Scalable HOGWILD! for Big Models
Chengjie Qin, Martin Torres, and Florin Rusu
University of California Merced

Background & Motivation
Example: Gradient in logistic regression where $\vec{x}$ is the training example; $\vec{w}$ is the model.

$$\sum_i e^{-y_i \vec{x} \cdot \vec{w}} \vec{x}$$

Model size grows beyond the memory of a single machine.

Scalable HOGWILD! Framework

Experimental Evaluation

Dataset | # Dims | Examples | Size | Model
--- | --- | --- | --- | ---
uniform | 1B | 80K | 4.2 GB | 12 GB
skewed | 1B | 1M | 4.5 GB | 12 GB
splice | 500K | 10M | 128 MB | 128 MB
MovieLens | 6K x 4K | 1M | 24 MB | 120 MB

Table: Datasets used in the experiments.

Background & Motivation

Immediate Extension of Big Model HOGWILD!

1. for $i = 1$ to $N$ do in parallel
2. for each non-zero feature $j \in \{1, \ldots, d\}$ in $\vec{x}_i$ do
3. get $\vec{w}(k)[j]$
4. compute $\nabla_{\vec{w}(k)} \frac{1}{N} \sum_i y_i \vec{w}(k) \cdot \vec{x}_i$
5. end for
6. $\vec{w}(k+1) \leftarrow \vec{w}(k) - \eta \nabla_{\vec{w}(k)} \frac{1}{N} \sum_i y_i \vec{w}(k) \cdot \vec{x}_i$
7. for each non-zero feature $j \in \{1, \ldots, d\}$ in $\vec{x}_i$ do
8. set $\vec{w}(k+1)[j]$
9. end for
10. end for

Online Model and Data-Parallel Asynchronous Training

Model Vertical Partitioning Algorithm

Require:
- Model index set $I = \{1, 2, \ldots, d\}$
- Sparse vector set $X = (x_1, x_2, \ldots, x_d)$

Ensure:
- Model partitions $P = (P_1, P_2, \ldots)$
- for each index $i \in I$ do $P_i$ $\rightarrow$ $i$

Main loop
1. while true do
2. Compute affinity matrix $AF$ for partitions in $P$
3. for each vector $x_i \in P$ do
4. Collect partitions accessed by $x_i$ in $P_1$
5. Compute affinity $AF([i])$ for all pairs $(P_i, P_j)$ in $P$
6. end for
7. for each partition $P_i \in P$ do
8. Compute cost $C_i = AF([i]) \cdot \text{cost}(P_i)$
9. Select the best pair of partitions to merge
10. for each pair $(P_i, P_j)$ in do
11. Compute cost $C_{ij} = AF([i] \cup [j]) \cdot \text{cost}(P_i \cup P_j)$
12. Compute reduction in cost if $P_i$ and $P_j$ are merged
13. if $\Delta C = C_i + C_j - C_{ij}$
14. Pick the pair $(P_i, P_j)$ with largest reduction in cost $\Delta C$
15. else Merge $P_i$ and $P_j$
16. end while
17. end for

Experimental Evaluation

Dataset | # Dims | Examples | Size | Model
--- | --- | --- | --- | ---
uniform | 1B | 80K | 4.2 GB | 12 GB
skewed | 1B | 1M | 4.5 GB | 12 GB
splice | 500K | 10M | 128 MB | 128 MB
MovieLens | 6K x 4K | 1M | 24 MB | 120 MB

Table: Vertical partitioning time (in Table): Execution time per iteration seconds as a function of the pruning ratio $K$. (c) Time per iteration. (d) Convergence over K.

Figure: SVM on skewed. (a) Convergence over time. (b) Speedup over serial KV. (c) Time per iteration. (d) Convergence over K.

Figure: LMF on MovieLens. (a) Convergence over time. (b) Speedup over serial KV. (c) Time per iteration. (d) Convergence over K.