CS 744: DISTRIBUTED DGL

Shivaram Venkataraman

Fall 2022

ADMINISTRIVIA

- Midterm grades out!
- Regrade requests (check question numbers)
  - Thu: After class, Roger's OH
  - Mon: Shivaram's OH, Roger's OH
  - Tue: After class

- Course Project: Check in by Nov 23th

EXAMPLE: LINK PREDICTION

Task: Predict potential connections in a social network

[0.25, 0.45, 0.30]

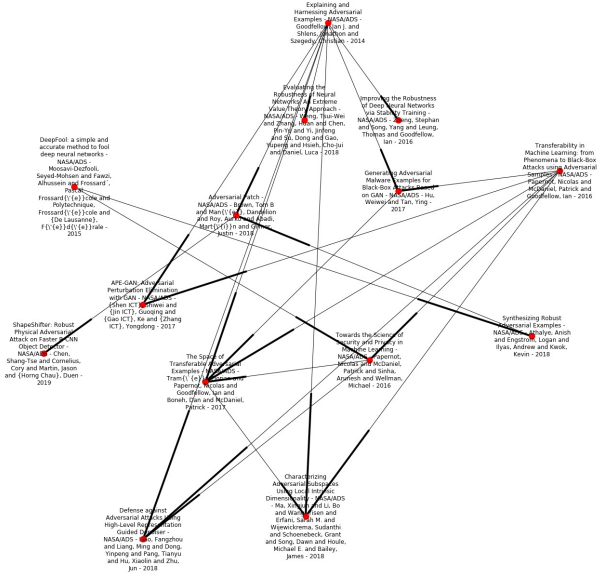
[0.15, 0.85, 0.92]

…

Find K-nearest neighbors

EXAMPLE: NODE CLASSIFICATION

Task: Classify papers in a citation graph by subject area



[0.25, 0.45, 0.30]

[0.15, 0.85, 0.92]

…

physics

math

From: https://github.com/lanstonchu/citation-graph

GRAPH EMBEDDING MODELS: DECODER-ONLY

Loss function

- Maximize score for edges in graph
- Minimize for others (negative edges)

$$\mathcal{L} = \sum_{e \in G} \sum_{e' \in S'_e} \max(f(e) - f(e') + \lambda, 0))$$

BACKGROUND: GRAPH NEURAL NETWORKS (GNN)

Graph Neural Networks: Use neural network to capture neighborhood structure

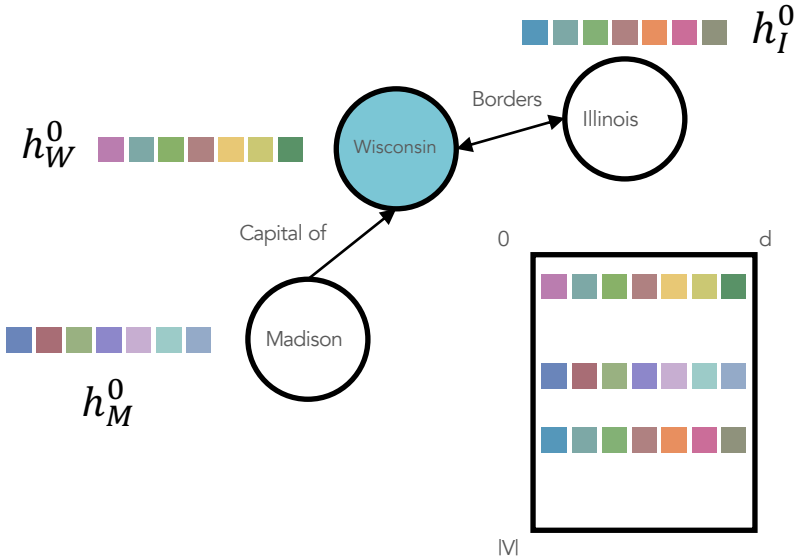
Input: h_i^0 (base node representations)

Model: $h_i^k = AGG(h_i^{k-1}, \{h_u^{k-1} : u \in N_i\})$

N_i one-hop neighborhood of *i*

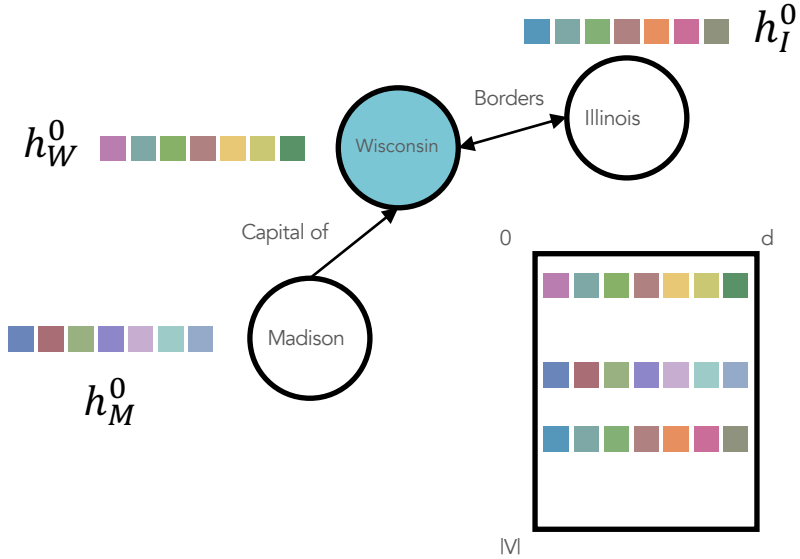
AGG parameterized aggregation func

$h_W^1 = AGG(h_W^0, \{h_I^0, h_M^0\})$



BACKGROUND: GRAPHSAGE

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$$



DISTDGL: DEEP GRAPH LIBRARY

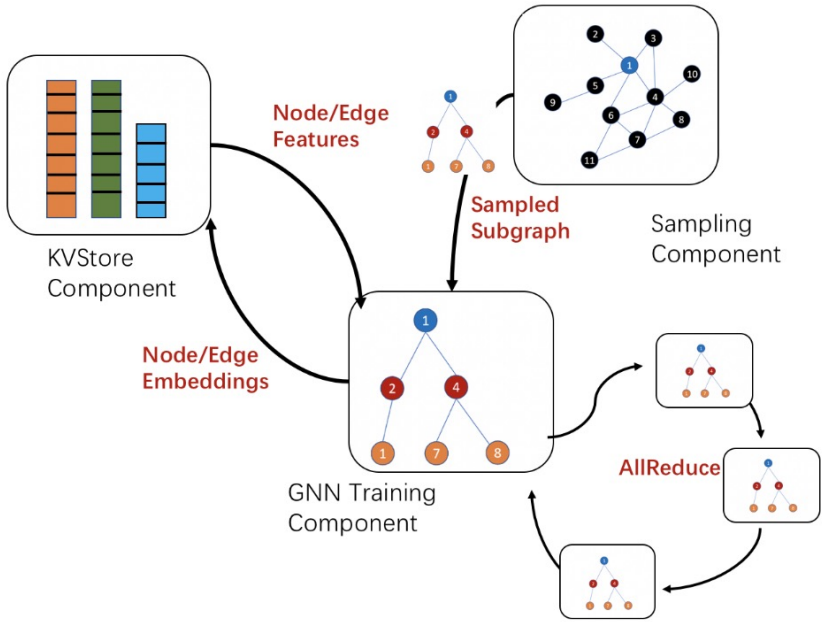
Distributed system for training GNNs

- KVStore
- Mini-batch sampler
- Trainer

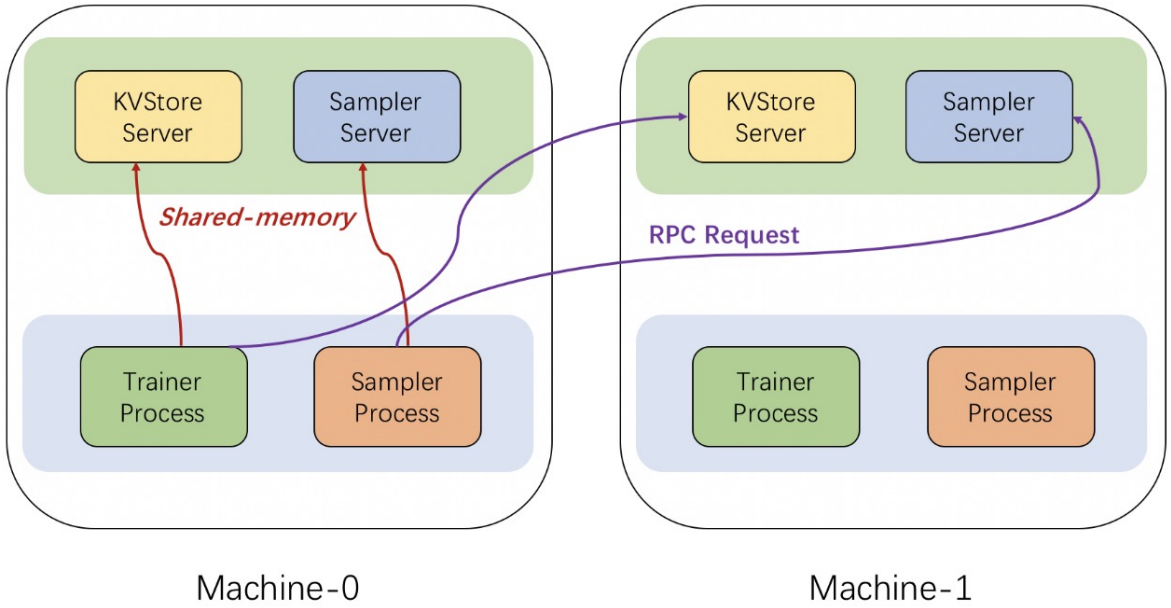
System Design

Key techniques

- Partitioning heterogeneous graphs
- Async mini-batch sampling



DISTDGL SYSTEM SETUP

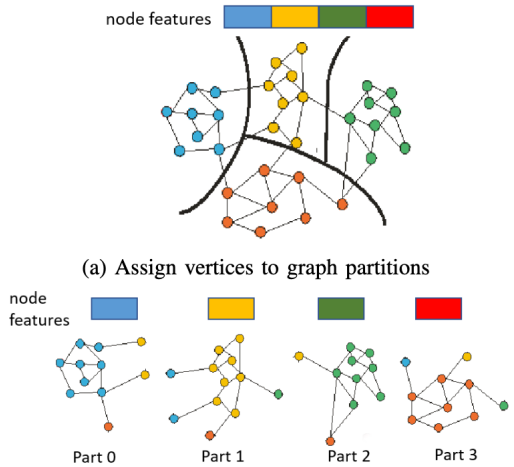


GRAPH PARTITIONING: METIS

Hierarchical METIS

Apply METIS to partition graph across machines

Re-apply METIS to partition within a machine

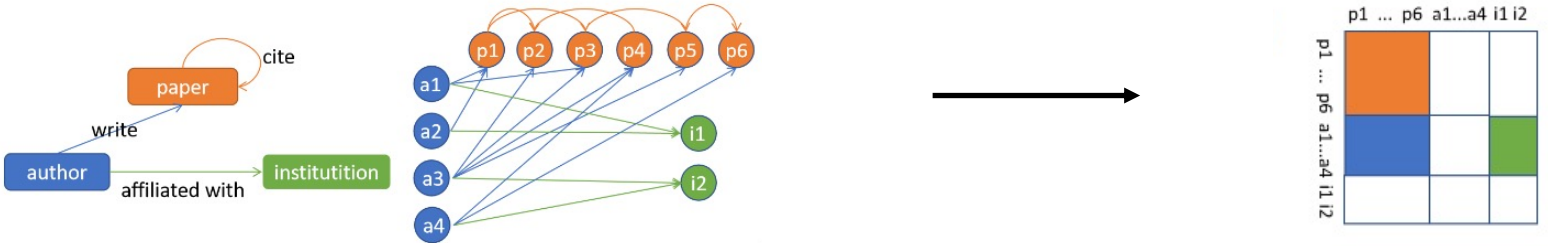


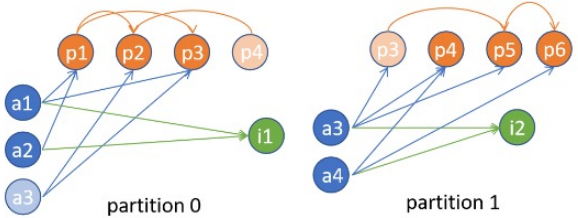
A fast and high quality multilevel scheme for partitioning irregular graphs

George Karypis and Vipin Kumar

SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359 - 392, 1999

HETEROGENEOUS GRAPH PARTITIONING

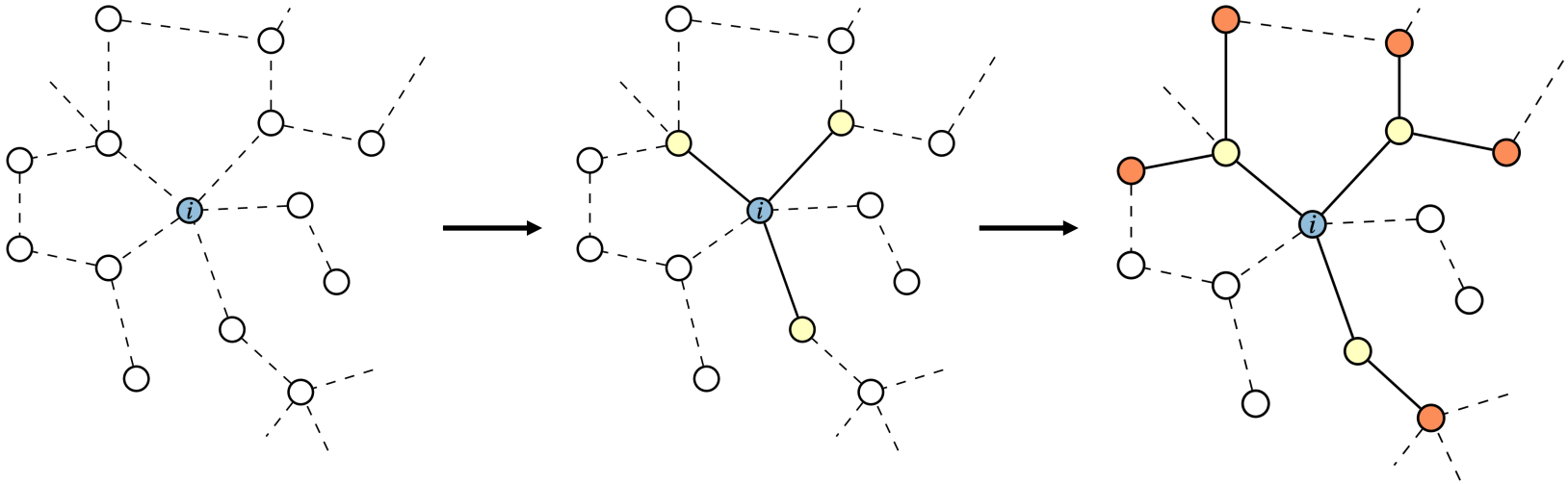


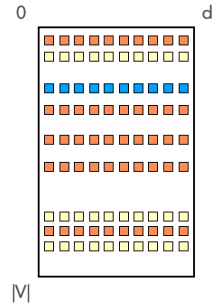


GNN MINI-BATCH PREPARATION

```
for batch in training_examples:
    sample_neighbors(batch)
    load_representations(batch)
    transfer_to_GPU(batch)
    loss = model(batch)
    transfer_to_CPU(batch)
    update_parameters(batch)
```

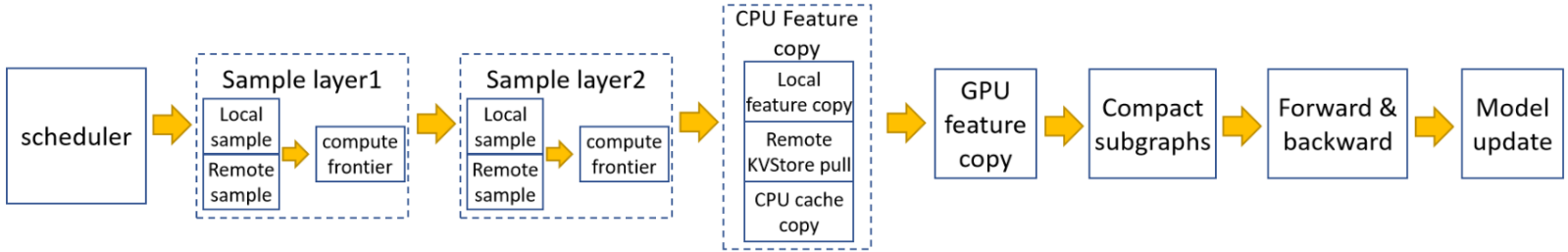
MINIBATCH SAMPLING FOR GNNS



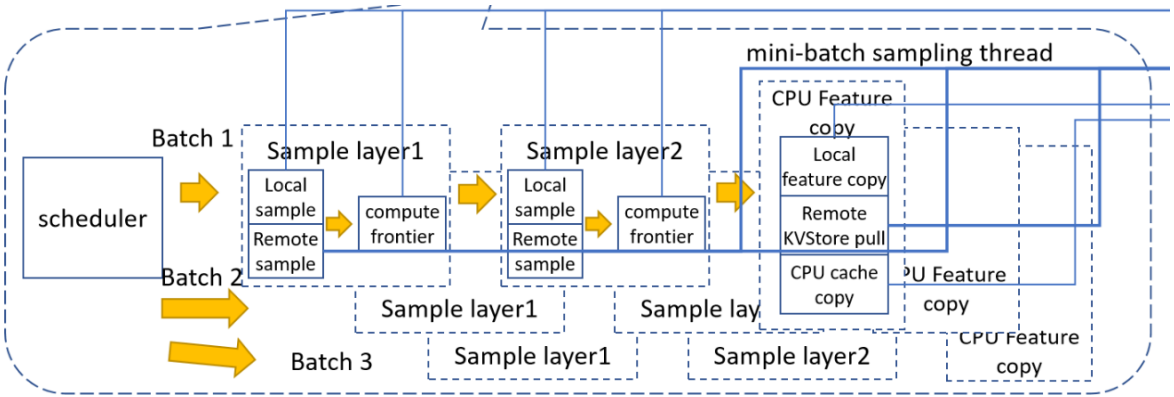


Base representations h^0

DISTDGL PIPELINE



Async mini-batch sampling with sync training



SUMMARY

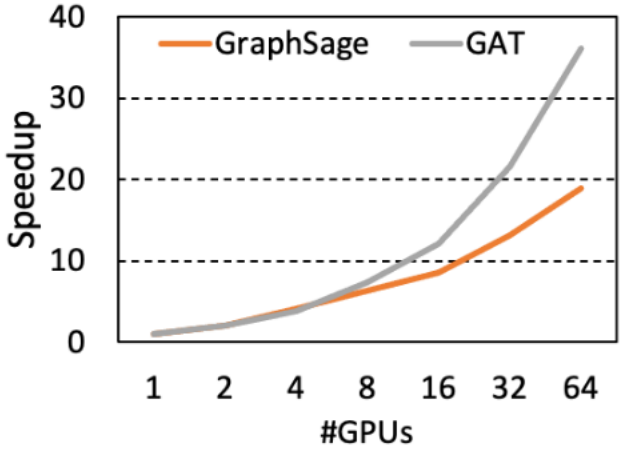
Graph NN: capture the structure of

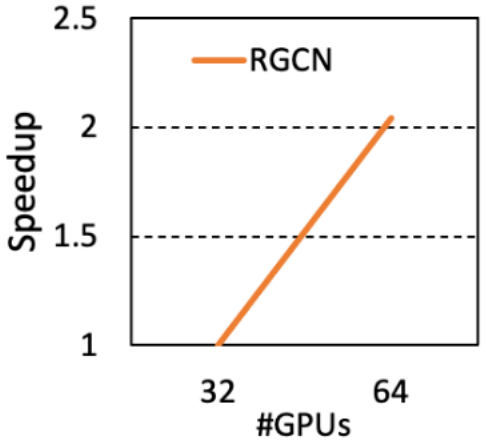
DistDGL: Distributed GNN training

- Partition graphs using METIS
- Pipelining to use CPU, GPU for sampling

DISCUSSION

https://forms.gle/Dp8qtqdpWuoVeys67


40
30
20
10
0
Speedup
GraphSage
GAT
1
2
4
8
16
32
64
#GPUs


2.5
2
1.5
1
Speedup
RGCN
32
64
#GPUs

If you wished to extend the design of Marius to support GNNs, how would you do that? What would be some challenges?

NEXT STEPS

Next class: Serverless computing

Project check-ins by Nov 23th