Incremental Gradient Methods

This project is aimed to be a way for us to better understand and thinker with the recent advances in the Stochastic Gradient Descent algorithms, specifically some of the newest Incremental Gradient Methods such as SAG [7], SVRG [4] and SAGA [2]. This class of algorithms have been developed to solve problems of the form

$$\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(x) + h(x), \tag{1}$$

where each f_i is convex and has Libschitz continuous derivatives with constant L or is strongly convex with constant μ; and h is a convex but potentially non-differentiable function (his proximal operator is however easy to compute). While computing the full gradient would be prohibitive due to large d and n, these iterative stochastic algorithms reduce the computational cost of optimization by only computing the gradient of a subset of the functions f_i at each step.

Many machine learning problems can be cast in (1), such as (constrained) Least-Square or Logistic Regressions with ℓ_1 or ℓ_2 regularization; where x would represent the model parameters, f_i the data fidelity term applied to a particular sample i, and h a regularization or indicator function of a convex set. As such, these methods are of use in our respective domains of expertise: Signal Processing on Graphs and Risk Analytics.

With the general setting in mind, we identify four directions relevant to our research in which we could contribute:

1. Play with the trade-off between the computational efficiency of SAGA and the memory efficiency of SVRG, especially relevant when working with large datasets, e.g. for $n > 10^6$ which is not uncommon in these days of Big Data. A first approach to compromise on the memory requirement of SAGA would be to store averaged gradients over mini-batches instead of the full gradient matrix. This task will involve the implementation and empirical testing of the devised scheme. A novel proof of convergence can be envisioned. This work is related to [6].
2. A distributed implementation of one of those algorithms. This would be useful to diminish the clock time needed to solve a given problem or to solve large-scale optimizations where the memory of one computer is not sufficient anymore. This goal will require the analysis of the inter-nodes communication cost as well as the design of a merging or synchronization scheme. Novel proofs of convergence could be required. It could be inspired by [1].
3. Explore the application of these algorithms to minimax problems which aim at finding saddle points [5]. The min-max formulation appears in the context of zero-sum games and robust optimization. Traditionally, robust optimization problems focus on converting the minimax problem to a minimization problem by leveraging duality theory. Instead, we aim to find the saddle points using incremental methods.
4. Use these methods to fit statistical models. In particular, we are interested to fit a Gaussian Mixture Model (GMM) viewed as a manifold optimization problem. Our goal would be to adapt one of the incremental methods to fit GMMs [3].

We do not expect to complete all of the above objectives. We plan to discuss with experts in the domain[1] and will then choose two of them to focus on two of them only.

[1]Such as the first author of [3], whom Soroosh met during his master studies. Or someone from the EPFL LIONS lab.

Roles. Each of us will pursue one of the mentioned goals from beginning to end; which includes any necessary theory, implementation, testing, writing and presentation. Our work (code, report and presentation) will be tracked by *git*, such that individual contributions can easily be spotted.

Milestones. Following are the milestones we envision for the completion of the aforementioned project.

- 2016-03-24 Proposal submitted.
- 2016-04-01 Proposal approved.
- 2016-04-08 Two directions chosen.
- 2016-04-22 Problems stated and solutions formulated.
- 2016-05-06 Solutions implemented (Jupyter notebooks, Python).
- 2016-05-20 Tested on real or synthetic data.
- 2016-05-27 Report written.
- 2016-06-03 Project presented.

References

[1] Pascal Bianchi, Walid Hachem, and Franck Iutzeler. "A Coordinate Descent Primal-Dual Algorithm and Application to Distributed Asynchronous Optimization". In: (2014). arXiv: 1407.0898.

[2] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. "SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives". In: *Advances in Neural Information Processing Systems 27*. 2014, pp. 1646–1654. URL: http://papers.nips.cc/paper/5258-saga-a-fast-incremental-gradient-method-with-support-for-non-strongly-convex-composite-objectives.pdf.

[3] Reshad Hosseini and Suvrit Sra. "Matrix Manifold Optimization for Gaussian Mixtures". In: *Advances in Neural Information Processing Systems 28*. 2015, pp. 910–918. URL: http://papers.nips.cc/paper/5812-matrix-manifold-optimization-for-gaussian-mixtures.pdf.

[4] Rie Johnson and Tong Zhang. "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction". In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 315–323. URL: http://papers.nips.cc/paper/4937-accelerating-stochastic-gradient-descent-using-predictive-variance-reduction.pdf.

[5] Arkadi Nemirovski et al. "Robust stochastic approximation approach to stochastic programming". In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1574–1609. URL: http://www.eecs.berkeley.edu/~brecht/cs294docs/week1/09.Nemirovski.pdf.

[6] Atsushi Nitanda. "Stochastic proximal gradient descent with acceleration techniques". In: *Advances in Neural Information Processing Systems*. 2014, pp. 1574–1582. URL: https://papers.nips.cc/paper/5610-stochastic-proximal-gradient-descent-with-acceleration-techniques.pdf.

[7] Mark Schmidt, Nicolas Le Roux, and Francis Bach. "Minimizing Finite Sums with the Stochastic Average Gradient". In: (Sept. 10, 2013). arXiv: 1309.2388.