Linear Programming

January 1, 2023

Linear Programming

- Canonical form,

$$\min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{c}^T\mathbf{x}, \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \text{ and } \mathbf{x} \geq 0 \tag{1}$$

  $\mathbf{A}$ is $m \times n$, $m < n$.

- Practically, a more general formulation

$$\min_{\mathbf{x}\in\mathbb{R}^n} \mathbf{c}^T\mathbf{x}, \text{ s.t. } \mathbf{b}_l \leq \mathbf{A}\mathbf{x} \leq \mathbf{b}_u \text{ and } \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \tag{2}$$

  - $\mathbf{b}_l \leq \mathbf{A}\mathbf{x} \leq \mathbf{b}_u$ are general linear constraints.
  - $\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}$ are simple bound constraints.

Linear Programming

Simplex Method: General Setup

- introduce slack and surplus variables to convert the general form (2) to canonical form (1)
- maintain index set $\mathcal{B}$ for m number of basic variables $\mathbf{x}_B$ and index set $\mathcal{N}$ for $n - m$ number of non-basis variables $\mathbf{x}_N$
- re-arrange (conceptually) the order of all variables such that $\mathbf{A}\mathbf{x} = \mathbf{b}$ becomes,

$$\left(\mathbf{B} \mid \mathbf{N} \right) \begin{pmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{pmatrix} = \mathbf{b}$$

- if $\mathbf{B}$ is nonsingular, fix $\mathbf{x}_N = 0$ and solve $\mathbf{B}\mathbf{x}_B = \mathbf{b}$ for a basic feasible solution.

Simplex Method: Step Update

- optimality conditions at iteration k:

$$\begin{aligned} \mathbf{B}_k^T \lambda_B &= \mathbf{c}_B \quad (3) \\ \mathbf{N}_k^T \lambda_B + \lambda_N &= \mathbf{c}_N \quad (4) \\ \lambda_N &\geq 0 \quad (5) \end{aligned}$$

- if $\exists q \in \mathcal{N}$ such that $\lambda_q < 0$, then there is a feasible descent direction $\mathbf{r} = -\mathbf{B}_k^{-1}\mathbf{A}_q$, where $\mathbf{A}_q$ is the q-th column of $\mathbf{A}$.
- moving along $\mathbf{r}$, $\mathbf{x}_q$ becomes positive hence q enters the set $\mathcal{B}$
- meanwhile a basic variable $\mathbf{x}_p$ hits 0 first, hence p quits the set $\mathcal{B}$ and joins to $\mathcal{N}$.
- pivot operation (p, q) updates $\mathbf{B}_{k+1}$ and $\mathbf{N}_{k+1}$ to start the iteration $k+1$

Linear Programming

Simplex Method: Performance Efficiency

- need repeatedly solve equation (3).
- maintain LU factorization $\mathbf{B}_k = \mathbf{L}_k \mathbf{U}_k$
- fast **LU** update to obtain $\mathbf{B}_{k+1} = \mathbf{L}_{k+1} \mathbf{U}_{k+1}$ after the pivot
- steepest edge method to select the enter index q.

Linear Programming

Simplex Method: Initial Feasibile Point

- introduce artificial variables $\mathbf{z}$ for all violated constraints.
- solve a phase-I feasibility problem,

$$\min_{\mathbf{x},\mathbf{z}} \mathbf{e}^T\mathbf{z}, \text{ s.t.} \quad \begin{aligned} \mathbf{A}\mathbf{x} \pm \mathbf{z} &= \mathbf{b} \\ \mathbf{x} &\geq 0, \\ \mathbf{z} &\geq 0 \end{aligned}$$

- pivoting $\mathbf{z}$ first, once $\mathbf{z}$ quits from $\mathcal{B}$, it never enters $\mathcal{B}$ again.

Linear Programming

Active Set Method: General Setup

- directly work on the general form (2)
- maintain an index set $\mathcal{W}$ of working constraints and the complement index set $\mathcal{N}$ for non-working constraints.
- working constraints $\mathbf{A}_{\mathcal{W}}\mathbf{x} = \mathbf{b}_{\mathcal{W}}$, where $\mathbf{b}_{\mathcal{W}}$ are the active bounds of these constraints.
- partition $\mathbf{x}$ to the set b of fixed variables $\mathbf{x}_b$ (equal to their bounds) and the complement set f of free variables $\mathbf{x}_f$
- rearrange (conceptually) the order of variables such that $\mathbf{Ax}$ becomes to

$$\left(\begin{array}{c|c} \mathbf{B} & \mathbf{F} \end{array} \right) \begin{pmatrix} \mathbf{x}_b \\ \mathbf{x}_f \end{pmatrix}$$

Linear Programming

Active Set Method: Search Direction $\mathbf{p}$

- $\mathbf{x}_b$ is not changed along $\mathbf{p}$, i.e. $\mathbf{p} = (0, \mathbf{p}_f)$
- looking for a null space direction wrt $\mathbf{F}_{\mathcal{W}}^T$ such that $\mathbf{F}_{\mathcal{W}}\mathbf{p}_f = 0$
- moving along $\mathbf{p}$ does not change the value of working constraints, $\mathbf{A}_{\mathcal{W}}\mathbf{x} = \mathbf{b}_{\mathcal{W}}$ still holds.
- find step length α along $\mathbf{p}$ to hit the first blocking constraint,
  - simple bound $j \in f$ is hit, add j to set b, $\mathbf{x}_j$ then becomes a fixed variable.
  - general constraint $i \in \mathcal{N}$ is hit, add i to set $\mathcal{W}$, $\mathbf{A}_i$ then becomes a working constraint.
  - there is no blocking constraint, declared the LP problem has unbounded solution.
- update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\mathbf{p}$ and relevant index sets.

Linear Programming

Active Set Method: Optimality Conditions

- stationary conditions:

$$\begin{pmatrix} \mathbf{I} & \mathbf{B}_{\mathcal{W}}^T \\ & \mathbf{F}_{\mathcal{W}}^T \end{pmatrix} \begin{pmatrix} \lambda_b \\ \lambda_{\mathcal{W}} \end{pmatrix} = \begin{pmatrix} \mathbf{c}_b \\ \mathbf{c}_f \end{pmatrix}$$

  - λ_b are the Lagrange multipliers of simple bound constraints of $\mathbf{x}_b$.
  - $\lambda_{\mathcal{W}}$ are the Lagrange multipliers of general working constraints $\mathbf{A}_{\mathcal{W}}$.
- for iteration at $\mathbf{x}_k$ to be optimal, Lagrange multipliers need be ≥ 0 for lower bounded constraints, ≤ 0 for upper bounded ones.
- otherwise, found the "worst" Lagrange multiplier,
  - for general constraint index $i \in \mathcal{W}$, if λ_i is the worst, move i from $\mathcal{W}$ to $\mathcal{N}$.
  - for simple bound index $j \in b$, if λ_j is the worst, move j from b to set f.
  - update the relevant index sets and start the $k+1$ iteration.

Active Set Method: QR Update

- maintain **QR** factorization

$$\mathbf{F}_{\mathcal{W}}^T = \underbrace{\left(\mathbf{Y} \mid \mathbf{Z} \right)}_{\mathbf{Q}} \begin{pmatrix} \mathbf{R} \\ 0 \end{pmatrix}$$

- seek a descent null space direction $\mathbf{p}_f = -\mathbf{Z}\mathbf{Z}^T\mathbf{c}_f$, $\mathbf{c}_f$ is the sub cost vector for free variables.
- $\mathbf{F}_{\mathcal{W}}$ changes whenever the working set $\mathcal{W}$ or the set b of fixed variables changes, economically update the **QR** factors, not refactoring.

Linear Programming

Active Set Method: Initial Working Set and Feasible Point

- given initial guess of $\mathbf{x}$, take slightly violated constraints to form the initial index sets $\mathcal{W}$ and b.
- solve $\mathbf{x}_f$ to satisfy $\mathbf{F}_{\mathcal{W}}\mathbf{x}_f = \mathbf{b}_{\mathcal{W}} - \mathbf{B}_{\mathcal{W}}\mathbf{x}_b$
- introduce artificial variables $\mathbf{y}$ and $\mathbf{z}$ and solve an active-set phase-I feasibility subproblem,

$$\min_{\mathbf{x},\mathbf{y},\mathbf{z}} \mathbf{e}^T\mathbf{y} + \mathbf{e}^T\mathbf{z}, \text{ s.t.} \quad \begin{aligned} \mathbf{b}_l \leq \mathbf{A}\mathbf{x} \pm \mathbf{z} \leq \mathbf{b}_u \\ \mathbf{l} \leq \mathbf{x} \pm \mathbf{y} \leq \mathbf{u}, \\ \mathbf{y} \geq 0 \\ \mathbf{z} \geq 0 \end{aligned}$$

References I

[1] David G. Luenberger: Linear and Nonlinear Programming, 2nd edition, Springer 2003

[2] Jorge Nocedal and Stephen J. Wright: Numerical Optimization, Springer, 1999

[3] Philip E. Gill, Walter Murray and Margaret H. Wright: Practical Optimization, Academic Press, 1981

[4] J.J. Forrest and J. A. Tomlin: Updated Triangular Factors of the Basis to Maintain Sparsity in the Product from Simplex Method, Mathematical Programming 2(1972) 263-278