CS 744: CLIPPER

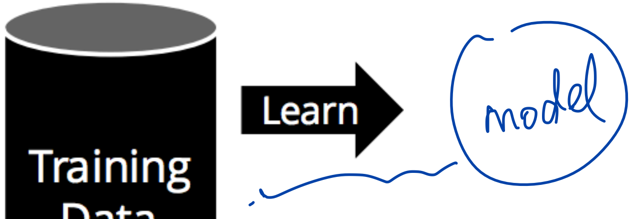
Shivaram Venkataraman

Fall 2019

ADMINISTRIVIA

- Assignment 2 grading
- Midterm details
- Course Project template - Due Thu

MACHINE LEARNING SO FAR



Systems
to
learn models

Shared memory

Distributed

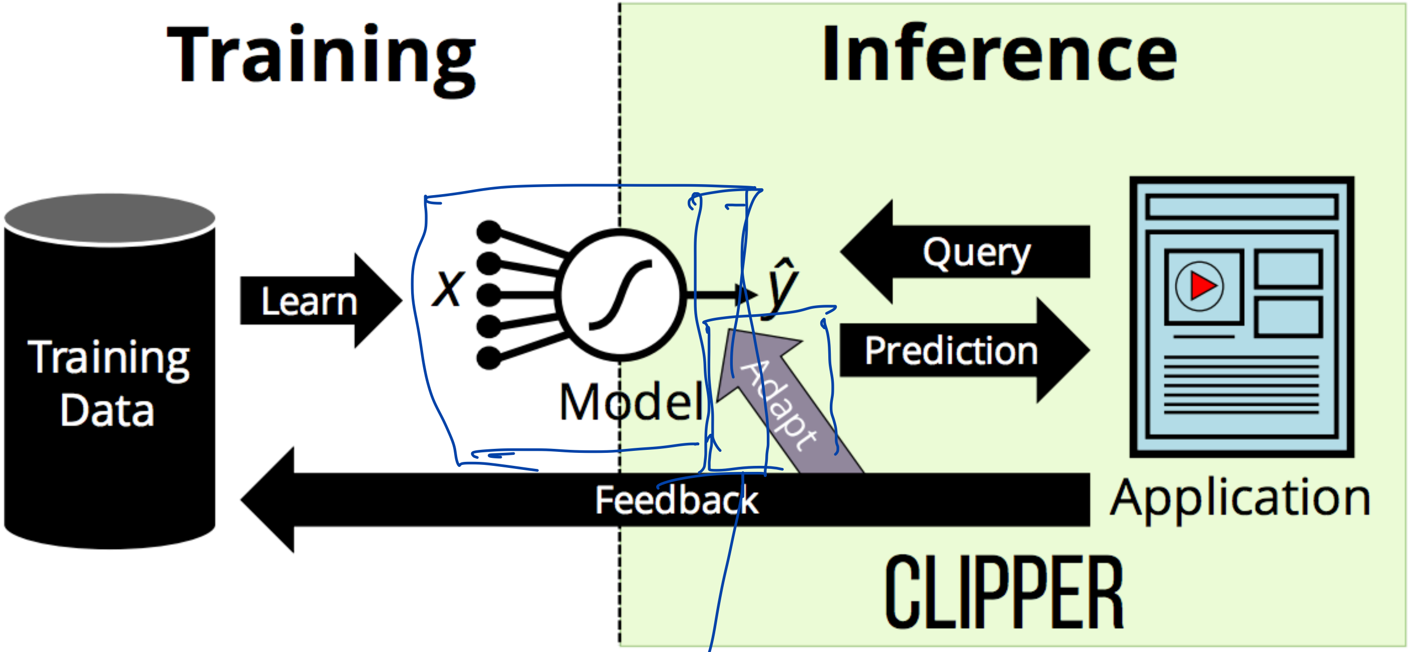
Flexible Programming Model

Reinforcement learning

MACHINE LEARNING: INFERENCE

Movie

Recommendation



Incrementally update the model / Prediction

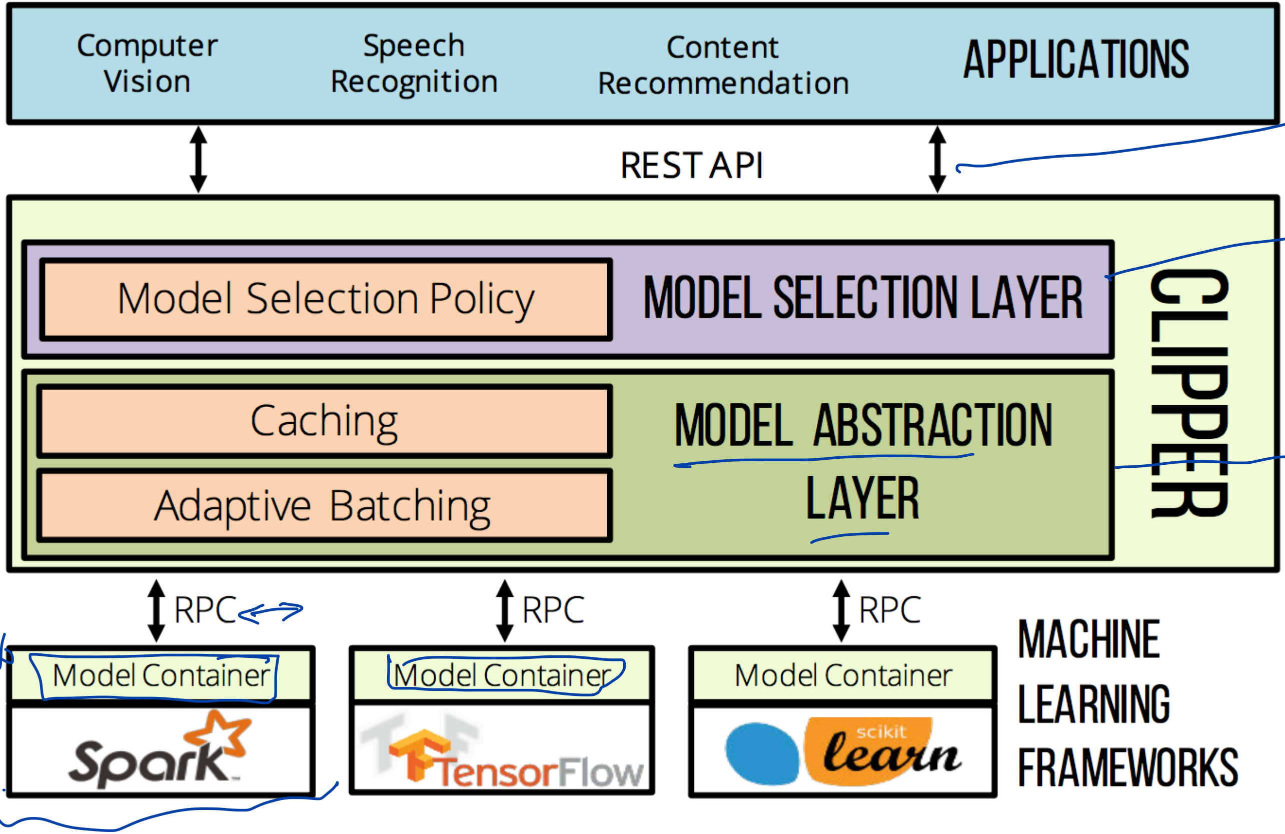
GOALS

- Interactive latencies (tail latency < 100ms) → Page load times
- High throughput to handle load → To handle many users

→ General

- Improved prediction accuracy
- Generality (?) → Support multiple ML frameworks

ARCHITECHTURE



Computer Vision

Speech Recognition

Content Recommendation

APPLICATIONS

REST API

Model Selection Policy

MODEL SELECTION LAYER

Caching

Adaptive Batching

MODEL ABSTRACTION LAYER

CLIPPER

RPC

RPC

RPC

Model Container

Model Container

Model Container

MACHINE LEARNING FRAMEWORKS

One server

frontend

Improve Accuracy

Many workers

Improving Performance

Common interface

Isolation

Generality goal

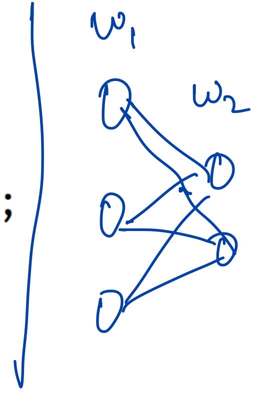
MODEL CONTAINERS

→ No workers / PS tasks inside MC

→ No training data

→ Assumption: weights fit in MC

Training $\xrightarrow{f}$ labels



```
interface Predictor<X,Y> {
  List<List<Y>> pred_batch(List<X> inputs);
}
```

for each input

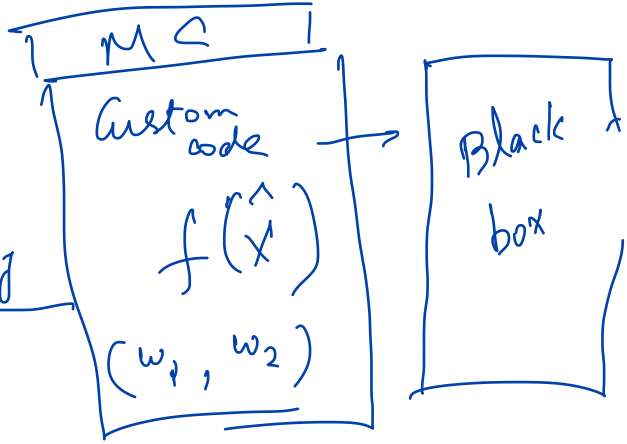
list of outputs

- Run using Docker containers

- Can be replicated across machines

↳ Replication factor could vary

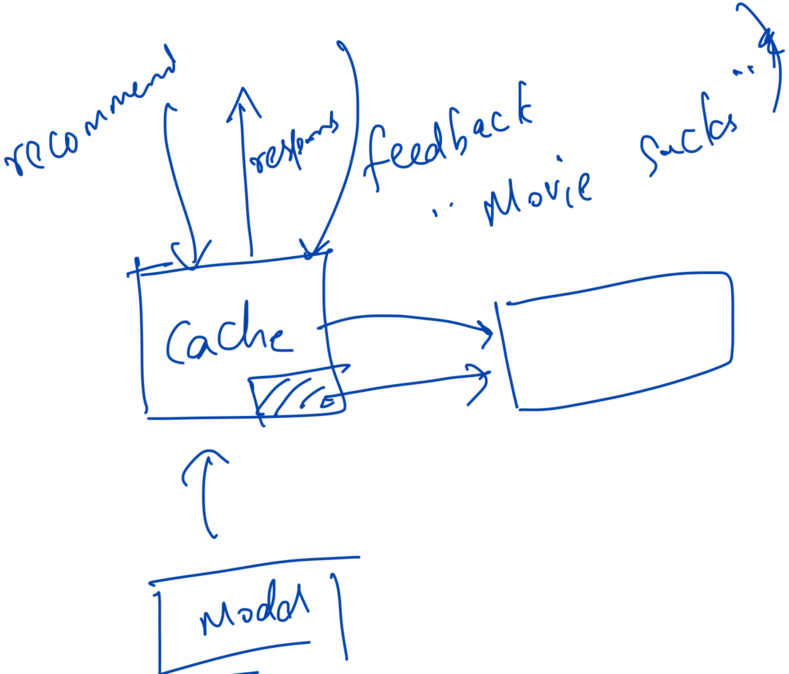
Pre-processing
+
Model inference



MODEL ABSTRACTION LAYER

Caching

- Improve performance for frequent queries
- LRU eviction policy
- Important for feedback



BATCHING, QUEUING

100 request
batch size : 50

= 100 RPCs w/o batching   2 RPCs w/ batching

Goals, Insight

- Increase latency (within SLO) for improved throughput
- Reduce RPC overheads
- GPU / BLAS acceleration

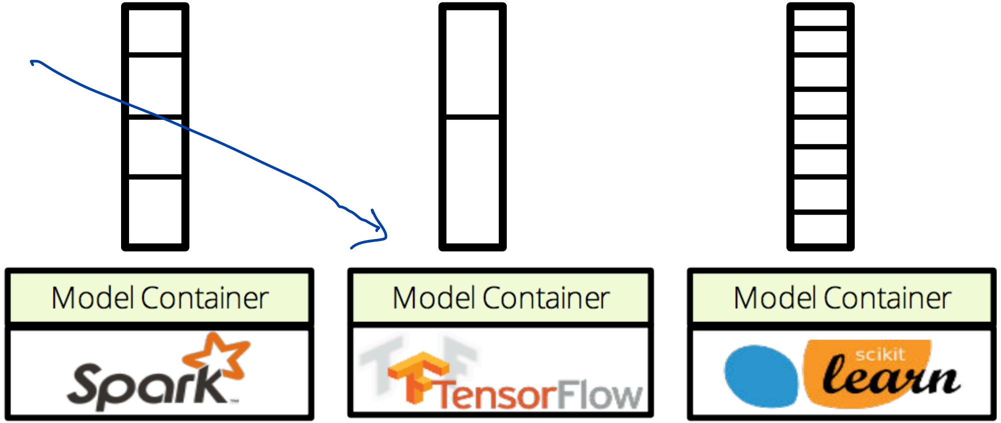
→ Opt Batch size different for each container

→ 500 batchsize : 20 ms

Approach

- Per container queues.
- Maximum batch size. → Bound latency
- Why?

TF



GPU

ADAPTIVE BATCHING

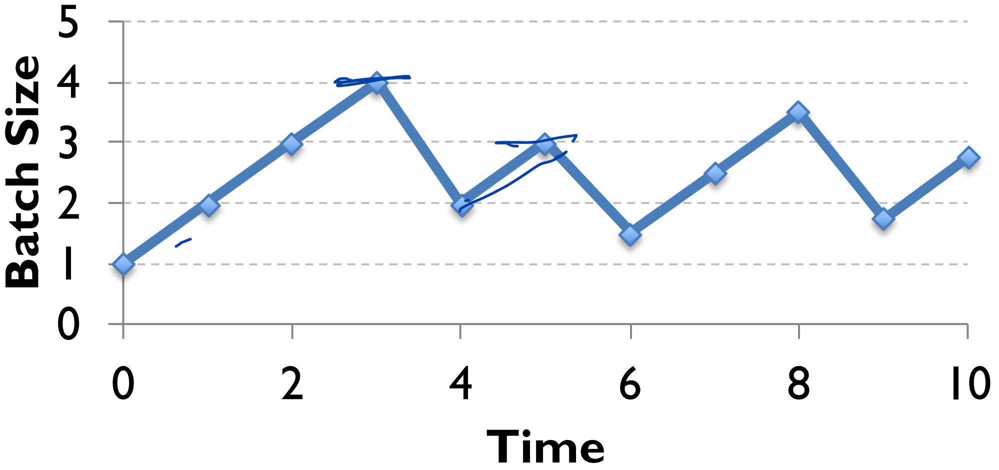
AIMD: Additive Inc Multiplicative Dec

Why ? ~ ~10% of

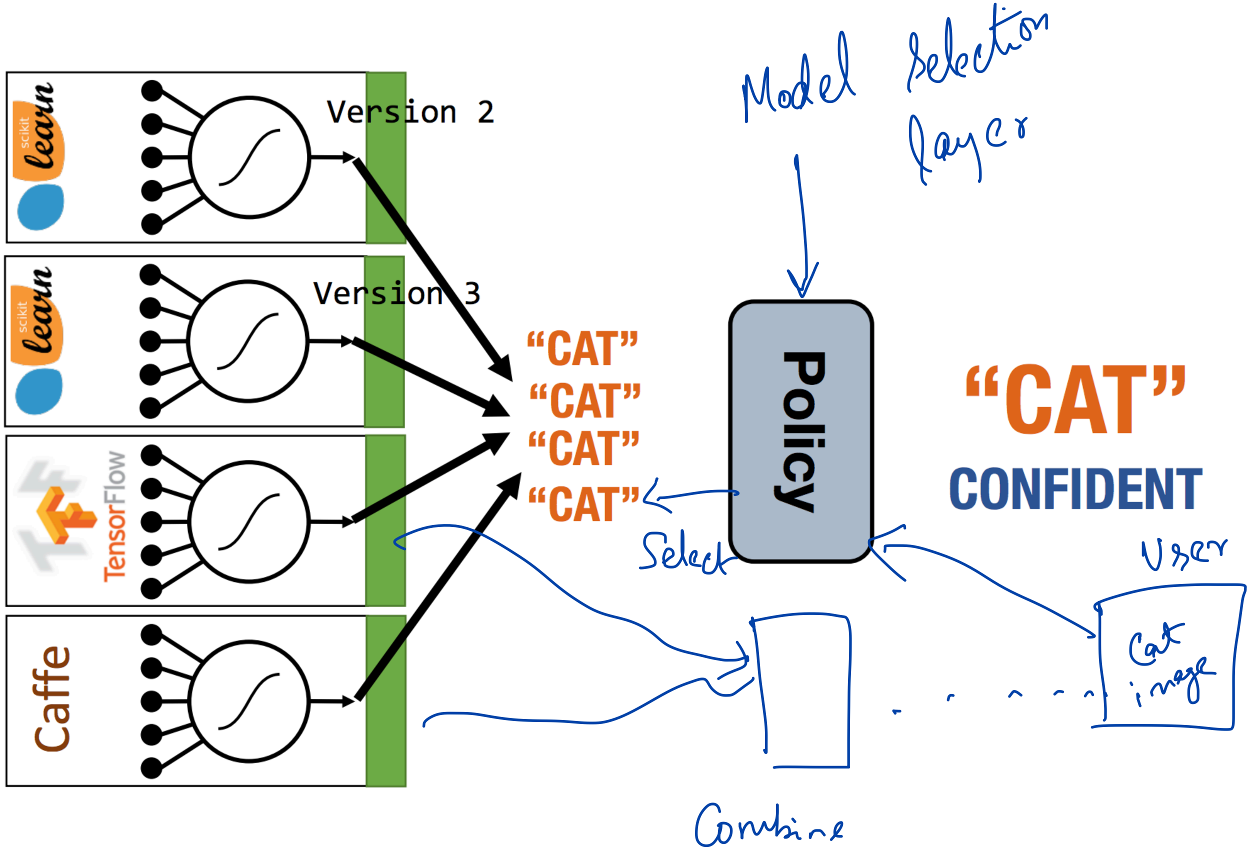
Runtime variations

Delayed: Wait until batch exists

Why?



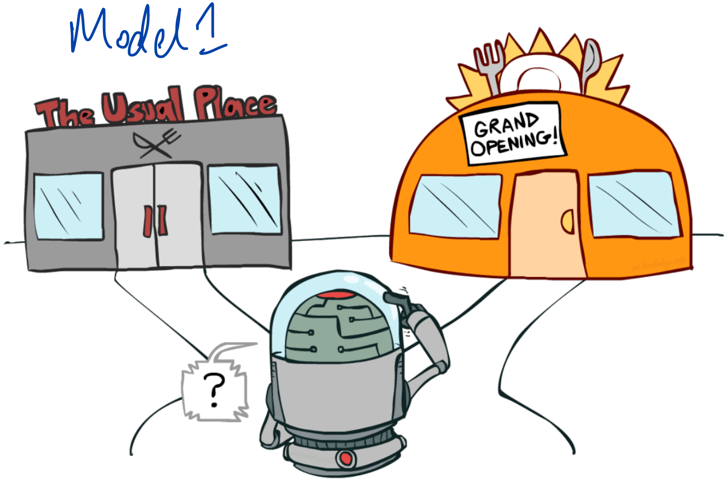
MODEL SELECTION



SINGLE MODEL SELECTION

Model 2

Model 1



Multi-Arm Bandit formulation

- Explore vs Exploit
- Regret: Loss by not picking optimal action
- Goal: Minimize regret

Clipper

- Exp3 algorithm → uses feedback to influence next selection.
- Single evaluation
- Scales to more models

MULTI MODELS

Ensemble

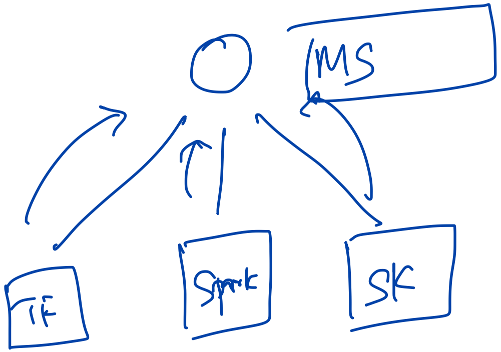
- Combine output from models (weighted average)
- How do we get the weights ? → Linear Combination

Robust Prediction

- React to model changes
- Output confidence score

STRAGGLER MITIGATION

Why do stragglers occur?



All of the requests finish

Approach

↳ Approx result not exact

↳ Better approx than late!

→ ML specific

→ Slower model could be more accurate!

Deadline → Pick the best based on results within deadline

TAKEAWAYS

- ML inference: Workloads + Requirements
- Layered architecture provides generality
- Caching, Batching, Replication to improve latency, throughput
- Multi-Arm bandits to improve accuracy

DISCUSSION

https://forms.gle/pZMuhCWcap2q3LQJ9

(Discussion question from last week)

Considering AllReduce using MPI as the baseline parallel programming task. Discuss the improvements made by MapReduce, Spark over MPI and discuss if/how Ray further contributes to the comparison.

- Ease of programming → MR
  ↳ Spark
- Fault tolerance → MR → disk
  Spark → Lineage
  Ray ↗
- Straggler mitigation → MR → Speculative tasks
- Network Usage
  → Controlled partition
  → Locality → Mem, disk → MR
  ↳ spark, Ray
  →

Scheduling benefits
↳ Locality
↳ Scale
↳ Ray

Consider a scenario where you run a model serving service that hosts a number of different ~~models~~. The traffic for some ~~models~~ is sporadic (e.g. only a few hours where they are used). What are some advantages / disadvantages of using Clipper for such a service?

application

apps

- Automatically scale up, scale down

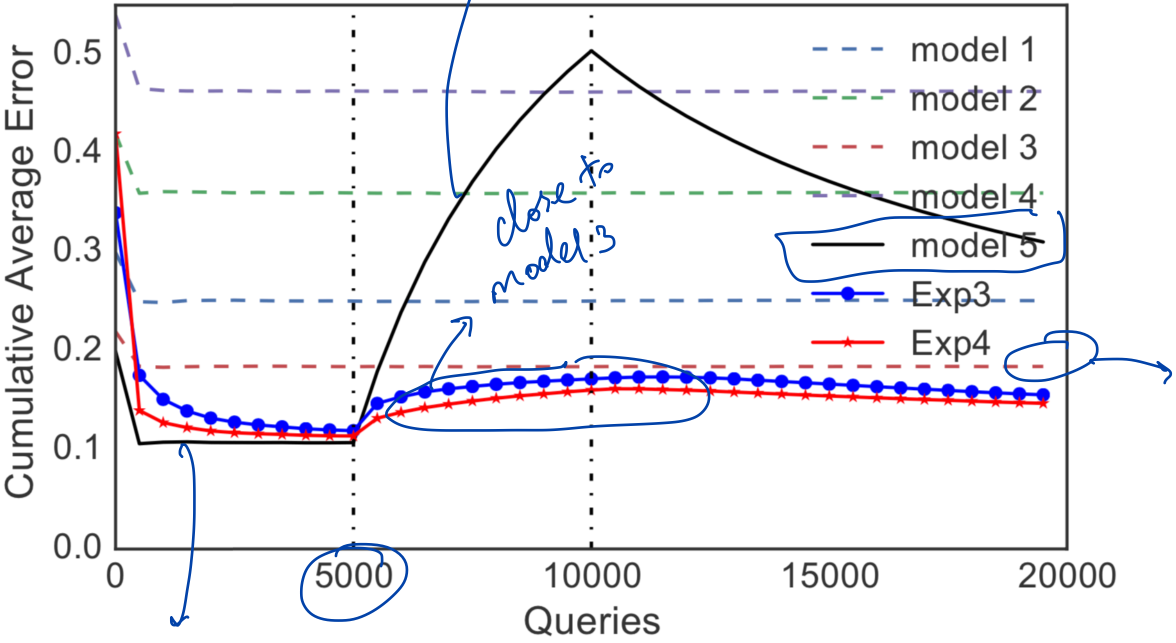
  At least 1 container active

- Batching is not effective, sporadic

- SLOs could be different →

  Caching might not be effective

model 5 is bad here



close to model 3

model 1 + 2 + 4 + model 5 +

model 3

exp3, exp4 is close to model 5