Convex Optimization and Applications

9 - Interior Point Methods

Guillaume Sagnol

Technische Universität Berlin

Outline

History

Interior Point Methods: Important Milestones

- Late 50's: Barrier method for nonlinear programming

History

Interior Point Methods: Important Milestones

- Late 50's: Barrier method for nonlinear programming
- 60's & 70's:
  - New, faster methods for NLP, (Augmented Lagrangian & Sequential Quadratic Programming)
  - For LPs, the simplex appears to be much better
  - So in the early 80's, everyone thinks that barrier methods have no future.

History

Interior Point Methods: Important Milestones

- Late 50's: Barrier method for nonlinear programming
- 60's & 70's:
  - New, faster methods for NLP, (Augmented Lagrangian & Sequential Quadratic Programming)
  - For LPs, the simplex appears to be much better
  - So in the early 80's, everyone thinks that barrier methods have no future.
- 1984: Karmakar publishes the first *efficient* polynomial-time algorithm for LP, based on an interior point method

History

Interior Point Methods: Important Milestones

- Late 50’s: Barrier method for nonlinear programming
- 60’s & 70’s:
  - New, faster methods for NLP, (Augmented Lagrangian & Sequential Quadratic Programming)
  - For LPs, the simplex appears to be much better
  - So in the early 80’s, everyone thinks that barrier methods have no future.
- 1984: Karmakar publishes the first *efficient* polynomial-time algorithm for LP, based on an interior point method
- 1994: Theory of self-concordance [Nesterov & Nemirovski], extension of polynomial path-following methods to conic programming problems.

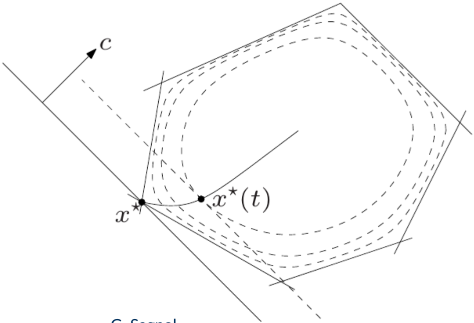
General idea: The central path

- Every convex optimization problem can be written as

$$\underset{\boldsymbol{x}\in\mathcal{X}}{\textbf{minimize}}\langle \boldsymbol{c}, \boldsymbol{x}\rangle,$$

  where $\mathcal{X}$ is a convex subset of $\mathbb{R}^n$.

- Assume $\mathcal{X}$ is equipped with a *barrier function* F:
  - $F : \textbf{int}\,\mathcal{X} \to \mathbb{R}$ is smooth and *strongly convex*;
  - $F(\boldsymbol{x}) \to \infty$ on boundary of $\mathcal{X}$.



Define the *central path*

$$\{\boldsymbol{x}^*(t) : t > 0\},$$

where:

$$\boldsymbol{x}^*(t) = \arg\min_{\boldsymbol{x}} t\langle \boldsymbol{c}, \boldsymbol{x}\rangle + F(\boldsymbol{x})$$

Strong convexity

Barrier functions are required to be twice differentiable and strongly convex:

Definition (ν-strong convexity).

For a twice differentiable convex function f, we say that f is *ν-strongly convex* for some $\nu > 0$ if

$$\nabla^2 f(\boldsymbol{x}) \succeq \nu I, \quad \forall \boldsymbol{x} \in \mathbf{dom}\, f.$$

Remark 1: definition can be extended to non-smooth functions:

$$f \ \nu\text{-strongly convex} \ :\Longleftrightarrow \ \boldsymbol{x} \mapsto f(\boldsymbol{x}) - \frac{\nu}{2}\|\boldsymbol{x}\|^2 \text{ convex.}$$

Remark 2:

$$\text{strong convexity} \implies \text{strict convexity} \implies \text{convexity.}$$

General idea: The central path

$$x^*(t) = \arg\min_{x} t\langle c, x\rangle + F(x)$$

Basic algorithm:

- Start with a small t_0 and a point $x^*(t_0)$ on the central path
- At kth iteration compute $t_k = \alpha t_{k-1}$ for some $\alpha > 1$
- Compute $x^*(t_k)$, using $x^*(t_{k-1})$ as a starting point.

General idea: The central path

$$\boldsymbol{x}^*(t) = \arg\min_{\boldsymbol{x}} t\langle \boldsymbol{c}, \boldsymbol{x}\rangle + F(\boldsymbol{x})$$

Basic algorithm:

- Start with a small t_0 and a point $\boldsymbol{x}^*(t_0)$ on the central path
- At kth iteration compute $t_k = \alpha t_{k-1}$ for some $\alpha > 1$
- Compute $\boldsymbol{x}^*(t_k)$, using $\boldsymbol{x}^*(t_{k-1})$ as a starting point.

Advandages:

- Can use Newton's method for smooth (here, twice differentiable) unconstrained optimization

General idea: The central path

$$\boldsymbol{x}^*(t) = \arg\min_{\boldsymbol{x}} t\langle \boldsymbol{c}, \boldsymbol{x}\rangle + F(\boldsymbol{x})$$

Basic algorithm:

- Start with a small t_0 and a point $\boldsymbol{x}^*(t_0)$ on the central path
- At kth iteration compute $t_k = \alpha t_{k-1}$ for some $\alpha > 1$
- Compute $\boldsymbol{x}^*(t_k)$, using $\boldsymbol{x}^*(t_{k-1})$ as a starting point.

Advandages:

- Can use Newton's method for smooth (here, twice differentiable) unconstrained optimization
- Problem becomes *harder* when t grows, but using $\boldsymbol{x}^*(t_{k-1})$ as a starting point for computing $\boldsymbol{x}^*(t_k)$ is a *good strategy*

General idea: The central path

$$x^*(t) = \arg\min_{x} t\langle c, x\rangle + F(x)$$

Basic algorithm:

- Start with a small t_0 and a point $x^*(t_0)$ on the central path
- At kth iteration compute $t_k = \alpha t_{k-1}$ for some $\alpha > 1$
- Compute $x^*(t_k)$, using $x^*(t_{k-1})$ as a starting point.

Advandages:

- Can use Newton's method for smooth (here, twice differentiable) unconstrained optimization
- Problem becomes *harder* when t grows, but using $x^*(t_{k-1})$ as a starting point for computing $x^*(t_k)$ is a *good strategy*
- In some cases (LP, SOCP, SDP, ECP), this yields a polynomial algorithm (w.r.t. <input size> and $\log \epsilon^{-1}$)

General idea: The central path

$$x^*(t) = \arg\min_{x} t\langle c, x\rangle + F(x)$$

Basic algorithm:

- Start with a small t_0 and a point $x^*(t_0)$ on the central path
- At kth iteration compute $t_k = \alpha t_{k-1}$ for some $\alpha > 1$
- Compute $x^*(t_k)$, using $x^*(t_{k-1})$ as a starting point.

Advandages:

- Can use Newton's method for smooth (here, twice differentiable) unconstrained optimization
- Problem becomes *harder* when t grows, but using $x^*(t_{k-1})$ as a starting point for computing $x^*(t_k)$ is a *good strategy*
- In some cases (LP, SOCP, SDP, ECP), this yields a polynomial algorithm (w.r.t. <input size> and $\log \epsilon^{-1}$)
- IPMs are good for both theoretical and practical purpose.

Outline

Newton method

For an iteration with current value t, we must minimize

$$f(\boldsymbol{x}) := t\langle \boldsymbol{c}, \boldsymbol{x}\rangle + F(\boldsymbol{x})$$

- Basic Newton iteration: $\boldsymbol{x}^+ \leftarrow \boldsymbol{x} - \nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$.
  - Condider quadratic approx of f around x:

    $$\hat{f}(\boldsymbol{y}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})\nabla^2 f(\boldsymbol{x})(\boldsymbol{y}-\boldsymbol{x}).$$

  - $\hat{f}(\boldsymbol{y})$ is minimized at $\boldsymbol{y}^* = \boldsymbol{x} - \left(\nabla^2 f(\boldsymbol{x})\right)^{-1} \nabla f(\boldsymbol{x})$.
  - It holds: $\hat{f}(\boldsymbol{x}) - \hat{f}(\boldsymbol{y}^*) = \frac{1}{2}\lambda(\boldsymbol{x})^2$, where

$$\lambda(\boldsymbol{x}) := \|\nabla f(\boldsymbol{x})\|_{\nabla^2 f(\boldsymbol{x})} = \sqrt{\nabla f(\boldsymbol{x})^T \; \nabla^2 f(\boldsymbol{x})^{-1} \; \nabla f(\boldsymbol{x})}$$

is called the *Newton decrement*.

Damped Newton method

In general, the Newton method can fail to converge if the starting point is not well chosen. A fix is to use

- Damped Newton iterations:

$$x^+ \leftarrow x - \delta \nabla^2 f(x)^{-1} \nabla f(x),$$

  where the step size $\delta \in [0, 1]$ can be set by line search.

- Interpretation: move in the direction of the minimizer of the local quadratic approximation of f.

Line search

In optimization, a *line search* is a procedure to select the step size of the next iteration:

At iteration k, once a search direction $-\boldsymbol{u}^{(k)}$ has been chosen, select $\delta_k > 0$ such that $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} - \delta_k \boldsymbol{u}^{(k)}$.

Different options exist:

- Exact line search: solve 1-dimensional problem at each iteration: $\delta_k := \operatorname{argmin}_{\delta > 0} f(\boldsymbol{x}^{(k)} - \delta \boldsymbol{u}^{(k)})$: *rarely used in practice.*

Line search

In optimization, a *line search* is a procedure to select the step size of the next iteration:

At iteration k, once a search direction $-\boldsymbol{u}^{(k)}$ has been chosen, select $\delta_k > 0$ such that $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} - \delta_k \boldsymbol{u}^{(k)}$.

Different options exist:

- Exact line search: solve 1-dimensional problem at each iteration: $\delta_k := \operatorname*{argmin}_{\delta > 0} f(\boldsymbol{x}^{(k)} - \delta \boldsymbol{u}^{(k)})$: *rarely used in practice*.
- Backtracking line search: Given $0 < \alpha \leq \frac{1}{2}$ and $0 < \beta < 1$, we start at $\delta = 1$, and we shrink $\delta \leftarrow \beta\delta$ until the criterion $f(\boldsymbol{x}^{(k+1)}) \leq f(\boldsymbol{x}^{(k)}) + \alpha\delta\nabla f(\boldsymbol{x}^{(k)})^T \Delta \boldsymbol{x}$ is fulfilled. *The method of choice in practice.*

Line search

In optimization, a *line search* is a procedure to select the step size of the next iteration:

At iteration k, once a search direction $-\boldsymbol{u}^{(k)}$ has been chosen, select $\delta_k > 0$ such that $\boldsymbol{x}^{(k+1)} \leftarrow \boldsymbol{x}^{(k)} - \delta_k \boldsymbol{u}^{(k)}$.

Different options exist:

- Exact line search: solve 1-dimensional problem at each iteration: $\delta_k := \underset{\delta>0}{\operatorname{argmin}} f(\boldsymbol{x}^{(k)} - \delta \boldsymbol{u}^{(k)})$: *rarely used in practice.*
- Backtracking line search: Given $0 < \alpha \leq \frac{1}{2}$ and $0 < \beta < 1$, we start at $\delta = 1$, and we shrink $\delta \leftarrow \beta\delta$ until the criterion $f(\boldsymbol{x}^{(k+1)}) \leq f(\boldsymbol{x}^{(k)}) + \alpha\delta\nabla f(\boldsymbol{x}^{(k)})^T \Delta \boldsymbol{x}$ is fulfilled. *The method of choice in practice.*
- Analytic formula: Set $\delta = \dfrac{1}{1+\lambda(\boldsymbol{x})}$: *useful for theory.*

Convergence analysis of Newton's method

- Typical behaviour:
  - Constant decrease phase: There is a constant $\gamma > 0$ such that the function decreases by at least γ at each iteration
  - Quadratic convergence phase: When $\lambda(\boldsymbol{x})$ becomes small enough, number of accurate digits doubles at each iteration !

- But... analysis depend on unknown strong convexity parameters and Lipschitz constants; Moreover these constants become worse as t grows.

- Convenient framework for the analysis: SELF-CONCORDANCE !

Self concordance

Definition (Self-concordance).

A function F is called self-concordant on $\mathcal{X}$ if it is a barrier function for $\mathcal{X}$, is C^3, and for all $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{h} \in \mathbb{R}^n$, the restriction G of F to the line $\boldsymbol{x} + t\boldsymbol{h}$, i.e. $G(t) = F(\boldsymbol{x} + t\boldsymbol{h})$ satisfies

$$|G'''(t)| \leq 2G''(t)^{3/2} \iff \left|\frac{d}{dt} G''(t)^{-1/2}\right| \leq 1, \qquad \forall t \in \mathbf{dom}\, G.$$

Remark. Note that the notion of self-concordance is not immune to *scaling*:

$$f \text{ self-concordant} \not\Rightarrow \lambda f \text{ self-concordant for } \lambda < 1.$$

Hence, problems with self-concordant functions have a *natural scale*. This allows one to make an *affine-invariant* analysis of Newton's method.

Newton's method for self-concordant f

The analysis of Newton's method applied to a self-concordant function f relies on the following propositions:

Proposition 1

f self-concordant, $\lambda(\boldsymbol{x}) \leq 0.68 \quad \Longrightarrow \quad f(\boldsymbol{x}) - p^* \leq \lambda(\boldsymbol{x})^2.$

Proposition 2

f self-concordant, $\lambda(\boldsymbol{x}) \geq 0.25$

$$\Longrightarrow \begin{cases} \boldsymbol{x}^+ \in \mathbf{int\,dom}\, f \\ f(\boldsymbol{x}) - f(\boldsymbol{x}^+) \geq (\frac{1}{4} - \log\frac{5}{4}) = 0.026856 \end{cases}$$

Proposition 3

f self-concordant, $\lambda(\boldsymbol{x}) \leq 0.25 \quad \Longrightarrow \; 2\lambda(\boldsymbol{x}^+) \leq (2\lambda(\boldsymbol{x}))^2.$

Newton's method for self-concordant f

With these three propositions, it is easy to prove:

Theorem (Convergence Analysis).

Given $x^{(0)} \in \textbf{int dom} f$, the number of damped Newton steps

$$x^{(k+1)} \leftarrow x^{(k)} + \frac{1}{1+\lambda(x^{(k)})} \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

required to find an $\epsilon-$suboptimal solution to the problem of minimizing $f(x)$ is upper bounded by

$$O(1)[f(x_0) - p^*] + \log_2 \log_2 \frac{1}{\epsilon},$$

where the hidden constant does not depend on f, and is ≤ 38.

Newton's method for self-concordant f

Proof of the convergence analysis:

- Proposition 1: $\lambda(\boldsymbol{x})^2 \leq \epsilon$ is a valid stopping criterion.

Newton's method for self-concordant f

Proof of the convergence analysis:

- Proposition 1: $\lambda(\boldsymbol{x})^2 \leq \epsilon$ is a valid stopping criterion.
- Proposition 2: There is a 1st phase during which $\lambda(\boldsymbol{x}) \geq 0.25$. Due the the *sufficient decrease* property, this phase lasts at most $\ell = \underbrace{\frac{1}{0.026856}}_{\leq 38}(f(\boldsymbol{x}_0) - p^*)$ iterations.

Newton's method for self-concordant f

Proof of the convergence analysis:

- Proposition 1: $\lambda(\boldsymbol{x})^2 \leq \epsilon$ is a valid stopping criterion.
- Proposition 2: There is a 1st phase during which $\lambda(\boldsymbol{x}) \geq 0.25$. Due the the *sufficient decrease* property, this phase lasts at most $\ell = \underbrace{\frac{1}{0.026856}}_{\leq 38}(f(\boldsymbol{x}_0) - p^*)$ iterations.
- Proposition 3: In the second phase, called *quadratic convergent phase*, we have $\lambda(\boldsymbol{x}^{(k)}) \leq 0.25$, and

$$2\lambda(\boldsymbol{x}^{(k+1)}) \leq (2\lambda(\boldsymbol{x}^{(k)}))^2 \implies \lambda(\boldsymbol{x}^{(k)})^2 \leq \left(\frac{1}{2}\right)^{2^{k-\ell+1}}, \forall k \geq \ell.$$

  Hence, the stopping criterion is reached after at most $\log_2 \log_2 1/\epsilon$ iterations of the second phase.

Proof of propositions

See Blackboard and/or Handout.

Outline

Feasible Newton Direction

Constrained equality problem: $p^* = \inf\{f(\boldsymbol{x}) : A\boldsymbol{x} = \boldsymbol{b}\}$.

To adapt Newton's method, we search the minimum of the quadratic approximation of f over the feasible affine subspace:

$$\begin{aligned} \min_{u} \quad & \hat{f}(\boldsymbol{x}+\boldsymbol{u}) := f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{u} + \frac{1}{2}\boldsymbol{u}^T \nabla^2 f(\boldsymbol{x})\boldsymbol{u} \\ s.t. \quad & A(\boldsymbol{x}+\boldsymbol{u}) = \boldsymbol{b}. \end{aligned}$$

Given feasible iterate $\boldsymbol{x}$, the KKT conditions are: $\exists \boldsymbol{\mu}$:

$$\begin{aligned} A\boldsymbol{u} = 0 \quad & \text{(primal feasibility)} \\ \nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x})\boldsymbol{u} + A^T\boldsymbol{\mu} = 0 \quad & \text{(gradient of Lagrangian vanishes).} \end{aligned}$$

Feasible Newton Direction

Given feasible iterate $\boldsymbol{x}$, the KKT conditions are: $\exists \boldsymbol{\mu}$:

$$\begin{aligned} A\boldsymbol{u} = 0 &\quad \text{(primal feasibility)} \\ \nabla f(\boldsymbol{x}) + \nabla^2 f(\boldsymbol{x})\boldsymbol{u} + A^T\boldsymbol{\mu} = 0 &\quad \text{(gradient of Lagrangian vanishes).} \end{aligned}$$

The solution of this system is the *feasible Newton direction* $\Delta\boldsymbol{x}$. In practice, we find it by solving the symmetric KKT system:

$$\begin{pmatrix} \nabla^2 f(\boldsymbol{x}^{(k)}) & A^T \\ A & 0 \end{pmatrix} \begin{bmatrix} \Delta\boldsymbol{x} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} -\nabla f(\boldsymbol{x}^{(k)}) \\ \mathbf{0} \end{bmatrix}$$

Equality Constrained Newton's method

At iteration k,

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \delta_k \Delta \boldsymbol{x},$$

where δ_k is selected by line search. The convergence analysis *is the same* as for unconstrained problems.

Equality Constrained Newton's method

At iteration k,

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \delta_k \Delta \boldsymbol{x},$$

where δ_k is selected by line search. The convergence analysis *is the same* as for unconstrained problems. Indeed,

$$A\boldsymbol{x} = \boldsymbol{b} \iff \exists \boldsymbol{z} \in \mathbb{R}^r : \boldsymbol{x} = U\boldsymbol{z} + \boldsymbol{x}_0,$$

for some matrix $U \in \mathbb{R}^{n \times r}$ and a vector $\boldsymbol{x}_0 \in \mathbb{R}^n$.

Equality Constrained Newton's method

At iteration k,

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \delta_k \Delta \boldsymbol{x},$$

where δ_k is selected by line search. The convergence analysis *is the same* as for unconstrained problems. Indeed,

$$A\boldsymbol{x} = \boldsymbol{b} \iff \exists \boldsymbol{z} \in \mathbb{R}^r : \boldsymbol{x} = U\boldsymbol{z} + \boldsymbol{x}_0,$$

for some matrix $U \in \mathbb{R}^{n \times r}$ and a vector $\boldsymbol{x}_0 \in \mathbb{R}^n$.

And it's easy to show that the equality-constrained Newton method produces the same iterates as Netwton's method for the equivalent unconstrained problem $\mathbf{minimize}_{\boldsymbol{z}} f(U\boldsymbol{z} + \boldsymbol{x}_0)$:

$$\boldsymbol{x}^{(k)} = U\boldsymbol{z}^{(k)} + \boldsymbol{x}_0, \quad \forall k.$$

Outline

Barrier functions

- When feasible region is described by convex inequalities

$$\mathcal{X} = \{\boldsymbol{x} : f_i(\boldsymbol{x}) \leq 0,\ \forall i \in [m]\}.$$

  A barrier function is $F(\boldsymbol{x}) = -\sum_{i=1}^{m} \log(-f_i(\boldsymbol{x}))$.

  - Works well in practice, but does not yield polytime algo.

- When $\mathcal{X}$ is defined by a conic inequality

$$\mathcal{X} = \{\boldsymbol{x} : A\boldsymbol{x} \succeq_K \boldsymbol{b}\}.$$

  we can use a barrier function ϕ for K, so that
  $F(\boldsymbol{x}) = \phi(A\boldsymbol{x} - \boldsymbol{b})$ is a barrier for $\mathcal{X}$.

Barrier method for convex problem with m inequality constraints

$$\begin{aligned} p^* = \inf \quad & f_0(\boldsymbol{x}) \\ s.t. \quad & f_i(\boldsymbol{x}) \leq 0 \quad (i = 1, \ldots, m) \\ & A\boldsymbol{x} = \boldsymbol{b}, \end{aligned} \tag{1}$$

Assumption: Problem strictly feasible, f_i's twice differentiable.

Barrier method for convex problem with m inequality constraints

$$\begin{aligned} p^* = \inf \quad & f_0(\boldsymbol{x}) \\ s.t. \quad & f_i(\boldsymbol{x}) \leq 0 \quad (i = 1, \ldots, m) \\ & A\boldsymbol{x} = \boldsymbol{b}, \end{aligned} \tag{1}$$

Assumption: Problem strictly feasible, f_i's twice differentiable.
Denote by $\boldsymbol{x}^*(t)$ the unique[1] optimal solution to:

$$\begin{aligned} \underset{\boldsymbol{x}}{\textbf{minimize}} \quad & F_t(\boldsymbol{x}) := tf_0(\boldsymbol{x}) + \sum_{i=1}^{m} -\log(-f_i(\boldsymbol{x})) \\ & A\boldsymbol{x} = \boldsymbol{b}, \end{aligned} \tag{Q_t}$$

[1] uniqueness guaranteed if at least one f_i is strictly convex

Barrier method for convex problem with m inequality constraints

$$
\begin{aligned}
p^* = \inf \quad & f_0(\boldsymbol{x}) \\
s.t. \quad & f_i(\boldsymbol{x}) \leq 0 \quad (i = 1, \ldots, m) \\
& A\boldsymbol{x} = \boldsymbol{b},
\end{aligned}
\tag{1}
$$

Assumption: Problem strictly feasible, f_i's twice differentiable.

Denote by $\boldsymbol{x}^*(t)$ the unique[1] optimal solution to:

$$
\begin{aligned}
\underset{\boldsymbol{x}}{\textbf{minimize}} \quad & F_t(\boldsymbol{x}) := t f_0(\boldsymbol{x}) + \sum_{i=1}^{m} -\log(-f_i(\boldsymbol{x})) \\
& A\boldsymbol{x} = \boldsymbol{b},
\end{aligned}
\tag{Q_t}
$$

Proposition

$\boldsymbol{x}^*(t)$ is feasible for (1), and $\quad f_0(\boldsymbol{x}^*(t)) - p^* \leq \dfrac{m}{t}.$

[1] uniqueness guaranteed if at least one f_i is strictly convex

Barrier method for convex problem with m inequality constraints

Proposition

$x^*(t)$ is feasible for (1), and $\quad f_0(x^*(t)) - p^* \leq \frac{m}{t}.$

- We'll only prove the counterpart of this result for conic programming problems
- Idea t large enough $\implies$ satisfactory solution.
- But problem hard to solve for large t
- Using $x(t_{k-1})$ as a starting point for computing $x(t_k)$ works well in practice, but without the affine-invariant analysis of the Newton method relying on self-concordance, we do not know if this method can be made polynomial.

Barrier method for conic problems

We turn to the study of the barrier method for conic programming problems.

$$\begin{array}{lll} \min & \boldsymbol{c}^T\boldsymbol{x} & (P) \\ s.t. & F\boldsymbol{x} = \boldsymbol{g} & \\ & A\boldsymbol{x} \succeq_K \boldsymbol{b} & \end{array} \qquad \begin{array}{lll} \max & \boldsymbol{g}^T\boldsymbol{y} + \boldsymbol{b}^T\boldsymbol{z} & (D) \\ s.t. & F^T\boldsymbol{y} + A^T\boldsymbol{z} = \boldsymbol{c} & \\ & \boldsymbol{z} \succeq_{K*} 0 & \end{array}$$

We assume that both problems are strictly feasible, which guarantees the existence of primal and dual optimal solutions:

$$p^* = \boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{g}^T\boldsymbol{y}^* + \boldsymbol{b}^T\boldsymbol{z}^*.$$

To show that the path following method yields an ϵ-suboptimal solution in polynomial time, we need an additional property on the proper cone K.

θ-normal barrier

Definition (θ-normal barrier).

ϕ is a $\theta-$normal barrier for K if it is a self-concordant barrier function over **int** K, with

$$\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta \log t, \quad \forall t > 0, \boldsymbol{x} \in \textbf{int}\, K.$$

θ-normal barrier

Definition (θ-normal barrier).

ϕ is a $\theta-$normal barrier for K if it is a self-concordant barrier function over **int** K, with

$$\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta \log t, \quad \forall t > 0, \boldsymbol{x} \in \textbf{int}\, K.$$

Examples:

- $\phi(\boldsymbol{x}) = \sum -\log x_i$ is a m-normal barrier for $K = \mathbb{R}_+^m$.

θ-normal barrier

Definition (θ-normal barrier).

ϕ is a $\theta-$normal barrier for K if it is a self-concordant barrier function over **int** K, with

$$\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta \log t, \quad \forall t > 0, \boldsymbol{x} \in \textbf{int}\, K.$$

Examples:

- $\phi(\boldsymbol{x}) = \sum -\log x_i$ is a m-normal barrier for $K = \mathbb{R}_+^m$.
- $\phi(\boldsymbol{x}) = -\log(t^2 - \|\boldsymbol{x}\|_2^2)$ is a 2-normal barrier for the Lorentz cone $K = \mathbb{L}_+^m := \{(\boldsymbol{x}, t) \in \mathbb{R}^{m-1} \times \mathbb{R} : \|\boldsymbol{x}\|_2 \leq t\}$.

θ-normal barrier

Definition (θ-normal barrier).

ϕ is a $\theta-$normal barrier for K if it is a self-concordant barrier function over **int** K, with

$$\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta \log t, \quad \forall t > 0, \boldsymbol{x} \in \textbf{int}\, K.$$

Examples:

- $\phi(\boldsymbol{x}) = \sum -\log x_i$ is a m-normal barrier for $K = \mathbb{R}_+^m$.
- $\phi(\boldsymbol{x}) = -\log(t^2 - \|\boldsymbol{x}\|_2^2)$ is a 2-normal barrier for the Lorentz cone $K = \mathbb{L}_+^m := \{(\boldsymbol{x}, t) \in \mathbb{R}^{m-1} \times \mathbb{R} : \|\boldsymbol{x}\|_2 \leq t\}$.
- $\phi(X) = -\log \det X$ is a m-normal barrier for $K = \mathbb{S}_+^m$.

θ-normal barrier

Definition (θ-normal barrier).

ϕ is a $\theta-$normal barrier for K if it is a self-concordant barrier function over **int** K, with

$$\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta \log t, \quad \forall t > 0, \boldsymbol{x} \in \textbf{int}\, K.$$

Examples:

- $\phi(\boldsymbol{x}) = \sum -\log x_i$ is a m-normal barrier for $K = \mathbb{R}_+^m$.
- $\phi(\boldsymbol{x}) = -\log(t^2 - \|\boldsymbol{x}\|_2^2)$ is a 2-normal barrier for the Lorentz cone $K = \mathbb{L}_+^m := \{(\boldsymbol{x}, t) \in \mathbb{R}^{m-1} \times \mathbb{R} : \|\boldsymbol{x}\|_2 \leq t\}$.
- $\phi(X) = -\log \det X$ is a m-normal barrier for $K = \mathbb{S}_+^m$.
- $\phi(x, y, z) = -\log(y \log z/y - x) - \log z - \log y$ is a normal barrier for $K_{\exp}$, with $\theta = 3$.

Properties of normal barriers

- If ϕ_1 is θ_1-normal for K_1 and ϕ_2 is θ_2-normal for K_2, then $\phi : (\boldsymbol{x}, \boldsymbol{y}) \mapsto \phi_1(\boldsymbol{x}) + \phi_2(\boldsymbol{y})$ is $(\theta_1 + \theta_2)$-normal for $K_1 \times K_2$.

Properties of normal barriers

- If ϕ_1 is θ_1-normal for K_1 and ϕ_2 is θ_2-normal for K_2, then $\phi : (\boldsymbol{x}, \boldsymbol{y}) \mapsto \phi_1(\boldsymbol{x}) + \phi_2(\boldsymbol{y})$ is $(\theta_1 + \theta_2)$-normal for $K_1 \times K_2$.

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \dfrac{1}{t}\nabla\phi(\boldsymbol{x});$
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta;$
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*;$

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 1:

Differentiate $\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta\log(t)$ w.r.t. $\boldsymbol{x}$:

$$t\nabla\phi(t\boldsymbol{x}) = \nabla\phi(\boldsymbol{x}).$$

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 2:

Differentiate $\phi(t\boldsymbol{x}) = \phi(\boldsymbol{x}) - \theta\log(t)$ w.r.t. t:

$$\boldsymbol{x}^T\nabla\phi(t\boldsymbol{x}) = -\theta/t.$$

$\rightarrow$ desired result for $t = 1$.

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 3: Differentiate 1 w.r.t. t: $\nabla^2\phi(t\boldsymbol{x})\boldsymbol{x} = -\frac{1}{t^2}\nabla\phi(\boldsymbol{x})$.

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 3: Differentiate 1 w.r.t. t: $\nabla^2\phi(t\boldsymbol{x})\boldsymbol{x} = -\frac{1}{t^2}\nabla\phi(\boldsymbol{x})$.

At $t = 1$, this gives $\nabla^2\phi(\boldsymbol{x})\boldsymbol{x} = -\nabla\phi(\boldsymbol{x})$.

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 3: Differentiate 1 w.r.t. t: $\nabla^2\phi(t\boldsymbol{x})\boldsymbol{x} = -\frac{1}{t^2}\nabla\phi(\boldsymbol{x})$.

At $t = 1$, this gives $\nabla^2\phi(\boldsymbol{x})\boldsymbol{x} = -\nabla\phi(\boldsymbol{x})$.
Combining with 2 yields $\boldsymbol{x}^T\nabla^2\phi(\boldsymbol{x})\boldsymbol{x} = \theta$.

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 3: Differentiate 1 w.r.t. t: $\nabla^2\phi(t\boldsymbol{x})\boldsymbol{x} = -\frac{1}{t^2}\nabla\phi(\boldsymbol{x})$.

At $t = 1$, this gives $\nabla^2\phi(\boldsymbol{x})\boldsymbol{x} = -\nabla\phi(\boldsymbol{x})$.

Combining with 2 yields $\boldsymbol{x}^T\nabla^2\phi(\boldsymbol{x})\boldsymbol{x} = \theta$.

Then, $\theta = \boldsymbol{x}^T\nabla^2\phi(\boldsymbol{x})\boldsymbol{x} = \underbrace{\boldsymbol{x}^T\nabla^2\phi(\boldsymbol{x})}_{=-\nabla\phi(\boldsymbol{x})^T}(\nabla^2\phi(\boldsymbol{x}))^{-1}\underbrace{\nabla^2\phi(\boldsymbol{x})\boldsymbol{x}}_{=-\nabla\phi(\boldsymbol{x})}$

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 4: Let $\boldsymbol{y} \in \textbf{int}\, K$

Convexity of ϕ: $\phi(\boldsymbol{y}) \geq \phi(t\boldsymbol{x}) + \langle\nabla\phi(t\boldsymbol{x}), \boldsymbol{y} - t\boldsymbol{x}\rangle$.

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 4: Let $\boldsymbol{y} \in \textbf{int}\, K$

Convexity of ϕ: $\phi(\boldsymbol{y}) \geq \phi(t\boldsymbol{x}) + \langle\nabla\phi(t\boldsymbol{x}), \boldsymbol{y} - t\boldsymbol{x}\rangle$.

Combining with 1 and 2: $\phi(\boldsymbol{y}) \geq \phi(\boldsymbol{x}) - \theta\log(t) + \frac{1}{t}\nabla\phi(\boldsymbol{x})^T\boldsymbol{y} + \theta$.

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 4: Let $\boldsymbol{y} \in \textbf{int}\, K$

Convexity of ϕ: $\phi(\boldsymbol{y}) \geq \phi(t\boldsymbol{x}) + \langle\nabla\phi(t\boldsymbol{x}), \boldsymbol{y} - t\boldsymbol{x}\rangle$.

Combining with 1 and 2: $\phi(\boldsymbol{y}) \geq \phi(\boldsymbol{x}) - \theta\log(t) + \frac{1}{t}\nabla\phi(\boldsymbol{x})^T\boldsymbol{y} + \theta$.

This implies $\nabla\phi(\boldsymbol{x})^T\boldsymbol{y} < 0$ (otherwise contradiction for $t \to 0$)

Properties of normal barriers

Proposition

Let ϕ be $\theta-$normal for proper cone K, $t > 0$, $\boldsymbol{x} \in \textbf{int}\, K$. Then,

1. $\nabla\phi(t\boldsymbol{x}) = \frac{1}{t}\nabla\phi(\boldsymbol{x})$;
2. $\boldsymbol{x}^T\nabla\phi(\boldsymbol{x}) = -\theta$;
3. $\nabla\phi(\boldsymbol{x})^T\nabla^2\phi(x)^{-1}\nabla\phi(\boldsymbol{x}) = \theta$
4. $-\nabla\phi(\boldsymbol{x}) \in \textbf{int}\, K^*$;

Proof of 4: Let $\boldsymbol{y} \in \textbf{int}\, K$

Convexity of ϕ: $\phi(\boldsymbol{y}) \geq \phi(t\boldsymbol{x}) + \langle\nabla\phi(t\boldsymbol{x}), \boldsymbol{y} - t\boldsymbol{x}\rangle$.

Combining with 1 and 2: $\phi(\boldsymbol{y}) \geq \phi(\boldsymbol{x}) - \theta\log(t) + \frac{1}{t}\nabla\phi(\boldsymbol{x})^T\boldsymbol{y} + \theta$.

This implies $\nabla\phi(\boldsymbol{x})^T\boldsymbol{y} < 0$ (otherwise contradiction for $t \to 0$)

Hence, $\forall \boldsymbol{y} \in \textbf{int}\, K, -\nabla\phi(\boldsymbol{x})^T\boldsymbol{y} > 0 \implies -\nabla\phi(\boldsymbol{x}) \in K^*$.
And we can show that $-\nabla\phi(\boldsymbol{x}) \in \partial K^*$ contradicts $\nabla^2\phi(\boldsymbol{x}) \succ 0$.

Barrier method for Conic Programming

Let K be a proper cone with a θ-normal barrier ϕ.

$$\begin{array}{llr} \min & \boldsymbol{c}^T \boldsymbol{x} & (P) \\ s.t. & F\boldsymbol{x} = \boldsymbol{g} & \\ & A\boldsymbol{x} \succeq_K \boldsymbol{b} & \end{array} \qquad \begin{array}{llr} \max & \boldsymbol{g}^T \boldsymbol{y} + \boldsymbol{b}^T \boldsymbol{z} & (D) \\ s.t. & F^T \boldsymbol{y} + A^T \boldsymbol{z} = \boldsymbol{c} & \\ & \boldsymbol{z} \succeq_{K*} 0 & \end{array}$$

Denote by $\boldsymbol{x}^*(t)$ the unique optimal solution to:

$$\begin{array}{llr} \textbf{minimize} & t\, c^T \boldsymbol{x} + \phi(A\boldsymbol{x} - \boldsymbol{b}) & (P_t) \\ s.t. & F\boldsymbol{x} = \boldsymbol{g}. & \end{array}$$

Proposition

$\boldsymbol{x}^*(t)$ is feasible for (P), and $\quad \boldsymbol{c}^T \boldsymbol{x}^*(t) - p^* \leq \dfrac{\theta}{t}.$

Barrier method for Conic Programming

Proposition

$\boldsymbol{x}^*(t)$ is feasible for (P), and $\quad \boldsymbol{c}^T \boldsymbol{x}^*(t) - p^* \leq \dfrac{\theta}{t}.$

Proof.
cf. Blackboard

Short steps

- Assume *exact*-centering is used, i.e., $\boldsymbol{x}^*(t_k)$ has been computed exactly.

Short steps

- Assume *exact*-centering is used, i.e., $\boldsymbol{x}^*(t_k)$ has been computed exactly.
- To show polynomial-time convergence, we show that if we use *short steps*, i.e., $t_{k+1} = \alpha t_k$ for some small enough $\alpha > 1$, then $\boldsymbol{x}^*(t_k)$ lies in the region of quadratic convergence for the centering step at t_{k+1}.

Short steps

- Assume *exact*-centering is used, i.e., $\boldsymbol{x}^*(t_k)$ has been computed exactly.
- To show polynomial-time convergence, we show that if we use *short steps*, i.e., $t_{k+1} = \alpha t_k$ for some small enough $\alpha > 1$, then $\boldsymbol{x}^*(t_k)$ lies in the region of quadratic convergence for the centering step at t_{k+1}.
- This ensures that $\boldsymbol{x}^*(t_{k+1})$ can be computed (to a reasonable precision ϵ within $\log_2 \log_2 \epsilon^{-1} \simeq 6$ iterations of Newton's method when $\boldsymbol{x}^*(t_k)$ is used as starting point.

Short steps

- Assume *exact*-centering is used, i.e., $\boldsymbol{x}^*(t_k)$ has been computed exactly.
- To show polynomial-time convergence, we show that if we use *short steps*, i.e., $t_{k+1} = \alpha t_k$ for some small enough $\alpha > 1$, then $\boldsymbol{x}^*(t_k)$ lies in the region of quadratic convergence for the centering step at t_{k+1}.
- This ensures that $\boldsymbol{x}^*(t_{k+1})$ can be computed (to a reasonable precision ϵ within $\log_2 \log_2 \epsilon^{-1} \simeq 6$ iterations of Newton's method when $\boldsymbol{x}^*(t_k)$ is used as starting point.

Proposition

Let $\boldsymbol{x}^*(t)$ be a solution of (P_t), and let $\omega = 1 + \dfrac{1}{4\sqrt{\theta}}$. Then,

$\lambda_{\omega t}(\boldsymbol{x}^*(t)) \leq \dfrac{1}{4}$, where λ_τ is the Newton decrement for Problem (P_τ).

Convergence Analysis with θ-normal barrier

Summary

Start with $t_0 > 0$ and a point $\boldsymbol{x}^*(t_0)$ on the central path. Then, for $k = 1, 2, \ldots,$

- $t_k = (1 + \frac{1}{4\sqrt{\theta}})t_{k-1}$;
- Compute $\boldsymbol{x}^*(t_k)$ by using a few (say ≤ 6) Newton steps to solve Problem (P_{t_k}), starting at $\boldsymbol{x}^*(t_{k-1})$;
- Stop if $\frac{\theta}{t_k} \leq \epsilon$

Convergence Analysis with θ-normal barrier

Summary

Start with $t_0 > 0$ and a point $\boldsymbol{x}^*(t_0)$ on the central path. Then, for $k = 1, 2, \ldots,$

- $t_k = (1 + \frac{1}{4\sqrt{\theta}})t_{k-1}$;
- Compute $\boldsymbol{x}^*(t_k)$ by using a few (say ≤ 6) Newton steps to solve Problem (P_{t_k}), starting at $\boldsymbol{x}^*(t_{k-1})$;
- Stop if $\frac{\theta}{t_k} \leq \epsilon$

Theorem [Nesterov & Nemirovski]

Assuming *exact centering steps*, the number of Newton steps required to get an $\epsilon-$suboptimal solution is

$$N_{\text{newton}} = O(1)\sqrt{\theta}\log\frac{\theta}{\epsilon t_0}.$$

Interior Point methods – conclusion

We just exposed the most basic version of IPM for conic programming.

Further improvements include:

- Inexact centering steps (i.e., the iterate $\boldsymbol{x}^{(k)}$ is reasonably close, but not equal to $\boldsymbol{x}^*(t_k)$);
- Large step methods (i.e., $t_k = \omega t_{k-1}$ for a large ω), still guaranteeing polytime convergence, with the help of predictor-corrector methods;
- Primal-dual interior point methods that take Newton steps simultaneously on both the primal and the dual problem for symmetric, self-dual cones.