Good morning!

CS 744: PYTORCH-BIGGRAPH

Shivaram Venkataraman

Fall 2020

ADMINISTRIVIA

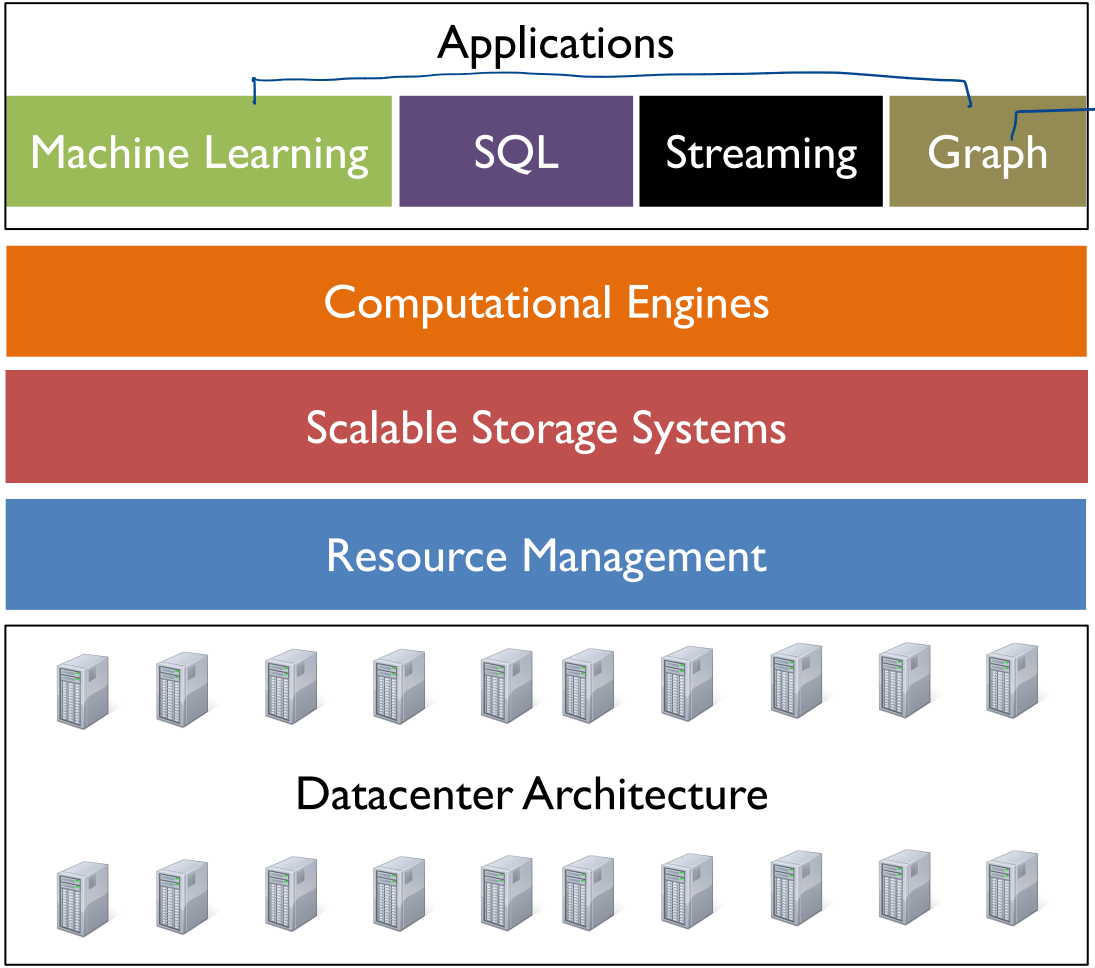
Course Project: Check in by Nov 20th

↳ Instructions for this by tomorrow.

Regrades - a few?

Graphs +

ML



Powergraph

GraphX

Graph

Analytics

GRAPH EMBEDDINGS

Link prediction

≡ Friend recommendation

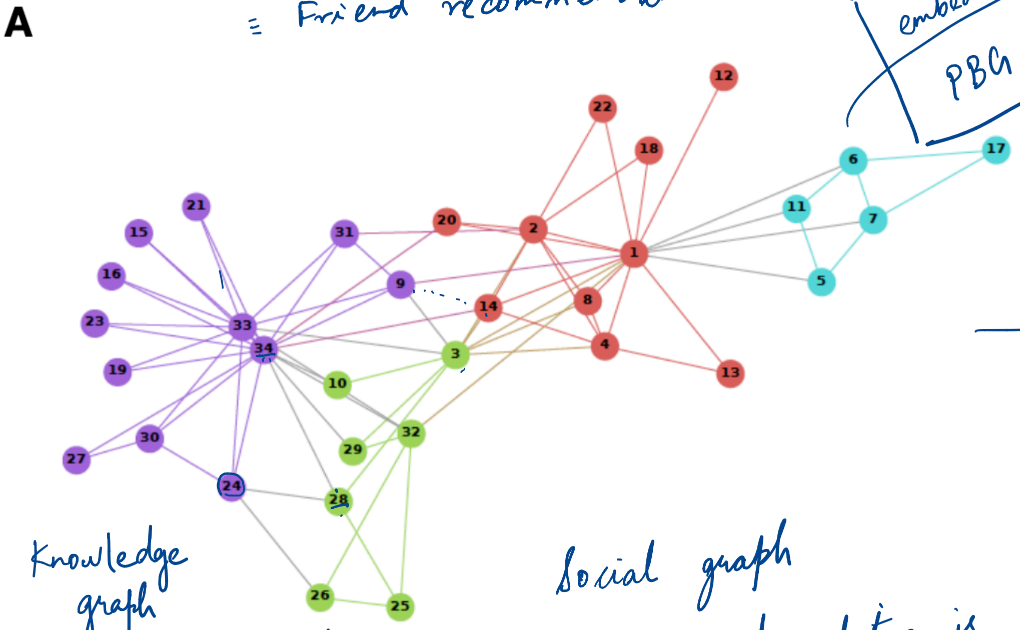
embedding

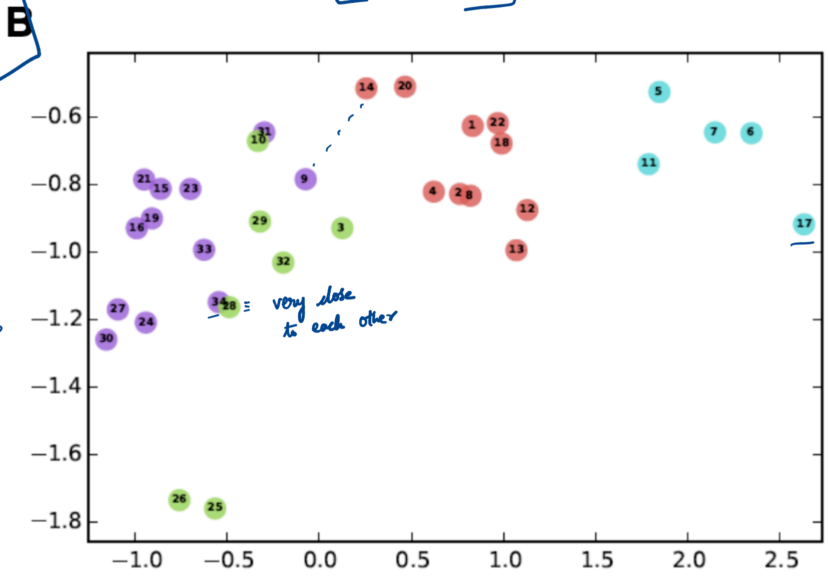
PBG

n data points

dimension - features

ML Method





Knowledge graph

→ different kinds of edges

→ multiple edges

Social graph

≡ friend relation is an edge

≡ can learn 100 dim embedding

2d embedding

(~100 dim is common)

TRAINING GRAPH EMBEDDINGS

initialize random

Vertex	Embedding
12 / Id	<0.0, 0.25>
14	<0.33, 0.5>
16	<0.45, 0.6>
…	…

2d vector

list of edges

Source	Dest
12 / Id	14 / Id
12	18
16	14
…	…

initialize emb. randomly

SGD

sample edges

↳ compute scores, negatives

↳ update embedding

⋮

score function

similarity

embedding!

$$f(\theta_s, \cancel{\theta_r}, \theta_d) = sim\left(g_{(s)}(\theta_s, \cancel{\theta_r}), g_{(d)}(\theta_d, \cancel{\theta_r})\right)$$

sim function = dot product

sim (12, 14) = <0.0, 0.25> dot <0.33, 0.5>

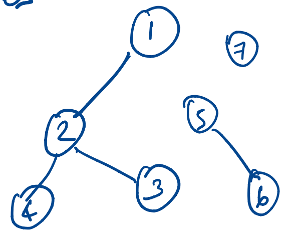
score should be higher for vertices that are "more" connected or closer

NEGATIVE SAMPLING

negatives

$\boxed{1 \to 3}$

$3 \to 4$



positives

$1 \to 2$

$2 \to 3$

$2 \to 4$

- Sample from edges not in the graph!
  → Corrupt positive edges with a new src or dest

- maximize score for true edges / positive
- minimize score for negative edges

Two options

1. According to data distribution (degree)
   ↳ Very rarely see negative edges between rare nodes

Hybrid

2. Uniformly
   ↳ model will tend to predict high degree vertices

→ link prediction
⇒ always recommend very popular users

loss function ↑

regularizer

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

set of negative edges for every +ve edge

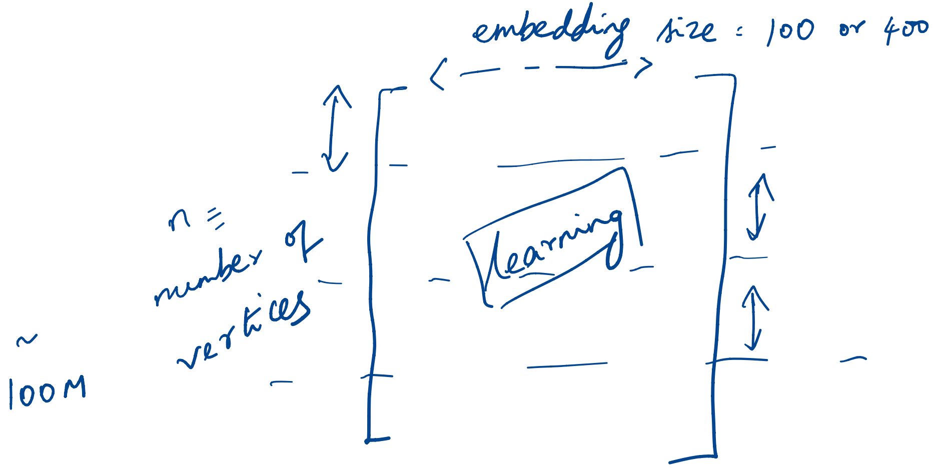
score for a +ve edge

score of a negative edge

SCALING CHALLENGES

Fast enough to embed graphs with 10^{11}–10^{12} edges in a reasonable time

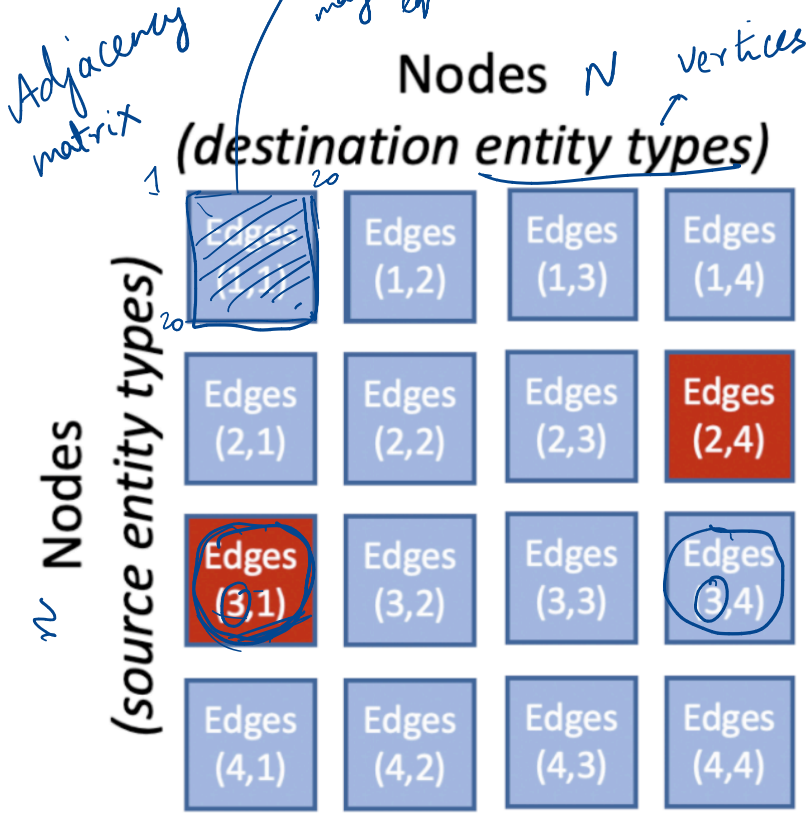
~100 embedding parameters per node → require 800GB of memory!

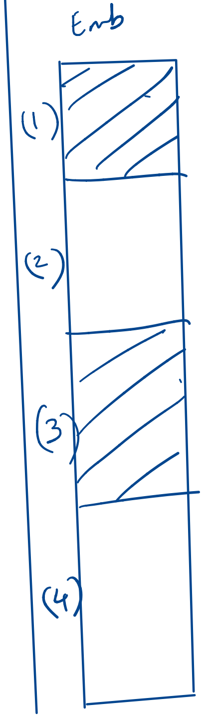


= 800 GB of memory

GRAPH PARTITIONING

num edges per bucket may not be equal!





Consider Edges (3,1) bucket

→ sample some edges from this bucket

→ To compute score, vertex embeddings of vertex (3) and vertex (1)

only these two

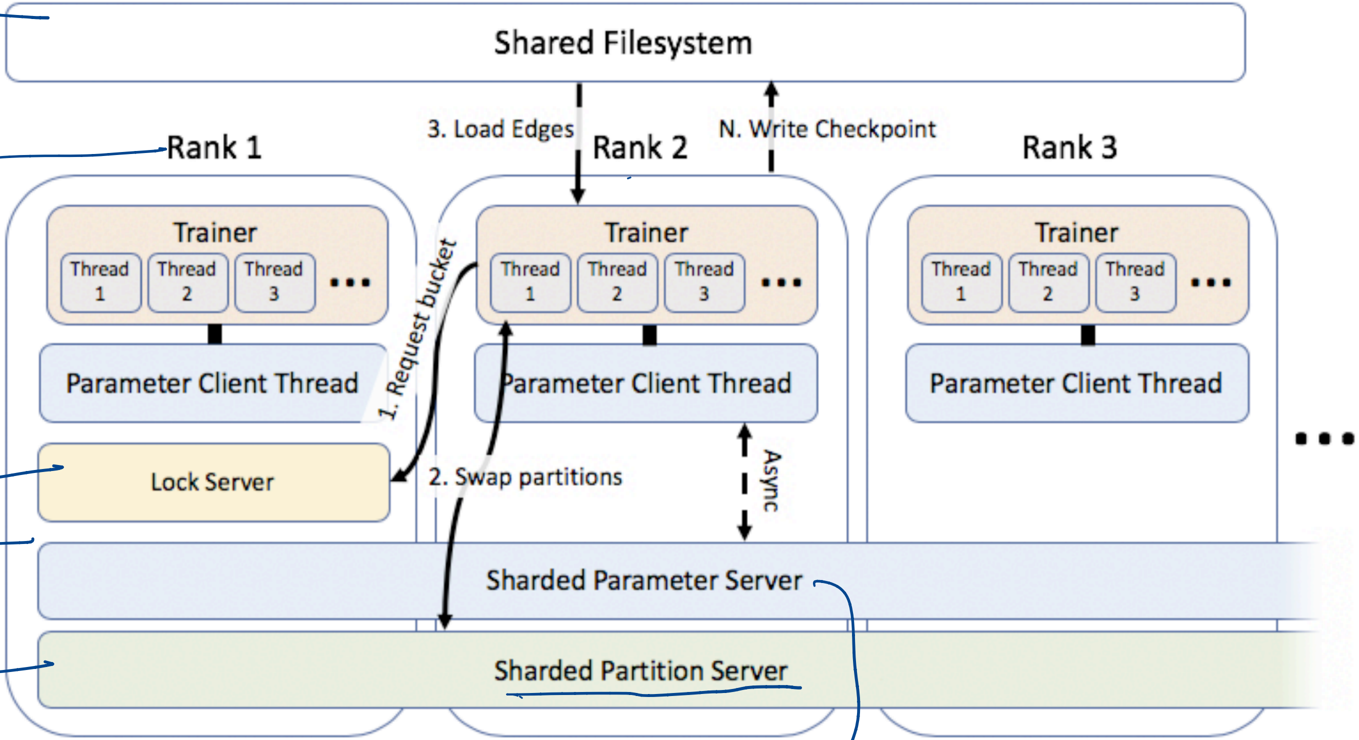
non-overlapping edge buckets be processed in parallel

partitions of vertex embeddings

SYSTEM DESIGN

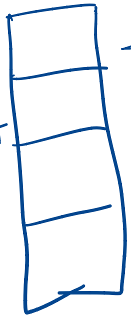
all edges of graph

Parallel workers



Assign partitions to workers

emb



vert ex

~GFS where each partition is on diff server

other relation embeddings on parameter server

DISTRIBUTED EXECUTION

At a worker

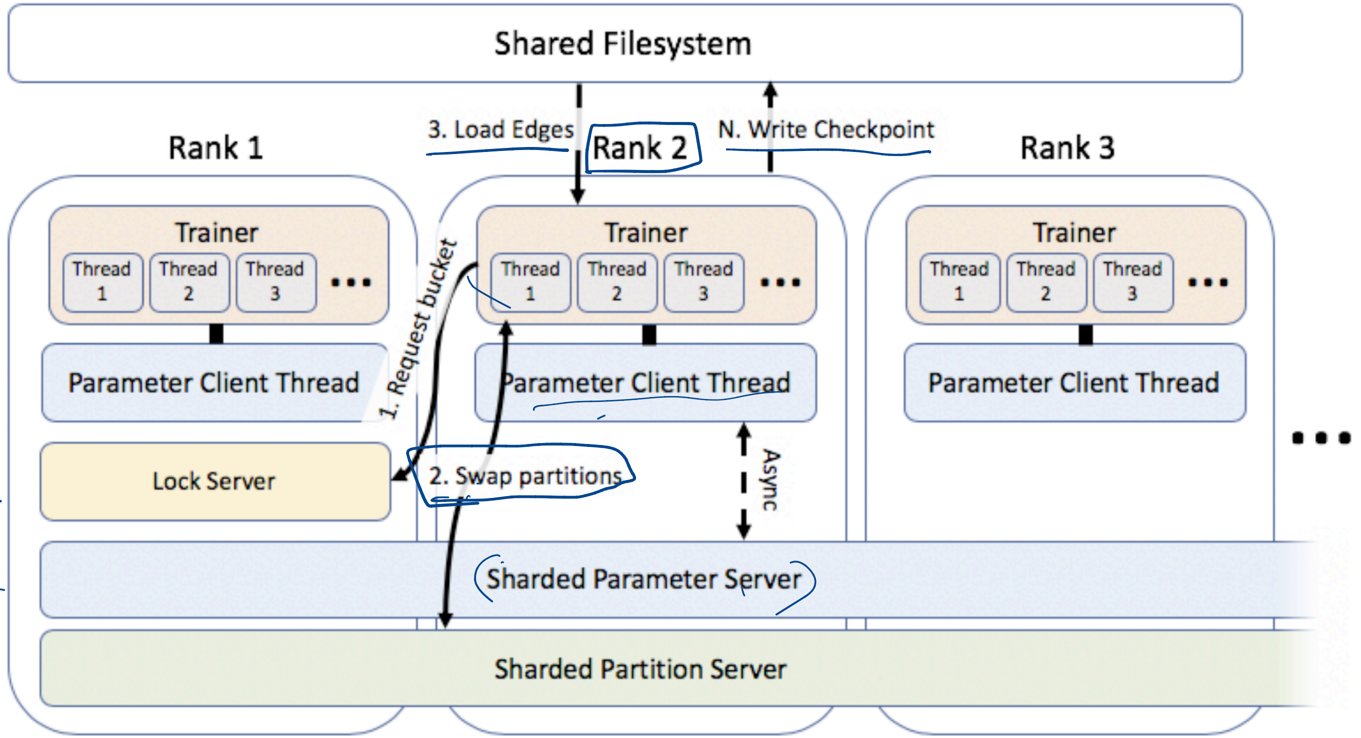
(1) Get a edge bucket from lock server

(2) Fetch vertex embedding partitions

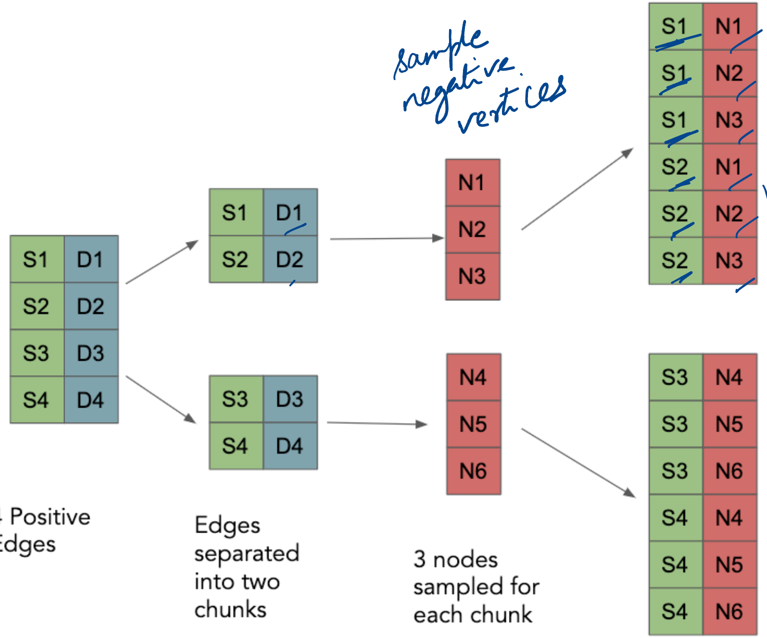
(3) Load positive edges from FS

DO TRAINING

Checkpoints to FS



BATCH NEGATIVE SAMPLING



4 Positive Edges

Edges separated into two chunks

3 nodes sampled for each chunk

Each positive edge now has 3 corresponding negative edges

sample negative vertices

1

3

(3, 1)

2x3

→ Also need to fetch emb. for negative samples!

↳ Only going to corrupt src/dest with vertices from same partitions

↳ Re-use negative edges by batching them.

you can also corrupt the source!

Thanks Jason!

Sample few

Get a lot of negatives

SUMMARY

Graph Embeddings: Learn embeddings from graph data for ML

Partition graph into buckets for scalability

Distributed execution with shared partition server

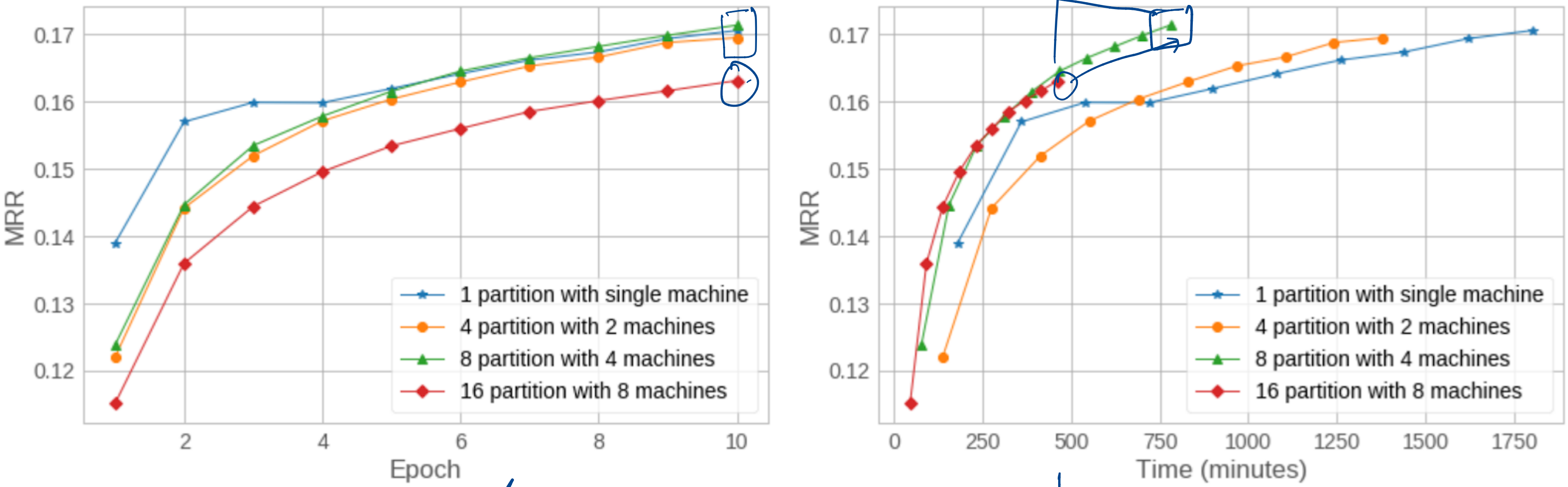
Batched negative sampling

DISCUSSION

https://forms.gle/YEtECXCUtksoL6Sr8

sweet spot!

Running for a fixed epochs may not be ideal?

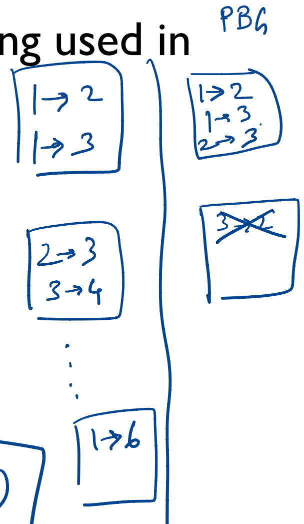


Accuracy degrades for 16 partitions

Time per epoch much lower with more machines

How does the partitioning scheme used in this paper differ from partitioning used in PowerGraph and why? (from review)

vertex cut →



Power Graph
↳ update vertex / edge state

here we update embeddings !

→ Powergraph : not all vertices might get updated
↳ Order of traversal didn't matters (barrier)

→ Powergraph
→ vertex had to look at all the neighbors

PBG → edge only look at emb. for src, dst

NEXT STEPS

Next class: New module!

Project check-ins by Nov 20th