ROYAL INSTITUTE
OF TECHNOLOGY

EP3260: Machine Learning Over Networks

Computer Assignments

Hossein S. Ghadikolaei

Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Stockholm, Sweden

`https://sites.google.com/view/mlons/home`

January – March 2019

Summary

- ML problems involve a wide ranges of optimization problems with various optimization landscape and constraints. CAs are designed to provide numerical insights on the pros and cons of various algorithmic solutions for ML problems, covered in the course.
- CAs 1 and 2 are from Lecture 3 (Centralized convex ML)
- CA 3 is from Lecture 4 (Centralized nononvex ML)
- CA 4 is from Lecture 5 (Distributed ML)
- CA 5 is from Lectures 6 (ADMM) and 7 (Communication Efficiency)
- CA 6 is from Lecture 8 (DNNs)

CA 1: Closed-form solution vs iterative approaches

Consider $\boldsymbol{x}^\star = \underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\ \frac{1}{N}\sum_{i\in[N]} \|\boldsymbol{w}^T\boldsymbol{x}_i - \boldsymbol{y}_i\|^2 + \lambda\|\boldsymbol{w}\|_2^2$ for dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$.

a) Find a closed-form solution for this problem.

b) Consider "Communities and Crime" dataset ($N = 1994$, $d = 128$) and find the optimal linear regressor from the closed-form expression.

c) Repeat b) for "Individual household electric power consumption" dataset ($N = 2075259$, $d = 9$) and observe the scalability issue of the closed-form expression.

d) How would you address even bigger datasets?

CA 2: Deterministic/stochastic algorithms

Consider logistic ridge regression $f(\boldsymbol{w}) = \frac{1}{N}\sum_{i\in[N]} f_i(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_2^2$ where $f_i(\boldsymbol{w}) = \log\left(1+\exp\{-y_i\boldsymbol{w}^T\boldsymbol{x}_i\}\right)$ for "Individual household electric power consumption" dataset.

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

b) Tune a bit hyper-parameters (including λ).

c) Compare these solvers in terms complexity of hyper-parameter tunning, convergence time, convergence rate (in terms of # outer-loop iterations), and memory requirement.

CA 3: Which algorithm to choose?

Consider optimization problem

$$\underset{\boldsymbol{W}_1,\boldsymbol{W}_2,\boldsymbol{w}_3}{\text{minimize}} \frac{1}{N}\sum\nolimits_{i\in[N]} \|\boldsymbol{w}_3\boldsymbol{s}(\boldsymbol{W}_2\boldsymbol{s}(\boldsymbol{W}_1\boldsymbol{x}_i) - \boldsymbol{y}_i\|_2^2,$$

where $\boldsymbol{s}(\boldsymbol{x}) = 1/(1+\exp(-\boldsymbol{x}))$. You may add your choice of regularizer. Consider both "Communities and crime" and "Individual household electric power consumption" regression datasets.

a) Try to solve this optimization task with proper choices of size of decision variables (matrix $\boldsymbol{W}_1$, matrix $\boldsymbol{W}_2$, and vector $\boldsymbol{w}_3$) using GD, perturbed GD, SGD, SVRG, and block coordinate descent. For the SGD method, you may use the mini-batch version. Notice that you may need t o compute derivative w.r.t a matrix.

b) Compare these solvers in terms complexity of hyper-parameter tunning, convergence time, convergence rate (in terms of # outer-loop iterations), and memory requirement

CA 4: Sensitivity to outliers

Split "MNIST" dataset to 10 random disjoint subsets, each for one worker, and consider SVM classifier in the form of $\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i \in [N]} f_i(\boldsymbol{w})$ with $N = 10$. Consider the following outlier model: each worker i at every iteration independently and randomly with probability p adds a zero-mean Gaussian noise with a large variance R to the information it shares, i.e., ∇f_i and $\boldsymbol{w}_{j,k}$ in the cases of Algorithm 1 and decentralized subgradient method of Lecture 5, respectively.

a) Run decentralized gradient descent (Algorithm 1) with 10 workers.

   Characterize the convergence against p and R.

   Propose an efficient approach to improve the robustness of Algorithm 1 and characterize its convergence against p and R.

b) Consider a two-star topology with communication graph (1,2,3,4)-5-6-(7,8,9,10) and run decentralized subgradient method.

   Characterize the convergence against p and R.

   Propose an efficient approach to improve the robustness to outliers and characterize its convergence against p and R.

c) Assume that we can protect only three workers in the sense that they would always send the true information. Which workers you protect in Algorithm 1 and which in the two-star topology, running decentralized subgradient method?

CA 5 Communication efficiency

Split "MNIST" dataset to 10 random disjoint subsets, each for one worker, and consider SVM classifier in the form of $\min_{\boldsymbol{w}} \frac{1}{N} \sum_{i \in [N]} f_i(\boldsymbol{w})$ with $N = 10$.

a) Run decentralized GD (from Lecture 5) with 10 workers. Characterize the convergence against the total number of signaling exchanges among all nodes, denoted by T.

b) Consider a two-star topology with communication graph (1,2,3,4)-5-6-(7,8,9,10) and run decentralized subgradient method (from Lecture 5) and ADMM over network (from Lecture 6). Characterize the convergence against T. Tune hyperparameters to improve the convergence rate.

c) Propose an approach to reduce T with a marginal impact on the convergence. Do not limit your imaginations and feel free to propose any change or any solution. While being nonsense in some applications, your solution may actually make very sense in some other applications. Discuss pros and cons of your solution and possibly provide numerical evidence that it reduces T.

d) An alternative approach to improve communication-efficiency is to compress the information message to be exchanged (usually gradients – either in primal or dual forms). Consider two compression/quantization methods for a vector: (Q1) keep only K values of a vector and set the rest to zero and (Q2) represent every element with fewer bits (e.g., 4 bits instead of 32 bits).

e) Repeat parts a-b using Q1 and Q2. Can you integrate Q1/Q2 to your solution in part d? Discuss.

f) How do you make SVRG and SAG communication efficient for large-scale ML?

CA 6: Deep neural networks

Consider "MNIST" dataset. Consider a DNN with J layers and $\{N_j\}$ neurons on layer j.

a) Train DNN using SGD and your choices of hyper-parameters, L, and $\{N_j\}_{j\in[J]}$. Report the convergence rate on the training as well as the generalization performance. Feel free to change SGD to any other solver of your choice (give explanation for the choice).

b) Repeat part a with mini-batch GD of your choice of the mini-batch size, retrain DNN, and show the performance measures. Compare the training performance (speed, accuracy) using various adaptive learning rates (constant, diminishing, AdaGrad, RMSProp).

c) Consider design of part a. Fix $\sum_j N_j$. Investigate shallower networks (smaller J) each having potentially more neurons versus deeper network each having fewer neurons per layer, and discuss pros and cons of these two DNN architectures.

d) Split the dataset to 6 random disjoint subsets, each for one worker, and repeat part a on master-worker computational graph.

e) Consider a two-star topology with communication graph (1,2)-3-4-(5,6). Repeat part a using your choice of distributed optimization solver. You can add communication-efficiency to the iterations, if you like!

f) To promote sparse solutions, you may use l_1 regularization or a so-called dropout technique. Explain how you incorporate each of these approaches in the training? Compare their training performance and the size of the final trained models.

g) Improving the smoothness of an optimization landscape may substantially improve the convergence properties of first-order iterative algorithms. Batch-normalization is a relatively simple technique to smoothen the landscape [Santurkar-2018]. Using the materials of the course, propose an alternative approach to improve the smoothness. Provide numerical justification for the proposed approach.

Some references

- L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," SIAM Review, 2018.
- S. Bubeck, "Convex optimization: Algorithms and complexity," FoT in Machine Learning, 2015.
- L. Zhang, M. Mahdavi, and R. Jin, "Linear convergence with condition number independent access of full gradients," NIPS 2013.
- M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," Mathematical Programming, 2017.
- Stephen J. Wright, "Coordinate descent algorithm" , Math. Program., 2015.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," FoT in Machine learning, 2011.
- A. Nedic, A. Olshevsky, and M. G. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," Proceedings of the IEEE, 2018.
- E. Ghadimi, A. Teixeira, I. Shames, and M. johansson, "Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems," IEEE Transactions on Automatic Control, 2015.
- S. Santurkar, D. Tsipras, A. Ilyas, A. Madry, "How does batch normalization help optimization?" NIPS, 2018.