Regularized Nonlinear Acceleration

Damien Scieur
INRIA & D.I., UMR 8548,
École Normale Supérieure, Paris, France.
damien.scieur@inria.fr

Alexandre d'Aspremont
INRIA & D.I., UMR 8548,
École Normale Supérieure, Paris, France.
aspremon@di.ens.fr

Francis Bach
INRIA & D.I., UMR 8548,
École Normale Supérieure, Paris, France.
francis.bach@inria.fr

***Abstract*—We describe a convergence acceleration technique for generic optimization problems. Our scheme computes estimates of the optimum from a nonlinear average of the iterates produced by any optimization method. The weights in this average are computed via a simple and small linear system, whose solution can be updated online.**

I. Introduction

Suppose we want to solve the following optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

in the variable $x \in \mathbb{R}^n$, where $f(x)$ is strongly convex with respect to the Euclidean norm with parameter μ, and has a Lipschitz continuous gradient with parameter L with respect to the same norm. Assume we solve this problem using an iterative algorithm of the form

$$x_{i+1} = g(x_i), \quad \text{for } i = 1, ..., N, \tag{2}$$

where $x_i \in \mathbb{R}^n$ and N the number of iterations and g the algorithm, with fixed-point x^*. Here, we will focus on the problem of estimating the solution to (1) (which is also x^*) by tracking only the sequence of iterates x_i produced by g. This will lead to an acceleration of the speed of convergence, since we will be able to extrapolate more accurate solutions without any calls to the oracle $g(x)$.

II. Minimal polynomial extrapolation

The main idea of Minimal Polynomial Extrapolation (MPE, see [1], [2]) algorithm is to accelerate the linear version of algorithm (2),

$$x_i - x^* = A(x_{i-1} - x^*) + r(x_i) = A^i(x_0 - x^*) + r(x_i) \tag{3}$$

where x^* is the fixed point of g, $r(x) = O(\|x - x^*\|^2)$ and $A = g'(x^*)$, the Jacobian of g at x^*. If we drop $r(x_i)$ and average the N first equations (3) with coefficients c_i (with unitary sum), we obtain

$$\textstyle\sum_{i=0}^N c_i x_i - x^* = \sum_{i=0}^N c_i A^i(x_0 - x^*) = p(A)(x_0 - x^*) \tag{4}$$

where $p(A)$ is a matrix polynomial where $p(1) = 1$. The goal of MPE is to minimize this polynomial using only the sequence $\{x_i\}$ (meaning we cannot acces to A). However, the differences follow

$$x_{i+1} - x_i = (x_{i+1} - x^*) - (x_i - x^*) = (A - I)(x_i - x^*), \quad i = 1 \dots N.$$

Their combination with coefficient c_i thus becomes

$$\textstyle\sum_{i=0}^N c_i(x_{i+1} - x_i) = (A - I)p(A)(x_0 - x^*)$$

The previous equation means that if the combination of differences is small, so the polynomial is also small, so the weighted mean (4) is close to the solution x^*. Let $U = [..., x_{i+1} - x_i, ...]$ the matrix of differences. The MPE algorithm solves

$$c^* = \arg\min_c \|Uc\|_2 = (U^TU)^{-1}\mathbf{1}_N / (\mathbf{1}_N^T(U^TU)^{-1}\mathbf{1}_N), \tag{5}$$

where $\mathbf{1}_N$ is a vector of N ones, then returns $x_{\text{extr}} = \sum_{i=1}^N c_i^* x_i$. A important advantage of this method is its complexity: if $n \gg N$ (with n the dimension of the space), the complexity is linear in the dimension. Even better, if the vectors are p-sparse, then the complexity thus becomes $O(p)$.

III. Regularized minimal polynomial extrapolation

The main problem of MPE is the inversion of matrix U^TU of size $N \times N$. Even if its size is small (in numerical experiments, N is typically 5), its condition number is extremely large. Thus, when some perturbations is added in the system the impact on the solution c^* is huge. The perturbations come, for example, from the remainder in (3). This algorithm has been shown to be extremely unstable in practice (see Figure 2). The Regularized MPE (RMPE) algorithm thus solves a similar problem, with Tikhonov regularization

$$c^* = \arg\min_c \|Uc\|_2^2 + \lambda\|c\|_2^2$$

whose solution is also computed via a linear system, similar to (5). This regularization term is able to control the impact of perturbations, leading to a significant improvement in terms of performances. The regularization also allows to derive an upper bound on the performances of the method (see [3]), which appears to be asymptotically optimal (if applied with gradient method on strongly, smooth functions with Lipschitz-continuous Hessian).

IV. Numerical performances and Conclusion

In [3], the RMPE-N method (applied to gradient method) is compared to the Nesterov's accelerated gradient method [4]. The RMPE-N is a variant of RMPE, which restart the extrapolation each N steps (more details in [3]) and find λ using inexact line search. In Figures 1 and 3 is reported the optimization of the logistic regression, on Madelon and Sido0 datasets. We see that without any information on the strong-convexity constant of the function (unlike Nesterov's accelerated gradient), the RMPE-N algorithm outperforms the other methods used. Moreover, even without any theoretical guarantee, RMPE-N performs well also on non-smooth functions, for example on MAXCUT problem (see Figure 4). In all of these examples, we see that the post-processing step works well, and the impact of the complexity of computing the non-linear average on the conputation time is completely marginal.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7-PEOPLE-2013-ITN) under grant agreement n° 607290 SpaRTaN

References

[1] S. Cabay and L. Jackson, "A polynomial extrapolation method for finding limits and antilimits of vector sequences," *SIAM Journal on Numerical Analysis*, vol. 13, no. 5, pp. 734–752, 1976.

[2] M. Mešina, "Convergence acceleration for the iterative solution of the equations x= ax+ f," *Computer Methods in Applied Mechanics and Engineering*, vol. 10, no. 2, pp. 165–173, 1977.

[3] D. Scieur, A. d'Aspremont, and F. Bach, "Regularized nonlinear acceleration," in *Advances In Neural Information Processing Systems*, pp. 712–720, 2016.

[4] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Springer, 2003.

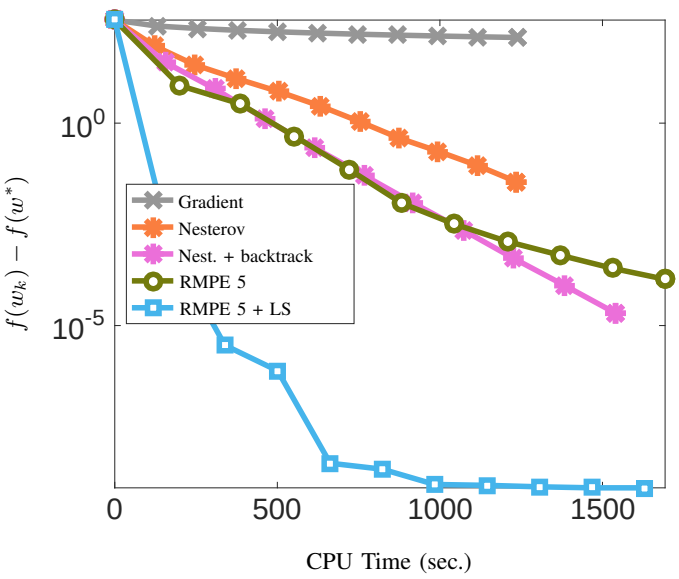

Fig. 1. Solving logistic regression on Madelon dataset (500 features, 4400 data points, condition number = $1.5 \cdot 10^5$). Here, RMPE is used on gradient method.

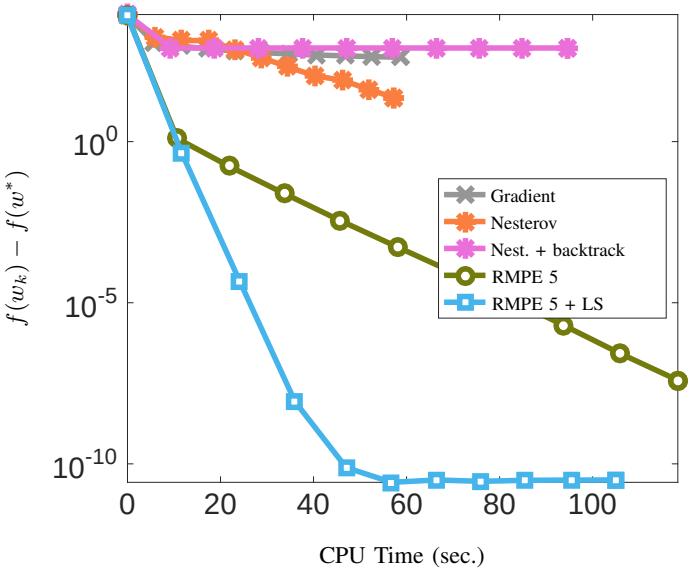

Fig. 3. Solving Logistic regression on sido0 dataset (4932 features, 12678 data points, condition number = $1.5 \cdot 10^5$). Here, RMPE is used on gradient method.

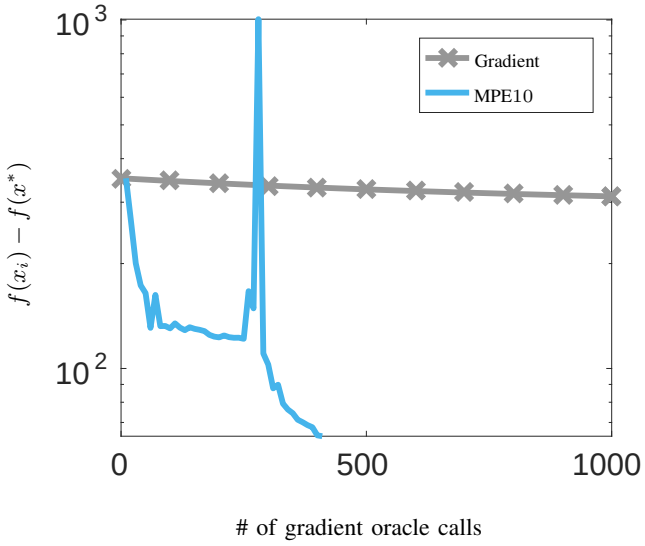

Fig. 2. Application of MPE (non-regularized acceleration) for solving logistic regression on Madelon dataset. We see that MPE looks unstable at the beginning, then completely diverge after 500 iterations.

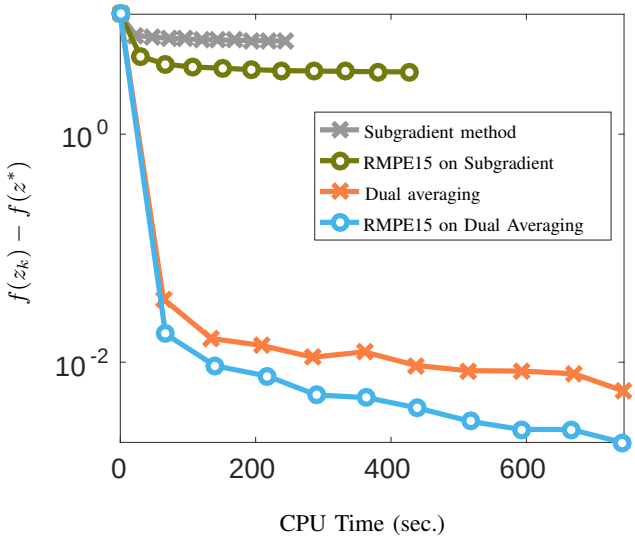

Fig. 4. Solving MAXCUT (dual) problem using algorithms for non-smooth functions. Even if the improvement of RMPE is not as impressive as in Figures 1 and 3, the speed of convergence becomes more or less twice faster. Here, RMPE is used on sub-gradient and dual averaging methods.