Automated Lyapunov analysis of primal-dual optimization algorithms

An interpolation approach

Bryan Van Scoy
Miami University

John W. Simpson-Porco
University of Toronto

Laurent Lessard
Northeastern University

Optimization

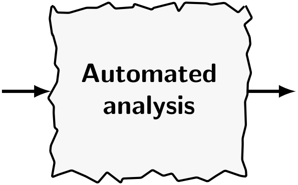
$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X \end{array}$$

- **Black-box setting:** can only obtain information by sampling oracles that return information about the objective/constraint at a point
- **Performance measure:** a measure of distance from the optimal solution (e.g., $f(x) - f_\star$ or $\|\nabla f(x)\|^2$)
- **Iteration complexity:** number of iterations to compute a solution such that the performance measure is less than some tolerance
- **Worst-case analysis:** bound the worst-case iteration complexity over a class of problems

Systematic algorithm analysis

Problem specifications

- function class
- oracle
- algorithm
- performance measure

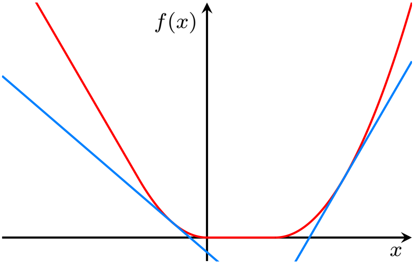


Worst-case performance bound

(Drori and Teboulle, 2014), (Lessard et al., 2016), (Taylor et al., 2017), . . .

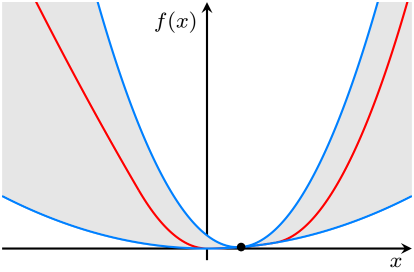
Function classes

A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if $f(y) \geq f(x) + \nabla f(x)^\mathsf{T}(y - x)$.



Convex functions have no local minimizers.

- A function is m-strongly convex if $f(x) - \frac{m}{2}\|x\|^2$ is convex.
- A function is L-smooth if $\frac{L}{2}\|x\|^2 - f(x)$ is convex.



The condition ratio L/m characterizes the variation in curvature.

At each point, the function is bounded by quadratics of curvature m and L.

Linearly constrained convex optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

Assumptions

- f is L-smooth and m-strongly convex, denoted $f \in \mathcal{F}(m, L)$
- A has full row rank and singular values in $[\underline{\sigma}, \overline{\sigma}]$, denoted $A \in \mathcal{A}(\underline{\sigma}, \overline{\sigma})$

Condition ratios $\kappa(f) = \frac{L}{m}$ and $\kappa(A) = \frac{\overline{\sigma}}{\underline{\sigma}}$ characterize problem difficulty.

Algorithms

- Projected gradient descent

$$x_{k+1} = \mathrm{proj}_X(x_k - \alpha_k \nabla f(x_k))$$

- Dual ascent

$$\begin{aligned} x_{k+1} &\in \arg\min_x \; f(x) + \lambda_k^\mathsf{T}(Ax - b) \\ \lambda_{k+1} &= \lambda_k + \alpha_k(Ax_{k+1} - b) \end{aligned}$$

- Method of multipliers

$$\begin{aligned} x_{k+1} &\in \arg\min_x \; f(x) + \lambda_k^\mathsf{T}(Ax - b) + \tfrac{\mu}{2}\|Ax - b\|^2 \\ \lambda_{k+1} &= \lambda_k + \mu(Ax_{k+1} - b) \end{aligned}$$

Projection and minimization oracles are computationally expensive.

(Boyd et al., 2010)

Primal-dual algorithms

A pair $(x_\star, \lambda_\star)$ is optimal if and only if it is a saddle point of the Lagrangian

$$L(x, \lambda) = f(x) + \lambda^\mathsf{T}(Ax - b)$$

Primal descent

$$\begin{aligned} x^+ &= x - \alpha_x \, \nabla_x L(x, \lambda) \\ &= x - \alpha_x(\nabla f(x) + A^\mathsf{T}\lambda) \end{aligned}$$

Dual ascent

$$\begin{aligned} \lambda^+ &= \lambda + \alpha_\lambda \, \nabla_\lambda L(x, \lambda) \\ &= \lambda + \alpha_\lambda(Ax - b) \end{aligned}$$

Only requires computing ∇f and multiplication by A and A^T.

Generalizations

- Apply the primal-dual algorithm to the augmented problem

$$\begin{array}{ll} \text{minimize} & f(x) + \frac{\mu}{2}\|Ax - b\|^2 \\ \text{subject to} & Ax = b \end{array}$$

- Apply dual ascent to an extrapolated point

$$\lambda^+ = \lambda + \alpha_\lambda \nabla_\lambda L\big(x + \gamma\,(x^+ - x), \lambda\big)$$

$$x^+ = x - \alpha_x \left[\nabla f(x) + A^\mathsf{T}\lambda + \mu A^\mathsf{T}(Ax - b)\right]$$
$$\lambda^+ = \lambda + \alpha_\lambda \left[A\,(x + \gamma\,(x^+ - x)) - b\right]$$

More parameters so (potentially) better convergence.

Literature review

With no Lagrangian augmentation ($\mu = 0$) and no extrapolation ($\gamma = 0$), the Lyapunov function

$$V(x, \lambda) = \|x - \nabla f^*(-A^\mathsf{T}\lambda)\| + c\,\|\lambda - \lambda_\star\|$$

where f^* is the convex conjugate of f converges with rate

$$V(x_k, \lambda_k) = \mathcal{O}\left(\left(1 - \frac{1}{12\kappa(f)^3\,\kappa(A)^4}\right)^k\right)$$

(Du and Hu, 2019)

Literature review

With extrapolation $\gamma = 1$ and no Lagrangian augmentation (for simplicity), the Lyapunov function

$$V(x, \lambda) = \sqrt{(1 - \alpha_x \alpha_\lambda \overline{\sigma}^2)\|x - x_\star\|^2 + \|\lambda - \lambda_\star\|^2}$$

converges with rate

$$V(x_k, \lambda_k) = \mathcal{O}(\rho^k) \quad \text{where} \quad \rho^2 = \max\{1 - \alpha_x m\,(1 - \alpha_x L),\ 1 - \alpha_x \alpha_\lambda \underline{\sigma}^2\}$$

Choosing the stepsizes to optimize this bound yields

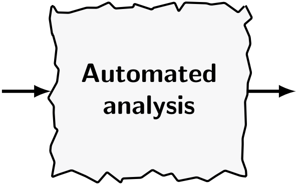
$$\rho^2 = \begin{cases} 1 - \frac{1}{4\,\kappa(f)} & \text{if } \kappa(A) \leq \sqrt{2} \\ 1 - \frac{1}{\kappa(f)}\left(\frac{1}{\kappa(A)^2} - \frac{1}{\kappa(A)^4}\right) & \text{otherwise} \end{cases}$$

(Alghunaim and Sayed, 2020)

Systematic analysis of primal-dual algorithms

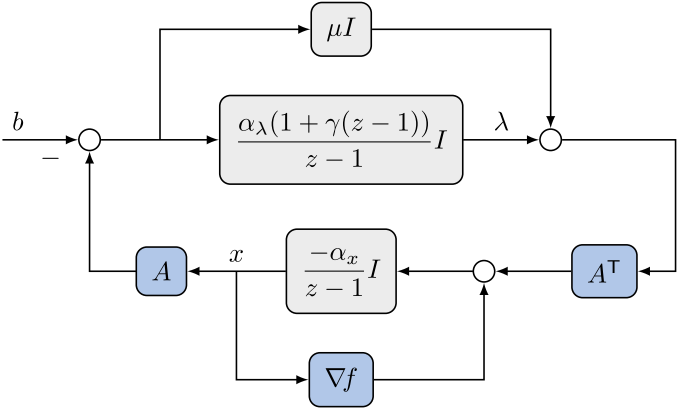
Problem specifications

- function class
- oracle
- algorithm
- performance measure



Worst-case performance bound

$$x^+ = x - \alpha_x \left[\nabla f(x) + A^\mathsf{T}\lambda + \mu A^\mathsf{T}(Ax - b)\right]$$
$$\lambda^+ = \lambda + \alpha_\lambda \left[A\left(x + \gamma\left(x^+ - x\right)\right) - b\right]$$



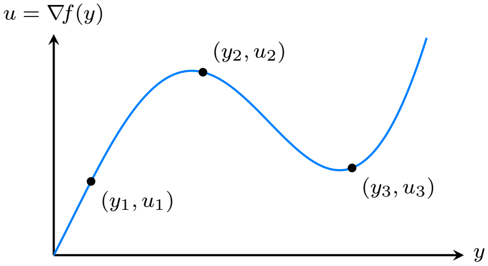
The algorithm is an LTI system in feedback with ∇f, A, and A^T.

Main idea

- Replace these components with constraints on their inputs/outputs
- Use these constraints to search for a Lyapunov function

Interpolation

When does there exist $f \in \mathcal{F}(m, L)$ such that $u_i = \nabla f(y_i)$ for all i?



$$0 \leq f(y_i) - f(y_j) - \nabla f(y_j)^\mathsf{T}(y_i - y_j) - \frac{1}{2(1-\frac{m}{L})}\Big(\frac{1}{L}\|\nabla f(y_i) - \nabla f(y_j)\|^2 + m\|y_i - y_j\|^2 - 2\frac{m}{L}(\nabla f(y_i) - \nabla f(y_j))^\mathsf{T}(y_i - y_j)\Big)$$

(Taylor et al., 2017)

Quadratic constraints

For what matrices M does the quadratic inequality

$$0 \leq \operatorname{tr}\left(M \begin{bmatrix} Y \\ U \end{bmatrix}^\mathsf{T} \begin{bmatrix} Y \\ U \end{bmatrix}\right) \qquad \text{with} \qquad \begin{aligned} Y &= \begin{bmatrix} y_1 & \dots & y_n \end{bmatrix} \\ U &= \begin{bmatrix} u_1 & \dots & u_n \end{bmatrix} \end{aligned}$$

hold for iterates such that $u_i = \nabla f(y_i)$ for some $f \in \mathcal{F}(m, L)$?

Result:

$$M = \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \otimes R + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \otimes S$$

where R is diagonally dominant with zero excess and S is skew symmetric

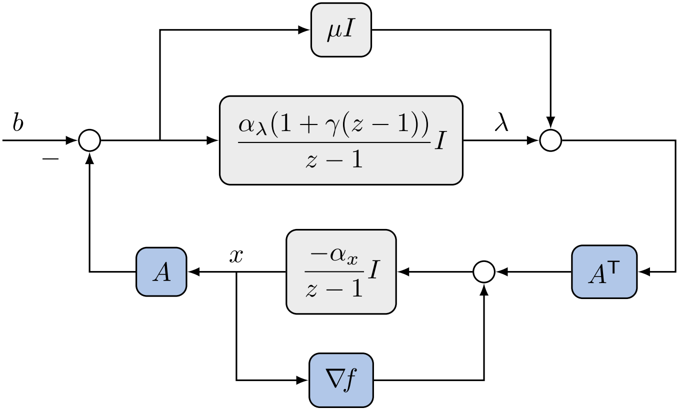
Proof (idea): take nonnegative sum of interpolation conditions such that function values cancel

R is symmetric, $R\mathbf{1} = 0$, $R_{ii} \geq 0$ for all i, and $R_{ij} \leq 0$ for all $i \neq j$

Issue: no (convex) interpolation conditions for $A \in \mathcal{A}(\underline{\sigma}, \overline{\sigma})$

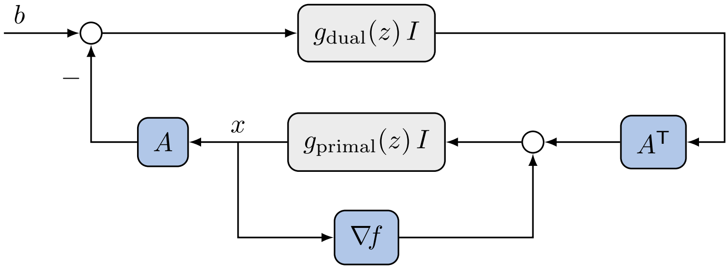
- interpolation for linear operators in (Bousselmi et al., 2023)
- can bound maximum singular value, but not minimum

Fix: transform the uncertainty to a form for which we have interpolation conditions



First, let's simplify the block diagram. Define

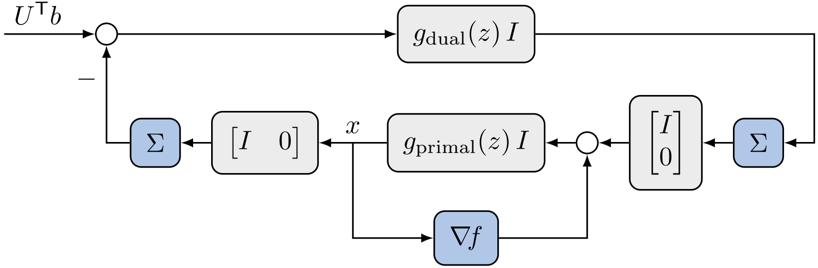
$$g_{\text{primal}}(z) = \frac{-\alpha_x}{z-1} \qquad g_{\text{dual}}(z) = -\mu - \frac{\alpha_\lambda(1+\gamma(z-1))}{z-1}$$



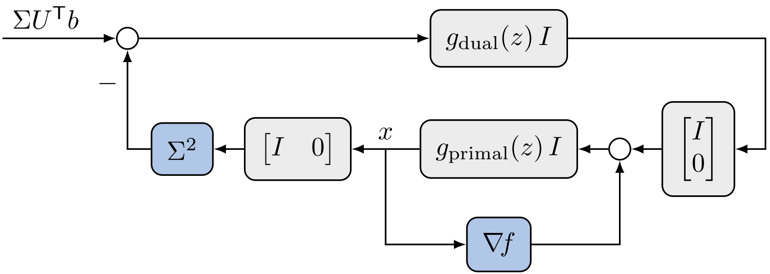
Compact SVD (A is full row rank by assumption)

$$A = U \begin{bmatrix} \Sigma & 0 \end{bmatrix} \begin{bmatrix} V_1^\mathsf{T} \\ V_2^\mathsf{T} \end{bmatrix}$$

Can move A^T around top loop to form $A^\mathsf{T} A$ (singular), but not AA^T (nonsingular).



The objective function $f(x)$ transforms to $f(Vx)$.



The algorithm is an LTI system in feedback with ∇f and Σ^2.

Main idea

- Replace these components with constraints on their inputs/outputs
- Use these constraints to search for a Lyapunov function

Linear interpolation

Interpolation: There exists $A \in \mathcal{A}(\underline{\sigma}, \overline{\sigma})$ such that $u_i = \Sigma^2 y_i$ for all i iff

$$U^\mathsf{T} Y = Y^\mathsf{T} U \qquad \text{and} \qquad (U - \underline{\sigma}^2 Y)^\mathsf{T} (\overline{\sigma}^2 Y - U) \succeq 0$$

Quadratic constraints: The quadratic inequality

$$0 \leq \operatorname{tr}\left(M \begin{bmatrix} Y \\ U \end{bmatrix}^\mathsf{T} \begin{bmatrix} Y \\ U \end{bmatrix} \right)$$

holds for iterates such that $u_i = \Sigma^2 y_i$ for some $A \in \mathcal{A}(\underline{\sigma}, \overline{\sigma})$ iff

$$M = \begin{bmatrix} -2\underline{\sigma}^2\overline{\sigma}^2 & \overline{\sigma}^2 + \underline{\sigma}^2 \\ \overline{\sigma}^2 + \underline{\sigma}^2 & -2 \end{bmatrix} \otimes R + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \otimes S$$

where R is symmetric positive semidefinite and S is skew symmetric.

(Bousselmi et al., 2023)

Analysis via Lyapunov functions

$$x_{k+1} = \phi(x_k)$$

A function $V(x)$ is a Lypapunov function if the following holds for all x:

$$V(x) \geq \|x\|^2 \qquad \text{(positivity condition)}$$
$$V(\phi(x)) \leq \rho^2 V(x) \qquad \text{(decrease condition)}$$

If there exists a Lyapunov function, then $\|x_k\| = O(\rho^k)$.

Proof: $\|x_k\|^2 \leq V(x_k) \leq \rho^2 V(x_{k-1}) \leq \cdots \leq \rho^{2k} V(x_0)$

Lyapunov functions are also known as *potential* or *energy* functions.

$$x_{k+1} = Ax_k + Bu_k \quad \text{subject to} \quad \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\mathsf{T} M_i \begin{bmatrix} x_k \\ u_k \end{bmatrix} \ge 0$$

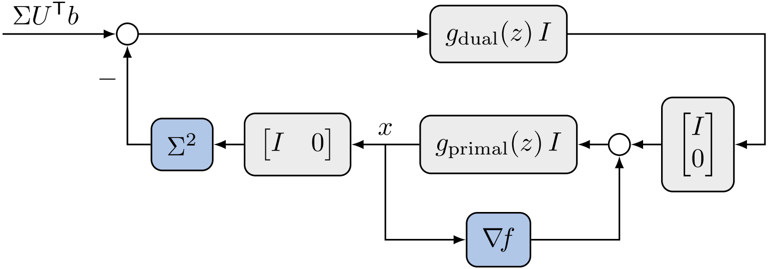
If there exist symmetric matrix P and scalars $\lambda_i \ge 0$ and $\mu_i \ge 0$ such that

$$\begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^\mathsf{T} \begin{bmatrix} P & 0 \\ 0 & -\rho^2 P \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} + \sum_i \lambda_i \, M_i \preceq 0 \qquad \text{(decrease)}$$

$$\begin{bmatrix} I & 0 \end{bmatrix}^\mathsf{T} (I - P) \begin{bmatrix} I & 0 \end{bmatrix} + \sum_i \mu_i \, M_i \preceq 0 \qquad \text{(positivity)}$$

then $V(x) = x^\mathsf{T} P x$ is a Lyapunov function.

For LTI systems subject to quadratic constraints, searching for a quadratic Lyapunov function is a semidefinite program.



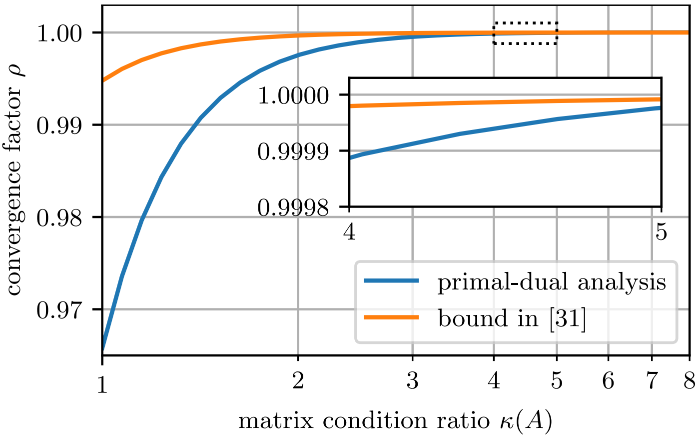
To apply more constraints, filter the inputs and outputs of ∇f and Σ^2 by

$$\psi(z) = \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-\ell} \end{bmatrix}$$

The Lyapunov function then depends on the state and ℓ previous inputs and outputs of both ∇f and Σ^2.

Comparison with known bounds

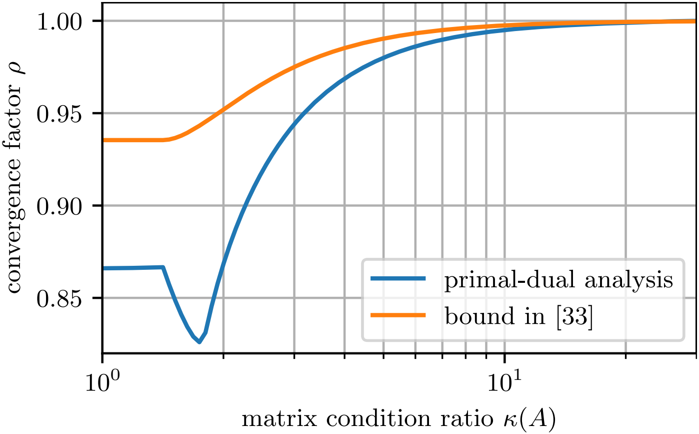
Parameters: $\kappa(f) = 2$, $\mu = 0$, $\gamma = 0$



(Du and Hu, 2019)

Comparison with known bounds

Parameters: $\kappa(f) = 2$, $\mu = 0$, $\gamma = 1$



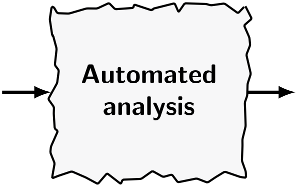
(Alghunaim and Sayed, 2020)

Summary

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

Problem specifications

- function class
- oracle
- algorithm
- performance measure



Worst-case performance bound

$$x^+ = x - \alpha_x \left[\nabla f(x) + A^\mathsf{T}\lambda + \mu A^\mathsf{T}(Ax - b)\right]$$
$$\lambda^+ = \lambda + \alpha_\lambda \left[A\left(x + \gamma\left(x^+ - x\right)\right) - b\right]$$