Assignment 3 of MATP6960: programming

(Due together with theoretical questions at 11:59PM on Nov-19-2022)

Instruction: Each student needs to submit the source code file and a report by Latex. Include your solutions to the theoretical questions in the report file. 40% of the total credit will be for the theoretical questions and 60% for the programming. **You will be evaluated based on whether you have all the results in the requirements below, and also the report**. Compress your source files (**DO NOT send the data.**) and PDF report file into a single .zip file, name it as "MATP6960_Assignment3_YourLastNameInitial", and send it to `optimization.rpi@gmail.com`

Problem description

In the class, we introduced the cooperative stochastic approximation (CSA) for solving expectation-constrained problem [1, Algorithm 1], one stochastic gradient method (MSA) for minimax problem [2, Eqn. 3.7], and a primal-dual stochastic gradient method (PDSG) for solving many-constrained problems [3, Algorithm 1]. In this assignment, you are asked to implement these three algorithms to solve the quadratically constrained quadratic programming (QCQP):

$$\min_{\mathbf{x}\in X} \ \frac{1}{2N}\sum_{i=1}^{N}\|\mathbf{H}_i\mathbf{x}-\mathbf{c}_i\|^2, \ \text{s.t.} \ \frac{1}{2}\mathbf{x}^\top\mathbf{Q}_j\mathbf{x}+\mathbf{a}_j^\top\mathbf{x}\le b_j, \ j=1,\ldots,M, \tag{1}$$

where N and M are both large integers, and each $\mathbf{Q}_j$ is a positive semidefinite matrix.

The PDSG can be directly used to solve (1); the CSA method can be applied to solve the equivalent problem

$$\min_{\mathbf{x}\in X} \ \frac{1}{2N}\sum_{i=1}^{N}\|\mathbf{H}_i\mathbf{x}-\mathbf{c}_i\|^2, \ \text{s.t.} \ \frac{1}{M}\sum_{j=1}^{M}\left[\frac{1}{2}\mathbf{x}^\top\mathbf{Q}_j\mathbf{x}+\mathbf{a}_j^\top\mathbf{x}-b_j\right]_+\le 0; \tag{2}$$

and the MSA can be applied to the minimax problem by using the Lagrangian function

$$\min_{\mathbf{x}\in X}\max_{\mathbf{z}} \ \frac{1}{2N}\sum_{i=1}^{N}\|\mathbf{H}_i\mathbf{x}-\mathbf{c}_i\|^2+\frac{1}{M}\sum_{j=1}^{M}z_j\left(\frac{1}{2}\mathbf{x}^\top\mathbf{Q}_j\mathbf{x}+\mathbf{a}_j^\top\mathbf{x}-b_j\right). \tag{3}$$

Requirements

Suppose the data $\{\mathbf{H}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{Q}\}$ involved in (1) are given. You are required to do the follows.

1. Write a solver (in MATLAB or Python) for each of the CSA, MSA, and PDSG methods on solving the QCQP problem. For PDSG, a nonadaptive version and an adaptive version were introduced in class. You are required to implement the adaptive version. Treat the data $\{\mathbf{H}, \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{Q}\}$ and other parameters (like max epoch) as the input of your solver. You can set X to the box constraint set $\{\mathbf{x} \in \mathbb{R}^n : -10 \leq x_i \leq 10, \forall\, i = 1, \ldots, n\}$. You can stop the algorithm simply based on the max epoch. Or you can set a stopping tolerance and check if the violation to the KKT system is below the tolerance. Compute the objective value and constraint violation at the actual iterate and the weighted-averaged iterate after each epoch and return these values as output of your solver.

2. Use the provided file to test your solver. The instructor's implemented adaptive PDSG is provided to approximately compute the optimal objective value. You can also use other methods to compute the optimal objective value and report what method you use. Print the objective error and constraint violation for each method. You need to specify what values you use for the algorithm parameters, including stepsize and minibatch size. [**Remark**: The provided files are written in Matlab. If you code up the solvers in Python, you will need to write a Python test file based on the provided Matlab test file. Also, you can run your solver to enough epochs to obtain an approximate optimal objective value].

3. Discuss what you observe in the test. For example, how do you tune the algorithm parameters, how these parameters affect the performance of these algorithms, which algorithm performs better in term of a certain measure?

Bonus question

You can earn up to 25% extra credits if you also implement the APriD method in Algorithm 1 of [4] and compare to the three methods mentioned above. Try different values of the clipping parameter θ and discuss how it affects the performance of the algorithm.

References

[1] G. Lan and Z. Zhou. Algorithms for stochastic optimization with function or expectation constraints. *Computational Optimization and Applications*, pages 1–38, 2020.

[2] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

[3] Y. Xu. Primal-dual stochastic gradient method for convex programs with many functional constraints. *SIAM Journal on Optimization*, 30(2):1664–1692, 2020.

[4] Y. Yan and Y. Xu. Adaptive primal-dual stochastic gradient method for expectation-constrained convex stochastic programs. *Mathematical Programming Computation*, 14:319–363, 2022.