An Optimal Control Framework for First Order Methods

Robert Bassett

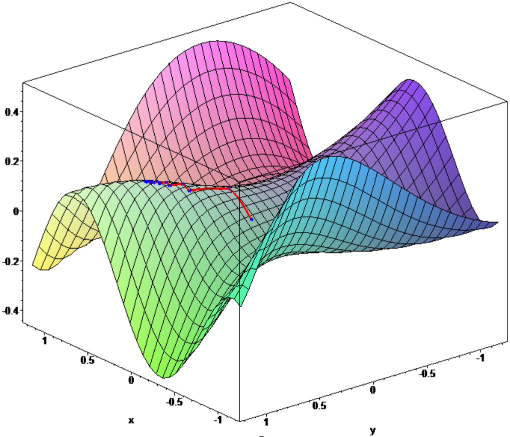
UC Davis Department of Mathematics

March 2015

Introduction

Problem: Analyzing first order methods is hard!

Question: Is there a unified framework for investigating these algorithms?



Dynamical System Basics

Definition

A *Linear Dynamical System* is a set of recursive linear equations

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

$$y_k = C\xi_k + Du_k.$$

u_k in the *input*, y_k is the *output*, and ξ_k is the *state* at time k.

We can connect this linear system in *feedback* with a nonlinearity Δ by including

$$u_k = \Delta(y_k)$$

in the rules above.

For our purposes, the nonlinearity has the form $\Delta(y) = \nabla f(y)$.

Some Basics

Definition

$S(m, L)$ is the set of continuously differentiable, strongly convex with parameter m and have Lipschitz gradients with parameter L. In other words,

$$m\,||x-y||^2 \leq (\nabla f(x) - \nabla f(y))^T (x-y) \leq L\,||x-y||^2 \,.$$

All the functions we consider will be in this class.

First Order Methods

Gradient Descent

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

Nesterov's Accelerated Gradient Descent

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

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

Heavy-Ball Method

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

Each of the above methods can be written as a linear dynamical system in feedback for choice of A, B, C, D.

Big Picture Summary

- Cast a first order method as a linear dynamical system in feedback.
- Use *Integral Quadratic Constraints* to overcome non-linear feedback.
- Convergence rates can be found by establishing feasiblity of a certain *Semi-Definite Program*.

Quadratic Problems

Assume that f is a convex, quadratic function

$$f(x) = \frac{1}{2}x^T Qx - p^T x + r.$$

- $\nabla f(x) = Qx - p$
- The optimal solution is $x_* = Q^{-1}p$.

We assume that $D = 0$, as is the case in GD, NAGD, and HBM.

$$\begin{aligned} \xi_{k+1} &= A\xi_k + Bu_k \\ y_k &= C\xi_k + Du_k \\ u_k &= Qy_k - p \end{aligned} \quad \rightarrow \xi_{k+1} = A\xi_k + BQy_k - p \rightarrow \xi_{k+1} = (A+BQC)\xi_k - c$$

If ξ_* is a fixed point of the dyn. sys. then $\xi_* = (A + BQC)\xi_* - c$.

Quadratic Problems

$$\xi_{k+1} - \xi_* = (A + BQC)(\xi_k - \xi_*)$$

A necessary and sufficient condition for $\xi_k \to \xi_*$ is that $T := A + BQC$ has spectral radius strictly less than one.

FACTS:

- $\rho(M) \leq \left|\left|M^K\right|\right|^{1/k}$ for all k
- $\rho(M) = \lim_{k\to\infty} \left|\left|M^K\right|\right|^{1/k}$

So for any ϵ and k large enough, we can bound the convergence rate

$$||\xi_k - \xi_*|| = \left|\left|T^K(\xi_0 - \xi_*\right|\right| \leq \left|\left|T^k\right|\right| ||\xi_0 - \xi_*|| \leq (\rho(T) + \epsilon)^k ||\xi_0 - \xi_*|| .$$

Linear Matrix Inequalities

The following theorem connects the spectral radius to the feasibility of an SDP.

Theorem

$\rho(T) < \rho$ if and only if there exists a $P \succ 0$ satisfying

$$T^T P T - \rho^2 P \prec 0.$$

Integral Quadratic Constraints

Integral Quadratic Constraints cope with the nonlinearity of the gradient in the non-quadratic case.

Idea: Replace nonlinear component by a quadratic constraint on its inputs and outputs that is know to be satisfied by all possible instances of the component.

There are different types of IQCs

$$\{\text{Pointwise IQCs}\} \subset \{\rho - \text{Hard IQCs}\} \subset \{\text{Hard IQCs}\} \subset \{\text{all soft IQCs}\}.$$

Main Theorem

Theorem

Main Theorem Suppose ϕ satisfies a certain ρ-hard IQC and $0 \le \rho \le 1$. If

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

is feasible for some $P \succ 0$ and $\lambda \ge 0$, then for any ξ_0

$$||\xi_k - \xi_*|| \le \sqrt{cond(P)}\rho^k \, ||\xi_0 - \xi_*||$$

for all k, where $cond(P)$ is the condition number of P

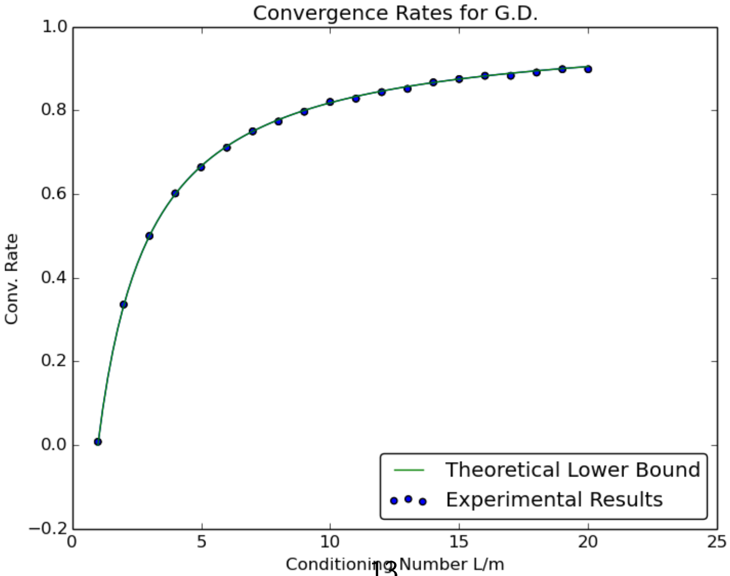
$\hat{A}, \hat{B}, \hat{C}, \hat{D}$, and M all come from the IQC.

My Experiments

I confirmed the results for Gradient Descent myself.

- Pointwise IQC (suffices for the GD case)
- Used Convex in Julia, which is a frontend for solving convex problems in julia (open source).
- Solver: SCS = splitting conic solver (open source, developed by Stanford Univeristy Convex Optimization Group).
- I computed the best ρ and compared it with the theoretical rate of $\frac{L-m}{L+m}$.

Results



Convergence Rates for G.D.
Conv. Rate
Conditioning Number L/m
Theoretical Lower Bound
Experimental Results
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25

References

L. Lessard, B. Recht, A. Packard, *Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints*. http://arxiv.org/abs/1408.3595

julialang.org

Udell, Mohan, Zeng, Hong, Diamond, Boyd, *Convex Optimization in Julia* Proceedings of the 1st First Workshop for High Performance Technical Computing in Dynamic Languages, 2014.