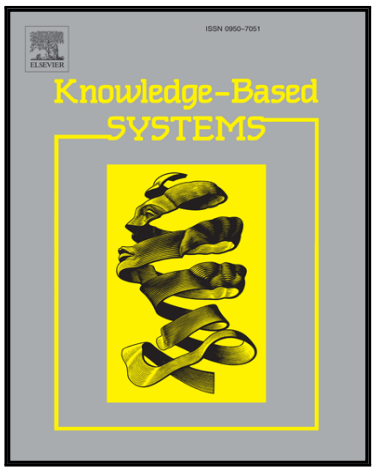

Graph Embedding Techniques, Applications, and Performance: A Survey

Palash Goyal, Emilio Ferrara

PII: S0950-7051(18)30154-0
DOI: 10.1016/j.knosys.2018.03.022
Reference: KNOSYS 4271

To appear in: *Knowledge-Based Systems*

Received date: 10 December 2017
Revised date: 13 March 2018
Accepted date: 15 March 2018

Please cite this article as: Palash Goyal, Emilio Ferrara, Graph Embedding Techniques, Applications, and Performance: A Survey, *Knowledge-Based Systems* (2018), doi: 10.1016/j.knosys.2018.03.022

Graph Embedding Techniques, Applications, and Performance: A Survey

Palash Goyal and Emilio Ferrara

University of Southern California, Information Sciences Institute
4676 Admiralty Way, Suite 1001. Marina del Rey, CA. 90292, USA

Abstract

Graphs, such as social networks, word co-occurrence networks, and communication networks, occur naturally in various real-world applications. Analyzing them yields insight into the structure of society, language, and different patterns of communication. Many approaches have been proposed to perform the analysis. Recently, methods which use the representation of graph nodes in vector space have gained traction from the research community. In this survey, we provide a comprehensive and structured analysis of various graph embedding techniques proposed in the literature. We first introduce the embedding task and its challenges such as scalability, choice of dimensionality, and features to be preserved, and their possible solutions. We then present three categories of approaches based on factorization methods, random walks, and deep learning, with examples of representative algorithms in each category and analysis of their performance on various tasks. We evaluate these state-of-the-art methods on a few common datasets and compare their performance against one another. Our analysis concludes by suggesting some potential applications and future directions. We finally present the open-source Python library we developed, named GEM (*Graph Embedding Methods*, available at `https://github.com/palash1992/GEM`), which provides all presented algorithms within a unified interface to foster and facilitate research on the topic.

Keywords: Graph embedding techniques, Graph embedding applications, Python Graph Embedding Methods GEM Library

1. Introduction

Graph analysis has been attracting increasing attention in the recent years due the ubiquity of networks in the real world. Graphs (a.k.a. networks) have been used to denote information in various areas including biology (Protein-Protein interaction networks)[1], social sciences (friendship networks)[2] and linguistics (word co-occurrence networks)[3]. Modeling the interactions between entities as graphs has enabled researchers to understand the various network systems in a systematic manner[4]. For example, social networks have been used for applications like friendship or content recommendation, as well as for advertisement [5]. Graph analytic tasks can be broadly abstracted into the following four categories: *(a)* node classification[6], *(b)* link prediction[5], *(c)* clustering[7], and *(d)* visualization[8]. Node classification aims at determining the label of nodes (a.k.a. vertices) based on other labeled nodes and the topology of the network. Link prediction refers to the task of predicting missing links or links that are likely to occur in the future. Clustering is used to find subsets of similar nodes and group them together; finally, visualization helps in providing insights into the structure of the network.

In the past few decades, many methods have been proposed for the tasks defined above. For node classification, there are broadly two categories of approaches — methods which use random walks to propagate the labels[9, 10], and methods which extract features from nodes and apply classifiers on them[11, 12]. Approaches for link prediction include similarity based methods[13, 14], maximum likelihood models[15, 16], and probabilistic models[17, 18]. Clustering methods include attribute based models[19] and methods which directly maximize (resp., minimize) the inter-cluster (resp., intra-cluster) distances[7, 20]. This survey will provide a taxonomy that captures these application domains and the existing strategies.

Typically, a model defined to solve graph-based problems either operates on the original graph adjacency matrix or on a derived vector space. Recently, the methods based on representing networks in vector space, while preserving their properties, have become widely popular[21, 22, 23]. Obtaining such an embedding is useful in the tasks defined above.[1] The embeddings are input as features to a model and the parameters are learned based on the training data. This obviates the need for complex classification models which are applied directly on the graph.

1.1. Challenges

Obtaining a vector representation of each node of a graph is inherently difficult and poses several challenges which have been driving research in this field:

(i) Choice of property: A "good" vector representation of nodes should preserve the structure of the graph and the connection between individual nodes. The first challenge is choosing

[1] The term *graph embedding* has been used in the literature in two ways: to represent an entire graph in vector space, or to represent each individual node in vector space. In this paper, we use the latter definition since such representations can be used for tasks like node classification, differently from the former representation.