Regularized Nonlinear Acceleration

Damien Scieur, Alexandre D'Aspremont, Francis Bach

Regularized Nonlinear Acceleration

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

Alexandre d'Aspremont
CNRS & 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. This acceleration scheme runs in parallel to the base algorithm, providing improved estimates of the solution on the fly, while the original optimization method is running. Numerical experiments are detailed on classical classification problems.

1 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. This class of function is often encountered, for example in regression where $f(x)$ is of the form

$$f(x) = \mathcal{L}(x) + \Omega(x),$$

where $\mathcal{L}(x)$ is a smooth convex loss function and $\Omega(x)$ is a smooth strongly convex penalty function.

Assume we solve this problem using an iterative algorithm of the form

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

where $x_i \in \mathbb{R}^n$ and k the number of iterations. Here, we will focus on the problem of estimating the solution to (1) by tracking only the sequence of iterates x_i produced by an optimization algorithm. 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 $g(x)$.

Since the publication of Nesterov's optimal first-order smooth convex minimization algorithm [1], a significant effort has been focused on either providing more explicit interpretable views on current acceleration techniques, or on replicating these complexity gains using different, more intuitive schemes. Early efforts were focused on directly extending the original acceleration result in [1] to broader function classes [2], allow for generic metrics, line searches or simpler proofs [5, 6], produce adaptive accelerated algorithms [7], etc. More recently however, several authors [8, 9] have started

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

using classical results from control theory to obtain numerical bounds on convergence rates that match the optimal rates. Others have studied the second order ODEs obtained as the limit for small step sizes of classical accelerated schemes, to better understand their convergence [10, 11]. Finally, recent results have also shown how to wrap classical algorithms in an outer optimization loop, to accelerate convergence [12] and reach optimal complexity bounds.

Here, we take a significantly different approach to convergence acceleration stemming from classical results in numerical analysis. We use the iterates produced by any (converging) optimization algorithm, and estimate the solution directly from this sequence, assuming only some regularity conditions on the function to minimize. Our scheme is based on the idea behind Aitken's Δ^2 algorithm [13], generalized as the Shanks transform [14], whose recursive formulation is known as the ε-algorithm [15] (see e.g. [16, 17] for a survey). In a nutshell, these methods fit geometrical models to linearly converging sequences, then extrapolate their limit from the fitted model.

In a sense, this approach is more statistical in nature. It assumes an approximately linear model holds for iterations near the optimum, and estimates this model using the iterates. In fact, Wynn's algorithm [15] is directly connected to the Levinson-Durbin algorithm [18, 19] used to solve Toeplitz systems recursively and fit autoregressive models (the Shanks transform solves Hankel systems, but this is essentially the same problem [20]). The key difference here is that estimating the autocovariance operator is not required, as we only focus on the limit. Moreover, the method presents strong links with the conjugate gradient when applied to unconstrained quadratic optimization.

We start from a slightly different formulation of these techniques known as minimal polynomial extrapolation (MPE) [17, 21] which uses the minimal polynomial of the linear operator driving iterations to estimate the optimum by nonlinear averaging (i.e., using weights in the average which are nonlinear functions of the iterates). So far, for all the techniques cited above, no proofs of convergence of these estimates were given in the case where the iterates made the estimation process unstable.

Our contribution here is to add a regularization in order to produce explicit bounds on the distance to optimality by controlling the stability through the regularization parameter, thus explicitly quantifying the acceleration provided by these techniques. We show in several numerical examples that these stabilized estimates often speed up convergence by an order of magnitude. Furthermore this acceleration scheme thus runs in parallel to the original algorithm, providing improved estimates of the solution on the fly, while the original method is progressing.

The paper is organized as follows. In section 2.1 we recall basic results behind MPE for linear iterations and we will introduce in section 2.2 a formulation of the approximate version of MPE and make a link with the conjugate gradient method. Then, in section 2.3, we generalize these results to generic nonlinear iterations and show, in section 2.4, how to fully control the impact of nonlinearity. We use these results to derive explicit bounds on the acceleration performance of our estimates.

2 Approximate Minimal Polynomial Extrapolation

In what follows, we recall the key arguments behind *minimal polynomial extrapolation (MPE)* as derived in [22] or also [21]. We also explain a variant called *approximate minimal polynomial extrapolation (AMPE)* which allows to control the number of iterates used in the extrapolation, hence reduces its computational complexity. We begin by a simple description of the method for linear iterations, then extend these results to the generic nonlinear case. Finally, we fully characterize the acceleration factor provided by a regularized version of AMPE, using regularity properties of the function $f(x)$, and the result of a Chebyshev-like, tractable polynomial optimization problem.

2.1 Linear Iterations

Here, we assume that the iterative algorithm in (2) is in fact linear, with

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

where $A \in \mathbb{R}^{n\times n}$ (not necessarily symmetric) and $x^* \in \mathbb{R}^n$. We assume that 1 is not an eigenvalue of A, implying that (3) admits a unique fixed point x^*. Moreover, if we assume that $\|A\|_2 < 1$, then x_k converge to x^* at rate $\|x_k - x^*\|_2 \leq \|A\|_2^k \|x_0 - x^*\|$. We now recall the *minimal polynomial extrapolation* (MPE) method as described in [21], starting with the following definition.

Definition 2.1 *Given $A \in \mathbb{R}^{n\times n}$, s.t. 1 is not an eigenvalue of A and $v \in \mathbb{R}^n$, the minimal polynomial of A with respect to the vector v is the lowest degree polynomial $p(x)$ such that*

$$p(A)v = 0, \qquad p(1) = 1.$$

Note that the degree of $p(x)$ is always less than n and the condition $p(1) = 1$ makes p unique. Notice that because we assumed that 1 is not an eigenvalue of A, having $p(1) = 1$ is not restrictive since we can normalize each minimal polynomial with the sum of its coefficients (see Lemma A.1 in the supplementary material). Given an initial iterate x_0, MPE starts by forming a matrix U whose columns are the increments $x_{i+1} - x_i$, with

$$u_i = x_{i+1} - x_i = (A - I)(x_i - x^*) = (A - I)A^i(x_0 - x^*). \tag{4}$$

Now, let p be the minimal polynomial of A with respect to the vector u_0 (where p has coefficients c_i and degree d), and $U = [u_0, u_1, ..., u_d]$. So

$$\textstyle\sum_{i=0}^d c_i u_i = \sum_{i=0}^d c_i A^i u_0 = p(A)u_0 = 0\ , \qquad p(1) = \sum_{i=0}^d c_i = 1. \tag{5}$$

We can thus solve the system $Uc = 0$, $\sum_i c_i = 1$ to find p. In this case, the fixed point x^* can be computed *exactly* as follows

$$\begin{aligned} 0 = \textstyle\sum_{i=0}^d c_i A^i u_0 &= \textstyle\sum_{i=0}^d c_i A^i (A - I)(x_0 - x^*) \\ &= (A - I)\textstyle\sum_{i=0}^d c_i A^i (x_0 - x^*) = (A - I)\sum_{i=0}^d c_i (x_i - x^*). \end{aligned}$$

Hence, using the fact that 1 is not an eigenvalue of A and $p(1) = 1$,

$$(A - I)\textstyle\sum_{i=0}^d c_i (x_i - x^*) = 0 \quad \Leftrightarrow \quad \sum_{i=0}^d c_i (x_i - x^*) = 0 \quad \Leftrightarrow \quad \sum_{i=0}^d c_i x_i = x^*.$$

This means that x^* is obtained by *averaging* iterates using the coefficients in c. The averaging in this case is called nonlinear, since the coefficients of c vary with the iterates themselves.

2.2 Approximate Minimal Polynomial Extrapolation (AMPE)

Suppose now that we only compute a fraction of the iterates x_i used in the MPE procedure. While the number of iterates k might be smaller than the degree of the minimal polynomial of A with respect to u_0, we can still try to make the quantity $p_k(A)u_0$ small, where $p_k(x)$ is now a polynomial of degree at most k. The corresponding difference matrix $U = [u_0, u_1, ..., u_k] \in \mathbb{R}^{n\times(k+1)}$ is rectangular.

This is also known as the Eddy-Mešina method [3, 4] or reduced rank extrapolation with arbitrary k (see [21, §10]). The objective here is similar to (5), but the system is now overdetermined because $k < \deg(P)$. We will thus choose c to make $\|Uc\|_2 = \|p(A)u_0\|_2$, for some polynomial p such that $p(1) = 1$, as small as possible, which means solving for

$$c^* \triangleq \operatorname{argmin}\ \|Uc\|_2 \quad \text{s.t. } \mathbf{1}^T c = 1 \tag{AMPE}$$

in the variable $c \in \mathbb{R}^{k+1}$. The optimal value $\|Uc^*\|_2$ of this problem is decreasing with k, satisfies $\|Uc^*\|_2 = 0$ when k is greater than the degree of the minimal polynomial, and controls the approximation error in x^* with equation (4). Setting $u_i = (A - I)(x_i - x^*)$, we have

$$\begin{aligned} \|\textstyle\sum_{i=0}^k c_i^* x_i - x^*\|_2 &= \|(I - A)^{-1}\textstyle\sum_{i=0}^k c_i^* u_i\|_2 \\ &\leq \left\|(I - A)^{-1}\right\|_2 \|Uc^*\|_2. \end{aligned}$$

We can get a crude bound on $\|Uc^*\|_2$ from Chebyshev polynomials, using only an assumption on the range of the spectrum of the matrix A. Assume A symmetric, $0 \preceq A \preceq \sigma I \prec I$ and $\deg(p) \leq k$. Indeed,

$$\|Uc^*\|_2 = \|p^*(A)u_0\|_2 \leq \|u_0\|_2 \min_{p:p(1)=1} \|p(A)\|_2 \leq \|u_0\|_2 \min_{p:p(1)=1} \max_{A:0\preceq A\preceq\sigma I} \|p(A)\|_2, \tag{6}$$

where p^* is the polynomial with coefficients c^*. Since A is symmetric, we have $A = Q\Lambda Q^T$ where Q is unitary. We can thus simplify the objective function:

$$\max_{A:0\preceq A\preceq\sigma I} \|p(A)\|_2 = \max_{\Lambda:0\preceq\Lambda\preceq\sigma I} \|p(\Lambda)\|_2 = \max_{\Lambda:0\preceq\Lambda\preceq\sigma I} \max_i |p(\lambda_i)| = \max_{\lambda:0\leq\lambda\leq\sigma} |p(\lambda)|.$$

We thus have

$$\|Uc^*\|_2 \leq \|u_0\|_2 \min_{p:p(1)=1} \max_{\lambda:0\leq\lambda\leq\sigma} |p(\lambda)|.$$

So we must find a polynomial which takes small values in the interval $[0, \sigma]$. However, Chebyshev polynomials are known to be polynomials for which the maximal value in the interval $[0, 1]$ is the smallest. Let C_k be the Chebyshev polynomial of degree k. By definition, $C_k(x)$ is a monic polynomial[1] which solves

$$C_k(x) = \operatorname*{argmin}_{p:p \text{ is monic}} \max_{x:x\in[-1,1]} |p(x)|.$$

We can thus use a variant of $C_k(x)$ in order to solve the minimax problem

$$\min_{p:p(1)=1} \max_{\lambda:0\leq\lambda\leq\sigma} |p(\lambda)|. \tag{7}$$

The solution of this problem is given in [23] and admits an explicit formulation:

$$\mathcal{T}(x) = \frac{C_k(t(x))}{C_k(t(1))}, \quad t(x) = \frac{2x-\sigma}{\sigma}.$$

Note that $t(x)$ is simply a linear mapping from interval $[0, \sigma]$ to $[-1, 1]$. Moreover,

$$\min_{p:p(1)=1} \max_{\lambda:0\leq\lambda\leq\sigma} |p(\lambda)| = \max_{\lambda:0\leq\lambda\leq\sigma} |T_k(\lambda)| = |T_k(\sigma)| = \frac{2\zeta^k}{1+\zeta^{2k}}, \tag{8}$$

where ζ is

$$\zeta = (1-\sqrt{1-\sigma})/(1+\sqrt{1-\sigma}) < \sigma. \tag{9}$$

Since $\|u_0\|_2 = \|(A-I)(x_0-x^*)\|_2 \leq \|A-I\|_2\|x_0-x^*\|$, we can bound (6) by

$$\|Uc^*\|_2 \leq \|u_0\|_2 \min_{p:p(1)=1} \max_{\lambda:0\leq\lambda\leq\sigma} |p(\lambda)| \leq \|A-I\|_2 \frac{2\zeta^k}{1+\zeta^{2k}} \|x_0-x^*\|_2.$$

This leads to the following proposition.

Proposition 2.2 *Let A be symmetric, $0 \preceq A \preceq \sigma I \prec I$ and c_i be the solution of* (AMPE)*. Then*

$$\left\| \sum_{i=0}^k c_i^* x_i - x^* \right\|_2 \leq \kappa(A-I) \frac{2\zeta^k}{1+\zeta^{2k}} \|x_0-x^*\|_2, \tag{10}$$

where $\kappa(A-I)$ is the condition number of the matrix $A-I$ and ζ is defined in (9)*.*

Note that, when solving quadratic optimization problems, the rate in this bound matches that obtained using the optimal method in [6]. Also, the bound on the rate of convergence of this method is exactly the one of the conjugate gradient with an additional factor $\kappa(A-I)$.

Remark: This method presents a strong link with the conjugate gradient. Denote $\|v\|_B = \sqrt{v^T B v}$ the norm induced by the definite positive matrix B. By definition, at the k-th iteration, the conjugate gradient computes an approximation s of x^* which follows

$$s = \operatorname{argmin} \|x-x^*\|_A \quad \text{s.t. } x \in \mathcal{K}_k\,,$$

where $\mathcal{K}_k = \{Ax^*, A^2x^*, ..., A^kx^*\}$ is called a Krylov subspace. Since $x \in \mathcal{K}_k$, we have that x is a linear combination of the element in $\mathcal{K}_k$, so $x = \sum_{i=1}^k c_i A^i x^* = q(A)x^*$, where $q(x)$ is a polynomial of degree k and $q(0) = 0$. So conjugate gradient solves

$$s = \operatorname{argmin}_{q:q(0)=0} \|q(A)x^* - x^*\|_A = \operatorname{argmin}_{\hat{q}:\hat{q}(0)=0} \|\hat{q}(A)x^*\|_A,$$

which is very similar to equation (AMPE). However, while conjugate gradient has access to an oracle which gives the result of the product between matrix A and any vector v, the AMPE procedure can only use the iterations produced by (3) (meaning that the AMPE procedure does not need to know A). Moreover, we analyze the convergence of AMPE in another norm ($\|\cdot\|_2$ instead of $\|\cdot\|_A$). These two reasons explain why a condition number appears in the rate of convergence of AMPE (10).

[1] A monic polynomial is a univariate polynomial in which the coefficient of highest degree is equal to 1.

2.3 Nonlinear Iterations

We now go back to the general case where the iterative algorithm is nonlinear, with

$$\tilde{x}_{i+1} = g(\tilde{x}_i), \quad \text{for } i = 1, ..., k, \tag{11}$$

where $\tilde{x}_i \in \mathbb{R}^n$ and the function g has a symmetric Jacobian at point x^*. We also assume that the method has a unique fixed point written x^* and linearize these iterations around x^*, to get

$$\tilde{x}_i - x^* = A(\tilde{x}_{i-1} - x^*) + e_i, \tag{12}$$

where A is now the Jacobian matrix (i.e., the first derivative) of g taken at the fixed point x^* and $e_i \in \mathbb{R}^n$ is a second order error term $\|e_i\|_2 = O(\|\tilde{x}_{i-1} - x^*\|_2^2)$. Note that, by construction, the linear and nonlinear models share the same fixed point x^*. We write x_i the iterates that would be obtained using the asymptotic linear model (starting at x_0)

$$x_i - x^* = A(x_{i-1} - x^*).$$

Running the algorithm described in (11), we thus observe the iterates $\tilde{x}_i$ and build $\tilde{U}$ from their differences. As in (AMPE) we then compute $\tilde{c}$ using matrix $\tilde{U}$ and finally estimate

$$\tilde{x}^* = \textstyle\sum_{i=0}^k \tilde{c}_i \tilde{x}_i.$$

In this case, our estimate for x^* is based on the coefficient $\tilde{c}$, computed using the iterates $\tilde{x}_i$. We will now decompose the error made by the estimation by comparing it with the estimation which comes from the linear model:

$$\left\|\textstyle\sum_{i=0}^k \tilde{c}_i \tilde{x}_i - x^*\right\|_2 \leq \left\|\textstyle\sum_{i=0}^k (\tilde{c}_i - c_i) x_i\right\|_2 + \left\|\textstyle\sum_{i=0}^k \tilde{c}_i (\tilde{x}_i - x_i)\right\|_2 + \left\|\textstyle\sum_{i=0}^k c_i x_i - x^*\right\|_2. \tag{13}$$

This expression shows us that the precision is comparable to the precision of the AMPE process in the linear case (third term) with some perturbation. Also, if $\|e_i\|_2$ is small then $\|x_i - \tilde{x}_i\|_2$ is small as well. But we need more information about $\|c\|_2$ and $\|\tilde{c} - c\|_2$ if we want to go further.

We now show the following proposition computing the perturbation $\Delta c = (\tilde{c}^* - c^*)$ of the optimal solution of (AMPE), c^*, induced by $E = \tilde{U} - U$. It will allow us to bound the first term on the right-hand side of (13) (see proof A.2 in the Appendix). For simplicity, we will use $P = \tilde{U}^T\tilde{U} - U^T U$.

Proposition 2.3 *Let c^* be the optimal solution to* (AMPE)

$$c^* = \operatorname*{argmin}_{\mathbf{1}^T c = 1} \|Uc\|_2$$

for some matrix $U \in \mathbb{R}^{n \times k}$. Suppose U becomes $\tilde{U} = U + E$ and write $c^* + \Delta c$ the perturbed solution to (AMPE)*. Let $M = \tilde{U}^T\tilde{U}$ and the perturbation matrix $P = \tilde{U}^T\tilde{U} - U^T U$. Then,*

$$\Delta c = -\left(I - \frac{M^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^T M^{-1}\mathbf{1}}\right) M^{-1} P c^*. \tag{14}$$

We see here that the perturbation can be potentially large. Even if $\|c^*\|_2$ and $\|P\|_2$ can be potentially small, $\|M^{-1}\|_2$ is huge in general. It can be shown that $U^T U$ (the square of a Krylov-like matrix) presents an exponential condition number (see [24]) because the minimal eigenvalue decays very fast. Moreover, the eigenvalues are perturbed by P, leading to a potential huge perturbation Δc, especially if $\|P\|_2$ is comparable (or bigger) to $\lambda_{\min}(U^T U)$.

2.4 Regularized AMPE

The condition number of the matrix $U^T U$ in problem (AMPE) can be arbitrary large. Indeed, this condition number is related to the one of Krylov matrices which has been proved in [24] to be exponential in k. By consequence, this conditioning problem coupled with nonlinear errors lead to highly unstable solutions c^* (which we observe in our experiments). We thus study a regularized formulation of problem (AMPE), which reads

$$\begin{array}{ll} \text{minimize} & c^T(U^T U + \lambda I)c \\ \text{subject to} & \mathbf{1}^T c = 1 \end{array} \tag{RMPE}$$

The solution of this problem may be computed with a linear system, and the regularization parameter controls the norm of the solution, as shown in the following Lemma (see proof A.3 in Appendix).

Lemma 2.4 *Let c_λ^* be the optimal solution of problem (RMPE). Then*

$$c_\lambda^* = \frac{(U^TU + \lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(U^TU + \lambda I)^{-1}\mathbf{1}} \qquad \textit{and} \qquad \|c_\lambda^*\|_2 \leq \sqrt{\frac{\lambda + \|U\|_2^2}{k\lambda}}. \tag{15}$$

This allows us to obtain the following corollary extending Proposition 2.3 to the regularized AMPE problem in (RMPE), showing that the perturbation of c is now controlled by the regulaization parameter λ.

Corollary 2.5 *Let c_λ^*, defined in (15), be the solution of problem* (RMPE)*. Then the solution of problem* (RMPE) *for the perturbed matrix $\tilde{U} = U + E$ is given by $c_\lambda^* + \Delta c_\lambda$ where*

$$\Delta c_\lambda = -WM_\lambda^{-1}Pc_\lambda^* = -M_\lambda^{-1}W^TPc_\lambda^* \qquad \textit{and} \qquad \|\Delta c_\lambda^*\|_2 \leq \tfrac{\|P\|_2}{\lambda}\|c_\lambda^*\|_2,$$

where $M_\lambda = (U^TU + P + \lambda I)$ and $W = \left(I - \frac{M_\lambda^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^TM_\lambda^{-1}\mathbf{1}}\right)$ is a projector of rank $k-1$.

These results lead us to the following simple algorithm.

Algorithm 1 Regularized Approximate Minimal Polynomial Extrapolation (**RMPE**)

Input: Sequence $\{x_0, x_1, ..., x_{k+1}\}$, parameter $\lambda > 0$
Compute $U = [x_1 - x_0, ..., x_{k+1} - x_k]$
Solve the linear system $(U^TU + \lambda I)z = \mathbf{1}$
Set $c = z/(z^T\mathbf{1})$
Output: $\sum_{i=0}^k c_i x_i$, the approximation of the fixed point x^*

The computational complexity (with online updates or in batch mode) of the algorithm is $O(nk^2)$ and some strategies (batch and online) are discussed in the Appendix A.3. Note that the algorithm never calls the oracle $g(x)$. It means that, in an optimization context, the acceleration does not require $f(x)$ or $f'(x)$ to compute the extrapolation. Moreover, it does not need a priori information on the function, for example L and μ (unlike Nesterov's method).

2.5 Convergence Bounds on Regularized AMPE

To fully characterize the convergence of our estimate sequence, we still need to bound the last term on the right-hand side of (13), namely $\|\sum_{i=0}^k c_i x_i - x^*\|_2$. A coarse bound can be provided using Chebyshev polynomials, however the norm of the Chebyshev's coefficients grows exponentially as k grows. Here we refine this bound to better control the quality of our estimate.

Let $g(x^*) \preceq \sigma I$. Consider the following Chebyshev-like optimization problem, written

$$S(k,\alpha) \triangleq \min_{\{q \in \mathbb{R}_k[x]:\, q(1)=1\}} \left\{ \max_{x\in[0,\sigma]} \; ((1-x)q(x))^2 + \alpha\|q\|_2^2 \right\}, \tag{16}$$

where $\mathbb{R}_k[x]$ is the ring of polynomials of degree at most k and $q \in \mathbb{R}^{k+1}$ is the vector of coefficients of the polynomial $q(x)$. This problem can be solved exactly using a semi-definite solver because it can be reduced to a SDP program (see Appendix A.4 for the details of the reduction). Our main result below shows how $S(k,\alpha)$ bounds the error between our estimate of the optimum constructed using the iterates $\tilde{x}_i$ in (RMPE) and the optimum x^* of problem (1).

Proposition 2.6 *Let matrices $X = [x_0, x_1, ..., x_k]$, $\tilde{X} = [x_0, \tilde{x}_1, ..., \tilde{x}_k]$, $\mathcal{E} = (X - \tilde{X})$ and scalar $\kappa = \|(A-I)^{-1}\|_2$. Suppose $\tilde{c}_\lambda^*$ solves problem* (RMPE)

$$\begin{array}{ll} \textit{minimize} & c^T(\tilde{U}^T\tilde{U} + \lambda I)c \\ \textit{subject to} & \mathbf{1}^Tc = 1 \end{array} \quad \Rightarrow \quad \tilde{c}_\lambda^* = \frac{(\tilde{U}^T\tilde{U} + \lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(\tilde{U}^T\tilde{U} + \lambda I)^{-1}\mathbf{1}} \tag{17}$$

in the variable $c \in \mathbb{R}^{k+1}$, with parameters $\tilde{U} \in \mathbb{R}^{n\times(k+1)}$. Assume A symmetric with $0 \preceq A \prec I$. Then

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \left(\kappa^2 + \frac{1}{\lambda}\left(1 + \frac{\|P\|_2}{\lambda}\right)^2\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2\right)^{\frac{1}{2}}\left(S(k,\lambda/\|x_0 - x^*\|_2^2)\right)^{\frac{1}{2}}\|x_0 - x^*\|_2,$$

with P is defined in Corollary 2.5 and $S(k,\alpha)$ is defined in (16)*.*

We have that $S(k, \lambda/\|x_0 - x^*\|_2^2)^{\frac{1}{2}}$ is similar to the value $\mathcal{T}_k(\sigma)$ (see (8)) so our algorithm achieves a rate similar to the Chebyshev's acceleration up to some multiplicative scalar. We thus need to choose λ so that this multiplicative scalar is not too high (while keeping $S(k, \lambda/\|x_0 - x^*\|_2^2)^{\frac{1}{2}}$ small).

We can analyze the behavior of the bound if we start close to the optimum. Assume

$$\|\mathcal{E}\|_2 = O(\|x_0 - x^*\|_2^2), \qquad \|U\|_2 = O(\|x_0 - x^*\|_2) \quad \Rightarrow \quad \|P\|_2 = O(\|x_0 - x^*\|_2^3).$$

This case is encountered when minimizing a smooth strongly convex function with Lipchitz-continuous Hessian using fixed-step gradient method (this case is discussed in details in the Appendix, section A.6). Also, let $\lambda = \beta\|P\|_2$ for $\beta > 0$ and $\|x_0 - x^*\|$ small. We can thus approximate the right parenthesis of the bound by

$$\lim_{\|x-x^*\|_2 \to 0} \left(\|\mathcal{E}\|_2 + \kappa \frac{\|P\|_2}{2\sqrt{\lambda}} \right) = \lim_{\|x-x^*\|_2 \to 0} \left(\|\mathcal{E}\|_2 + \kappa \frac{\sqrt{\|P\|_2}}{2\sqrt{\beta}} \right) = \frac{\kappa\sqrt{\|P\|_2}}{2\sqrt{\beta}}.$$

Therefore, the bound on the precision of the extrapolation is approximately equal to

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \quad \lesssim \quad \kappa \left(1 + \frac{(1+\frac{1}{\beta})^2}{4\beta^2} \right)^{1/2} \sqrt{S\left(k, \frac{\beta\|P\|_2}{\|x_0 - x^*\|_2^2}\right)} \|x_0 - x^*\|_2$$

Also, if we use equation (8), it is easy to see that

$$\sqrt{S\left(k, 0\right)} \leq \min_{\{q \in \mathbb{R}_k[x]:\, q(1)=1\}} \max_{x \in [0, \sigma_1]} |q(x)| = \mathcal{T}_k(t(\sigma)) = \frac{2\zeta^k}{1+\zeta^{2k}},$$

where ζ is defined in (9). So, when $\|x_0 - x^*\|_2$ is close to zero, the regularized version of AMPE tends to converge as fast as AMPE (see equation (10)) up to a small constant.

3 Numerical Experiments

We test our methods on a regularized logistic regression problem written

$$f(w) = \textstyle\sum_{i=1}^m \log\left(1 + \exp(-y_i \xi_i^T w)\right) + \frac{\tau}{2}\|w\|_2^2,$$

where $\Xi = [\xi_1, ..., \xi_m]^T \in \mathbb{R}^{m \times n}$ is the design matrix and y is a $\{-1, 1\}^n$ vector of labels. We used the *Madelon* UCI dataset, setting $\tau = 10^2$ (in order to have a ratio L/τ equal to 10^9). We solve this problem using several algorithms, the fixed-step gradient method for strongly convex functions [6, Th. 2.1.15] using stepsize $2/(L+\mu)$, where $L = \|\Xi\|_2^2/4 + \tau$ and $\mu = \tau$, the accelerated gradient method for strongly convex functions [6, Th. 2.2.3] and our nonlinear acceleration of the gradient method iterates using RMPE in Proposition 2.6 with restarts.

This last algorithm is implemented as follows. We do k steps (in the numerical experiments, k is typically equal to 5) of the gradient method, then extrapolate a solution $\tilde{X}\tilde{c}_\lambda^*$ where $\tilde{c}_\lambda^*$ is computed by solving the RMPE problem (17) on the gradient iterates $\tilde{X}$, with regularization parameter λ determined by a grid search. Then, this extrapolation becomes the new starting point of the gradient method. We consider it as one iteration of RMPEk using k gradient oracle calls. We also analyze the impact of an inexact line-search (described in Appendix A.7) performed after this procedure.

The results are reported in Figure 1. Using very few iterates, the solution computed using our estimate (a nonlinear average of the gradient iterates) are markedly better than those produced by the Nesterov-accelerated method. This is only partially reflected by the theoretical bound from Proposition 2.6 which shows significant speedup in some regions but remains highly conservative (see Figure 3 in section A.6). Also, Figure 2 shows us the impact of regularization. The AMPE process becomes unstable because of the condition number of matrix M, which impacts the precision of the estimate.

4 Conclusion and Perspectives

In this paper, we developed a method which is able to accelerate, under some regularity conditions, the convergence of a sequence $\{x_i\}$ without any knowledge of the algorithm which generates this sequence. The regularization parameter present in the acceleration method can be computed easily using some inexact line-search.

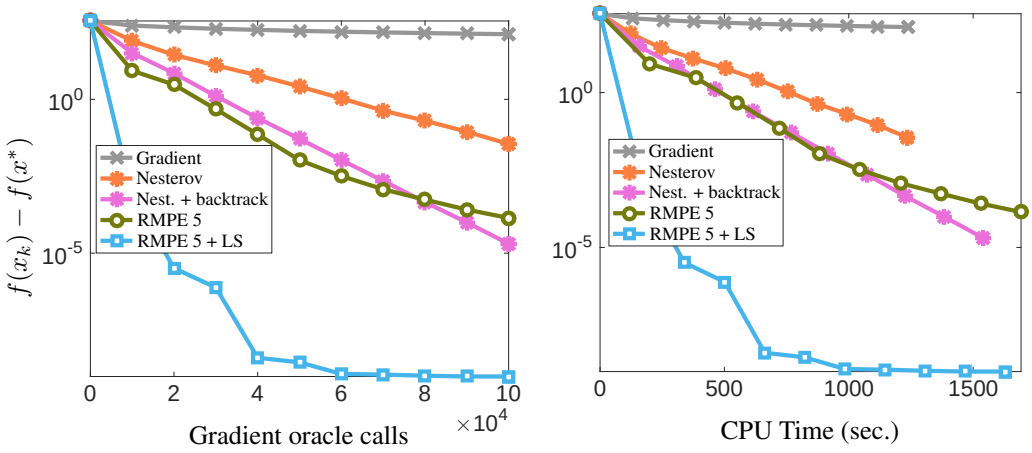

Figure 1: Solving logistic regression on *UCI Madelon dataset* (500 features, 2000 data points) using the gradient method, Nesterov's accelerated method and RMPE with $k = 5$ (with and without line search over the stepsize), with penalty parameter τ equal to 10^2 (Condition number is equal to $1.2 \cdot 10^9$). Here, we see that our algorithm has a similar behavior to the conjugate gradient: unlike the Nesterov's method, where we need to provide parameters μ and L, the RMPE algorithm adapts himself in function of the spectrum of $g(x^*)$ (so it can exploit the good local strong convexity parameter), without any prior specification. We can, for example, observe this behavior when the global strong convexity parameter is bad but not the local one.

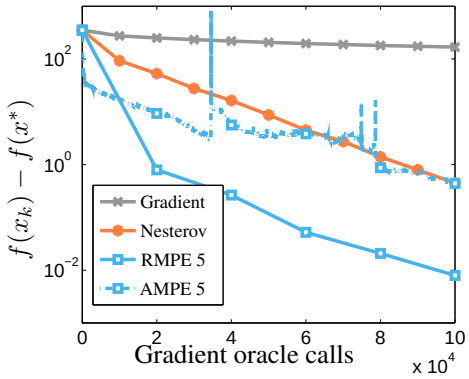

Figure 2: Logistic regression on *Madelon UCI Dataset*, solved using Gradient method, Nesterov's method and AMPE (i.e. RMPE with $\lambda = 0$). The condition number is equal to $1.2 \cdot 10^9$. We see that without regularization, AMPE is unstable because $\|(\tilde{U}^T\tilde{U})^{-1}\|_2$ is huge (see Proposition 2.3).

The algorithm itself is simple. By solving only a small linear system we are able to compute a good estimate of the limits of the sequence $\{x_i\}$. Also, we showed (using the gradient method on logistic regression) that the strategy which consists in alternating the algorithm and the extrapolation method can lead to impressive results, improving significantly the rate of convergence.

Future work will consist in improving the performance of the algorithm by exploiting the structure of the noise matrix E in some cases (for example, using the gradient method, the norm of the column E_k in the matrix E is decreasing when k grows), extending the algorithm to the constrained case, the stochastic case and to the non-symmetric case. We will also try to refine the term (16) present in the theoretical bound.

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, as well as support from ERC SIPA and the chaire *Économie des nouvelles données* with the *data science* joint research initiative with the *fonds AXA pour la recherche*.

References

[1] Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[2] AS Nemirovskii and Y. E Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, 25(2):21–30, 1985.

[3] Mešina, M. [1977], 'Convergence acceleration for the iterative solution of the equations $x = ax + f$', *Computer Methods in Applied Mechanics and Engineering* **10**(2), 165–173.

[4] Eddy, R. [1979], 'Extrapolating to the limit of a vector sequence', *Information linkage between applied mathematics and industry* pp. 387–396.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

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

[7] Y. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1-2):381–404, 2015.

[8] Yoel Drori and Marc Teboulle. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

[9] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[10] Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

[11] Andre Wibisono and Ashia C Wilson. On accelerated methods in optimization. Technical report, 2015.

[12] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3366–3374, 2015.

[13] Alexander Craig Aitken. On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1927.

[14] Daniel Shanks. Non-linear transformations of divergent and slowly convergent sequences. *Journal of Mathematics and Physics*, 34(1):1–42, 1955.

[15] Peter Wynn. On a device for computing the $e_m(s_n)$ transformation. *Mathematical Tables and Other Aids to Computation*, 10(54):91–96, 1956.

[16] C Brezinski. Accélération de la convergence en analyse numérique. *Lecture notes in mathematics*, (584), 1977.

[17] Avram Sidi, William F Ford, and David A Smith. Acceleration of convergence of vector sequences. *SIAM Journal on Numerical Analysis*, 23(1):178–196, 1986.

[18] N Levinson. The Wiener RMS error criterion in filter design and prediction, appendix b of wiener, n.(1949). *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, 1949.

[19] James Durbin. The fitting of time-series models. *Revue de l'Institut International de Statistique*, pages 233–244, 1960.

[20] Georg Heinig and Karla Rost. Fast algorithms for Toeplitz and Hankel matrices. *Linear Algebra and its Applications*, 435(1):1–59, 2011.

[21] David A Smith, William F Ford, and Avram Sidi. Extrapolation methods for vector sequences. *SIAM review*, 29(2):199–233, 1987.

[22] Stan Cabay and LW Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM Journal on Numerical Analysis*, 13(5):734–752, 1976.

[23] Gene H Golub and Richard S Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):157–168, 1961.

[24] Evgenij E Tyrtyshnikov. How bad are Hankel matrices? *Numerische Mathematik*, 67(2):261–269, 1994.

[25] Y. Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000.

[26] J. B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[27] P. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

[28] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization : analysis, algorithms, and engineering applications*. MPS-SIAM series on optimization. SIAM, 2001.

A Supplementary material

This section contains proofs and additional details on some of the material discussed above.

A.1 Proofs of minimal polynomial normalization

Lemma A.1 *Let $p(x)$ be the minimal polynomial of the matrix M. Assume that M does not have the eigenvalue 1. Then we can always normalize p by the sum of its coefficients, $p(1)$.*

Proof. Let p be the minimal polynomial of M. Then for all eigenvalue λ_i of M, we have $p(\lambda_i) = 0$. Since p is minimal, we cannot find q with $\deg q < \deg p$ satisfying the above condition.

Now, assume by contradiction that $p(1) = 0$. It means that 1 is a root of p, so p can be written

$$p(x) = (1 - x)q(x).$$

Because 1 is not an eigenvalue of M, we have $p(\lambda_i) = 0$ if and only if $q(\lambda_i) = 0$. However $\deg q < \deg p$, so p is not minimal. ■

A.2 Proofs of regularization results

Proposition A.2 *Let c^* be the optimal solution to* (AMPE)

$$c^* = \operatorname*{argmin}_{\mathbf{1}^T c = 1} \|Uc\|_2$$

for some matrix $U \in \mathbb{R}^{n,k}$. Suppose U becomes $\tilde{U} = U + E$ and write $c^* + \Delta c$ the perturbed solution to (AMPE)*, then*

$$\Delta c = -\left(I - \frac{M^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^T M^{-1}\mathbf{1}}\right) M^{-1}(U^T E + E^T U + E^T E)c^* \tag{18}$$

where $M = (U + E)^T(U + E)$ and

$$\left(I - \frac{M^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^T M^{-1}\mathbf{1}}\right)$$

is a projector of rank $k - 1$.

Proof. Let μ be the dual variable corresponding to the equality constraint. Both $c^* + \Delta c$ and $\mu^* + \Delta\mu$ must satisfy the KKT system

$$\begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta\mu \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

writing $P = U^T E + E^T U + E^T E$, this means again

$$\begin{aligned} \begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta\mu \end{pmatrix} &= \begin{bmatrix} 2P & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta\mu \end{pmatrix} + \begin{bmatrix} 2U^T U & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} c^* + \Delta c \\ \mu^* + \Delta\mu \end{pmatrix} \\ &= \begin{pmatrix} 2P(c^* + \Delta c) \\ 0 \end{pmatrix} + \begin{bmatrix} 2U^T U & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} \Delta c \\ \Delta\mu \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

hence

$$\begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{pmatrix} \Delta c \\ \Delta\mu \end{pmatrix} = \begin{pmatrix} -2Pc^* \\ 0 \end{pmatrix}$$

The block matrix can be inverted explicitly, with

$$\begin{bmatrix} 2M & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}^{-1} = \frac{1}{\mathbf{1}^T M^{-1}\mathbf{1}} \begin{bmatrix} \frac{1}{2}M^{-1}\left((\mathbf{1}^T M^{-1}\mathbf{1})I - \mathbf{1}\mathbf{1}^T M^{-1}\right) & M^{-1}\mathbf{1} \\ \mathbf{1}^T M^{-1} & -2 \end{bmatrix}$$

leading to an expression of Δc and $\Delta\mu$ in terms of c^* and μ^*:

$$\begin{pmatrix}\Delta c\\ \Delta\mu\end{pmatrix} = \frac{1}{\mathbf{1}^T M^{-1}\mathbf{1}} \begin{bmatrix} \frac{1}{2}M^{-1}\left((\mathbf{1}^T M^{-1}\mathbf{1})I - \mathbf{1}\mathbf{1}^T M^{-1}\right) & M^{-1}\mathbf{1} \\ \mathbf{1}^T M^{-1} & -2 \end{bmatrix} \begin{pmatrix}-2Pc^*\\ 0\end{pmatrix}$$

After some simplification, we get

$$\Delta c = -\left(I - \frac{M^{-1}\mathbf{1}\mathbf{1}^T}{\mathbf{1}^T M^{-1}\mathbf{1}}\right) M^{-1}Pc^* = -WM^{-1}Pc^*$$

where W is a projector of rank $k-1$, which is the desired result. ■

Lemma A.3 *Let c_λ^* be the optimal solution of problem (RMPE). Then*

$$c_\lambda^* = \frac{(U^TU + \lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(U^TU + \lambda I)^{-1}\mathbf{1}} \tag{19}$$

Therefore,

$$\|c_\lambda^*\|_2 \leq \sqrt{\frac{\lambda + \|U\|_2^2}{k\lambda}} \tag{20}$$

Proof. Let c_λ^* the optimal solution of the primal and ν_λ^* the optimal dual variable of problem (RMPE). Let $M_\lambda = U^TU + \lambda I$. Then both c_λ^* and ν_λ^* must satisfy the KKT system:

$$\begin{bmatrix}2M_\lambda & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}\begin{pmatrix}c_\lambda^*\\ \mu_\lambda^*\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}.$$

We have thus

$$\begin{pmatrix}c_\lambda^*\\ \mu_\lambda^*\end{pmatrix} = \begin{bmatrix}2M_\lambda & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}^{-1}\begin{pmatrix}0\\1\end{pmatrix}.$$

The block matrix can be inverted explicitly:

$$\begin{bmatrix}2M_\lambda & \mathbf{1}\\ \mathbf{1}^T & 0\end{bmatrix}^{-1} = \frac{1}{\mathbf{1}^T M_\lambda^{-1}\mathbf{1}}\begin{bmatrix}\frac{1}{2}M_\lambda^{-1}((\mathbf{1}^T M_\lambda^{-1}\mathbf{1})I - \mathbf{1}\mathbf{1}^T M_\lambda^{-1}) & M_\lambda^{-1}\mathbf{1}\\ \mathbf{1}^T M_\lambda^{-1} & -2\end{bmatrix},$$

leading to

$$c_\lambda^* = \frac{M_\lambda^{-1}\mathbf{1}}{\mathbf{1}^T M_\lambda^{-1}\mathbf{1}}.$$

Since

$$\|M_\lambda^{-1}\|_2 \leq \frac{1}{\sigma_{\min}(U^TU) + \lambda} \leq \frac{1}{\lambda}$$

and

$$\mathbf{1}^T M_\lambda^{-1}\mathbf{1} \geq \frac{\|\mathbf{1}\|^2}{\sigma_{\max}(M_\lambda)} \geq \frac{k}{\|U\|_2^2 + \lambda}$$

We obtain

$$\|c_\lambda^*\|_2 = \frac{\|M_\lambda^{-1/2}M_\lambda^{-1/2}\mathbf{1}\|_2}{\mathbf{1}^T M_\lambda^{-1}\mathbf{1}} \leq \frac{\|M_\lambda^{-1}\|_2^{1/2}}{\sqrt{\mathbf{1}^T M_\lambda^{-1}\mathbf{1}}} \leq \sqrt{\frac{\lambda + \|U\|_2^2}{k\lambda}}$$

which is the desired result. ■

A.3 Computational Complexity for RMPE Algorithm

In Algorithm 1, computing the coefficients $\tilde{c}_\lambda^*$ means solving the $k \times k$ system $(\tilde{U}^T\tilde{U} + \lambda I)z = \mathbf{1}$. We then get $\tilde{c}_\lambda^* = z/(\mathbf{1}^T z)$. This can be done in both batch and online mode.

Online updates. Here, we receive the vectors u_i one by one from the optimization algorithm. In this case, we perform low-rank updates on the Cholesky factorization of the system matrix. At iteration i, we have the Cholesky factorization $LL^T = \tilde{U}^T\tilde{U} + \lambda I$. We receive a new vector u_+ and we want

$$L_+L_+^T = \begin{bmatrix} L & 0 \\ a^T & b \end{bmatrix}\begin{bmatrix} L^T & a \\ 0 & b \end{bmatrix} = \begin{bmatrix} \tilde{U}^T\tilde{U} + \lambda I & \tilde{U}^Tu_+ \\ (\tilde{U}^Tu_+)^T & u_+^Tu_+ + \lambda \end{bmatrix} \Rightarrow\ a = L^{-1}\tilde{U}^Tu_+,\ b = a^Ta + \lambda.$$

The complexity of this update is thus $O(in + i^2)$, i.e. the matrix-vector multiplication of $\tilde{U}^Tu_+$ and solving the triangular system. Since we need to do it k times, the final complexity is thus $O(nk^2+k^3)$. Notice that, at the end, it takes only $O(k^2)$ iteration to solve the system $LL^Tz = \mathbf{1}$.

Batch mode. The complexity is divided in two parts: First, we need to build the linear system itself. Since $U \in \mathbb{R}^{n\times k}$, it takes $O(nk^2)$ flops to perform the multiplication. Then we need to solve the linear system $(\tilde{U}^T\tilde{U} + \lambda I)z = \mathbf{1}$ which can be done by a direct solver like Gaussian elimination (if k is small) or Cholesky factorization, or using an iterative method like conjugate gradient method. It takes $O(k^3)$ flops to solve the linear system in the worst case, meaning that the complexity at the end is $O(nk^2 + k^3)$. In practice, the eigenvalues of the system tend to be clustered around λ, which means that the conjugate gradient solver converges very quickly to a good solution.

A.4 Regularized Chebychev Polynomials

We first briefly recall basic results on Sum of Squares (SOS) polynomials and moment problems [25, 26, 27], which will allow us to formulate problem (16) as a (tractable) semidefinite program. A univariate polynomial is positive if and only if it is a sum of squares. Furthermore, if we let $m(x) = (1, x, \ldots, x^k)^T$ we have, for any $p(x) \in \mathbb{R}_{2k}[x]$,

$$\begin{gathered} p(x) \geq 0,\ \text{for all } x \in \mathbb{R} \\ \Updownarrow \\ p(x) = m(x)^TCm(x),\ \text{for some } C \succeq 0, \end{gathered}$$

which means that checking if a polynomial is positive on the real line is equivalent to solving a linear matrix inequality (see e.g. [28, §4.2] for details). We can thus write the problem of computing the maximum of a polynomial over the real line as

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & t - p(x) = m(x)^TCm(x), \quad \text{for all } x \in \mathbb{R} \\ & C \succeq 0, \end{array} \tag{21}$$

which is a semidefinite program in the variables $p \in \mathbb{R}^{k+1}$, $C \in \mathbf{S}_{k+1}$ and $t \in \mathbb{R}$, because the first contraint is equivalent to a set of linear equality constraints. Then, showing that $p(x) \geq 0$ on the segment $[0, \sigma]$ is equivalent to showing that the rational fraction $p\left(\sigma x^2/(1+x^2)\right)$ is positive on the real line, or equivalently, that the polynomial is positive on the real line. Overall, this implies that problem (16) can be written

$$\begin{array}{lll} S(k,\alpha) = & \text{min.} & t^2 + \alpha^2\|q\|_2^2 \\ & \text{s.t.} & (1+x^2)^{k+1}\left(\left(1 - \frac{\sigma x^2}{1+x^2}\right)\, q\left(\frac{\sigma x^2}{1+x^2}\right)\right) = m(x)^TCm(x), \quad \text{for all } x \in \mathbb{R} \\ & & \mathbf{1}^Tq = 1,\ C \succeq 0, \end{array} \tag{22}$$

which is a semidefinite program in the variables $q \in \mathbb{R}^{k+1}$, $C \in \mathbf{S}_{k+2}$ and $t \in \mathbb{R}$.

A.5 Proof of the convergence result

Proposition A.4 *Let matrices $X = [x_0, x_1, ..., x_k]$, $\tilde{X} = [x_0, \tilde{x}_1, ..., \tilde{x}_k]$, $\mathcal{E} = (X - \tilde{X})$ and scalar $\kappa = \|(A - I)^{-1}\|_2$. Suppose $\tilde{c}_\lambda^*$ solves problem* (RMPE)

$$\begin{array}{ll} \textit{minimize} & c^T(\tilde{U}^T\tilde{U} + \lambda I)c \\ \textit{subject to} & \mathbf{1}^Tc = 1 \end{array} \quad \Rightarrow \quad c_\lambda^* = \frac{(\tilde{U}^T\tilde{U} + \lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(\tilde{U}^T\tilde{U} + \lambda I)^{-1}\mathbf{1}} \tag{23}$$

in the variable $c \in \mathbb{R}^{k+1}$, with parameters $\tilde{U} \in \mathbb{R}^{n\times(k+1)}$. Assume A symmetric with $0 \preceq A \prec I$. Then

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \left(\kappa^2 + \frac{1}{\lambda}\left(1+\frac{\|P\|_2}{\lambda}\right)^2\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2\right)^{\frac{1}{2}}\left(S(k,\lambda/\|x_0-x^*\|_2^2)\right)^{\frac{1}{2}}\|x_0-x^*\|_2$$

with P is defined in Corollary 2.5 and $S(k,\alpha)$ is defined in (16)*.*

Proof. Let us write the error decomposition (13) in matrix format:

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \|Xc_\lambda^* - x^*\|_2 + \|(X-X^*)\Delta c\|_2 + \|\mathcal{E}\tilde{c}_\lambda^*\|_2.$$

The first term can be bounded as follow:

$$\begin{aligned}
\|Xc_\lambda^* - x^*\|_2 &\leq \kappa\|Uc_\lambda^*\|_2 \\
&\leq \kappa\sqrt{\|Uc_\lambda^*\|_2^2 + (\lambda-\lambda)\|c_\lambda^*\|_2^2} \\
&\leq \kappa\sqrt{\|(A-I)p(A)\|_2^2\|x_0-x^*\|_2^2 + \lambda\|c_\lambda^*\|_2^2 - \lambda\|c_\lambda^*\|_2^2} \\
&\leq \kappa\sqrt{S(k,\lambda/\|x_0-x^*\|_2^2)\|x_0-x^*\|_2^2 - \lambda\|c_\lambda^*\|_2^2}.
\end{aligned}$$

The second one becomes, if we use Corollary 2.5,

$$\begin{aligned}
\|(X-X^*)\Delta c_\lambda^*\|_2 &\leq \kappa\|U\Delta c_\lambda^*\|_2 \\
&\leq \kappa\|U(U^TU+\lambda I+P)^{-1}\tilde{W}^TP\|_2\|c_\lambda^*\|_2 \\
&\leq \kappa\|U(U^TU+\lambda I+P)^{-1}\|_2\|P\|_2\|c_\lambda^*\|_2.
\end{aligned}$$

Let us write $(U^TU+\lambda I+P)^{-1} = [(U^TU+\lambda I)^{-1}+S]$ for some perturbation S. Indeed,

$$((U^TU+\lambda I)^{-1}+S)(U^TU+\lambda I+P) = I,$$

which leads to

$$S = -(U^TU+\lambda I)^{-1}P(U^TU+\lambda I+P)^{-1}.$$

If we plug this expression in $\|U(U^TU+\lambda I+P)^{-1}\|_2$ we obtain

$$\begin{aligned}
\|U(U^TU+\lambda I+P)^{-1}\|_2 &= \|U(U^TU+\lambda I)^{-1}(I-P(U^TU+\lambda I+P)^{-1})\|_2 \\
&\leq \|U(U^TU+\lambda I)^{-1}\|_2\left(1+\|P\|_2\|(U^TU+\lambda I+P)^{-1}\|_2\right) \\
&\leq \frac{\sigma}{\sigma^2+\lambda}\left(1+\frac{\|P\|_2}{\lambda}\right).
\end{aligned}$$

For any value of $\sigma \in [\sigma_{\min}^{1/2}(U^TU), \sigma_{\max}^{1/2}(U^TU)]$. The maximum is attained at $\sigma = \sqrt{\lambda}$, so it becomes

$$\|U(U^TU+\lambda I+P)^{-1}\|_2 \leq \frac{1}{2\sqrt{\lambda}}\left(1+\frac{\|P\|_2}{\lambda}\right).$$

So the second term can be bounded by

$$\|(X-X^*)\Delta c_\lambda^*\|_2 \leq \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\left(1+\frac{\|P\|_2}{\lambda}\right)\|c_\lambda^*\|.$$

The third term can be bounded as follow:

$$\begin{aligned}
\|\mathcal{E}\tilde{c}_\lambda^*\|_2 &\leq \|\mathcal{E}\|_2(\|c_\lambda^*\|_2 + \|\Delta c_\lambda^*\|_2) \\
&\leq \|\mathcal{E}\|_2\left(1+\frac{\|P\|_2}{\lambda}\right)\|c_\lambda^*\|_2.
\end{aligned}$$

If we combine all bounds, we obtain

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2^2 \leq \kappa\sqrt{S(k,\lambda/\|x_0-x^*\|_2^2)\|x_0-x^*\|_2^2 - \lambda\|c_\lambda^*\|_2^2} + \|c_\lambda^*\|_2\left(1+\frac{\|P\|_2}{\lambda}\right)\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right).$$

We will now find the value of $\|c_\lambda^*\|_2$ which maximize the bound. For more simplicity, let us write the bound using parameters a, b and $c = \|c_\lambda^*\|_2$:

$$\kappa\sqrt{a^2 - \lambda c^2} + bc.$$

We want to solve

$$\max_{c:0\leq c\leq (a/\sqrt{\lambda})} \kappa\sqrt{a^2 - \lambda c^2} + bc,$$

and the solution is given by

$$c = \frac{a}{\sqrt{\lambda}}\frac{b}{\sqrt{\kappa^2\lambda + b^2}} \in \left[0, \frac{a}{\sqrt{\lambda}}\right].$$

The optimal value becomes

$$\max_{c:0\leq c\leq (a/\sqrt{\lambda})} \kappa\sqrt{a^2 - \lambda c^2} + bc = \frac{a}{\sqrt{\lambda}}\sqrt{\kappa^2\lambda + b^2}.$$

In other words, if we replace a, b and c, we have

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \sqrt{S(k, \lambda/\|x_0 - x^*\|_2^2)}\|x_0 - x^*\|_2\sqrt{\kappa^2 + \frac{1}{\lambda}\left(1 + \frac{\|P\|_2}{\lambda}\right)^2\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2},$$

which is the desired result. ■

A.6 Explicit bounds for the gradient method

Here, we make our bounds explicit in the case where we use the simple gradient method for smooth and strongly convex function with Lipchitz-continuous Hessian. In this scenario, the fixed point function becomes

$$\tilde{x}_{k+1} = \tilde{x}_k - \frac{1}{L}f'(\tilde{x}_k)$$

where $\mu I \preceq f''(x) \preceq LI$ (and we assume the bounds thigh at $x = x^*$ for simplicity). Moreover,

$$\|f''(y) - f''(x)\|_2 \leq M\|y - x\|_2$$

We have thus $A = I - \frac{1}{L}f''(x^*)$, meaning that $\|A\|_2 \leq 1 - \frac{\mu}{L}$. The rate of this method is

$$\|\tilde{x}_k - x^*\|_2 \leq \left(\sqrt{\frac{L-\mu}{L+\mu}}\right)^k \|x_0 - x^*\|_2 = r^k\|x_0 - x^*\|_2$$

Notice that this is not the optimal fixed-step gradient method, however it allows us a much simpler analysis. Now let us bound $\|\tilde{X} - X^*\|_2$, $\|U\|_2$ and $\|E\|_2$. Indeed,

$$\begin{aligned}
\|\tilde{X} - X^*\|_2 &\leq \sum_{i=0}^{k}\|\tilde{x}_i - x^*\|_2 \\
&= \frac{1 - r^k}{1 - r}\|x_0 - x^*\|_2 \\
\|U\|_2 &\leq \|A - I\|_2\sum_{i=0}^{k}\|x_i - x^*\|_2 \\
&\leq \sum_{i=0}^{k}\|A\|^i\|x_0 - x^*\|_2 \\
&\leq \frac{1 - \|A\|_2^k}{1 - \|A\|_2}\|x_0 - x^*\|_2 \\
&\leq \frac{L}{\mu}\left(1 - \left(1 - \frac{\mu}{L}\right)^k\right)\|x_0 - x^*\|_2
\end{aligned}$$

$$\begin{aligned}
\tilde{x}_{i+1} - x_{i+1} &= \tilde{x}_i - \frac{1}{L}f'(\tilde{x}_i) - x_i + \frac{1}{L}f''(x^*)(x_i - x^*) \\
&= \tilde{x}_i - x_i - \frac{1}{L}(f'(\tilde{x}_i) - f''(x^*)(x_i - x^*)) \\
&= (I - \frac{f''(x^*)}{L})(\tilde{x}_i - x_i) - \frac{1}{L}(f'(\tilde{x}_i) - f''(x^*)(\tilde{x}_i - x^*))
\end{aligned}$$

Since our function has a Lipchitz-continuous Hessian, it is possible to show that ([6], Lemma 1.2.4)

$$\|f'(\tilde{y}) - f'(x) - f''(x)(\tilde{y} - x)\|_2 \leq \frac{M}{2}\|y - x\|^2$$

We can thus bound the norm of the error at the i^{th} iteration:

$$\begin{aligned}
\|x_{i+1} - \tilde{x}_{i+1}\|_2 &\leq \left\|I - \frac{f''(x^*)}{L})\right\|_2 \|x_i - \tilde{x}_i\|_2 + \frac{1}{L}\|f'(\tilde{x}_i) - f''(x^*)(\tilde{x}_i - x^*)\|_2 \\
&= \left\|I - \frac{f''(x^*)}{L})\right\|_2 \|x_i - \tilde{x}_i\|_2 + \frac{1}{L}\|f'(\tilde{x}_i) - f'(x^*) - f''(x^*)(\tilde{x}_i - x^*)\|_2 \\
&\leq (1 - \frac{\mu}{L})\|x_i - \tilde{x}_i\|_2 + \frac{M}{2L}\|\tilde{x}_i - x^*\|_2^2 \\
&\leq (1 - \frac{\mu}{L})\|x_i - \tilde{x}_i\|_2 + \frac{M}{2L}(r^2)^i\|x_0 - x^*\|_2^2 \\
&= \frac{M}{2L}\sum_{j=1}^{i}(1 - \frac{\mu}{L})^{i-j}(r^2)^j\|x_0 - x^*\|_2^2
\end{aligned}$$

The sum starts at $j = 1$ because, by definition, $\|e_0\|_2 = 0$. In order to have a simpler analysis, let us use the fact that

$$\frac{r^2}{1 - \frac{\mu}{L}} = \frac{1}{1 + \frac{\mu}{L}} < 1$$

We can bound $\|x_{i+1} - \tilde{x}_{i+1}\|_2$ with a simpler expression:

$$\begin{aligned}
\|x_{i+1} - \tilde{x}_{i+1}\|_2 &\leq (1 - \frac{\mu}{L})^i \frac{M}{2L}\sum_{j=1}^{i}\left(\frac{r^2}{1 - \frac{\mu}{L}}\right)^j \|x_0 - x^*\|_2^2 \\
&\leq (1 - \frac{\mu}{L})^i \frac{M}{2L}\left(\sum_{j=0}^{i}\left(\frac{r^2}{1 - \frac{\mu}{L}}\right)^j\right)\|x_0 - x^*\|_2^2 \\
&= (1 - \frac{\mu}{L})^i \frac{M}{2L}\left(\sum_{j=0}^{i}\left(\frac{1}{1 + \frac{\mu}{L}}\right)^j\right)\|x_0 - x^*\|_2^2 \\
&= (1 - \frac{\mu}{L})^i \frac{M}{2L}\left(\frac{1 - \left(\frac{1}{1+\frac{\mu}{L}}\right)^i}{1 - \frac{1}{1+\frac{\mu}{L}}}\right)\|x_0 - x^*\|_2^2 \\
&= \frac{M}{2L}\left(\frac{(1 - \frac{\mu}{L})^i - \left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}}\right)^i}{1 - \frac{1}{1+\frac{\mu}{L}}}\right)\|x_0 - x^*\|_2^2 \\
&= \left(1 + \frac{L}{\mu}\right)\frac{M}{2L}\left((1 - \frac{\mu}{L})^i - \left(\frac{1 - \frac{\mu}{L}}{1 + \frac{\mu}{L}}\right)^i\right)\|x_0 - x^*\|_2^2
\end{aligned}$$

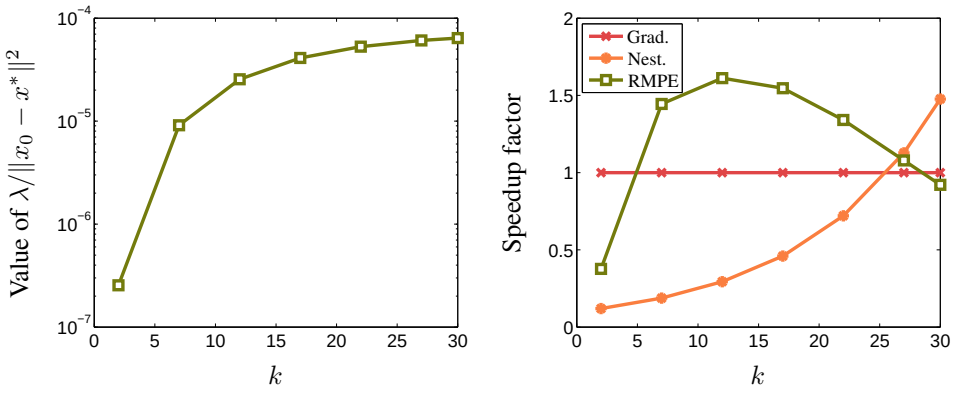

Figure 3: *Left:* Relative value for the regularization parameter λ used in the theoretical bound. *Right:* Convergence speedup relative to gradient, for Nesterov's accelerated method and the theoretical RMPE bound in Proposition 2.6. We see that our algorithm performs a significant speedup (even in comparison with Nesterov's method) when k is well chosen.

If we sum all the errors we get

$$\begin{aligned}
\|\mathcal{E}\|_2 &\leq \sum_{i=0}^{k} \|x_i - \tilde{x}_i\|_2 \\
&\leq \left(1+\frac{L}{\mu}\right)\frac{M}{2L}\left(\sum_{i=0}^{k}(1-\frac{\mu}{L})^i - \sum_{i=0}^{k}\left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}}\right)^i\right)\|x_0 - x^*\|_2^2 \\
&= \left(1+\frac{L}{\mu}\right)\frac{M}{2L}\left(\frac{L}{\mu}\left(1-\left(1-\frac{\mu}{L}\right)^k\right) - \frac{L}{2\mu}\left(1-\left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}}\right)^k\right)\right)\|x_0 - x^*\|_2^2 \\
&\leq \left(1+\frac{L}{\mu}\right)^2\frac{M}{2L}\left(1-\left(1-\frac{\mu}{L}\right)^k - \frac{1}{2}\left(1-\left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}}\right)^k\right)\right)\|x_0 - x^*\|_2^2 \\
&= \left(1+\frac{L}{\mu}\right)^2\frac{M}{2L}\left(\frac{1}{2} - \left(1-\frac{\mu}{L}\right)^k + \frac{1}{2}\left(\frac{1-\frac{\mu}{L}}{1+\frac{\mu}{L}}\right)^k\right)\|x_0 - x^*\|_2^2
\end{aligned}$$

We can also deduce that

$$\|\tilde{U} - U\|_2 = \|E\|_2 \leq 2\|\mathcal{E}\|_2 = \|\tilde{X} - X\|_2$$

Let us fix μ, L, M and $\|x_0 - x^*\|_2$ to some values:

- $L = 100$,
- $\mu = 10$,
- $M = 10^{-1}$,
- $\|x_0 - x^*\|_2 = 10^{-4}$.

We also decided to put $\lambda = \|P\|_2$.

In figure 3 (left) we can see the relative value for λ (i.e. $\|P\|_2/\|x_0 - x^*\|^2$) using the above parameters. In this case, we can compute the rate of convergence of the AMPE method using proposition 2.6. This rate of convergence is showed in figure 3 (right).

A.7 Additional numerical experiments

We test our methods on a regularized logistic regression problem written

$$f(w) = \sum_{i=1}^{m} \log\left(1 + \exp(-y_i \xi_i^T w)\right) + \frac{\tau}{2}\|w\|_2^2,$$

Optimization is done on dataset Madelon (500 features, 4400 points), sido0 (4932 features, 12678 points) and sonar (60 features, 208 points) with different values of τ. The starting point is always $w = 0$. The optimization is done on raw data. We compare different algorithms:

- fixed-step gradient method with stepsize = $2/(L + \mu)$,
- Nesterov's method for strongly convex functions with fixed coefficients,
- Nesterov's method using backtracking strategy on the smoothness parameter L,
- RMPE with $k = 5$ (i.e. RMPE5), where λ is found using grid search,
- RMPE5 using grid search for λ and line-search for the stepsize.

The line-search on the stepsize works as follow. We compute an extrapolation $\bar{x}$ =RMPE(X, k, λ) using iterates $X = [x_0, ..., x_k]$, then we find a coefficient $c \in \{1, 2, 4, ..., 2^i, ...\}$ which minimize the objective function $f(x_0 + c(\bar{x} - x_0))$. If we assume that the computation of the objective function value is much cheaper than the computation of its gradient, then this line-search has almost no impact on the total complexity of the algorithm.

In all experiments, we compare the rate of convergence of these methods in function of the number of gradient oracle call ot in function of time.

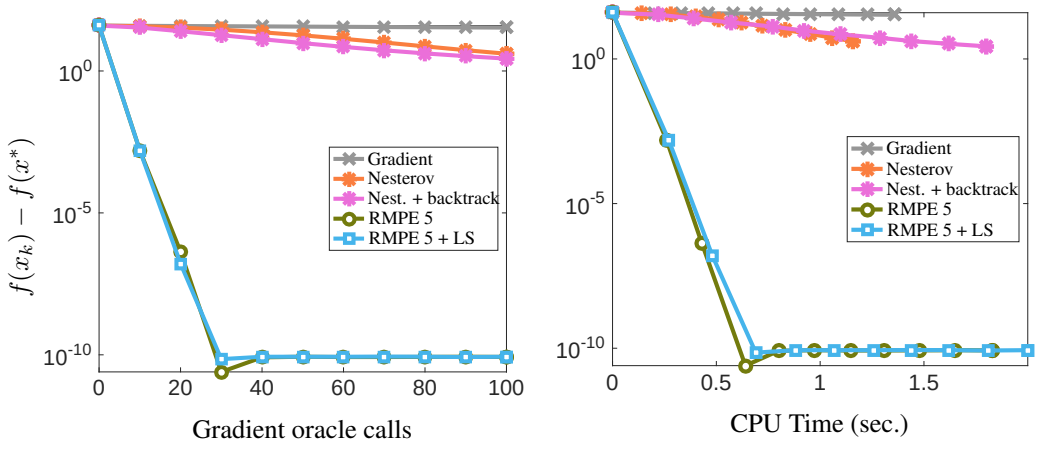

Figure 4: Logistic regression on Madelon dataset, with $\tau = 10^7$ (condition number = $6 \cdot 10^3$).

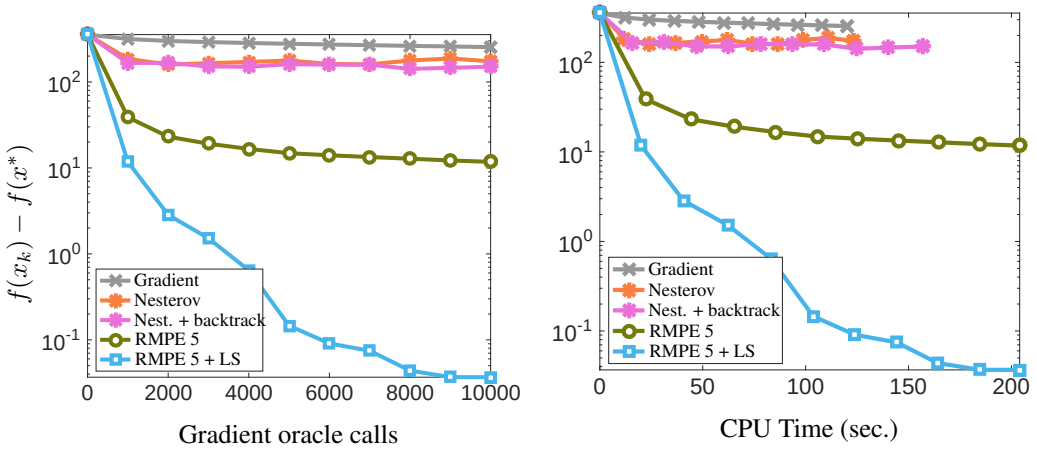

Figure 5: Logistic regression on Madelon dataset, with $\tau = 10^{-3}$ (condition number = $6 \cdot 10^{13}$).

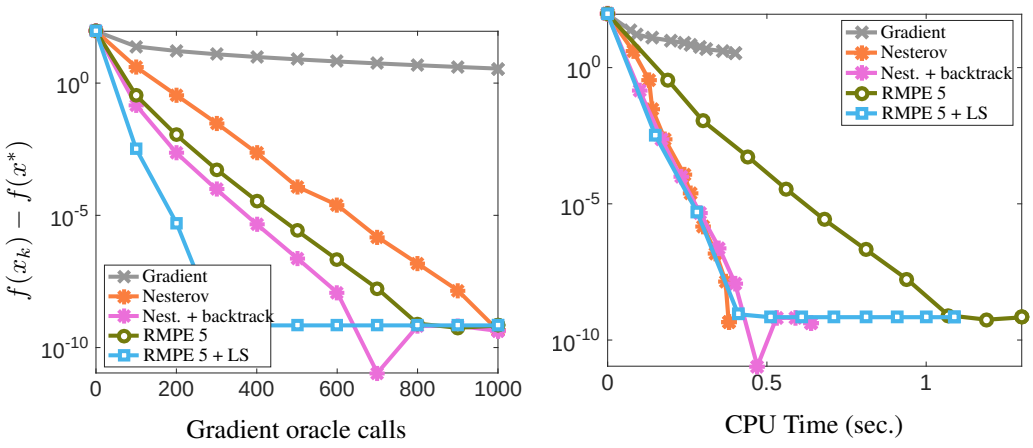

Figure 6: Logistic regression on sonar dataset, with $\tau = 10^{-1}$ (condition number = $7 \cdot 10^3$)

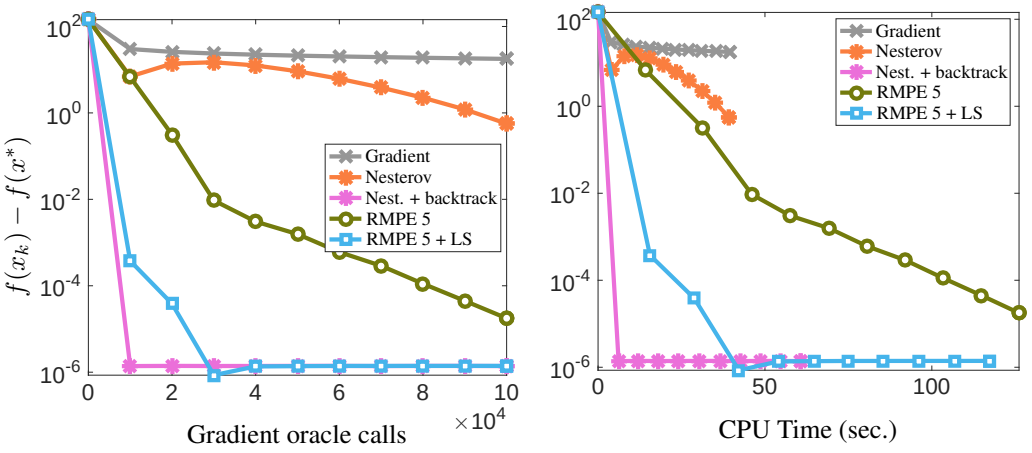

Figure 7: Logistic regression on sonar dataset, with $\tau = 10^{-6}$ (condition number = $7 \cdot 10^8$)

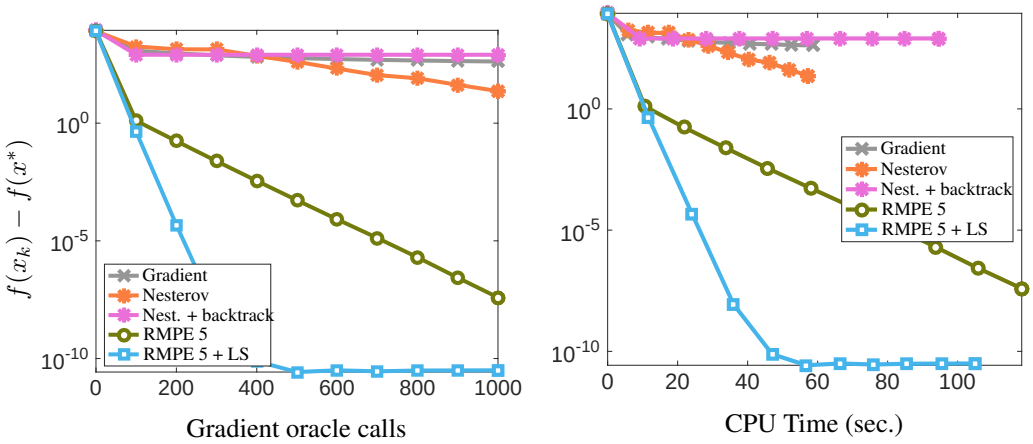

Figure 8: Logistic regression on sido0 dataset, with $\tau = 10^2$ (condition number = $1.5 \cdot 10^5$)