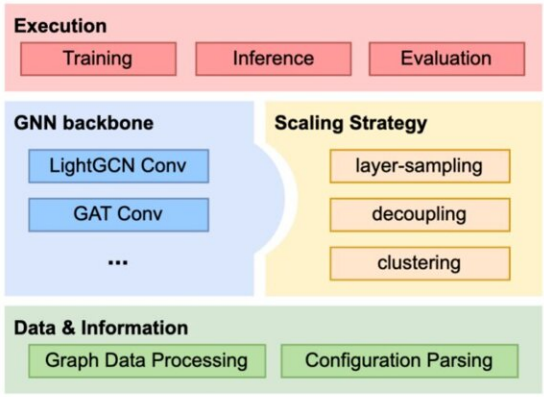
Team proposes Python-based library for large-scale graph neural network recommendations

April 1 2024



(a) Overall framework of the XGCN library

```
config = parse_arguments()              # parse arguments
model = XGCN.create_model(config)       # model initialization
model.fit()                             # model training
test_results = model.test()             # model testing
# infer top-k recommendations
score, topk_node = model.infer_topk(k=100, src=5)
```

(b) Usage example

XGCN's overall framework and usage example. Credit: *Frontiers of Computer Science* (2024). DOI: 10.1007/s11704-024-3803-z

Graph neural networks (GNNs) have gained widespread adoption in recommendation systems. When it comes to processing large graphs, GNNs may encounter the scalability issue stemming from their multi-layer message-passing operations. Consequently, scaling GNNs has emerged as a crucial research area in recent years, with numerous scaling strategies being proposed.

To promote the study of GNN recommendations, a number of open-

source recommendation libraries have incorporated GNNs as a key model category. However, when dealing with large graphs, most of the existing libraries still have two limitations.

The first one is inadequate consideration of scaling strategies. Most libraries only implement specific GNN algorithms without taking scaling strategies into account. The second limitation is the lack of large graph processing optimizations. Many official model implementations overlook certain coding details, leading to scaling issues when processing large graphs.

To solve these problems, a research team led by Hong Huang and Hai Jin published their new research on 14 March 2024 in *Frontiers of Computer Science*.

The team proposed a Python-based library named XGCN to help users quickly build and run large-scale GNNs in a single-machine environment. The library supports various scaling strategies, offers optimized implementations for large graphs, and has an easy-to-use interface for running and development.

Specifically, XGCN includes 16 embedding models in total, covering a broad spectrum of types: from shallow models to common GNNs with three kinds of mainstream scaling strategies: layer-sampling-based, decoupling-based, and clustering-based methods. It incorporates optimizations in models' implementation, such as a set of Numba-accelerated operation functions, making them more suitable for large graphs. Detailed documents for usage guidance and model running scripts are also provided.

Experiments are performed to evaluate several different implementations of a classic graph convolution operation by using datasets of different scales, ranging from 0.15 million nodes to 3 million

nodes. It is observed that existing implementations tend to experience out-of-memory issues, while XGCN consistently showcases superior time and memory efficiency.

More information: Xiran Song et al, XGCN: a library for large-scale graph neural network recommendations, *Frontiers of Computer Science* (2024). DOI: 10.1007/s11704-024-3803-z

Provided by Frontiers Journals

Citation: Team proposes Python-based library for large-scale graph neural network recommendations (2024, April 1) retrieved 2 May 2024 from https://techxplore.com/news/2024-04-team-python-based-library-large.html

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.