Complexity Guarantees for Polyak Steps with Momentum

Mathieu Barré, Adrien Taylor and Alexandre d'Aspremont

Conference on Learning Theory (COLT) - July 2020

What is this work about?

What is this work about?

Adaptive optimization strategy with smooth and strongly convex objective based on Polyak steps.

What is this work about?

Adaptive optimization strategy with smooth and strongly convex objective based on Polyak steps.

(B. Polyak 1987), (Nedic & Bertsekas 2001), . . .

What is this work about?

Adaptive optimization strategy with smooth and strongly convex objective based on Polyak steps.

(B. Polyak 1987), (Nedic & Bertsekas 2001), ...

Computer-aided worst case analysis.

What is this work about?

Adaptive optimization strategy with smooth and strongly convex objective based on Polyak steps.

(B. Polyak 1987), (Nedic & Bertsekas 2001), . . .

Computer-aided worst case analysis.

(Drori & Teboulle 2014), (Lessard, Recht & Packard 2016), (Taylor, Hendrickx & Glineur 2017), and a few others.

Polyak stepsizes

Famous stepsizes rule for solving the convex problem

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)$$

using gradient steps.

Polyak stepsizes

Famous stepsizes rule for solving the convex problem

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)$$

using gradient steps.

Iterates of the form $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$. With $\gamma_k = \frac{f(x_k) - f_\star}{\|\nabla f(x_k)\|^2}$ (Polyak step).

Polyak stepsizes

Famous stepsizes rule for solving the convex problem

$$f_\star = \min_{x \in \mathbb{R}^d} f(x)$$

using gradient steps.

Iterates of the form $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$. With $\gamma_k = \frac{f(x_k) - f_\star}{\|\nabla f(x_k)\|^2}$ (Polyak step).

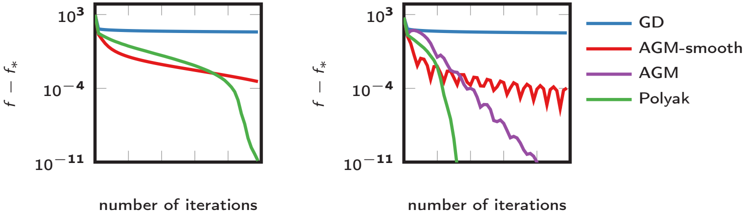

Figure: Regularized logistic regression. Left: regularization parameter 10^{-7}. Right: regularization parameter 10^{-4}.

Performance Estimation approach on Polyak steps

For simplicity study the variant $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$, $\gamma_k = 2\frac{f(x_k)-f_\star}{\|\nabla f(x_k)\|^2}$.

Performance Estimation approach on Polyak steps

For simplicity study the variant $x_{k+1} = x_k - \gamma_k \nabla f(x_k)\,, \gamma_k = 2\frac{f(x_k)-f_\star}{\|\nabla f(x_k)\|^2}$.

Looking for the smallest $\rho \geq 0$ such that $\|x_{k+1} - x_*\|^2 \leq \rho \|x_k - x_*\|^2$.

Performance Estimation approach on Polyak steps

For simplicity study the variant $x_{k+1} = x_k - \gamma_k \nabla f(x_k)\,, \gamma_k = 2\,\frac{f(x_k)-f_\star}{\|\nabla f(x_k)\|^2}$.

Looking for the smallest $\rho \geq 0$ such that $\|x_{k+1} - x_*\|^2 \leq \rho \|x_k - x_*\|^2$.

$$\rho := \quad \text{maximize} \quad \frac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2}$$

Performance Estimation approach on Polyak steps

For simplicity study the variant $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$, $\gamma_k = 2\frac{f(x_k)-f_\star}{\|\nabla f(x_k)\|^2}$.

Looking for the smallest $\rho \geq 0$ such that $\|x_{k+1} - x_*\|^2 \leq \rho \|x_k - x_*\|^2$.

$$\rho := \quad \text{maximize} \quad \frac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2}$$
$$\text{subject to} \quad x_{k+1} = x_k - 2\frac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2}\nabla f(x_k),$$

Performance Estimation approach on Polyak steps

For simplicity study the variant $x_{k+1} = x_k - \gamma_k \nabla f(x_k)\,, \gamma_k = 2\,\frac{f(x_k)-f_\star}{\|\nabla f(x_k)\|^2}$.

Looking for the smallest $\rho \geq 0$ such that $\|x_{k+1} - x_*\|^2 \leq \rho \|x_k - x_*\|^2$.

$$
\begin{aligned}
\rho := \quad & \text{maximize} \quad && \frac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2} \\
& \text{subject to} \quad && x_{k+1} = x_k - 2\,\frac{f(x_k)-f_*}{\|\nabla f(x_k)\|^2}\nabla f(x_k), \\
& && f \in \mathcal{F}_{\mu,L},\; x_k \in \mathbb{R}^d, d \in \mathbb{N}.
\end{aligned}
$$

Performance Estimation approach on Polyak steps

For simplicity study the variant $x_{k+1} = x_k - \gamma_k \nabla f(x_k)$, $\gamma_k = 2 \frac{f(x_k) - f_\star}{\|\nabla f(x_k)\|^2}$.

Looking for the smallest $\rho \geq 0$ such that $\|x_{k+1} - x_*\|^2 \leq \rho \|x_k - x_*\|^2$.

$$
\begin{aligned}
\rho := \quad & \text{maximize} \quad && \frac{\|x_{k+1} - x_*\|^2}{\|x_k - x_*\|^2} \\
& \text{subject to} \quad && x_{k+1} = x_k - 2 \tfrac{f(x_k) - f_*}{\|\nabla f(x_k)\|^2} \nabla f(x_k), \\
& && f \in \mathcal{F}_{\mu,L},\ x_k \in \mathbb{R}^d, d \in \mathbb{N}.
\end{aligned}
$$

Problem : Infinite dimensional

Performance Estimation approach on Polyak steps

Work with discrete version of f (Drori & Teboulle 2014), (Taylor, Hendrickx & Glineur 2017).

Performance Estimation approach on Polyak steps

Work with discrete version of f (Drori & Teboulle 2014), (Taylor, Hendrickx & Glineur 2017).

Optimization problem can be relaxed and cast to a SDP.

$$
\begin{aligned}
\rho := \quad & \text{maximize} && 1 + 4\frac{f_k - f_*}{G_k}\frac{GX_k}{X_k} + 4\frac{(f_k - f_*)^2}{G_k X_k} \\
& \text{subject to} && f_k - f_* + GX_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\
& && f_* - f_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\
& && \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0
\end{aligned}
$$

in the variables $X_k, G_k, GX_k, f_k, f_* \in \mathbb{R}$.

Performance Estimation approach on Polyak steps

Work with discrete version of f (Drori & Teboulle 2014), (Taylor, Hendrickx & Glineur 2017).

Optimization problem can be relaxed and cast to a SDP.

$$
\begin{array}{rll}
\rho := & \text{maximize} & 1 + 4\frac{f_k - f_*}{G_k}\frac{GX_k}{X_k} + 4\frac{(f_k - f_*)^2}{G_k X_k} \\
& \text{subject to} & f_k - f_* + GX_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\
& & f_* - f_k + \frac{1}{2L}G_k + \frac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \frac{2}{L}GX_k + \frac{1}{L^2}G_k\right) \leq 0 \\
& & \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0
\end{array}
$$

in the variables $X_k, G_k, GX_k, f_k, f_* \in \mathbb{R}$.

Problem : nonlinear objective.

Performance Estimation approach on Polyak steps

Add $\gamma = 2\frac{f_k - f_\star}{G_k}$ as constraint.

Performance Estimation approach on Polyak steps

Add $\gamma = 2\frac{f_k - f_\star}{G_k}$ as constraint.

For every step size value γ, we can solve the linear SDP

$$\begin{aligned}
\rho(\gamma) := \quad \max. \quad & 1 + 2\gamma GX_k + 2(f_k - f_*)\gamma \\
\text{s.t.} \quad & f_k - f_* + GX_k + \tfrac{1}{2L} G_k + \tfrac{\mu}{2(1-\frac{\mu}{L})} \left(X_k + \tfrac{2}{L} GX_k + \tfrac{1}{L^2} G_k \right) \leq 0 \\
& f_* - f_k + \tfrac{1}{2L} G_k + \tfrac{\mu}{2(1-\frac{\mu}{L})} \left(X_k + \tfrac{2}{L} GX_k + \tfrac{1}{L^2} G_k \right) \leq 0 \\
& \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0 \\
& G_k \gamma = 2(f_k - f_*)
\end{aligned}$$

Note that $\rho = \max_\gamma \rho(\gamma)$.

Performance Estimation approach on Polyak steps

Add $\gamma = 2\frac{f_k - f_\star}{G_k}$ as constraint.

For every step size value γ, we can solve the linear SDP

$$\begin{aligned}\rho(\gamma) := \quad \max. \quad & 1 + 2\gamma GX_k + 2(f_k - f_*)\gamma \\ \text{s.t.} \quad & f_k - f_* + GX_k + \tfrac{1}{2L} G_k + \tfrac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \tfrac{2}{L} GX_k + \tfrac{1}{L^2} G_k\right) \leq 0 \\ & f_* - f_k + \tfrac{1}{2L} G_k + \tfrac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \tfrac{2}{L} GX_k + \tfrac{1}{L^2} G_k\right) \leq 0 \\ & \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0 \\ & G_k\gamma = 2(f_k - f_*)\end{aligned}$$

Note that $\rho = \max_\gamma \rho(\gamma)$.

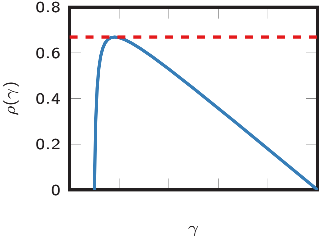

Figure: $\mu = 0.1$ and $L = 1$.

Performance Estimation approach on Polyak steps

Add $\gamma = 2\frac{f_k - f_\star}{G_k}$ as constraint.

For every step size value γ, we can solve the linear SDP

$$
\begin{aligned}
\rho(\gamma) := \quad \max. \quad & 1 + 2\gamma GX_k + 2(f_k - f_*)\gamma \\
\text{s.t.} \quad & f_k - f_* + GX_k + \tfrac{1}{2L} G_k + \tfrac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \tfrac{2}{L} GX_k + \tfrac{1}{L^2} G_k\right) \leq 0 \\
& f_* - f_k + \tfrac{1}{2L} G_k + \tfrac{\mu}{2(1-\frac{\mu}{L})}\left(X_k + \tfrac{2}{L} GX_k + \tfrac{1}{L^2} G_k\right) \leq 0 \\
& \begin{pmatrix} X_k & GX_k \\ GX_k & G_k \end{pmatrix} \succcurlyeq 0 \\
& G_k \gamma = 2(f_k - f_*)
\end{aligned}
$$

Note that $\rho = \max_\gamma \rho(\gamma)$.

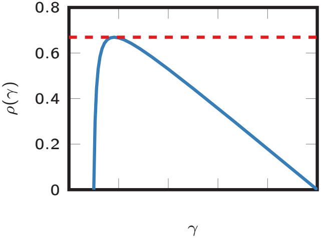

Figure: $\mu = 0.1$ and $L = 1$.

(see the paper for an explicit expression of $\rho(\gamma)$)

Limit of worst case analysis

One can show $\rho = \left(\frac{L-\mu}{L+\mu}\right)^2$.

Convergence rate ρ doesn't explain why classical Polyak steps behave so well in pratice.

Limit of worst case analysis

One can show $\rho = \left(\frac{L-\mu}{L+\mu}\right)^2$.

Convergence rate ρ doesn't explain why classical Polyak steps behave so well in pratice.

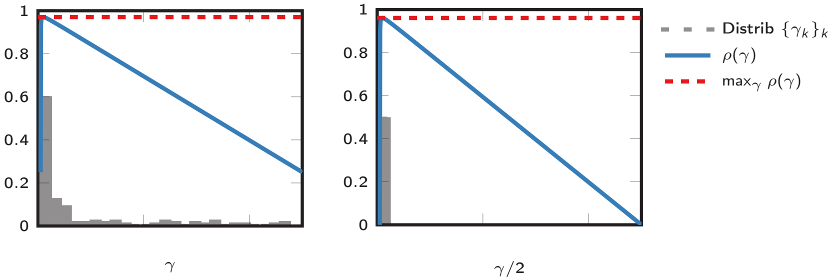

Figure: Empirical distribution of stepsizes $\{\gamma_k\}_k$. Left : Classical Polyak. Right : Variant with extra 2.

Accelerated algorithm with Polyak steps style momentum

Introduce strong convexity estimate in Accelerated gradient descent with momentum (Nesterov 2018).

Accelerated algorithm with Polyak steps style momentum

Introduce strong convexity estimate in Accelerated gradient descent with momentum (Nesterov 2018).

Algorithm 2 Accelerated gradient method with Polyak steps momentum

Input: $x_0 \in \mathbb{R}^n$, $f_* \in \mathbb{R}$, L smoothness constant.

$y_0 = x_0$,

for $k \geq 0$ **do**

$y_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$

$\tilde{\mu}_k = \frac{\|\nabla f(y_{k+1})\|^2}{2(f(y_{k+1}) - f_\star)}$ and $\beta_k = \frac{\sqrt{L} - \sqrt{\tilde{\mu}_k}}{\sqrt{L} + \sqrt{\tilde{\mu}_k}}$

$x_{k+1} = y_{k+1} + \beta_k(y_{k+1} - y_k)$

end for

Output: y_{k+1}

Accelerated algorithm with Polyak steps style momentum

Complexity bounds (B.,Taylor,d'Aspremont 2020)

$$f(y_N) - f_* \leq C \left(1 + \sqrt[4]{\frac{\mu}{L}}\right)^{-N}$$

Accelerated algorithm with Polyak steps style momentum

Complexity bounds (B.,Taylor,d'Aspremont 2020)

$$f(y_N) - f_* \leq C\left(1 + \sqrt[4]{\frac{\mu}{L}}\right)^{-N}$$

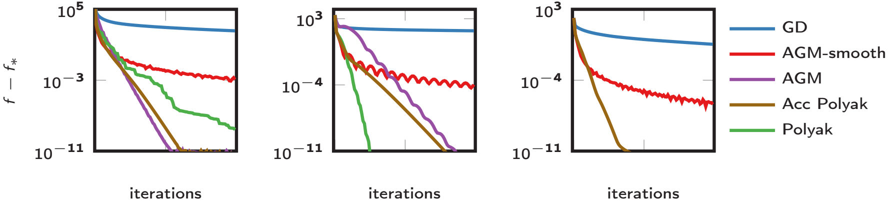

Figure: Numerical experiments on Musk Dataset.
Left : Linear reg. Middle : Log reg. Right : LASSO.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.
- ☺ Fast in practice.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.
- ☺ Fast in practice.
- ☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.
- ☺ Fast in practice.
- ☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate.
- ☹ Requires L.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.
- ☺ Fast in practice.
- ☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate.
- ☹ Requires L.
- ☹☹ Requires knowledge of $f_\star$.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.
- ☺ Fast in practice.
- ☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate. $\rightarrow$ Might be artifact from the proof's form
- ☹ Requires L.
- ☹☹ Requires knowledge of $f_\star$.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.
- ☺ Comes with complexity guarantees.
- ☺ Fast in practice.
- ☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate. $\rightarrow$ Might be artifact from the proof's form
- ☹ Requires L. $\rightarrow$ Classical backtracking arguments do not apply as is.
- ☹☹ Requires knowledge of $f_\star$.

Accelerated algorithm with Polyak steps style momentum

☺ Simple formulation, no tuning.

☺ Comes with complexity guarantees.

☺ Fast in practice.

☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate. $\rightarrow$ Might be artifact from the proof's form

☹ Requires L. $\rightarrow$ Classical backtracking arguments do not apply as is.

☹☹ Requires knowledge of $f_\star$.

$\rightarrow$ *(i)* Can do a 2 phases algorithm with $\sim 1 - \frac{1}{2}\sqrt[4]{\frac{\mu}{L}}$ rate not using f_* but not as fast in practice.

Accelerated algorithm with Polyak steps style momentum

- ☺ Simple formulation, no tuning.

- ☺ Comes with complexity guarantees.

- ☺ Fast in practice.

- ☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate. $\rightarrow$ Might be artifact from the proof's form

- ☹ Requires L. $\rightarrow$ Classical backtracking arguments do not apply as is.

☹☹ Requires knowledge of $f_\star$.

$\rightarrow$ *(i)* Can do a 2 phases algorithm with $\sim 1 - \frac{1}{2}\sqrt[4]{\frac{\mu}{L}}$ rate not using f_* but not as fast in practice.
$\rightarrow$*(ii)* Can also use different estimates $\tilde{\mu}_k$ that do not use f_* with fast performances but no accelerated rate yet.

Accelerated algorithm with Polyak steps style momentum

☺ Simple formulation, no tuning.

☺ Comes with complexity guarantees.

☺ Fast in practice.

☹ $\sqrt[4]{\cdot}$ in guaranteed rate instead of $\sqrt{\cdot}$ in convergence rate. $\rightarrow$ Might be artifact from the proof's form

☹ Requires L. $\rightarrow$ Classical backtracking arguments do not apply as is.

☹☺ Requires knowledge of $f_\star$.

$\rightarrow$ *(i)* Can do a 2 phases algorithm with $\sim 1 - \frac{1}{2}\sqrt[4]{\frac{\mu}{L}}$ rate not using f_* but not as fast in practice.
$\rightarrow$*(ii)* Can also use different estimates $\tilde{\mu}_k$ that do not use f_* with fast performances but no accelerated rate yet.

Conclusion

Why Polyak steps ?

Conclusion

Why Polyak steps ? Probably simplest adaptive algorithm

Conclusion

Why Polyak steps ? Probably simplest adaptive algorithm $\rightarrow$ good start.

Conclusion

Why Polyak steps ? Probably simplest adaptive algorithm $\rightarrow$ good start.

- Used Performance Estimation Program in the context of adaptive methods.

Conclusion

Why Polyak steps ? Probably simplest adaptive algorithm $\rightarrow$ good start.

- Used Performance Estimation Program in the context of adaptive methods.
- Derive optimal bounds for gradient descent with Polyak steps.

Conclusion

Why Polyak steps ? Probably simplest adaptive algorithm $\rightarrow$ good start.

- Used Performance Estimation Program in the context of adaptive methods.
- Derive optimal bounds for gradient descent with Polyak steps.
- A step in the direction of (proved) simple and fully adaptive accelerated algorithm.

Thanks!

Happy to answer (almost live) questions

"Complexity Guarantess for Polyak Steps with Momentum"