A Brief Introduction to Interior Point Methods for Linear Programming

杨俊锋

南京大学数学系
(jfyang@nju.edu.cn)

October 12, 2014

Outline

Outline

Introduction

- Complexity theory is the foundation of computer algorithms.
- The goal of complexity theory is to develop criteria for measuring
  - the effectiveness of various algorithms;
  - the difficulty of various problems.
- The term complexity refers to the amount of resources required by a computation, e.g.,
  - running time or iteration number required to obtain an exact solution or an approximate solution of desired accuracy;
  - number of arithmetic operations.
- Linear programming (LP) plays a very important role in complexity analysis because it can be treated as
  - continuous optimization (minimization a linear objective over a convex polytope)
  - discrete (combinatorial) optimization (selecting an extreme point among a finite set of vertices)

Introduction

- The fundamental theorem of LP implies that to find an optimal solution it is sufficient to examine the vertices.
- However, the total number of vertices, though no more than C_n^m, is huge, and thus direct search is not possible in almost all cases.
- The simplex method is a procedure for examining optimal candidate vertices in an intelligent fashion.
- On average the number of vertices or iterations visited by the simplex method seems to be roughly linear in m and perhaps logarithmic in n.
- Though works very well in practice, the simplex method will indeed examine every vertex when applied to certain linear programs, which implies that the simplex method is not a polynomial time algorithm in theory, and, in the worst case, its performance can be extremely bad.

Introduction

- Klee and Minty showed in 1972 by examples that for certain linear programs the simplex method will examine every vertex. For these examples, the simplex method requires a number of steps that is exponential in the size of the problem.
- In view of Klee and Minty's result, many researchers believed that a good algorithm might be devised whose number of steps would be polynomial rather than exponential in the program's size, i.e., the time or iteration number required to compute an optimal solution would be bounded above by a polynomial in the size of the problem.

Introduction

- In 1979, based on previous works developed during the 1970s by the Soviet Union mathematicians, Khachiyan invented the so-called *ellipsoid method*.
- The method constructs a sequence of shrinking ellipsoids each of which contains the optimal solution set and each member of the sequence is smaller in volume than its predecessor by at least a certain fixed factor. Therefore, the solution set can be found to any desired degree of approximation by continuing the process.
- Khachiyan proved that the ellipsoid method is a polynomial-time algorithm for linear programming.
- However, practical experience was disappointing. In almost all cases, the simplex method was much faster than the ellipsoid method.

Introduction

- The ellipsoid method showed that polynomial time algorithms for LP do exist.
- It left open the question of whether one could be found that, in practice, was comparable with or faster than the simplex method.
- Karmarkar announced in 1984 a new polynomial time algorithm, called *interior point method* (IPM), which was said to have the potential to beat the simplex method. Karmarkar's announcement of the invention of IPM made frontpage news in major newspapers and magazines throughout the world.
- Since 1984, the research of interior point type methods remain very active for more than 20 years. IPMs had been applied to LP, SDP, SOCP, convex optimization, general nonlinear optimization, and many other areas as well.

Outline

Basic concepts of complexity theory

- Let $\mathcal{P}$ be a problem class and $\mathcal{Z}_{\mathcal{P}}$ be the data set of $\mathcal{P}$. For example, $\mathcal{Z}_{\mathcal{P}}$ and $\mathcal{P}$ for LP in standard form are, respectively, defined by

$$\begin{aligned} \mathcal{Z}_{\mathcal{P}} &= \{Z = (A, b, c) : (A, b, c) \in R^{m\times n} \times R^m \times R^n\}, \\ \mathcal{P} &= \left\{\min\{c^T x : \ s.t.\ Ax = b, x \geq 0\} : \ Z = (A, b, c) \in \mathcal{Z}_{\mathcal{P}}\right\}. \end{aligned}$$

- A member $p \in \mathcal{P}$ is called an instance of $\mathcal{P}$, which is determined by an element $Z \in \mathcal{Z}_{\mathcal{P}}$ and will be denoted by $p(Z)$ whenever necessary.

Basic concepts of complexity theory

- Corresponding to every instance $p \in \mathcal{P}$ (or $p(Z) \in \mathcal{P}$) there is an optimal solution set S_p (could be empty):
  - Linear equations $Ax = b$: $S_p = \{x : Ax = b\}$;
  - Nonlinear equations $f(x) = 0$: $S_p = \{x : f(x) = 0\}$;
  - Linear least squares $\min \|Ax - b\|^2$: $S_p = \{x : A^T Ax = A^T b\}$;
  - Linear programming:

    $$S_p = \{x : Ax = b, x \geq 0, \exists\ \lambda \text{ such that } A^T \lambda \leq c, c^T x = b^T \lambda\};$$

  - Many other problem classes:
    - Conic LP (SOCP, SDP)
    - Quadratic programming
    - Linear/nonlinear complementarity problem
    - Nonlinear programming
    - Variational inequality
    - Equilibrium problem
    - ...

Basic concepts of complexity theory

- An algorithm is a list of instructions to solve a problem class $\mathcal{P}$.
- For each instance $p \in \mathcal{P}$, an algorithm for solving $\mathcal{P}$ either determines that the optimal solution set S_p is empty or generates an *approximate solution* x (i.e., x is close to S_p in certain measure).
- Let $\mathcal{A}_{\mathcal{P}}$ be the collection of all possible algorithms for solving every instance in $\mathcal{P}$ and $A \in \mathcal{A}_{\mathcal{P}}$, i.e., A is an algorithm for solving $\mathcal{P}$.
- The *operation complexity* of an algorithm $A \in \mathcal{A}_{\mathcal{P}}$ for solving an instance $p(Z) \in \mathcal{P}$ is defined as the total arithmetic operations ($+, -, *, /$ and comparison on real numbers). Denote operation complexity by $c_o(A, Z)$.
- The *iteration complexity* of an algorithm $A \in \mathcal{A}_{\mathcal{P}}$ for solving an instance $p(Z) \in \mathcal{P}$ is defined as the total number of iterations taken. Denote it by $c_i(A, Z)$. We assume that each iteration costs a polynomial number (in problem dimension m and n) of arithmetic operations.

Worst-case complexity

- The worst-case complexity of algorithm $A \in \mathcal{A}_{\mathcal{P}}$ is defined as

$$c(A) := \sup\{c(A, Z) : Z \in \mathcal{Z}_{\mathcal{P}}\},$$

  where $c(A, Z)$ denotes certain complexity of algorithm A when applied to data Z.
- The worst-case complexity of $\mathcal{P}$ is defined as

$$c(\mathcal{P}) := \inf\{c(A) : A \in \mathcal{A}_{\mathcal{P}}\}.$$

- The analysis for both $c(A)$ and $c(\mathcal{P})$ is challenging because $\mathcal{P}$ is immense and $\mathcal{A}_{\mathcal{P}}$ is also an unknown domain for a given $\mathcal{P}$.

Turing Machine Model for computation

- Classify $\mathcal{P}$ using its data bit-size L, where the data are assumed rational. Note that an integer n needs approximately $\log_2 n$ bits on computer. Analysis of this type is called *size-based complexity*.
- Let m and n be integers representing the problem dimension.
- Suppose we have found an upper bound $f_A(m, n, L)$ (in terms of m, n and L for the size based complexity of algorithm A) which satisfies

$$c(A, L) := \sup\{c(A, Z) : \ Z \in \mathcal{Z}_{\mathcal{P}}, \text{size}(Z) \leq L\} \leq f_A(m, n, L).$$

Then, the size based complexity of problem $\mathcal{P}$ has a relation

$$c(\mathcal{P}, L) := \inf\{c(A, L) : \ A \in \mathcal{A}_{\mathcal{P}}\} \leq f_A(m, n, L),$$

which implies that the complexity of algorithms is an upper bound for the complexity of the problem class $\mathcal{P}$.

Turing Machine Model for computation

- Find such upper bound as tight as possible.
- Another direction active in computer science is the analysis of the lower bound of the complexity of the problem class $\mathcal{P}$.
- If $f_A(m, n, L)$ is a polynomial in m, n and L then we say that algorithm A is a polynomial time or polynomial algorithm and problem $\mathcal{P}$ is polynomially solvable.
- If $f_A(m, n, L)$ is independent of L and polynomial in m and n then we say that algorithm A is a strongly polynomial time algorithm.
- The existence of strongly polynomial time algorithm for LP is still open.

Real number model

- Let $\epsilon > 0$ be a parameter which controls the accuracy of a solution.
- Let $c(A, Z, \epsilon)$ be the total number of operations of algorithm A for generating an ϵ-approximate solution, with a well defined measure, to the problem determined by Z. Then

$$c(A, \epsilon) := \sup_{Z \in \mathcal{Z}_{\mathcal{P}}} c(A, Z, \epsilon) \leq f_A(m, n, \epsilon) \qquad \text{for any } \epsilon > 0,$$

  is called *error-based* complexity analysis model.
- If $f_A(m, n, \epsilon)$ is a polynomial in m, n and $\log(1/\epsilon)$, then algorithm A is called a polynomial algorithm and problem $\mathcal{P}$ is polynomially solvable. If $f_A(m, n, \epsilon)$ is independent of ϵ and polynomial in m and n then we say that algorithm A is a strongly polynomial algorithm.
- The existence of strongly polynomial time algorithm for LP in the real number model is still open.
- Besides time complexity, space complexity of algorithms is also very important.

Other types of complexity analysis - condition-based

- In the Turing Machine Model the parameters are selected as the number of variables, the number of constraints, and the bit size of the data of an instance.
- However, the difficulty of an instance is not determined by these parameters.
- Two instances with the same size may result in drastically different performances by the same algorithm.

Other types of complexity analysis

- Condition-based complexity analysis is to derive a complexity bound (arithmetic operations or iterations) related to the *condition number* of a problem.
- A famous example is to use steepest descent method to minimize a positive definite quadratic function $q(x) = 0.5x^T Qx + c^T x$. The sequence $\{x^k\}$ generated by steepest descent satisfies

$$\|x^k - x^*\|_Q \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x^0 - x^*\|_Q,$$

where x^* is the solution and $\kappa = \lambda_n/\lambda_1$ (the maximum eigenvalue of Q divided by the minimum one) is the condition number of Q.

Other types of complexity analysis

- Average complexity. Let $\mathcal{Z}_{\mathcal{P}}$ be a random sample space, then one can define the average or expected complexity of the algorithm for the problem as

$$c_{average}(A) := E_{Z \in \mathcal{Z}_{\mathcal{P}}}(c(A, Z)).$$

- Asymptotic complexity.
  - Most algorithms are iterative in nature. They generate a sequence of ever improving points $x^0, x^1, \ldots, x^k, \ldots$ approaching the solution set.
  - For many optimization problems and/or algorithms, the sequence will never exactly reach the solution set.
  - Local or asymptotic convergence analysis is concerned with *the rate at which the optimality error of the generated sequence converges to zero*.
  - Quadratic convergence, superlinear convergence, linear convergence, sublinear convergence, etc.

The simplex method is not polynomial-time

- At one time researchers believed and attempted to prove that the simplex algorithm (or some variant thereof) always requires a number of iterations that is bounded by a polynomial expression in the problem size.
- However, Klee and Minty exhibited a class of linear programs each of which requires an exponential number of iterations when solved by the conventional simplex method.
- One form of the Klee-Minty example is

$$\begin{aligned} \min \quad & \sum_{j=1}^{n} 10^{n-j} x_j \\ s.t. \quad & 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1}, \ i = 1, 2, \ldots, n \\ & x_j \geq 0, \ j = 1, 2, \ldots, n. \end{aligned}$$

- After adding slack variables, the problem is in standard form.

The simplex method is not polynomial-time

- This LP takes $2^n - 1$ pivot steps, which is equal to the number of vertices minus one (which is the starting vertex).
- For $n = 50$, we have $2^{50} - 1 \approx 10^{15}$. In a year with 365 days, there are approximately 3×10^7 seconds. If a computer ran continuously, performing a million pivots of the simplex algorithm per second, it would take approximately

$$\frac{10^{15}}{3 \times 10^7 \times 10^6} \approx 33 \text{ years}$$

to solve a problem of this class using the greedy pivot selection rule.

Outline

The ellipsoid method

- Consider a system of linear inequalities

$$\Omega = \{y \in R^m : y^T a_j \leq c_j, j = 1, 2, \ldots, n\}.$$

- Finding a point of Ω is equivalent to solving a linear programming problem (consider the optimality system).
- Two assumptions:
  1. There exist a vector $y_0 \in R^m$ and a scalar $M > 0$ such that

$$E_0 = S(y_0, M) := \{y : \|y - y_0\| \leq M\} \supset \Omega.$$

  2. If Ω is nonempty, there is a known scalar $r > 0$ such that Ω contains a ball of the form $S(y^*, r)$ with center at y^* and radius r. [1]

[1] This assumption implies that if Ω is nonempty, then it has a nonempty interior and its volume is at least $\text{vol}(S(0, r))$.

The ellipsoid method

- An ellipsoid in R^m is a set of the form

$$E = \{y \in R^m : (y - z)^T Q(y - z) \leq 1\}$$

  where $z \in R^m$ is the ellipsoid center and $Q \in S^n_{++}$ (symmetric and pd). Denote this ellipsoid by $ell(z, Q)$.
- The axes of $ell(z, Q)$ are the eigenvectors of Q and the lengths of the axes are $\lambda_i^{-1/2}$, $i = 1, 2, \ldots, m$, where λ_i's are the eigenvalues of Q. The volume of E is given by

$$vol(E) = vol(S(0, 1)) \det(Q^{-1/2}).$$

  Here $\det(\cdot)$ means determinant.
- Clearly, $E_0 = ell(y_0, \frac{1}{M^2} I)$. Thus, $vol(E_0) = vol(S(0, 1))M^m$.

The ellipsoid method

- In the ellipsoid method, a series of ellipsoids $E_k = ell(y_k, B_k^{-1})$ is defined, where B_k is symmetric and positive definite.
- At each iteration of the algorithm, $\Omega \subset E_k$ holds. It is then possible to check whether $y_k \in \Omega$. If so, we have found an element of Ω as required. If not, there is at least one constraint that is violated. Suppose $a_j^T y_k > c_j$. Then

$$\Omega \subset \frac{1}{2} E_k = \{y \in R^m : a_j^T y \leq a_j^T y_k\} \cap E_k$$

  is half of E_k, obtained by cutting the ellipsoid in half through its center y_k.
- The next ellipsoid E_{k+1} is defined to be the minimum volume ellipsoid containing $(1/2)E_k$. E_{k+1} can be constructed analytically.

The ellipsoid method

- Let $\tau = \frac{1}{m+1}$, $\delta = \frac{m^2}{m^2-1}$ and $\sigma = 2\tau$. Define

$$\begin{aligned} y_{k+1} &= y_k - \frac{\tau}{\sqrt{a_j^T B_k a_j}} B_k a_j \\ B_{k+1} &= \delta \left(B_k - \sigma \frac{B_k a_j a_j^T B_k}{a_j^T B_k a_j} \right). \end{aligned}$$

It can be shown that $E_{k+1} = ell(y_{k+1}, B_{k+1}^{-1})$ is the minimum volume ellipsoid containing $(1/2)E_k$. Moreover,

$$\frac{vol(E_{k+1})}{vol(E_k)} < \exp\left(-\frac{1}{2(m+1)}\right) < 1.$$

The ellipsoid method

- Notation: $g(x) = O(x)$ means that there exists a constant $C > 0$ such that $g(x) < Cx$.
- It is easy to show that ellipsoid method reduces the volume of an ellipsoid to one-half of its initial value in $O(m)$ iterations.
- $S(y^*, r) \subset \Omega \subset E_0$ implies that

$$vol(S(0,1))r^m \leq vol(\Omega) \leq vol(S(0,1))M^m.$$

- Simple calculation shows that in no more than $O(m^2 \log_2(M/r))$ iterations the center y_k of E_k will be in Ω, i.e., a solution to a system of linear inequalities is found.
- Roughly, a single iteration requires $O(m^2)$ arithmetic operations. Hence the entire process requires $O(m^4 \log_2(M/r))$ arithmetic operations.

Outline

The analytic center

- The new interior-point algorithms introduced by Karmarkar move by successive steps inside the feasible region.
- It is the interior of the feasible set rather than the vertices and edges that plays a dominant role in this type of algorithms.
- In fact, these algorithms purposely avoid the boundary points of the set, only eventually converging to one as a solution.
- Analytic center focuses on the interior of a set. As the name implies, the center is always away from the boundary.
- The analytic center also introduces a special structure, termed a *barrier* or *potential* that is fundamental to interior-point methods.

The analytic center

Let S be a subset of R^n and be defined by a set of equalities and inequalities, i.e.,

$$S := \{x \in R^n : h_i(x) = 0, i = 1, 2, \ldots, m_1; g_j(x) \geq 0, j = 1, 2, \ldots, m_2\},$$

where h_i and g_j are continuous functions from R^n to R. The interior of S is defined by[2]

$$S^\circ := \{x \in R^n : h_i(x) = 0, i = 1, 2, \ldots, m_1; g_j(x) > 0, j = 1, 2, \ldots, m_2\}$$

which we assume is nonempty. The following function, which is defined on S°, is called the potential function of S:

$$\psi(x) := -\sum_{j=1}^{m_2} \log g_j(x) \qquad \text{for } x \in S^\circ.$$

[2]Note that this is different from the topological interior of a set.

The analytic center

Definition (analytic center)

The analytic center of S is defined by the vector (or set of vectors) that minimizes its potential function, i.e.,

$$\arg\min_{x \in R^n} \{\psi(x) : x \in S^\circ\}.$$

Example (analytic center of a cube)

The cube $S = [0,1]^n$ can be represented by

$$S = \{x \in R^n : x_i \geq 0, 1 - x_i \geq 0, i = 1, 2, \ldots, n\}.$$

The analytic center of S is $(1/2, 1/2, \ldots, 1/2)$.

The analytic center

Remarks on the definition of analytic center:

- In general, the analytic center of a set S depends on how it is described. For example, the cube can also be represented by

$$S = \{x \in R^n : x_i \geq 0, (1 - x_i)^3 \geq 0, i = 1, 2, \ldots, n\},$$

in which case the analytic center of S becomes

$$x = (1/4, 1/4, \ldots, 1/4).$$

- Also, redundant inequalities (e.g., repeating a given inequality) can also change the location of the analytic center.
- In short, the analytic center is not uniquely determined by the geometric shape of S and thus is not “geometric center".

A Lemma

Lemma

Let $f : R^n \to R$ be a differentiable convex function, $A \in R^{m \times n}$ and $b \in R^m$. The gradient of f at a point x is given by

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)^T .$$

Then, $x^* \in R^n$ is an optimal solution of $\min_x \{f(x) : s.t.\ Ax = b\}$ if and only if there exists $y^* \in R^m$ such that

$$\begin{aligned} \nabla f(x^*) &= A^T y^* \\ Ax^* &= b, \end{aligned}$$

i.e., $\nabla \mathcal{L}(x^*, y^*) = 0$, where $\mathcal{L}(x, y) := f(x) - y^T(Ax - b)$. [3]

[3] Proof will be given in later lectures.

The analytic center

Example

The analytic center of $S = \{x \in R^n : Ax = b, x \geq 0\}$, i.e., the set of minimizers of

$$\min_x \{-\sum_{i=1}^{n} \log x_i : Ax = b, x > 0\},$$

is determined by the following system of equations:

$$\begin{aligned} Ax &= b \\ A^T y + s &= \mathbf{0} \\ x \circ s &= \mathbf{1}, \end{aligned}$$

where $y \in R^m$, $x, s \in R^n$, $\mathbf{1} = (1, 1, \ldots, 1)^T \in R^n$, and

$$x \circ s := (x_1 s_1, x_2 s_2, \ldots, x_n s_n)^T.$$

The analytic center

Example

The analytic center of $S = \{(y, s) \in R^m \times R^n : A^T y + s = c, s \geq 0\}$, i.e., the set of minimizers of

$$\min_{y,s} \{ -\sum_{i=1}^{n} \log s_i : A^T y + s = c, s > 0 \},$$

is determined by the following system of equations:

$$\begin{aligned} Ax &= 0 \\ A^T y + s &= c \\ x \circ s &= \mathbf{1}, \end{aligned}$$

where $y \in R^m$, $x, s \in R^n$, $\mathbf{1} = (1, 1, \ldots, 1)^T \in R^n$, and

$$x \circ s := (x_1 s_1, x_2 s_2, \ldots, x_n s_n)^T.$$

Primal central path

Consider a primal linear program in standard form

$$\textbf{LP} \quad p^* := \min_x \quad c^T x \\ s.t. \quad Ax = b, x \geq 0.$$

Denote the feasible region by $\mathcal{F}_p$ and assume that

$$\mathcal{F}_p^\circ = \{x \in R^n : Ax = b, x > 0\}$$

is nonempty. We also assume that the optimal solution set to **LP** is bounded. For each $\mu \geq 0$, we define a barrier problem of the form

$$\textbf{BP} \quad \min_x \quad c^T x - \mu \sum_{j=1}^n \log x_j \\ s.t. \quad Ax = b, x > 0.$$

Primal central path

- Clearly, $\mu = 0$ corresponds to the original problem **LP**.
- As $\mu \to +\infty$, the solution of **BP** approaches the analytic center of the feasible region (when $\mathcal{F}_p$ is bounded).
- As μ is varied continuously (from a positive value) toward 0, there is a path $x(\mu)$ defined by the solution to **BP**.
- The path $\{x(\mu) \in R^n : \mu \geq 0\}$ is called the primal central path.
- $\lim_{\mu \to 0+} x(\mu) =$ the analytic center of the optimal face

$$\{x \in R^n : c^T x = p^*, Ax = b, x \geq 0\}.$$

- A strategy for solving **LP** is to solve **BP** for smaller and smaller values of μ and thereby approach a solution to **LP**. This is the
basic idea of interior-point methods.

Primal central path

It can be shown that the optimality conditions for **BP** are

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ x \circ s &= \mu \mathbf{1}. \end{aligned}$$

Note that $x > 0$ and $s > 0$ must hold. Thus, x and (y, s) are, respectively, primal and dual feasible. The duality gap corresponds to this pair of primal and dual feasible points is

$$x^T s = \sum_{i=1}^{n} x_i s_i = n\mu.$$

A toy example

Consider maximizing x_1 within the unit square $[0,1]^2$. The problem is formulated as

$$\begin{aligned} \min_x \quad & -x_1 \\ s.t. \quad & x_1 + x_3 = 1 \\ & x_2 + x_4 = 1 \\ & x_i \geq 0, i = 1,2,3,4. \end{aligned}$$

The optimality conditions for $x(\mu)$ are

$$\begin{aligned} x_1 + x_3 = 1, \quad & x_2 + x_4 = 1 \\ y_1 + s_1 = -1, \quad & y_2 + s_2 = 0 \\ y_1 + s_3 = 0, \quad & y_2 + s_4 = 0 \\ x_i s_i = \mu, \quad & i = 1,2,3,4. \end{aligned}$$

(A system of 10 equations with 10 variables.)

A toy example

Direct calculations give

$$\begin{aligned} x_1(\mu) &= \frac{1 - 2\mu + \sqrt{1 + 4\mu^2}}{2} \\ x_2(\mu) &= 1/2. \end{aligned}$$

- $\lim_{\mu \to 0+} x(\mu) = (1, 1/2)^T$, which is not a corner of the cube but the analytic center of the optimal face $\{x : x_1 = 1, x_2 \in [0, 1]\}$.
- $\lim_{\mu \to +\infty} x(\mu) = (1/2, 1/2)^T$, which is the analytic center of $[0, 1]^2$.
- Hence, the central path in this case is a straight line progressing from the analytic center of the square ($\mu \to +\infty$) to the analytic center of the optimal face ($\mu \to 0+$).

Dual central path

Consider the dual linear program of **LP**:

$$\textbf{LD} \quad d^* := \max_{y,s} \quad b^T y$$
$$s.t. \quad A^T y + s = c, s \geq 0.$$

Denote the feasible region by $\mathcal{F}_d$ and assume that

$$\mathcal{F}_d^\circ = \{(y, s) \in R^m \times R^n : A^T y + s = c, s > 0\}$$

is nonempty. We also assume that the optimal solution set to **LD** is bounded. For each $\mu \geq 0$, we define a barrier problem of the form

$$\textbf{BD} \quad \max_{y,s} \quad b^T y + \mu \sum_{j=1}^n \log s_j$$
$$s.t. \quad A^T y + s = c, s > 0.$$

Dual central path

- Clearly, $\mu = 0$ corresponds to the dual problem **LD**.
- As $\mu \to +\infty$, the solution of **BD** approaches the analytic center of the feasible region (when $\mathcal{F}_d$ is bounded).
- As μ is varied continuously (from a positive value) toward 0, there is a path $\{(y(\mu), s(\mu)) \in R^m \times R^n : \mu \geq 0\}$ defined by the solution to **BD** and is called the dual central path.
- $\lim_{\mu \to 0+}(y(\mu), s(\mu)) =$ the analytic center of the optimal face

$$\{(y, s) \in R^m \times R^n : b^T y = d^*, A^T y + s = c, s \geq 0\}.$$

- A strategy for solving **LD** is to solve **BD** for smaller and smaller values of μ and thereby approach a solution to **LD**, which is the
basic idea of interior-point methods.

Dual central path

It can be shown that the optimality conditions for **BD** are

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ x \circ s &= \mu \mathbf{1}. \end{aligned}$$

These are identical to the optimality conditions for the primal central path. Note that $x > 0$ and $s > 0$ must hold. Thus, x and (y, s) are, respectively, primal and dual feasible. The duality gap corresponds to this pair of primal and dual feasible points is

$$x^T s = \sum_{i=1}^{n} x_i s_i = n\mu.$$

A toy example

Consider maximizing x_1 within the unit square $[0,1]^2$. The dual linear program is

$$\begin{array}{ll} \max\limits_{y} & y_1 + y_2 \\ s.t. & y_1 \leq -1, y_2 \leq 0. \end{array}$$

The optimality conditions for $(y(\mu), s(\mu))$ are identical to the primal central path system. Simple calculations show that

$$\begin{aligned} y_1(\mu) &= -1 - \mu / x_1(\mu) = -1 - \frac{\sqrt{1+4\mu^2} - (1-2\mu)}{2} \\ y_2(\mu) &= -2\mu. \end{aligned}$$

Clearly, it holds that $\lim_{\mu \to 0+} y(\mu) = (-1, 0)^T$, which is the unique optimal solution to the dual linear program. On the other hand, $y(\mu)$ is unbounded as $\mu \to +\infty$ because in this case the dual feasible set is itself unbounded.

Primal dual central path

Suppose the feasible region of the primal **LP** has interior points and its optimal solution set is bounded. Then, the dual feasible region also has interior points (because $s > 0$ in the optimality system of primal central path). The primal-dual central path is defined to be the set of vectors $(x(\mu), y(\mu), s(\mu))$ that satisfy the conditions

$$\begin{aligned} Ax &= b \\ A^T y + s &= c \\ x \circ s &= \mu \mathbf{1} \\ x &\geq 0 \\ s &\geq 0, \end{aligned}$$

for $0 \leq \mu \leq +\infty$. Hence the central path is defined without explicit reference to an optimization problem. It is simply defined in terms of the set of equality and inequality conditions.

Primal dual central path

The primal-dual central path can be split into two components by projecting onto the relevant space.

Theorem

Suppose the feasible sets of the primal and dual programs contain interior points. Then the following results hold

1. *The primal-dual central path* $(x(\mu), y(\mu), s(\mu))$ *exists for all* $0 \leq \mu < +\infty$*;*
2. $x(\mu)$ *is the primal central path, and* $(y(\mu), s(\mu))$ *is the dual central path;*
3. $x(\mu)$ *and* $(y(\mu), s(\mu))$ *converge to the analytic centers of the optimal primal solution and dual solution faces, respectively, as* $\mu \to 0+$*;*
4. *The duality gap* $c^T x(\mu) - b^T y(\mu)$ *is* $n\mu$*.*

Outline

Primal Barrier Method

- A direct approach is to use the barrier construction and solve the barrier problem for a very small value of μ:

$$\begin{aligned} \min \quad & c^T x - \mu \sum_{j=1}^{n} \log x_j \\ s.t. \quad & Ax = b, x \geq 0. \end{aligned}$$

  - Theoretically, to reduce the duality gap to ε it is only necessary to solve this problem with $\mu = \varepsilon/n$.
  - Unfortunately, when μ is small, the barrier problem could be highly ill-conditioned because the optimality conditions are nearly singular. This makes it difficult to directly solve the problem for small μ.

Primal Barrier Method

- A feasible strategy is to start with a moderately large μ (say $\mu = \mu_0$) and solve that problem approximately. The corresponding solution is approximately on the primal central path, but it is likely to be quite distant from the point corresponding to the limit of $\mu \to 0$.
- The (approximate) solution point at a former (larger) μ can be used as the starting point for the problem with a slightly smaller μ, because this point is likely to be close to the solution of the new problem.
- The value of μ can be reduced at each stage by a specific factor, e.g., $\mu_{k+1} = \gamma \mu_k$ for some fixed $\gamma \in (0, 1)$. Here k is the stage count.
- With this strategy, at the kth stage we have $\mu_k = \gamma^k \mu_0$. Hence to reduce μ_k/μ_0 to below ε requires $k = \frac{\log \varepsilon}{\log \gamma}$ stages.

Primal Barrier Method

Therefore, we have an algorithm with two loops:

- Outer loop: decrease μ via $\mu_k = \gamma^k \mu_0$.
- Inner loop: for each fixed μ, find an approximate solution satisfying the central path conditions

$$\begin{aligned} x \circ s &= \mu \mathbf{1} \\ Ax &= b \\ A^T y + s &= c. \end{aligned}$$

Define $X = \mathrm{diag}(x_1, x_2, \ldots, x_n)$. Note that the central path equations are equivalent to $F(x, y, s) = 0$, where $F : R^n \times R^m \times R^n \to R^n \times R^m \times R^n$ is defined by

$$F(x, y, s) = \begin{pmatrix} s - \mu X^{-1} \mathbf{1} \\ Ax - b \\ A^T y + s - c \end{pmatrix}.$$

Primal Barrier Method

From a given point (x, y, s) satisfying $x \in \mathcal{F}_p^\circ$ and $A^T y + s = c$ ($s > 0$ and/or $x \circ s = \mu \mathbf{1}$ may be violated), Newton's method moves to a closer point $x^+ = x + d_x \in \mathcal{F}_p^\circ$, where d_x is obtained via solving

$$\begin{aligned} \mu X^{-2} d_x + d_s &= \mu X^{-1} \mathbf{1} - s \\ A d_x &= 0 \\ A^T d_y + d_s &= 0, \end{aligned}$$

which is the linearized version of the central path equations, i.e., (d_x, d_y, d_s) satisfies

$$F'(x, y, s) \begin{pmatrix} d_x \\ d_y \\ d_s \end{pmatrix} = -F(x, y, s).$$

Primal Barrier Method

Three steps for solving the Newton's system:

1. solve for d_y from $(AX^2A^T)d_y = -\mu AX\mathbf{1} + AX^2 s$;
2. compute $d_s = -A^T d_y$;
3. solve for d_x from $\mu X^{-2} d_x + d_s = \mu X^{-1}\mathbf{1} - s$.

Total cost for solving each Newton system is $O(nm^2 + m^3)$:

- $O(nm^2)$ for forming AX^2A^T;
- $O(m^3)$ for solving for d_y.

Outline

Primal-Dual Path-Following Method

Assume that both $\mathcal{F}_p^\circ$ and $\mathcal{F}_d^\circ$ are nonempty. The central path can be expressed as

$$\mathcal{C} = \left\{ (x, y, s) : x \in \mathcal{F}_p^\circ, (y, s) \in \mathcal{F}_d^\circ, x \circ s = \frac{x^T s}{n} \mathbf{1} \right\}.$$

On the path we have $x \circ s = \mu \mathbf{1}$ and hence $x^T s = n\mu$. A neighborhood of the central path $\mathcal{C}$ is of the form

$$\mathcal{N}(\eta) = \left\{ (x, y, s) \;\middle|\; \begin{array}{l} x \in \mathcal{F}_p^\circ, (y, s) \in \mathcal{F}_d^\circ, \\ |x \circ s - \mu \mathbf{1}| < \eta\mu, \text{where } \mu = x^T s/n \end{array} \right\},$$

for some $\eta \in (0, 1)$. This can be thought of as a tube whose center is the central path.

Primal-Dual Path-Following Method

- The idea of the path-following method is to move within a tubular neighborhood of the central path toward the solution point.
- A suitable initial point $(x^0, y^0, s^0) \in \mathcal{N}(\eta)$ can be found by solving the barrier problem for some fixed μ_0 or from an initialization phase.
- After that, step by step moves are made, alternating between a predictor step and a corrector step.
- After each pair of steps, the point achieved is again in the fixed given neighborhood of the central path, but closer to the linear program's solution set.

Primal-Dual Path-Following Method

- The predictor step is designed to move essentially parallel to the true central path. The search direction $d = (d_x, d_y, d_s)$ is determined from the linearized version of the primal-dual central path equations, i.e.,

$$\begin{aligned} s \circ d_x + x \circ d_s &= \gamma\mu\mathbf{1} - x \circ s \\ Ad_x &= 0 \\ A^T d_y + d_s &= 0, \end{aligned}$$

  where $\gamma = 0$ is used in prediction step.
- The new point is then found by taking a step in the direction of d, as $(x^+, y^+, s^+) = (x, y, s) + \alpha(d_x, d_y, d_s)$, where $\alpha > 0$ is the step-size.
- Note that $d_x^T d_s = -d_x^T A^T d_y = 0$. Thus

$$(x^+)^T s^+ = x^T s + \alpha(d_x^T s + x^T d_s) = (1 - \alpha)x^T s,$$

  i.e., the predictor step reduces the duality gap by a factor $1 - \alpha$.

Primal-Dual Path-Following Method

- The maximum possible step-size α in that direction is made in that parallel direction without going outside of the neighborhood $\mathcal{N}(2\eta)$.
- The corrector step essentially moves perpendicular to the central path in order to get closer to it. This step moves the solution back to within the neighborhood $\mathcal{N}(\eta)$ and the step is determined by selecting $\gamma = 1$ in the Newton system with $\mu = x^T s/n$.
- This corrector step is identical to one step of the barrier method.
- The predictor-corrector method requires only one sequence of steps, each consisting of a single predictor and corrector. This contrasts with the barrier method which requires a complete sequence for each μ to get back to the central path, and then an outer sequence to reduce the μ's.

Primal-Dual Path-Following Method

- It can be shown that for any $(x, y, s) \in \mathcal{N}(\eta)$ with $\mu = x^T s/n$, the step-size in the predictor stop satisfies $\alpha \geq \frac{1}{2\sqrt{n}}$.
- After k iterations, it holds that $(x^k)^T s^k \leq (1 - \frac{1}{2\sqrt{n}})^k (x^0)^T s^0$.
- To achieve $(x^k)^T s^k / (x^0)^T s^0 \leq \varepsilon$, it is sufficient that $(1 - \frac{1}{2\sqrt{n}})^k \leq \varepsilon$.
- Thus, the iteration complexity of the primal-dual path-following method is $O(\sqrt{n}\log(1/\varepsilon))$ to achieve

$$\frac{\text{current duality gap}}{\text{initial duality gap}} \leq \varepsilon.$$

- Moreover, one can prove that the step-size $\alpha \to 1$ as $x^T s \to 0$, i.e., the duality gap reduction speed is accelerated as it becomes smaller.

Outline

Primal-Dual Potential Reduction Method

- In this method a primal-dual potential function is used to measure the solution's progress. The potential is reduced at each iteration.
- There is no restriction on either neighborhood or step-size during the iterative process as long as the potential is reduced. The greater the reduction of the potential function, the faster the convergence of the algorithm.
- Thus, from a practical point of view, potential-reduction algorithms may have an advantage over path-following algorithms where iterates are confined to lie in certain neighborhoods of the central path.

Primal-Dual Potential Reduction Method

For $x \in \mathcal{F}_p^\circ$, $(y, s) \in \mathcal{F}_d^\circ$ and $\rho \geq 0$, the primal-dual potential function is defined by

$$\psi_{n+\rho}(x, s) := (n + \rho) \log(x^T s) - \sum_{j=1}^{n} \log(x_j s_j).$$

It can be shown from the arithmetic and geometric mean inequality that

$$n \log(x^T s) - \sum_{j=1}^{n} \log(x_j s_j) \geq n \log n.$$

Then $\psi_{n+\rho}(x, s) \geq \rho(x^T s) + n \log n$. Thus, for $\rho > 0$, $\psi_{n+\rho}(x, s) \to -\infty$ implies that $x^T s \to 0$. More precisely, we have

$$x^T s \leq \exp\left(\frac{\psi_{n+\rho}(x, s) - n \log n}{\rho}\right).$$

Primal-Dual Potential Reduction Method

- The objective of this method is to drive the potential function down toward $-\infty$.
- The strategy is to apply a version of Newton's method to the primal-dual central path equations.
- In this case $\gamma = n/(n+\rho)$ is usually used, which is a combination of a predictor ($\gamma = 0$) and corrector ($\gamma = 1$) choice. This seems logical, for the predictor moves parallel to the central path toward a lower duality gap, and the corrector moves perpendicular to get close to the central path. This new method does both at once.

Primal-Dual Potential Reduction Method

- For $\rho \geq \sqrt{n}$, it can be shown that

$$\psi_{n+\rho}(x, s) - \psi_{n+\rho}(x^+, s^+) \geq \delta \geq 0.2,$$

  where $(x^+, s^+) = (x, s) + (d_x, d_s)$ and (d_x, d_s) is the Newton step.
- In practice, a line search strategy is frequently used to accelerate the reduction process of the potential function, i.e.,

$$(x^+, s^+) = (x, s) + \bar{\alpha}(d_x, d_s),$$

  (y will be updated as well) where

$$\bar{\alpha} := \arg\min_{\alpha \geq 0} \psi_{n+\rho}(x_k + \alpha d_x, s_k + \alpha d_s).$$

Primal-Dual Potential Reduction Method

- Suppose the initial point $(x^0, y^0, s^0) \in \mathcal{F}_p^\circ \times \mathcal{F}_d^\circ$ satisfies

$$\psi_{n+\rho}(x^0, s^0) \leq \rho \log(\langle s^0, x^0 \rangle) + n \log n + O(\sqrt{n} \log n).$$

- It can be shown that the algorithm terminates in at most $O(\rho \log(n/\varepsilon))$ iterations with

$$\frac{\text{current duality gap}}{\text{initial duality gap}} = \frac{\langle x^k, s^k \rangle}{\langle x^0, s^0 \rangle} \leq \varepsilon.$$

- Since this holds for any $\rho \geq \sqrt{n}$, by choosing $\rho = \sqrt{n}$, the iteration complexity bound of the primal-dual potential-reduction method becomes $O(\sqrt{n} \log(n/\varepsilon))$.

Primal-Dual Potential Reduction Method

Three steps for solving the Newton system:

1. solve for d_y from $(AS^{-1}XA^T)d_y = b - \gamma\mu AS^{-1}\mathbf{1}$;
2. compute $d_s = -A^T d_y$;
3. solve for d_x from $Sd_x + Xd_s = \gamma\mu\mathbf{1} - XS\mathbf{1}$.

The per iteration cost for solving each Newton system is $O(nm^2 + m^3)$:

1. Forming the coefficient matrix $AS^{-1}XA^T$ cost $O(m^2 n)$.
2. Solving $(AS^{-1}XA^T)d_y = b - \gamma\mu AS^{-1}\mathbf{1}$ cost m^3.

Primal-Dual Potential Reduction Method

By using an advanced rank-one technique to update the approximate inverse of $AS^{-1}XA^T$ during the iterative process, one can reduce the average number of arithmetic operations per iteration to $O(\sqrt{n}m^2)$. Thus, we have the following total arithmetic operation complexity bound to solve a linear program:

Theorem

Let $\rho = \sqrt{n}$. Then, the primal-dual potential reduction method terminates in at most $O(nm^2 \log(n/\varepsilon))$ arithmetic operations to obtain a solution satisfying

$$\begin{aligned} Ax^k &= b, \quad x^k > 0 \\ A^T y^k + s^k &= c, \quad s^k > 0 \\ \langle x^k, s^k \rangle &\leq \varepsilon \langle x^0, s^0 \rangle. \end{aligned}$$

The end

Other issues include:

- Termination
- Initialization
- Infeasible IPMs
- Homogeneous Self-Dual Algorithm

After nearly 30 years of development, now IPMs have become an important class of methods for solving LP, SOCP, SDP and convex optimization problems.

参考书


David G. Luenberger and Yinyu Ye
Linear and nonlinear programming.


Yinyu Ye,
Interior point algorithms: theory and analysis,
John Wiley and Sons, New York, 1997.