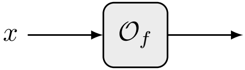
Systematic Analysis of Iterative Black-Box Optimization Algorithms using Control

Bryan Van Scoy
Miami University

Black-box optimization

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

Can only obtain information by sampling oracles.



Oracles: function value, gradient, Hessian, coordinate derivative, proximal operator, projection, noisy (stochastic or adversarial)

Algorithm analysis

Iteration complexity: number of iterations such that

$$\text{performance measure} \quad \leq \quad \text{tolerance}$$

Performance measures

- distance from optimizer: $\|x_k - x_\star\|$ where $x_\star$ is an optimizer
- optimality gap: $f(x_k) - f_\star$ where $f_\star$ is the optimal value
- distance from stationary point: $\|\nabla f(x_k)\|$

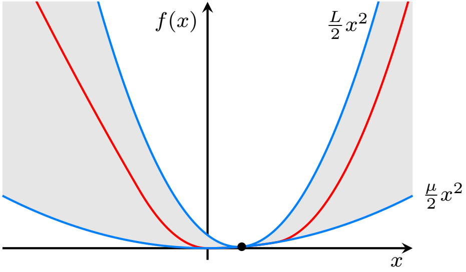
Worst-case algorithm analysis

Bound the worst-case iteration complexity over all problem instances in some class.

- **Function classes:** linear, quadratic, smooth, (strongly) convex, quadratically upper bounded, Lipschitz continuous, convex indicator, convex support functions, restricted secant inequality, error bound
- **Constraint classes:** convex, cone, polytope, half-plane, affine space

Example: Smooth strongly convex functions

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



The condition ratio $\kappa = L/\mu$ characterizes the variation in curvature.

Example: Algorithm analysis

$$\text{minimize} \quad f(x)$$

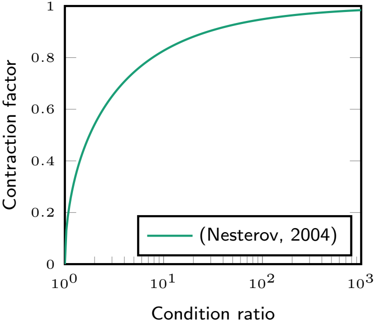
Problem specification

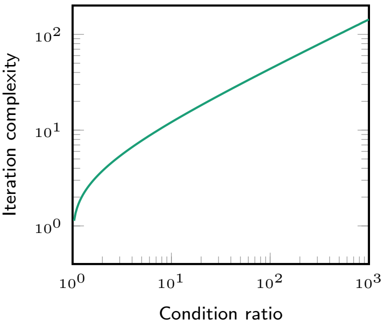
- Function class: f is L-smooth and μ-strongly convex
- Oracle: gradient $\nabla f(x)$
- Algorithm: fast gradient method

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

- Performance measure: $f(x_k) - f_\star$

Performance bound





$$f(x_k) - f_\star \leq c\left(1 - \sqrt{\frac{\mu}{L}}\right)^k$$

Condition ratio: $\kappa = \frac{L}{\mu}$, contraction factor: $\rho^2 = 1 - \sqrt{\frac{\mu}{L}}$, iteration complexity: $-\frac{1}{\log \rho}$

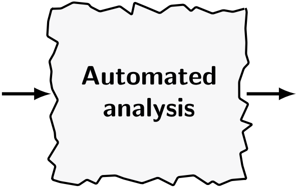
Traditional algorithm analysis

- requires expert knowledge and insights
- performed on a case-by-case basis
- bounds may not be tight

Systematic algorithm analysis

Problem specifications

- function class
- oracle
- algorithm
- performance measure



Worst-case performance bound

Main ideas

- interpret optimization algorithms as dynamical systems
- use tools from robust control to study convergence properties

Literature

Optimization

- performance estimation problem (PEP)
- searching for worst-case problem instance is an optimization problem
- originally formulated in (Drori and Teboulle, 2014)
- tight bounds using interpolation in (Taylor, Hendrickx, Glineur, 2017)

Controls

- integral quadratic constraints (IQCs)
- tools for robust control (Megretski and Rantzer, 1997)
- algorithms are dynamical systems (Lessard, Packard, Recht, 2016)
- worst-case analysis using robust control

Outline

Preliminaries

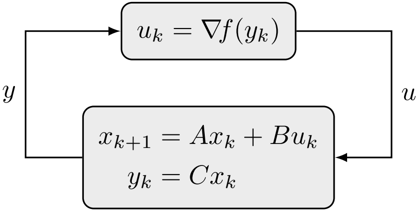
- iterative algorithms as dynamical systems
- interpolation
- worst-case performance analysis via Lyapunov functions

Case studies

- consensus optimization
- sensitivity to gradient noise

Iterative algorithms as dynamical systems

$$\xi_{k+1} = \xi_k - \alpha \nabla f\big(\xi_k + \eta(\xi_k - \xi_{k-1})\big) + \beta(\xi_k - \xi_{k-1})$$



fixed points

$\updownarrow$

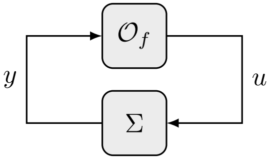
optimizers

$$x_k = \begin{bmatrix} \xi_k \\ \xi_{k-1} \end{bmatrix} \qquad A = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} -\alpha \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 1+\eta & -\eta \end{bmatrix}$$

Special cases: gradient descent, Nesterov or Polyak acceleration

(Lessard, Packard, Recht, 2016)

Worst-case performance analysis



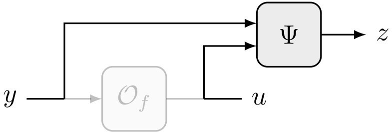
- $\Sigma = (A, B, C)$ is the system
- $\mathcal{O}_f$ is the oracle applied to a function f in a function class $\mathcal{F}$

Bound the worst-case performance over the function class $\mathcal{F}$.

Main ideas

- Replace the oracle with constraints on its (filtered) input and output.
- Use the constraints to search for a Lyapunov function.

Filter



- Choose Ψ as the linear time-invariant system with transfer function

$$\Psi(z) = \begin{bmatrix} \psi(z) & 0 \\ 0 & \psi(z) \end{bmatrix} \qquad \text{where} \qquad \psi(z) = (1, z^{-1}, \dots, z^{-\ell})$$

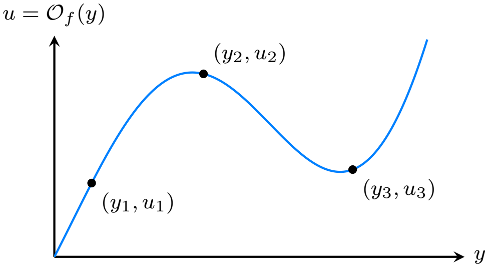
- ℓ trades off tightness and computational efficiency of the analysis

$$z_k = (\underbrace{y_k, y_{k-1}, \dots, y_{k-\ell}}_{\text{past } \ell \text{ inputs}}, \underbrace{u_k, u_{k-1}, \dots, u_{k-\ell}}_{\text{past } \ell \text{ outputs}})$$

(Van Scoy and Lessard, 2022)

Interpolation

When does there exist $f \in \mathcal{F}$ such that $u_k = \mathcal{O}_f(y_k)$ for all k?



First-order oracle:

$$u_k = \mathcal{O}_f(y_k) = (f_k, g_k) \quad \text{with} \quad f_k = f(y_k) \quad \text{and} \quad g_k = \nabla f(y_k)$$

Interpolation is also known as *function extension*.

Convex interpolation

$$f(y) \geq f(x) + \nabla f(x)^{\mathsf{T}}(y - x) \quad \text{for all } x, y \in \mathbb{R}^d$$

The conditions for interpolation are the discretization

$$f_i \geq f_j + g_j^{\mathsf{T}}(y_i - y_j) \quad \text{for all } i, j$$

If these conditions hold, then an interpolating convex function is

$$f(y) = \max_k \left\{ f_k + g_k^{\mathsf{T}}(y - y_k) \right\}$$

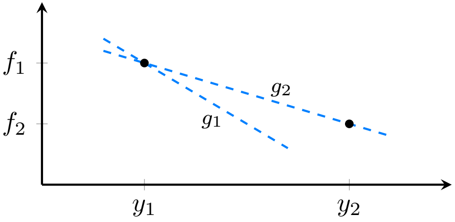
(Taylor, Hendrickx, Glineur, 2017)

Smooth convex interpolation

- Convex: $f(y) \geq f(x) + \nabla f(x)^{\mathsf{T}}(y - x)$
- Lipschitz gradient: $\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$

Naive discretization does not yield interpolation conditions.

Counterexample



$$(y_1, f_1, g_1) = (1, 2, -2)$$
$$(y_2, f_2, g_2) = (2, 1, -1)$$

Smooth strongly convex interpolation

A function is L-smooth and μ-strongly convex iff, for all $x, y \in \mathbb{R}^d$,

$$0 \leq f(y) - f(x) - \nabla f(x)^\mathsf{T}(y-x) - \frac{1}{2(1-\frac{\mu}{L})}\bigg(\frac{1}{L}\|\nabla f(y) - \nabla f(x)\|^2 + \mu\|y-x\|^2 - 2\frac{\mu}{L}(\nabla f(x) - \nabla f(y))^\mathsf{T}(x-y)\bigg)$$

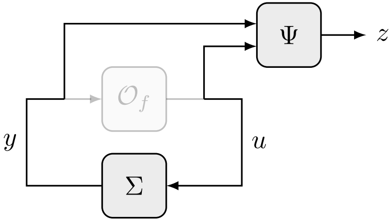
Discretizing this inequality yields interpolation conditions.

Special cases

- convex: $\mu = 0$ and $L = +\infty$
- smooth and convex: $\mu = 0$ and L finite

(Taylor, Hendrickx, Glineur, 2017)

Algorithm analysis



- the output of the filter is the past ℓ inputs and outputs of the oracle
- the constraints on z_k are the interpolation conditions for the oracle
- for first-order oracles, these are typically linear–quadratic constraints

$$\left\langle \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{g}_k \end{bmatrix}, M_i \begin{bmatrix} \boldsymbol{y}_k \\ \boldsymbol{g}_k \end{bmatrix} \right\rangle + \langle m_i, \boldsymbol{f}_k \rangle \geq 0$$

- search for a Lyapunov function of the same form

$$V(\boldsymbol{x}, \boldsymbol{f}) = \langle \boldsymbol{x}, P\boldsymbol{x} \rangle + \langle p, \boldsymbol{f} \rangle$$

Bold quantities consist of the past ℓ iterates.

$V(\boldsymbol{x}, \boldsymbol{f})$ is a Lyapunov function iff there exist $\lambda_i \geq 0$ and $\mu_i \geq 0$ such that

- **Decrease condition**

$$V(\boldsymbol{x}_{k+1}, \boldsymbol{f}_{k+1}) - \rho^2 V(\boldsymbol{x}_k, \boldsymbol{f}_k) + \sum_i \lambda_i \, (\text{constraint}_i) \leq 0$$

- **Positivity condition**

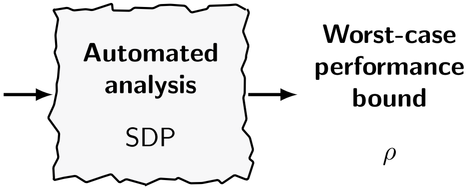
$$(\text{performance measure}) - V(\boldsymbol{x}_k, \boldsymbol{f}_k) + \sum_i \mu_i \, (\text{constraint}_i) \leq 0$$

Searching for a linear–quadratic Lyapunov function is a semidefinite program.

(Van Scoy, Taylor, Lessard, 2018)

Problem specifications

- function class (interpolation)
- oracle (first-order)
- algorithm (A, B, C)
- performance $\|x_k - x_\star\|$



$$\|x_k - x_\star\| = O(\rho^k)$$

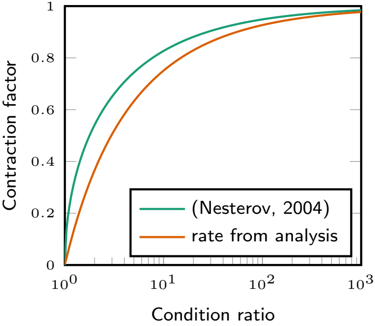
Efficiency

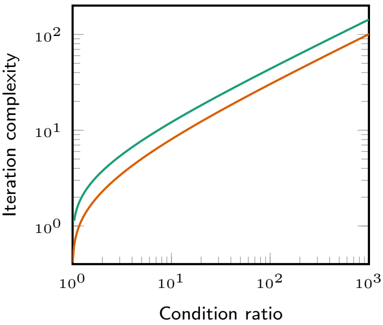
- Size of the SDP does **not** depend on dimension of the domain of f.
- Size scales with ℓ, but $\ell > 2$ does not appear to improve the bound.
- To obtain the best bound, perform bisection over ρ.

The automated analysis involves solving a semidefinite program that can be done in fractions of a second.

Function class: L-smooth and μ-strongly convex

Algorithm: fast gradient method





Algorithm design

$$\begin{array}{ll} \text{minimize} & \rho \\ \text{subject to} & \mathrm{SDP}(\rho, A, B, C) \end{array}$$

Challenges

- The problem is not jointly convex in ρ.
- In principle, solution is a **semialgebraic set**.
  - matrix inequalities are equivalent to sets of polynomial inequalities (principle minors)
  - optimal solution is characterized by the active constraints
- This polynomial system is not always solvable analytically.

Find algorithms with simple algebraic expressions
(avoid numeric solutions) that are close to optimal.

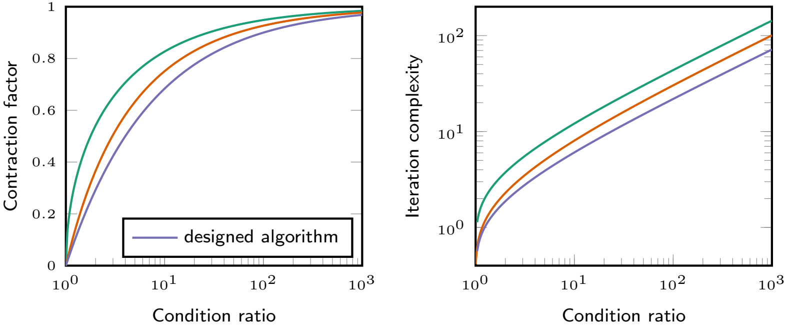
General strategy

- Fix function class parameters (e.g., μ and L).
- Numerically find locally optimal algorithm parameters.
- Write SDP as polynomial optimization problem.
- Use numerical solution to find active constraints.
- Look for analytic solution to system of active constraints.

Function class: L-smooth and μ-strongly convex

Algorithm: triple momentum (TM) with $\rho = 1 - \sqrt{\mu/L}$

$$x_{k+1} = x_k + \frac{\rho^2}{2-\rho}(x_k - x_{k-1}) - \frac{1+\rho}{L}\nabla f\Big(x_k + \frac{\rho^2}{(1+\rho)(2-\rho)}(x_k - x_{k-1})\Big)$$



The designed algorithm has the optimal rate for this function class.

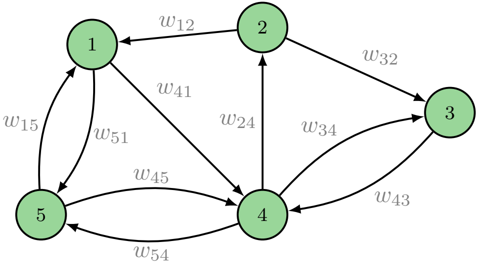
(Van Scoy, Freeman, Lynch, 2017) and (Drori and Taylor, 2022)

Case study

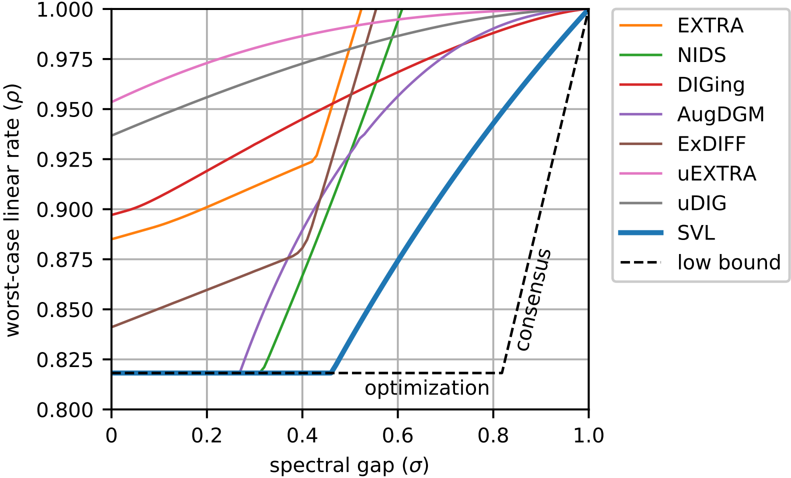
Consensus optimization

Consensus optimization

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{n} f_i(x_i) \\ \text{subject to} & x_1 = x_2 = \ldots = x_n \end{array}$$



Want each agent to compute the global optimizer by communicating with local neighbors and performing local computations.



(Sundararjan, Van Scoy, Lessard 2020)

Case study

Sensitivity to gradient noise

Sensitivity to gradient noise

$$\text{minimize} \quad f(x)$$

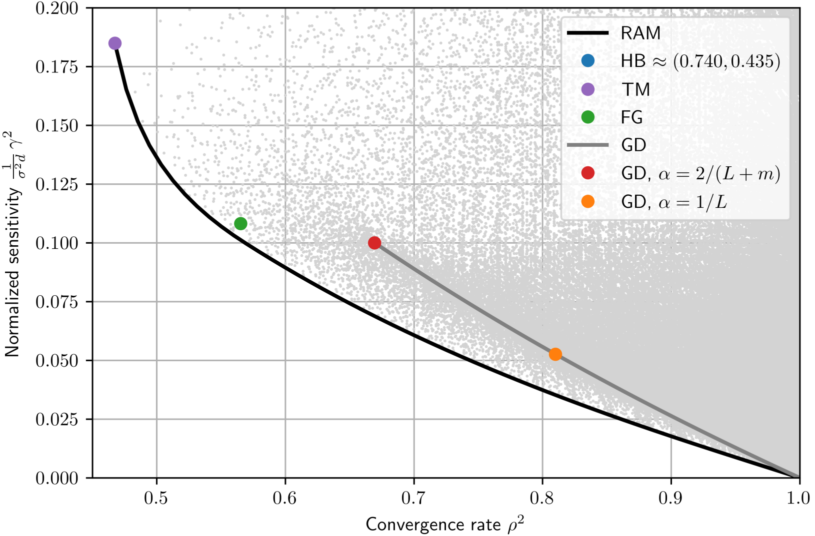
Noisy oracle: $\mathcal{O}_f(x) = \nabla f(x) + w$

- w is zero-mean and independent across queries
- $\mathbb{E}\, ww^\mathsf{T} \preceq \sigma^2 I_d$ for some known σ

Use cases

- perturb gradient for privacy
- gradient only available through noisy measurements
- risk minimization; minimize expected loss over population distribution

Robust Accelerated Method (RAM)

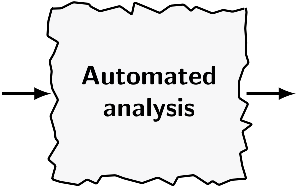


(Van Scoy and Lessard, 2021)

Summary

Problem specifications

- function class
- oracle
- algorithm
- performance measure



Worst-case performance bound

`vanscoy.github.io`