Introduction to Practical Interior-Point Methods

Professor Yin Zhang
Department of Computational and Applied Math,
Rice University

Typed by Yuelin Li

September 2020

1 Newton's Method

Newton's method solves linear systems at each iteration. What Newton did: A formula for approximating a root of a polynomial (no derivative, no iteration, unpublished) Now standard form: solve $F(x) = 0$ for $F : \mathbb{R}^n \to \mathbb{R}^n$

$$x^{k+1} = x^k - \alpha_k F'\left(x^k\right)^{-1} F\left(x^k\right), \quad k = 1, 2, \cdots$$

Note: it is powerful if affordable.
Source of Power: Fast local convergence rate, it is able to reach higher accuracy, and demand much fewer iterations with "more costly" iterations. The quadratic convergence: $\forall k > \hat{k}$

$$\left\|x^{k+1} - x^*\right\| \leq C \left\|x^k - x^*\right\|^2$$

under standard conditions $\left(F'\left(x^*\right)^{-1}\right.$ exists, ...).
Systems: equivalent but unequal $x^p = 0, p > 0$ are equivalent systems (same root)
Pure Newton ($\alpha_k \equiv 1$)

$$x^{k+1} = \left(1 - \frac{1}{p}\right) x^k = \left(1 - \frac{1}{p}\right)^k x^1, \forall x^1$$

Behavior varies with p -value: for $\cdot p \in (0, 0.5]$: $\left\{x^k\right\}$ is divergent; for $\cdot p \in (0.5, \infty)$: $\left\{x^k\right\} \to 0$. Speed can be arbitrarily fast or arbitrarily slow. Nonlinear transformations can greatly affect Newton's behavior.

2 Linear Programming

Consider standard-form LP pair:
$\min c^T x$ $\max b^T y$ and $Ax = b$, $x \geq 0$, $A^T y + z = c$, $z \geq 0$, where $A \in \mathbb{R}^{m*n} (m < n)$. To apply Newton's method, we need square systems. The idea is adding log-barrier in objective to enforce strict feasibility.

Classic Barrier Method By Replacing inequalities by barriers with parameter $\mu > 0$:

$$\begin{array}{lll} \min & c^T x - \mu \sum \log x_i & \max \quad b^T y + \mu \sum_{\log(c - A^T y)_i} \\ \text{s.t.} & Ax = b (x > 0) & \left(A^T y < c\right) \end{array}$$

Barrier systems: element-wise vector inverse)

$$c - \mu x^{-1} - A^T y = 0, \quad b - \mu A z^{-1} = 0$$
$$Ax - b = 0 \quad z = c - A^T y$$

or

$$F_p(x, y; \mu) = 0 \quad F_d(y, z; \mu) = 0$$

Barrier method: apply Newton to barrier system with $\mu \to 0^+$ (while keeping $x > 0$ or $z > 0$).

Perturbed KKT System Introducing new variable + transform + vector product o:

$$z = \mu x^{-11} \left(x = \mu z^{-1}\right) \Longrightarrow x \circ z = \mu e$$

($x^{-1} \circ x = e$ unit vector: $e \circ v = v \circ e = v$ for all v) Both barrier systems transformed into:

$$\begin{aligned} A^T y + z - c &= 0 \\ Ax - b &= 0 \\ x \circ z &= \mu e \\ \text{or } F(x, y, z; \mu) \quad &= 0 \quad \text{(PKKT)} \end{aligned}$$

Note: $F(x, y, z; 0) = 0$ is nothing but the KKT system.

Three Equivalent Systems As $\mu \to 0$, central path leads to solution. (1) $F_p(x, y; \mu) = 0 : (x(\mu), y(\mu)) \to (x^*, y^*)$ (2) $F_d(y, z; \mu) = 0 : (y(\mu), z(\mu))_{\text{i}} \to (y^*, z^*)$ (3) $F(x, y, z; \mu) = 0 : (x(\mu), y(\mu), z(\mu)) \to (x^*, y^*, z^*)$ When following the central path closely, the 3 equivalent systems produce similar complexity results. However, the differences are that equivalent systems are unequal for Newton, Newton's method enjoys more freedom in (c). Also, it took over 20 years to shift from (a) to (c).

In barrier systems, PKKT looks nicer

$$z_i^* = \lim_{\mu \to 0} \frac{\mu}{x_i(\mu)} \text{ or } x_i^* = \lim_{\mu \to 0} \frac{\mu}{z_i(\mu)}$$

However, both F_p' and F_d' blow up as $\mu \to 0$. On the other hand, $x \circ z = \mu e$ is symmetric in the PKKT, Jacobian $F'(x, y, z; \mu)$ is independent of μ. And $F'\left(x^*, y^*, z^*; 0\right)$ is nonsingular for non-degeneracy. Note that the quadratic or superlinear convergence is possible. [5]An overall superlinear rate is impossible for barrier methods. [3] In theory, PKKT looks nicer. Considering applying the pure Newton's method:

Theorem 1. Assume LP non-degeneracy and denote the radii of the sphere of convergence for the two systems by $r_B(\mu), r(\mu)$. Then $F_B(x,y;\mu) = 0, r_B(\mu) = \Omega(\mu)$, $F(x,y,z;\mu) = 0, r(\mu) \geq c$.

2.1 Infeasible, p-d, interior-point algorithm

Infeasible, p-d, interior-point algorithm:
Initialize (x,y,z) so that $x, z > 0$
WHILE "stopping criteria" not satisfied, DO
Step 1 Choose $\sigma \in (0,1)$ and set $\mu = \sigma x^T z/n$
Step 2 Solve for $dv = (dx, dy, dz)$,

$$F'(x,y,z)[dx; dy; dz] = -F(x,y,z) + [0; 0; \mu e] \tag{1}$$

Step 3 Compute the primal and dual step-length, $\hat{\alpha}_p$ and $\hat{\alpha}_d$, to the non-negative orthant boundary. For $\tau \in (0,1)$, set $\alpha_p = \min(\tau\hat{\alpha}_p, 1), \alpha_d = \min(\tau\hat{\alpha}_d, 1)$
Step 4 Update,

$$[x; y; z] \Longleftarrow [x; y; z] + [\alpha_p dx; \alpha_d dy; \alpha_d dz] \tag{2}$$

The algorithm is practically efficient in that:
(1) starting Infeasible while keeping polynomiality [4]
(2) long steps in wide-neighborhood of central path μ can be reduced at each iteration substantially
(3) local convergence rate acceleration: $\sigma \to 0, \tau \to 1$ (at a sufficient rate) $\Longrightarrow$ fast local rate
(4) adequate numerical stability in linear system solving: relative error $\epsilon \leq 10^{-8}$ or even better. In practice, the performance $\gg$ theoretical complexity bounds.

2.2 Composite Newton and Predictor-Corrector

Classic composite Newton for $F(v) = 0$

$$\begin{aligned} \hat{v} &= v^k - F'\left(v^k\right)^{-1} F\left(v^k\right) \\ v^{k+1} &= v^k - F'\left(v^k\right)^{-1} F\left(v^k\right) - F'\left(v^k\right)^{-1} F(\hat{v}) \end{aligned}$$

Cubic convergence (under suitable conditions):

$$\left\|v^{k+1}-v^{*}\right\| \leq C\left\|v^{k}-v^{*}\right\|^{3}$$

Merotra's Predictor-Corrector for PKKT: [2]

$$\begin{aligned}\hat{v} &= v^{k}-F'\left(v^{k}\right)^{-1} F\left(v^{k}\right), \text{ compute } \mu(\hat{v}) \\ v^{k+1} &= v^{k}-\alpha_{k} F'\left(v^{k}\right)^{-1}\left(F\left(v^{k}\right)+F(\hat{v})-\mu p\right)\end{aligned}$$

Note: adaptive μ -value, cubic rate observable occasionally.
Summary:
Algorithm: PKKT based, primal-dual framework Features and effects:
•infeasible starting $cb \nearrow \quad ct \searrow$
•long-step path-following $cb \nearrow \quad ct \searrow$
•predictor-corrector$cb \nearrow \quad ct \searrow$
(proven) complexity bound : cb (practical) computing time : ct
Further discussion: How to numerically solve sparse linear systems?

3 Conic Programming

Standard-form conic program pair:

$$\min_{c \cdot x} \max \quad b \cdot y$$

s.t. $\quad \mathcal{A}(x)=b$ s.t. $\quad \mathcal{A}^{*}(y)+z=c$

$$x \in \mathcal{K} \quad z \in \mathcal{K}^{*}$$

where $\mathcal{A}$ is a linear operator with adjoint $\mathcal{A}^{*}$, and
•$\mathcal{K}$ is a proper cone: interior $\neq \varnothing$, closed, pointed, convex;
•$\mathcal{K}^{*}$ is the dual cone: $\{y \mid x \cdot y \geq 0, \forall x \in \mathcal{K}\}$, convex
•LP: $\mathcal{K}=\mathbb{R}_{+}^{n}=\mathcal{K}^{*}$, self-dual.
Optimality system: $x \in \mathcal{K}, z \in \mathcal{K}^{*}$

$$\begin{aligned}\mathcal{A}(x)=b & \quad (\text{ primal feasibility }) \\ \mathcal{A}^{*}(y)+z=c & \quad (\text{ dual feasibility }) \\ x \cdot z=0 & \quad (\text{ strong duality })\end{aligned}$$

Non-square system: 1 eqn. in strong duality, $\dim(\mathcal{K})$ needed Recall in LP: $x, z \in \mathbb{R}^n_+ = \mathcal{K} = \mathcal{K}^*$

$$x \cdot z = 0 \Longleftrightarrow x \circ z = 0 \quad (\# \text{ eqn} = n)$$

A component-wise vector product squares the system. In general, how to expand "dot product" into "vector product"?

Second-Order Cone Programming

$$\min_x c \cdot x, \quad \text{s.t. } \mathcal{A}(x) = b, \quad x \in \mathcal{K}$$

where $\mathcal{K} = \mathcal{Q}^{p_1} \times \mathcal{Q}^{p_2} \cdots \times \mathcal{Q}^{p_k}$. A single quadratic cone $(k = 1)$:

$$\mathcal{Q}^p = \left\{(x_0, \bar{x}) \in \mathbb{R} \times \mathbb{R}^{p-1} \mid x_0 \geq \|\bar{x}\|_2\right\} = (\mathcal{Q}^p)^*$$

For $x, z = (z_0, \bar{z}) \in \mathcal{Q}^p$, define vector product, then

$$x \circ z \overset{\text{def}}{=} \begin{pmatrix} x^T z \\ z_0\bar{x} + x_0\bar{z} \end{pmatrix} \qquad (e = [1; 0] \in \mathbb{R}^p)$$
$$x \cdot z = 0 \Leftrightarrow x \circ z = 0 \quad (x \circ e = e \circ x = x, \forall x).$$

Perturbed System: For $\mathcal{K} = \mathcal{Q}^{p_1} \times \cdots \times \mathcal{Q}^{p_k}$, $x = \left[\left(x_0^{(1)}; \bar{x}^{(1)}\right); \cdots \left(x_0^{(k)}; \bar{x}^{(k)}\right);\right]$

$$x \cdot z = 0 \Leftrightarrow x \circ z = 0, \text{ and } x \circ z = \mu e \Leftrightarrow \begin{bmatrix} {x^{(1)}}^T z^{(1)} \\ z_0^{(1)}\bar{x}^{(1)} + x_0^{(1)}\bar{z}^{(1)} \\ \vdots \\ {x^{(k)}}^T z^{(k)} \\ z_0^{(k)}\bar{x}^{(k)} + x_0^{(k)}\bar{z}^{(k)} \end{bmatrix} = \mu \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$$

Now the strong duality expanded, and system is squared. $x \cdot z = 0 \Leftrightarrow x \circ z = 0$.

Semi-Definite Programming (SDP)

$$\min_x C \cdot X, \text{ s.t. } \mathcal{A}(X) = b, \quad X \in \mathcal{K}$$
$$\mathcal{K} = \mathcal{S}^n_+ = \left\{X \in \mathbb{R}^{n \times n} \mid X = X^T \succeq 0\right\} = \mathcal{K}^*$$

For $X, Z \in \mathcal{K}$, $\quad X \cdot Z = 0 \Leftrightarrow XZ = 0$, too many equations. The following vector product squares strong duality in $\mathcal{K}$:

$$X \circ Z \overset{\text{def}}{=} \frac{1}{2}(XZ + ZX)$$

The unit vector is the identity Id, giving perturbed equation $X \circ Z = \mu Id$.

4 Nonlinear Programming

Consider inequality constrained:

$$\begin{aligned}&\min_{x\in\mathbb{R}^n}\phi(x)\\&\text{s.t. } f_i(x)\le 0,\quad i=1,\cdots,m\end{aligned}$$

Lagrangian: $L(x,y)=\phi(x)+\sum y_i f_i(x)$,
Perturbed KKT system $F(x,y,z)=0$

$$\begin{aligned}\nabla_x L(x,y)&=0\\ f(x)+z&=0\\ y\circ z&=\mu e\end{aligned}$$

Jacobian of the perturbed KKT system:

$$F'(x,y,z)=\begin{pmatrix}\nabla^2_{xx}L(x,y) & J(x)^T & 0\\ J(x) & 0 & 1\\ 0 & Z & Y\end{pmatrix}$$

For $y,z>0, F'(x,y,z)$ is nonsingular if and only if so is

$$\nabla^2\phi(x)+\sum y_i\nabla^2 f_i(x)+J(x)^T YZ^{-1}J(x)$$

This framework should reliably solve mid-sized, smooth convex programs to a high accuracy. It can also solve some nonconvex problems. [1]

Maximum Volume Ellipsoid in Polytope Finding the largest ellipsoid, centered at x with $E\succ 0$, $\mathcal{E}(x,E)=\{v\in\mathbb{R}^n : v=x+Es, \|s\|\le 1\}$ inside a polytope: $\mathcal{P}=\left\{v\in\mathbb{R}^n : a_i^T v\le b_i, i=1:m\right\}$.
Convex program:

$$\max_{x,E}\log\det E\quad (E\succ 0)$$

s.t. $\quad a_i^T x+\|Ea_i\|_2\le b_i\quad i=1:m$
m quadratic cones in $\mathbb{R}^n$: $\|s_i\|_2\le t_i=a_i^T x-b_i, s_i=Ea_i\in\mathbb{R}^n$.

An Equivalent KKT System Eliminating E from KKT:

$$E(y)=\left(A^T\operatorname{diag}(y)A\right)^{-1/2},\quad y\in\mathbb{R}^m$$

Equivalent KKT system: for $y, z \geq 0$

$$F(x,y,z) = \begin{bmatrix} A^T \operatorname{diag}(y)h(y) \\ Ax + h(y) + z - b \\ y \circ z \end{bmatrix} = 0$$

where $h(y) = (\|E(y)a_1\|, \cdots, \|E(y)a_m\|)^T \in \Re^m$, ipm4mve (A Matlab IPM solver) is built for solving PKKT.

Performance Comparision
...

5 Remarks

Open problems in SDP, symmetric cone programming, what is the convergence of AHO-based long-step alg? (we have results for short-step.) To explore superlinear rate for other directions.

References

[1] Tsuchiya El-Bakry, Tapia and Zhang, *On formulation and theory of newton interior-point method for nonlinear programming*, JOTA **89** (1996), 507–541.

[2] Mehrotra, *On the implementation of a primaldual interior point method*, SIAM J. on Optimization **2**, 575–601.

[3] Tapia Villalobos and Zhang, *Local behavior of the newton method on two equivalent systems from lp*, JOTA **121** (2002), 239–263.

[4] Y. Zhang, *On the convergence of a class of infeasible interior-point methods*, SIAM J. on Optimization **4** (1994), 208–227.

[5] Y. Zhang and D. Tapia, *On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms*, SIAM J. on Optimization (1999).