Constrained Optimization

Introduction

- Hunting for a solution in *n*-dimensional space
  - Linear systems
  - Linear Least Squares
  - Nonlinear Unconstrained Optimization
  - Constrained optimization
- In addition to objective function, we have constraints
  - Point that minimizes objective function may not satisfy constraints
  - Among all points that satisfy constraints (are “feasible”) find the best one
  - Alternately, find the one that is locally best
- Solutions may not exist
  - Constraints may be too restrictive

Remarks

- Usually harder to solve than the unconstrained case
- Sometimes, it is possible to transform the problem, so that it can be converted to unconstrained case
  - Problem 1: Find (x_1,x_2) that minimize $f_1(x_1,x_2)$ subject to $x_1^2 + x_2^2 = 1$
  - Problem 2: Find θ that minimizes $f_2(\theta)$
- More sophisticated versions of this approach
  - Reduced variable methods
  - Manifold methods
  - Use expansions in “basis functions” that satisfy constraints

Statement and Notation

$$\min_{\mathbf{x}} f(\mathbf{x})$$

$$\begin{aligned} c_i(\mathbf{x}) &= 0, \quad i \in \mathcal{E} \\ c_i(\mathbf{x}) &\geq 0, \quad i \in \mathcal{I} \end{aligned}$$

Equality constraints

Inequality constraints

where f and c_i are $\mathcal{C}^2$ functions from $\mathcal{R}^n$ into $\mathcal{R}^1$.

We say that $\mathbf{x}_{opt}$ is a solution to our problem if

- $\mathbf{x}_{opt}$ satisfies all of the constraints.

feasibility

- For some $\epsilon > 0$, if $\|\mathbf{y} - \mathbf{x}_{opt}\| \leq \epsilon$, and if $\mathbf{y}$ satisfies the constraints, then $f(\mathbf{y}) \geq f(\mathbf{x}_{opt})$.

Local optimality

In other words, $\mathbf{x}_{opt}$ is feasible and locally optimal.

Constraints may be active or inactive

Constraints that are active belong to the “active set”

KKT (Karush–Kuhn–Tucker) conditions

- Lagrangian

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \phi(\mathbf{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\mathbf{x}).$$

- KKT conditions necessary for a minimum:

$$\begin{aligned}
\boldsymbol{\nabla}_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \boldsymbol{\lambda}^*) &= \mathbf{0}, \\
c_i(\mathbf{x}^*) &= 0, \quad \forall i \in \mathcal{E}, \\
c_i(\mathbf{x}^*) &\geq 0, \quad \forall i \in \mathcal{I}, \\
\lambda_i^* &\geq 0, \quad \forall i \in \mathcal{I}, \\
\lambda_i^* c_i(\mathbf{x}^*) &= 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}.
\end{aligned}$$

Linear Programming

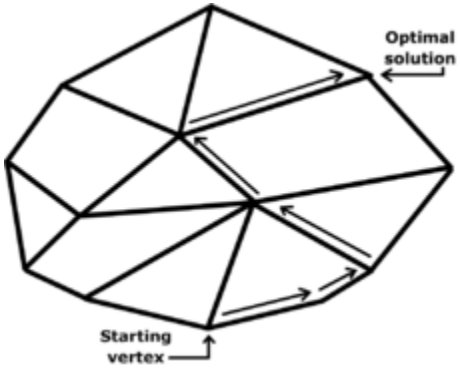
- Case 1: Linear constraints and linear f.

  This is a linear programming problem. There are two popular types of algorithm:

  - Simplex method for linear programming.
    This was the most popular up to the 1990s.
  - Interior point methods (IPMs).
    These are generally faster on large problems and remain an active area of research.

  Both of these methods are implemented in Matlab's `linprog`.

- Simplex method – constraints define a polytope in n dimensions
- A major problem in "Operations Research" and Business
- Open questions in theoretical computer science



The general case

- Case 2: General .
  Note: Unless both f and the feasible region are convex, there is no guarantee of finding the global solution. You may obtain a local solution that is far from the best.

  Some algorithmic approaches:

  – Idea 1: Reduced variable methods eliminate constraints by reducing the number of variables.

  – Idea 2: Barrier methods and interior point methods eliminate constraints through Lagrangians.

- Langrangian methods are also called “Penalty function” methods
- Example Problem

Let

$$
\begin{aligned}
f(\mathbf{x}) &= x_1^2 - 2x_1x_2 + 4x_2^2 \\
c_1(\mathbf{x}) &= x_1 + x_2 - 1 = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 1
\end{aligned}
$$

Approach 1: The feasible direction formulation

If $x_1 + x_2 = 1$, then all feasible points have the form

$$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Straight Line equation
Parametrize points with α

This formulation works because

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \alpha \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = 1$$

and all vectors **x** that satisfy the constraints have this form.

Substitute in objective function and find α

We obtain this formulation for feasible **x** by taking a particular solution

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

and adding on a linear combination of vectors that span the null space of the matrix

$$\begin{bmatrix} 1 & 1 \end{bmatrix}.$$

More general view – null space of equality constraints

The null space defines the set of feasible directions, the directions in which we can step without immediately stepping outside the feasible space.

The feasible direction approach

In general, if our constraints are $\mathbf{Ax} = \mathbf{b}$, to get feasible directions, we express $\mathbf{x}$ as

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{Zv}$$

where

- $\bar{\mathbf{x}}$ is a particular solution to the equations $\mathbf{Ax} = \mathbf{b}$ (any one will do),
- the columns of $\mathbf{Z}$ form a basis for the nullspace of $\mathbf{A}$ (any basis will do),
- $\mathbf{v}$ is an arbitrary vector of dimension $(n - m) \times 1$.

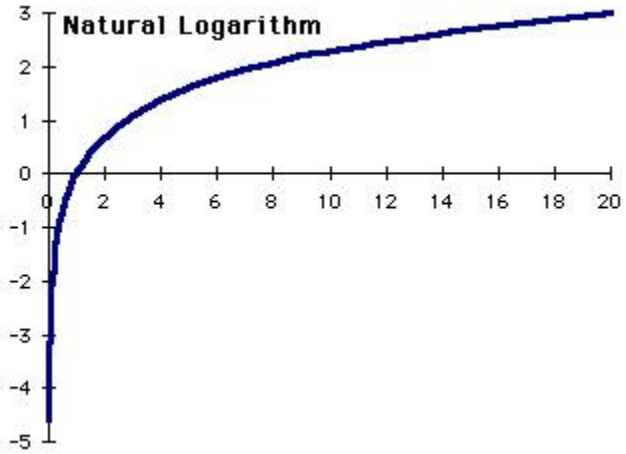
Then we have succeeded in reformulating our constrained problem as an unconstrained one with a smaller number of variables:

$$\min_{\mathbf{v}} f(\bar{\mathbf{x}} + \mathbf{Zv})$$

Lagrange multipliers

Barrier Function Method

- Suppose the constraint is an inequality constraint
- Recall behavior of the log function
  - Blows up at 0.
  - Undefined below 0
  - Negative for arguments less than 1
  - Grows slowly to infinity



$$\begin{aligned} B_\mu(\mathbf{x}) &= f(\mathbf{x}) - \mu \log c_1(\mathbf{x}) \\ &= x_1^2 - 2x_1x_2 + 4x_2^2 - \mu \log(x_1 + x_2 - 1)\,. \end{aligned}$$

If we minimize $B_\mu(\mathbf{x})$, starting from a point at which $x_1 + x_2 - 1 > 0$, and for a sufficiently small value of μ, then we will approximately solve our original problem.

- a **barrier function's v**alue increases to infinity as the argument approaches the boundary of the feasible region

Barrier methods

The problem:

$$\min_{\mathbf{x}} f(\mathbf{x})$$
$$\mathbf{c}(\mathbf{x}) \geq \mathbf{0}$$

The Lagrangian:

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \boldsymbol{\lambda}^T \mathbf{c}(\mathbf{x})$$

Log Barrier function:

$$B_\mu(\mathbf{x}) = f(\mathbf{x}) - \mu \sum_{i=1}^{m} \log(c_i(\mathbf{x}))$$

The severity of the barrier is controlled by the choice of μ:

- large μ: gradual barrier
- small μ: sharp barrier

The tradeoff

- If μ is large and the minimizer is near the boundary:
  - the barrier function looks very different from f.
- If μ is small and the minimizer is near the boundary:
  - steep gradients.
  - ill-conditioning and therefore hard to solve the problem.

Because of this, we usually solve a sequence of problems:

- Start with a large μ to guide us near the solution.
- Gradually reduce μ, using our previous solution as a starting point.

This is an important computational paradigm: we replace one hard problem (constrained minimization) by a sequence of easier problems, each of which gives a hint at the solution to the next.