Large-scale nonlinear programming

IPOPT, short for "**I**nterior **P**oint **OPT**imizer, pronounced I-P-Opt", is a software library for large scale nonlinear optimization of continuous systems.

It is written in C++ (after migrating from Fortran and C) and is released under the EPL (formerly CPL). IPOPT implements a primal-dual interior point method, and uses line searches based on Filter methods (Fletcher and Leyffer).

fmincon Algorithms

`fmincon` has five algorithm options:

- `'interior-point'` (default)
- `'trust-region-reflective'`
- `'sqp'`
- `'sqp-legacy'`
- `'active-set'`

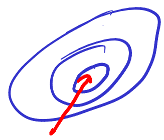
Sequential Quadratic Programming (SQP)

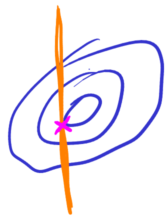
(notation from EDO)

Key Ideas:

- Like Newton's method, but with constraints
- An optimization problem with
  - Quadratic objective
  - Linear Equality Constraints

  can be solved with one matrix inversion
- Inequality constraints in the active set behave just-like equality constraints





General NLP

$$\text{minimize} \quad f(x)$$
$$\text{s.t.} \quad h(x) = 0$$
$$\cancel{g(x) \le 0}$$

$$\mathcal{L}(x,\lambda) = f(x) + \lambda^T h(x)$$

$H_{\mathcal{L}}$ Hessian of $\mathcal{L}$ w.r.t. x

J_h of h

KKT: $\nabla_x \mathcal{L} = 0$, $h(x) = 0$

Locally, near some x

$$\text{minimize} \quad \tfrac{1}{2} p^T H_{\mathcal{L}} p + \nabla_x \mathcal{L}^T p$$
$$\text{s.t.} \quad J_h p + h = 0$$

Equality - Constrained QP

$$\text{minimize} \quad x^T Q x + q^T x$$
$$\text{s.t.} \quad Ax + b = 0$$

$\Longleftrightarrow$

$$\mathcal{L}(x,\lambda) = \tfrac{1}{2} x^T Q x + q^T x + \lambda^T (Ax + b)$$

$$\text{KKT:} \begin{cases} \nabla_x \mathcal{L}(x,\lambda) = Qx + q + A^T\lambda = 0 \\ Ax + b = 0 \end{cases}$$

$$\longrightarrow \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -q \\ -b \end{bmatrix}$$

solve this to get x^*, λ^*

$$\begin{bmatrix} H_{\mathcal{L}} & J_h^T \\ J_h & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_\lambda \end{bmatrix} = \begin{bmatrix} -\nabla_x \mathcal{L} \\ -h \end{bmatrix} \quad (1)$$

SQP with equality constraints

loop

calculate $H_{\mathcal{L}}, J_h$ at x_k, λ_k

solve ① to find p_x, p_λ

$x_{k+1} \leftarrow x_k + \alpha p_x$

$\lambda_{k+1} \leftarrow \lambda_k + \alpha_k p_\lambda$

↑ chosen through line search

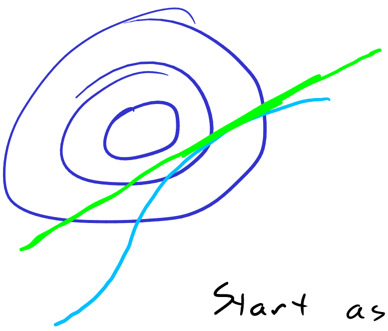
Inequality Constraints

Key: identifying active set

Local approximation of NLP Lagrangian

$$\begin{aligned} \underset{s}{\text{minimize}} \quad & \tfrac{1}{2} s^T H_{\mathcal{L}} s + \nabla_x \mathcal{L}^T s \\ \text{s.t.} \quad & J_h s + h = 0 \\ & J_g s + g \leq 0 \end{aligned}$$

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \tfrac{1}{2} x^T Q x + q^T x \\ \text{s.t.} \quad & Ax + b = 0 \\ & Cx + d \leq 0 \end{aligned}$$



W_k = working set

approximation of active set

$C_w x + d_w = 0$

↑ active set rows

$C_n x + d_n < 0$

↑ inactive set rows

Start assuming that we have x_k that satisfies

$$A x_k + b = 0$$
$$C_w x_k + d_w = 0$$
$$C_n x_k + d_n < 0$$

$$\begin{aligned} \rightarrow \quad \underset{p}{\text{minimize}} \quad & \tfrac{1}{2}(x_k + p)^T Q (x_k + p) + q^T (x_k + p) \\ \text{s.t.} \quad & A(\cancel{x_k} + p) + \cancel{b} = 0 \\ & C_w(\cancel{x_k} + p) \cancel{+ d_w} = 0 \end{aligned}$$

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2} p^T Q p + q^T p \\ \text{s.t.} \quad & Ap = 0 \\ & C_w p = 0 \end{aligned}$$

$$\begin{bmatrix} Q & A^T & C_w^T \\ A & 0 & 0 \\ C_w & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ \lambda \\ \sigma \end{bmatrix} = \begin{bmatrix} -q - Q^T x_k \\ 0 \\ 0 \end{bmatrix}$$

If $C_n(\vec{x}_k + \vec{p}) + \vec{d}_n \le 0$

$\vec{x}_{k+1} \leftarrow \vec{x}_k + \vec{p}$

else

$\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha \vec{p}$

how to choose

note $Ax + b = 0$

$C_w x + d_w = 0$ are satisfied $\forall \alpha$

$c_i \equiv i_{th}$ row of C_n

$c_i^T(\vec{x}_k + \alpha\vec{p}) + d_i \le 0 \quad \forall\, i$ in inactive set (n)

$$\alpha \underbrace{c_i^T\vec{p}}_{\text{positive if interesting}} \le -\underbrace{(c_i^T\vec{x}_k + d_i)}_{\text{always negative}}$$

$$\alpha_i \le -\frac{c_i^T\vec{x}_k + d_i}{c_i^T\vec{p}}$$

If $\alpha_i < 1$, i is a blocking constraint

add i to w

If $p \approx 0$

if $\sigma \le 0$ ← constraints in w that are no longer active

remove constraint with most negative σ from w

else

optimal solution!

Algorithm 5.4 Active-set solution method for an inequality constrained QP

Inputs:

Q, q, A, b, C, D: Matrices and vectors defining the QP (Eq. 5.77); Q must be positive definite

ε: Tolerance used for termination and for determining whether constraint is active

Outputs:

x^*: Optimal point

$k = 0$
$x_k = x_0$
$W_k = i$ for all i where $(c_i{}^\intercal x_k + d_i) > -\varepsilon$ and $\texttt{length}(W_k) \le n_x$ — One possible initial working set

while true **do**
  set $C_w = C_{i,*}$ and $d_w = d_i$ for all $i \in W_k$ — Select rows for working set
  Solve the KKT system (Eq. 5.81)
  if $\|p\| < \varepsilon$ **then**
    if $\sigma \ge 0$ **then** — Satisfied KKT conditions
      $x^* = x_k$
      return
    else
      $i = \operatorname{argmin} \sigma$
      $W_{k+1} = W_k \setminus \{i\}$ — Remove i from working set
      $x_{k+1} = x_k$
    end if
  else
    $\alpha = 1$ — Initialize with optimum step
    $B = \{\}$ — Blocking index
    for $i \notin W_k$ **do** — Check constraints outside of working set
      if $c_i^\intercal p > 0$ **then** — Potential blocking constraint
        $\alpha_b = \frac{-(c_i^\intercal x_k + d_i)}{c_i^\intercal p}$ — c_i is a row of C_n
        if $\alpha_b < \alpha$ **then**
          $\alpha = \alpha_b$
          $B = i$ — Save or overwrite blocking index
        end if
      end if
    end for
    $W_{k+1} = W_k \cup \{B\}$ — Add B to working set (if linearly independent)
    $x_{k+1} = x_k + \alpha p$
  end if
  $k = k + 1$
end while

How do we decide to accept new iterations?

start $\vec{x}_k$

solve QP to find $\vec{p}, \alpha$

$\vec{x}_{k+1} \leftarrow \vec{x}_k + \alpha \vec{p}$

Want to accept $\vec{x}_{k+1}$ that

1) decreases objective
2) satisfies constraints

Merit Function

$$\hat{f}(\vec{x};\mu) = f(\vec{x}) + \mu \, \|\bar{g}(\vec{x})\|_p$$ ← 1 or 2

$$\bar{g}_j(\vec{x}) = \begin{cases} h_j(\vec{x}) & \text{for equality} \\ \max(0, g_i(\vec{x})) & \text{for inequality} \end{cases}$$

how to chose μ?

Filter