Mini-batch & distributed SAGA

Incremental Gradient Methods

Michaël Defferrard[†], Soroosh Shafieezadeh Abadeh[†]

[†] Ecole Polytechnique Federale de Lausanne

Student Presentation

3^{rd} June

Introduction

$$\text{Problem:}\quad \min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(x) + h(x)$$

SAGA

Given a learning rate γ, the value of x^k and of each $f_i'(\phi_i^k)$ at the end of iteration k, the updates for iteration $k+1$ are given by:

1. Pick a j uniformly at random.
2. Take $\phi_j^{k+1} = x^k$, and store $f_j'(\phi_j^{k+1})$ in the table.
3. Update x:

$$w^{k+1} = x^k - \gamma\left[f_j'(\phi_j^{k+1}) - f_j'(\phi_j^k) + \frac{1}{n}\sum_{i=1}^{n} f_i'(\phi_i^k)\right],$$

$$x^{k+1} = \operatorname{prox}_\gamma^h(w^{k+1}).$$

Improvements

- Memory-efficiency $\rightarrow$ **mini-batch** SAGA
- Time-efficiency $\rightarrow$ **distributed** SAGA

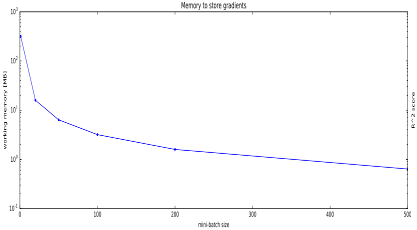
Mini-batch SAGA

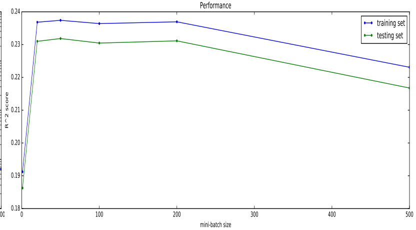
1. Pick a i uniformly at random in $[1, \frac{n}{m}]$.
2. Take $\phi_j^{k+1} = x^k \ \forall \ j \in \mathcal{B}_i$, and store $\frac{1}{m} \sum_{j \in \mathcal{B}_i} f_j'(\phi_j^{k+1})$ in the table.
3. Update x:

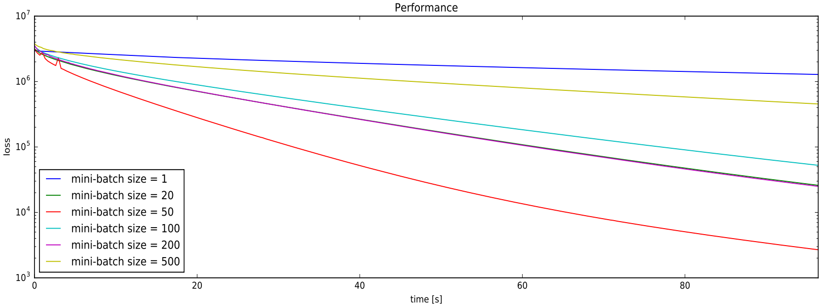
$$w^{k+1} = x^k - \gamma \left[\frac{1}{m} \sum_{j \in \mathcal{B}_i} f_j'(\phi_j^{k+1}) - \frac{1}{m} \sum_{j \in \mathcal{B}_i} f_j'(\phi_j^{k}) + \frac{1}{n} \sum_{i=1}^{m} \sum_{j \in \mathcal{B}_i} f_j'(\phi_j^{k}) \right],$$

$$x^{k+1} = \mathrm{prox}_\gamma^h(w^{k+1}).$$

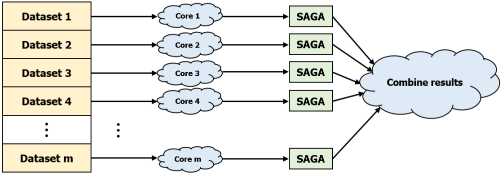
Results

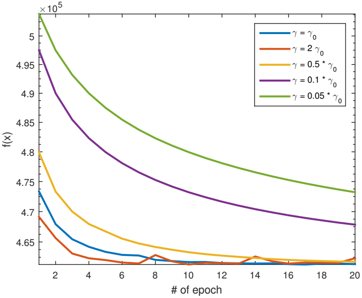

Memory to store gradients
working memory [MB]
mini-batch size


Performance
R^2 score
mini-batch size
training set
testing set

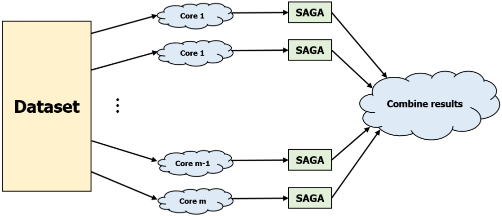

Performance
loss
time [s]
mini-batch size = 1
mini-batch size = 20
mini-batch size = 50
mini-batch size = 100
mini-batch size = 200
mini-batch size = 500

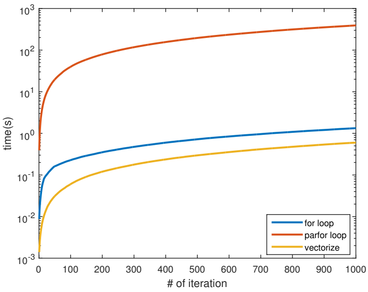
Variant I

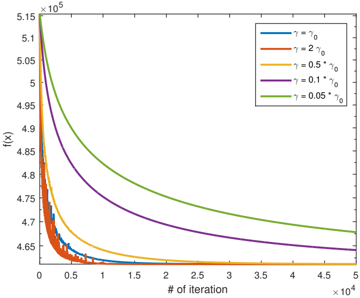

Dataset 1
Dataset 2
Dataset 3
Dataset 4
Dataset m
Core 1
Core 2
Core 3
Core 4
Core m
SAGA
Combine results


$\times 10^5$
f(x)
# of epoch
$\gamma = \gamma_0$
$\gamma = 2\ \gamma_0$
$\gamma = 0.5 * \gamma_0$
$\gamma = 0.1 * \gamma_0$
$\gamma = 0.05 * \gamma_0$

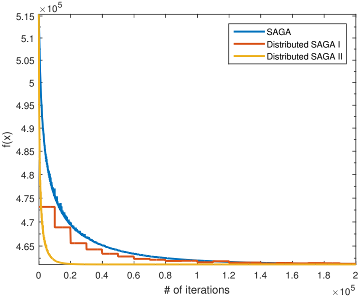
Variant II

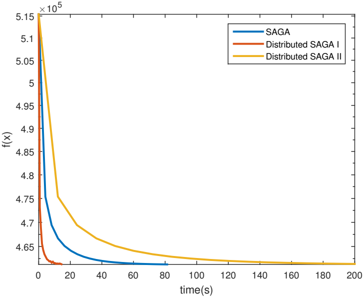

Dataset
Core 1
Core 1
Core m-1
Core m
SAGA
SAGA
SAGA
SAGA
Combine results


time(s)
# of iteration
for loop
parfor loop
vectorize


f(x)
# of iteration
γ = γ0
γ = 2 γ0
γ = 0.5 * γ0
γ = 0.1 * γ0
γ = 0.05 * γ0

Comparison


×10^5
5.15
5.1
5.05
5
4.95
4.9
4.85
4.8
4.75
4.7
4.65
f(x)
SAGA
Distributed SAGA I
Distributed SAGA II
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
# of iterations
×10^5


×10^5
5.15
5.1
5.05
5
4.95
4.9
4.85
4.8
4.75
4.7
4.65
f(x)
SAGA
Distributed SAGA I
Distributed SAGA II
0
20
40
60
80
100
120
140
160
180
200
time(s)