CS150-02 Final Presentation

CallGraphNet: A GNN Approach for Call Graph Embedding from Trace Data

(Linda) Chuyi Zhao
05/05/2023

Outline ~10p, 7min

- Intro - what I did: Call Graph embedding extraction for similarity comparison and microservice pattern analysis
- Related work
- BG - Alibaba dataset & reconstructed trace -> call graph
- Method
  - Baseline: directed graph edit distance
  - Loss: Earth Mover's Distance (EMD) or Wasserstein distance.
  - Overview: graph classification
- Experiments
  - Overview: dataset split, train, test, result
- Evaluation/ Contribution/ Conclusion (Compared to baseline)
  - Visualization available
  - Transductive
  - Time and space efficiency

Outline

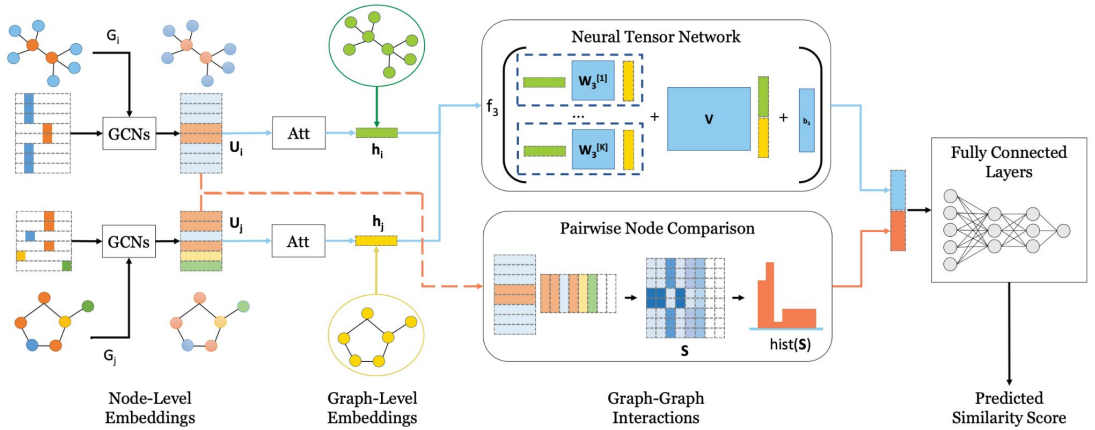
- Introduction
- Related work
- Background
- Methodology
- Experiment
- Evaluation/ Contributions

Introduction

- Trace data
  - Generated during the execution of distributed systems, containing performance information of microservices within large and complex service clusters
  - One Trace - one user’s request, consisting of “calls/ invocations” and “returns”
- Call Graph
  - Constructed from a set of calls, depending on the upstream microservice (UM) and downstream microservice (DM) of each call within a trace
- Call Graph Embeddings
  - Application: program analysis, vulnerability/ anomaly detection, code similarity detection, and Recommender systems

Related Work

- Categorizing and differencing system behaviours
  - Clustering sets of request graphs, to further explore automated problem diagnosis in complex distributed systems
- Graph Similarity based on embeddings
  - SimGNN - predicts a pairwise similarity score based on node-level embeddings and graph-level embeddings



Background

Dataset: Alibaba Trace Program - `cluster-trace-microservices-v2021`

- contain the detailed runtime metrics of nearly 200,000 microservices
- collected from Alibaba production clusters
  - 10,000 bare-metal nodes
  - during 12h in 2021
    - 5min x 145 files
- Reconstruct call graphs
  - Nearly 100,000 x 145 files

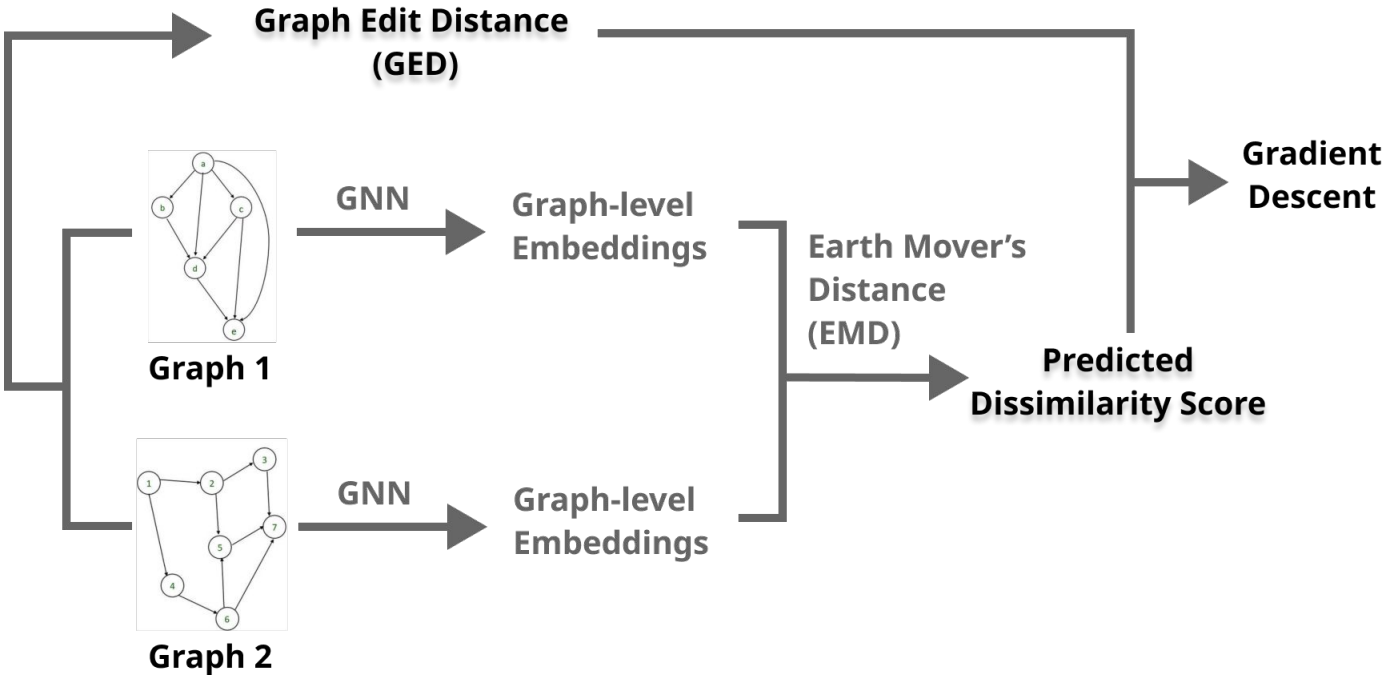
MS_CallGraph_Table:

columns	Example Entry
timestamp	16397576
traceid	015101cd15919399974329000e
rpcid	0.1.1.2.50
um	35114acfb54c54fb9618f23cd28bbc57c765f597df140977d7030dcc52775ed4
rpctype	rpc
interface	af42b5e3e0eb334d38619733586d78d1414f6549f24d31b39a5294454638bc59
dm	b65fdc9bfef6b4974c3e90e1ec7b92d30e639789da5a78c1d4685857e19c75a0
rt	13

clusterdata/cluster-trace-microservices-v2021 at master · alibaba/clusterdata (github.com)

Method - Overview

- Overview:



Method - Ground Truth Construction

Graph edit distance (GED) is a measure of similarity (or dissimilarity) between two graphs. It is defined as the minimum cost of a sequence of graph edit operations required to transform one graph into another. The basic graph edit operations include:

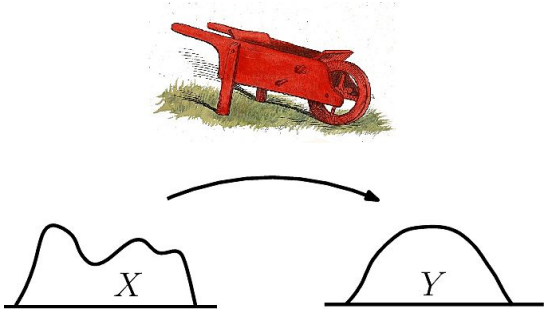
1. Node insertion
2. Node deletion
3. Node substitution
4. Edge insertion
5. Edge deletion
6. Edge substitution

$$GED(g_1, g_2) = \min_{(e_1,\ldots,e_k)\in\mathcal{P}(g_1,g_2)} \sum_{i=1}^{k} c(e_i)$$

Method - Predicted Dissimilarity

Earth Mover's Distance (EMD) or **Wasserstein distance**, is a measure of similarity between two probability distributions.

- Based on *optimal transport* – find the minimum "work" required to transform one distribution into another.
- Like the minimum amount of "earth" that must be moved from one distribution to another, with the cost of moving each unit of "earth" proportional to the distance it is moved.
- Use EMD to calculate the distance between graph embeddings
    - To capture some graph structure information



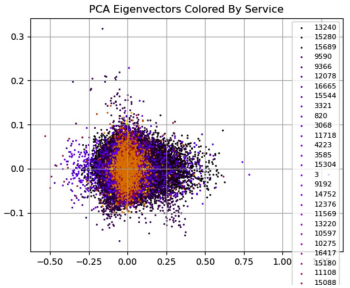
Experiments

- Dataset
  - Due to the time limitation, I used 1,000 call graphs, as 500 pairs with GED labels
    - Training set: testing set = 8:2
  - The learning is unstable, so I used a validation set for model selection and avoiding overfitting
    - Training set: validation set = 8:2
- Results
  - Accuracy – defined as the percentage of error less than a threshold
    - Error = |GED - prediction| / GED x 100%

	Validation accuracy	Testing accuracy
GCNConv	0.6153	0.4694
SAGEConv	0.5714	0.4898
GATConv	0.54762	0.5510
GATv2Conv	0.47619	0.44897

Evaluation/ Contributions

- Visualization available
  - Provides meaningful graphics visualization for further categorized exploration
- Transductive
  - Applies to any new-coming call graph not in the training set
- Time and space efficiency



	GED	CallGraphNet
Time	Nearly 50% takes 10min+ Mean - 297.483998s	< 3 min in total for data loading, training, validation, and testing
Space	Have to construct a giant hashmap for each possible pair, O(n*n)	Only need to store pretrained model state dictionary `model.pt` O(1)

References and Acknowledgments

- Chami Lamelas - help with trace data processing and call graph reconstruction

- Characterizing Microservice Dependency and Performance: Alibaba Trace Analysis (Paper Link) (Dataset)
- Categorizing and differencing system behaviours (Paper Link)
- SimGNN: A Neural Network Approach to Fast Graph Similarity Computation (Paper Link)
- Graph Edit Distance (Website), Earth Mover's Distance (Website) from Wikipedia

Thank you!