Network Completion: Beyond Matrix Completion

Cong Tran[1,2] and Won-Yong Shin[2]
[1]Dankook University, [2]Yonsei University

trancong208@gmail.com, wy.shin@yonsei.ac.kr

Abstract

The problem of recovering a partially observable network in which some nodes and corresponding edges are missing is known as network completion in the literature. We review two network completion algorithms. We also present performance comparison between the two algorithms via experimental evaluation.

Ⅰ. Introduction

In this paper, we study the problem of network completion [1], where we observe a partial sample of a true underlying network and try to infer the remainder. For example, we may raise the following question: how do we estimate the structure of Facebook social network by only observing a part of it via web crawling? More specifically, suppose that there is a complete network, represented by an adjacency matrix A_T, from which some nodes and corresponding edges are missing. We then aim to fill in the missing part of A_T based on the information of the observable entries as illustrated in Fig. 1. In this context, network completion is related to the problem of matrix completion [2]. However, while existing matrix completion methods exploit information from observable entries in the same row or column, such information is not available when there is a missing node in the network with no associated edges.

Ⅱ. Network Completion

As the most influential study, KronEM, was suggested by Kim and Leskovec [1] to solve the network completion problem by applying the expectation maximization (EM) algorithm. More specifically, the observed part of the network is used to fit a Kronecker graph model. Then, two following iterative steps are executed: 1) the missing part of the network is sampled from the fitted model and 2) the parameters of the model are re-estimated according to new samples.

Recently, DeepNC was introduced for inferring the missing parts of an underlying network based on a deep generative model of graphs [3]. Specifically, the method first learns a likelihood over edges via an autoregressive generative model, and then identifies the graph that maximizes the learned likelihood conditioned on the observable graph topology. This procedure is executed via a computationally efficient algorithm, where individual nodes and corresponding edges that maximize the likelihood are consecutively generated in a greedy manner.

Ⅲ. Experimental Results

To evaluate the performance of network completion, we adopt the graph edit distance (GED), which quantifies the degree of agreement between the recovered network and the original one without deleting any nodes or edges. Fig. 1 shows the comparison of two network completion algorithms, including DeepNC and KronEM, using GED via two synthetic datasets and two real-world ones, whose detailed statistics can be found in [3]. The result demonstrates the superior of DeepNC over KronEM.

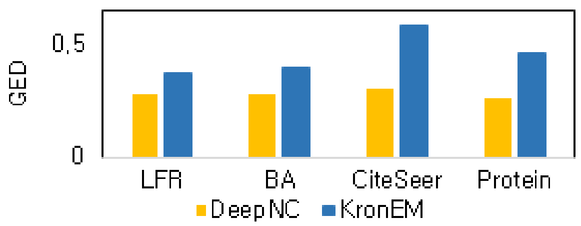

Figure 1. Performance comparison in terms of GED

ACKNOWLEDGMENT

This work was supported by the Republic of Korea's MSIT (Ministry of Science and ICT), under the High-Potential Individuals Global Training Program (No. 2020-0-01463) supervised by the IITP (Institute of Information and Communications Technology Planning Evaluation) and by the Yonsei University Research Fund of 2020 (2020-22-0101).

REFENCES

[1] M. Kim and J. Leskovec, "The network completion problem: Inferring missing nodes and edges in networks," in *Proc. SIAM Int. Conf. Data Mining (SDM '11)*, Apr. 2011, pp. 47-58.
[2] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 1-49, 2008.
[3] C. Tran, W.-Y. Shin, A. Spitz, and M. Gertz, "DeepNC: Deep Generative Network Completion," preprint. [Online]. Available: https://arxiv.org/abs/1907.07381.