Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints

Based on papers by Lessard, Packard, Recht, Nishihara, Jordan

Matthew Staib

Massachusetts Institute of Technology

OPTML++, November 30, 2015

Table of Contents

Context

- Balance robustness, accuracy, speed

Context

- Balance robustness, accuracy, speed
- Current: analyze methods algorithm-by-algorithm

Context

- Balance robustness, accuracy, speed
- Current: analyze methods algorithm-by-algorithm
- Reliance on optimization experts for proofs

Main idea

- Frame first-order methods as dynamical systems

Main idea

- Frame first-order methods as dynamical systems
- Replace nonlinear parts with integral quadratic constraints (IQCs)

Main idea

- Frame first-order methods as dynamical systems
- Replace nonlinear parts with integral quadratic constraints (IQCs)
- Prove a linear convergence rate by solving a small SDP

Main idea

- Frame first-order methods as dynamical systems
- Replace nonlinear parts with integral quadratic constraints (IQCs)
- Prove a linear convergence rate by solving a small SDP
- Optimize over algorithm parameters for convergence rate

Main idea

- Frame first-order methods as dynamical systems
- Replace nonlinear parts with integral quadratic constraints (IQCs)
- Prove a linear convergence rate by solving a small SDP
- Optimize over algorithm parameters for convergence rate
  - Subject to strong convexity and Lipschitz properties

Main idea

- Frame first-order methods as dynamical systems
- Replace nonlinear parts with integral quadratic constraints (IQCs)
- Prove a linear convergence rate by solving a small SDP
- Optimize over algorithm parameters for convergence rate
  - Subject to strong convexity and Lipschitz properties
  - Subject to extent of noise

Table of Contents

Linear dynamical systems

$$\xi_{k+1} = A\xi_k + Bu_k \tag{1}$$
$$y_k = C\xi_k + Du_k \tag{2}$$

$(u_k, y_k, \xi_k) =$ input, output, state

Linear dynamical systems (with nonlinear feedback)

$$\xi_{k+1} = A\xi_k + Bu_k \tag{3}$$
$$y_k = C\xi_k + Du_k \tag{4}$$
$$u_k = \Delta(y_k) \tag{5}$$

$(u_k, y_k, \xi_k) =$ input, output, state
$\Delta =$ (nonlinear) map

Linear dynamical systems (for first order methods)

$$\xi_{k+1} = A\xi_k + Bu_k \tag{6}$$
$$y_k = C\xi_k + Du_k \tag{7}$$
$$u_k = \Delta(y_k) \tag{8}$$

$(u_k, y_k, \xi_k) =$ input, output, state
$\Delta(z) = \nabla f(z)$

Gradient descent

- Start with gradient descent update:

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

Gradient descent

- Start with gradient descent update:

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

- Expand to input, output, state:

$$\begin{aligned} \xi_{k+1} &= \xi_k - \alpha u_k \\ y_k &= \xi_k \\ u_k &= \nabla f(y_k) \end{aligned}$$

Gradient descent

- Start with gradient descent update:

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

- Expand to input, output, state:

$$\begin{aligned} \xi_{k+1} &= \xi_k - \alpha u_k \\ y_k &= \xi_k \\ u_k &= \nabla f(y_k) \end{aligned}$$

- Block form: $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} I_d & -\alpha I_d \\ \hline I_d & 0_d \end{array}\right]$

Nesterov's method

- Start with update:

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

Nesterov's method

- Start with update:

$$\begin{aligned} x_{k+1} &= y_k - \alpha_k \nabla f(y_k) \\ y_k &= (1+\beta)x_k - \beta x_{k-1} \end{aligned}$$

- Expand to input, output, state:

$$\begin{aligned} \xi_{k+1}^{(1)} &= (1+\beta)\xi_k^{(1)} - \beta\xi_k^{(2)} - \alpha u_k \\ \xi_{k+1}^{(2)} &= \xi_k^{(1)} \\ y_k &= (1+\beta)\xi_k^{(1)} - \beta\xi_k^{(2)} \\ u_k &= \nabla f(y_k) \end{aligned}$$

Nesterov's method

- Start with update:

$$\begin{aligned} x_{k+1} &= y_k - \alpha_k \nabla f(y_k) \\ y_k &= (1+\beta)x_k - \beta x_{k-1} \end{aligned}$$

- Expand to input, output, state:

$$\begin{aligned} \xi^{(1)}_{k+1} &= (1+\beta)\xi^{(1)}_k - \beta\xi^{(2)}_k - \alpha u_k \\ \xi^{(2)}_{k+1} &= \xi^{(1)}_k \\ y_k &= (1+\beta)\xi^{(1)}_k - \beta\xi^{(2)}_k \\ u_k &= \nabla f(y_k) \end{aligned}$$

- Block form: $\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} (1+\beta)I_d & -\beta I_d & -\alpha I_d \\ I_d & 0_d & 0_d \\ \hline (1+\beta)I_d & -\beta I_d & 0_d \end{array}\right]$

Necessary conditions for convergence

- For convex problems, we need $u_\star = \nabla f(y_\star) = 0$

Necessary conditions for convergence

- For convex problems, we need $u_\star = \nabla f(y_\star) = 0$
- Plug this into update rule: $\xi_\star = A\xi_\star,\ y_\star = C\xi_\star$

Quadratic case

- Suppose $f(y) = \frac{1}{2}y^T Qy - p^T y + r$ with $mI_d \preceq Q \preceq LI_d$.

Quadratic case

- Suppose $f(y) = \frac{1}{2}y^T Qy - p^T y + r$ with $mI_d \preceq Q \preceq LI_d$.
- Then $\nabla f(y) = Qy - p = Q(y - y_\star)$

Quadratic case

- Suppose $f(y) = \frac{1}{2}y^T Qy - p^T y + r$ with $mI_d \preceq Q \preceq LI_d$.
- Then $\nabla f(y) = Qy - p = Q(y - y_\star)$
- $y_k = C\xi_k$, so $u_k = QC(\xi_k - \xi_\star)$

Quadratic case

- Suppose $f(y) = \frac{1}{2}y^T Qy - p^T y + r$ with $mI_d \preceq Q \preceq LI_d$.
- Then $\nabla f(y) = Qy - p = Q(y - y_\star)$
- $y_k = C\xi_k$, so $u_k = QC(\xi_k - \xi_\star)$
- From state update: $\xi_{k+1} - \xi_\star = (A + BQC)(\xi_k - \xi_\star)$

Quadratic case

- Suppose $f(y) = \frac{1}{2} y^T Q y - p^T y + r$ with $mI_d \preceq Q \preceq LI_d$.
- Then $\nabla f(y) = Qy - p = Q(y - y_\star)$
- $y_k = C\xi_k$, so $u_k = QC(\xi_k - \xi_\star)$
- From state update: $\xi_{k+1} - \xi_\star = (A + BQC)(\xi_k - \xi_\star)$
- Hence the spectral radius $\rho(T)$ of $T := A + BQC$ determines convergence rate

Quadratic case

- Suppose $f(y) = \frac{1}{2}y^T Qy - p^T y + r$ with $mI_d \preceq Q \preceq LI_d$.
- Then $\nabla f(y) = Qy - p = Q(y - y_\star)$
- $y_k = C\xi_k$, so $u_k = QC(\xi_k - \xi_\star)$
- From state update: $\xi_{k+1} - \xi_\star = (A + BQC)(\xi_k - \xi_\star)$
- Hence the spectral radius $\rho(T)$ of $T := A + BQC$ determines convergence rate
- Using given properties of Q, we can analytically tune the parameters and determine rate ρ for e.g. gradient descent

An alternative approach

Theorem

The spectral radius $\rho(T) < \rho$ *if and only if there exists* $P \succeq 0$ *such that* $T^T P T - \rho^2 P \preceq 0$.

- If $\xi_{k+1} - \xi_\star = T(\xi_k - \xi_\star)$ then

$$(\xi_{k+1} - \xi_\star)^T P(\xi_{k+1} - \xi_\star) < \rho^2(\xi_k - \xi_\star)^T P(\xi_k - \xi_\star)$$

An alternative approach

Theorem

The spectral radius $\rho(T) < \rho$ if and only if there exists $P \succeq 0$ such that $T^T PT - \rho^2 P \preceq 0$.

- If $\xi_{k+1} - \xi_\star = T(\xi_k - \xi_\star)$ then

$$(\xi_{k+1} - \xi_\star)^T P(\xi_{k+1} - \xi_\star) < \rho^2(\xi_k - \xi_\star)^T P(\xi_k - \xi_\star)$$

- Iterating this, if $\rho < 1$, then

$$\|\xi_k - \xi_\star\| < \sqrt{\mathrm{cond}(P)}\,\rho^k \|\xi_0 - \xi_\star\|$$

Table of Contents

Unknown nasty function

- Suppose $u = \phi(y)$ (u and y are sequences and ϕ is nasty)

Unknown nasty function

- Suppose $u = \phi(y)$ (u and y are sequences and ϕ is nasty)
  - ϕ is static and memoryless: $\phi(y_0, y_1, \dots) = (g(y_0), g(y_1), \dots)$

Unknown nasty function

- Suppose $u = \phi(y)$ (u and y are sequences and ϕ is nasty)
  - ϕ is static and memoryless: $\phi(y_0, y_1, \dots) = (g(y_0), g(y_1), \dots)$
  - Further, g is L-Lipschitz: $\|g(y_1) - g(y_2)\| \leq L\|y_1 - y_2\|$

Unknown nasty function

- Suppose $u = \phi(y)$ (u and y are sequences and ϕ is nasty)
  - ϕ is static and memoryless: $\phi(y_0, y_1, \dots) = (g(y_0), g(y_1), \dots)$
  - Further, g is L-Lipschitz: $\|g(y_1) - g(y_2)\| \leq L\|y_1 - y_2\|$
- If $u_\star = g(y_\star)$ then for any k,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} L^2 I_d & 0_d \\ 0_d & -I_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0$$

Unknown nasty function

- Suppose $u = \phi(y)$ (u and y are sequences and ϕ is nasty)
  - ϕ is static and memoryless: $\phi(y_0, y_1, \dots) = (g(y_0), g(y_1), \dots)$
  - Further, g is L-Lipschitz: $\|g(y_1) - g(y_2)\| \leq L\|y_1 - y_2\|$
- If $u_\star = g(y_\star)$ then for any k,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} L^2 I_d & 0_d \\ 0_d & -I_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0$$

- This gives constraints on (y, u) – in fact, on each pair (y_k, u_k)

Unknown nasty function

- Suppose $u = \phi(y)$ (u and y are sequences and ϕ is nasty)
  - ϕ is static and memoryless: $\phi(y_0, y_1, \dots) = (g(y_0), g(y_1), \dots)$
  - Further, g is L-Lipschitz: $\|g(y_1) - g(y_2)\| \leq L\|y_1 - y_2\|$
- If $u_\star = g(y_\star)$ then for any k,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} L^2 I_d & 0_d \\ 0_d & -I_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0$$

- This gives constraints on (y, u) – in fact, on each pair (y_k, u_k)
- "Reference point" $(y_\star, u_\star)$ should make you think of arg min

Core idea

- Instead of analyzing a system containing ϕ, throw away ϕ but keep the constraints on some auxiliary sequence $z = \Psi(y, u)$

Core idea

- Instead of analyzing a system containing ϕ, throw away ϕ but keep the constraints on some auxiliary sequence $z = \Psi(y, u)$
- Any analysis that is valid for the constrained system is valid for the original

Modifying our dynamical system

- Auxiliary sequences ζ, z and map Ψ so that $\zeta_0 = \zeta_\star$,

$$\begin{aligned} \zeta_{k+1} &= A_\Psi \zeta_k + B_\Psi^y y_k + B_\Psi^u u_k \\ z_k &= C_\Psi \zeta_k + D_\Psi^y y_k + D_\Psi^u u_k \end{aligned}$$

Modifying our dynamical system

- Auxiliary sequences ζ, z and map Ψ so that $\zeta_0 = \zeta_\star$,

$$\begin{aligned} \zeta_{k+1} &= A_\Psi \zeta_k + B_\Psi^y y_k + B_\Psi^u u_k \\ z_k &= C_\Psi \zeta_k + D_\Psi^y y_k + D_\Psi^u u_k \end{aligned}$$

- If $\rho(A_\Psi) < 1$ then reference point $(\zeta_\star, z_\star)$ determined by $(y_\star, u_\star)$

Definition of IQCs

Definition

Let $u = \phi(y)$ and $z = \Psi(y, u)$. We say that ϕ satisfies the

Definition of IQCs

Definition

Let $u = \phi(y)$ and $z = \Psi(y, u)$. We say that ϕ satisfies the

- **Pointwise IQC** defined by $(\Psi, M, y_\star, u_\star)$ if for all sequences y,

$$(z_k - z_\star)^T M (z_k - z_\star) \geq 0 \ \forall k$$

Definition of IQCs

Definition

Let $u = \phi(y)$ and $z = \Psi(y, u)$. We say that ϕ satisfies the

- **Pointwise IQC** defined by $(\Psi, M, y_\star, u_\star)$ if for all sequences y,

$$(z_k - z_\star)^T M (z_k - z_\star) \geq 0 \;\forall k$$

- ρ-**Hard IQC** defined by $(\Psi, M, \rho, y_\star, u_\star)$ if for all sequences y,

$$\sum_{t=0}^{k} \rho^{-2t} (z_t - z_\star)^T M (z_t - z_\star) \geq 0 \;\forall k$$

Definition of IQCs

Definition

Let $u = \phi(y)$ and $z = \Psi(y, u)$. We say that ϕ satisfies the

- **Pointwise IQC** defined by $(\Psi, M, y_\star, u_\star)$ if for all sequences y,

$$(z_k - z_\star)^T M(z_k - z_\star) \geq 0 \ \forall k$$

- ρ-**Hard IQC** defined by $(\Psi, M, \rho, y_\star, u_\star)$ if for all sequences y,

$$\sum_{t=0}^{k} \rho^{-2t}(z_t - z_\star)^T M(z_t - z_\star) \geq 0 \ \forall k$$

- **Hard IQC** if satisfies ρ-Hard IQC for $\rho = 1$

Revisiting dynamical systems for first order methods

- Recall:

$$\begin{aligned} \xi_{k+1} &= A\xi_k + Bu_k \\ y_k &= C\xi_k \end{aligned}$$

Revisiting dynamical systems for first order methods

- Recall:

$$\xi_{k+1} = A\xi_k + Bu_k$$
$$y_k = C\xi_k$$

- Combine with the map Ψ and eliminate y:

$$\begin{bmatrix} \xi_{k+1} \\ \zeta_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ B_\Psi^y C & A_\Psi \end{bmatrix} \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} + \begin{bmatrix} B \\ B_\Psi^u \end{bmatrix} u_k$$
$$z_k = \begin{bmatrix} D_\Psi^y C & C_\Psi \end{bmatrix} \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} + D_\Psi^u u_k$$

Revisiting dynamical systems for first order methods

- Recall:

$$\xi_{k+1} = A\xi_k + Bu_k$$
$$y_k = C\xi_k$$

- Combine with the map Ψ and eliminate y:

$$\begin{bmatrix} \xi_{k+1} \\ \zeta_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ B_\Psi^y C & A_\Psi \end{bmatrix} \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} + \begin{bmatrix} B \\ B_\Psi^u \end{bmatrix} u_k$$
$$z_k = \begin{bmatrix} D_\Psi^y C & C_\Psi \end{bmatrix} \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} + D_\Psi^u u_k$$

- More succinctly:

$$x_{k+1} = \hat{A}x_k + \hat{B}u_k$$
$$z_k = \hat{C}x_k + \hat{D}u_k$$

Main result

Theorem

Suppose $(\xi_\star, \zeta_\star, y_\star, u_\star, z_\star)$ *is a fixed point of the system. Suppose* ϕ *satisfies the* ρ*-hard IQC defined by* $(\Psi, M, \rho, y_\star, u_\star)$ *for* $\rho \in [0, 1]$*. If the LMI*

$$\begin{bmatrix} \hat{A}^T P \hat{A} - \rho^2 P & \hat{A}^T P \hat{B} \\ \hat{B}^T P \hat{A} & \hat{B}^T P \hat{B} \end{bmatrix} + \lambda \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix}^T M \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix} \preceq 0$$

is feasible for some $P \succ 0$ *and* $\lambda \geq 0$*, then for any* ξ_0 *we have*

$$\|\xi_k - \xi_\star\| \leq \sqrt{\mathrm{cond}(P)} \rho^k \|\xi_0 - \xi_\star\| \ \forall k.$$

Main result

Theorem

Suppose $(\xi_\star, \zeta_\star, y_\star, u_\star, z_\star)$ *is a fixed point of the system. Suppose* ϕ *satisfies the* ρ*-hard IQC defined by* $(\Psi, M, \rho, y_\star, u_\star)$ *for* $\rho \in [0,1]$*. If the LMI*

$$\begin{bmatrix} \hat{A}^T P \hat{A} - \rho^2 P & \hat{A}^T P \hat{B} \\ \hat{B}^T P \hat{A} & \hat{B}^T P \hat{B} \end{bmatrix} + \lambda \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix}^T M \begin{bmatrix} \hat{C} & \hat{D} \end{bmatrix} \preceq 0$$

is feasible for some $P \succ 0$ *and* $\lambda \geq 0$*, then for any* ξ_0 *we have*

$$\|\xi_k - \xi_\star\| \leq \sqrt{\mathrm{cond}(P)}\rho^k \|\xi_0 - \xi_\star\| \ \forall k.$$

Proof.

Multiply on both sides by $\begin{bmatrix} (x_k - x_\star)^T & (u_k - u_\star)^T \end{bmatrix}$ and its transpose. Then use the definition of ρ-hard IQC to find that $\|x_k - x_\star\| \leq \sqrt{\mathrm{cond}(P)}\rho^k \|x_0 - x_\star\|$. Finally, use $\zeta_0 = \zeta_\star$, $x = (\xi, \zeta)$, and the triangle inequality. □

A few notes

- Pointwise IQC satisfied $\implies$ ρ-hard IQC satisfied for any ρ, so find the smallest ρ with the LMI feasible

A few notes

- Pointwise IQC satisfied $\implies$ ρ-hard IQC satisfied for any ρ, so find the smallest ρ with the LMI feasible
- Hard IQC means 1-hard IQC, which implies bounded iterates but not convergence

A few notes

- Pointwise IQC satisfied $\implies$ ρ-hard IQC satisfied for any ρ, so find the smallest ρ with the LMI feasible
- Hard IQC means 1-hard IQC, which implies bounded iterates but not convergence
- If ϕ satisfies multiple IQCs, replace λM with a block diagonal matrix with $\lambda_i M_i$ on the diagonal

Lemma (Sector IQC)

Suppose $f_k \in S(m, L)$ and $u_\star = \nabla f_k(y_\star)$ for all k. Let $\phi = (\nabla f_0, \nabla f_1, \dots)$. If $u = \phi(y)$, then ϕ satisfies the pointwise IQC defined by

$$\Psi = \begin{bmatrix} LI_d & -I_d \\ -mI_d & I_d \end{bmatrix} \text{ and } M = \begin{bmatrix} 0_d & I_d \\ I_d & 0_d \end{bmatrix}.$$

This corresponds to the constraint that for all sequences y,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} -2mLI_d & (L+m)I_d \\ (L+m)I_d & -2I_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0.$$

Lemma (Sector IQC)

Suppose $f_k \in S(m, L)$ and $u_\star = \nabla f_k(y_\star)$ for all k. Let $\phi = (\nabla f_0, \nabla f_1, \dots)$. If $u = \phi(y)$, then ϕ satisfies the pointwise IQC defined by

$$\Psi = \begin{bmatrix} L I_d & -I_d \\ -m I_d & I_d \end{bmatrix} \text{ and } M = \begin{bmatrix} 0_d & I_d \\ I_d & 0_d \end{bmatrix}.$$

This corresponds to the constraint that for all sequences y,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} -2mL I_d & (L+m) I_d \\ (L+m) I_d & -2 I_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0.$$

Note: this Ψ corresponds to no ζ, and

$$z = \Psi \begin{bmatrix} y \\ u \end{bmatrix} = \begin{bmatrix} Ly - u \\ -my + u \end{bmatrix}$$

Sector IQC proof

Proof.

If f has L-Lipschitz gradient, then we have

$$(x_1 - x_2)^T(\nabla f(x_1) - \nabla f(x_2)) \geq \frac{1}{L}\|\nabla f(x_1) - \nabla f(x_2)\|^2$$

which is known as *co-coercivity*. Note $f(x) - \frac{m}{2}\|x\|^2 \in S(0, L-m)$ has Lipschitz gradient with parameter $L - m$. By co-coercivity, and replacing x_1, x_2 with $y_k, y_\star$, etc., we see that

$$(m + L)(y_k - y_\star)^T(u_k - u_\star) \geq mL\|y_k - y_\star\|^2 + \|u_k - u_\star\|^2$$

which we can rearrange into matrix form. □

Lemma (IQC for general convex functions)

Suppose $f_k \in S(0, \infty)$ and $u_\star \in \partial f_k(y_\star)$ for all k. Let ϕ be such that $u_k \in \partial f_k(y_k)$ for all k. Then ϕ satisfies the pointwise IQC defined by

$$\Psi = \begin{bmatrix} I_d & 0 \\ 0 & I_d \end{bmatrix} = I_{2d} \text{ and } M = \begin{bmatrix} 0_d & I_d \\ I_d & 0_d \end{bmatrix}.$$

This corresponds to the constraint that for all sequences y,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} 0_d & I_d \\ I_d & 0_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0.$$

Lemma (IQC for general convex functions)

Suppose $f_k \in S(0, \infty)$ and $u_\star \in \partial f_k(y_\star)$ for all k. Let ϕ be such that $u_k \in \partial f_k(y_k)$ for all k. Then ϕ satisfies the pointwise IQC defined by

$$\Psi = \begin{bmatrix} I_d & 0 \\ 0 & I_d \end{bmatrix} = I_{2d} \textit{ and } M = \begin{bmatrix} 0_d & I_d \\ I_d & 0_d \end{bmatrix}.$$

This corresponds to the constraint that for all sequences y,

$$\begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix}^T \begin{bmatrix} 0_d & I_d \\ I_d & 0_d \end{bmatrix} \begin{bmatrix} y_k - y_\star \\ u_k - u_\star \end{bmatrix} \geq 0.$$

Proof.

This is equivalent to $(y_k - y_\star)^T(u_k - u_\star) \geq 0$, i.e. that the subdifferential of a convex function is a monotone operator. (combine $f(y_\star) \geq f(y_k) + u_k^T(y_\star - y_k)$ and vice-versa per EE236C) □

Table of Contents

SDP tractibility

- We prove convergence by finding $P \succ 0$ ... how big is P?

SDP tractibility

- We prove convergence by finding $P \succ 0$ ... how big is P?
- Our LMI has the term $\hat{A}^T P \hat{A}$, where $\hat{A}$ operates on (ξ, ζ). Hence P is $n \times n$ where $(\xi, \zeta) \in \mathbb{R}^n$.

SDP tractibility

- We prove convergence by finding $P \succ 0$ ... how big is P?
- Our LMI has the term $\hat{A}^T P \hat{A}$, where $\hat{A}$ operates on (ξ, ζ). Hence P is $n \times n$ where $(\xi, \zeta) \in \mathbb{R}^n$.
- Better than Drori and Teboulle '13, where the SDP scales with the number of time steps, but still too large to e.g. analyze gradient descent in high dimensions.

Structure in our linear maps

- First-order methods in dynamical system form often have block-diagonal structure

Structure in our linear maps

- First-order methods in dynamical system form often have block-diagonal structure
- Nesterov's accelerated gradient method has

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} (1+\beta)I_d & -\beta I_d & -\alpha I_d \\ I_d & 0_d & 0_d \\ \hline (1+\beta)I_d & -\beta I_d & 0_d \end{array}\right]$$

Structure in our linear maps

- First-order methods in dynamical system form often have block-diagonal structure
- Nesterov's accelerated gradient method has

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} (1+\beta)I_d & -\beta I_d & -\alpha I_d \\ I_d & 0_d & 0_d \\ \hline (1+\beta)I_d & -\beta I_d & 0_d \end{array}\right]$$

- For example,

$$A = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} \otimes I_d$$

Structure in our linear maps

- First-order methods in dynamical system form often have block-diagonal structure
- Nesterov's accelerated gradient method has

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} (1+\beta)I_d & -\beta I_d & -\alpha I_d \\ I_d & 0_d & 0_d \\ \hline (1+\beta)I_d & -\beta I_d & 0_d \end{array}\right]$$

- For example,

$$A = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix} \otimes I_d$$

- Even our IQCs have this form, e.g.

$$\Psi = \begin{bmatrix} L & -1 \\ -m & 0 \end{bmatrix} \otimes I_d \text{ and } M = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes I_d$$

for the sector IQC

Making the SDP small

- If each matrix $(\hat{A}, \hat{B}, \hat{C}, \hat{D}, M)$ from the LMI has the form e.g. $\hat{A} = \hat{A}_0 \otimes I_d$ then we can instead solve the smaller LMI (which is the equivalent of the $d = 1$ case):

$$\begin{bmatrix} \hat{A}_0^T P_0 \hat{A}_0 - \rho^2 P_0 & \hat{A}_0^T P_0 \hat{B}_0 \\ \hat{B}_0^T P_0 \hat{A}_0 & \hat{B}_0^T P_0 \hat{B}_0 \end{bmatrix} + \lambda \begin{bmatrix} \hat{C}_0 & \hat{D}_0 \end{bmatrix}^T M_0 \begin{bmatrix} \hat{C}_0 & \hat{D}_0 \end{bmatrix} \preceq 0$$

Making the SDP small

- If each matrix $(\hat{A}, \hat{B}, \hat{C}, \hat{D}, M)$ from the LMI has the form e.g. $\hat{A} = \hat{A}_0 \otimes I_d$ then we can instead solve the smaller LMI (which is the equivalent of the $d = 1$ case):

$$\begin{bmatrix} \hat{A}_0^T P_0 \hat{A}_0 - \rho^2 P_0 & \hat{A}_0^T P_0 \hat{B}_0 \\ \hat{B}_0^T P_0 \hat{A}_0 & \hat{B}_0^T P_0 \hat{B}_0 \end{bmatrix} + \lambda \begin{bmatrix} \hat{C}_0 & \hat{D}_0 \end{bmatrix}^T M_0 \begin{bmatrix} \hat{C}_0 & \hat{D}_0 \end{bmatrix} \preceq 0$$

- We can get feasible P_0 from P and vice-versa, so solving this smaller SDP is completely equivalent

Making the SDP small

- If each matrix $(\hat{A}, \hat{B}, \hat{C}, \hat{D}, M)$ from the LMI has the form e.g. $\hat{A} = \hat{A}_0 \otimes I_d$ then we can instead solve the smaller LMI (which is the equivalent of the $d = 1$ case):

$$\begin{bmatrix} \hat{A}_0^T P_0 \hat{A}_0 - \rho^2 P_0 & \hat{A}_0^T P_0 \hat{B}_0 \\ \hat{B}_0^T P_0 \hat{A}_0 & \hat{B}_0^T P_0 \hat{B}_0 \end{bmatrix} + \lambda \begin{bmatrix} \hat{C}_0 & \hat{D}_0 \end{bmatrix}^T M_0 \begin{bmatrix} \hat{C}_0 & \hat{D}_0 \end{bmatrix} \preceq 0$$

- We can get feasible P_0 from P and vice-versa, so solving this smaller SDP is completely equivalent
- In the first order methods we have looked at so far, this means P_0 is no bigger than 2×2

Analytic results for gradient descent

- Using the sector IQC and the dimensionality reduction, the LMI for gradient descent is

$$\begin{bmatrix} (1-\rho^2)P & -\alpha P \\ -\alpha P & \alpha^2 P \end{bmatrix} + \lambda \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \preceq 0$$

Analytic results for gradient descent

- Using the sector IQC and the dimensionality reduction, the LMI for gradient descent is

$$\begin{bmatrix} (1-\rho^2)P & -\alpha P \\ -\alpha P & \alpha^2 P \end{bmatrix} + \lambda \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \preceq 0$$

- For $\alpha = \frac{2}{L+m}$ (optimal for f quadratic), we find $\lambda \geq \frac{2}{(L+m)^2}$ and $\rho^2 \geq \frac{1}{2}\lambda(L-m)^2$ which yields optimal $\rho = \frac{L-m}{L+m}$.

Analytic results for gradient descent

- Using the sector IQC and the dimensionality reduction, the LMI for gradient descent is

$$\begin{bmatrix} (1-\rho^2)P & -\alpha P \\ -\alpha P & \alpha^2 P \end{bmatrix} + \lambda \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \preceq 0$$

- For $\alpha = \frac{2}{L+m}$ (optimal for f quadratic), we find $\lambda \geq \frac{2}{(L+m)^2}$ and $\rho^2 \geq \frac{1}{2}\lambda(L-m)^2$ which yields optimal $\rho = \frac{L-m}{L+m}$.
- Can reformulate LMI so that it is linear in $(\rho^2, \lambda, \alpha)$. Hence, can answer "what range of stepsizes yield a given rate?" etc.

Analyzing Nesterov's method

- Analyze $\alpha = 1/L$ and $\beta = (\sqrt{L} - \sqrt{m})/(\sqrt{L} + \sqrt{m})$ which are optimal when f is quadratic

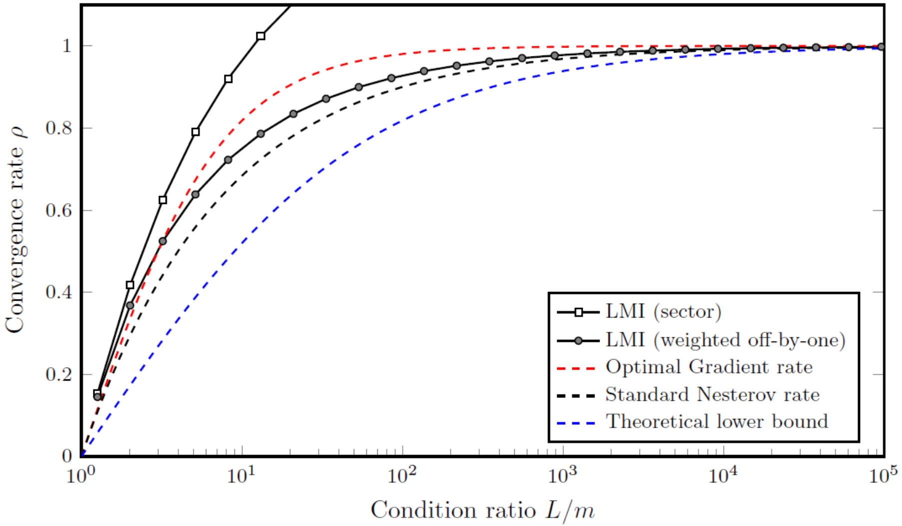
Analyzing Nesterov's method

- Analyze $\alpha = 1/L$ and $\beta = (\sqrt{L} - \sqrt{m})/(\sqrt{L} + \sqrt{m})$ which are optimal when f is quadratic
- Solve the LMI numerically. LMI is no longer linear in ρ^2 but can find optimal via bisection search

Analyzing Nesterov's method

- Analyze $\alpha = 1/L$ and $\beta = (\sqrt{L} - \sqrt{m})/(\sqrt{L} + \sqrt{m})$ which are optimal when f is quadratic
- Solve the LMI numerically. LMI is no longer linear in ρ^2 but can find optimal via bisection search
- Sector IQC actually fails for high $\kappa = L/m$, but more sophisticated *weighted off-by-one IQC* works

Convergence rate v. condition ratio



Robustness of Nesterov's method

- Sector IQC (which allows different functions f_k for each k) fails for large κ, unlike gradient descent

Robustness of Nesterov's method

- Sector IQC (which allows different functions f_k for each k) fails for large κ, unlike gradient descent
- Optimal parameters $\alpha = 4/(3L + m)$ and $\beta = \frac{\sqrt{3\kappa+1}-2}{\sqrt{3\kappa+1}+2}$ cause sector IQC to fail faster

Robustness of Nesterov's method

- Sector IQC (which allows different functions f_k for each k) fails for large κ, unlike gradient descent
- Optimal parameters $\alpha = 4/(3L+m)$ and $\beta = \frac{\sqrt{3\kappa+1}-2}{\sqrt{3\kappa+1}+2}$ cause sector IQC to fail faster
- In some sense, gradient descent, and even the suboptimal parameters α, β more robust than fully optimal Nesterov

Robustness of heavy ball method

- Recall the heavy ball method:

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

Robustness of heavy ball method

- Recall the heavy ball method:

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

- For quadratic-optimal α, β for heavy ball method, not even weighted off-by-one IQC can guarantee convergence for $\kappa = L/m$ at least ≈ 18.

Robustness of heavy ball method

- Recall the heavy ball method:

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

- For quadratic-optimal α, β for heavy ball method, not even weighted off-by-one IQC can guarantee convergence for $\kappa = L/m$ at least ≈ 18.
- Informs a function $f(x)$ with piecewise-linear gradient and $\kappa = L/m = 25$ for which heavy ball method optimized for quadratics does not converge

ADMM background

- ADMM seeks to solve the problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

ADMM background

- ADMM seeks to solve the problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- Updates are of the form:

$$\begin{aligned} x_{k+1} &= \arg\min_x f(x) + \frac{\rho}{2}\|Ax + Bz_k - c + u_k\|^2 \\ z_{k+1} &= \arg\min_z g(z) + \frac{\rho}{2}\|Ax_{k+1} + Bz - c + u_k\|^2 \\ u_{k+1} &= u_k + Ax_{k+1} + Bz_{k+1} - c \end{aligned}$$

ADMM background

- ADMM seeks to solve the problem

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & Ax + Bz = c \end{array}$$

- Updates are of the form:

$$\begin{aligned} x_{k+1} &= \arg\min_x f(x) + \frac{\rho}{2}\|Ax + Bz_k - c + u_k\|^2 \\ z_{k+1} &= \arg\min_z g(z) + \frac{\rho}{2}\|Ax_{k+1} + Bz - c + u_k\|^2 \\ u_{k+1} &= u_k + Ax_{k+1} + Bz_{k+1} - c \end{aligned}$$

- Over-relaxed ADMM given by replacing Ax_{k+1} with $\alpha Ax_{k+1} - (1-\alpha)(Bz_k - c)$ in z and u updates. Typically $\alpha \in (0, 2]$

ADMM as a dynamical system

- Assume $f \in S(m, L)$ and $g \in S(0, \infty)$. Then instead of one sequence u_k of gradients of y_k, instead have two sequences $\beta_k = \nabla \hat{f}(r_k)$ and $\gamma_k \in \partial \hat{g}(s_k)$ ($\hat{f}$ and $\hat{g}$ are versions of f, g scaled by A, B, ρ)

ADMM as a dynamical system

- Assume $f \in S(m, L)$ and $g \in S(0, \infty)$. Then instead of one sequence u_k of gradients of y_k, instead have two sequences $\beta_k = \nabla \hat{f}(r_k)$ and $\gamma_k \in \partial \hat{g}(s_k)$ ($\hat{f}$ and $\hat{g}$ are versions of f, g scaled by A, B, ρ)
- Then we can write x, z iterates (now called r, s) in terms of β, γ, e.g.

$$x_{k+1} = A^{-1} \arg\min_r f(A^{-1} r) + \frac{\rho}{2} \|r + s_k - c + u_k\|^2$$

ADMM as a dynamical system

- Assume $f \in S(m, L)$ and $g \in S(0, \infty)$. Then instead of one sequence u_k of gradients of y_k, instead have two sequences $\beta_k = \nabla \hat{f}(r_k)$ and $\gamma_k \in \partial \hat{g}(s_k)$ ($\hat{f}$ and $\hat{g}$ are versions of f, g scaled by A, B, ρ)
- Then we can write x, z iterates (now called r, s) in terms of β, γ, e.g.

$$x_{k+1} = A^{-1} \arg\min_r f(A^{-1}r) + \frac{\rho}{2}\|r + s_k - c + u_k\|^2$$

$$\implies r_{k+1} = \arg\min_r \hat{f}(r) + \frac{1}{2}\|r + s_k - c + u_k\|^2$$

ADMM as a dynamical system

- Assume $f \in S(m, L)$ and $g \in S(0, \infty)$. Then instead of one sequence u_k of gradients of y_k, instead have two sequences $\beta_k = \nabla \hat{f}(r_k)$ and $\gamma_k \in \partial \hat{g}(s_k)$ ($\hat{f}$ and $\hat{g}$ are versions of f, g scaled by A, B, ρ)
- Then we can write x, z iterates (now called r, s) in terms of β, γ, e.g.

$$x_{k+1} = A^{-1} \arg\min_r f(A^{-1}r) + \frac{\rho}{2}\|r + s_k - c + u_k\|^2$$

$$\implies r_{k+1} = \arg\min_r \hat{f}(r) + \frac{1}{2}\|r + s_k - c + u_k\|^2$$

and via optimality conditions implies

$$r_{k+1} = -s_k - u_k + c - \beta_{k+1}.$$

IQCs for ADMM

- One IQC for each of f, g

IQCs for ADMM

- One IQC for each of f, g
- Sector IQC for $f \in S(m, L)$ and corresponding iterates

IQCs for ADMM

- One IQC for each of f, g
- Sector IQC for $f \in S(m, L)$ and corresponding iterates
- More general pointwise IQC for $g \in S(0, \infty)$ and corresponding iterates

IQCs for ADMM

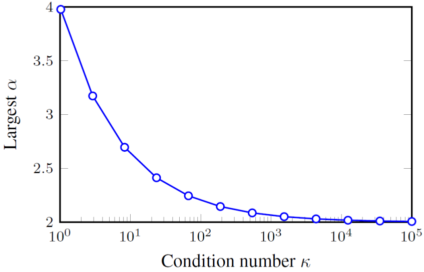
- One IQC for each of f, g
- Sector IQC for $f \in S(m, L)$ and corresponding iterates
- More general pointwise IQC for $g \in S(0, \infty)$ and corresponding iterates
- Put M_1 and M_2 into a block diagonal matrix and solve

IQCs for ADMM

- One IQC for each of f, g
- Sector IQC for $f \in S(m, L)$ and corresponding iterates
- More general pointwise IQC for $g \in S(0, \infty)$ and corresponding iterates
- Put M_1 and M_2 into a block diagonal matrix and solve
- Given fixed α, ρ, m, L, can bisection search on convergence rates τ.

Some results for ADMM

- Prior work limits us to $\alpha \in (0, 2)$ but depending on κ, we can find convergent α larger than 2



Some results for ADMM

- Prior work limits us to $\alpha \in (0, 2)$ but depending on κ, we can find convergent α larger than 2
- Also able to analytically construct certificates λ, P that work for large enough κ (for $\alpha \in (0, 2)$ and specific choice of ρ

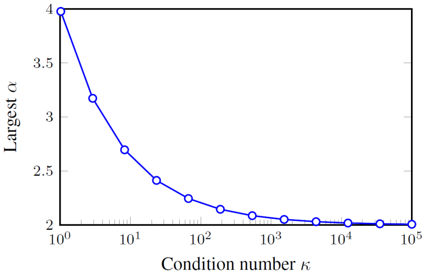


Table of Contents

Multiplicative gradient noise

- Suppose instead of observing $\nabla f(y)$, we see $u_k = \nabla f(y_k) + r_k$, where $\|r_k\| \leq \delta \|\nabla f(y_k)\|$

Multiplicative gradient noise

- Suppose instead of observing $\nabla f(y)$, we see $u_k = \nabla f(y_k) + r_k$, where $\|r_k\| \leq \delta \|\nabla f(y_k)\|$
- If w_k is true gradient, we observe u_k with $\|u_k - w_k\| \leq \delta \|w_k\|$

Multiplicative gradient noise

- Suppose instead of observing $\nabla f(y)$, we see $u_k = \nabla f(y_k) + r_k$, where $\|r_k\| \leq \delta \|\nabla f(y_k)\|$
- If w_k is true gradient, we observe u_k with $\|u_k - w_k\| \leq \delta \|w_k\|$
- In IQC form:

$$\begin{bmatrix} w_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \delta^2 - 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} w_k \\ u_k \end{bmatrix} \geq 0$$

Multiplicative gradient noise

- Suppose instead of observing $\nabla f(y)$, we see $u_k = \nabla f(y_k) + r_k$, where $\|r_k\| \leq \delta \|\nabla f(y_k)\|$
- If w_k is true gradient, we observe u_k with $\|u_k - w_k\| \leq \delta \|w_k\|$
- In IQC form:

$$\begin{bmatrix} w_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \delta^2 - 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} w_k \\ u_k \end{bmatrix} \geq 0$$

- We can use nearly the same LMI after augmenting our state with w, i.e. we keep track of (y, u, w), and instead solve for 3×3 P for e.g. Nesterov's method

Nesterov's method convergence rates with noisy gradient

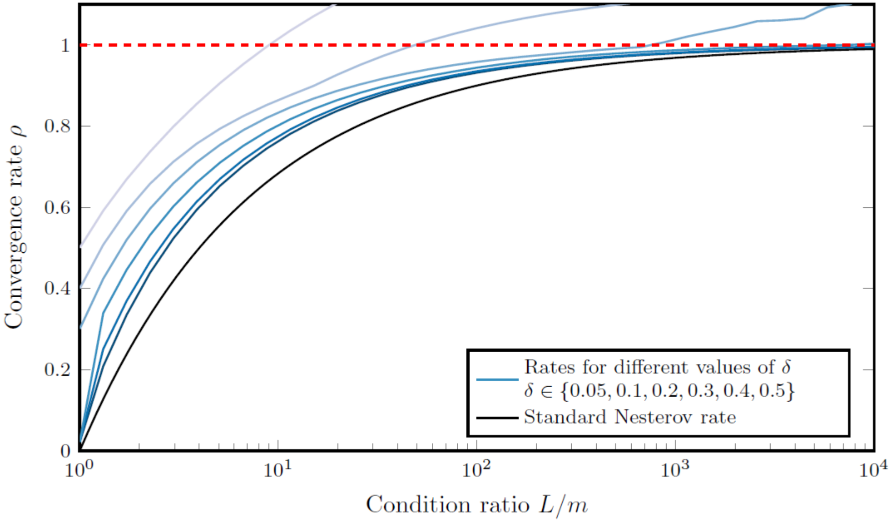


Table of Contents

Summary

- Many optimization methods are (almost) linear dynamical systems

Summary

- Many optimization methods are (almost) linear dynamical systems
- IQCs can replace nonlinearities in these systems

Summary

- Many optimization methods are (almost) linear dynamical systems
- IQCs can replace nonlinearities in these systems
- IQCs exist which capture standard properties of convex functions

Summary

- Many optimization methods are (almost) linear dynamical systems
- IQCs can replace nonlinearities in these systems
- IQCs exist which capture standard properties of convex functions
- Automatic numerical convergence rate bounds whenever we have bounds on m, L and (in the noisy case) δ

Summary

- Many optimization methods are (almost) linear dynamical systems
- IQCs can replace nonlinearities in these systems
- IQCs exist which capture standard properties of convex functions
- Automatic numerical convergence rate bounds whenever we have bounds on m, L and (in the noisy case) δ
- Hence easy parameter tuning/algorithm design

Future work

- Analytic proofs doable if we can solve small SDPs in closed form

Future work

- Analytic proofs doable if we can solve small SDPs in closed form
- We don't usually know m, L; connections to e.g. adaptive control?

Future work

- Analytic proofs doable if we can solve small SDPs in closed form
- We don't usually know m, L; connections to e.g. adaptive control?
- More sophisticated parameter search needed if we want more steps of memory

Future work

- Analytic proofs doable if we can solve small SDPs in closed form
- We don't usually know m, L; connections to e.g. adaptive control?
- More sophisticated parameter search needed if we want more steps of memory
- Noise analysis is far from complete; IQCs that are valid in expectation?

Future work

- Analytic proofs doable if we can solve small SDPs in closed form
- We don't usually know m, L; connections to e.g. adaptive control?
- More sophisticated parameter search needed if we want more steps of memory
- Noise analysis is far from complete; IQCs that are valid in expectation?
- We translated convexity properties into IQCs; are there useful IQCs for certain nonconvex functions?