Physica A 427 (2015) 100–112

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Noise-tolerant model selection and parameter estimation for complex networks

Sadegh Aliakbary *, Sadegh Motallebi, Sina Rashidian, Jafar Habibi, Ali Movaghar

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran

HIGHLIGHTS

- We integrated various network features and then applied machine learning algorithms in order to develop a model fitting method for complex networks.
- We presented comprehensive empirical evaluations based on synthesized graphs and case studies for real networks.
- Using distance based (nearest neighbor) classification increases the accuracy of model prediction.
- Supervised machine learning algorithms are effective in developing an accurate model selection and parameter estimation method.

ARTICLE INFO

Article history:
Received 28 September 2014
Received in revised form 14 December 2014
Available online 10 February 2015

Keywords:
Complex networks
Network models
Model selection
Parameter estimation
Machine learning
Social networks

ABSTRACT

Real networks often exhibit nontrivial topological features that do not occur in random graphs. The need for synthesizing realistic networks has resulted in development of various network models. In this paper, we address the problem of selecting and calibrating the model that best fits a given target network. The existing model fitting approaches mostly suffer from sensitivity to network perturbations, lack of the parameter estimation component, dependency on the size of the networks, and low accuracy. To overcome these limitations, we considered a broad range of network features and employed machine learning techniques such as genetic algorithms, distance metric learning, nearest neighbor classification, and artificial neural networks. Our proposed method, which is named ModelFit, outperforms the state-of-the-art baselines with respect to accuracy and noise tolerance in different network datasets.

1. Introduction

Many systems in the real world can be modeled as networks. Real-world networks often exhibit non-trivial topological features which are missing in random graphs [1]. Such graphs, which are called complex networks, appear in social networks, biological networks, and many other domains. In the past two decades, network models are proposed in the literature to synthesize realistic networks [2–6]. The artificial networks are supposed to follow the non-trivial topological features of real networks. For example, Barabási–Albert model [2] synthesizes scale-free networks with long-tail degree distribution, and Watts–Strogatz model [3] generates small-world networks with high clustering. Despite the advances in the field, there is no universal network model suitable for all applications. Thus, a prerequisite of network generation is the step of model

* Corresponding author.
E-mail addresses: aliakbary@ce.sharif.edu (S. Aliakbary), motallebi@ce.sharif.edu (S. Motallebi), rashidian@ce.sharif.edu (S. Rashidian), jhabibi@sharif.edu (J. Habibi), movaghar@sharif.edu (A. Movaghar).

http://dx.doi.org/10.1016/j.physa.2015.02.032
0378-4371/

selection. Additionally, appropriate parameters for the selected model should be estimated. However, model fitting is not a trivial task, and intelligent methods are required for model selection and parameter estimation.

In any application that involves network generation, we should select an appropriate network model and then tune its parameters in order to generate networks with desired properties. Particularly, it is often required to fit a model to a target network instance, or to fit it to a set of target feature values. In this case, the fitted model is supposed to synthesize networks similar to the target network. The "model fitting" process aims at finding the best model being capable of generating networks similar to the target network (*Model Selection*), and then estimating the model parameters which generate the most similar graphs to the target network (*Parameter Estimation*). Applications of network model fitting include simulation of network dynamics [7–9], graph summarization [10,5,11–13], network sampling [14–18], network anonymization [19–22], and network comparison [23,10,24–27,11].

Most of the existing model fitting methods lack the parameter estimation component and only provide a model selection method (e.g., Refs. [28–31]). Additionally, the existing methods are sensitive to network perturbations, and their accuracy drops considerably if we inject noise to the network by making random changes in network edges. However, noise tolerance is an important requirement for model fitting methods. This is because, in most cases, the target network (e.g., a real network) is fully compatible with no network model and shows a level of difference (noise) to any candidate model.

In this paper, we propose an intelligent and noise-tolerant model fitting method for complex networks, which is named *ModelFit*. ModelFit consists of both model selection and parameter estimation components. It utilizes a diverse set of local and global network features, and employs various machine learning algorithms for model selection and parameter estimation. The model selection component of ModelFit is based on genetic algorithms, distance metric learning, and nearest neighbor classification. The parameter estimation is performed by learning an artificial neural network. As a result, ModelFit has become an accurate, robust-to-noise, and size-independent method.

The structure of the paper in the following sections is as follows: Section 2 reviews the literature on the problem of model fitting. In Section 3 we propose an intelligent method for model selection. The comprehensive evaluations of the proposed model selection method are presented in Section 4 along with comparison to baseline methods. In Section 5, a parameter estimation method is proposed and the accuracy of the method is evaluated. Finally, we conclude the paper in Section 8.

2. Related works

The existing model fitting methods follow different approaches and techniques. The set of considered network features is a source of diversity in the existing methods. Although some methods use both local and global network features in model selection [28,13], many existing methods are based on graphlet-counting features [29–31,25,26]. Graphlet counting is an inefficient approach, often accelerated by sampling [29] or approximation algorithms [32,29] which result in accuracy drops [28].

A network model can be regarded as a class of networks, and model selection is actually a classification problem in which the target network is assigned to one of the candidate models. The classification method is another source of difference among the existing model selectors. Some existing model selection methods are based on supervised machine learning algorithms of classification, such as decision tree learning [28–30]. Some other methods are based on lazy learning and distance-based classification [31]. In this approach, the target network is compared with a set of generated networks, and the most similar model is selected. In this process, the model of the target network is selected either based on its average distance to networks of different models (e.g., Ref. [31]) or according to the class of the nearest neighbors (e.g., our proposed method). It is also worth noting that some of the existing model fitting methods are sensitive to the size of the target network (e.g., Ref. [29]), or dependent on network density (e.g., Refs. [28–30]).

3. Model selection

In this section, we describe our proposed method for model selection. Given a target network instance, or its structural features, ModelFit finds the best model that is able to generate similar networks. ModelFit selects the best fitting class (model) based on the distance of the target network to various generated network instances. In the proposed method, a network distance metric is first learned which is able to separate networks of different classes. In this way, a genetic algorithm (GA) is utilized as an intelligent search method in order to find the best network distance metric. The resulting distance function, called NetDistance, is then utilized in nearest neighbor classification of the networks. In this phase, different network instances are generated using various network models, and among them, the most similar instances to the target network are selected. This process describes the well-known "k nearest neighbor (KNN)" classification algorithm. KNN is a simple machine learning algorithm that classifies an object by a majority vote of its k neighbors. Fig. 1 shows the methodology of ModelFit for model selection.

3.1. Network features

In the proposed method, the structural properties of each network instance is translated into a set of numerical features. Various measures are defined in the literature to quantify the structural properties of networks. We utilized a diverse set