EE/AA547-W22 2022.02.17

Topic: NLP and Intro to DP

Lecturer: L.J. Ratliff

1 Review

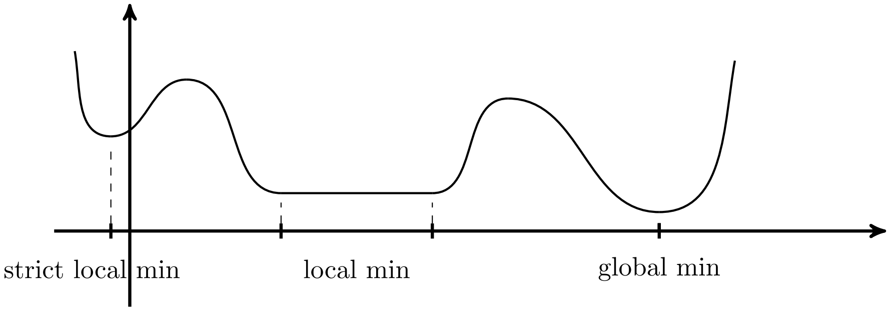
Some great references:

- Bertsekas Nonlinear Programming
- Bertsekas Dynamic Programming and Optimal Control Vols I, II
- Bertsekas Neurodynamic Programming
- Kirk Optimal Control

1.1 Non-Linear Programming Overview

The basics of nonlinear programming you have probably seen in your calculus sequence even if you did not know it. The idea is that for an unconstrained problem, we can simply look at derivatives (first and second order) to determine necessary and sufficient conditions for (global/local) "optimality"—i.e., a critical point that is either a (global/local) minimum or maximum.

For example consider the sufficiently smooth cost function $F : \mathbb{R}^n \to \mathbb{R}$, and suppose we want to find a minimum of this cost. Consider the following cartoon graphic:



Definition 1. A vector $x^\star$ is a local minimum of F if it is no worse than its neighbors; i.e., if there exists an $\epsilon > 0$ such that

$$F(x^\star) \leq F(x) \quad \forall\ x : \ \|x - x^\star\| \leq \epsilon$$

It is a global minimum if this inequality holds for all x. And, it is strict if the inequality is strict for all $x \neq x^\star$.

Necessary Conditions:

- $DF(x^\star) = 0$
- $D^2F(x^\star) \succeq 0$

Sufficient Conditions:

- $DF(x^\star) = 0$

- $D^2F(x^\star) \succ 0$

These sufficient conditions in particular are equivalent to the following: there exists $\gamma > 0$ and $\epsilon > 0$ such that

$$F(x) \geq F(x^\star) + \frac{\gamma}{2}\|x - x^\star\|^2, \quad \forall\ x \text{ with } \|x - x^\star\| < \epsilon$$

1.1.1 Constrained NLP

The more interesting and more apropos NLP setting is the constrained setting. This will be relevant for us since we typically are seeking an optimal controller for an objective of the form $F(x, u)$ where $f(x, u) = 0$ is the constrained defined by the dynamics—e.g., $f(x, u) = \dot{x} - (Ax + Bu)$. And in particular we seek to minimize costs $J(u)$ were the state constraints implicit specify x in terms of u.

Consider the optimization problem

$$\begin{aligned} \min_u \ & F(x, u) \\ \text{s.t.} \ & f(x, u) = 0 \end{aligned}$$

There are several types of ”constraint satisfaction/qualification” conditions specifying sufficient conditions for such problems. We will take the approach of writing out the Lagrangian and looking at the sufficient conditions of the corresponding unconstrained problem. Indeed, let $\lambda \in \mathbb{R}^n$ (where there are n constraints) be the Lagrange multiplier.

$$D_x f(x, u) = 0 \quad \Longrightarrow$$

Then the Lagrangian is

$$L(x, u, \lambda) = F(x, u) + \lambda^\top f(x, u)$$

Since $f(x, u) = 0$ and we are optimizing over u here we can use the elimination method to remove one of the variables. The best way to see this is with a simple example.

Problem 1. (Elimination Method.)Consider the optimization problem

$$\begin{aligned} \min_u \ & F(x, u) \\ \text{s.t.} \ & Gx + Hu = b \end{aligned}$$

where G is invertible. The matrix $[G\ H]$ is an $n \times (n + m)$ matrix where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. Suppose it has linearly independent rows and $b \in \mathbb{R}^n$ is given. Here $G \in \mathbb{R}^{n\times n}$ and $H \in \mathbb{R}^{n\times m}$. Write out first order optimality conditions.
Solution.

Optimality Conditions. The general optimality conditions we care about are the following:

- optimal Lagrange multiplier:

$$\nabla_x L(x,u,\lambda) = \nabla_x F(x,u) + \lambda^\top \nabla_x f(x,u) = 0 \implies (\lambda^\star)^\top = -\nabla_x F(x,u)(\nabla_x f(x,u))^{-1}$$

- Constraint satisfaction:

$$f(x,u) = 0$$

- Optimality of u:

$$\nabla_u L(x,u,\lambda) = \nabla_u F(x,u) + \lambda^\top \nabla_u f(x,u) = 0$$

1.2 DT LQR Problem

Consider

$$x(t+1) = Ax(t) + Bu(t),\ x(0) = x_0$$

and

$$J(u) = x_N^\top Q_f x_N + \sum_{k=0}^{N-1} \left(x_k^\top Q x_k + u_k^\top R u_k \right)$$

where $u = (u_0, \ldots, u_{N-1})$ and $Q = Q^\top \succeq 0$, $Q_f = Q_f^\top \succeq 0$, $R = R^\top \succ 0$ are the given state cost, final state cost, and input cost matrices.

- N is the time horizon
- first term measures state deviation
- second term measures input size or actuator authority
- last term measures final state deviation
- Q, R set relative weights of state deviation and input usage
- $R \succ 0$ means any (non-zero) input adds cost to J

LQR Problem: find $u^\star$ that minimizes $J(u)$.

Problem 2. (LQR as an NLP.) Write out the LQR problem as an NLP. In particular a least squares problem.

Solution.