Lyapunov Functions for First-Order Methods
Tight Automated Convergence Guarantees

Adrien Taylor  Bryan Van Scoy  Laurent Lessard

35th International Conference on Machine Learning
Stockholm, Sweden
July 11, 2018
We consider first-order methods to solve the following problem:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in \mathbb{R}^d
\end{align*}
\]

**This work**

**Automatic** analysis of optimization methods by solving a small-sized semidefinite program. Convergence rate is **provably tight** when

- \( f \) is \( L \)-smooth and \( \mu \)-strongly convex (denoted \( f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \)), and
- the method is **iterative** with **fixed stepsizes**.

Combines:
- performance estimation problems (Drori & Teboulle, 2014)
- integral quadratic constraints (Lessard, Recht, Packard, 2016)

Uses smooth strongly convex interpolation (Taylor, Hendrickx, Glineur, 2017)
First-order iterative fixed-step method

\[ y_k = \sum_{j=0}^{N} \gamma_j x_{k-j} \]
\[ x_{k+1} = \sum_{j=0}^{N} \beta_j x_{k-j} - \alpha \nabla f(y_k) \]

- **degree** \( N \)
- **stepsizes** \( \alpha, \beta_j, \gamma_j \)
- **initial conditions** \( x_j \in \mathbb{R}^d \) for \( j = -N, \ldots, 0 \)

**Extensions:**
- linesearch (or subspace search)
- restart (fixed/adaptive)
First-order iterative fixed-step method

\[ y_k = \sum_{j=0}^{N} \gamma_j \, x_{k-j} \]

\[ x_{k+1} = \sum_{j=0}^{N} \beta_j \, x_{k-j} - \alpha \nabla f(y_k) \]

Main result

\[
\text{SDP}(\mu, L, \rho, \alpha, \beta_j, \gamma_j) \text{ is feasible} \]

\[
\downarrow
\]

the method converges linearly for all \( f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d) \) with rate at least \( \rho \)
First-order iterative fixed-step method

\[
y_k = \sum_{j=0}^{N} \gamma_j x_{k-j}
\]

\[
x_{k+1} = \sum_{j=0}^{N} \beta_j x_{k-j} - \alpha \nabla f(y_k)
\]

Main result

SDP(\(\mu, L, \rho, \alpha, \beta_j, \gamma_j\)) is feasible

\[\Leftrightarrow\]

there exists a quadratic Lyapunov function

\[\Downarrow\]

the method converges linearly for all \(f \in \mathcal{F}_{\mu,L}(\mathbb{R}^d)\) with rate at least \(\rho\)
Lyapunov function

Fundamental tool from control theory that can be used to verify stability of a dynamical system (Kalman & Bertram, 1960).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Control</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>ξ</td>
<td>state</td>
<td>iterate, function, &amp; gradient values</td>
</tr>
<tr>
<td>ξ⋆</td>
<td>fixed-point</td>
<td>optimal solution</td>
</tr>
<tr>
<td>V(ξ)</td>
<td>energy in system</td>
<td>distance from optimality</td>
</tr>
</tbody>
</table>

- **nonnegative** \( V(ξ) \geq 0 \) for all \( ξ \)
- **zero at fixed-point** \( V(ξ) = 0 \) if and only if \( ξ = ξ⋆ \)
- **radially unbounded** \( V(ξ) \to \infty \) as \( ||ξ|| \to \infty \)
- **decreasing** \( V(ξ_{k+1}) \leq ρ^2 V(ξ_k) \) for all \( k \)

Existence of a Lyapunov function provides a certificate of convergence.
Lyapunov function

Fundamental tool from control theory that can be used to verify stability of a dynamical system (Kalman & Bertram, 1960).

<table>
<thead>
<tr>
<th>Notation</th>
<th>Control</th>
<th>Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi$</td>
<td>state</td>
<td>iterate, function, &amp; gradient values</td>
</tr>
<tr>
<td>$\xi_*$</td>
<td>fixed-point</td>
<td>optimal solution</td>
</tr>
<tr>
<td>$V(\xi)$</td>
<td>energy in system</td>
<td>distance from optimality</td>
</tr>
</tbody>
</table>

Suppose we choose the state as $\xi_k = \{(x_i, f(y_i), \nabla f(y_i))\}_{i=k-N}^N$ where $x_k$ are the iterates. We then use the quadratic Lyapunov function candidate:

$$V(\xi_k) = \begin{bmatrix} x_k - x_* \\ \vdots \\ x_{k-N} - x_* \\ \nabla f(y_k) \\ \vdots \\ \nabla f(y_{k-N}) \end{bmatrix}^T (P \otimes I_d) \begin{bmatrix} x_k - x_* \\ \vdots \\ x_{k-N} - x_* \\ \nabla f(y_k) \\ \vdots \\ \nabla f(y_{k-N}) \end{bmatrix} + p^T \begin{bmatrix} f(y_k) - f(x_*) \\ \vdots \\ f(y_{k-N}) - f(x_*) \end{bmatrix}$$
Numerical results

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

<table>
<thead>
<tr>
<th>Method</th>
<th>( \alpha )</th>
<th>( \beta )</th>
<th>( \gamma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>( \frac{1}{L} )</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HBM</td>
<td>( \frac{4}{(\sqrt{L} + \sqrt{\mu})^2} )</td>
<td>( \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^2 )</td>
<td>0</td>
</tr>
<tr>
<td>FGM</td>
<td>( \frac{1}{L} )</td>
<td>( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} )</td>
<td>( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} )</td>
</tr>
<tr>
<td>TMM</td>
<td>( \frac{2\sqrt{L} - \sqrt{\mu}}{L\sqrt{L}} )</td>
<td>( \frac{(\sqrt{\kappa} - 1)^2}{\kappa + \sqrt{\kappa}} )</td>
<td>( \frac{(\sqrt{\kappa} - 1)^2}{2\kappa + \sqrt{\kappa} - 1} )</td>
</tr>
</tbody>
</table>
**GM with linesearch**

\[ \alpha = \arg\min_\alpha f(x_k - \alpha \nabla f(x_k)) \]

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

**HBM with linesearch**

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

\[ x_{k+1} = x_k + \beta (x_k - x_{k-1}) - \alpha \nabla f(x_k) \]
FGM (Nesterov, 1983) with restarts every $N$ iterations

$$x_{k+1} = y_k - \frac{1}{L} \nabla f(y_k)$$

$$y_{k+1} = x_{k+1} + \beta_{k+1} (x_{k+1} - x_k)$$

Upper bound (O’Donoghue and Candès, 2015)

$$\rho(N^*) \leq \exp\left(\frac{-1}{e \sqrt{8 \kappa}}\right)$$

Condition ratio $\kappa$

Evaluations to converge

$N = 1$
$N = 5$
$N = 10$
$N = 20$
$\min N$

Upper bound
Summary

**Automatic** analysis of optimization methods by solving a small-sized semidefinite program. Convergence rate is *provably tight* when

- $f$ is $L$-smooth and $\mu$-strongly convex, and
- the method is *iterative* with *fixed stepsizes*.

Same methodology may be used (possibly without tightness) to analyze:
- line/subspace search
- restart (fixed/adaptive)
- other function classes (besides smooth strongly convex)

Code

https://github.com/QCGroup/quad-lyap-first-order