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

Tharun Medini[1], Qixuan Huang[1], Yiqiu Wang[2], Vijai Mohan[3], Anshumali Shrivastava[1]

[1]Rice University, [2]MIT, [3]Amazon Search

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]

Scale Challenge

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

Existing Methods

- Embedding Models – Training data explodes, and negative sampling is required
- Parabel – Partial Tree based 1-vs-all classifier, not GPU friendly

Our Method: Merged Average Classifiers via Hashing (MACH)

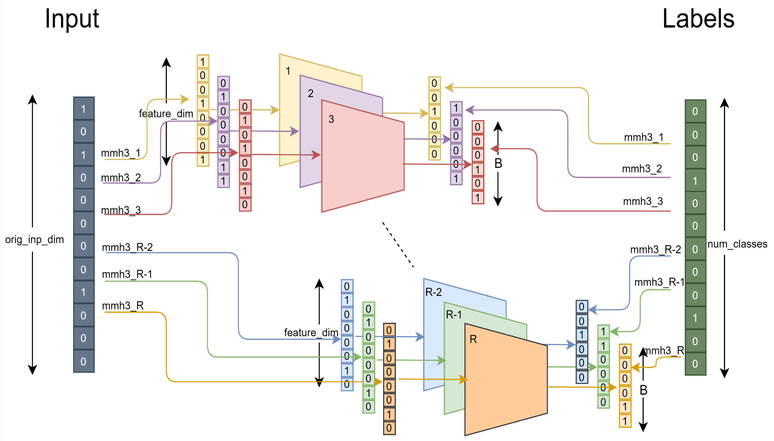
- Generic classification framework that provably scales O(logK)
- Facilitates zero-communication model parallelism
- MACH learns to predict Count-Min Sketch (CMS) matrix of the sparse K-dimensional label vector
- Retrieves the heavy-hitters during inference

Count-Min Sketch

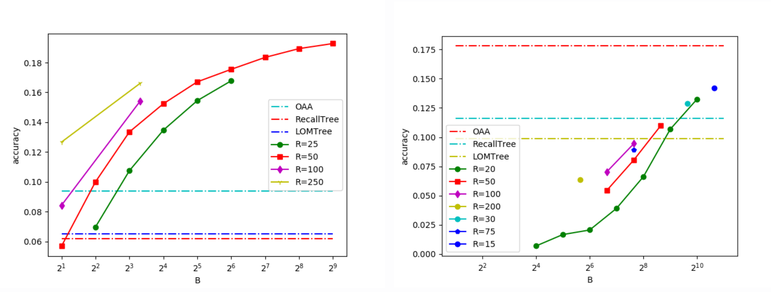
	H1	H2	H3	H4
A	1	6	3	1
B	1	2	4	6
C	3	4	1	6
D	6	2	4	1

	0	1	2	3	4	5	6
H1	0	1+1+1+1=4		1+1 = 2	0	0	1
H2	0	0	1+1=2	0	1+1 = 2	0	1+1+1=3
H3	0	1+1 = 2	0	1+1+1=3	1+1 = 2	0	0
H4	0	1+1+1+1=4	0	0	0	0	1+1+1=3

Methodology



Multiclass Datasets



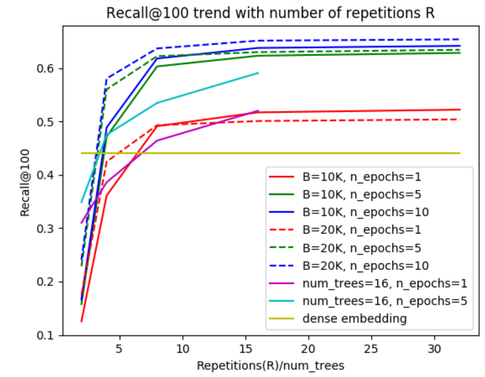
Accuracy-Resource tradeoff with MACH with varying settings of R and B. Left: ODP Dataset. Right: Imagenet Dataset

Multilabel Datasets

Dataset	Precision@K	MACH	Parabel	DisMEC	PfastreXML	FastXML
Wiki10-31K	P@1	**0.8544**	0.8431	0.8520	0.8357	0.8303
	P@3	0.7142	0.7257	**0.7460**	0.6861	0.6747
	P@5	0.6151	0.6339	**0.6590**	0.5910	0.5776
Delicious-200K	P@1	0.4366	**0.4697**	0.4550	0.4172	0.4307
	P@3	**0.4018**	0.4008	0.3870	0.3783	0.3866
	P@5	**0.3816**	0.3663	0.3550	0.3558	0.3619
Amazon-670K	P@1	0.4141	**0.4489**	0.4470	0.3946	0.3699
	P@3	**0.3971**	**0.3980**	**0.3970**	0.3581	0.3328
	P@5	**0.3632**	0.3600	0.3610	0.3305	0.3053

Comparison of MACH and popular extreme classification algorithms on few public datasets. MACH mostly preserves the precision and slightly betters the best algorithms on half of the cases. These numbers also establish the limitations of pure tree-based approaches FastXML and PfastreXML

Amazon – 50 MM dataset



MACH vs DSSM vs Parabel

Model	Epochs	wRecall @100	Training time	Peak Memory-Training	Peak Memory-Eval
DSSM – 256d	5	0.441	316.6 hrs	**40 GB**	286 GB
Parabel, 16 trees	5	0.5810	232.4 hrs	350 GB	426 GB
MACH, B=10K, R=32	10	**0.6419**	**31.8 hrs**	150 GB	**80 GB**
MACH, B=20K, R=32	10	**0.6541**	**34.2 hrs**	180 GB	90 GB

References

[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

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

For More Details

Please attend ML with Guarantees Workshop for Theoretical Discussion

Tharun Medini: tharun.medini@rice.edu

Anshumali Shrivastava: anshumali@rice.edu

RUSH-LAB: rush.rice.edu