Extreme Classification in Log-Memory using Count-Min Sketch: A Case Study of Amazon Search with 50MM products

Tharun Medini, Qixuan Huang, Yiqiu Wang, Vijai Mohan, Anshumali Shrivastava

tharun.medini@rice.edu

24th Nov 2019

What is Extreme Classification?

- Classification with a large number of classes (often running into millions!).
- Examples: Product Search[1,2], Search Query Suggestions[3], Ad Predictions[4]

[1] Nigam et al., *Semantic Product Search*. KDD 2019

[2] McAuley et al., *Image-based Recommendations on Styles and Substitutes*. SIGIR 2015

[3] Jain et al., *Slice: Scalable Linear Extreme Classifiers trained on 100 Million Labels for Related Searches*. WSDM 2019

[4] Prabhu et al., *Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising*. WSDM 2018

Scale Challenge

- The state-of-art models scale linearly with the number of classes. Hence, they cannot train beyond million classes.
- For 50 MM classes, a penultimate layer size of 2000 would require 100 billion parameters!
- Momentum based optimizers require 2x additional memory.
- **Needs 1.2TB GPU memory!**

Existing Methods

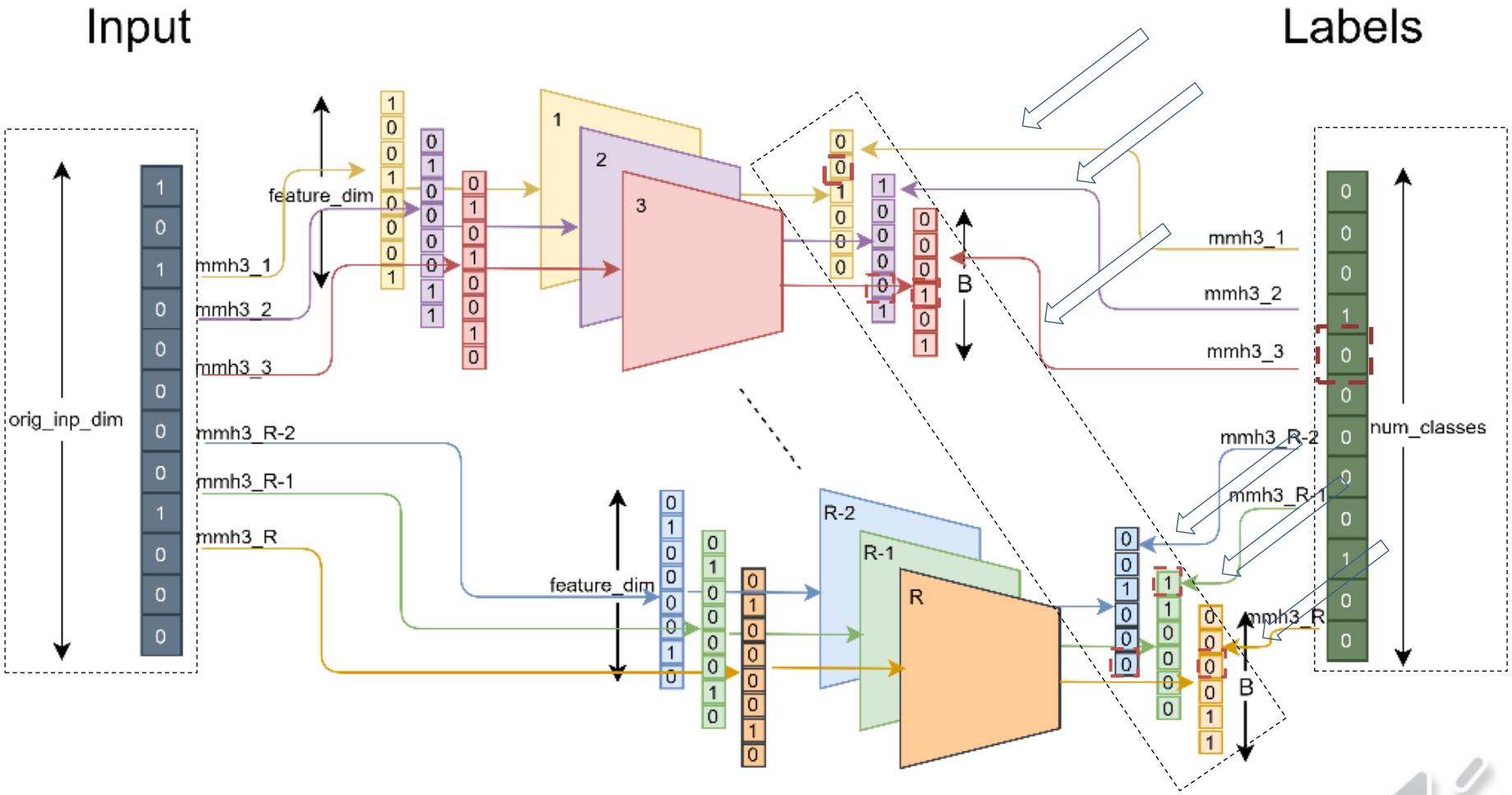
- Embedding Based Models: Learn a dense mapping for both inputs and labels and perform Approx-NN in the new embedding space.
    - Issues – Pairwise loss leads to large number of training data points. Need to do smart negative sampling.
- Parabel - Partial Tree Based Methods: Create a partial hierarchy of labels and train a 1-vs-all classifier for each of the leaf nodes.
    - Issues - Tree Based Methods are not conducive to GPUs.

Our Proposal:MACH

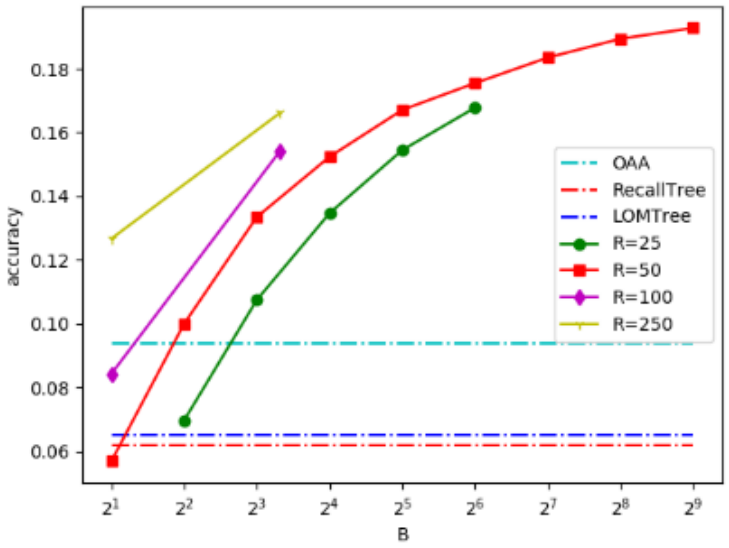
- MACH stands for Merged Average Classifiers via Hashing.
- It is a generic classification framework that scales O(logK), K being the number of classes.
- MACH facilitates zero-communication model parallelism.
- MACH learns to predict the Count-Min Sketch (CMS)[1] matrix of the sparse K-dimensional label vector.
- Retrieve heavy hitters during inference.

[1] Cormode et al., *An improved data stream summary: the count-min sketch and its applications*. Journal of Algorithms, 2005.

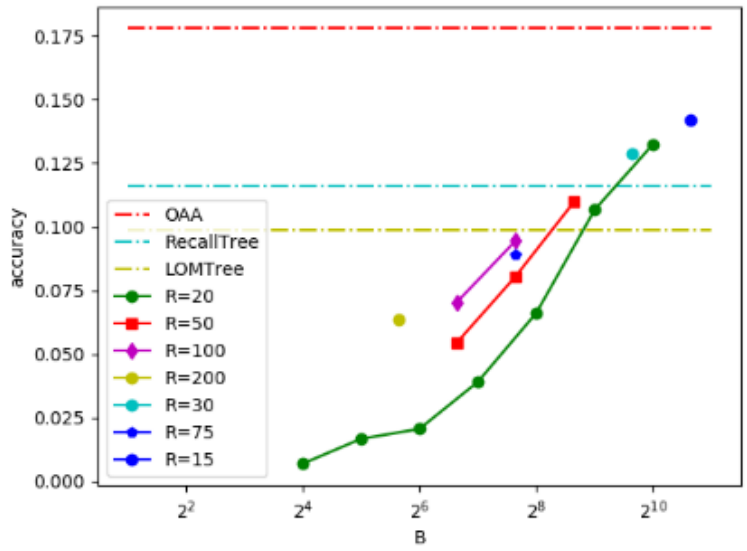
Schema


Input
Labels
orig_inp_dim
feature_dim
mmh3_1
mmh3_2
mmh3_3
mmh3_R-2
mmh3_R-1
mmh3_R
1
2
3
B
num_classes
R-2
R-1
R

Experiments – Public Datasets



ODP Dataset (105K classes)



Imagenet (22K classes)

Dataset	(B, R)	Model size Reduction	Training Time	Prediction Time per Query	Accuracy
ODP	(32, 25)	125x	7.2hrs	2.85ms	15.446%
Imagenet	(512, 20)	2x	23hrs	8.5ms	10.675%

Table 1: Wall Clock Execution Times and accuracies for two runs of MACH on a single Titan X.

Experiments – Amazon Search

- Anonymized, aggregated and sub-sampled Search Dataset from 5 different categories on Amazon Search Engine.
- 70 MM training samples, 50 MM classes, 20K test samples.

Model	epochs	wRecall	Total training time	Memory(Train)	Memory (Eval)	#Params
DSSM, 256 dim	5	0.441	316.6 hrs	**40 GB**	286 GB	200 M
Parabel, num_trees=16	5	0.5810	232.4 hrs (all 16 trees in parallel)	350 GB	426 GB	-
MACH, B=10K, R=32	10	**0.6419**	**31.8 hrs** (all 32 repetitions in parallel)	150 GB	**80 GB**	5.77 B
MACH, B=20K, R=32	10	**0.6541**	**34.2 hrs** (all 32 repetitions in parallel)	180 GB	90 GB	6.4 B

- Nigam et al., *Semantic Product Search*. KDD 2019
- Prabhu et al., *Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising*. WSDM 2018

Further Discussions

- Please refer to our paper for detailed experiments, metrics and theoretical discussion about why MACH works.
- Our code is hosted at https://github.com/Tharun24/MACH/
- Please contact tharun.medini@rice.edu/anshumali@rice.edu for further discussions.