Snap ML: A Hierarchical Framework for Machine Learning
C. Dünnér*, T. Parnell*, D. Sarigiannis*, N. Ioannou*, A. Anghel*, G. Ravi†, M. Kandasamy‡, H. Pozidis*

Contributions
✓ Snap ML is a new framework for efficient training of generalized linear models.
✓ Snap ML implements novel out-of-core techniques to enable GPU acceleration at scale.
✓ Snap ML is built on a novel hierarchical version of the popular CoCoA framework to enable multi-level distributed training.
✓ Snap ML can train a logistic regression classifier on the Criteo Terabyte Click Logs data in 1.5 minutes.

The Snap ML Framework

e.g. SVM, Lasso, Ridge Regression, Logistic Regression

The Hierarchical Optimization Framework

Hierarchical Optimization Framework

Consider Algorithm 1 applied to the log-linear objective where the local subproblems are solved with relative accuracy η in each iteration. Let f be β-smooth and convex and gi be general convex functions. Then, after T1 outer iterations with T2 inner iterations each the suboptimality is bounded as

E[ci] ≤ \left(1 + \frac{\beta K_0}{1 - (1 - (1 - 0.5) R_g)}\right) \frac{1}{T_1}

Furthermore, if gi are µ-strongly convex this rate improves to

E[ci] ≤ \left(1 + \frac{\beta K_0}{1 - (1 - (1 - 0.5) R_g)}\right) \frac{1}{T_1}

Trade-off parameters θ and T2

→ Take advantage of non-uniform interconnects

Algorithm 1: Hierarchical Distributed Framework

1. Input: Data matrix d ∈ R^{n×d} distributed columns-wise across K/2 workers
2. Model vector α ∈ R^d and shared vectors v ∈ R^d.
3. Stopping point ϵ ≥ 0, v ≥ 0.
4. for t = 0, 1, 2, . . . do
5. for i = 0, 1, 2, . . . , T1 do
6. for l ∈ {1, 2, . . . , L} in parallel across workers do
7. Update model parameter α_{l,i} = α_{l,i-1} - ϵ \nabla_{α_{l,i}} f_{l,i}(α_{l,i-1}, v)
8. end for
9. end for
10. Update shared parameter v = ϵ \sum_{l=1}^{L} \nabla_{v} f_{l,i}(α_{l,i}, v)
11. end for
12. end for

Disjoint partitions {\{(E_i(E), f_{l,E})\}, E = 1, 2, . . . , n} of the data domain. Each worker only needs to communicate with its neighbors in the graph, analogous to the local subtasks defined by recursively applying a block-separable upper-bound to similar to [4].

Disjoint partitions {\{(E_i(E), f_{l,E})\}, E = 1, 2, . . . , n} of the data domain. Each worker only needs to communicate with its neighbors in the graph, analogous to the local subtasks defined by recursively applying a block-separable upper-bound to similar to [4].

Tera-Scale Benchmark

Data: # examples 4.5 billion
# features 1 million
~3TB

Performance of Snap ML in comparison to other frameworks and previously published results for training a logistic regression classifier on the Terabyte Click Logs dataset.