CS 744: BIG DATA SYSTEMS

Shivaram Venkataraman
Fall 2018
- Assignment 1 Grades
- Midterm review session
- Course Project Template
SETTING: INFERENCE

Training

- Training Data
- Learn
- Model
- Learn

Inference

- Query
- Prediction
- Feedback
- CLIPPER

Application
GOALS

- Interactive latencies (< 100ms)
- High throughput to handle load
- Improved prediction accuracy
- Generality (?)
MODEL CONTAINERS

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

- Run using Docker containers
- Can be replicated across machines
MODEL ABSTRACTION LAYER

Caching
- Improve performance for frequent queries
- LRU eviction policy
- Important for feedback
BATCHING, QUEUING

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

Approach
- Per container queues
- Maximum batch size
ADAPTIVE BATCHING

AIMD
- Additive Increase
  Multiplicative Decrease
- Why?

Delayed Batching
- Wait until batch exists
- Useful for high startup overheads?
MODEL SELECTION

Version 2

Version 3

"CAT"
"CAT"
"CAT"

"CAT"
CONFIDENT
SINGLE MODEL SELECTION

Multi-Arm Bandit formulation
- Explore vs Exploit
- Regret: Loss by not picking optimal action
- Goal: Minimize regret

Clipper
- Exp3 algorithm
- Single evaluation
- Scales to more models
MULTI MODELS

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

Robust Prediction
 - React to model changes
 - Output confidence score
STRAGGLER MITIGATION

Setup
- Ensembles result in more queries
- Requires more replicas / scaling → Stragglers

Approach
- Better inaccurate than late
- Policy handles values missing at deadline
TAKEAWAYS

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