RESEARCH ARTICLE

# A Noise-Filtering Method for Link Prediction in Complex Networks

**Bo Ouyang[1]*, Lurong Jiang[2☯], Zhaosheng Teng[1☯]**

**1** College of Electrical and Information Engineering, Hunan University, Changsha, Hunan Province, China, **2** School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, Zhejiang Province, China

☯ These authors contributed equally to this work.
* ouyangbo@hnu.edu.cn

## Abstract

Link prediction plays an important role in both finding missing links in networked systems and complementing our understanding of the evolution of networks. Much attention from the network science community are paid to figure out how to efficiently predict the missing/future links based on the observed topology. Real-world information always contain noise, which is also the case in an observed network. This problem is rarely considered in existing methods. In this paper, we treat the existence of observed links as known information. By filtering out noises in this information, the underlying regularity of the connection information is retrieved and then used to predict missing or future links. Experiments on various empirical networks show that our method performs noticeably better than baseline algorithms.

OPEN ACCESS

**Citation:** Ouyang B, Jiang L, Teng Z (2016) A Noise-Filtering Method for Link Prediction in Complex Networks. PLoS ONE 11(1): e0146925. doi:10.1371/journal.pone.0146925

**Editor:** Wen-Bo Du, Beihang University, CHINA

**Received:** November 13, 2015

**Accepted:** December 24, 2015

**Published:** January 20, 2016

**Data Availability Statement:** All relevant data are within the paper.

**Funding:** This work was supported by Fundamental Research Funds for the Central Universities of China, Grant No. 531107040852. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

## Introduction

About one and a half decades ago, Barabási and Albert pointed out that the property of scale-invariance of many real networked systems originates from a specific growth process, named preferential attachment [1]. Since then, the study of complex networks has led to dramatic changes in many different fields [2–7], and also, many facets of node attractiveness in growing networks, rather than preferential attachment, have been revealed, e.g. similarity [8]. Since different growing processes often result in networks with strikingly different macroscopic properties, how real-world networks are evolved is a fundamental question in understanding our complex world. Link prediction, one of whose capabilities is to rank the best candidates of future links, plays an important role in revealing the evolution processes of networks [9, 10].

On the other hand, many applications have to predict missing links in networked systems [11–13]. Determining whether a link exists in such networks is usually very costly, yet the answer is crucial. For example, knowing the map of protein-protein interactions will reveal many aspects of the cellular function [14], but little has been studied. Link prediction are also widely used in these applications [15, 16].

The problem of link prediction has received much attention from the network science community in the past few years [9, 12, 17, 18]. In general, both topological feature and node attributes can be used in the prediction. However, the latter is usually unavailable or unreliable. For example, in online social networks, the personal information of users are inaccessible due to privacy policies. Thus, many algorithms consider only topological features.

Basically, there are two classes of topological methods—similarity-based and likelihood-based algorithms. Similarity based algorithms assume that two nodes are likely to be connected if they are similar. It assigns a score $s_{xy}$ to each pair of nodes $x$ and $y$, which is defined as the similarity between them. All non-observed links are ranked according to their scores, and the links connecting more similar nodes are supposed to be of higher existence likelihoods. A wealth of methods of this type have been proposed. For example, CN (Common Neighbours) [19] uses the number of common neighbours to rank the similarity of nodes and the likelihood that they are/will be linked. Many variations of CN are also proposed: AA (Adamic-Adar) [20], Resource Allocation (RA) [19] give more importance to common neighbours with lower degree, and Jaccard's index is a normalised CN. Only local structural information are used in these methods. There are also methods utilizing quasi-global or global information. For example, the Local Path method defines the similarity as the number of paths passing through two nodes, whose length may be larger than 2.

Recently, the organization patterns existing in many real-world networks are utilized in predicting missing links. Likelihood-based methods make assumptions of the structure, with specific parameters obtained by maximising the likelihood of the known structure. Predictions of the non-observed links are made based on the presumed pattern and the parameters. For example, Ref. [21] utilizes the hierarchical structure existing in many networks to predict missing links. And Cannistraci et al. propose the local-community-paradigm to improve the performance of classical predictors [13].

We know that real-world information always contains noise, which is also the case in an observed network. However, this problem is rarely considered in existing methods. In Ref. [18], the authors use the average of the eigen-decomposition of perturbed adjacency matrix (by removing some links) to suppress the noise. However, the underlying physical meaning is not clear, say, why should the eigenvectors of the adjacency matrix reflect the regularity of a network, if they actually are sensitive to perturbation [22]? Besides, it has a high computational complexity. In this paper, by treating the existence of observed links as known "information" (as in [23, 24]), and filtering out the noise in it, we obtain similarity scores for all non-observed links. We give a more theoretical analysis of the link prediction problem and a more meaningful demonstration of the noise-filtering (NF) method. Our method outperforms the typical predictors.

# Materials and Methods

## Metrics

In this paper, two metrics are used to compare the performance of the base-line algorithms and the proposed noise-filtering method.

Consider that we are given an simple network $G(V, E)$, where $V$ and $E$ are the set of nodes and links, respectively. By "simple", we mean there are no self-loops or multi-links in the network. In a similarity-based algorithm, for each pair of nodes $x, y \in V$ without a link, a similarity score is assigned. Then all unlinked pairs are ranked in descending order according to their scores, and the links on the top are considered as the ones with the highest likelihoods to be connected.

To test the accuracy of a predictor, we randomly divide the observed links in the network into a training set $E^T$ and a probe set $E^P$. Here, $E^T$ is treated as known information while $E^P$ is only used to test the accuracy. Clearly, we have $E^T \cup E^P = E$ and $E^T \cap E^P = \emptyset$.

In this study, we use two metrics, AUC (Area Under the Receiver operating characteristic curve) and precision to evaluate the performance of a predictor. They are defined as follows.

- AUC: AUC is a metric in the receiver operating characteristics (ROC) analysis [25]. Taking the top $L$ links as predicted links, a ROC curve is obtained by plotting true positive rates versus false positive rates for varying $L$ values. Thus AUC can be interpreted as the probability that a randomly chosen missing link (i.e., a link in $E^P$) has a higher score than a randomly chosen non-existent link (i.e., a link in $U - E$), in the rank of all non-observed links. In the algorithmic implementation, if among $n$ times of independent comparisons, there are $n'$ times in which the score of the missing link is higher than that of the non-existent link and $n''$ times in which the two have the same score, then AUC can be expressed as

  $$AUC = \frac{n' + n''}{n} \ . \tag{1}$$

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

- 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. Thus if we choose the top-$L$ links in the rank, and there are $L_r$ links correctly predicted, then

  $$Precision = L_r/L \ . \tag{2}$$

  Clearly, higher precision means higher accuracy. In this paper, $L$ is always set to the size of the probe set.

## Data Description

Networks from different fields are considered in the experiment, including biological, social, and technological networks. The original networks are turned into undirected, and simple (with multiple links or loops removed) networks. These networks are described in the following. i) Karate [26]: A social network of a university karate club. ii) FoodWeb [27]: A food web in Florida Bay during the rainy season. iii) Jazz [28]: A collaboration network of jazz musicians. iv) Neural [29]: The neural network of C.elegans. v) USAir [30]: The US Air transportation network. vi) Metabolic: The metabolic network of C.elegans. vii) Email [31]: A network of Alex Arenas's email. viii) PB [32]: A network of US political blogs. ix) Yeast [33]: A protein-protein interaction network. x) EPA [34]: A network of web pages linking to the website www.epa.gov. xi) Router [35]: The router-level topology of the Internet. xii) WikiVote [36, 37]: The network contains all the Wikipedia voting data from its inception till January 2008. Their basic topological parameters are summarized in Table 1.

## Baseline Algorithms for Comparison

In this paper, six representative similarity indices are considered for performance comparison, including the Common Neighbours (CN), Adamic-Adar (AA) [20], Resource Allocation (RA) [19], Preferential Attachment (PA) [38], Local Path (LP) [39], and Katz [40]. The first four are local indices, the fifth is a quasi-local index, and the last is a global index. Some of them are briefly introduced earlier. Here we present the details of these algorithms.

1. CN index. The CN index follows the intuition that two nodes $x$ and $y$ are more likely to have connection if their nearest neighbours overlap substantially. The similarity score is obtained by

   $$s_{xy} = |\Gamma(x) \cap \Gamma(y)| \ , \tag{3}$$

   where $\Gamma(x)$ is the set of neighbours of $x$ and $|\cdot|$ denotes the cardinality of a set.

**Table 1. Topological parameters of the real-world networks.**

| | \|*V*\| | \|*E*\| | *C* | *r* | ⟨*k*⟩ | *H* |
|---|---|---|---|---|---|---|
| Karate | 34 | 78 | 0.571 | -0.476 | 4.588 | 1.693 |
| FoodWeb | 128 | 2075 | 0.335 | -0.112 | 32.422 | 1.237 |
| Jazz | 198 | 2742 | 0.617 | 0.020 | 27.697 | 1.395 |
| Neural | 297 | 2148 | 0.292 | -0.163 | 14.465 | 1.801 |
| USAir | 332 | 2126 | 0.625 | -0.208 | 12.807 | 3.464 |
| Metabolic | 453 | 2025 | 0.646 | -0.226 | 8.940 | 4.485 |
| Email | 1133 | 5451 | 0.220 | 0.078 | 9.622 | 1.942 |
| PB | 1490 | 16715 | 0.263 | -0.221 | 22.436 | 3.622 |
| Yeast | 2361 | 6646 | 0.130 | -0.099 | 5.630 | 2.944 |
| EPA | 4772 | 8909 | 0.064 | -0.303 | 3.734 | 7.573 |
| Router | 5022 | 6258 | 0.012 | -0.138 | 2.492 | 5.503 |
| WikiVote | 8297 | 100762 | 0.121 | -0.083 | 24.289 | 5.985 |

|*V*| and |*E*| are the number of nodes and links. *C* is the clustering coefficient and *r* the degree-degree correlation coefficient. ⟨*k*⟩ is the average degree, ⟨*d*⟩ is the average shortest distance, and *H* is the degree heterogeneity $H = \langle k^2\rangle/\langle k\rangle^2$.

doi:10.1371/journal.pone.0146925.t001

2. AA index. AA is a variation of CN: it gives less importance to common neighbours with high degree:

$$s_{xy} = \sum_{s\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\log|\Gamma(s)|} \ . \tag{4}$$

3. RA index. Similar to AA, the only difference is that RA punishes high-degree common neighbours to a higher extent:

$$s_{xy} = \sum_{s\in\Gamma(x)\cap\Gamma(y)} \frac{1}{|\Gamma(s)|} \ . \tag{5}$$

4. PA index. The PA index supposes that popular nodes are more likely to be connected to. This index is defined as

$$s_{xy} = |\Gamma(x)||\Gamma(y)| \ . \tag{6}$$

5. LP index. Unlike the previous indices, LP uses second order information (information about neighbours of the neighbours) to improve performance. It is defined by

$$s_{xy} = (\mathbf{A}^2)_{xy} + \epsilon(\mathbf{A}^2)_{xy} \ . \tag{7}$$

6. Katz index. This index sums over the number of paths (including loops) between two nodes, with each number exponentially damped by the path length

$$s_{xy} = \beta\mathbf{A}_{xy} + \beta^2(\mathbf{A}^2)_{xy} + \beta^3(\mathbf{A}^3)_{xy} + \cdots = (\mathbf{I} - \beta\mathbf{A})^{-1} - \mathbf{I} \ . \tag{8}$$

Note that the LP index and Katz are both parameter-dependent.

## Results

### Link Prediction via Noise Filtering

In many networks, the formation of links usually embodies both regularities and irregularities. Only the former shows a uniform pattern, which is called the intrinsic pattern. For a specific link, if its existence does not correspond with this pattern, then its existence should be treated as noise. For a specific link, if its existence does not correspond with the connection pattern of the whole network, then its existence is treated as noise. A large body of link prediction methods (i.e. common neighbor method) assumes that nodes are linked if they are similar. Following this assumption, we treat links connecting dissimilar nodes as noise. By filtering out the noise, we can obtain the intrinsic connection pattern, which can be further used to predict missing or future links.

To this end, one has to define a measure to quantify the degree to which a link connects dissimilar nodes.

For every node in the network, assume that its topological features are captured by some vectors in $\mathbb{R}^m$. Define the feature matrix $\mathbf{X}$ to be an $n$-by-$m$ matrix whose rows are the feature vectors of nodes. Thus, $\mathbf{X}_{ik}$ is the $k$-th feature of node $i$, and $\mathbf{X}_{\bullet k}$, the $k$-th column vector of $\mathbf{X}$, is the $k$-th feature of all nodes. In real-world cases, features usually contain noise.

In some typical link prediction methods (i.e. common neighbor method), nodes are assumed to be linked because they are similar. Now focusing on the $k$-th feature, we may measure to what degree dissimilar nodes are linked in the whole network by

$$D'_k = \sum_{i\sim j} (\mathbf{X}_{ik} - \mathbf{X}_{jk})^2 = \mathbf{X}_{\bullet k}^T \mathbf{L} \mathbf{X}_{\bullet k} \ ,$$

where $i \sim j$ indicates that $i$ and $j$ are neighbors, and $\mathbf{L}$ is the Laplacian matrix [41]. However, this measure is biased. In the rhs of the first equation, the feature $\mathbf{X}_{ik}$ of node $i$ appears in $d_i$ different terms in the summation, where $d_i$ is the degree of node $i$. So features of high-degree nodes dominate the value of $D'_k$, while in many real-world networks, most nodes are of low degree [1]. Thus the value of $D'_k$ does not properly count the similarity of the features from the majority.

The rightmost term in the above equation is the quadratic form of the Laplacian. To treat features from different nodes equally, a natural alternative is using the quadratic form of the normalised Laplacian matrix $\tilde{\mathbf{L}}$ [41],

$$D_k = \mathbf{X}_{\bullet k}^T \tilde{\mathbf{L}} \mathbf{X}_{\bullet k} \ . \tag{9}$$

The quadratic form of $\tilde{\mathbf{L}}$ has similar interpretation of that of $\mathbf{L}$, so larger $D_k$ indicates to a larger extent, dissimilar nodes are linked together. Thus $D_k$ can be used as a non-biased dissimilarity measure of the $k$-th feature.

In signal processing, to filter out noise, the signal is decomposed into a set of sine waves with different frequencies. For higher frequencies, the sine waves oscillate much more rapidly. Then the waves with frequencies that are considered within the band of noise are filtered out. In our case, the eigenvectors of the normalised Laplacian provide a similar notion of frequency. To understand this, denote by $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ the eigenvalues of the normalised Laplacian matrix $\tilde{\mathbf{L}}$, and $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$ the corresponding eigenvectors. The Courant-Fischer Theorem [42] tells us that

$$\mathbf{v}_1 = \arg \min_{\mathbf{x}:\|\mathbf{x}\|_2=1} \mathbf{x}^T \tilde{\mathbf{L}} \mathbf{x} \ , \tag{10}$$

and

$$\mathbf{v}_l = \arg \min_{\mathbf{x}:\|\mathbf{x}\|_2=1, \mathbf{x}\perp \mathrm{span}\{\mathbf{v}_1,\ldots \mathbf{v}_{l-1}\}} \mathbf{x}^T \tilde{\mathbf{L}} \mathbf{x} \tag{11}$$

So, if $\mathbf{X}_{\bullet k} = \mathbf{v}_1$, then $D_k$ achieves its smallest, which indicates that $\mathbf{v}_1$ oscillates slowly among connected nodes (since $D_k$ is a dissimilarity measure). The eigenvectors associated with larger eigenvalues oscillate more rapidly.

Similar to filtering noise in signal processing, we can project $\mathbf{X}_{\bullet k}$ onto $\{\mathbf{v}_i\}$, and filter out the components with high "frequency", i.e., the components on $\mathbf{v}_i$ with large subscript $i$, since we treat the existence of links connecting dissimilar nodes as noise. Denote the cut-off threshold by $t$, the noise-filtered $\mathbf{X}_{\bullet k}$ reads

$$\hat{\mathbf{X}}_{\bullet k} = (\mathbf{v}_1^T \mathbf{X}_{\bullet k})\mathbf{v}_1 + (\mathbf{v}_2^T \mathbf{X}_{\bullet k})\mathbf{v}_2 + \cdots + (\mathbf{v}_t^T \mathbf{X}_{\bullet k})\mathbf{v}_t = \mathbf{V}_t \mathbf{V}_t^T \mathbf{X}_{\bullet k} \ , \tag{12}$$

in which $\mathbf{V}_t = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_t]$ is a matrix whose columns are the first $t$ eigenvectors of $\mathbf{L}$ with the smallest eigenvalues.

Since no prerequisite is required for $k$, we can easily generalise the above derivation for the $k$-th feature to any other feature. Then we obtain the noise-filtered features for the whole network

$$\hat{\mathbf{X}} = \mathbf{V}_t \mathbf{V}_t^T \mathbf{X} \ . \tag{13}$$

For any node $i$, its connections with all other nodes in the same network are totally characterized by the corresponding rows in the adjacency matrix $\mathbf{A}$. So one may use these rows as the feature vectors for nodes, as in [43, 44], and interpret the $k$-th feature of node $i$ as whether it is a neighbour of $k$. But there are some minor issues with this choice. Recall that the above derivation is based on the minimisation of the dissimilarity measure of all linked nodes (see Eq (9)). We now consider two linked nodes $i$ and $j$, which have exactly the same neighbourhoods, so we expect the dissimilarity of them is 0. However, their $i$-th feature will not be the same, since the $i$-th feature of $i$ is 0 while the $i$-th feature of $j$ is 1. This is the same with the $j$-th feature. We can see from this analysis that one can use the rows of $\mathbf{A} + \mathbf{I}$ rather than $\mathbf{A}$ as the feature vectors for nodes. So the $k$-th feature of node $i$ can be interpreted as whether its to node $k$ is no more than 1. This is further demonstrated in Fig 1.

Apply the above methodology, we have

$$\hat{\mathbf{S}} = \mathbf{V}_t \mathbf{V}_t^T (\mathbf{A} + \mathbf{I}) \ . \tag{14}$$

Entries of $\hat{\mathbf{S}}$ reflect the intrinsic connection pattern, so they can be used to predict missing links. However, since we are focusing on undirected networks, there is still one problem with $\hat{\mathbf{S}}$. We can see that according to Eq (14), it might not be symmetric. So we will make predictions based on entries of $\frac{1}{2}(\hat{\mathbf{S}} + \hat{\mathbf{S}}^T)$ instead of $\hat{\mathbf{S}}$.

## Experimental Results

To compare the performance of the Noise-Filtering (NF) method and some well-known algorithms, 12 real-world networks, including biological, social, and technological networks, are considered in the experiments. They are transformed into undirected, and simple (with multiple links or loops removed) networks. The resulting networks are summarized in Table 1.

Table 2 shows the prediction accuracy measured by AUC. Results measured by another widely used metric, *precision*, is presented in Table 3. These metrics are introduced in the Methods section. The highest AUC/precision for each network (in each column) is shown in

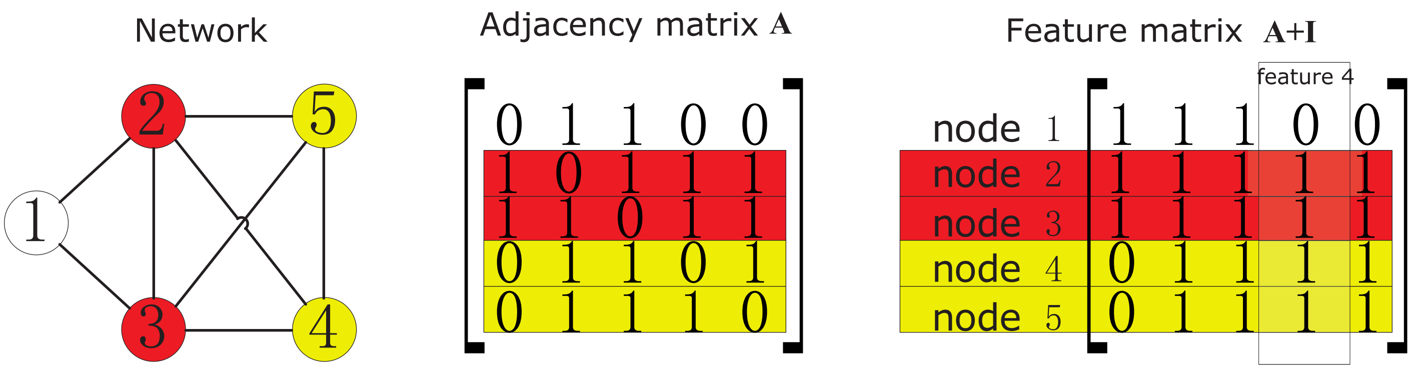

**Fig 1. Demonstration of using rows of A + I as the feature vectors for nodes.** In the network, nodes 4 and 5 are topologically equivalent. However, the 4th row of **A** reads [0, 1, 1, 0, 1], and the 5th reads [0, 1, 1, 1, 0], which are different. By adding **I**, the 4th and 5th rows of **A** + **I** now are both [0, 1, 1, 1, 1], which is exactly what we want. This is also the case for nodes 2 and 3. The *k*-th feature of a node can be interpreted as whether the distance between it and node *k* is no more than 1. For example, the distance between node 1 and 4 is greater than 1, while the distance between all the other nodes and node 4 are within 1, so the 4th feature is $[0, 1, 1, 1, 1]^T$.

doi:10.1371/journal.pone.0146925.g001

**Table 2. Comparison of the prediction accuracy under the AUC metric in real-world networks.**

| | CN | AA | RA | PA | LP | Katz | NF |
|---|---|---|---|---|---|---|---|
| Karate | 0.6994(162) | 0.7338(202) | 0.7281(182) | 0.7006(297) | 0.7206(200) | 0.7375(284) | **0.8113(211)** |
| FoodWeb | 0.6104(11) | 0.6094(11) | 0.6120(8) | 0.7332(9) | 0.6235(11) | 0.6770(10) | **0.8150(8)** |
| Jazz | 0.9545(2) | 0.9619(2) | **0.9701(1)** | 0.7668(8) | 0.9591(1) | 0.9485(2) | 0.9663(1) |
| Neural | 0.8441(4) | 0.8589(4) | 0.8644(4) | 0.7529(7) | 0.8595(6) | 0.8575(5) | **0.8847(4)** |
| USAir | 0.9359(3) | 0.9477(3) | 0.9537(2) | 0.8856(5) | 0.9427(3) | 0.9242(3) | **0.9599(2)** |
| Metabolic | 0.9198(3) | 0.9506(2) | **0.9544(2)** | 0.8172(7) | 0.9233(3) | 0.9195(4) | 0.9319(2) |
| Email | 0.8442(1) | 0.8464(1) | 0.8467(1) | 0.7779(3) | **0.8974(1)** | 0.8942(2) | 0.8973(1) |
| PB | 0.9368(0) | 0.9396(0) | 0.9398(0) | 0.9325(0) | 0.9495(0) | **0.9500(0)** | 0.9336(1) |
| Yeast | 0.7061(0) | 0.7066(0) | 0.7061(1) | 0.7865(3) | **0.8357(1)** | 0.8184(2) | 0.7989(3) |
| EPA | 0.5860(0) | 0.5865(0) | 0.5868(0) | 0.7371(2) | 0.7855(0) | 0.7376(1) | **0.7915(2)** |
| Router | 0.5580(0) | 0.5579(0) | 0.5579(0) | 0.4694(3) | 0.6320(0) | 0.3738(3) | **0.6654(6)** |
| Wikivote | 0.9337(0) | 0.9347(0) | 0.9344(0) | 0.9484(0) | 0.9616(0) | 0.9584(0) | **0.9646(0)** |

Each value is obtained by averaging over 100 implementations with independent random divisions of the training set(90%) and the probe set(10%). The method proposed in this paper is in the last column, NF (Noise Filtering). The best result achieved for each network data is in boldface. The numbers in the brackets denote the standard deviations. For example, 0.6994(162) means that the AUC value is 0.6994 and the standard deviation is $162 \times 10^{-4}$.

doi:10.1371/journal.pone.0146925.t002

boldface. Under the AUC metric, NF performs best in 7 out of 12 networks, while under the precision metric, NF performs best in 9 of them. Figs 2 and 3 compare prediction accuracy of different algorithms under varied partitioning ratio. It can be seen that the proposed method is either the best or very close to the best, except for only one network—PB. Moreover, the robustness of the proposed method can also be verified by Figs 2 and 3. Since in most networks, the accuracy of the proposed method is either the best or very close to the best, even with the size of training sets varied.

Intuitively, the more the amount of known information, the higher the prediction accuracy. But in Fig 3, we see that most of the time, the precisions do not increase with the size of training sets. This is due to different sizes of probe sets (follow a conventional way, we always set *L* in

**Table 3. Comparison of the prediction accuracy under the precision metric in real-world networks.**

| | CN | AA | RA | PA | LP | Katz | NF |
|---|---|---|---|---|---|---|---|
| Karate | 0.1525(96) | 0.1538(156) | 0.1538(146) | 0.0863(68) | **0.1750(100)** | 0.1613(123) | 0.1487(93) |
| FoodWeb | 0.0707(2) | 0.0755(2) | 0.0754(3) | 0.1607(4) | 0.0758(2) | 0.1023(3) | **0.1762(5)** |
| Jazz | 0.5044(6) | 0.5244(6) | 0.5393(5) | 0.1300(4) | 0.5120(7) | 0.4920(6) | **0.6225(5)** |
| Neural | 0.0962(2) | 0.1039(3) | 0.1025(3) | 0.0575(2) | 0.0985(3) | 0.1027(2) | **0.1262(3)** |
| USAir | 0.3730(8) | 0.3898(8) | **0.4505(9)** | 0.3164(7) | 0.3738(9) | 0.3695(8) | 0.3905(9) |
| Metabolic | 0.1378(4) | 0.1932(4) | 0.2680(5) | 0.0999(4) | 0.1449(5) | 0.1408(4) | **0.2113(6)** |
| Email | 0.1392(2) | 0.1552(2) | 0.1400(2) | 0.0174(0) | 0.1469(1) | 0.1355(2) | **0.1503(2)** |
| PB | 0.1729(0) | 0.1716(0) | 0.1493(0) | 0.0652(0) | 0.1735(0) | **0.1744(0)** | 0.0861(11) |
| Yeast | 0.0924(0) | 0.0912(0) | 0.0736(0) | 0.0093(0) | 0.0950(1) | 0.0925(0) | **0.1070(1)** |
| EPA | 0.0090(0) | 0.0148(0) | 0.0198(0) | 0.0044(0) | 0.0135(0) | 0.0136(0) | **0.0642(0)** |
| Router | 0.0166(0) | 0.0162(0) | 0.0096(0) | 0.0096(0) | 0.0212(0) | 0.0226(0) | **0.0253(0)** |
| Wikivote | 0.1009(0) | 0.0999(0) | 0.0833(0) | 0.0616(0) | 0.1005(0) | 0.1028(0) | **0.1352(0)** |

Each value is obtained by averaging over 100 implementations with independent random divisions of the training set(90%) and the probe set(10%). The method proposed in this paper is in the last column, NF (Noise Filtering). The best result achieved for each network data is in boldface. The numbers in the brackets denote the standard deviations. For example, 0.1525(96) means that the precision value is 0.1525 and the standard deviation is 96 × $10^{-4}$.

doi:10.1371/journal.pone.0146925.t003

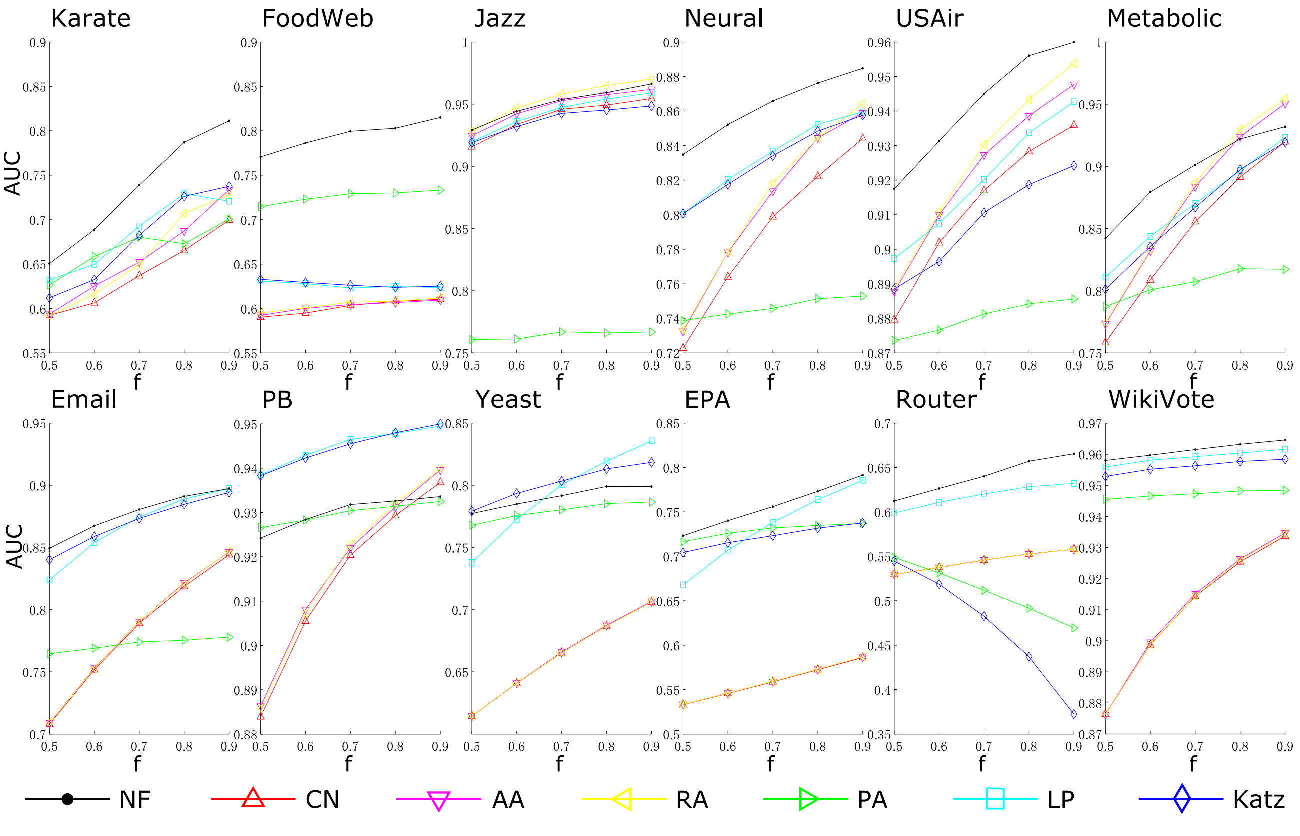

**Fig 2. Comparison of prediction accuracy under the AUC metric.** The fraction of training sets *f* is varied from 0.5 to 0.9.

doi:10.1371/journal.pone.0146925.g002

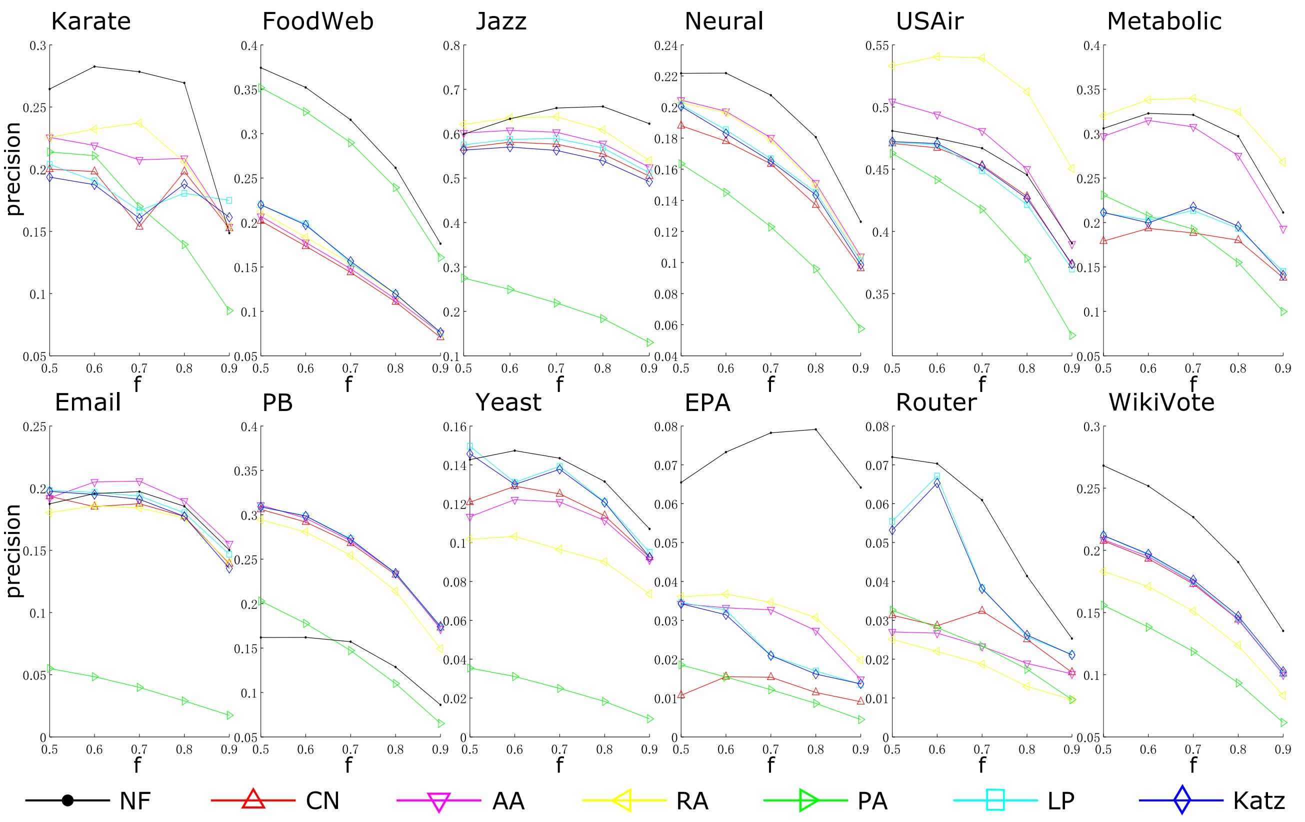

**Fig 3. Comparison of prediction accuracy under the precision metric.** The fraction of training sets *f* is varied from 0.5 to 0.9.

doi:10.1371/journal.pone.0146925.g003

Eq (2) to the size of the probe set). Thus with different sizes of training set, the precisions cannot be compared [45].

For all the parameter dependent methods considered in the experiment, i.e., LP, Katz, and NF, the results correspond to the optimal parameter, subject to the highest prediction accuracy. The optimal parameter can be found through a process similar to the *K*-fold validation. For example, in the proposed method NF, the training set is first partitioned into *K* units, a single unit is retained as the validation data for testing the method with specific *t*, and the remaining $K - 1$ units are used as known information. The cross-validation is then repeated *K* times (the folds), with each of the *K* units used exactly once as the validation data. The *K* results from the folds are then averaged. This whole process is repeated several times to find the optimal value of *t* (the value of optimal *t* is manually bounded in the range [1, 125], so the computation complexity is relatively small). In Fig 4, we see that for the two metrics considered here, the optimal *t* is robust, since the value of *t* where the prediction accuracy peaks does not change with the choice of the size of the training set. So there is no need to search for an optimal *t* in every single run of the simulation. Once the optimal *t* is found, it is set to this same value in all subsequent simulations, even with the size of the training set varied.

The experiments are conducted on a workstation with 64 GB RAM and an Intel (R) Xeon (R) E5-2687W @ 3.10 GHz 8-core processor. The comparison of computational time is summarized in Table 4. We see that the proposed method NF has similar run time with the global index Katz, especially on large networks, but having better performance.

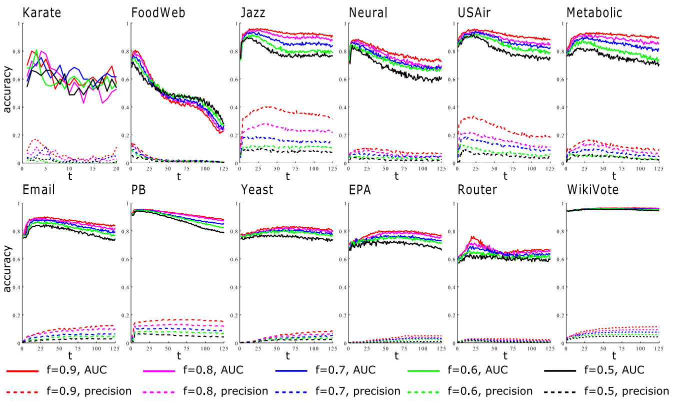

**Fig 4. Prediction accuracy with different cutoff threshold *t* in the proposed noise-filtering method.** The symbol *f* denotes the fraction of links in the training sets.

doi:10.1371/journal.pone.0146925.g004

**Table 4. Comparison of the computational efficiency in real-world networks.**

| | CN | AA | RA | PA | LP | Katz | NF |
|---|---|---|---|---|---|---|---|
| Karate | 0.2722 | 0.0863 | 0.0794 | 0.0741 | 0.0765 | 0.0767 | 0.3953 |
| FoodWeb | 0.1265 | 0.1332 | 0.1319 | 0.1408 | 0.1511 | 0.1659 | 0.2652 |
| Jazz | 0.1928 | 0.1939 | 0.2207 | 0.2117 | 0.2357 | 0.2596 | 0.4017 |
| Neural | 0.2041 | 0.2295 | 0.2302 | 0.2499 | 0.2491 | 0.3566 | 0.4580 |
| USAir | 0.2428 | 0.2789 | 0.2220 | 0.2788 | 0.3358 | 0.3118 | 0.5731 |
| Metabolic | 0.3635 | 0.3644 | 0.3881 | 0.4841 | 0.5499 | 0.5712 | 0.6719 |
| Email | 1.3969 | 1.6700 | 1.5221 | 3.2462 | 2.4013 | 5.0422 | 3.1099 |
| PB | 4.5587 | 5.0003 | 5.1569 | 6.0813 | 6.8293 | 9.7084 | 6.0244 |
| Yeast | 4.9859 | 6.8925 | 6.1101 | 13.9745 | 7.1093 | 21.4607 | 12.9680 |
| EPA | 20.3863 | 29.6148 | 26.7357 | 53.4191 | 24.4295 | 89.9699 | 50.5754 |
| Router | 19.2990 | 33.6029 | 25.6877 | 64.5513 | 23.8481 | 72.9081 | 89.1175 |
| WikiVote | 366.9862 | 387.7545 | 386.7545 | 447.2611 | 453.6359 | 704.4210 | 526.8122 |

Each value is the total time (in seconds) for 100 runs, with independent random divisions of the training set(90%) and the probe set(10%). The method proposed in this paper is in the last column, NF (Noise Filtering).

doi:10.1371/journal.pone.0146925.t004

## Discussion

Real-world information always contains noise. This is also the case when making observation of a network structure. This problem is rarely considered in existing link prediction methods. To address this issue, we treat the connection of a given network as known information, and filter out the noises in it, based on an assumption that connected nodes should have similar neighbourhoods. The underlying regularity of the connection information is then retrieved and used to predict missing or future links. Experimental results show that it performs better than typical algorithms. Future works include how to improve the performance of existing methods based on the same idea of noise filtering.

## Acknowledgments

We acknowledge B. Y. Zhu, for his helpful suggestions. This work is supported by the Fundamental Research Funds for the Central Universities of China, Grant No. 531107040852.

## Author Contributions

Conceived and designed the experiments: BO LJ. Performed the experiments: BO LJ. Analyzed the data: BO ZT. Contributed reagents/materials/analysis tools: BO ZT. Wrote the paper: BO LJ ZT. Drew figures: LJ ZT.

## References

1. Barabási AL, Albert R. Emergence of scaling in random networks. Science. 1999; 286(5439):509–512. doi: 10.1126/science.286.5439.509 PMID: 10521342
2. Sachtjen ML, Carreras BA, Lynch VE. Disturbances in a power transmission system. Physical Review E. 2000; 61(5):4877. doi: 10.1103/PhysRevE.61.4877
3. Kinney R, Crucitti P, Albert R, Latora V. Modeling cascading failures in the North American power grid. The European Physical Journal B. 2005; 46(1);101–107. doi: 10.1140/epjb/e2005-00237-9
4. Huang X, Vodenska I, Havlin S, Stanley HE. Cascading failures in bi-partite graphs: model for systemic risk propagation. Scientific Reports. 2013; 3:1219. doi: 10.1038/srep01219 PMID: 23386974
5. Borrvall C, Ebenman B, Jonsson T. Biodiversity lessens the risk of cascading extinction in model food webs. Ecology Letters. 2000; 3(2):131–136. doi: 10.1046/j.1461-0248.2000.00130.x
6. Gao Y, Du W-B, Yan G. Selectively-informed particle swarm optimization. Scientific reports. 2015; 5:9295. doi: 10.1038/srep09295 PMID: 25787315
7. Du W-B, Gao Y, Liu C, Zheng Z, Wang Z. Adequate is better: particle swarm optimization with limited-information. Applied Mathematics and Computation. 2015; 268:832–838. doi: 10.1016/j.amc.2015.06.062
8. Papadopoulos F, Kitsak M, Serrano MÁ, Boguñá M, Krioukov D. Popularity versus similarity in growing networks. Nature. 2012; 489(7417):537–540. doi: 10.1038/nature11459 PMID: 22972194
9. Zhang QM, Lü L, Wang WQ, Zhu YX, Zhou T, et al. Potential theory for directed networks. PloS one. 2013; 8(2):e55437. doi: 10.1371/journal.pone.0055437 PMID: 23408979
10. Zhang QM, Xu XK, Zhu YX, Zhou T. Measuring multiple evolution mechanisms of complex networks. arXiv preprint arXiv:14103519. 2014.
11. Guimerà R, Sales-Pardo M. Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences. 2009; 106(52):22073–22078. doi: 10.1073/pnas.0908366106
12. Lü L, Medo M, Yeung CH, Zhang YC, Zhang ZK, Zhou T. Recommender systems. Physics Reports. 2012; 519(1):1–49. doi: 10.1016/j.physrep.2012.02.006
13. Cannistraci CV, Alanis-Lobato G, Ravasi T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. Scientific reports. 2013; 3. doi: 10.1038/srep01613 PMID: 23563395
14. Westermarck J, Ivaska J, Corthals GL. Identification of protein interactions involved in cellular signaling. Molecular & Cellular Proteomics. 2013; 12(7):1752–1763. doi: 10.1074/mcp.R113.027771
15. Lü L, Zhou T. Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and its Applications. 2011; 390(6):1150–1170. doi: 10.1016/j.physa.2010.11.027
16. Wang P, Xu B, Wu Y, Zhou X. Link prediction in social networks: the state-of-the-art. Science China Information Sciences. 2015; 58(1):1–38.
17. Wang WQ, Zhang QM, Zhou T. Evaluating network models: A likelihood analysis. EPL (Europhysics Letters). 2012; 98(2):28004. doi: 10.1209/0295-5075/98/28004
18. Lü L, Pan L, Zhou T, Zhang YC, Stanley HE. Toward link predictability of complex networks. Proceedings of the National Academy of Sciences. 2015; 112(8):2325–2330. doi: 10.1073/pnas.1424644112
19. Zhou T, Lü L, Zhang YC. Predicting missing links via local information. Eur Phys J B. 2009; 71(4):623–630. doi: 10.1140/epjb/e2009-00335-8
20. Adamic LA, Adar E. Friends and neighbors on the web. Social networks. 2003; 25(3):211–230. doi: 10.1016/S0378-8733(03)00009-1
21. Clauset A, Moore C, Newman ME. Hierarchical structure and the prediction of missing links in networks. Nature. 2008; 453(7191):98–101. doi: 10.1038/nature06830 PMID: 18451861
22. Restrepo JG, Ott E, Hunt BR. Characterizing the dynamical importance of network nodes and links. Physical Review Letters. 2006; 97(9):094102. doi: 10.1103/PhysRevLett.97.094102 PMID: 17026366
23. Tan F, Xia Y, Zhu B. Link Prediction in Complex Networks: A Mutual Information Perspective. PLoS ONE. 2014; 9:e107056. doi: 10.1371/journal.pone.0107056 PMID: 25207920

24. Zhu B, Xia Y. An information-theoretic model for link prediction in complex networks. Scientific reports. 2015; 5:13707. doi: 10.1038/srep13707 PMID: 26335758

25. Marbach D, Costello JC, Küffner R, Vega NM, Prill RJ, Camacho DM, et al. Wisdom of crowds for robust gene network inference. Nature methods. 2012; 9(8):796–804. doi: 10.1038/nmeth.2016 PMID: 22796662

26. Zachary WW. An information flow model for conflict and fission in small groups. Journal of anthropological research. 1977;p. 452–473.

27. Ulanowicz RE, DeAngelis DL. Network analysis of trophic dynamics in south florida ecosystems. US Geological Survey Program on the South Florida Ecosystem. 2005; 114.

28. Gleiser PM, Danon L. Community structure in jazz. Advances in complex systems. 2003; 6(04):565–573. doi: 10.1142/S0219525903001067

29. Watts DJ, Strogatz SH. Collective dynamics of'small-world' networks. Nature. 1998; 393(6684):440–442. doi: 10.1038/30918 PMID: 9623998

30. Vladimir B, Andrej M. Pajek datasets; 2006. http://vlado.fmf.uni-lj.si/pub/networks/data/

31. Duch J, Arenas A. Community detection in complex networks using extremal optimization. Physical review E. 2005; 72(2):027104. doi: 10.1103/PhysRevE.72.027104

32. Adamic LA, Glance N. The political blogosphere and the 2004 US election: divided they blog. In: Proceedings of the 3rd international workshop on Link discovery. ACM; 2005. p. 36–43.

33. Von Mering C, Krause R, Snel B, Cornell M, Oliver SG, Fields S, et al. Comparative assessment of large-scale data sets of protein–protein interactions. Nature. 2002; 417(6887):399–403. doi: 10.1038/nature750 PMID: 12000970

34. Vladimir B. Pajek datasets; Accessed 2015 Aug 19. http://vlado.fmf.uni-lj.si/pub/networks/data/mix/mixed.htm

35. Spring N, Mahajan R, Wetherall D. Measuring ISP topologies with Rocketfuel. In: ACM SIGCOMM Computer Communication Review. vol. 32. ACM; 2002. p. 133–145.

36. Leskovec J, Huttenlocher D, Kleinberg J. Predicting positive and negative links in online social networks. In: Proceedings of the 19th international conference on World wide web. ACM; 2010. p. 641–650.

37. Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM; 2010. p. 1361–1370.

38. Newman ME. Clustering and preferential attachment in growing networks. Phys Rev E. 2001; 64 (2):025102. doi: 10.1103/PhysRevE.64.025102

39. Lü L, Jin CH, Zhou T. Similarity index based on local paths for link prediction of complex networks. Physical Review E. 2009; 80(4):046122. doi: 10.1103/PhysRevE.80.046122

40. Katz L. A new status index derived from sociometric analysis. Psychometrika. 1953; 18(1):39–43. doi: 10.1007/BF02289026

41. Chung FR. Spectral graph theory. vol. 92. American Mathematical Soc.; 1997.

42. Horn RA, Johnson CR. Matrix analysis. Cambridge university press; 2012.

43. Burt RS. Positions in networks. Social forces. 1976; 55(1):93–122. doi: 10.1093/sf/55.1.93

44. Scott J. Social network analysis. Sage; 2012.

45. Zhao J., Miao L., Yang J., Fang H., Zhang Q.-M., Nie M., et al. Prediction of Links and Weights in Networks by Reliable Routes Scientific reports. 2015;5:12261.