selected. When the network is extremely large and computing the full spectrum unpractical, we calculate the σ_c of the sampled subnetwork to determine its link predictability. Figure S2 shows how the subnetwork sampled using the random walk method can successfully recover the σ_c of the original network. This enables us to approximate the predictability of large networks using the sampled subnetwork.

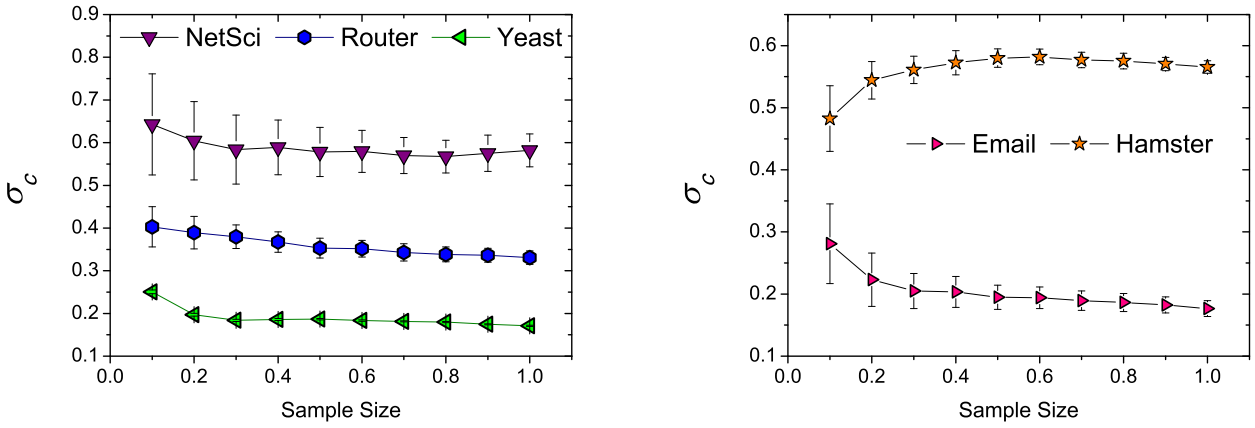

FIG. S2. (Color online). Structural consistency of sampled subnetwork for different sample sizes. When the sample size equals to 1 the sampled network is identical to the original network. Each point is obtained by averaging over 20 realizations.

IX. DEPENDENCE OF σ_c ON THE SIZE OF PERTURBATION SET p^H

We have shown how σ_c is related to the predictability of empirical networks in the main text. The calculation of σ_c relies on a structural perturbation of the given network. To confirm the relevance of σ_c in revealing the inherent link predictability of networks, we show that σ_c is not affected by the size of perturbation set E^H. We investigate the dependence of σ_c on the ratio of the perturbation set ΔE. We change p^H from 0.05 to 0.45 and find that the differences in σ_c are relatively small, see the results for ten real-world networks in Fig. S3. Note that σ_c varies slowly and steadily with p^H, indicating that σ_c is a robust metric for different sizes of perturbation sets. In practice, therefore, we can select approximately p^H of links from the given network to calculate σ_c. Since σ_c is not sensitive to the size of ΔE, we can use σ_c as a metric of link predictability of the network.

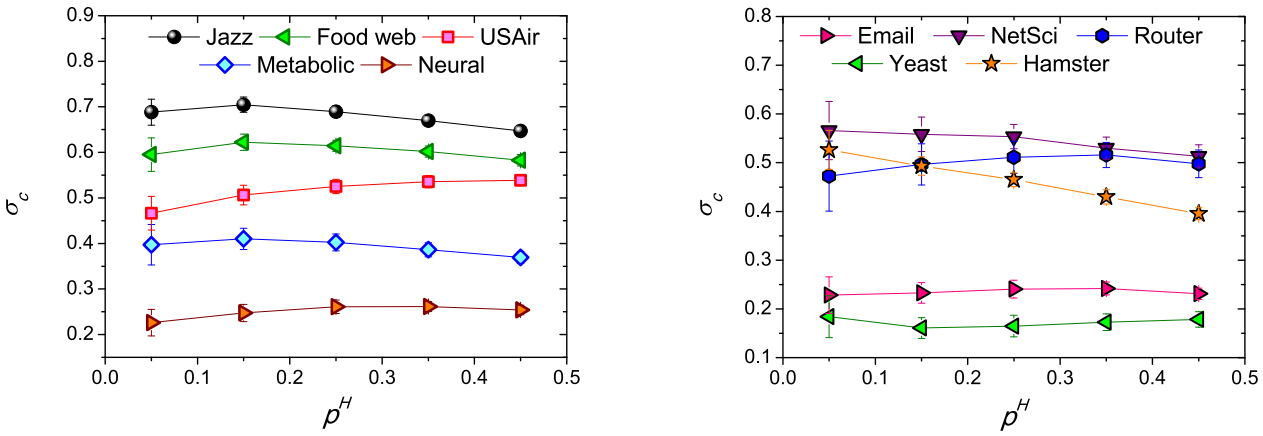

FIG. S3. (Color online). Dependence of network consistency on the size of perturbation set. The standard deviation is show as the Y-error bar. Each point is obtained by averaging over 100 realizations.

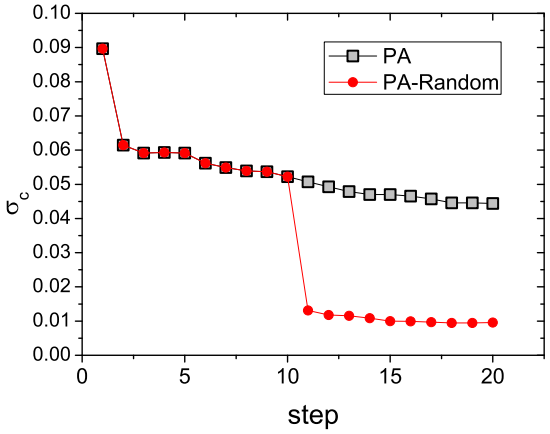

FIG. S4. The structural consistency for an artificial evolving network. This initial network is constructed by a configuration model with $N = 1000$, $\langle k \rangle = 4$ and $p(k) \sim k^{-3}$. In each time step, we added 1000 links into the network. The black squares represent the case we continuously using the preferential attachment (PA) mechanism to add new links for 20 time steps, while the red circles stand for the case we change from PA mechanism to random connecting strategy after the 10^{th} time step.

X. MONITOR THE SUDDEN CHANGES OF EVOLVING NETWORKS WITH σ_c

The index σ_c can be used to monitor the changes of network structure during its evolving. Figure S4 gives an example. We artificially build up an evolving model that starting from a configuration network [19] with size $N = 1000$, average degree $\langle k \rangle = 4$ and degree distribution $p(k) \sim k^{-3}$. In each step, we will add 1000 new links which are selected according to their popularity (preferential attachment mechanism), that is to say, the probability to add a link connecting nodes (i, j) is proportional to the degree product $k_i \cdot k_j$. After the 10^{th} step (10,000 links are added), we change to random connecting strategy. As shown in figure S4, the structural consistency can well detect this sudden change.

XI. TABLE OF NOTATIONS

The definition of some notations used in this paper are summarized in Table S4.

TABLE S4. Notations used in the paper.

Notations	Description
N	The number of nodes in given network
V	The set of nodes in given network
E	The set of edges of the given network
ΔE	The perturbation link set
E^R	The set of edges in $E - \Delta E$
ΔA	The adjacency matrix corresponding to ΔE
A^R	The adjacency matrix corresponding to E^R
A	The adjacency matrix of the given network, corresponding to E
E^T	The training set for link prediction
E^P	The probe set for link prediction
$\tilde{A}$	The perturbed network matrix
x_k	The eigenvectors of A^R
λ_k	The eigenvalues of A^R, corresponding to eigenvector x_k
T	The matrix transposition
σ_c	The structural consistency which measures the link predictability of network
p^H	The fraction of links that constitute the perturbation set

[1] MacArthur BD, Sánchez-García RJ (2009) Spectral characteristics of network redundancy. *Phys Rev E* 80(2):026117.

[2] Hanely JA, McNeil BJ (1983) A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology* 148(3):839-843.

[3] Herlocker JL, Konstann JA, Terveen K, Riedl JT (2004) Evaluating collaborative filtering recommender systems. *ACM Trans Inf Syst* 22(1):5-53.

[4] Gleiser P, Danon L (2003) Community structure in Jazz. *Advances in complex systems* 6(04):565.

[5] Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. *Phys Rev E* 72(2):027104.

[6] Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. *Nature* 393(6684):440-442.

[7] http://vlado.fmf.uni-lj.si/pub/networks/data/

[8] Ulanowicz RE, Bondavalli C, Egnotovich MS (1998) Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 97: The Florida Bay Ecosystem. *Technical report* CBL:98-123. http://www.cbl.umces.edu/ atlss/FBay701.html

[9] konect:2013:petster-friendships-hamster, Hamsterster friendships unique network dataset – KONECT, (2013). http://konect.uni-koblenz.de/networks/petster-friendships-hamster

[10] Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys Rev E* 74(3):036104.

[11] Sun SW, Ling LJ, Zhang N, Li GJ, Chen RS (2003) Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research* 31(9):2443-2450

[12] Guimerà R, Danon L, Diaz-Guilera A, Giralt F, Arenas A (2003) Self-similar community structure in a network of human interactions. *Phys Rev E* 68(6):065103.

[13] Spring N, Mahajan R, Wetherall D, Anderson T (2004) Measuring ISP topologies with Rocketfuel. *IEEE/ ACM Trans Networking* 12(1):2-16.

[14] Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1(1):2. (http://konect.uni-koblenz.de/networks/ca-cit-HepTh)

[15] Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in Facebook. *In Proc. Workshop on Online Social Networks*, pp 37-42. (http://konect.uni-koblenz.de/networks/facebook-wosn-wall)

[16] Klimt B, Yang Y (2004) The Enron corpus: A new dataset for email classification research. *In Proc. European Conf. on Machine Learning*, pp 217-226. (http://konect.uni-koblenz.de/networks/enron)

[17] Newman MEJ (2002) Assortative mixing in networks. *Phys Rev Lett* 89:208701.

[18] Leskovec J, Faloutsos C (2006) Sampling from Large Graphs. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*(ACM, New York), pp 631-636.

[19] Catanzaro M, Boguna M, Pastor-Satorras R (2005) Generation of uncorrelated random scale-free networks. *Phys Rev E* 71(2):027103.