Accelerating the Gradient Projection Iterative Sketch for large scale constrained Least-squares

Junqi Tang, Mohammad Golbabaee, Mike Davies
Instituttte of Digital Communications, The University of Edinburgh, EH9 3JE, UK

***Abstract*—This paper proposes an accelerated sketched gradient method [1] which was based on equipping a combination of the meta-algorithms *Classical Sketch* (CS) [2] and *Iterative Hessian Sketch* (IHS) [3] with the *Projected / Proximal Gradient Descent* (PGD) algorithm and Nesterov's acceleration scheme for efficiently solving large scale constrained Least-squares and regularized Least-squares. As a first order solver, the PGD can provide us flexibility in handling the constraints and scalability in computation. The proposed algorithm satisfies a number of our expectations as an efficient large scale constrained/regularized LS solver, which are mainly inherited from the scalability and flexibility of the PGD combined with dimensionality reducing properties of the sketching techniques: (a) computational efficiency, (b) efficiency on high speed storage, and (c) flexibly to incorporate a wide range of constraints and non-smooth regularization.**

I. Introduction

Consider a noisy linear measurement model for a vector x_{gt} (ground truth) which belongs to a convex constrained set $\mathcal{K}$, an n by d linear operator matrix A, and additive noise denoted by $w \in \mathcal{R}^{n\times 1}$:

$$y = Ax_{gt} + w, \quad x_{gt} \in \mathcal{K}, \quad A \in \mathcal{R}^{n\times d}. \tag{1}$$

In the context of imaging applications such as CT or MRI, the vector y denotes a set of n physical measurements collected from an image x_{gt} through the measurement operator A, and in the context of machine learning, A is often a training data matrix used for setting the regression parameters x_{gt} from the observations y. The Least-square (LS) estimator for x_{gt} is:

$$x^\star = \arg\min_x \|y - Ax\|_2^2 + f_{\mathcal{K}}(x), \tag{2}$$

where the convex (could be non-smooth) function $f_{\mathcal{K}}$ enforces the constraint into the Least-squares estimator. If the constraint is exactly known, the $f_{\mathcal{K}}$ can be set as the indicator function of the set $\mathcal{K}$, if not, we can set it as a regularizer.

II. Sketched gradient with Nesterov's Acceleration scheme

A standard first order solver for (2) is the PGD algorithm which can be defined for any convex constrained set $\mathcal{K}$, as long as the projection (or proximal operation) onto the set is efficient:

$$x_{j+1} = Prox_{f_{\mathcal{K}}}(x_j - \eta A^T(Ax_j - y)). \tag{3}$$

The PGD is known to be flexible to various constraint sets, but it faces two major challenges: 1) when the operator A is large, the computational cost of the iterates can be large; 2) when A is ill-conditioned, the PGD may take a very large number of iterations to converge. Moreover when the computational cost of the projection/proximal operator is non-trivial, we also wish to reduce the number of iterations as much as possible (the stochastic gradient algorithms usually demands a small batch size which will lead to a large number of iterations). The proposed algorithm is aimed at tackling both reducing the cost of the gradient calculation and the number of iterations.

Algorithm 1:

Initialization: $p_0^t = 1$ for all t, $x_0^0 = 0$, $z_0^0 = 0$;
Given $A \in \mathcal{R}^{n\times d}$, sketch size $m \ll n$;
Generate a random sketching matrix $S^0 \in \mathcal{R}^{m\times n}$;
Calculate $S^0 A$, $S^0 y$;
while $i = 0 : k_0 - 1$ **do**
  $x_{i+1}^0 = Prox_{f_{\mathcal{K}}}(z_i^0 - \eta(S^0 A)^T(S^0 A z_i^0 - S^0 y))$;
  $p_{i+1}^0 = \frac{-(p_i^0)^2 + \sqrt{(p_i^0)^4 + 4(p_i^0)^2}}{2}$;
  $\tau_{i+1}^0 = \frac{p_i^0(1-p_i^0)}{(p_i^0)^2 + p_{i+1}^0}$;
  $z_{i+1}^0 = x_{i+1}^0 + \tau_{i+1}^0(x_{i+1}^0 - x_i^0)$;
end
$x_0^1 = z_0^1 = x_{k_0}^0$;
while $t = 1 : N - 1$ **do**
  Calculate $g = A^T(Ax_0^t - y)$;
  Generate a random sketching matrix $S^t \in \mathcal{R}^{m\times n}$;
  Calculate $A_s^t = S^t A$;
  while $i = 0 : k_t - 1$ **do**
    $x_{i+1}^t = Prox_{f_{\mathcal{K}}}(z_i^t - \eta({A_s^t}^T A_s^t(z_i^t - x_0^t) + mg))$;
    $p_{i+1}^t = \frac{-(p_i^t)^2 + \sqrt{(p_i^t)^4 + 4(p_i^t)^2}}{2}$;
    $\tau_{i+1}^t = \frac{p_i^t(1-p_i^t)}{(p_i^t)^2 + p_{i+1}^t}$;
    $z_{i+1}^t = x_{i+1}^t + \tau_{i+1}^t(x_{i+1}^t - x_i^t)$;
  end
  $x_0^{t+1} = z_0^{t+1} = x_k^t$;
end
Return x_k^N;

As shown in the Algorithm 1, we use the classical sketching [2] and iterative sketching [3] framework as we have done in the [1], and then since the sketched least-square problem is fixed in every outer loop, the *Nesterov's acceleration scheme* [4][5] is in theory directly applicable to provide acceleration in both strict constrained setting and proximal setting.

We have also tested its performance through numerical experiments, and observe that the proposed algorithm achieves a further speed-up onto the GPIS / GPIS-prox algorithm in all the experiments.

Acknowledgment

Junqi Tang would like to acknowledge the support from H2020-MSCA-ITN Machine Sensing Training Network (MacSeNet), project 642685

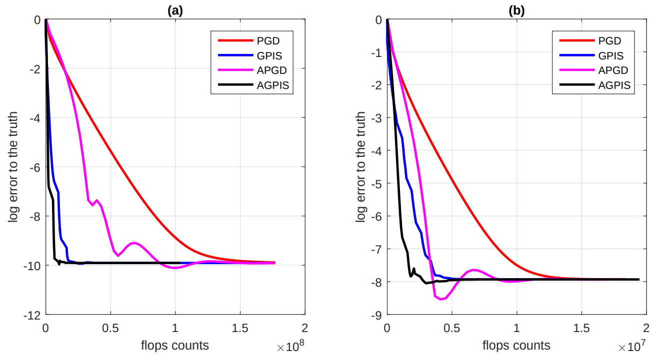

Fig. 1. Experimental results on a synthetic l_1 constrained Least-square regression problem

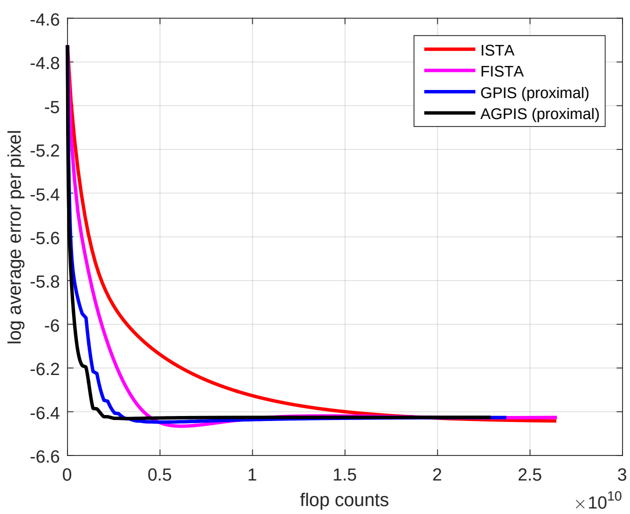

Fig. 2. Experimental results on a fan-beam CT image reconstruction (Regularized least-squares)

References

[1] Junqi Tang, Mohammad Golbabaee, and Mike Davies, "Gradient projection iterative sketch for large scale constrained least-squares," *arXiv preprint arXiv:1609.09419*, 2016.

[2] Mert Pilanci and Martin J Wainwright, "Randomized sketches of convex programs with sharp guarantees," *Information Theory, IEEE Transactions on*, vol. 61, no. 9, pp. 5096–5115, 2015.

[3] Mert Pilanci and Martin J Wainwright, "Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares," *Journal of Machine Learning Research*, vol. 17, no. 53, pp. 1–38, 2016.

[4] Yurii Nesterov et al., "Gradient methods for minimizing composite objective function," Tech. Rep., UCL, 2007.

[5] Yurii Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, 2013.