automating the analysis and design of large-scale optimization algorithms

Laurent Lessard
University of Wisconsin–Madison

with Ben Recht and Andy Packard, UC Berkeley

ICCOPT 2016 – Tokyo, Japan
August 8, 2016
In this talk: a general framework for obtaining performance guarantees for first-order optimization algorithms.

- uses a dynamical systems perspective and tools from robust control theory and semidefinite programming.
- universal: the same method can analyze a variety of algorithms under various assumptions.
- efficient: requires solving a very small LMI.
- it works! recovers or improves on existing results and provides new insights.
Gradient method
\[ x_{k+1} = x_k - \alpha \nabla f(x_k) \]

Heavy ball method
\[ x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1}) \]

Nesterov’s accelerated method
\[ y_k = x_k + \beta (x_k - x_{k-1}) \]
\[ x_{k+1} = y_k - \alpha \nabla f(y_k) \]

contours of \( f(x) \)
(quadratic)
Robust algorithm selection

\[ f \in \mathcal{S} : \text{function we’d like to minimize} \]

\[ G \in \mathcal{G} : \text{algorithm we’re going to use} \]

\[ G_{\text{opt}} = \arg \min_{G \in \mathcal{G}} \left( \max_{f \in \mathcal{S}} \text{cost}(f, G) \right) \]

Similar problem for a finite number of iterations:

- Drori, Teboulle (2012)
- Taylor, Hendrickx, Glineur (2016)
$G \in \mathcal{G}$

Gradient method

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

Heavy ball method

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1})$$

Nesterov’s accelerated method

$$x_{k+1} = x_k - \alpha \nabla f(x_k + \beta (x_k - x_{k-1})) + \beta (x_k - x_{k-1})$$

$f \in \mathcal{S}$

Analytically solvable:

Quadratic functions: $f(x) = x^T Q x - p^T x$

with the constraint: $mI \leq Q \leq LI$
Convergence rate: \[ \|x_k - x_\star\| \leq C\rho^k\|x_0 - x_\star\| \]

Iterations to convergence \( \propto \frac{1}{\log \rho} \)
Robust algorithm selection

\( f \in S \): function we’d like to minimize
\( G \in \mathcal{G} \): algorithm we’re going to use

\[
G_{\text{opt}} = \arg \min_{G \in \mathcal{G}} \left( \max_{f \in S} \text{cost}(f, G) \right)
\]

(1) mathematical representation for \( \mathcal{G} \)
(2) mathematical representation for \( S \)
(3) main robustness result
Dynamical system interpretation

Heavy ball: \( x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1}) \)

Define \( u_k := \nabla f(x_k) \) and \( p_k := x_{k-1} \)
Dynamical system interpretation

Heavy ball: \( x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1}) \)

Define \( u_k := \nabla f(x_k) \) and \( p_k := x_{k-1} \)

algorithm (linear, known, \textbf{decoupled})

\[
\begin{bmatrix}
(x_{k+1})_i \\
(p_{k+1})_i
\end{bmatrix}
= \begin{bmatrix}
1 + \beta & -\beta \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
(x_k)_i \\
(p_k)_i
\end{bmatrix}
+ \begin{bmatrix}
-\alpha \\
0
\end{bmatrix}
(u_k)_i
\]

\[
(y_k)_i = \begin{bmatrix}
1 & 0
\end{bmatrix}
\begin{bmatrix}
(x_k)_i \\
(p_k)_i
\end{bmatrix}
\]

\( i = 1, \ldots, N \)

function (nonlinear, uncertain, \textbf{coupled})

\( u_k = \nabla f(y_k) \)
\[
\begin{align*}
\xi_{k+1} &= A\xi_k + Bu_k \\
y_k &= C\xi_k \\
u_k &= \nabla f(y_k)
\end{align*}
\]

For the matrices:

\[
\begin{bmatrix}
A & B \\
C & 0
\end{bmatrix}
= \left\{ \begin{array}{c}
\begin{bmatrix}
1 & -\alpha \\
1 & 0
\end{bmatrix} \\
1 + \beta & -\beta & -\alpha \\
1 & 0 & 0 \\
1 & 0 & 0 \\
1 + \beta & -\beta & -\alpha \\
1 & 0 & 0 \\
1 + \beta & -\beta & 0
\end{array} \right\}
\]

- **Gradient**
- **Heavy ball**
- **Nesterov**
\[ \nabla f(x) : \square \subset \nabla f(x) : \text{linear} \]

\[ \nabla f(x) : \square \subset \nabla f(x) : \text{sector bounded + slope restricted} \]

\[ \nabla f(x) : \square \subset \nabla f(x) : \text{sector bounded} \]

\[ f(x) : \square \subset f(x) : \text{quadratic} \]

\[ f(x) : \square \subset f(x) : \text{strongly convex + Lipschitz gradients} \]

\[ f(x) : \square \subset f(x) : \text{radially quasiconvex} \]
Representing function classes

express as quadratic constraints on \((y, u)\)

\(\nabla f\) is a \textbf{passive} function:

\[u_k y_k \geq 0\]
Representing function classes

express as quadratic constraints on \((y, u)\)

\(\nabla f\) is sector-bounded:

\[
\begin{bmatrix}
y_k \\
u_k
\end{bmatrix}^T \begin{bmatrix}
-2mL & m + L \\
m + L & -2
\end{bmatrix} \begin{bmatrix}
y_k \\
u_k
\end{bmatrix} \geq 0
\]
Representing function classes express as quadratic constraints on \((y, u)\)

\(\nabla f\) is sector-bounded + slope-restricted: constraint on \((y_k, u_k)\) depends on history \((y_0, \ldots, y_{k-1}, u_0, \ldots, u_{k-1})\).
Introduce extra dynamics

- Design dynamics $\Psi$ and multiplier matrix $M$.
- Instead of using $q(u_k, y_k)$, use $z_k^T M z_k$.
- Systematic way of doing this for strong convexity via Zames-Falb multipliers (1968).
- General theory: Integral Quadratic Constraints (Megretski & Rantzer 1997)
\[ \nabla f(u) \]

\[ \begin{bmatrix}
1 & -\alpha \\
1 & 0
\end{bmatrix} \]

\[ \begin{bmatrix}
1 + \beta & -\beta & -\alpha \\
1 & 0 & 0 \\
1 + \beta & -\beta & 0
\end{bmatrix} \]

\[ (\Psi, M) \]

- Gradient
- Heavy ball
- Nesterov
\[
\begin{align*}
\xi_{k+1} &= A\xi_k + Bu_k \\
y_k &= C\xi_k \\
u_k &= \nabla f(y_k) \\
\zeta_{k+1} &= A\Psi\zeta_k + B^y\Psi y_k + B^u\Psi u_k \\
z_k &= C\Psi\zeta_k + D^y\Psi y_k + D^u\Psi u_k
\end{align*}
\]

where \( x_k := \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} \) and \( z \) is quadratically constrained.
Main result

Suppose \( \{x_0, x_1, \ldots\} \) satisfies dynamics
\[
x_{k+1} = \hat{A} x_k + \hat{B} u_k
\]
\[
z_k = \hat{C} x_k + \hat{D} u_k
\]
where \( \{z_0, z_1, \ldots\} \) is constrained by
\[
\sum_{k=0}^{T} \rho^{-2k} (z_k - z_*)^T M (z_k - z_*) \geq 0 \quad \text{for all } T
\]

If there exists \( P > 0 \) and \( \lambda \geq 0 \) such that
\[
\begin{bmatrix}
\hat{A}^T P \hat{A} - \rho^2 P & \hat{A}^T P \hat{B} \\
\hat{B}^T P \hat{A} & \hat{B}^T P \hat{B}
\end{bmatrix} + \lambda \begin{bmatrix}
\hat{C} & \hat{D}
\end{bmatrix}^T M \begin{bmatrix}
\hat{C} & \hat{D}
\end{bmatrix} \leq 0
\]
then \( \|x_k - x_*\| \leq \sqrt{\text{cond}(P)} \rho^k \|x_0 - x_*\| \) for all \( k \).

Size of LMI does not grow with problem dimension! e.g. \( P \in \mathbb{S}^{3 \times 3} \), LMI \( \in \mathbb{S}^{4 \times 4} \)
main results:
analytic and numerical
Gradient method

\[ x_{k+1} = x_k - \alpha \nabla f(x_k) \]

analytic solution! Same rate for: quadratics, strongly convex, or quasiconvex functions.
Nesterov’s method

\[ x_{k+1} = x_k - \alpha \nabla f(x_k + \beta(x_k - x_{k-1})) + \beta(x_k - x_{k-1}) \]

- Cannot certify stability for quasiconvex functions
- IQC bound **improves** upon best known bound!
Heavy ball method

\[ x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta (x_k - x_{k-1}) \]

- Cannot certify stability for quasiconvex functions
- Cannot certify stability for strongly convex functions
The heavy ball method is **not** stable!

counterexample: \( f(x) = \begin{cases} \frac{25}{2} x^2 & x < 1 \\ \frac{1}{2} x^2 + 24x - 12 & 1 \leq x < 2 \\ \frac{25}{2} x^2 - 24x + 36 & x \geq 2 \end{cases} \)

and start the heavy ball iteration at \( x_0 = x_1 \in [3.07, 3.46] \).

- \( L/m = 25 \)
- heavy ball iterations converge to a limit cycle
uncharted territory:
noise robustness and algorithm design
Noise robustness

The $\Delta_\delta$ block is uncertain multiplicative noise:

$$\|u_k - w_k\| \leq \delta\|w_k\|$$

How does an algorithm perform in the presence of noise?
Gradient method, \( \alpha = \frac{2}{L+m} \) (optimal stepsize with no noise)

Gradient method, \( \alpha = \frac{1}{L} \) (more conservative stepsize)
Nesterov’s method (strongly convex \( f \), with noise)

- Nesterov’s method is not robust to noise.

  can we have it all? (robustness AND performance)
Brute force approach

- test all strictly proper $G$ of degree 2
- parameterization in terms of $(\alpha, \beta_1, \beta_2)$:

$$x_{k+1} = x_k - \alpha \nabla f(x_k + \beta_2(x_k - x_{k-1})) + \beta_1(x_k - x_{k-1})$$

Special cases:

$$(\alpha, \beta_1, \beta_2) = \begin{cases} 
(\alpha, 0, 0) & \text{Gradient} \\
(\alpha, \beta, 0) & \text{Heavy ball} \\
(\alpha, \beta, \beta) & \text{Nesterov}
\end{cases}$$
Optimal designs over \((\alpha, \beta_1, \beta_2)\)

- Faster than the gradient method \textbf{and}
- more robust to noise than Nesterov’s method
- automatic algorithm design is possible!
What we have (so far!)

L, Recht, Packard (SIOPT’16)
  • unified framework for algorithm analysis
  • gradient, heavy ball, Nesterov
  • constrained optimization, noise robustness

Nishihara, L, Recht, Packard, Jordan (ICML’15)
  • application to robust ADMM tuning

Boczar, L, Recht (CDC’15)
  • control theory version
Thank you!

- Manuscripts + code available: www.laurentlessard.com
- If you’re interested, come talk to me!