

Peer Review – Group1_CA2

1) Solve the optimization problem using GD, stochastic GD, SVRG, and SAG.

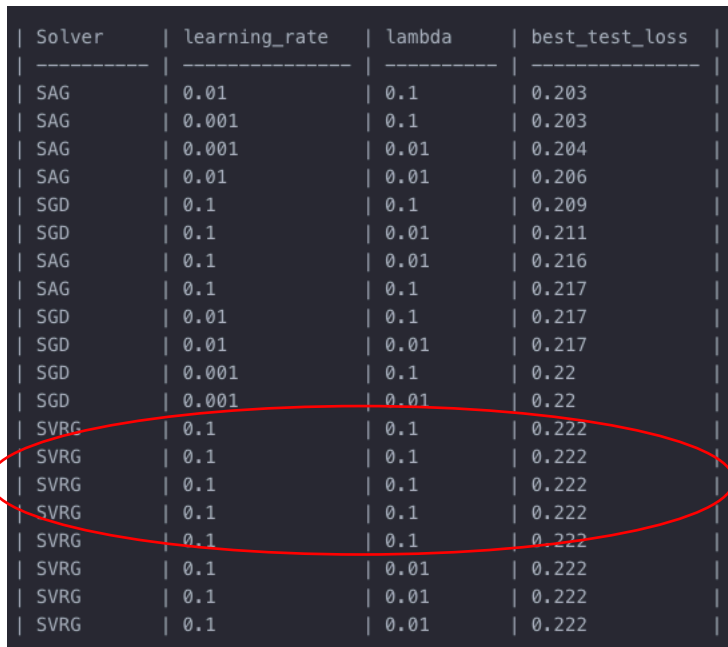
The code is clear and well-structure.

We all feel impressed.

All solvers are correct.

2) Tune a bit hyper-parameters.

- There is one more HP: number of iterations
- In the table, there are some repeated results, we are wonder the reason behind.
- Suggestion: #1. jointly tuning method Gridsearch could tune several HPs together; #2. HP tuning, in order to reduce the error, also could consider cross validation



The table displays the best test loss for various solvers (SAG, SGD, SVRG) across different learning rates and lambda values. A red circle highlights a group of rows where SVRG consistently achieves a best test loss of 0.222, regardless of the learning rate or lambda value.

Solver	learning_rate	lambda	best_test_loss
SAG	0.01	0.1	0.203
SAG	0.001	0.1	0.203
SAG	0.001	0.01	0.204
SAG	0.01	0.01	0.206
SGD	0.1	0.1	0.209
SGD	0.1	0.01	0.211
SAG	0.1	0.01	0.216
SAG	0.1	0.1	0.217
SGD	0.01	0.1	0.217
SGD	0.01	0.01	0.217
SGD	0.001	0.1	0.22
SGD	0.001	0.01	0.22
SVRG	0.1	0.1	0.222
SVRG	0.1	0.1	0.222
SVRG	0.1	0.1	0.222
SVRG	0.1	0.1	0.222
SVRG	0.1	0.1	0.222
SVRG	0.1	0.01	0.222
SVRG	0.1	0.01	0.222
SVRG	0.1	0.01	0.222

3) Compare theses solvers in terms complexity of hyper-parameter tuning, convergence time, convergence rate, and memory requirement.

- The figure is really good, including different solvers and both training set and test set loss. It is very convenient to compare different solvers performance.
- Problem: To our understanding, the figure is based on the same HPs set up, but it shows some confusing results. For example, in figure one, SGD is faster than GD. SGD is a cheaper but faster with, which means it need more iterations, but less computation time. We think SGD should shows slower performance, w.r.t. iterations.