Hemingway
Modeling Distributed Optimization Algorithms

Xinghao Pan, Shivaram Venkataraman, Zizheng Tai, Joseph Gonzalez
Problem: Train Model from Big Data

- ADMM
- L-BFGS
- Minibatch SGD
- Gradient Descent
- Splash
- CoCoA
- Distributed SVRG

Data → Model
Problem: Choose Algorithm and Parallelism

Given target accuracy, choose algorithm and parallelism to minimize \( \text{time} / \text{cost} \).
\[
\min_{m, A, \text{iter}} \text{cost}_A(m, \text{iter}) \quad \text{s.t.} \quad \text{accuracy}_A(m, \text{iter}) \geq \epsilon
\]

Given time / cost budget, choose algorithm and parallelism to maximize accuracy.
\[
\max_{m, A, \text{iter}} \text{accuracy}_A(m, \text{iter}) \quad \text{s.t.} \quad \text{cost}_A(m, \text{iter}) \leq C
\]
Parallelism: Good, Bad, Ugly

Parallelism improves Computation
Parallelism impairs Communication

Parallelism impacts Convergence

More cores = Poorer convergence
Modeling Distributed Optimization

\[
\text{TimeToAccuracy}(m, \text{eps}) = \text{TimePerIteration}(m) \times \text{IterationsToAccuracy}(m, \text{eps})
\]

Ernest
Computation + Communication model

Hemingway
Convergence model

Data driven approach
Trained separately, predict jointly
Facilitates reuse
Ernest
Computation + Communication model

\[
\text{TimePerIteration}(m) = \frac{\theta_0}{m} + \theta_1 m + \theta_2 \log(m) + \theta_3
\]

# machines
Data meta-data
Data values

Ernest

TimePerIteration\( (m) = \theta_0 / m + \theta_1 m + \theta_2 \log(m) + \theta_3 \)

Collect training data

Fit linear regression
Hemingway
Convergence model

AccuracyAtIteation\((m, \text{iter})\)
\[= \sum_j \lambda_j \phi_j(m, \text{iter})\]

Guided but not constrained by theoretical convergence rates
Upper bound rates \(\neq\) observed convergence
Additional non-linear \(\phi_j\) terms

Collect training data
Fit Lasso
Collect training data

Fit Lasso
Collect training data

Fit Lasso

Cross Validation Fit

Sub-optimality

Iteration

m=4, actual
m=4, estimated
m=16, actual
m=16, estimated
m=64, actual
m=64, estimated
m=128, actual
m=128, estimated
Collect training data up to $i$th iteration

Fit Lasso, predict $(i+1)$th iter
Collect training data up to \( i \)th iteration

Fit Lasso, predict \((i+10)\)th iter
Ernest Hemingway: Future Time

Predicting 1000ms Ahead

Collect training data up to time $t$

Fit Lasso, predict time $t+1000$
Ernest Hemingway: Future Time

**Predicting 5000ms Ahead**

Collect training data up to time $t$

Fit Lasso, predict time $t+5000$
Ernest Hemingway Vision

Goal: Maximize accuracy in minimum time
Challenges + Directions

Adaptive learning

Reinforcement learning
- Explore vs exploit
- Modeling and learning rewards

Non-convex problems
- Non-monotone, non-smooth convergence rate

Asynchronous and other algorithms
- Systems vs Convergence breakdown valid?
- Ernest systems model valid?

Other input hyperparameters

Ernest Hemingway

Distributed running time depends on Computation, Communication, Convergence depends on Algorithm and Parallelism

Approach: Data driven modeling of distributed optimization

Ernest: White-box systems model

Hemingway: Black-box convergence model

Train separately, predict jointly

xinghao@berkeley.edu