MINING STATIC GRAPHS

Marek Kadlčík, Andrej Kubanda, Petra Krátká, Adam Hájek, Matúš Šikyňa

Masaryk University, Botanická 68a, 602 00
{485294,468985,484070,485410,485591}@mail.muni.cz

Introduction

Graph is a data structure containing a set of objects (called nodes or vertices) and their relations (called edges). Graphs are a useful representation for a variety of data, such as social networks, websites with hyperlinks, computer networks, city maps or molecules.
In our work, we give an overview of data mining and machine learning methods for processing static graph-based data. Tasks that can be solved using the described techniques include regression and classification of nodes, edges or graphs, finding similar graphs or nodes, creating meaningful graph embeddings and discovering frequent patterns in data.

Frequent subgraph mining [1]

Subgraphs that frequently occur in one or multiple graphs in the graph database might represent an essential characteristic of the source data. For example, consider a set of graphs representing the chemical structure of drugs developed for treating a specific disease. If some subgraph in the chemical structure of those drugs occurs frequently, it might be a good candidate for the essential chemical structure needed to cure the disease.
There are two main categories of algorithms for subgraph mining in static graph databases: Apriori-based or BFS algorithms (FSG, AGM), and Depth-first search algorithms (gSpan, MoFa or MoSS, FFSM).
Frequent subgraph mining is widely used in many different domains. The first algorithms were mainly used for the chemo-informatics domain problems, but they were quickly generalised to various applications like bioinformatics, social networks, computer vision or security.

Graph Kernels [2]

For tasks typical for graphs, we are often interested in quantifying similarities between structures such as molecules, complex proteins, or communities on social networks. This similarity can be computed on a pair of structures and expressed as a real number using so-called kernels. Subsequently, graph kernel methods are machine learning techniques that deal with finding suitable and computationally efficient graph kernels that capture the semantics of graph substructures. Consequently, many different graph kernel families have been developed based on different approaches, such as neighborhood aggregation, subgraph patterns, assignment and matching of components, walks and paths in graph traversals, and others, including deep learning inspired techniques.

DeepWalk [3]

DeepWalk is an algorithm, that learns latent representations of a graph's nodes, capturing their neighborhood similarity and community membership. These latent representations encode social relations in a continuous vector space with a relatively small number of dimensions, which is easily exploitable by statistical models such as logistic regression.
To compute the information about local community structures and structural regularities of the graph, the algorithm utilises a randomised path traversing technique, i.e. random walk.
DeepWalk takes the set of random walks as the corpus and the nodes as the vocabulary. The algorithm then uses the SkipGram language model to determine sequences of vertices for a given vertex. The algorithm uses Hierarchical Softmax to approximate probability distribution and to speed up the training time of the latent representations.

Graph neural networks [6, 5]

In recent decades, artificial neural networks have been successfully applied to machine learning problems on various data. Standard multi-layer perceptrons operate on fixed-sized input vectors, convolutional neural networks are commonly used for processing images and recurrent neural networks work with sequences. Graph neural networks (GNNs) are a recent class of neural architectures that can natively process graph-based data.

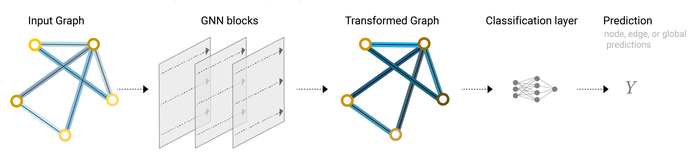

Fig. 1: A scheme of graph-level classification using GNN. Source: [4]

The core building block of GNNs is message passing. It is a trainable function represented by feedforward NN that propagates information on the input graph. How it aggregates data from node's neighbourhood and how it is applied is what distinguishes GNN architectures (RGNNs, CGNNs, GATs). Message passing creates latent representation of nodes (or edges) that can be used for predictions. For whole-graph prediction, graph global pooling needs to be used.

In our work, we discuss differences between architectures, give overview on global pooling layers, and mention application areas and open-source tools. GNNs have been successfully applied to traffic predictions, molecular data, social or citation networks, processing of structured documents, visual scene understanding, processing point clouds and recommender systems.

Explainability of GNNs [7]

A considerable disadvantage of neural networks is the so-called black-box problem. We know the inputs and outputs of a network but we can hardly interpret the reasoning behind the prediction, debug it, or explain it to end users.
We mostly focus on methods for instance-level explanations in our work. Gradient-based techniques visualize to which parts of input is prediction sensitive. Perturbation methods serve similar purpose, but use masking instead of gradients. Surrogate methods train a simpler interpretable model that agrees with the prediction. Decomposition methods directly inspect model parameters and try to expose the relationship between inputs and outputs.

References

[1] Ali Jazayeri and Chris Yang. "Frequent Subgraph Mining Algorithms in Static and Temporal Graph-Transaction Settings: A Survey". In: *IEEE Transactions on Big Data* (2021), pp. 1–1. DOI: https://doi.org/10.1109/TBDATA.2021.3072001.

[2] Nils M. Kriege, Fredrik D. Johansson, and Christopher Morris. "A Survey on Graph Kernels". In: *CoRR* abs/1903.11835 (2019). arXiv: 1903.11835. URL: http://arxiv.org/abs/1903.11835.

[3] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "DeepWalk". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Aug. 2014. DOI: 10.1145/2623330.2623732. URL: https://doi.org/10.1145%2F2623330.2623732.

[4] Benjamin Sanchez-Lengeling et al. "A Gentle Introduction to Graph Neural Networks". In: *Distill* (2021). https://distill.pub/2021/gnn-intro. DOI: 10.23915/distill.00033.

[5] Petar Veličković et al. *Graph Attention Networks*. 2017. DOI: 10.48550/ARXIV.1710.10903. URL: https://arxiv.org/abs/1710.10903.

[6] Zonghan Wu et al. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (Jan. 2021), pp. 4–24. DOI: 10.1109/tnnls.2020.2978386. URL: https://doi.org/10.1109%5C%2Ftnnls.2020.2978386.

[7] Hao Yuan et al. "Explainability in Graph Neural Networks: A Taxonomic Survey". In: *ArXiv* abs/2012.15445 (2020).