Interior-Point Methods
and Semidefinite Programming

Yin Zhang
Rice University
SIAM Annual Meeting
Puerto Rico, July 14, 2000

Outline

- The problem
- What are interior-point methods?
- Complexity theory for convex optimization
- Narrowing the gap between theory and practice
- How practical are interior-point methods?

This presentation

- is focused on a brief overview and a few selected topics
- will inevitably omit many important topics and works

Part 1. The problem

- Constrained optimization
- Conic programming
- They are "equivalent"

Constrained Optimization (CO)

$$\min\{f(x) : x \in Q \subseteq R^n\}$$

The set Q can be defined by equalities and inequalities. Inequalities create most difficulties.

$$Q = \{x : h(x) = 0, g(x) \le 0\}$$

$$h : R^n \to R^p, g : R^n \to R^q$$

Conic Programming (CP)

$$\min\{<c,x>: Ax=b, x\in K\}$$

$$(\text{cone }: x\in K \Rightarrow tx\in K, \forall t\geq 0)$$

- CP has a linear objective function
- Constraint set Q = {affine space} $\bigcap$ {cone}
- Difficulties hidden in the cone

"Natural" Conic Programs

$$\min\{< c, x >: Ax = b, x \in K\}$$

- LP --- nonnegative orthant:

$$K = \{x \in R^n : x \geq 0\}$$

- SDP --- semidefinite matrix cone:

$$K = \{X \in R^{n\times n} : X = X^T \succeq 0\}$$

- SOCP --- second order (ice cream) cone:

$$K = \{(x,t) \in R^{n+1} : \|x\| \leq t\}$$

- Direct sums of the above

Linear Objective Function

$$\min\{ f(x) : x \in Q \subseteq \mathrm{R}^n \}$$

$$\Updownarrow$$

$$\min\{ t : f(x) \le t, x \in Q \}$$

W.L.O.G, we can assume f(x)=<c,x>, i.e.,

$$\min\{ < c, x >: x \in Q \subseteq R^n \}$$

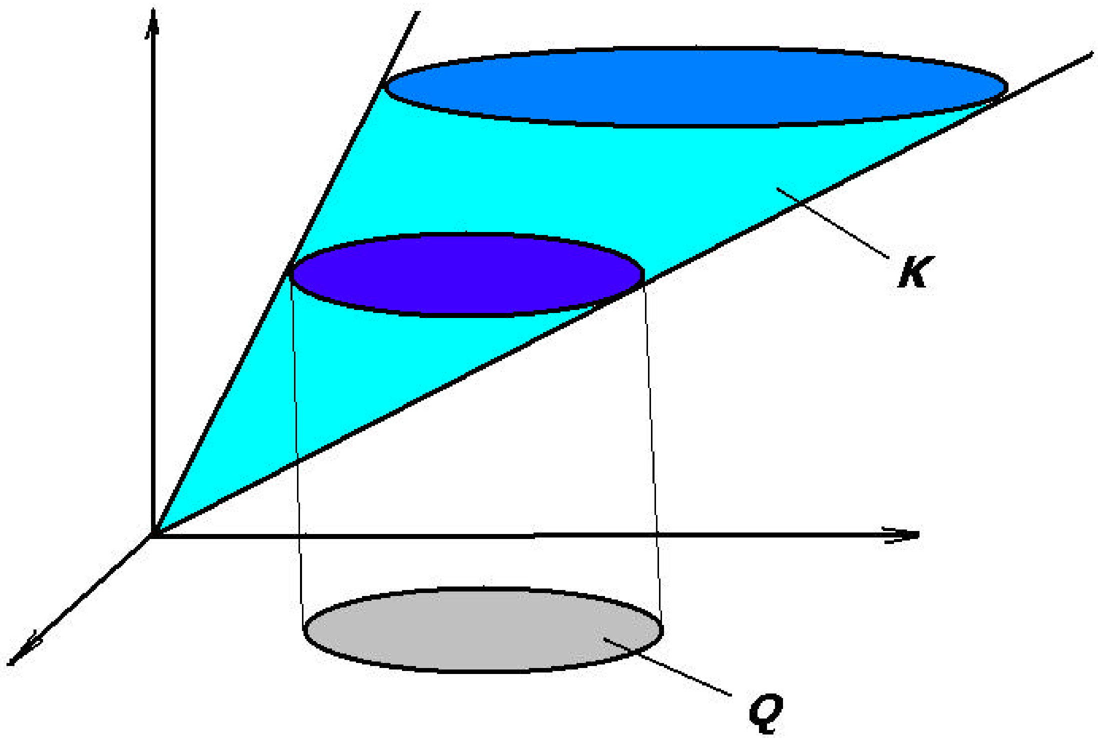
Q → Affine Space ∩ Cone

$$Q \Leftrightarrow \{(x,1) : x \in Q\} \subset R^{n+1}$$
$$\Leftrightarrow \{t(x,1) : x \in Q, t = 1\}$$
$$\Leftrightarrow K \cap \{(x,t) : t = 1\}$$

where K is the cone :

$$K \equiv \{(tx,t) : t \geq 0, x \in Q\}$$

Q → Affine Space ∩ Cone



Hence, CO = CP (theoretically convenient)

Convex or Nonconvex

$$\text{CO}: \quad \min\{f(x) : x \in Q\}$$

$$\text{CP}: \quad \min\{\langle c, x \rangle : Ax = b, x \in K\}$$

- Complexity theory exists for convex programs
  - f and Q are convex, or K is convex
- Local minima will all be global

Part 2: What are Interior-Point Methods? (IPMs)?

- Main ideas
- Classifications:
  - Primal and primal-dual methods
  - Feasible and infeasible methods

Main Ideas: Interior + Newton

CP: min{<c,x> : Ax = b, x in K }

- Keep iterates in the interior of K
- Apply Newton's method (How?)

Newton's Method

- Square Nonlinear systems:
  - F(x) = 0 (#equations = #variables)
- Unconstrained optimization:
  - $\min f(x) \Rightarrow \nabla f(x) = 0$
  - Sufficient when f is convex
- Equality constrained optimization:
  - 1^{st} order necessary (KKT) conditions form a square system F(x,y) = 0
- Inequality constrained optimization:
  - KKT → system with inequalities
  - How do we apply Newton's method?

Primal IPMs (Barrier Methods)

Consider CP: min{<c,x> : Ax=b, x in K }

- Solutions are necessarily on the boundary

- Barrier function: $F(x) : \mathrm{int}(K) \to R$

$$F(x) \to +\infty \ \text{ as } \ x \to \partial K$$

- Subproblems for t > 0 (equalities only):
  - min {<c,x> +t F(x): Ax = b } → x(t)
  - Under suitable conditions, the so-called central path x(t) exists, and as t → 0, x(t) → x*
- Newton’s method becomes applicable

Log-Barriers have a long history

- Early works on Log-barrier methods:
  - Frisch 1955 $F(x) = -\log x \ \text{ on } \ \{x > 0\}$
  - Fiacco & McCormick, 1968
  - Murray, Wright, 70s
  - Convergence results exist (e.g. x(t) → x*)
  - No computational complexity results
- Modern complexity theory for IPMs:
  - Karmarkar on LP, 1984
  - Many authors on LP, QP, LCP, SDP
  - General theory: Nesterov and Nemirovskii, 1993

Primal and Dual Conic Programs

- Primal and Dual CPs:

  (P): min { <c,x>: Ax = b, x in K }

  (D): max { <b,y>: A*y + s = c, s in K* }

where A* is the adjoint of A ,

  - < Ax, y > = < x, A*y >

and K* is the dual cone of K ,

  - $K^{*} = \{ y : \langle y, x \rangle \geq 0, \forall x \in K \}$

- Weak Duality holds: $\langle c, x \rangle \geq \langle b, y \rangle$

Primal-Dual IPMs

- They solve the primal and dual together
  - Most efficient in practice
  - First proposed for LP

  (Kojima/Mizuno/Yoshise 1990)
- They require strong duality:
  - $\langle c, x^* \rangle = \langle b, y^* \rangle$
  - Strong duality holds under reasonable conditions for the usual cones (LP, SDP, ……)

Semidefinite Programming (SDP)

$$(\mathrm{P}) \;:\; \min\{ \langle C, X \rangle : \mathcal{A}(X) = b,\, X \in K \}$$

$$(\mathrm{D}) \;:\; \max\{ \langle b, y \rangle : \mathcal{A}^*(y) + S = C,\, S \in K \}$$

$$K = K^* = \{ X \in R^{n \times n} : X = X^T \succeq 0 \}$$

- Optimization over matrix variables
- Applications:
  - Systems and Control theory, statistics
  - Structural (truss) optimization
  - Combinatorial optimization
  - …………

SDP (continued):

- Optimality conditions: X, S in K
  - Primal feasibility: $A(X) - b = 0$
  - Dual feasibility: $A^*(y) + S - C = 0$
  - Complementarity: $XS = 0 \quad + tI$
- Primal-dual methods for SDP:
  - Keep X, S in K (positive definite)
  - Perturb and apply Newton to equalities
- Keep an eye on the P-D central path (X(t),y(t),S(t))

An SDP Complication

- The system is non-square

  A(X) – b = 0, A*(y) + S – C = 0, XS – tI = 0

- Many remedies:
  - Helmberger/Rendl/Vanderbei/Wolkowisz, Kojima/Shida/Hara, Monteiro, Nesterov/Todd, Alizahde/Haeberly/Overton,……
  - Polynomial complexity bounds established
- A unification scheme: (Monteiro, YZ, 1996)

$$PXSP^{-1} + (PXSP^{-1})^T - 2tI = 0$$

IPMs: Feasible vs. Infeasible

CP: min{<c,x> : Ax = b, x in K }

We already require iterates to stay in K.

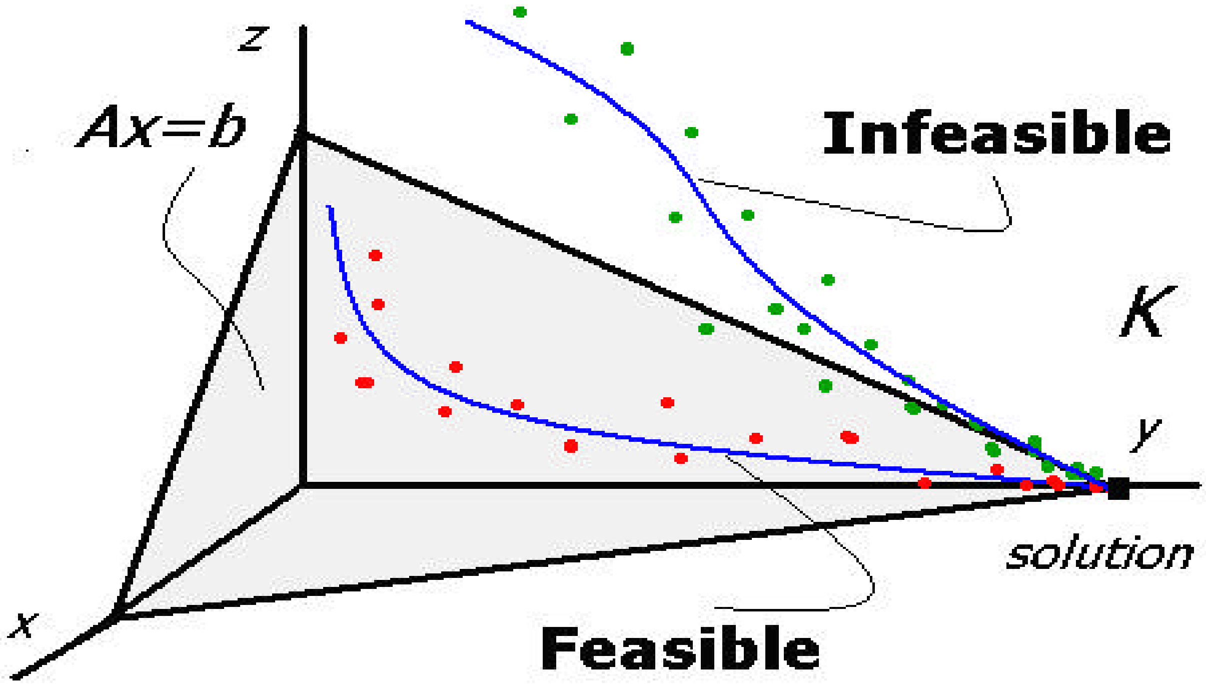
How about the affine space?

- Feasible IPMs:
  - Require iterates to stay in the affine space
  - Easier to analyze, stronger results
- Infeasible IPMs:
  - Not require iterates to stay in the affine space
  - Easier to implement, more practical

Feasible and Infeasible IPMs

CP: min{<c,x> : Ax = b, x in K }

e.g., 3x+y+2z = 1, K = {(x,y,z) >= 0}



Part 3: Complexity theory for convex programming

- Two wings make IPMs fly:
  - In theory, they work great
  - In practice, they work even better

Theory for Primal IPMs

- For $t \to 0$, apply Newton's method to:

  $\min \{ \langle c,x \rangle + t F(x) : Ax = b \}$

  - F and K hold the keys
- Q1: What barriers are good for Newton?
- Q2: What cones permit good barriers?

General theory by Nesterov & Nemirovskii:

- A1: Self-concordant barrier functions
- A2: Essentially all convex cones

  (that contain a non-empty interior but no lines)

Self-concordant Barrier Function

- Strictly convex function in interior of K :
  - The Hessian F″ varies slowly (good for Newton)

$$(F'''(x)[hhh])^2 \le 4(F''(x)[hh])^3$$

  - The gradient F′ is bounded in a special norm (implying F′ varies slowly near the central path)

$$q = \sup_{x \in \mathrm{int}(K)} \left\langle F'(x), [F''(x)]^{-1} F'(x) \right\rangle < \infty$$

- They guarantee good behavior of Newton on the function <c,x> + t F(x) for varying t

Examples

- Nonnegative Orthant:

$$K = \{x : x \geq 0\} \in R^n$$

$$F(x) = -\sum_{i=1}^{n} \log(x_i) \quad \Rightarrow \quad q = n$$

- Symmetric, positive semidefinite cone:

$$K = \{X \in R^{n \times n} : X^T = X \succeq 0\}$$

$$F(x) = -\log(\det(X)) \Rightarrow q = n$$

- Log-barriers are optimal (achieving smallest theta value possible)

Complexity Results for Primal IPMs (N&N 1993, simplified)

- Assume $x_1 \approx x(t_1)$
- Worst-case iteration number for $t < e\, t_1$:

$$O(\sqrt{q}\log e^{-1}) \text{ or } O(q \log e^{-1})$$

short-step methods long-step methods

- Different strategies exist to force t → 0
- A gap exists between theory and practice

Elegant theory has limitations

- Self-concordant barriers are not computable for general cone
- Polynomial bounds on iteration number do not necessarily mean polynomial algorithms
- A few nice cones (LP, SDP, SOCP, …) are exceptions

General Theory for Primal-Dual IPMs (Nesterov & Todd 98)

- Theory applies to symmetric cones:
  - Convex, self-dual ($K = K^*$) , homogeneous
  - Only 5 such basic symmetric cones exist
  - LP, SDP, SOCP, …, are covered
- Requires strong duality: $\langle c, x^* \rangle = \langle b, y^* \rangle$
- Same polynomial bounds on #iterations hold
- Polynomial bounds exist for #operations
- A gap still exists between theory & practice

Part 4: Narrowing the Gap Between Theory and Practice

- Infeasible algorithms
- Asymptotic complexity
  (terminology used by Ye)

Complexity of Infeasible Primal-Dual algorithms:

- All early complexity results require feasible starting points (hard to get)
- Practical algorithms only require starting points in the cone (easy)
- Can polynomial complexity be proven for infeasible algorithms?
- Affirmative answers would narrow the gap between theory and practice

The answers are indeed affirmative

- For LP: YZ 1992, also for SDP: YZ 1996
- Numerous works since 1992
- Polynomial bounds are weaker than feasible case

- There are many infeasible paths in the cone, e.g.,

$$A(X) - b = 0 \quad + t(A(X_1) - b)/t_1$$
$$A^*(y) + S - C = 0 \quad + t(A^*(y_1) + S_1 - C)/t_1$$
$$XS = 0 \quad + tI$$

Satisfied by (X_1, y_1, S_1, t_1) if $X_1 S_1 = t_1 I$

Asymptotic Complexity

- Why primal-dual algorithms are more efficient than primal ones in practice?
- Why long-step algorithms are more efficient than short-step ones in practice?
- Traditional complexity theory does not provide answers
- An answer lies in asymptotic behavior (i.e., local convergence rates)

IPMs Are Not Really Newtonian

- Nonlinear system is parameterized
- Full steps cannot be taken
- Jacobian is often singular at solutions
- Can the asymptotic convergence rate be higher than linear? Quadratic? Higher?
- Affirmative answers would explain why far less iterations taken by good algorithms than predicted by worst-case bounds
- A fast local rate accelerates convergence

Answers are all affirmative

- LP: Quadratic and higher rates attainable
  - YZ/Tapia/Dennis 92, YZ/Tapia 93, YZ/Tapia/Potra 93, Ye/Guler/Tapia/YZ 93, Mehrotra 93, YZ/D.Zhang/96, Wright/YZ 96, ……
- Extended to SDP and beyond
- IPMs can be made asymptotically close to Newton method or composite Newton methods

$$F'(x)\Delta\hat{x} = -F(x)$$
$$F'(x)\Delta x = -F(x + \Delta\hat{x})$$

- Idea: fully utilizing factorizations

Part 5: Practical Performance of IPMs

- Remarkably successful on “natural” CPs
- IPMs in Linear programming:
  - Now in every major commercial code
  - Brought an end to the Simplex era
- SDP: enabling technology
- Are there efficient interior-point algorithms for general convex programs in practice?
- How about for nonconvex programs?

Nonlinear Programming

$$\min\{f(x) : h_i(x) \le 0, i = 1,\ldots,m\}$$

$$\text{where } f, h_i : R^n \to R \text{ (possibly convex)}$$

KKT conditions form a nonlinear system with non-negativity constraints

Interior-point framework:

- Perturb and Apply Newton
- Keep iterates in the cone

KKT system and Perturbation

Optimality (KKT) conditions:

$$\nabla f(x) + \nabla h(x) y = 0$$

$$h(x) + z = 0$$

$$y \circ z = 0 \quad + te$$

$$y, z \geq 0$$

- Perturb KKT, then apply Newton
- Hopefully, (x(t),y(t),z(t)) → (x*,y*,z*)

(There is a close connection to log-barrier)

Does it work?

- General convex programming:
  - Yes, provided that derivatives are available and affordable
  - Global convergence can be established under reasonable conditions (but not poly. complexity)

    (e.g. El-Bakry/Tsuchiya/Tapia/YZ, 1992, …….)
- Nonconvex programming:
  - Yes, locally speaking

    (local optima, local convergence)
  - Continuing research (Session MS68 today)

Recent Books on IPMs

- Nesterov and Nemirovskii, "Interior Point Methods in Convex Programming", SIAM 1993
- Wright, "Primal-Dual Interior Point Methods", SIAM, 1997
- Ye, "Interior Point Algorithms: Theory and Analysis", John Wiley, 1997
- Roos/Terlaky/Vial, " Theory and Algorithms for Linear Optimization: An Interior Point Approach" , John Wiley, 1999
- Renegar, "Mathematical View of Interior Point Methods in Convex Programming", SIAM, 2000
- Also in many new linear programming books

A 15-line MATLAB code for LP

min {<c,x >: Ax = b, x >= 0 }

```
t0=cputime; [m,n]=size(A); x=sqrt(n)*ones(n,1); y=zeros(m,1);
z = x; p = symmmd(A*A'); bc = 1 + max(norm(b),norm(c));
for iter = 1:100
    Rd=A'*y+z-c; Rp=A*x-b; Rc=x.*z; residual=norm([Rd;Rp;Rc])/bc;
    fprintf('iter %2i:  residual = %9.2e',iter,residual);
    fprintf('\tobj=%14.6e\n',c'*x); if residual<5.e-8 break;end;
    gap=mean(Rc); Rc=Rc-min(.1,100*gap)*gap; d=min(5.e+15,x./z);
    B = A*sparse(1:n,1:n,d)*A'; R = cholinc(B(p,p),'inf');
    t1  = x.*Rd - Rc; t2 = -(Rp + A*(t1./z)); dy = zeros(m,1);
    dy(p)=R\(R'\t2(p)); dx=(x.*(A'*dy)+t1)./z; dz=-(z.*dx+Rc)./x;
    tau = max(.9995, 1-gap); ap = -1/min(min(dx./x),-1);
    ad = -1/min(min(dz./z),-1); ap = tau*ap; ad = tau*ad;
    x = x + ap*dx; z = z + ad*dz; y = y + ad*dy;
End
fprintf('Done!\t[m n] = [%g %g]\tCPU = %g\n',m,n,cputime-t0);
```