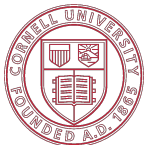
Optimizing Sparse/Graph Kernels with TVM

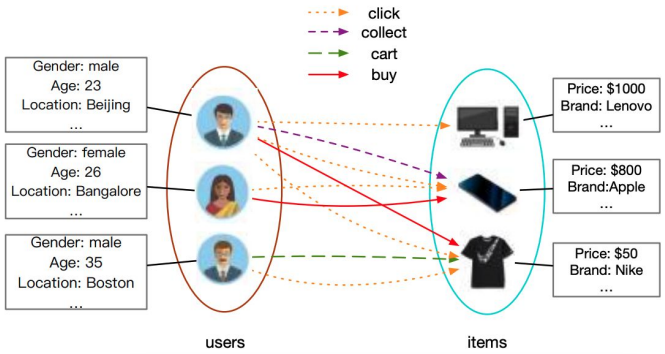
Yuwei Hu
Advisor: Zhiru Zhang
Internship Mentor: Yida Wang



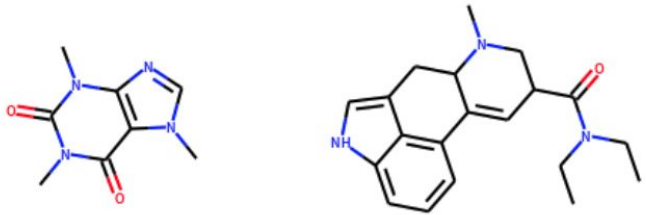
Cornell University

aws

Graph Neural Networks (GNNs) are getting popular



[1]



[2]

Diverse Applications

DeepGraphLibrary [3]

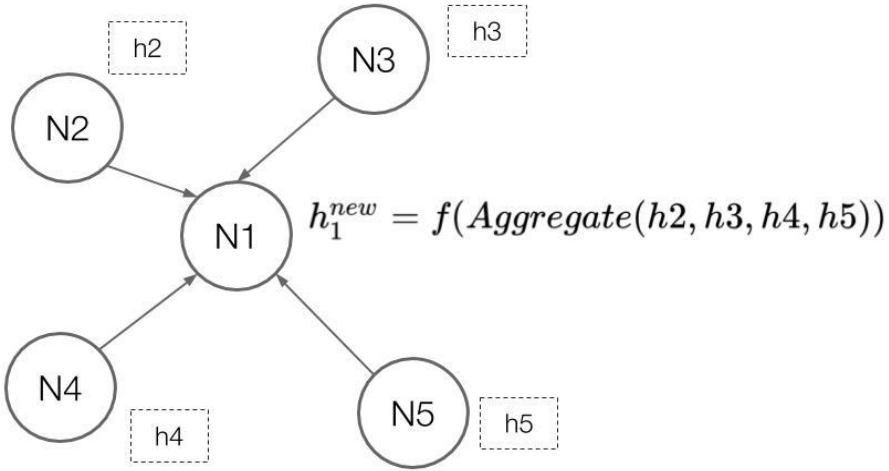
PyTorch geometric [4]

Paddle Graph Learning [5]

Emerging Frameworks

1. AliGraph: A Comprehensive Graph Neural Network Platform
2. Interpolate between two molecules with pre-trained JTNN
3. https://www.dgl.ai
4. https://pytorch-geometric.readthedocs.io
5. https://github.com/PaddlePaddle/PGL

Two Key Kernels in GNNs



$$q_j = W_q \cdot x_j$$
$$k_i = W_k \cdot x_i$$
$$v_i = W_v \cdot x_i$$
$$\text{score} = q_j^T k_i$$

$$w_{ji} = \frac{\exp\{\text{score}_{ji}\}}{\sum_{(k,i)\in E} \exp\{\text{score}_{ki}\}}$$

Message-passing is doing SpMM (sparse-dense matrix multiply)

More precisely, it is SpMM-like if we use a customized aggregation(reduce) function other than sum

Dot-product attention is doing SDDMM (sampled dense-dense matrix multiply)

Challenges

Existing deep learning frameworks have very limited and inflexible support for sparse computation

Existing graph processing frameworks are not aware of the feature dimension

Challenges

Existing deep learning frameworks have very limited and inflexible support for sparse computation

Existing graph processing frameworks are not aware of the feature dimension

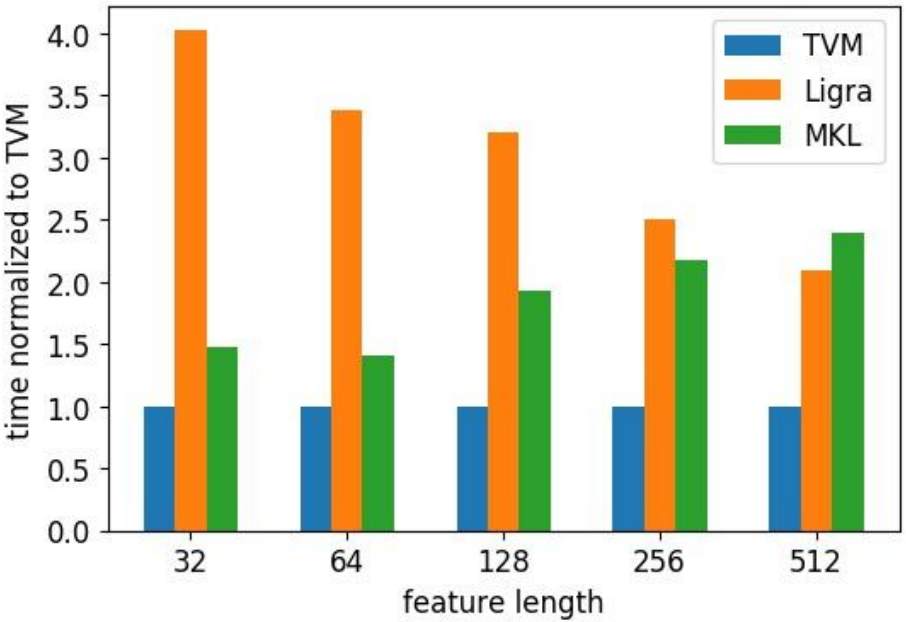
Exploring the Feature Dimension

CPU: feature dimension tiling to improve cache utilization

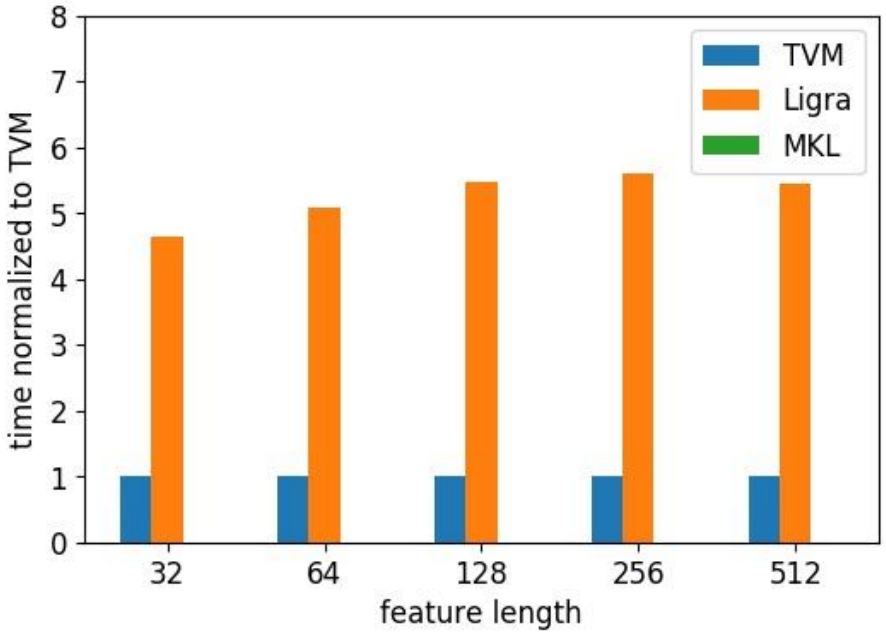
GPU: parallelization strategies specialized for computation patterns

Preliminary Results

Tested on c5.18xlarge instance with 36 cores and 140GB DRAM
Dataset is reddit with 233K vertices and 115M edges



Single-threaded SpMM



Single-threaded SDDMM

Plan

Integration into DGL

- Frontend: message passing programming interface in DGL
- Backend: optimized sparse kernels written in TVM

Plan

Integration into DGL

Frontend: message passing programming interface in DGL

Backend: optimized sparse kernels written in TVM

Native Sparse Support in TVM

Current implementation is using IR builder

Optimization techniques can be abstracted into schedule primitives. For example, we can introduce sparse split.

Ongoing efforts from UW to build an infrastructure for sparse representation and computation: RFC

Credits and Thanks

Jiali Yu @ SJTU, for helping with benchmarking

Andrew Tulloch @ Facebook, for contributing the blocked sparse kernel in TVM, which greatly inspired this work

Zihao Ye @ AWS, for discussing DGL integration

Leyuan Wang @ AWS, for discussing GPU optimizations