Regularized Nonlinear Acceleration.

Alexandre d'Aspremont,
CNRS & D.I. ENS.

with Damien Scieur & Francis Bach.

Support from ERC SIPA and ITN MacSeNet.

Jobs

Laplace Junior Faculty positions (two to three years) in **data science**.

At Ecole Normale Supérieure in Paris, competitive salary, travel fund.

See `https://data-ens.github.io/jobs/`

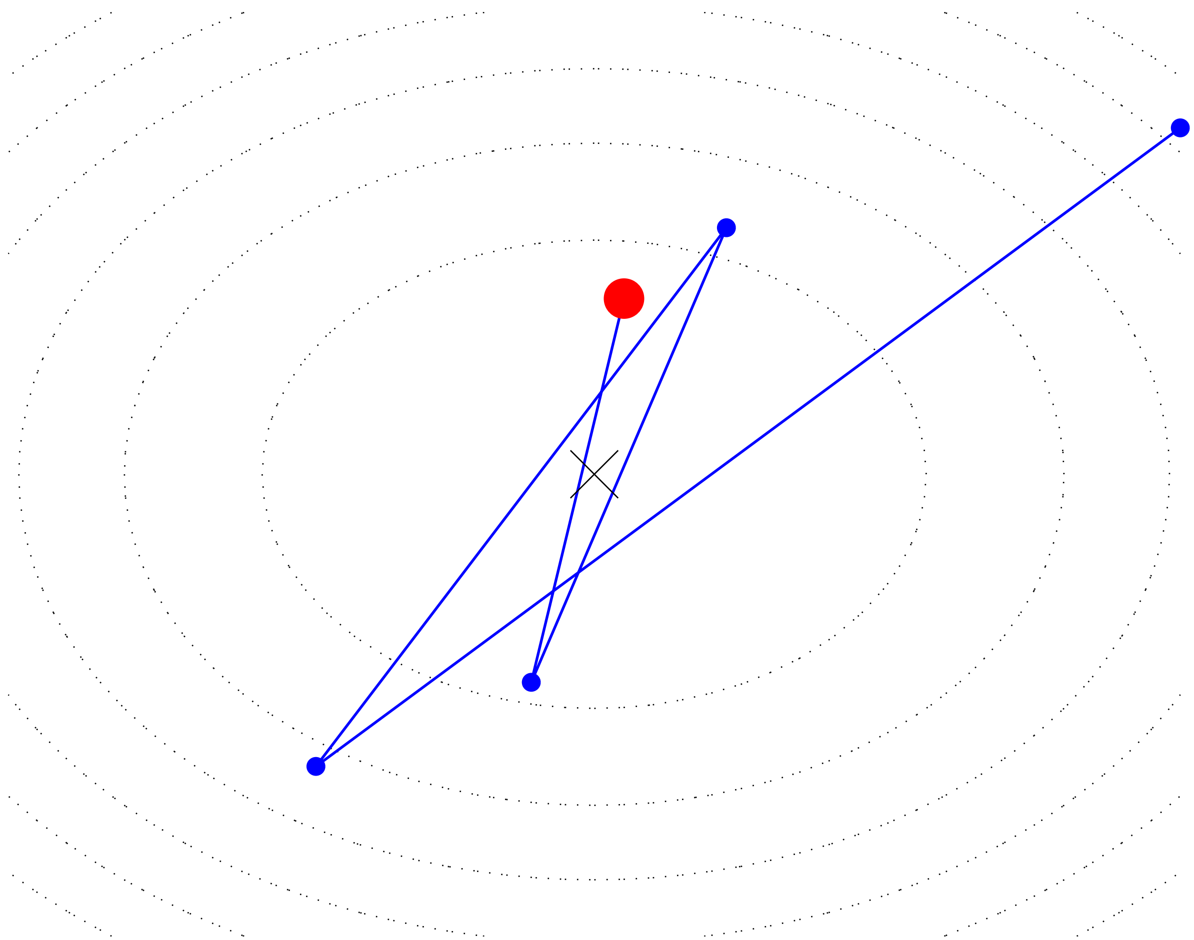
Introduction

Generic convex optimization problem

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

Introduction

Algorithms produce a **sequence** of iterates.



We only keep the last (or best) one. . .

Introduction

Aitken's Δ^2 [Aitken, 1927]. Given a sequence $\{s_k\}_{k=1,\ldots} \in \mathbb{R}^{\mathbb{N}}$ with limit s_*, and suppose

$$s_{k+1} - s_* = a\,(s_k - s_*), \quad \text{for } k = 1, \ldots$$

We can compute a using

$$s_{k+1} - s_k = a\,(s_k - s_{k-1}) \qquad \Rightarrow \qquad a = \frac{s_{k+1} - s_k}{s_k - s_{k-1}}$$

and get the limit s^* by solving

$$s_{k+1} - s^* = \frac{s_{k+1} - s_k}{s_k - s_{k-1}}(s_k - s^*)$$

which yields

$$s^* = \frac{s_{k-1}s_{k+1} - s_k^2}{s_{k+1} - 2s_k + s_{k-1}}$$

This is **Aitken's Δ^2** and allows us to **compute s_* from $\{s_{k+1}, s_k, s_{k-1}\}$.**

Introduction

Aitken's Δ^2 [Aitken, 1927], again. Given a sequence $\{s_k\}_{k=1,\ldots} \in \mathbb{R}^{\mathbb{N}}$ with limit s_*, and suppose that for $k = 1, \ldots,$

$$a_0\left(s_k - s_*\right) + a_1\left(s_{k+1} - s_*\right) = 0 \quad \text{and } a_0 + a_1 = 1 \quad \text{(normalization)}$$

We have

$$\begin{aligned} \underbrace{(a_0 + a_1)}_{=1} \; s_* &= a_0 s_{k-1} + a_1 s_k \\ 0 &= a_0(s_k - s_{k-1}) + a_1(s_{k+1} - s_k) \end{aligned}$$

We get s^* using

$$\begin{bmatrix} 0 & s_{k+1} - s_k & s_k - s_{k-1} \\ -1 & s_k & s_{k-1} \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} s^* \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \Leftrightarrow \quad s^* = \frac{\begin{vmatrix} s_{k+1} - s_k & s_k - s_{k-1} \\ s_k & s_{k-1} \end{vmatrix}}{\begin{vmatrix} s_{k+1} - s_k & s_k - s_{k-1} \\ 1 & 1 \end{vmatrix}}$$

Same formula as before, but generalizes to **higher dimensions.**

Introduction

Convergence acceleration. Consider

$$s_k = \sum_{i=0}^{k} \frac{(-1)^i}{(2i+1)} \quad \xrightarrow{k\to\infty} \quad \frac{\pi}{4} = 0.785398\ldots$$

we have

k	$\frac{(-1)^k}{(2k+1)}$	$\sum_{i=0}^{k} \frac{(-1)^i}{(2i+1)}$	Δ^2
0	1	1.0000	–
1	-0.33333	0.66667	–
2	0.2	0.86667	**0.7**9167
3	-0.14286	**0.7**2381	**0.78**333
4	0.11111	0.83492	**0.78**631
5	-0.090909	**0.7**4401	**0.78**492
6	0.076923	0.82093	**0.785**68
7	-0.066667	**0.7**5427	**0.785**22
8	0.058824	0.81309	**0.785**52
9	-0.052632	**0.7**6046	**0.7853**1

Introduction

Convergence acceleration.

- Similar results apply to sequences satisfying

$$\sum_{i=0}^{k} a_i(s_{n+i} - s_*) = 0$$

  using Aitken's ideas recursively.

- This produces **Wynn's $\varepsilon-$algorithm** [Wynn, 1956].
- See [Brezinski, 1977] for a survey on acceleration, extrapolation.
- Directly related to the Levinson-Durbin algo on AR processes.
- **Vector case:** focus on **Minimal Polynomial Extrapolation** [Sidi et al., 1986].

Overall: a simple **postprocessing** step.

Outline

- Introduction
- **Minimal Polynomial Extrapolation**
- Regularized MPE
- Numerical results

Minimal Polynomial Extrapolation

Quadratic example. Minimize

$$f(x) = \frac{1}{2}\|Bx - b\|_2^2$$

using the basic gradient algorithm, with

$$x_{k+1} := x_k - \frac{1}{L}(B^T B x_k - b).$$

we get

$$x_{k+1} - x^* := \underbrace{\left(\mathbf{I} - \frac{1}{L}B^T B\right)}_{A}(x_k - x^*)$$

since $B^T B x^* = b$.

This means $x_{k+1} - x^*$ follows a **vector autoregressive process.**

Minimal Polynomial Extrapolation

We have

$$\sum_{i=0}^{k} c_i(x_i - x^*) = \sum_{i=1}^{k} c_i A^i (x_0 - x^*)$$

and setting $\mathbf{1}^T c = 1$, yields

$$\left(\sum_{i=0}^{k} c_i x_i\right) - x^* = p(A)(x_0 - x^*), \quad \text{where } p(v) = \textstyle\sum_{i=1}^{k} c_i v^i$$

- Setting c such that $p(A)(x_0 - x^*) = 0$, we would have

$$\mathbf{x^* = \sum_{i=0}^{k} c_i x_i}$$

- Get the limit by **averaging iterates** (using weights depending on x_k).
- We typically do not observe A (or x^*).
- How do we extract c from the iterates x_k?

Minimal Polynomial Extrapolation

We have

$$\begin{aligned} x_k - x_{k-1} &= (x_k - x^*) - (x_{k-1} - x^*) \\ &= (A - \mathbf{I})A^{k-1}(x_0 - x^*) \end{aligned}$$

hence if $p(A) = 0$, we must have

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

so if $(A - \mathbf{I})$ is nonsingular, the coefficient vector c solves the **linear system**

$$\begin{cases} \sum_{i=1}^{k} c_i(x_i - x_{i-1}) = 0 \\ \sum_{i=1}^{k} c_i = 1 \end{cases}$$

and $p(\cdot)$ is the **minimal polynomial** of A w.r.t. $(x_0 - x^*)$.

Approximate Minimal Polynomial Extrapolation

Approximate MPE.

- For k smaller than the degree of the minimal polynomial, we find c that **minimizes the residual**

$$\|(A-\mathbf{I})p(A)(x_0-x^*)\|_2 = \left\|\sum_{i=1}^{k} c_i(x_i - x_{i-1})\right\|_2$$

- Setting $U \in \mathbb{R}^{n\times k+1}$, with $U_i = x_{i+1} - x_i$, this means solving

$$c^* \triangleq \mathop{\mathrm{argmin}}_{\mathbf{1}^T c=1} \|Uc\|_2 \tag{AMPE}$$

  in the variable $c \in \mathbb{R}^{k+1}$.

- Also known as Eddy-Mešina method [Mešina, 1977, Eddy, 1979] or Reduced Rank Extrapolation with arbitrary k (see [Smith et al., 1987, §10]).

Uniform Bound

Chebyshev polynomials. Crude bound on $\|Uc^*\|_2$ using Chebyshev polynomials, to bound error as a function of k, with

$$\begin{aligned}
\left\|\sum_{i=0}^k c_i^* x_i - x^*\right\|_2 &= \left\|(I-A)^{-1}\sum_{i=0}^k c_i^* U_i\right\|_2\\
&\leq \left\|(I-A)^{-1}\right\|_2 \|p(A)(x_1-x_0)\|_2
\end{aligned}$$

We have

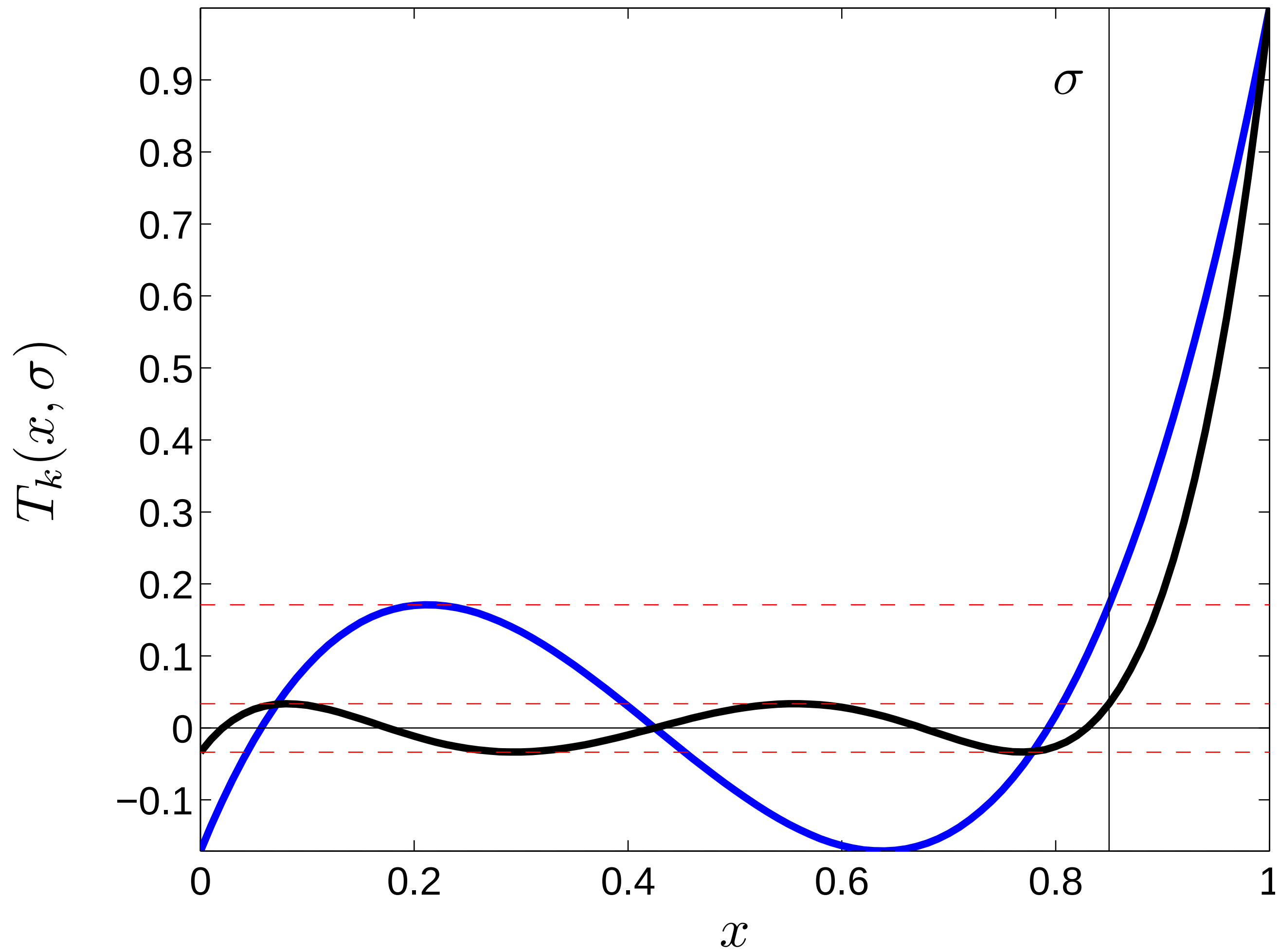
$$\begin{aligned}
\|p(A)(x_1-x_0)\|_2 &\leq \|p(A)\|_2\ \|(x_1-x_0)\|_2\\
&= \max_{i=1,\ldots,n} |p(\lambda_i)|\ \|(x_1-x_0)\|_2
\end{aligned}$$

where $0 \leq \lambda_i \leq \sigma$ are the eigenvalues of A. It suffices to find $p(\cdot) \in \mathbb{R}_k[x]$ solving

$$\inf_{\{p\in\mathbb{R}_k[x]:\, p(1)=1\}} \ \sup_{v\in[0,\sigma]} \ |p(v)|$$

Explicit solution using modified **Chebyshev polynomials.**

Uniform Bound using Chebyshev Polynomials



Chebyshev polynomials $T_3(x, \sigma)$ and $T_5(x, \sigma)$ for $x \in [0, 1]$ and $\sigma = 0.85$.
The maximum value of T_k on $[0, \sigma]$ decreases geometrically fast when k grows.

Approximate Minimal Polynomial Extrapolation

Proposition

AMPE convergence. *Let A be symmetric, $0 \preceq A \preceq \sigma I$ with $\sigma < 1$ and c^* be the solution of* (AMPE). *Then*

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

where $\kappa(A - I)$ is the condition number of the matrix $A - I$ and ζ is given by

$$\zeta = \frac{1 - \sqrt{1 - \sigma}}{1 + \sqrt{1 - \sigma}} < \sigma, \tag{2}$$

See also [Nemirovskiy and Polyak, 1984]. Gradient method, $\sigma = 1 - \mu/L$, so

$$\left\| \sum_{i=0}^{k} c_i^* x_i - x^* \right\|_2 \leq \kappa(A - I) \left(\frac{1 - \sqrt{\mu/L}}{1 + \sqrt{\mu/L}} \right)^k \|x_0 - x^*\|_2$$

Approximate Minimal Polynomial Extrapolation

AMPE versus Nesterov, conjugate gradient.

- Key difference with conjugate gradient: we do not observe A...
- Chebyshev polynomials satisfy a two-step recurrence. For quadratic minimization using the gradient method:

$$\begin{cases} z_{k-1} = y_{k-1} - \frac{1}{L}(By_{k-1} - b) \\ y_k = \dfrac{\alpha_{k-1}}{\alpha_k}\left(\dfrac{2z_{k-1}}{\sigma} - y_{k-1}\right) - \dfrac{\alpha_{k-2}}{\alpha_k} y_{k-2} \end{cases}$$

where $\alpha_k = \frac{2-\sigma}{\sigma}\alpha_{k-1} - \alpha_{k-2}$

- Nesterov's acceleration recursively computes a similar polynomial with

$$\begin{cases} z_{k-1} = y_{k-1} - \frac{1}{L}(By_{k-1} - b) \\ y_k = z_{k-1} + \beta_k(z_{k-1} - z_{k-2}), \end{cases}$$

see also [Hardt, 2013].

Approximate Minimal Polynomial Extrapolation

Accelerating optimization algorithms. For gradient descent, we have

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

- This means $\tilde{x}_{k+1} - x^* := A(\tilde{x}_k - x^*) + O(\|\tilde{x}_k - x^*\|_2^2)$ where

$$A = I - \frac{1}{L}\nabla^2 f(x^*),$$

  meaning that $\|A\|_2 \leq 1 - \frac{\mu}{L}$, whenever $\mu I \preceq \nabla^2 f(x) \preceq LI$.

- Approximation error is a sum of three terms

$$\left\|\sum_{i=0}^{k} \tilde{c}_i \tilde{x}_i - x^*\right\|_2 \leq \underbrace{\left\|\sum_{i=0}^{k} c_i x_i - x^*\right\|_2}_{\textbf{AMPE}} + \underbrace{\left\|\sum_{i=0}^{k} (\tilde{c}_i - c_i) x_i\right\|_2}_{\textbf{Stability}} + \underbrace{\left\|\sum_{i=0}^{k} \tilde{c}_i (\tilde{x}_i - x_i)\right\|_2}_{\textbf{Nonlinearity}}$$

Stability is key here.

Approximate Minimal Polynomial Extrapolation

Stability.

- The iterations span a Krylov subspace

$$\mathcal{K}_k = \operatorname{span}\left\{U_0, AU_0, ..., A^{k-1}U_0\right\}$$

  so the matrix U in AMPE is a **Krylov matrix.**

- Similar to **Hankel or Toeplitz** case. U^TU has a condition number typically growing exponentially with dimension [Tyrtyshnikov, 1994].

- In fact, the Hankel, Toeplitz and Krylov problems are directly connected, hence the link with Levinson-Durbin [Heinig and Rost, 2011].

- For generic optimization problems, eigenvalues are perturbed by deviations from the linear model, which can make the situation even worse.

Be wise, regularize . . .

Outline

- Introduction
- Minimal Polynomial Extrapolation
- **Regularized MPE**
- Numerical results

Regularized Minimal Polynomial Extrapolation

Regularized AMPE. Add a regularization term to AMPE.

- Regularized formulation of problem (AMPE),

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

- Solution given by a linear system of size $k+1$.

$$c^*_\lambda = \frac{(U^TU+\lambda I)^{-1}\mathbf{1}}{\mathbf{1}^T(U^TU+\lambda I)^{-1}\mathbf{1}} \tag{3}$$

Regularized Minimal Polynomial Extrapolation

RMPE algorithm.

Input: Sequence $\{x_0, x_1, ..., x_{k+1}\}$, parameter $\lambda > 0$

1: Form $U = [x_1 - x_0, \; ..., \; x_{k+1} - x_k]$

2: Solve the linear system $(U^T U + \lambda I)z = \mathbf{1}$

3: Set $c = z/(z^T \mathbf{1})$

Output: Return $\sum_{i=0}^{k} c_i x_i$, approximating the optimum x^*

Regularized Minimal Polynomial Extrapolation

Regularized AMPE. Define

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

Proposition [Scieur, d'Aspremont, and Bach, 2016]

Error bounds *Let matrices* $X = [x_0, x_1, ..., x_k]$, $\tilde{X} = [x_0, \tilde{x}_1, ..., \tilde{x}_k]$ *and scalar* $\kappa = \|(A-I)^{-1}\|_2$. *Suppose* $\tilde{c}^*_\lambda$ *solves problem* (RMPE) *and assume* $A = g'(x^*)$ *symmetric with* $0 \preceq A \preceq \sigma I$ *where* $\sigma < 1$. *Let us write the perturbation matrices* $P = \tilde{U}^T\tilde{U} - U^TU$ *and* $\mathcal{E} = (X - \tilde{X})$. *Then*

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

where

$$C(\mathcal{E},P,\lambda) = \left(\kappa^2 + \frac{1}{\lambda}\left(1+\frac{\|P\|_2}{\lambda}\right)^2\left(\|\mathcal{E}\|_2 + \kappa\frac{\|P\|_2}{2\sqrt{\lambda}}\right)^2\right)^{\frac{1}{2}}$$

Regularized Minimal Polynomial Extrapolation

Proposition [Scieur et al., 2016]

Asymptotic acceleration *Using the gradient method with stepsize in* $]0, \frac{2}{L}[$ *on a* L*-smooth,* μ*-strongly convex function* f *with Lipschitz-continuous Hessian of constant* M*.*

$$\|\tilde{X}\tilde{c}_\lambda^* - x^*\|_2 \leq \kappa \left(1 + \frac{(1+\frac{1}{\beta})^2}{4\beta^2}\right)^{1/2} \frac{2\zeta^k}{1+\zeta^{2k}} \|x_0 - x^*\|$$

with

$$\zeta = \frac{1-\sqrt{\mu/L}}{1+\sqrt{\mu/L}}$$

for $\|x_0 - x^*\|$ *small enough, where* $\lambda = \beta\|P\|_2$ *and* $\kappa = \frac{L}{\mu}$ *is the condition number of the function* $f(x)$*.*

We (asymptotically) recover the accelerated rate in [Nesterov, 1983].

Regularized Minimal Polynomial Extrapolation

Stochastic optimization. Noisy oracles on iterates (in practice, gradients) $\tilde{x}_{t+1} = g(\tilde{x}_t) + \eta_{t+1}$, where η_t is noise term (independent). Equivalent to

$$\tilde{x}_{t+1} = x^* + G(\tilde{x}_t - x^*) + \varepsilon_{t+1},$$

where $\|\mathbf{E}[\varepsilon_t]\| \leq \nu$ and ε_t has bounded variance $\Sigma_t \preceq (\sigma^2/d)I$ with

$$\tau \triangleq \frac{\nu + \sigma}{\|x_0 - x^*\|}.$$

Proposition [Scieur, d'Aspremont, and Bach, 2017]

Error bounds *The accuracy of AMPE applied to the sequence* $\{\tilde{x}_0, ..., \tilde{x}_k\}$ *is bounded by*

$$\frac{\mathbf{E}\left[\|\sum_{i=0}^k \tilde{c}_i^\lambda \tilde{x}_i - x^*\|\right]}{\|x_0 - x^*\|} \leq \left(S_\kappa(k,\bar{\lambda})\sqrt{\frac{1}{\kappa^2} + \frac{O(\tau^2(1+\tau)^2)}{\bar{\lambda}^3}} + O\left(\sqrt{\tau^2 + \frac{\tau^2(1+\tau^2)}{\bar{\lambda}}}\right)\right)$$

Regularized Minimal Polynomial Extrapolation

Stochastic optimization.

- When the noise scale $\tau \to 0$, if $\bar{\lambda} = \Theta(\tau^s)$ with $s \in]0, \frac{2}{3}[$, we recover the accelerated rate

$$\mathbf{E}\left[\|\sum_{i=0}^{k} \tilde{c}_i^{\lambda} \tilde{x}_i - x^*\|\right] \leq \frac{1}{\kappa} \left(\frac{1-\sqrt{\kappa}}{1+\sqrt{\kappa}}\right)^k \|x_0 - x^*\|.$$

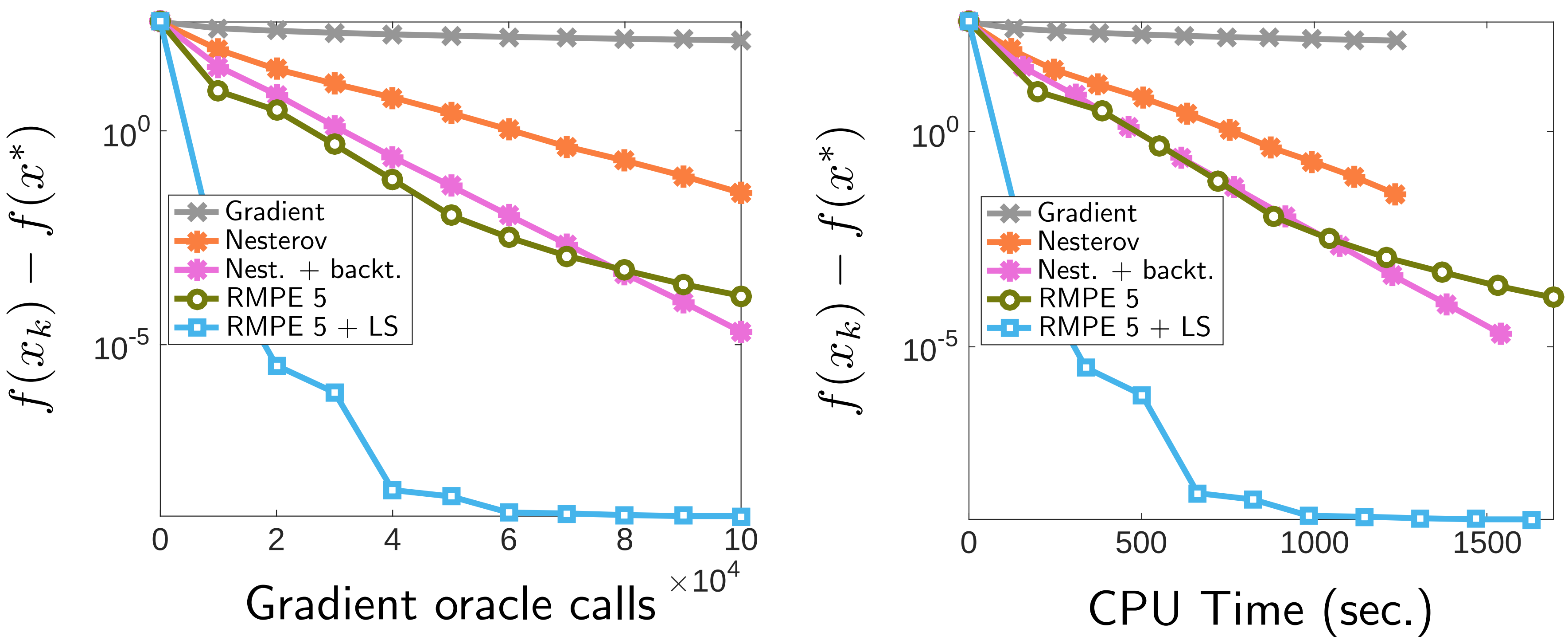
- If $\lambda \to \infty$, we recover the averaged gradient

$$\mathbf{E}\left[\|\sum_{i=0}^{k} \tilde{c}_i^{\lambda} \tilde{x}_i - x^*\|\right] \to \mathbf{E}\left[\left\|\frac{1}{k+1} \sum_{i=0}^{k} \tilde{x}_i - x^*\right\|\right]$$

Outline

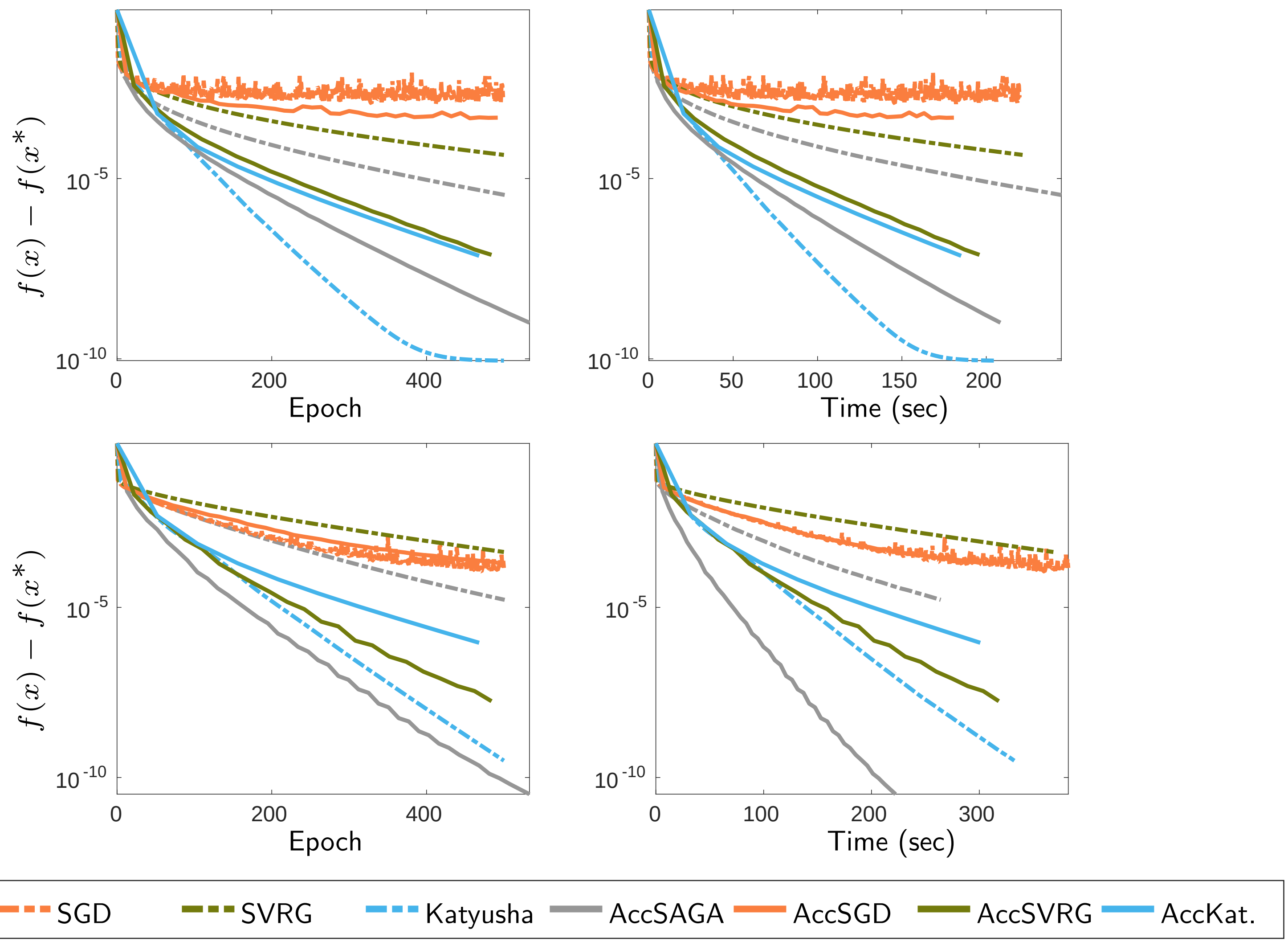
- Introduction
- Minimal Polynomial Extrapolation
- Regularized MPE
- **Numerical results**

Numerical Results



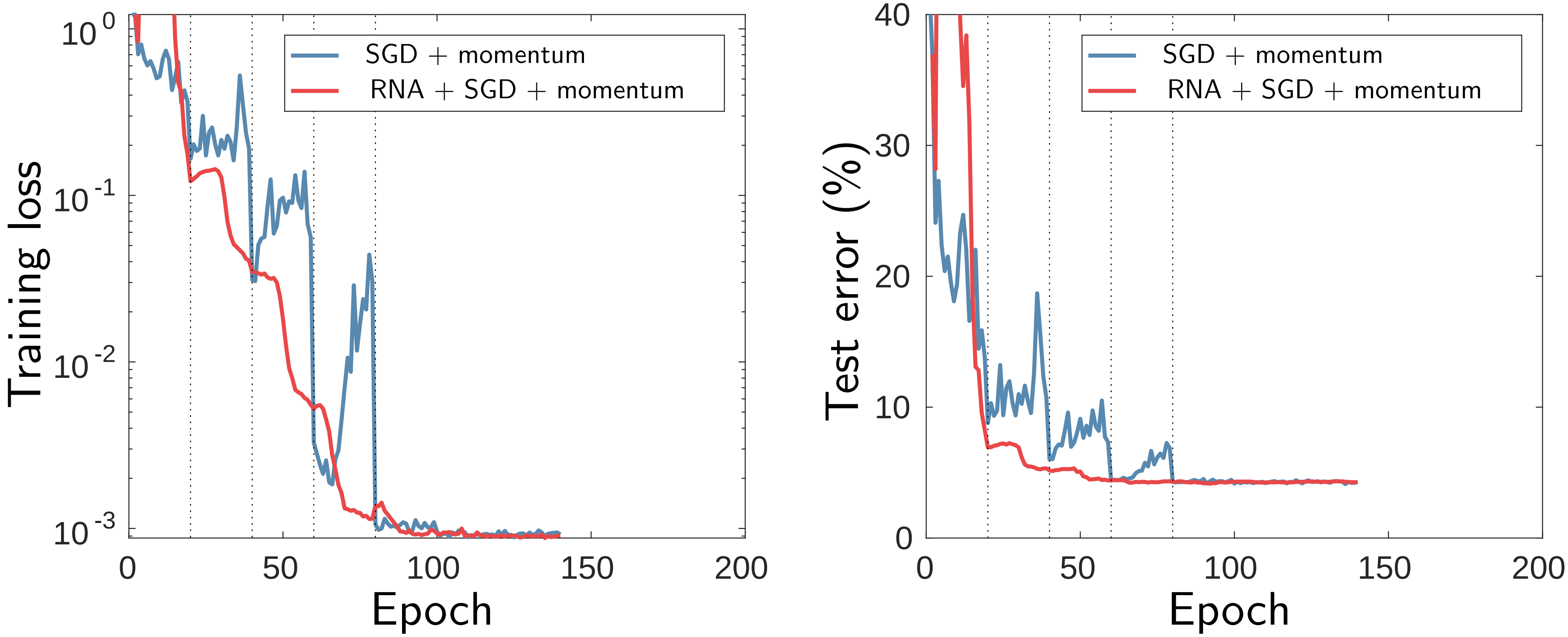
Logistic regression with ℓ_2 regularizartion, on *Madelon Dataset* (500 features, 2000 data points), solved using several algorithms. The penalty parameter has been set to 10^2 in order to have a condition number equal to 1.2×10^9.

Numerical Results



Optimization of quadratic loss (*Top*) and logistic loss (*Bottom*) with several algorithms, using the `Sid` dataset with bad conditioning. The experiments are done in Matlab. *Left:* Error vs epoch number. *Right:* Error vs time.

Numerical Results



Convergence acceleration. Training Resnet-28-10 on CIFAR data set. Value reached by the current iterate versus extrapolated one (from the last 15 iterates). Training loss on the *left*, testing error on the *right*. Restarting the training periodically at the extrapolated point. Vertical lines mark learning rate decreases.

Conclusion

Postprocessing works.

- Simple **postprocessing** step.
- Marginal complexity, can be performed in parallel.
- Significant convergence speedup over optimal methods.
- Adaptive. Does not need knowledge of smoothness parameters.

Work in progress. . .

- Extrapolating accelerated methods.
- Constrained problems.
- Better handling of smooth functions.
- . . .

Open problems

- **Regularization.** How do we account for the fact that we are estimating the limit of a VAR sequence with a fixed point?
- The VAR matrix A is formed implicitly, but we have some information on its spectrum through smoothness.
- Explicit bounds on the **regularized Chebyshev problem,**

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

Preprints on ArXiv, NIPS 2016, 2017.

*

References

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

C Brezinski. Accélération de la convergence en analyse numérique. *Lecture notes in mathematics (ISSN 0075-8434*, (584), 1977.

RP Eddy. Extrapolating to the limit of a vector sequence. *Information linkage between applied mathematics and industry*, pages 387–396, 1979.

M. Hardt. The zen of gradient descent. *Mimeo*, 2013.

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

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

Arkadi S Nemirovskiy and Boris T Polyak. Iterative methods for solving linear ill-posed problems under precise information. *ENG. CYBER.*, (4):50–56, 1984.

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.

D. Scieur, A. d'Aspremont, and F. Bach. *Regularized Nonlinear Acceleration*. NIPS, 2016.

Damien Scieur, Alexandre d'Aspremont, and Francis Bach. Nonlinear acceleration of stochastic algorithms. *arXiv preprint arXiv:1706.07270*, 2017.

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.

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

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

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.